The Journal of Open Source Software

cosmo-numba: B-modes and COSEBIs computations accelerated by Numba

Axel Guinot[1] and **Rachel Mandelbaum**[1]

1 Department of Physics, McWilliams Center for Cosmology and Astrophysics, Carnegie Mellon University, Pittsburgh, PA 15213, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Weak gravitational lensing is a widely used probe in cosmological analysis. It allows astrophysists to understand the content and evolution of the Universe. We are entering an era where we are not limited by the data volume but by systematic uncertainties. It is in this context that we present here a simple python-based software package to help in the computation of E-/B-mode decomposition, which can be use for systematic checks or science analysis. As we demonstrate, our implementation has both the high precision and speed required to perform this kind of analysis while avoiding a scenario wherein either numerical precision or computational time is a significant limiting factor.

Statement of need

The E-/B-mode composition for cosmic shear poses a significant computational challenge given the need for high precision (required to integrate oscillatory functions over a large integration range and achieve accurate results) and speed. `Cosmo-numba` meets this need, facilitating the computation of E-/B-mode decomposition using two methods. One of them is the Complete Orthogonal Sets of E-/B-mode Integrals (COSEBIs) as presented in P. Schneider et al. (2010). The COSEBIs rely on very high precision computation requiring more than 80 decimal places. P. Schneider et al. (2010) propose an implementation using `mathematica`. `cosmo-numba` uses a combination of `sympy` and `mpmath` to reach the required precision. This python version enables an easier integration within cosmological inference pipelines, which are commonly python-based, and facilitates the null tests.

This software package also enables the computation of the pure-mode correlation functions presented in Peter Schneider et al. (2022). Those integrals are less numerically challenging than the COSEBIs, but having a fast computation is necessary for their integration in an inference pipeline. Indeed, one can use those correlation functions for cosmological inference, in which case the large number of calls to the likelihood function will require a fast implementation.

State of the field

There are other implementations of the COSEBIs such as CosmoPipe[1] used in the KiDS-legacy analysis (Wright et al., 2025). Our implemetation is characterized by the use of `numba` that makes the computation of the filter functions described in section 6 faster. Regarding the pure E-/B-mode decomposition, we have not found a similar publicly available implementation. That being said, they are classicaly used as a one-time measure for null tests in various surveys. The implementation we are presenting would enable one to use this decomposition

[1] https://github.com/AngusWright/CosmoPipe

for cosmological inference, which requires computing several integrals at each likelyhood call. While the commonly used `scipy` library would make the computation untractable, the speed gain by switching to `numba` opens new opportunities such as this one.

Software design

This package has been designed around two constraints: precision and speed. As it can be difficult to reach both at the same time, the code is partitioned in a way that parts requiring high precision are done using python libraries such as `sympy` and `mpmath`. In contrast, parts of the code that do not require high precision leverage the power of Just-In-Time (JIT) compilation. `Numba` provides significant speed up compared to a classic python implementation. As this library is intended to provide tools for cosmological computation, it was important to provide meaningful unit tests and demonstrate a full coverage of the library. Providing an accurate coverage is challenging when using `numba` compiled code. Our implementation allows the developer to disable compilation for the targeted part of the code when performing coverage tests. This allows us to provide both high quality unit tests and good coverage to the users.

Testing setup

In the following two sections we make use of fiducial shear-shear correlation functions, $\xi_{\pm}(\theta)$, and power spectra, $P_{E/B}(\ell)$. They have been computed using the Core Cosmology Library[2] (Chisari et al., 2019). The cosmological parameters are taken from Aghanim et al. (2020). For tests that involved covariances we are using the Stage-IV Legacy Survey of Space and Time (LSST) Year 10 as a reference. The characteristics are taken from the LSST Dark Energy Science Collaboration (DESC) Science Requirements Document (SRD) (The LSST Dark Energy Science Collaboration et al., 2021).

COSEBIs

The COSEBIs are defined as:

$$E_n = \frac{1}{2}\int_0^{\infty} d\theta\,\theta[T_{n,+}(\theta)\xi_+(\theta) + T_{n,-}(\theta)\xi_-(\theta)], \tag{1}$$

$$B_n = \frac{1}{2}\int_0^{\infty} d\theta\,\theta[T_{n,+}(\theta)\xi_+(\theta) - T_{n,-}(\theta)\xi_-(\theta)]; \tag{2}$$

where $\xi_{\pm}(\theta)$ are the shear correlation functions, and $T_{n,\pm}$ are the weight functions for the COSEBI mode n. The complexity is in the computation of the weight functions. `Cosmo-numba` carries out the computation of the weight functions in a logarithmic scale defined by:

$$T_{n,+}^{\log}(\theta) = t_{n,+}^{\log}(z) = N_n \sum_{j=0}^{n+1} \bar{c}_{nj} z^j; \tag{3}$$

whare $z = \log(\theta/\theta_{\min})$, N_n is the normalization for the mode n, and $\bar{c}_{jn}$ are defined iteratively from Bessel functions (we refer the readers to P. Schneider et al. (2010) for more details).

We have validating our implementation against the original version in `Mathematica` from P. Schneider et al. (2010). In Figure 1 we show the impact of the precision going from 15 decimal places, which corresponds to the precision one could achieve using float64, up to 80 decimal

[2] https://github.com/LSSTDESC/CCL

places, the precision used in the original `Mathematica` implementation. We can see that classic float64 precision would not be sufficient, and with a precision of 80 our code exactly recovers the results from the original implementation. Given that the precision comes at very little computational cost, we default to the original implementation using high precision. The impact of the precision propagated to the COSEBIs is shown in Figure 2. We can see that using a lower precision than the default setting can incur a several percent error.

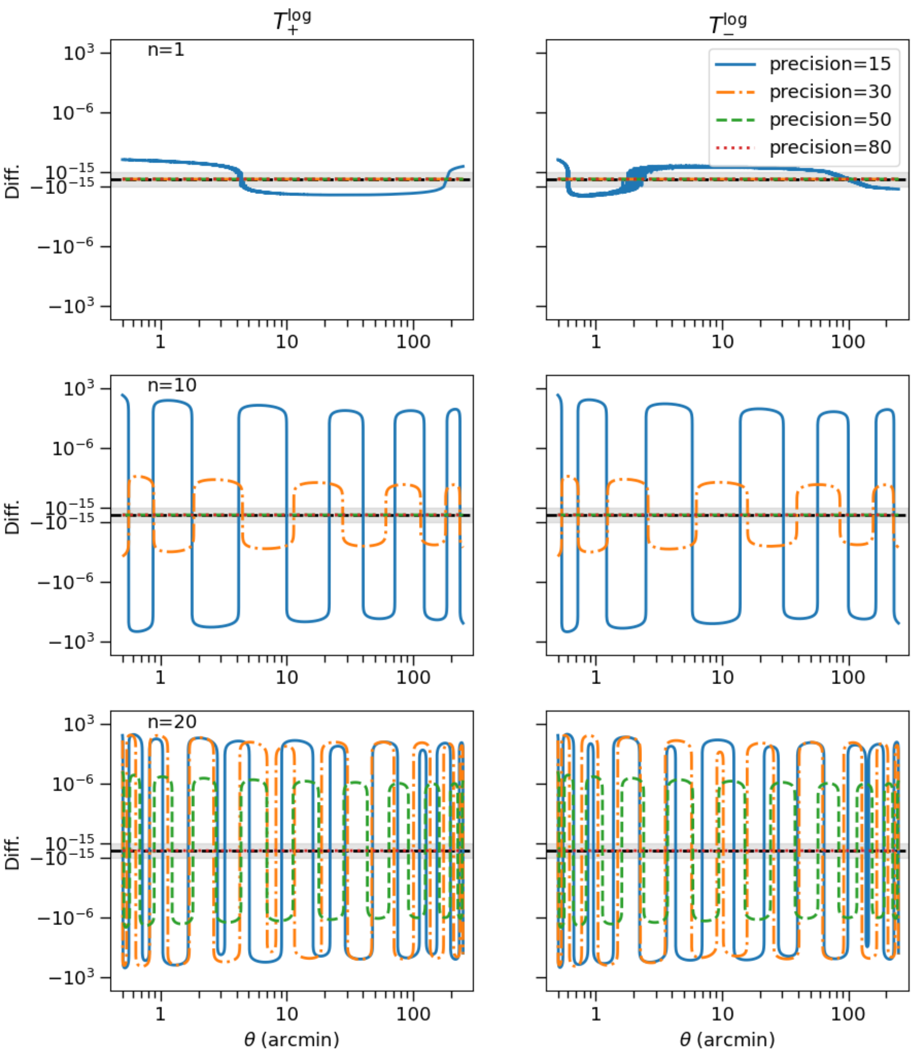

Figure 1: In this figure we show the impact of the precision in the computation of the weight functions $T_\pm^{\log}$. For comparion, a precision of 15 corresponds to what would be achieved using `numpy` float64. The difference is computed with respect to the original `Mathematica` implementation presented in P. Schneider et al. (2010). The figure uses symlog, with the shaded region representing the linear scale in the range $[-10^{-15}, 10^{-15}]$.

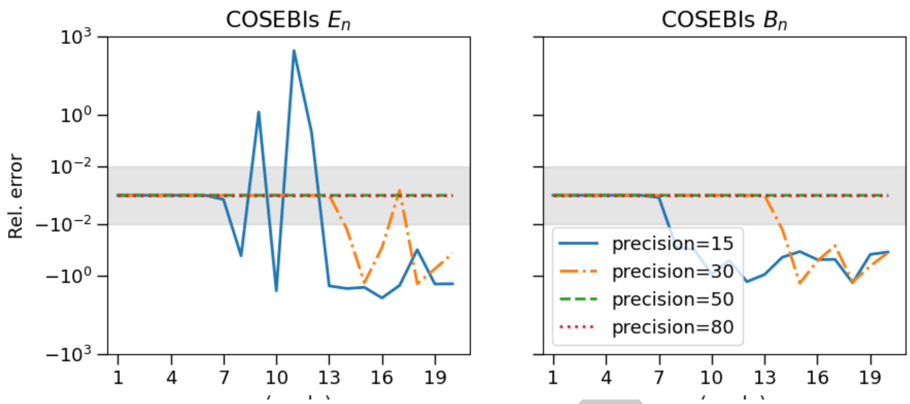

Figure 2: This figure shows the difference in the COSEBIs E- and B-modes relative to the original `Mathematica` implementation. We see that using only 15 decimal places would lead to several percent error, making an implementation based on `numpy` float64 not suitable. The figure uses symlog, with the shaded region representing the linear scale in the range $[-1, 1]$ percent.

COSEBIs can also be defined from the power spectrum as:

$$E_n = \int_0^\infty \frac{d\ell\,\ell}{2\pi} P_E(\ell) W_n(\ell); \tag{4}$$

$$B_n = \int_0^\infty \frac{d\ell\,\ell}{2\pi} P_B(\ell) W_n(\ell); \tag{5}$$

where $P_{E/B}(\ell)$ is the power spectrum of E- and B-modes and $W_n(\ell)$ are the filter functions which can be computed from $T_{n,+}$ as:

$$W_n(\ell) = \int_{\theta_{\mathrm{min}}}^{\theta_{\mathrm{max}}} d\theta\, \theta T_{n,+}(\theta) J_0(\ell\theta); \tag{6}$$

with $J_0(\ell\theta)$ the 0-th order Bessel function. The Equation 6 is a Hankel transform of order 0. It can be computed using the `FFTLog` algorithm presented in Hamilton (2000) implemented here in `Numba`. Figure 3 shows the comparison between the COSEBIs computed from $\xi_\pm(\theta)$ and from $C_{E/B}(\ell)$. We can see that the COSEBI E- & B-modes agree very well, with at most 0.3σ difference with respect to the LSST Y10 covariance. We consider that using either approach would not impact the scientific interpretation and both could be used for consistency checks.

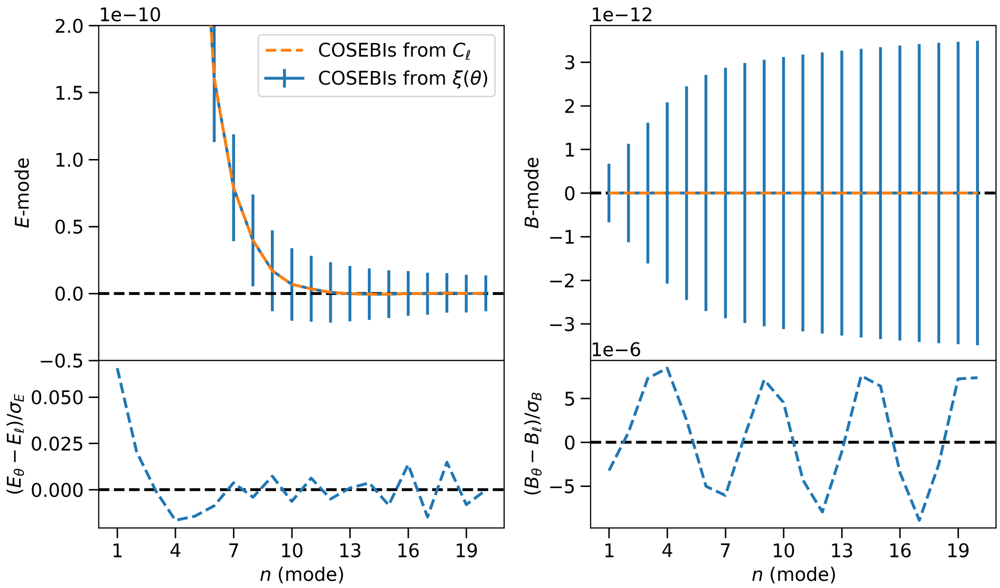

Figure 3: Comparison of the COSEBIs E- and B-mode computed from $\xi_\pm(\theta)$ and $C_{E/B}(\ell)$. The *upper* panel shows the COSEBIs E-/B-modes while the *bottom* panel shows the difference with respect to the statistical uncertainty based on the LSST Y10 covariance.

Finally, we have compared our implementation agains CosmoPipe[3] which make use of a different integration method to compute the filter functions such as Levin integration. We found that our implementation using numba is around 100 times faster for equivalent precision.

Pure-Mode Correlation Functions

In this section we describe the computation of the pure-mode correlation functions as defined in Peter Schneider et al. (2022). These are defined as follows:

$$\xi_+^E(\vartheta) = \frac{1}{2}\left[\xi_+(\vartheta) + \xi_-(\vartheta) + \int_\vartheta^{\vartheta_{\max}} \frac{d\theta}{\theta}\xi_-(\theta)\left(4 - \frac{12\vartheta^2}{\theta^2}\right)\right] - \frac{1}{2}\left[S_+(\vartheta) + S_-(\vartheta)\right], \quad (7)$$

$$\xi_+^B(\vartheta) = \frac{1}{2}\left[\xi_+(\vartheta) - \xi_-(\vartheta) - \int_\vartheta^{\vartheta_{\max}} \frac{d\theta}{\theta}\xi_-(\theta)\left(4 - \frac{12\vartheta^2}{\theta^2}\right)\right] - \frac{1}{2}\left[S_+(\vartheta) - S_-(\vartheta)\right], \quad (8)$$

$$\xi_-^E(\vartheta) = \frac{1}{2}\left[\xi_+(\vartheta) + \xi_-(\vartheta) + \int_{\vartheta_{\min}}^{\vartheta} \frac{d\theta\theta}{\vartheta^2}\xi_+(\theta)\left(4 - \frac{12\theta^2}{\vartheta^2}\right)\right] - \frac{1}{2}\left[V_+(\vartheta) + V_-(\vartheta)\right], \quad (9)$$

$$\xi_-^B(\vartheta) = \frac{1}{2}\left[\xi_+(\vartheta) - \xi_-(\vartheta) + \int_{\vartheta_{\min}}^{\vartheta} \frac{d\theta\theta}{\vartheta^2}\xi_+(\theta)\left(4 - \frac{12\theta^2}{\vartheta^2}\right)\right] - \frac{1}{2}\left[V_+(\vartheta) - V_-(\vartheta)\right]; \quad (10)$$

where $\xi_\pm(\theta)$ correspond to the shear-shear correlation function. The functions $S_\pm(\theta)$ and $V_\pm(\theta)$ are themselves defined by integrals and we refer the reader to Peter Schneider et al. (2022)

[3]The test has done on a Mac M3 Max using 16 cores. The script to run the test is available at: https://github.com/aguinot/cosmo-numba/blob/main/notebooks/cosebis_comparison_with_cosmopipe.ipynb

for more details about their definition. By contrast with the computation of the COSEBIs, these integrals are more stable and straightforward to compute but still require some level of precision. This is why we are using the qags method from QUADPACK (Piessens et al., 2012) with a 5th order spline interpolation. In addition, as one can see from the equations above, the implementation requires a loop over a range of ϑ values. This is why having a fast implementation will be required if one wants to use those correlation functions in cosmological inference. In Figure 4 we show the decomposition of the shear-shear correlation function into the E-/B-modes correlation functions and ambiguous mode.

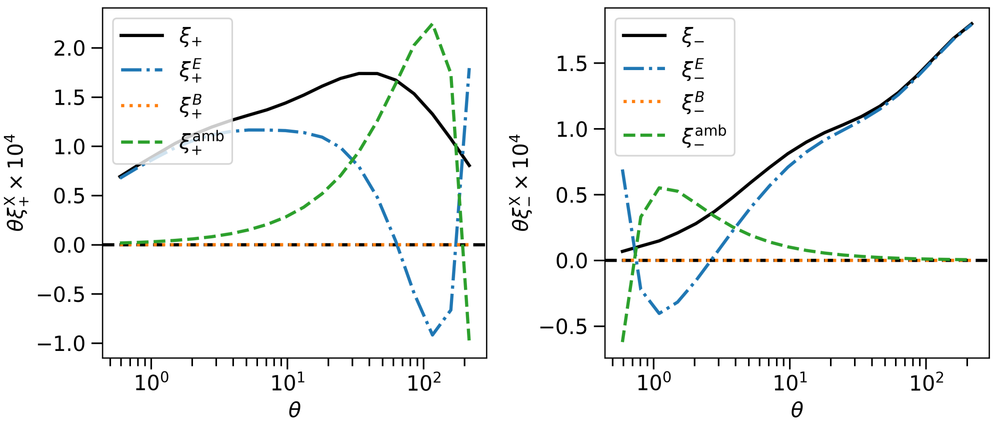

Figure 4: This figure shows the decomposition of the shear-shear correlation functions into E- and B-modes (and ambiguous mode).

To assess the speed improvement of our implementation, we have run the same computation using `Scipy` functions: `CubicSpline` for the interpolation and quad for the integration[4]. While the precision is comparable, our serial version is more than 8 time faster while the parallel version is more than 50 times faster.

Research impact statement

This software is being in the Utraviolet Near Infrared Optical Northern Survey (UNIONS) to validate the catalogue used for cosmological analysis (Daley & others, 2026). We are also planning to use this code in the Roman High Latitude Imaging Survey (HLIS). In addition to its current usage in science collaborations, we provide unit tests that not only validate the implementation but also validate the computation mathematically and provide a higher bound for the accuracy of the code. Fianlly, examples can be found in the code repository that provide comparison against alternative approaches and implementations. They show that the computation presented here is significantly faster than existing alternatives.

AI usage disclosure

Artificial Intelligence (AI) has been use to help with documentation, docstrings and for some of the unit tests.

[4]The test has been done on a Mac M3 Max using 16 cores. The script to run the test is available at: https://github.com/aguinot/cosmo-numba/blob/main/notebooks/pure_EB_modes_comparison_with_scipy.ipynb

Acknowledgements

The authors acknowledge the support of a grant from the Simons Foundation (Simons Investigator in Astrophysics, Award ID 620789).

References

Aghanim, N., Akrami, Y., Ashdown, M., Aumont, J., Baccigalupi, C., Ballardini, M., Banday, A. J., Barreiro, R. B., Bartolo, N., Basak, S., Battye, R., Benabed, K., Bernard, J.-P., Bersanelli, M., Bielewicz, P., Bock, J. J., Bond, J. R., Borrill, J., Bouchet, F. R., … Zonca, A. (2020). Planck 2018 results: VI. Cosmological parameters. *Astronomy &Amp; Astrophysics*, *641*, A6. https://doi.org/10.1051/0004-6361/201833910

Chisari, N. E., Alonso, D., Krause, E., Leonard, C. D., Bull, P., Neveu, J., Villarreal, A., Singh, S., McClintock, T., Ellison, J., Du, Z., Zuntz, J., Mead, A., Joudaki, S., Lorenz, C. S., Tröster, T., Sanchez, J., Lanusse, F., Ishak, M., … Wagoner, E. L. (2019). Core cosmology library: Precision cosmological predictions for LSST. *The Astrophysical Journal Supplement Series*, *242*(1), 2. https://doi.org/10.3847/1538-4365/ab1658

Daley, C., & others. (2026). UNIONS-3500 2D Cosmic Shear: III. B-mode tests and validation. *In Preparation*.

Hamilton, A. J. S. (2000). Uncorrelated modes of the non-linear power spectrum. *Monthly Notices of the Royal Astronomical Society*, *312*(2), 257–284. https://doi.org/10.1046/j.1365-8711.2000.03071.x

Piessens, R., Doncker-Kapenga, E. de, Überhuber, C. W., & Kahaner, D. K. (2012). *Quadpack: A subroutine package for automatic integration* (Vol. 1). Springer Science & Business Media.

Schneider, Peter, Asgari, M., Jozani, Y. N., Dvornik, A., Giblin, B., Harnois-Déraps, J., Heymans, C., Hildebrandt, H., Hoekstra, H., Kuijken, K., Shan, H., Tröster, T., & Wright, A. H. (2022). Pure-mode correlation functions for cosmic shear and application to KiDS-1000. *Astronomy &Amp; Astrophysics*, *664*, A77. https://doi.org/10.1051/0004-6361/202142479

Schneider, P., Eifler, T., & Krause, E. (2010). COSEBIs: Extracting the full e-/b-mode information from cosmic shear correlation functions. *Astronomy and Astrophysics*, *520*, A116. https://doi.org/10.1051/0004-6361/201014235

The LSST Dark Energy Science Collaboration, Mandelbaum, R., Eifler, T., Hložek, R., Collett, T., Gawiser, E., Scolnic, D., Alonso, D., Awan, H., Biswas, R., Blazek, J., Burchat, P., Chisari, N. E., Dell'Antonio, I., Digel, S., Frieman, J., Goldstein, D. A., Hook, I., Ivezić, Ž., … Troxel, M. A. (2021). *The LSST dark energy science collaboration (DESC) science requirements document*. https://arxiv.org/abs/1809.01669

Wright, A. H., Stölzner, B., Asgari, M., Bilicki, M., Giblin, B., Heymans, C., Hildebrandt, H., Hoekstra, H., Joachimi, B., Kuijken, K., Li, S.-S., Reischke, R., Wietersheim-Kramsta, M. von, Yoon, M., Burger, P., Chisari, N. E., Jong, J. de, Dvornik, A., Georgiou, C., … Zhang, Y.-H. (2025). KiDS-legacy: Cosmological constraints from cosmic shear with the complete kilo-degree survey. *Astronomy &Amp; Astrophysics*, *703*, A158. https://doi.org/10.1051/0004-6361/202554908