
Interactive
comment

Interactive comment on “JIBox v1.0: A Julia based mixed-phase atmospheric chemistry box-model” by Langwen Huang and David Topping

Anonymous Referee #1

Received and published: 29 November 2020

The paper presents JIBox - a box model solving the gas phase chemical reactions and the gas - aerosol interactions. The presented model is implemented in Julia, a fairly new high performance general purpose programming language that is well suited for computational science applications. JIBox is based on a Python box model PyBox.

The paper compares the PyBox vs JIBox results and performance. It also discusses the use of different solvers and preconditioners available in JIBox model. As shown throughout the paper, the use of Julia language itself, combined with exploiting different packages available in the Julia ecosystem, significantly improve the JIBox performance over PyBox. An especially interesting case is the application of the automatic differentiation algorithm available in one of the Julia packages to easily build the adjoint sensitivity algorithm in JIBox.

[Printer-friendly version](#)

[Discussion paper](#)



Interactive comment

The paper is written well and clearly introduces different options available in the model. The ecosystem of scientific packages available in Julia is growing and I think that JlBox model combined with the description provided in this paper would be a useful new tool for the community. I recommend the paper to be published and I leave some minor comments below.

As a side note - I'm an environmental scientist using Julia in my work, but I'm not an atmospheric chemist. I read the paper and wrote my review from a point of view of a new user trying to use the provided code. However, I cannot review the scientific merit of the presented equations or the numerical approaches used to solve them. Nevertheless, I hope that my comments will be somewhat useful.

Questions:

1) The authors mention that sometimes the time-step length has to be reduced in order to preserve the positive sign of the chemical compounds. I'm wondering how often does that happen and how big of a potential bottleneck could this become. Are there other algorithms that could be implemented to deal with this problem that would not require decreasing the time step?

2) I was wondering if the performance discussion could be extended a little. Table B2 provides simulation times for selected 4 modeling cases. As mentioned before, I'm not an atmospheric chemist, but I was wondering if a plot with the number of chemical compounds, reaction pathways or size bins on one axis and elapsed time on another for the different box models would be useful. This would show how the computational cost scales with the problem complexity for both Julia and Python implementations. I was also wondering, for the large scale simulations performed on the cluster would it be possible to show some scaling plots with the increasing number of processors? Additionally, the authors briefly mention the potential to extend the JlBox to GPU accelerators. What would be the expected performance gain over the current simulations. Are there any other models similar to JlBox that are already running on GPUs?

[Printer-friendly version](#)

[Discussion paper](#)



Minor Comments:

- line 38 - Is the text in italics a quote from somewhere? If yes, could you provide the reference? If not, why is it highlighted?
- ~line 56 - Add section 5 into the list of paper sections described here.
- line 109 - JIBox is written
- line 249 - previously different fonts were used for package names such as DifferentialEquations.jl
- line 262 - I would cross out "simply"
- subsection 4.1 - Maybe the header could be "Validation against existing box-models" the repeated model seems off
- Table 3 - Python library instead of python library, UManSysprop instead of Uman-sysprop, Numba should be capitalized too

Comments after cloning from the GitHub repo

I tried compiling and running the project following instructions from GitHub. I ran into several small problems, but in general I was able to execute the tests and example simulations. I'm listing the problems I had below as feedback, but it does not concern the manuscript itself.

When running the tests I got an error in the Gas Phase Sparse with Matrix-free operator test complaining about UndefVarError: AnalyticalJacVecOperator not defined. I manually updated DiffEqOperators which solved the problem. After that, all the tests passed.

When running one of the example simulations I got some additional complaints about packages not being installed. I manually added them via Julia package manager following what the error messages were suggesting. In general it seems that the build

[Printer-friendly version](#)

[Discussion paper](#)



Interactive comment

and test stage did not set up correctly all the dependencies for me but I was able to easily resolve that.

After that I started getting errors about/data/*.txt files being missing. The way I understood the "Get Started" section, it suggests executing the simulations from the JIBox folder by `include("example/Simulation_*.jl")`. But the simulations themselves look for the .txt files in/data/ folder. The correct way for me to execute the simulations without changing the files was to `include("Simulation_*.jl")` inside the example folder inside Julia REPL. It might be worth it to update the "Get Started" section on GitHub to clarify that. Everything worked otherwise.

As future work it would be great to include some example plotting scripts in Julia within the GitHub repo and to add a binder setup. This would allow the new users to run and plot the model from the web browser and showcase even better the strength of using Julia where both the high-performance computing and the analysis can be done in one programming language. It would also help a lot to get the new users up to speed in running and visualizing the simulations on their own.

I also think that package naming conventions in Julia suggest not having Julia in their name and instead ask for the package name to finish with .jl extension (<https://julialang.github.io/Pkg.jl/v1/creating-packages/#Package-naming-guidelines>). Might be too late to suggest changing the package name now, but I thought I should leave it as a comment.

Interactive comment on Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2020-344>, 2020.

[Printer-friendly version](#)

[Discussion paper](#)

