Assignment-2 (Part-2)

Advancing Scientific Discoveries using NumPy and SciPy

Ajay Choudhury (18018)
EECS Department
IISERB

Introduction

Python is one of the most important tools in every scientist's toolbox that comes in handy in every field of his/her research.

Python being very easy to understand and having a low learning curve is user-friendly and easily adaptable which in turn has led to its huge popularity.

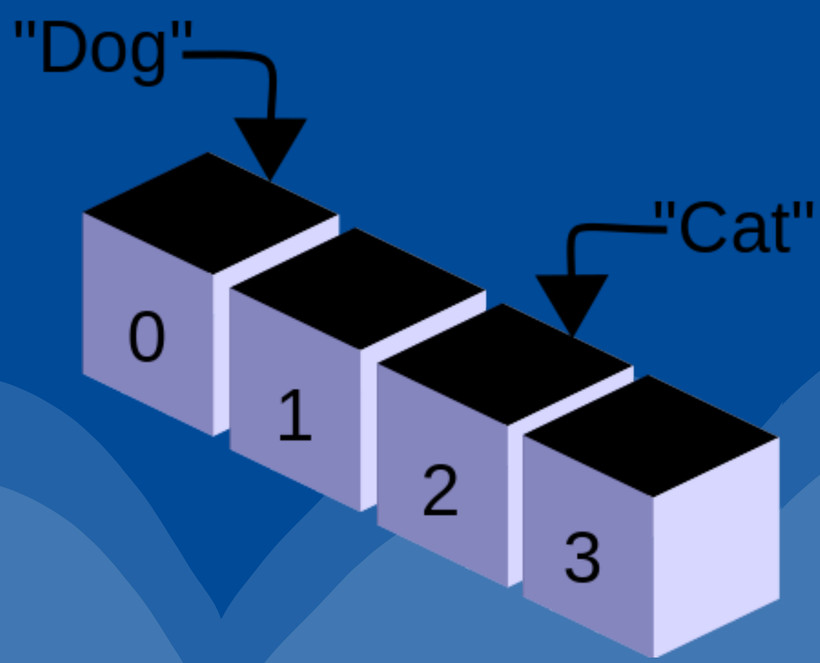
Being popular and carrying thousands and thousands of useful libraries and packages, it is now an indispensable language in any field, from market analysis to space probes, Python has got its feet down efficiently.

NumPy and **SciPy** lie at the core of all the scientific modules available in Python. They have gradually become the standards for scientific implementation in Python.

Implementation of NumPy using **C** programming language which makes it very fast and efficient in terms of memory. These are some of the prime reasons it has risen to be the standard in scientific-computing ecosystem.

Various standard libraries like **Matplotlib**, **Pandas**, **Scikit-Learn** and **Scikit-Image** use NumPy in their core.

Analysis of NumPy



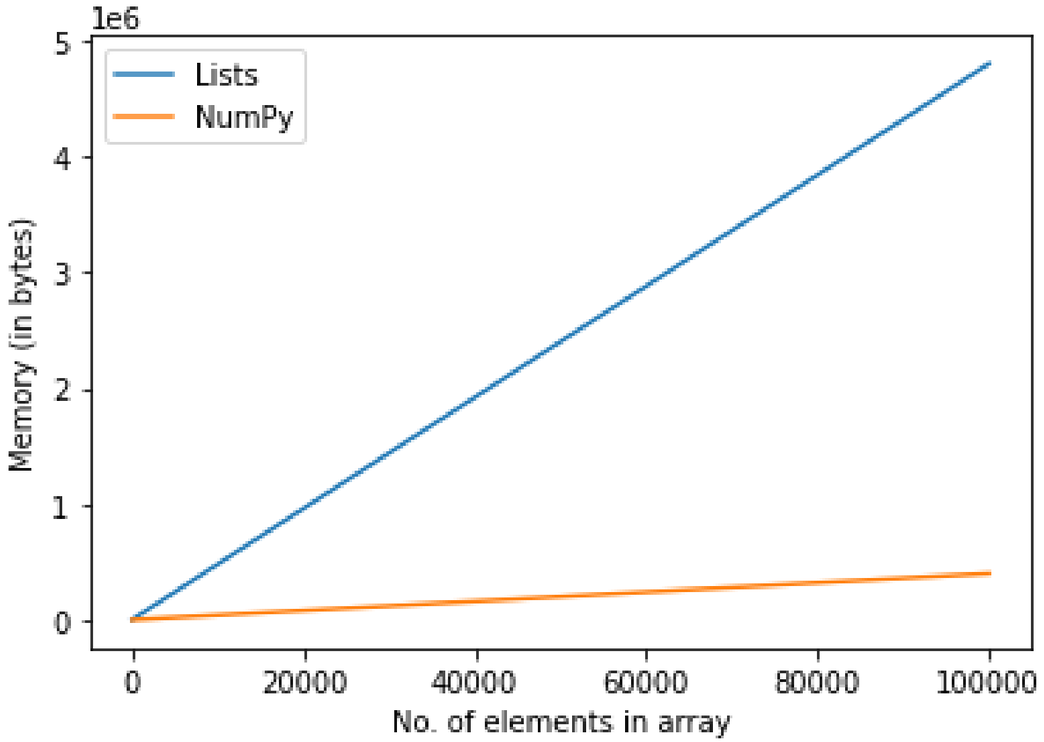
Comparing NumPy with Lists in Python

There are two basic factors upon which NumPy can be compared with Python's native Lists.

- **Speed**
- **Memory consumption**

We can further compare the speed of NumPy with that of Lists using the time module in Python and can get a graph of it as an illustration.

Comparison of Memory consumption



Native Lists in Python

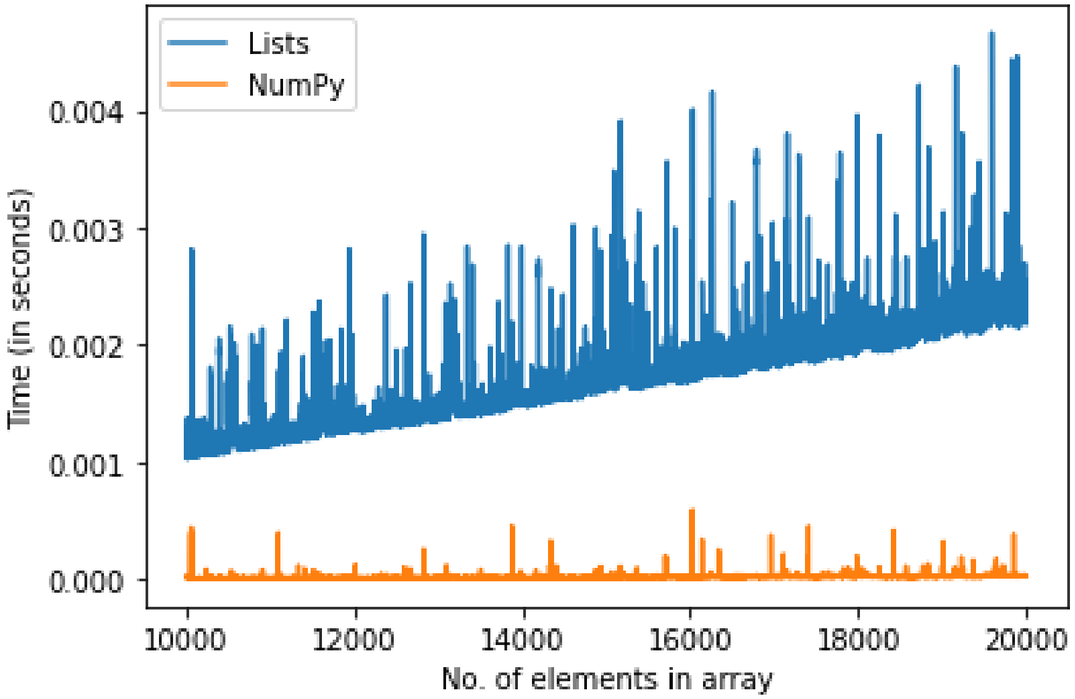
Lists consume considerably more memory than NumPy.

NumPy is more efficient than Lists

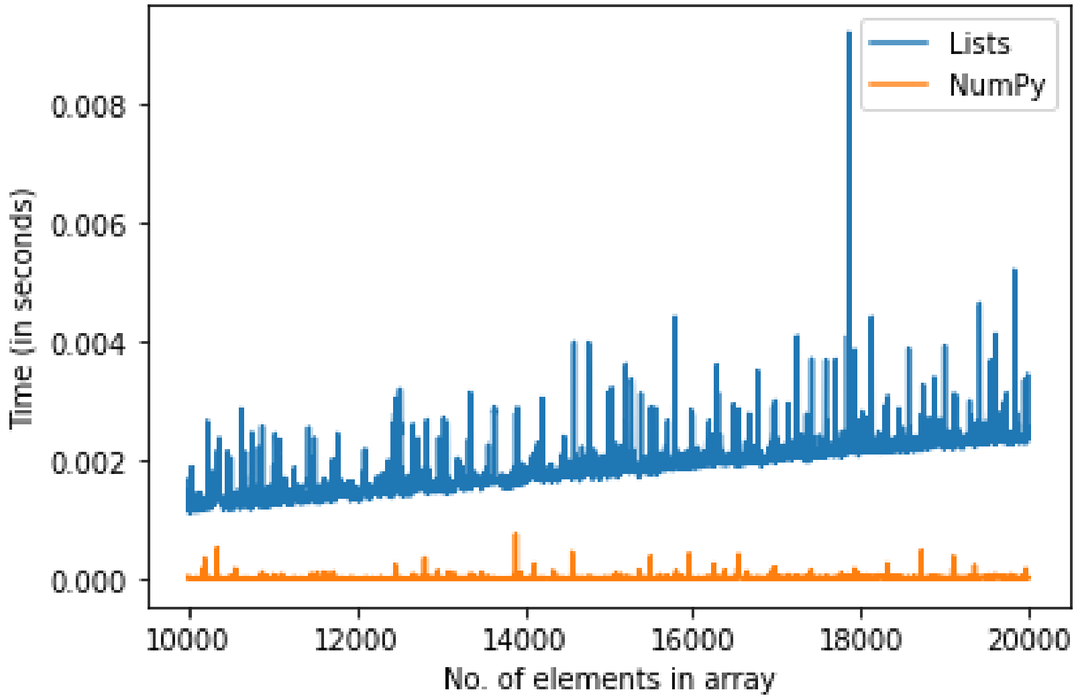
NumPy has a more efficient method to store data and occupies much lower space than lists.

Comparison of Speed of Operations

Lists in Python are comparatively slower in performing basic operations than Numpy. Here are a few illustrations to justify the fact:

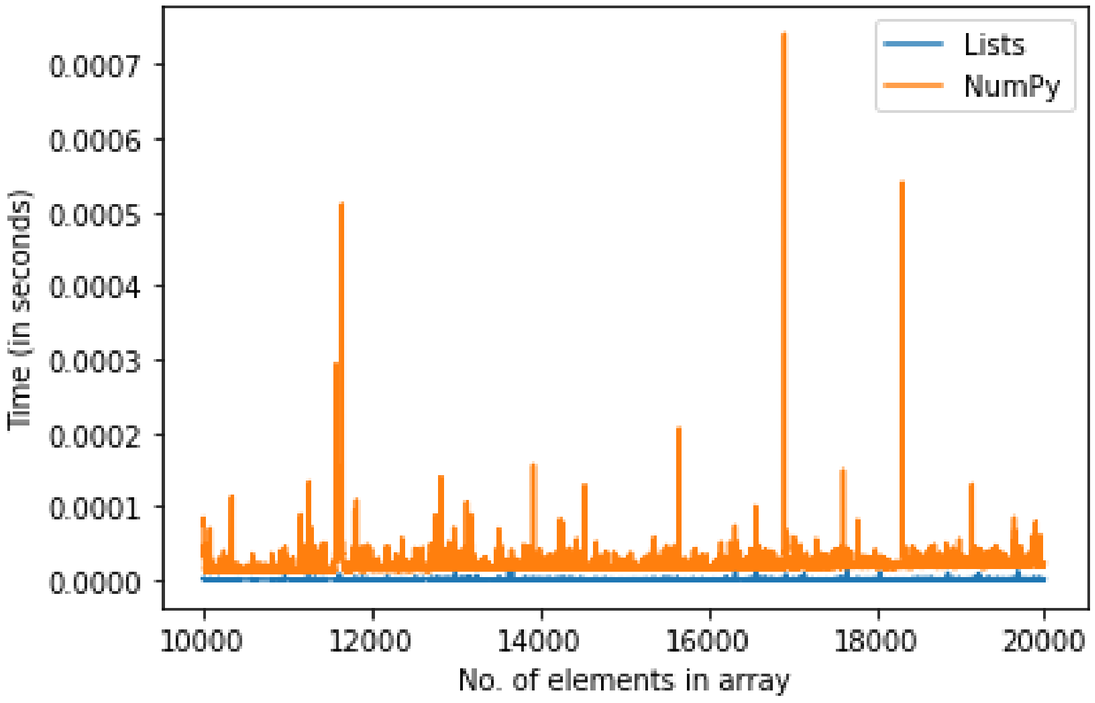


Sum Operation

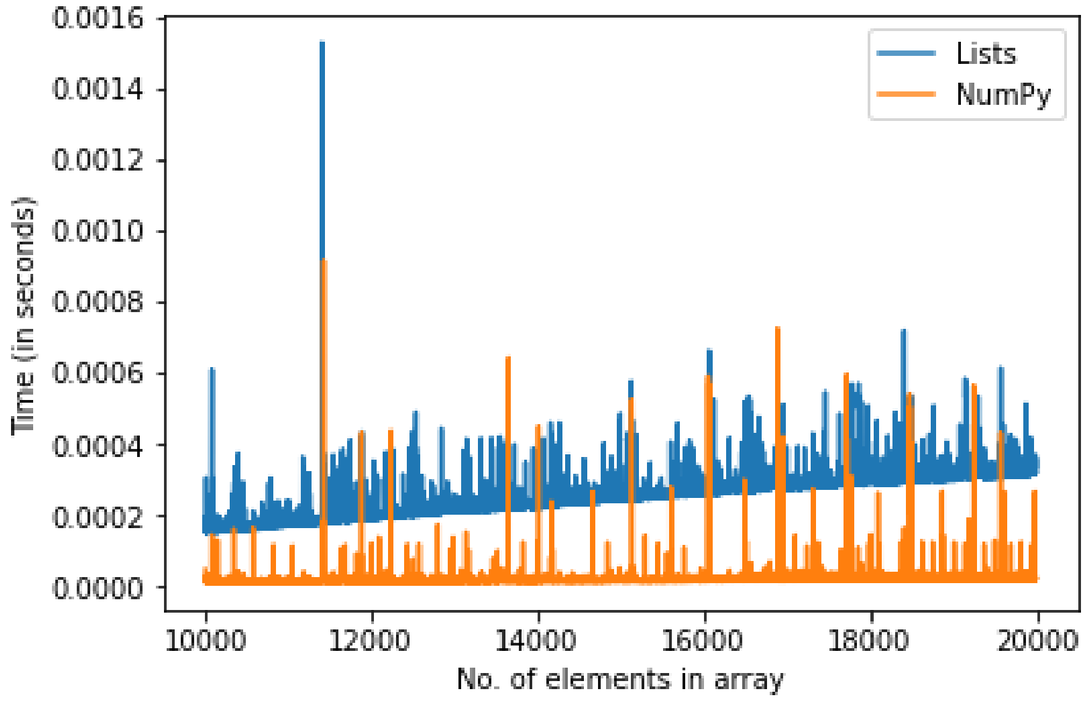


Multiplication

Comparison of Speed of Operations

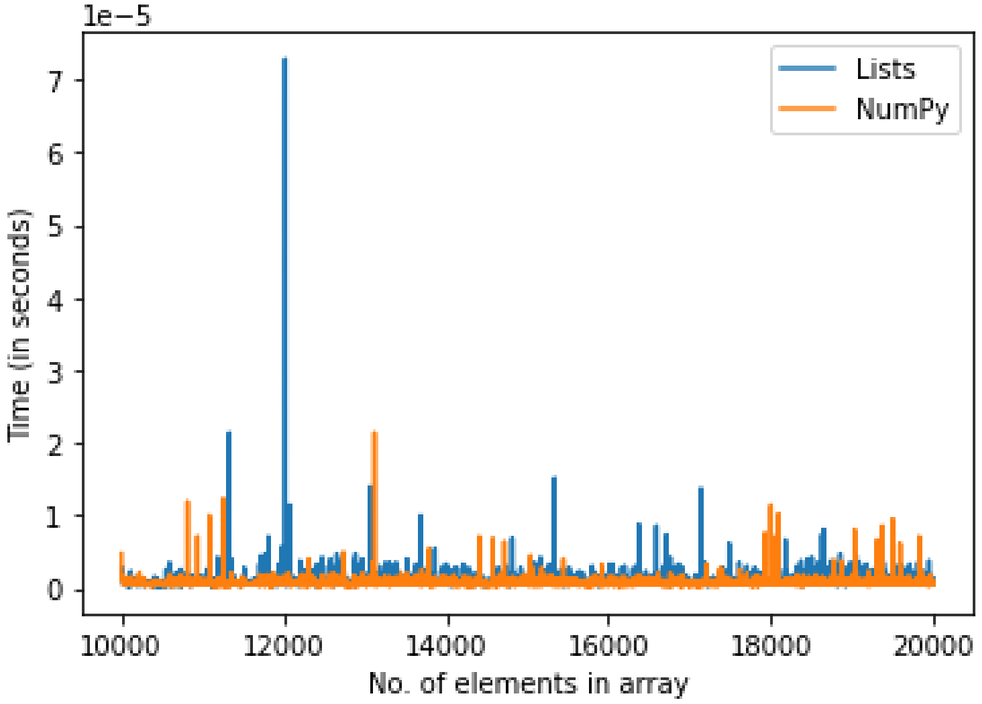


Appending

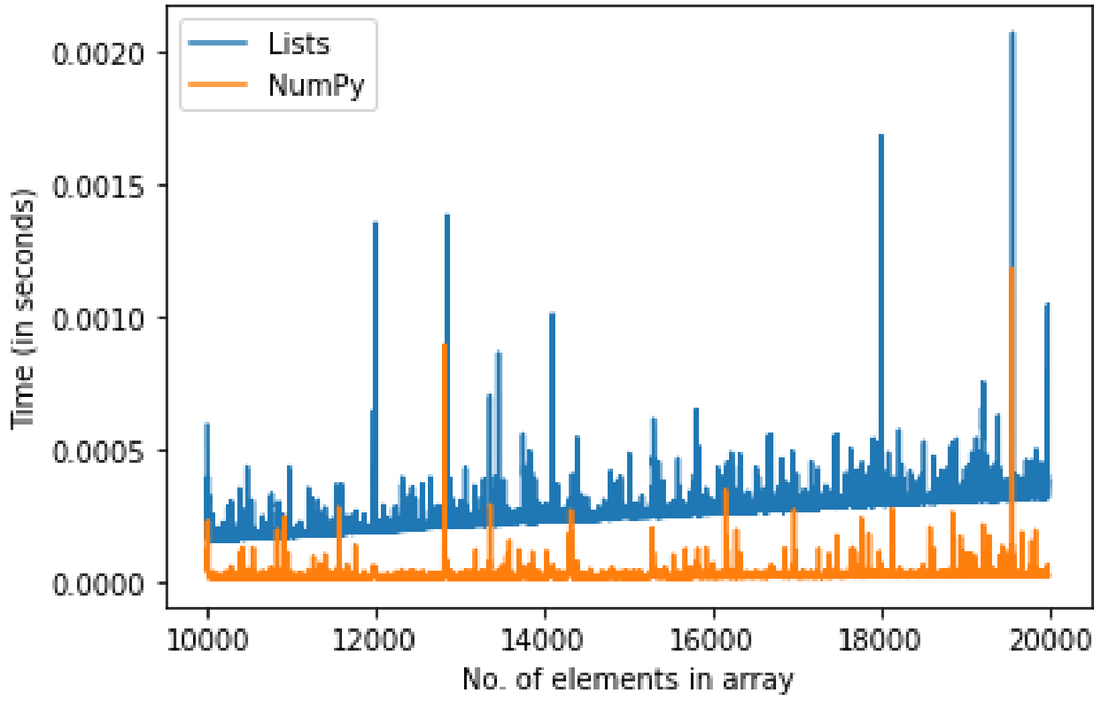


Searching

Comparison of Speed of Operations



Updation



Deletion

Unlike lists, NumPy arrays can be operated using arithmetic operations and it provides an efficient implementation of arrays.

But NumPy's append operation has a time complexity of O(n) whereas that of the list has O(1) making lists faster than NumPy arrays while appending elements.

For other tasks NumPy outperforms lists. With the development of modern data processing hardware such as GPU (Graphical Processing Unit), TPU (Tensor Processing Unit), FPGA (Field Programmable Gate Arrays). The need for NumPy to be compatible with such hardware has become essential for its sustenance in the future.

Discussion on SciPy

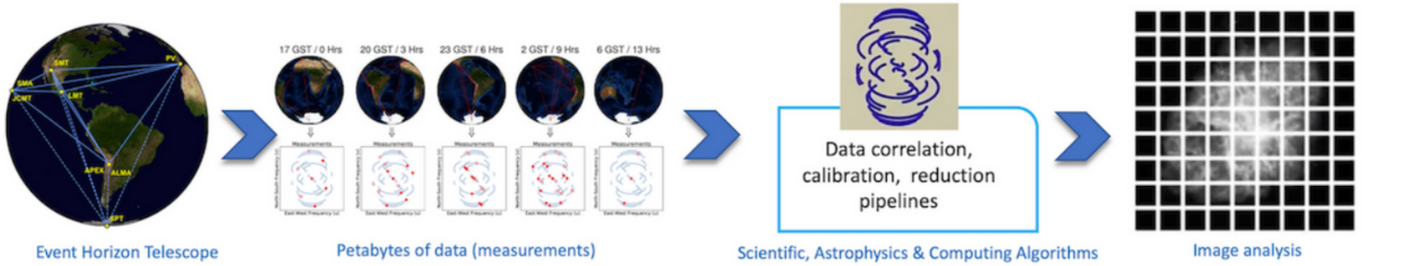
SciPy holds essential scientific algorithms and is built on top of NumPy.

It makes use of NumPy's fast numerical routines. Several Packages are build using SciPy, **Scikit-Learn** and **Scikit-Image** to name a few.

It has implementations for **image processing**, **statistics**, **machine learning**, **symbolic computation** and **network analysis**.

SciPy offers a one-stop solution for basic scientific computations for example Linear Algebra related algorithms.

SciPy : First Image of Black hole



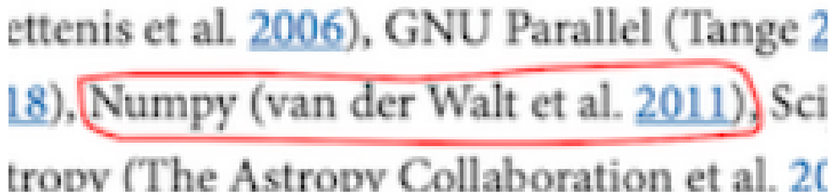
https://github.com/achael/eht-imaging/

Imaging, analysis and simulation software for radio interferometry

M87 – the first image of a black hole

Software: DiFX (Deller et al. 2011), CALC, PolConvert (Martí-Vidal et al. 2016), HOPS (Whitney et al. 2004), CASA (McMullin et al. 2007), AIPS (Greisen 2003), ParselTongue (Kettenis et al. 2006), GNU Parallel (Tange 2011), GILDAS, eht-imaging (Chael et al. 2016, 2018), Numpy (van der Walt et al. 2011), Scipy (Jones et al. 2001), Pandas (McKinney 2010), Astropy (The Astropy Collaboration et al. 2013, 2018), Jupyter (Kluyver et al. 2016), Matplotlib (Hunter 2007).

NumPy was one of the software used!


ettenis et al. 2006), GNU Parallel (Tange 2
18), Numpy (van der Walt et al. 2011), Sci
tropy (The Astropy Collaboration et al. 2(

A large amount of data needed to be collected and decisions need to be made on which data to use and which not.

Those pipelines were built on **pandas** and **SciPy**, the standard astronomical file formats and time transformations were handled by **Astropy**.

For plotting **matplotlib** module was used. All of these modules are built upon NumPy's arrays.

Conclusion

Interactive programming practices such as IPython and Jupyter Notebook has enabled further growth of python.

The emergence of GitHub and has accelerated the development of open-source packages such as **NumPy** and **SciPy**.

Its role towards accelerating scientific research is unquestionable, but to be sustainable in the future it will need to accommodate modern hardware architectures such as GPU and TPU.

The Scientific community focused on numerical computing have long neglected python due to its lack of speed, with the advent of Numba and further improvements in speed shall make python more favourable to the numerical computing community

References

01. Array programming with NumPy. Nature, 585:357–362, 2020.

02. Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17:261–272, 2020.