ndsplines: A Python Library for Tensor-Product B-Splines of Arbitrary Dimension

Benjamin W. L. Margolis[1] and Kenneth R. Lyons[1]

1 University of California, Davis

Summary

In order to use data driven models in analysis, the data must be projected onto a set of basis functions to allow the model to be evaluated over a continuous domain. Additionally, the known form of the basis functions enables the use of calculus and other symbolic operations to be performed on the representation of the data model.

B-Splines are a particularly useful basis because they have beneficial analytic properties and because many efficient algorithms for their creation, manipulation, and evaluation have already been developed (De Boor, 1978). Some of the beneficial properties of the B-Spline basis functions include: a closed support, smoothness, and amenability to efficient and stable algorithms. Strictly speaking, the B-Spline basis functions are 1-dimensional maps from the reals to the reals. However, they can be readily extended to N-dimensions by taking tensor products of B-Splines on orthogonal 1-dimensional spaces for each variable.

B-Splines are well studied in the field of computer graphics where they are particularly used in computer aided-design (CAD) and graphic design programs. Due to this popularity, many programming language ecosystems have implementations of 1- and 2-dimensional splines such as might be used to design curves or surfaces. For example, SciPy provides a number of interfaces to Dierckx's FORTRAN implementation as well as a custom 1-D implementation (Dierckx, 1995; Jones, Oliphant, Peterson, & others, n.d.). However, the authors are only aware of one general N-dimensional splines implementation (De Boor, 1990).

ndsplines is a Python package for multivariate B-splines with performant NumPy and C (via Cython) implementations (Behnel et al., 2011; Van Der Walt, Colbert, & Varoquaux, 2011). A primary goal of this package is to provide a unified API for tensor product splines of arbitrary input and output dimension. This implementation is slightly slower than the special-case 1-D and 2-D implementations available from SciPy, but this can be a worthwhile trade-off to only use one API to work with splines of any dimension. Test coverage is about 90%, and importantly verifies desirable mathematical properties including that the spline evaluation matches other implementations, that the fundamental theorem of calculus holds for any `NDSpline` object, that the accuracy of splines produced by `make_lsq_spline` improves with the signal-to-noise ratio of the sampled data, and that the splines produced by `make_interp_spline` actually interpolate the data.

A general N-dimensional spline implementation is useful for modeling dynamics from data. For example, the American Institute of Aeronautics and Astronautics (AIAA) has proposed a simulation description mark-up language standard (Jackson & Hildreth, 2002). One of the fundamental elements of the mark-up language is an interpolator of aerodynamic data, which often uses three to five independent variables to capture properties of the airflow and orientation.

One author has used a precursor to this software package along with SimuPy (Margolis, 2017) to simulate hypersonic entry dynamics in a manner similar to what was described in AIAA's

simulation description mark-up language (D'Souza et al., 2019; Margolis, Okolo, Nikaido, Barton, & D'Souza, 2019). Static stability analysis was also performed using this software package.

References

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The best of both worlds. *Computing in Science & Engineering*, *13*(2), 31. doi:10.1109/mcse.2010.118

De Boor, C. (1978). *A practical guide to splines* (Vol. 27). Springer-Verlag New York. doi:10.1002/zamm.19800600129

De Boor, C. (1990). *Spline toolbox: For use with MATLAB: User's guide, march 30, 1990*. MathWorks Incorporated.

Dierckx, P. (1995). *Curve and surface fitting with splines*. Oxford University Press.

D'Souza, S. N., Okolo, W., Nikaido, B., Yount, B., Tran, J., Margolis, B., Smith, B., et al. (2019). Developing an entry guidance and control design capability using flaps for the lifting Nano-ADEPT. In *AIAA aviation 2019 forum* (p. 2901). doi:10.2514/6.2019-2901

Jackson, E. B., & Hildreth, B. (2002). Flight dynamic model exchange using XML. In *AIAA modeling and simulation technologies conference and exhibit* (p. 4482). doi:10.2514/6.2002-4482

Jones, E., Oliphant, T., Peterson, P., & others. (n.d.). SciPy: Open source scientific tools for Python. Retrieved from http://www.scipy.org/

Margolis, B. W. (2017). SimuPy: A Python framework for modeling and simulating dynamical systems. *J. Open Source Software*, *2*(17), 396. doi:10.21105/joss.00396

Margolis, B. W., Okolo, W. A., Nikaido, B., Barton, J. D., & D'Souza, S. N. (2019). Control and simulation of a deployable entry vehicle with aerodynamic control surfaces. In *AAS/aiaa astrodynamics specialist conference, portland, me*.

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22. doi:10.1109/mcse.2011.37