Week 3: Python brush-up #2

02807 Computational Tools for Data Science

Exercises

1 Numpy package Install the numpy package and look at `http://www.numpy.org/` for help.

Write a python script which reads a matrix from a file like this one `https://pastebin.com/XAhwshXe` and solves the linear matrix equation $Ax = b$ where b is the last column of the input-matrix and A is the other columns. It is okay to use the solve()-function from numpy.linalg. Does the result make sense?

2 SciPy package Install the scipy package and look at `https://www.scipy.org/` for help.

Write a python script that reads in this list `https://pastebin.com/ENyYffaq` of points (x,y), fits/interpolates them with a polynomial of degree 3. Solve for the (real) roots of the polynomial numerically using Scipys optimization functions (not the root function in Numpy). Does the result make sense (plot something to check).?

3 Numba package Install the numba package and look at `https://numba.pydata.org/` for help.

Write a simple Python function for computing the sum $\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \ldots$ with 10,000 terms (this should be around 1.644), 2000 times in a row (to make the execution time measurable). Now use numba and see how much speedup you can achieve by this (you can use the time command in the terminal for this). How much faster can you make the Numba version?