




The Python Sky Model 3 software

Andrea Zonca¹, Ben Thorne², Nicoletta Krachmalnicoff^{3,4,5}, and Julian Borrill^{6,7}

1 San Diego Supercomputer Center, University of California San Diego, San Diego, USA 2 Department of Physics, University of California Davis, One Shields Avenue, Davis, CA 95616, USA 3 SISSA, Via Bonomea 265, 34136 Trieste, Italy 4 INFN, Via Valerio 2, 34127 Trieste, Italy 5 IFPU, Via Beirut 2, 34014 Trieste, Italy 6 Computational Cosmology Center, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA 7 Space Sciences Laboratory at University of California, 7 Gauss Way, Berkeley, CA 94720

DOI: [10.21105/joss.03783](https://doi.org/10.21105/joss.03783)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Christina Hedges](#) 

Reviewers:

- [@patricialarsen](#)
- [@smsharma](#)

Submitted: 24 July 2021

Published: 01 November 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Statement of Need

The Cosmic Microwave Background (CMB) radiation, emitted just 370 thousand years after the Big Bang, is a pristine probe of the Early Universe. After being emitted at high temperatures, the CMB was redshifted by the subsequent 13.8 billion years of cosmic expansion, such that it is brightest at microwave frequencies today. However, our own Milky Way galaxy also emits in the microwave portion of the spectrum, obscuring our view of the CMB. Examples of this emission are thermal radiation by interstellar dust grains and synchrotron emission by relativistic electrons spiraling in magnetic fields. Cosmologists need to create synthetic maps of the CMB and of the galactic emission based on available data and on physical models that extrapolate observations to different frequencies. The resulting maps are useful to test data reduction algorithms, to understand residual systematics, to forecast maps produced by future instruments, to run Monte Carlo analysis for noise estimation, and more.

Summary

The Python Sky Model (PySM) is a Python package used by Cosmic Microwave Background (CMB) experiments to simulate maps, in HEALPix ([Górski et al., 2005](#); [Zonca et al., 2019](#)) pixelization, of the various diffuse astrophysical components of Galactic emission relevant at CMB frequencies (i.e., dust, synchrotron, free-free and Anomalous Microwave Emission), as well as the CMB itself. These maps may be integrated over a given instrument bandpass and smoothed with a given instrument beam. The template emission maps used by PySM are based on Planck ([Planck Collaboration, 2018](#)) and WMAP ([Bennett et al., 2013](#)) data and are noise-dominated at small scales. Therefore, PySM simulation templates are smoothed to retain the large-scale information, and then supplemented with modulated Gaussian realizations at smaller scales. This strategy allows one to simulate data at higher resolution than the input maps.

PySM 2 ([Thorne et al., 2017](#)), released in 2016, has become the de-facto standard for simulating Galactic emission; it is used, for example, by CMB-S4, Simons Observatory, LiteBird, PICO, CLASS, POLARBEAR, and other CMB experiments, as shown by the [80+ citations of the PySM 2 publication](#). As the resolution of upcoming experiments increases, the PySM 2 software has started to show some limitations:

- Emission templates are provided at 7.9 arcminutes resolution (HEALPix $N_{side} = 512$), while the next generation of CMB experiments will require sub-arcminute resolution.

- The software is implemented in pure `numpy`, meaning that it has significant memory overhead and is not multi-threaded, precluding simply replacing the current templates with higher-resolution versions.
- Emission templates are included in the PySM 2 Python package, which is still practical when each of the roughly 40 input maps is ~ 10 Megabytes, but will not be if they are over 1 Gigabyte.

The solution to these issues was to reimplement PySM from scratch focusing of these features:

- Reimplement all the models with the `numba` (Lam et al., 2015) Just-In-Time compiler for Python to reduce memory overhead and optimize performance: the whole integration loop of a template map over the frequency response of an instrument is performed in a single pass in automatically compiled and multi-threaded Python code.
- Use MPI through `mpi4py` to coordinate execution of PySM 3 across multiple nodes, this allows supporting template maps at a resolution up to 0.4 arcminutes (HEALPix $N_{side} = 8192$).
- Rely on `libsharp` (Reinecke & Seljebotn, 2013), a distributed implementation of spherical harmonic transforms, to smooth the maps with the instrument beam when maps are distributed over multiple nodes with MPI.
- Employ the data utilities infrastructure provided by `astropy` (Astropy Collaboration et al., 2018, 2013) to download the input templates and cache them when requested.

At this stage we strive to maintain full compatibility with PySM 2, therefore we implement the exact same astrophysical emission models with the same naming scheme. In the extensive test suite we compare the output of each PySM 3 model with the results obtained by PySM 2.

Performance

As an example of the performance improvements achieved with PySM 3 over PySM 2, we run the following configuration:

- An instrument with 3 channels, with different beams, and a top-hat bandpass defined numerically at 10 frequency samples.
- A sky model with the simplest models of dust, synchrotron, free-free and AME [a1, d1, s1, f1 in PySM terms].
- Execute on a 12-core Intel processor with 12 GB of RAM.

The following tables shows the walltime and peak memory usage of this simulation executed at the native PySM 2 resolution of $N_{side} = 512$ and at two higher resolutions:

Output N_{side}	PySM 3	PySM 2
512	1m 0.7 GB	1m40s 1.45 GB
1024	3m30s 2.3 GB	7m20s 5.5 GB
2048	16m10s 8.5 GB	Out of memory

The models at $N_{side} = 512$ have been tested to be equal given a relative tolerance of $1e-5$.

At the moment it is not very useful to run at resolutions higher than $N_{side} = 512$ because there is no actual template signal at smaller scales. However, this demonstrates the performance improvements that will make working with higher resolution templates possible.

Future work

PySM 3 opens the way to implement a new category of models at much higher resolution. However, instead of just upgrading the current models to smaller scales, we want to also update them with the latest knowledge of Galactic emission and gather feedback from each of the numerous CMB experiments. For this reason we are collaborating with the Panexperiment Galactic Science group to lead the development of the new class of models to be included in PySM 3.

How to cite

If you are using PySM 3 for your work, please cite this paper for the software itself; for the actual emission modeling please also cite the original PySM 2 paper (Thorne et al., 2017). There will be a future paper on the generation of new PySM 3 astrophysical models.

Acknowledgments

- This work was supported in part by NASA grant 80NSSC18K1487.
- The software was tested, in part, on facilities run by the Scientific Computing Core of the Flatiron Institute.
- This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05C H11231.

References

- Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., VanderPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., ... Astropy Contributors. (2018). The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *The Astronomical Journal*, 156(3), 123. <https://doi.org/10.3847/1538-3881/aabc4f>
- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., ... Streicher, O. (2013). Astropy: A community Python package for astronomy. *Astronomy and Astrophysics*, 558, A33. <https://doi.org/10.1051/0004-6361/201322068>
- Bennett, C. L., Larson, D., Weiland, J. L., Jarosik, N., Hinshaw, G., Odegard, N., Smith, K. M., Hill, R. S., Gold, B., Halpern, M., Komatsu, E., Nolte, M. R., Page, L., Spergel, D. N., Wollack, E., Dunkley, J., Kogut, A., Limon, M., Meyer, S. S., ... Wright, E. L. (2013). Nine-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Final Maps and Results. *The Astrophysical Journal Supplement Series*, 208, 20. <https://doi.org/10.1088/0067-0049/208/2/20>
- Górski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. (2005). HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere. *The Astrophysical Journal*, 622, 759–771. <https://doi.org/10.1086/427976>

- Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. <https://doi.org/10.1145/2833157.2833162>
- Planck Collaboration. (2018). *Planck 2018 results. I. Overview and the cosmological legacy of Planck*. <http://arxiv.org/abs/1807.06205>
- Reinecke, M., & Seljebotn, D. S. (2013). Libsharp – spherical harmonic transforms revisited. *Astronomy and Astrophysics*, 554, A112. <https://doi.org/10.1051/0004-6361/201321494>
- Thorne, B., Dunkley, J., Alonso, D., & Næss, S. (2017). The Python Sky Model: software for simulating the Galactic microwave sky. *Monthly Notices of the Royal Astronomical Society*, 469(3), 2821–2833. <https://doi.org/10.1093/mnras/stx949>
- Zonca, A., Singer, L. P., Lenz, D., Reinecke, M., Rosset, C., Hivon, E., & Gorski, K. M. (2019). healpy: equal area pixelization and spherical harmonics transforms for data on the sphere in Python. *Journal of Open Source Software*, 4(35), 1298. <https://doi.org/10.21105/joss.01298>