Tobias Neckel Max-Planck, May 2019

Python Tutorial 7: OS Functions in Python

1 Managing your simulations with Python

In this tutorial we will use the packages `os` and `configparser` to run a large number of simulations and store the results in an orderly manner. Let's dive right in!

1.1 Solver

Write a Python script called `mySolver.py` containing a single function `mySolver(x, normalize)`. `x` will be a vector (1D *NumPy array*) and `normalize` a boolean. The function should do the following:

1. Create a square matrix `A` with random entries. You can use the function `numpy.random.rand(s,s)`, where `s` is the size of the vector `x`.
2. Perform the dot product of `A` with `x` using the function `numpy.dot(A,x)`.
3. If the variable `normalize` is equal to `True`, the result of the dot product should be normalized. This can be done with the function `np.linalg.norm(b)`.
4. Return the resulting vector.

1.2 Configuration file

Now create a settings file called `config.ini`. This file should contain only one section called `parameters` which should have three values `n`, `size` and `normalize`. `n` will be the number of simulations we will run; `size` will be the size of our vector x; and `normalize` will tell us whether our solver should normalize the result of the dot product. Set their values to `10`, `100` and `True` correspondingly.

1.3 Simulation manager

Now create a script called `runSimulations.py` and implement the following tasks:

1. Import the modules `os`, `configparser`, `numpy` and `mySolver`.
2. Define a function called `prepareSimulations(basename, n)`. This function should create `n` folders in the current directory using `basename` as the prefix for each folder name plus a unique number from 0 to $n-1$.
3. Finally, write a function `run()` that does the following:
   - It creates an object of the class `configparser`, which then reads in the values of `n`, `size` and `normalize` from your `config.ini` file. Don't forget to cast the values!

- It initializes a 1D *NumPy array* with `size` random entries.
- It calls the function `prepareSimulations` with a `basename` of your choice. Alternatively, you can add `basename` as a value in your `config.ini` file.
- For each of the `n` folders starting with `basename`, it enters the folder, calls your solver, writes the resulting vector in a file called `result.txt` and goes back to the original folder.

Now call your function `run` and play around with the parameters in the configuration file to make sure everything works.