Condition number and error analysis for linear systems

Write a Python script `ErrorAnalysis.py` that builds a matrix $\boldsymbol{A}$ with the following elements:

$$A_{ij} = \frac{(-1)^{i+j}}{i+2j},\ i,j = 1,\ldots,N$$

and let $\mathbf{c}$ be the first column of $\boldsymbol{A}$.

In the same script, compute the condition number of $\boldsymbol{A}$, for $N = 2,\ldots,10$, and solve

$$\boldsymbol{A}\mathbf{x} = \mathbf{c}$$

using LUP-decomposition (you can use `scipy.linalg.solve` function). The exact solution of this system of equations is $\mathbf{x} = \mathbf{e}_1$ (check, without solving the system explicitly, that this indeed must be the case). Compute the relative residual and error for all matrix sizes $N = 2,\ldots,10$.

Plot the relative error along with the maximal relative error (according to the theoretical upper bound discussed in Lecture 7) on a semilogarithmic scale (use `matplotlib`).

Push your Python script to your GitHub Classroom repository.

For discussion:

1. What happens to the condition number for increasing N? Is that bad news or good news when we are solving linear systems?
2. What happens to the relative errors and residuals for increasing N?
3. Up to what matrix size does the numerically obtained solution make sense?

One member of your group should post your group's discussion with figure on the Slack #tutorial-4 channel.