Numerical Linear Algebra
Master on Fundamentals of Data science
Universitat de Barcelona

Project 1 Corrections
NLA: direct methods in optimization with constraints.

Blai Ras

December 5, 2020

Note: this is the Blai Ras corrections of the project one of NLA. This project was coded between Pablo Alvarez and me, Blai Ras.

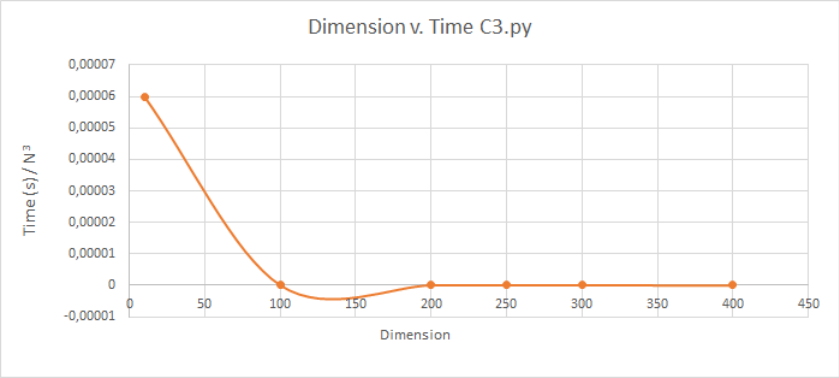
1 Feedback

1.1 You state that time increases exponentially with respect the dimension and no justification is provided. Have you perform a fit?

The mistake here I made was assuming that the time was growing exponentially without investigating any further. After looking at the graph I made and thanks to your appointment, I realized we can define this behavior as:

$$T \approx C \cdot N^3$$

Where T is the execution time, C is a constant and N is the dimension of our problem. To prove that, I isolated this constant and plot its value for every dimension and execution time of my previous tests:



We can see that this constant stabilizes around $\approx$ 5e-08 and therefore prove the previous equation, rather than saying that "grows like an exponential".

1.2 Report condition number for different strategies with respect to iterates of the Newton method

In the Condition Number section I reported different condition numbers at the first iteration of each code. I will extend this investigation performing the following steps:

- Gather all the condition numbers at each iteration.
- Compute the mean of all this condition numbers and set a unique condition number to a defined dimension.
- Plot the results to look at the behavior of this condition number when changing the dimension of the problem.

Therefore:

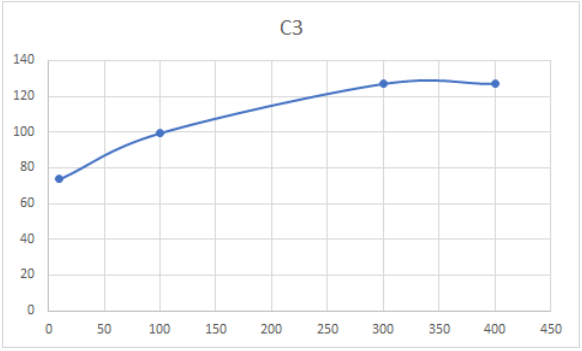


Figure 1.1: In C3.py we can see an incrementation of the condition number, stabilizing around 300-400 iterations

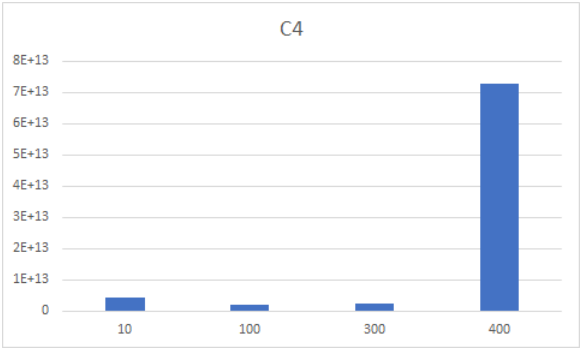


Figure 1.2: In the first strategy of C4, we see a weird behavior: it decreases from dimension 10 to 100 and stabilizes between 100 and 300. Nevertheless, it suddenly grows when inputting dimension 400. That's why I opted for a bar plot.

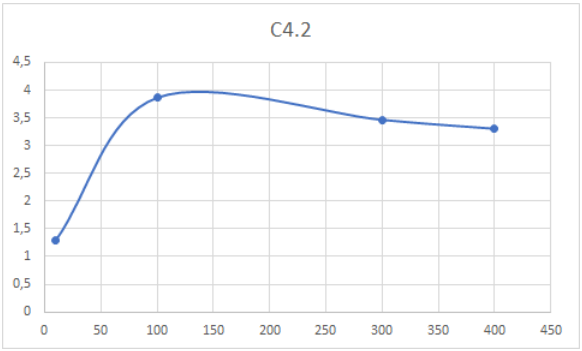


Figure 1.3: In the second strategy, we see much lower condition numbers. It stabilizes in general, it's hard to change the condition number by changing the dimension.

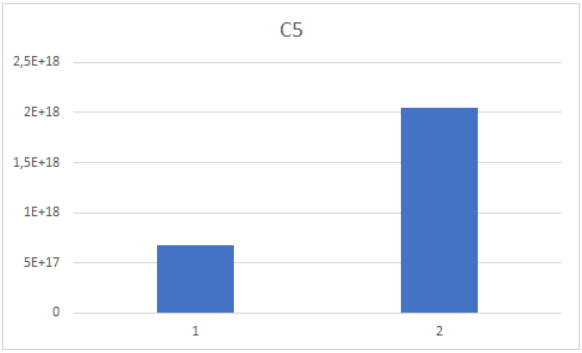


Figure 1.4: In the C5.py, when using the optrpr1 we see a much lower condition number than when using optpr2

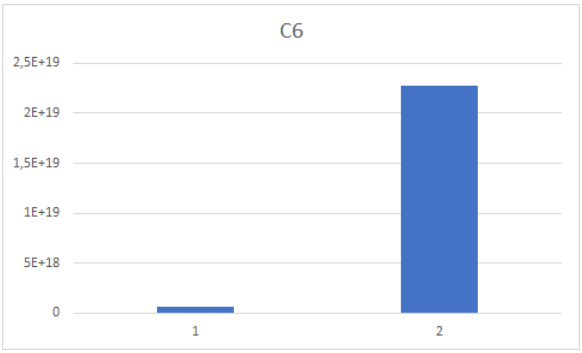


Figure 1.5: Same happens with C6.py, but this time the difference is higher