Science & Technology Facilities Council | UK Research and Innovation | NERC SCIENCE OF THE ENVIRONMENT

Read and Write Data

Working with data: binary formats

Why use a binary format?

Binary formats have the following **advantages**:

- They are very storage efficient (up to 10x that of text formats).
- Reading/writing data is typically faster.

And some **drawbacks**:

- They may not be portable between computing platforms.
- They are not human-readable.

We can create binary formats

It is easy to create binary format using Python, e.g.:

```
>>> import struct, random
>>> n = 100000
>>> my_data = [float(i) for i in range(n)]
>>> random.shuffle(my_data)
>>> binary_data = struct.pack("f" * n, *my_data)

>>> with open("output.dat", "wb") as writer:
...     writer.write(binary_data)
```

But should we? There are many reasons not to

Why shouldn't we create another binary format?

- Every *new format* requires documentation and code to support its usage.
- No software exists to automatically plot or analyse the *new format*.
- Other scientists will not know what to do with data in this *new format*.

We strongly recommend you use NetCDF.