The Journal of Open Source Software

ultrasphere and ultrasphere-harmonics: Python packages for Vilenkin–Kuznetsov–Smorodinsky polyspherical coordinates and hyperspherical harmonics techniques in array API

Hiromochi Itoh[1*], **Kei Matsushima**[2*], **and Takayuki Yamada**[3]

1 Department of Mechanical Engineering, Graduate School of Engineering, The University of Tokyo, Japan **2** Graduate School of Advanced Science and Engineering, Hiroshima University, Japan **3** Department of Strategic Studies, Institute of Engineering Innovation, Graduate School of Engineering, The University of Tokyo * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

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

Summary

Spherical harmonics, which are the solutions to the angular part of the laplace equation, have been widely used in various fields of science and engineering. Especially, hyperspherical harmonics, which are spherical harmonics in higher dimensions, have been applied to many-body problems in quantum mechanics and nuclear physics

Statement of need

`ultrasphere` and `ultrasphere-harmonics` are Python packages for hyperspherical coordinates and hyperspherical harmonics techniques. Our packages is that they support any type of Vilenkin–Kuznetsov–Smorodinsky polyspherical coordinate systems (Vilenkin & Klimyk, 1993). This allows to write codes that work in any type of polyspherical coordinates and thus in any number of dimensions. To demonstrate this, we implemented acoustic scattering from a single sphere for any type of polyspherical coordinates, which could be verified by command-line interface.

Our api is compatible with the array API standard (Meurer et al., 2023). This enables writing code which runs on multiple array libraries (e.g., NumPy(Harris et al., 2020), PyTorch(Paszke et al., 2019)) and multiple hardware (e.g., CPU, GPU). Our packages fully support vectorization for high performance computing.

Acknowledgements

This work used computational resources Supermicro ARS-111GL-DNHR-LCC and FUJITSU Server PRIMERGY CX2550 M7 (Miyabi) at Joint Center for Advanced High Performance Computing (JCAHPC) and TSUBAME4.0 supercomputer provided by Institute of Science Tokyo through Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures and High Performance Computing Infrastructure in Japan (Project ID: jh240031).

DRAFT

References

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Meurer, A., Reines, A., Gommers, R., Fang, Y.-L. L., Kirkham, J., Barber, M., Hoyer, S., Müller, A., Zha, S., Shanabrook, S., Gacha, S. J., Lezcano-Casado, M., Fan, T. J., Reddy, T., Passos, A., Kwon, H., Oliphant, T., & Standards, C. for P. D. A. (2023). Python array API standard: Toward array interoperability in the scientific python ecosystem. *Scipy*. https://doi.org/10.25080/gerudo-f2bc6f59-001

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd international conference on neural information processing systems* (pp. 8026–8037). Curran Associates Inc.

Vilenkin, N. Ja., & Klimyk, A. U. (1993). *Representation of lie groups and special functions* (Vol. 74). Springer Netherlands. https://doi.org/10.1007/978-94-017-2883-6

DRAFT