Advancing Scientific Discoveries with NumPy and SciPy: A review

Name:	**Dinesh Adhithya**
Registration No./Roll No.:	18097
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS

1 Introduction

Python has become an important tool in every scientist's toolbox.What makes python tick is it's small learning curve , user friendly syntax . Its versatile nature makes it indispensable , from being used in kinematics to astrophysics its efficient and easy to use modules make up for python's case. At the heart of all that is NumPy[1] and Scipy[2] which is used to build various modules in python. NumPy[3] has become python language's standard package for array implementation. It's speed and efficient memory consumption are prime reasons for its position and is now used to build various packages such as Pandas , Matplotlib , Networkx , Scikit-Learn and Scikit-Image cementing its place at the heart of python scientific computing ecosystem. Along with SciPy , python's standard library for scientific computation which holds efficient implementation of various scientific algorithms.

2 Analysis: NumPy

We understand python's central role in scientific research , NumPy being at the heart of many modules . In the below section we compare NumPy arrays with lists in python.Which would clearly show the importance of NumPy.

2.1 Memory

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

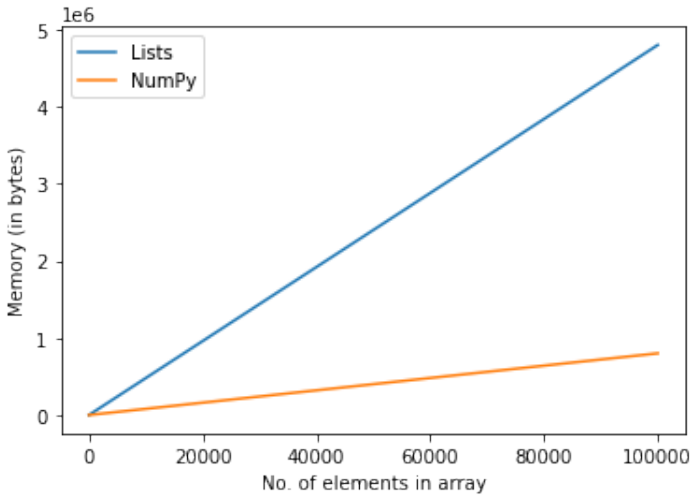

Figure 1: Figure showing the amount of space a NumPy array occupies to that of an list

2.2 Speed

Below we discuss the time taken for various fundamental operations by NumPy arrays and compare it with lists in python.

2.2.1 Multiplication

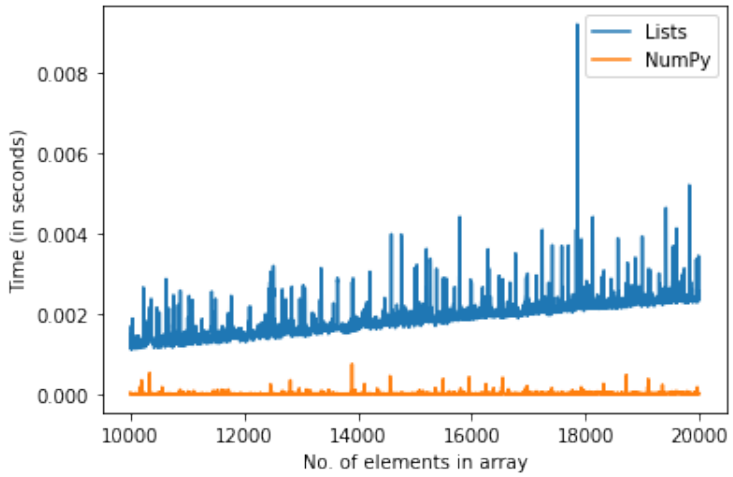

Figure 2: Figure showing the amount of time a NumPy array takes to that of an list to compute multiplication of an two arrays

2.2.2 Summation

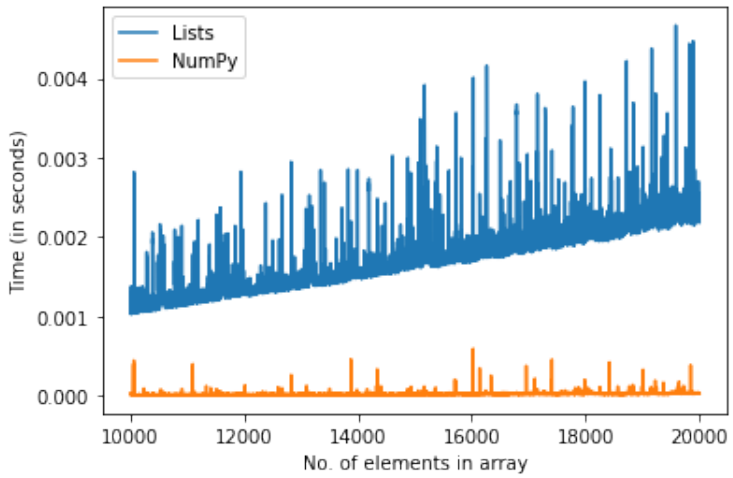

Figure 3: Figure showing the amount of time a NumPy array takes to that of an list to compute sum of its elements

2.2.3 Append

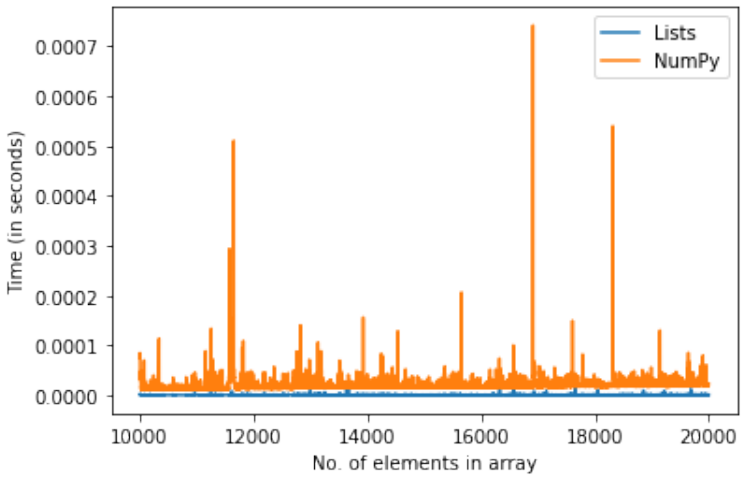

Figure 4: Figure showing the amount of time a NumPy array takes to that of an list to append a element at the end of an array.

2.2.4 Searching

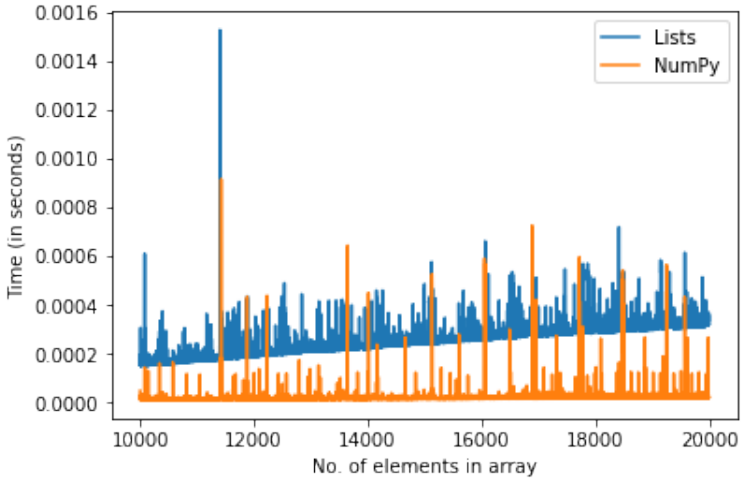

Figure 5: Figure showing the amount of time a NumPy array takes to that of an list to search a particular element

2.2.5 Deletion

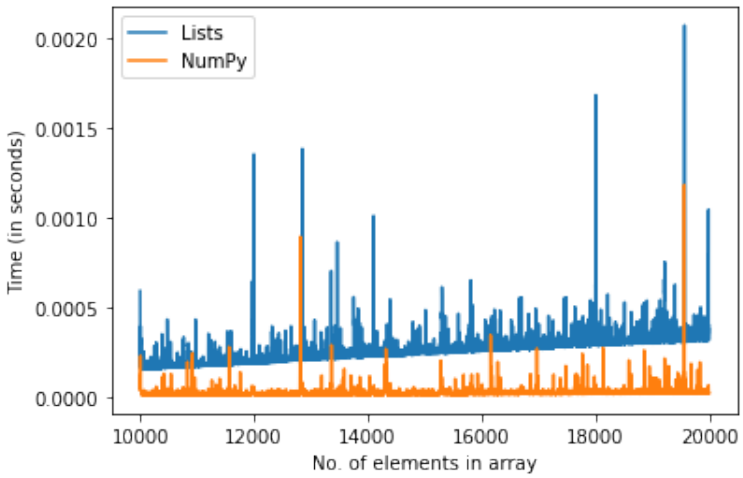

Figure 6: Figure showing the amount of time a NumPy array takes to that of an list to search a particular element

2.2.6 Update

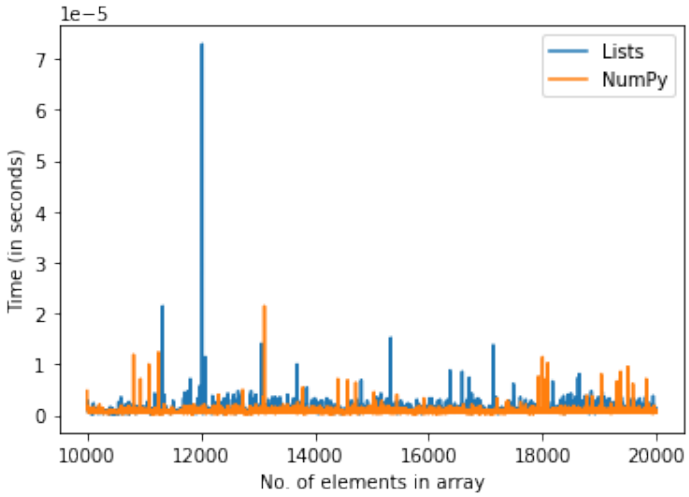

Figure 7: Figure showing the amount of time a NumPy array takes to that of an list to update a particular element

2.3 Discussions

2.3.1 NumPy

Unlike lists NumPy arrays can be operated using arithmetic operations and it provides efficient implementation of arrays. But NumPy's append operation has a time complexity of O(n) whereas that of list has O(1). Making lists faster than NumPy arrays while appending elements.For other tasks NumPy outperforms lists. The contribution of NumPy to the scientific community is of the highest value , in terms of providing a fast yet also pythonic syntax for its implementation. Yet , 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. NumPy versatile architecture allows users to process sparse arrays and Distributed arrays.

2.3.2 SciPy

SciPy package holds essential scientific algorithms and is built on top of NumPy and makes use of NumPy's fast numerical routines. Several Packages are build using SciPy , Scikit-Learn and Scikit-Image to name a few. It is quite diverse in its functionality , it has implementations for image processing , statistics , machine learning , symbolic computation and network analysis . Although there are specialised packages for each of the above task SciPy offers a one stop solution for basic scientific computations. Linear Algebra related algorithms are efficiently implemented in SciPy and offers complete coverage in this domain.

3 Case Study: First Image of a Black hole

With a large amount of data collected and decisions need to be make on which data to use and which not. Those pipelines were build on pandas and SciPy , the standard astronomical file formats and time transformations were handled by Astropy , for plotting matplotlib module was used. All build on NumPy's arrays.[1]

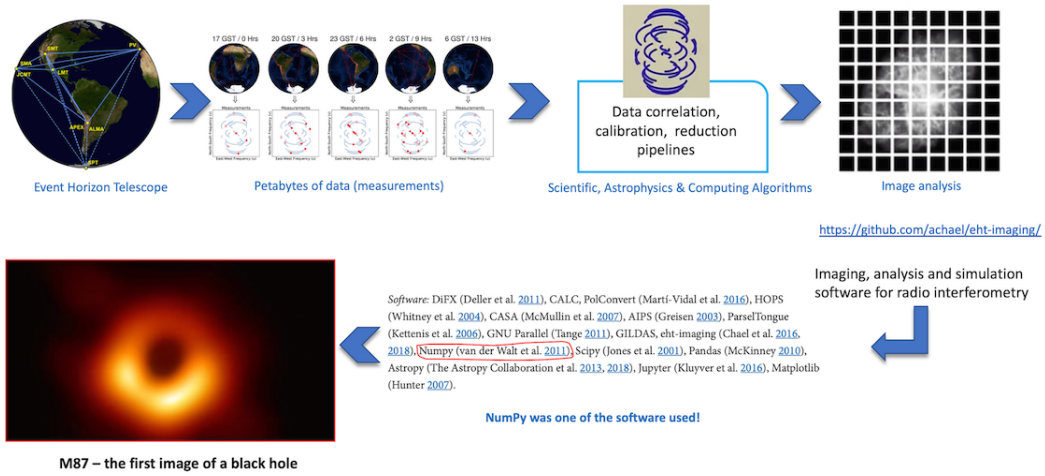

Figure 8: Figure showing the role played by NumPy package while developing the first image of a black hole.

4 Conclusion

With the advent of interactive programming practices such as IPython and Jupyter Notebook has enabled further growth of python , with emergence of GitHub and robust software development practices , has accelerated 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 its lack of speed , with advent of Numba and further improvements in speed shall make python more favourable to numerical computing community

References

[1] Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020.

[1] https://numpy.org/case-studies/blackhole-image/

[2] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[3] Travis E. Oliphant. Python for scientific computing. *Computing in Science Engineering*, 9(3):10–20, 2007.