Homework 2

In this homework, you will implement two classes for taking spatial derivatives. The classes will both be in the `finite.py` file. The `DifferenceUniformGrid` class can take derivatives of functions defined over a `UniformPeriodicGrid`, and the `DifferenceNonUniformGrid` class can take derivatives of functions defined over a `NonUniformPeriodicGrid`. In the former case, the grid is defined by the number of grid points `N` and the period `length`. The non-uniform grid is defined by the `values`, which are a monotonically increasing `numpy array` of grid points. The first grid point is assumed to be zero.

Difference classes are instantiated with the desired derivative order and convergence order. In all cases, you will construct centered difference operators (i.e., to estimate the derivative at x_i, you will use as many points to the right of x_i as to the left of x_i). We will make non-centered difference operators later in the course. For instance,

```
d2 = finite.DifferenceUniformGrid(2, 2, grid)
```

would make a second-order accurate, second derivative operator associated with `grid`, which is a `UniformPeriodicGrid`.

Your objective is to construct a `matrix` attribute which can be used to take derivatives. If `f` is a `numpy array` containing the data $f(x_i)$ for some function f, then we will estimate the second derivative of f using

```
d2f = d2.matrix @ f
```

where the `@` operator denotes a dot product. To look nicer, I've set up code that will compute this dot product with `d2 @ f`.

To construct the derivative matrices, you first need to determine a stencil that corresponds to the action of the derivative around a point x_i. The size of the stencil will depend both on the specified derivative and convergence orders. Then you should create a sparse matrix using the stencil. *Note:* The stencil sizes are not necessarily the same for the difference operators on uniform vs. non-uniform grids. Also, because we are limiting ourselves to centered difference schemes, not every order of convergence will be possible.

In addition to the tests that come with the homework, I encourage you to test your code by taking the derivative of simple functions numerically and analytically, and verifying that the error decreases as the grid spacing decreases with the expected convergence order.