

Response to referee comments on the manuscript “PyCHAM (v1.3.4): a Python box model for simulating aerosol chambers”

The authors thank the referees for their thoughtful and helpful comments. In response we have improved the manuscript and PyCHAM programme. The changes have resulted in an updated PyCHAM programme, for which we have updated the version number to v2.0.4, which is reflected in the new manuscript title. Another author – Yunqi Shao – has been added to the author list and acknowledgement section for her contribution to the improved nucleation section of the paper.

In the table below we address each referee comment. In the response column references to lines pertain to the revised paper and line numbers are inclusive.

We provide a modification in the revised paper that addresses the natural tendency for readers to compare outputs in this description paper against observations from experiments. Lines 68-71 emphasis the purpose and limits of the presented results: “Comparison of some results presented below with experimental observations is possible, however it is beyond the scope of this paper to investigate the accuracy of chemical mechanisms or estimation methods that PyCHAM can use, rather the examples below illustrate the utility of PyCHAM to test the sensitivity of such techniques”.

Referee Comment	Response
Referee # 1	
1. I could not find any clear description of how the gas-particle partitioning of multiple condensable organic and inorganic vapors are treated in PyCHAM. Do you consider the Kelvin effect and Raoult’s law for ideal mixtures? Possible I just missed this information. I think you should add a separate section which describe more in detail how the gas-particle partitioning onto different size bins are treated and how you keep track of the chemical composition in each size bin over time.	This information was very briefly mentioned in the original draft and we agree that further detail is beneficial. Lines 302-316 now provide an explanation.
2. Page 3, L78-79 and Fig. 1. Did you set the N ₂ O ₅ activity coefficient to zero or should it be unity, e.g. assuming ideal solution for N ₂ O ₅ . What activity coefficients do you use for the dissolved -+ inorganic ions, e.g. in this case NO ₃ ⁻ , H ⁺ ? Which base neutralize the dissolved HNO ₃ ? How did	Whilst trying to provide a realistic illustration of PyCHAM application to chamber experiments in a concise fashion, the referees have noted that the detail around Fig. 1 is insufficient. We have clarified and added information to the corresponding simulation explanation in lines 72-100. This includes explanation of simulation N ₂ O ₅ hydrolysis and estimating the

<p>you calculate the particle water content? How do you calculate the $[H^+]$ in the aqueous phase?</p> <p>$[H^+]$ is needed in order to calculate the effective Henry's law coefficient which in turn is needed to derive the condensation and dissolution growth for HNO_3 and N_2O_5 I presume.</p> <p>Based on the previous comments, does PyCHAM include any thermodynamics module which iteratively calculates the aqueous phase $[H^+]$ and particle water content etc., and if yes, how and when is it called?</p>	<p>accommodation coefficient for N_2O_5. It also includes detail on the absence of particle-phase chemistry or thermodynamic property estimation in PyHCAM – an important limitation that is now emphasised by these modifications.</p>
<p>3. P9-10, section 5.1 and Fig. 3. Considering that AtChem2 and PyCHAM use exactly the same MCM chemical scheme I am surprised that the deviation in the NO concentration is several percentage. Please discussion these results a bit more. What can the reasons be? Which error tolerances (solvers) did you use in PyCHAM and AtChem2?</p>	<p>The authors found that the wrong day of year had been used in the PyCHAM simulations for this comparison with AtChem2.</p> <p>The new results are shown in Figs. 3 & 4, with Fig. 3 showing that deviation reduced to around 2 % with this correction.</p> <p>Line 269 gives tolerances.</p>
<p>4. P14, 307-308. First, does not particle wall losses also directly influence the particle number size concentration? Secondly, would it not be more appropriate to write particle number concentration size distribution or just particle number size distribution?</p>	<p>Corrections made on line 302</p>
<p>5. P14, L319-L322, I agree that the full-moving particle distribution technique is not ideal when consider addition of new particles to the particle size distribution over time, but it is not impossible to use the full-moving approach as long as you add new particle size bins during the course of the simulation see e.g. Sect. 3.3 in Roldin et al., Atmos. Chem. Phys., 15, 10777– 10798, 2015. Hence, I would encourage you to add the option for users of PyCHAM to use the full-moving approach.</p>	<p>We thank the referee for this detail and have provided the full-moving size structure as an option in the new version of PyCHAM.</p> <p>Results involving a test/example of full-moving are presented in Figs. 6 and 11.</p>

<p>6. Section 7, Fig. 7 I was expecting the accumulation mode in Fig 7a to move towards the right because of condensational growth during the course of the simulation. But maybe this is not apparent when only looking at the particle volume size distribution. Can you also add a panel illustrating the particle number size distribution at the start and in the end?</p>	<p>Fig. 7 has become Fig. 6 to allow explanation of gas-particle partitioning prior to the section on gas-wall partitioning</p> <p>Plot (a) now contains both the volume- and the number-size distributions</p>
<p>7. I also think that it would have been more interesting and more appropriate, considering the intended use of PyCHAM, to show similar results from an idealized no-seed secondary organic aerosol formation experiment with new particle formation and consecutive growth.</p>	<p>The authors agree and have used a simple nucleation experiment from the Manchester aerosol chamber in the new Fig. 11 and Table 1 to compare PyCHAM against and to illustrate fitting of nucleation parameters.</p> <p>Whilst the main objective of section 10 is describing and exemplifying nucleation treatment in PyCHAM. Fig. 11 also acts to demonstrate a comparison of simulation output for a new particle formation and consecutive growth experiment with observations.</p>
<p>8. Section 8. I generally do not like numerical methods which is not mass conserving but I can accept the choice of the semiimplicit method of solving the coagulation process since you show that it is almost mass conserving. Still, I wonder what the advantage is of using this method in front of a simple fixed section method which will be both mass and number conserving? Is the semiimplicit approach by Jacobson (2005) resulting in less numerical diffusion problems compared to methods which are both mass and number conserving, but which rely on redistribution of the exact new single particle volumes onto the existing size bins using a fixed section approach (see e.g. Korhonen et al., Atmos. Chem. Phys., 4, 757–771, 2004)? Since, coagulation in contrast to condensation only change the size of a minor fraction of the total particle concentration each</p>	<p>This comment prompted us to detail further the semiimplicit method and the reason for this choice of method. Furthermore, we will investigate the feasibility of a mass-conserving approach. The relevant manuscript lines are 438-469.</p> <p>We also make the limitation of the semiimplicit approach clearer in the conclusion at line 686.</p>

<p>time step, numerical diffusion should not be a major concern with this fixed section approach and hence, I would encourage you to consider using such a mass and number conserving approach in future PyCHAM versions.</p>	
<p>9. Section 11, Fig. 11. To me this is the section which is least easy to understand. I think this is because Fig. 11 is not easy to follow. I think you should avoid to use a dark green background and red lines. I would suggest that you if possible replace this figure with one or several tables. You may then consider to mark the cells with less than 10 % deviation from the reference case.</p>	<p>Following this suggestion, we have substituted the original Fig. 11 for the new Tables 2-4.</p> <p>The contours of processing time provided in Fig. 11 were not particularly relevant due to the very simple chemical scheme used, therefore we leave Table 5 as an illustration of relevant processing times.</p>
<p>10. Table 1. Please write out the actual simulation time in hours instead of $\log_{10}(\text{time})$. $10^{3.4} \text{ s} \approx 0.7 \text{ h}$ for 2 size bins with 60 s time step is quite a long time for 6 hours of simulation I would say. I presume that this is mainly the gas-phase chemistry. I would like you to add the simulation time for a simulation without particles, e.g. only gas-phase chemistry and also add the simulation time for AtChem2 (no particles). I guess AtChem2 is written in Fortran or? Too me $10^{5.2} \text{ s} \approx 44 \text{ h}$ for 32 size bins with 60 s time step is not very user friendly. I guess it must have to do with the ode-solver and the stiffness of the differential equation system when you add many particle size bins. I would like to see some more discussion about these results and how they could be solved in future versions.</p>	<p>Table 1 is now Table 5. Presentation of processing times has been altered as suggested.</p> <p>Lines 657-670 make greater emphasis of the processing time limitation in PyCHAM, including comparison with AtChem for a like-for-like simulation.</p> <p>In these lines and in the conclusion (line 696) we suggest future work investigating incorporation of just in time compilers.</p>
<p>11. P11, L255. I do not understand the formulation “,with factor increases of 7 and 53 for 6×10^2 and $6 \times 10^1 \text{ s}$ resolutions, respectively.”</p>	<p>Corrected in line 295</p>

<p>12. P12, L286. “Third, k_w was held whilst” should it be Third, k_w was held constant whilst ... ?</p>	<p>Corrected on line 393</p>
<p>Referee # 2</p>	
<p>1. What is the file format of MCM that PyCHAM is taking, i.e., can it be directly used after downloaded the specific file from the MCM website or some formatting has to be made to the file before imported?</p>	<p>This explanation is now provided in line 148-151.</p>
<p>2. Fig. 1: since the size distribution uses $dN/d\log_{10}(D_p)$, the y-axis should be in \log_{10} scale instead of linear scale.</p>	<p>Fig. 1 now has this improvement</p>
<p>3. Is the chamber temperature coupled with the vapor molecule thermodynamic properties (e.g., the vapor pressure, diffusivity, etc.) in the PyCHAM?</p>	<p>Clarity around the effect of changing temperature is provided in line 183</p>
<p>4. What is the ODE solver used in the PyCHAM since most of the ODEs are very stiff? Are Jacobian matrices specified or numerically calculated in the code? What are the units used when simulating, e.g., ppm/min or pg/s or not the same for compounds in different phases?</p>	<p>These very important features are now better described in lines 116-121</p>
<p>5. L282: 0.5 μm seed particles are usually too big in a chamber experiment, which can cause significant particle-wall loss. Usually, the seed particles are around 100 nm. If no particle-wall loss is considered in this section, suggest using a smaller size of particles but with the same particle surface area concentration to test.</p>	<p>The authors agree that where possible simulations should represent those in real chamber experiments. Therefore, we use 100 nm diameter particles, whilst increasing number concentration to maintain surface area. The new results are shown in Fig. 7.</p> <p>This comment had referred to section 6 in the original draft, but this is now section 7 so that a description of gas-particle partitioning could precede it. In section 7.1 now we provide the updated Fig. 7 as a response and have adapted the main text (lines 385-390) accordingly.</p>

<p>6. Given different C_w values are used in Sec.6 for sensitivity test, Krechmer et al. (2016, DOI: 10.1021/acs.est.6b00606, Krechmer 2017 cited in the manuscript is not on this topic) have shown the vapor pressure-dependent C_w. Since the PyCHAM couples the molecule structure with the UManSysProp module, this character can be easily incorporated.</p>	<p>We thank the referee for their insight on this topic and have altered the references accordingly. We cite the mentioned paper but explain in lines 365-368 why its associated parametrisation for C_w is not included in this paper.</p>
<p>7. For both α-pinene and isoprene SOA experiments, please list the compounds (as well as their corresponding vapor pressures) that are allowed to partition to the particle phase.</p>	<p>This comment has prompted a significant upgrade to PyCHAM utility: a plotting script for a volatility basis set analysis of component contribution to particle-phase mass as a function of time and production of a comma separated value file containing individual component names, vapour pressures and their particle-phase mass concentration as a function of time.</p> <p>Using this plotting tool, we have included plot b in Fig. 7, which gives a time-dependent summary of mass contributions from components lumped by volatility. This plot prompts discussion in lines 399-424 which readers may find helpful in their own simulations.</p> <p>Whilst section 7 covers the isoprene experiment mentioned by the referee, the alpha-pinene simulations in section 5 deal only with the gas-phase, which we clarify now on line 270.</p> <p>The simulations using MCM schemes involve hundreds of components, all of which are allowed to partition to particles, which we now clarify on line 310. In the interest of readability we therefore do not provide the list suggested by the referee in the paper, however we hope that the new tool described above, which performs the same function with additional information and prompts new discussion in the paper (as detailed above), satisfies.</p>

8. L390: What are the timescales of coagulation versus condensation growth?	Line 495 includes this information which improves this section.
9. Fig. 10: Which experiment does this figure correspond to, α -pinene or isoprene?	Line 546 now better explains that this is a general plot independent of chemical scheme
10. Eq. 11: What is the unit of P_1 ? From the description, it looks like a rate. But Fig. 10 indicates it is a number concentration. If as a concentration, the nucleation rate would be changing over time (reaching the maximum around 150 s by differentiating curves in Fig. 10), how does that happen? Maybe explain this a bit more in this section.	The unit is now provided in line 544, whilst we better discuss the implications of Eq. 17 for rate of new particle formation in lines 548-550.
11. Sec. 11: The title "... spatial resolution" is confusing, which is actually "size" resolution. When appearing together with "temporal", one could think it as in a 3D space. Suggest to change it to an appropriate word.	The authors agree and have changed to number of size bins.
12. L471: Which method is used to interpolate the data points?	Linear interpolation is now specified on line 607
13. Eq. 13: Maybe use a different symbol (or sub-/super-scripts) for σ_g since it has been used to evaluate the Z & Y .	A new subscript has been used in Eq. 18 to make the labelling more intuitive.
14. Fig. 11 is a little bit difficult to follow: a). The simulation time in the context means the time consumed by the processor to finish the simulation, but one could think it as the time in the model to simulate. Maybe change to a different name. b). What is the number of size bin for different time interval simulation? c). If possible, contour plots of derivation/consuming time with the x-axis of the time interval and the y-axis of the number of size bin would be appreciated.	<p>The Fig. 11 of the original draft has been transformed to Tables 2-4, following both this comment and that of referee # 1.</p> <p>Simulation time has been changed to processing time, which is defined at line 134</p> <p>The processing times provided in Fig. 11 of the original draft were not particularly helpful as they related to a chemical scheme with only one</p>

	equation. Instead Table 5 has been improved and its surrounding discussion around processing times improved (lines 657-670).
15. Table 1: Instead of using log ₁₀ unit, units of hr or min are more intuitive to the readers.	Change made, note the original table 1 is now table 5.
Referee # 3	
1. Line 48: text in brackets should be clearer	Line 49 is now clearer and more accurate, since nucleation is allowed to occur in both seeded and unseeded experiments
2. Line 55-56: can the authors expand on how this fitting procedure is carried out? is there enough information for the user to be able to do this?	Lines 56-58 clarify processes requiring fitting and detail where further information can be found.
3. Line 72: define sufficient.	The entire introduction to Fig. 1 (lines 72-81) has been improved and the comment on sufficient concentrations for nucleation was seen as unnecessary, therefore please see the rewording on line 79.
4. Line 74: what do the authors mean by injected again later?	Clarity improved on line 80.
5. Line 79: give value of accommodation coefficient	Eq. 1 is provided and its surrounding text (lines 88-95) now provides more information about the accommodation coefficient for dinitrogen pentoxide
6. Figure 1: line up the x-axes to aid interpretation.	We thank the referee for this improvement which is provided in the revised Fig. 1
7. Line 101 doesn't appear to follow line 100??	Lines 123-126 provide improved wording of this line to aid readability.
8. Line 103: I guess photolysis rates are only constant for indoor lights?	Clarified on line 124.
9. Line 129: define process tendencies	Lines 159-160 now include a definition.
10. Line 185: define what relatively low means	Line 222 now includes a definition

11. line 286: is the word 'constant' missing?	Corrected on line 393
12. Line 313 - typo - predominantly?	Corrected on line 317.