Comment on egusphere-2022-143
Anonymous Referee #2

Referee comment on "MultilayerPy (v1.0): A Python-based framework for building, running and optimising kinetic multi-layer models of aerosols and films" by Adam Milsom et al., EGUsphere, https://doi.org/10.5194/egusphere-2022-143-RC2, 2022

General Comments

This manuscript develops an open-source package which can build kinetic multi-layer models. These models are useful in studying bulk-surface interactions for particles and the interplay between chemistry and thermodynamics. The ozonolysis of oleic acid system, which has been studied with both experimental work and kinetic modeling, is used as the most extensive example. Optimization of model parameters with experimental data is also explored via MCMC sampling techniques. Overall, this manuscript is easy to follow and this code base is going to be an extremely valuable tool for the community, though I believe a few additions to the notebooks code be helpful for those less familiar with code development (which is where this manuscript has the potential to have the biggest impact).

Major Comments

This manuscript intends to reduce the workload for those attempting to use kinetics modeling, but there are parts of the notebooks which I believe to be a bit difficult for novice users (regardless of their knowledge of the Python language). I spent quite a bit of time playing with the documents, and first would like to thank the authors, as I believe this tool is going to be quite useful for many in the field moving forward. I have a few major notes to make about the code base:

- What about the case where a user doesn’t want to consider a reaction? A good example for using this model could be in the case of water condensation, which is explored in the KM-GAP case studies (3.1). It seems that all chemical components need to be
defined via the reaction scheme, but in the case of water diffusing through an organic aerosol, perhaps you are not interested in any further chemistry. One could just create a set of reactions with rate constants all equal to 0 so the chemistry doesn’t occur, but this doesn’t seem like a ‘best practice’ way to code this up. Perhaps I missed how to do this more elegantly, though, in which case it would be good to highlight this in the code or manuscript.

- In general, the units are difficult to follow. While Table A1 provides the unit description for a KM-SUB application, Table S2 provides unit values. There is a challenge in the code in identifying what the true input values are. Take parameter ‘delta_1’: it is reported as 0.8 nm but you have to input the unit as cm. If you take both tables together then you can decipher what’s happening, but I think it might be useful just to comment off the units after the parameters in the dictionary. It was just difficult for me to change parameters on a whim to representative values.

- Consider a ‘docs’ sect would be useful beyond the provided notebooks to understanding this code base.

**Minor Comments:**

Line 200: You comment that MCMC cannot be used to compare two different models. What exactly do you imply by this? MCMC in the context of this paper is used to sample the posterior distribution and could for different reaction systems extract the values of, for instance, accommodation coefficients. These coefficients are themselves comparable to the extent that the model represents the physical system. So, in that sense you can compare ‘between models’. Essentially, I am just wondering what you originally intended to say.

Further, I have a question about the Emcee package: are the final extracted parameters based on the lowest value of the cost function in the markov chain? Or is a separate optimizer run on that cost function fit to optimize the parameters? While MCMC itself tends toward the best fit parameters, it does not create the best fit parameters.

Line 262: The authors state that differences in the model performance result from a lack of time-resolved particle size information after 17 min. While this point is subtle, the differences after 17 min are because of the physical differences in the model parameters, which in part result from the optimization on only 17 min of data.

Line 301: You reference the strong correlation observed in Fig S1. I am wondering if in the 2d histogram the burn in period is removed –it appears you can see the walk as evidenced in the extremities on both the 1 and 2D histograms. It is not untypical in many applications to remove the MCMC burn-in for visualizing the posterior. With more restricted bounds it may be easier to view the correlation.

Table S2: check the labeling of [3]nonanal and [4]products and the consistency in the subscripts of the variables in the table. I think they may be switched.
In the simulate code (`simulate.py`), if you turn the save figs on, the figures save on a 0-based indexing rather than a 1-based indexing. The problem here is that for someone not privy to coding, this may be confusing in interpreting which reaction component is which (because the components are listed into the model with a 1-based indexing).

Editorial

Line 11: insert 'the' before 'particle and film level'

Line 33: replace 'were' with 'are'

Line 306: Should be 'encourage' (drop the s).