

Homework linear equations

I have done all three parts of the exercise, using randomly generated matrices.

WARNING int part C I EXPLICITLY CALL I explicitly call `/usr/bin/time`, rather than `time`!!! this is because zsh overwrites time with its own (far inferior) version. IF YOU HAVE NOT INSTALLED THE TIME UTILITY IN THAT PLACE, THIS WILL FAIL.

Part A

Output, text file outA.txt, with results from basic random tests.

In part A I look at a 6 by 3 matrix and find the QR decomposition, I repeat this 1024 times. I also solve random linear equations on a 6 by 6 matrix, I also repeat this 1024 times, in all cases I check for any errors and find none.

Part B

Output, text file outB.txt, with results from basic random tests.

Here I test that the inverse is working.

Part C

Output, two png images `C_time_GNUmethod.png` and `C_time.png`. Both show the time cost of QR factorization using the C# build in timer, or the posix time utility.

I do part C twice, first method, use system clock internaly C# methods for timing ONLY the decomposition, advantages: compact, just one program with a for-loop inside, the setup of the matrix is NOT counted, and the internal clock has higher precision (milli seconds range). Disadvantage, if other programs run at the same time, the result will be effected.

Second method, (`C_time_single.cs`) I just set up and run ONCE, timing with `time`) I repeat this for as many matrix sizes as before. the time utility has WORSE precision (output is limited to 10 milli seconds)

WARNING, I explicitly call `/usr/bin/time`, rather than `time`!!! this is because zsh overwrites time with its own (far inferior) version

I test this (both ways) for matrices from 2x2 to a 300x300 .

In either version of C, I fit this to a $a x^3 + b$ function, this is done using my result from the next homework, which is included as `fit.cs`, I believe from that homework that this function is working.

The fits both work, showing that the function runs in the expected time (I did not really do a χ^2 test or anything, it just looks fine I guess) The second method with `time` is worse, because the time utility has worse resolution has

a noticably larger offset and slightly different coefficient a , likely due to time wasted reading from the input and setting up the matrix.