A Review of the Implementation of NumPy and SciPy Packages in Science and Math

Raed Waheed Kadhim [1], **Muna jaffar Raheem** [2], **Yasmin Makki Mohialden**[3] , **Nadia Mahmood Hussien** [4]

[1,2,3,4],Computer Science Department, College of Science, Mustansiriyah University, Baghdad-Iraq

raedwk.cs@uomustansiriyah.edu.iq [1]

munmnmun@uomustansiriyah.edu.iq[2]

ymmiraq2009@uomustansiriyah.edu.iq[3]
nadia.cs89@uomustansiriyah.edu.iq[4]

Abstract

In the Python programming language, there are a number of simple case studies of scientific computing. It gives you a multidimensional array object, a lot of organisms (like arrays and masked arrays), and many fast ways to work with arrays. SciPy, which is also called "Sigh Pie," is free math, science, and engineering software. The NumPy library is what the SciPy library is built on. This makes it easy and quick to work with arrays with N dimensions. The SciPy library is made to work with NumPy arrays in particular. It has a lot of easy-to-use and effective numerical methods, like scalar optimization and integration. They work well together, are free, and are easy to set up on all common operating systems. NumPy and SciPy are both easy to learn and use. This paper explains the most popular application of these packages in math-focused scientific research.

Keywords SciPy, Python, programming language, NumPy, statistics, data analysis

1. Introduction

NumPy is an extension of the popular Numeric Array object that aims to create a practical environment for scientific computing by introducing two basic components. These components are a global object, known as a Ufunc object, and an array object with N dimensions. The additional elements must be ready to work on top of any N-dimensional objects that are stored in a Ndarray. An array with n dimensions can be defined by the two basic pieces of information it contains. SciPy is an open-source library for scientific computing built on top of the Python programming language. Anyone can use it. Since its initial release in 2001, SciPy has established itself as the industry standard for implementing scientific algorithms in Python. It has more than 600 unique contributors who have written code for it, hundreds of packages that depend on it, more than 100,000 repositories that rely on it, and millions of downloads annually [1].
Numpy and Scipy both let you create algorithms for mathematical operations and machine learning [8]. The aim of this research is to present the most important applications that use these two functions in mathematics and science.

After the introduction, we briefly present the most important methods, previous sources, the application of these two functions, and the most important conclusions that were reached in recent years, Table 1 summarizes the functions of Numpy's sub-packages.

Table 1: This table gives a brief overview of the sub-packages that are part of the NumPy package.

Sub-package	Comments	purpose
core	All names exported to numpy	Basie objects
lib	All names exported numby	Other utilities
linalg	Old linear algebra from numeric	Base linear
fft	Old FFT from numeric	Discrete Fourier transforms
random	Old RandomArray from Numeric	Random number generators
distutils	Improvements built on distutils	Enhanced build and distribution
testing	Utility functions useful for testing	unit- testing
f2py	A useful utility needed by Scipy	Automatic wrapping of Fortran code

2. Related work

In this area, we show some studies that use NumPY and SCiPY packages: [Stefan van der Walt and others , 2011] showed that NumPy arrays are the usual approach to store numerical data and allow efficient implementation of numerical computations in a high-level language. This work shows that NumPy can be made more efficient by vectorizing calculations, reducing the amount of memory copying, and cutting down on the number of operations [4]. [Christian Bauckhage,2014] ranked different ways to use NumPy or SciPy to compute squared Euclidean distance matrices (EDMs). He talks about six code snippets that are progressively more difficult. The more difficult ones need more skill but are more effective. He closed our chat by advising when to employ each implementation [5].

[Christian Bauckhage explored K-meidos clustering , 2015] they described how to utilize NumPy to execute the technique. To show how this less-well-known method of clustering could be used in the real world, we talk about an application example in which we group strings based on how far apart their bi-grams are [6].

[M. Wood, R. Caputo, E. Charles, M. Di Mauro, J. Magill, and J. Perkins ,2017], Matthew Wood and others built a FermiPy tool based on the Fermi Science Tools, which NASA made public for the LAT project. Fermipy's high-level interface makes it straightforward and reproducible to evaluate LAT data. The current set of features includes techniques to collect spectral energy distributions and lightcurves, build test statistic maps, locate new source candidates, and fit source position and extension. Fermipy includes features from other scientific Python libraries including NumPy, SciPy, Matplotlib, and

Astropy. It's a community-developed package that follows open-source development [7].

[T. Bonald, N. de Lara, Q. Lutz, and B. Charpentier ,2019]Scikit-network analyzes huge graphs in Python like Scikit-Learn. SciPy's sparse CSR format stores graphs' adjacency matrices. The package includes techniques for ranking, clustering, classifying, embedding, and displaying network nodes. SciPy matrix-vector products, Cython-generated code, and parallel processing produce high performance. The BSD-licensed package requires NumPy and SciPy. Python 3.6 and later are supported. Online sources, documentation, and installation instructions [9].

[T. Gustafsson and G. D. Mcbain ,2020]Partial differential equations (PDEs) are the building blocks of modern physics and engineering. The Finite Element Method (FEM) discretizes and solves PDEs in complex spatial domains. The FEM translates time-independent PDEs into linear equations: Ax = b. Scikit-Fem is a lightweight Python library for creating A and B. The user imports a computational mesh chooses basis functions and supplies PDE's weak formulation. This creates SciPy-compatible sparse matrices and vectors[10].

[M. Chalela *et al* ,2011]They showed a regular grid search algorithm for quick nearest neighbor lookup with a fixed radius that was made in Python. This module indexes a set of k-dimensional points in a regular grid, with periodic conditions that can be turned on or off. This makes it easy to find the nearest neighbor. In this first part, they give you three different kinds of queries: bubble, shell, and nth-nearest. For these queries, they use three astronomy-related metrics: the Euclidean, the Haversine, and the Vincentian. The last two are given in spherical coordinates. They also give you the option to use your own distance function. This package gives useful results, especially for large datasets where a brute-force search wouldn't be practical. [11].

[C. R. Harris ,2020]Array programming provides a strong, compact, and expressive syntax for working with vectors, matrices, and higher-dimensional arrays. NumPy is Python's array library. It's used in physics, chemistry, astronomy, geoscience, biology, psychology, materials science, engineering, finance, and economics.

In astronomy, NumPy was utilized in the discovery of gravitational waves and the first black hole image. They discuss how the basic array of ideas leads to a simple and powerful programming paradigm for organizing, exploring, and analyzing scientific data. NumPy supports the scientific Python ecosystem. It's so popular that several programs have created NumPy-like interfaces and array objects. NumPy is a key part of the ecosystem, so it works as a layer of interoperability between array calculation libraries. Its API provides a flexible base for scientific and industrial analysis over the next ten years[12].

[P. LemenkovaT ,209]he goal of this study was to look at the data to see how different geological and tectonic factors might affect the way the Mariana Trench looks. Also, in marine geology and oceanography, the statistical analysis of the dataset needs a good plan for processing big data. In this case, the study recommends that GIS geospatial data analyses be performed using Python-based methods. The quantitative GIS part of the method led to a more accurate sample of the whole. a set of data with 25 cross-sections and 12,590 monitoring points for bathymetry. The second part of the methodology is about processing statistical data with the high-level programming language Python. The Pandas, NumPy, and SciPy libraries were used to do this search. Processing the data also makes sure that two more, disguised frames of data are taken from the large primary data set. The results showed that geological, tectonic, and oceanographic variables are all linked. With six full Python scripts [13]

3. Conclusions

Python is a powerful "routing" language that can be used with other languages that use scientific notation. On the other hand, this high-level language is great for writing code used in science and engineering. It is often fast enough to analyze data quickly enough to be useful right away, and it is flexible enough to add extensions that make it even faster. NumPy and SciPy can be used in many math and science programs.

Reviewer Comment

References

1. Oliphant, Travis E. A guide to NumPy. Vol. 1. USA: Trelgol Publishing, 2006.
2. R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, “CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations,”2017, p. 7.
3. P. Virtanen et al., “SciPy 1.0: fundamental algorithms for scientific computing in Python,” Nat. Methods, vol. 17, no. 3, Art. no. 3, Mar. 2020, doi: 10.1038/s41592-019-0686-2.
4. S. van der Walt, S. C. Colbert, and G. Varoquaux, “The NumPy Array: A Structure for Efficient Numerical Computation,” Comput. Sci. Eng., vol. 13, no. 2, pp. 22–30, Mar. 2011, doi: 10.1109/MCSE.2011.37.
5. Bauckhage, Christian. "NumPy/SciPy Recipes for Data Science: Squared Euclidean Distance Matrices." researchgate. net, Oct ,2014.
6. Bauckhage, Christian. "Numpy/scipy recipes for data science: k-medoids clustering." Researchgate. Net, February ,2015.
7. M. Wood, R. Caputo, E. Charles, M. Di Mauro, J. Magill, and J. Perkins, “Fermipy: An open-source Python package for analysis of Fermi-LAT Data.” arXiv, Jul. 29, 2017. doi: 10.48550/arXiv.1707.09551
8. J. Ranjani, A. Sheela, and K. P. Meena, “Combination of NumPy, SciPy and Matplotlib/Pylab -a good alternative methodology to MATLAB - A Comparative analysis,” in *2019 1st International Conference on Innovations in Information and*

Communication Technology (ICIICT), Apr. 2019, pp. 1–5. doi: 10.1109/ICIICT1.2019.8741475.

9. T. Bonald, N. de Lara, Q. Lutz, and B. Charpentier, “Scikit-network: Graph Analysis in Python,” p. 6.
10. T. Gustafsson and G. D. Mcbain, “scikit-fem: A Python package for finite element assembly,” J. Open Source Softw., vol. 5, no. 52, p. 2369, Aug. 2020, doi: 10.21105/joss.02369.
11. M. Chalela *et al.*, “GriSPy: A Python package for Fixed-Radius Nearest Neighbors Search.” arXiv, Dec. 23, 2020. doi: 10.48550/arXiv.1912.09585.
12. C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, Art. no. 7825, Sep. 2020, doi: 10.1038/s41586-020-2649-2.
13. P. Lemenkova, “Processing Oceanographic Data By Python Libraries Numpy, Scipy And Pandas,” *Aquat. Res.*, pp. 73–91, 2019, doi: 10.3153/AR19009.