

FYSS5120 Efficient Numerical Programming - Demo 2

Drop solutions before the demo session to the Nexcloud box ([link](#))

Please indicate clearly your name in the file name, so that I can tell who solved what. Commented Python scripts (.py or .ipy) and commented Jupyter Notebooks (.ipynb) are fine.

1. Time NumPy `sum()` method vs. Python `sum()` function.

```
import numpy as np
N = 10000000
A = np.random.random(N)
```

and compare the execution times of `A.sum()` and `sum(A)`.

2. Write a Python code that computes the distances of 1000 particles in three-dimensional space. The particle coordinates are in the NumPy `NxD` array `x`.

Time the five ways to compute particle distances using, for example, `time.perf_counter`:

- (a) NumPy broadcasting used in sample code `potential_simple.py`, in Python examples directory ([link](#))

```
d = x[:,np.newaxis,:]-x
r = np.sqrt((d**2).sum(2))
rs = r[np.triu_indices_from(r,1)]
```

- (b) A version that uses `numpy.linalg.norm()` to compute `r` in the previous code. Link to documentation: [numpy.linalg.norm](#)
- (c) Compute `r` using `numpy.einsum()`

```
r = np.sqrt(np.einsum('ijk,ijk->ij',d,d))
```

- (d) A version that uses `scipy.spatial.distance.pdist`. Link to documentation: [scipy.spatial.distance.pdist](#)
- (e) A basic `for`-loop, but with Numba `@jit` decorator.