Lecture 3: Lists, arrays, and optimisation with NumPy

Dr Benjamin J. Morgan[1] and Dr Andrew R. McCluskey[1,2]

[1] *Department of Chemistry, University of Bath, email: b.j.morgan@bath.ac.uk*
[2] *Diamond Light Source, email: andrew.mccluskey@diamond.ac.uk*

July 16, 2019

Aim

This lecture will introduce lists, arrays, and show how the NumPy library can be used to make your code faster.

1 Lists

The Python programming language natively includes the ability to group together a series of objects. These are `lists` and are one of the most powerful Python objects. Lists are an ordered set of objects, from which it is possible to pick all, one, or many values. A list is defined as follows,

```python
# Making a list

elements = ["Hydrogen", "Helium", "Lithium", "Beryllium",
    "Boron", "Carbon", "Nitrogen", "Oxygen"]
```

Having defined the list, it is then possible to select individual items of the list by using the following syntax,

```python
# Printing some items

print(elements[0], elements[4], elements[-1])
```

Note, that Python starts counting from the number 0, and using the minus sign we can ask Python to count from the end. This means that the above code should print, `"Hydrogen", "Boron", "Oxygen"`. This counting from 0 means that in the above list, the string `"Hydrogen"` would be referred to as the zeroth object in the list, while `"Helium"` would be the first.

In addition to making use of single objects from within a list, it is also possible to create sublists, for example,

```
# Just the first 4 elements

print(elements[0:4])
```

Note that above, the numbers on either side of the colon the list indices. However, rather strangely, the sublist created is **inclusive** of the first number and **exclusive** of the second. Additionally, it is possible to select non-consecutive objects from a list by placing commas between the indices,

```
# Just the gases

print(elements[0, 1, 6, 7])
```

The final point about `lists` is that the data that they hold does not all need to be the same time. For example, the list below contains a `float`, two `str`, a `complex` number and an `int`,

```
# List of many types

a_new_list = ['hello', 12.41242, 5 + 8j, 'sadness', 2]
print(a_new_list)
```

2 NumPy Arrays

NumPy (or `numpy` or more commonly `np`) is a library that Python can use that is designed and optimised for doing numerical operations.[1] Over this course you will be introduced to many other Python libraries, in order to use any of these you must `import` them,

```
# Import NumPy

import numpy as np
```

This asks the Python interperator to go and find the NumPy library, then in order to reduce the amount of typing (programmers are lazy), we give the library the alias `np`.

One of the most powerful features of the NumPy library is the `array`, these are similar to lists but with some important differences.

References

[1] T. E. Oliphant, *A guide to NumPy*, Trelgol Publishing USA, 2006, vol. 1.