Programming #4 Comparing interpreted and compiled codes

In this assignment, you will build three programs which perform Gaussian elimination with back substitution. You will use system calls to measure time to capture the time it takes to run the programs with different data sizes. Make sure you use an appropriate measure tool that give you reasonable time granularity. This will allow us to compare interpreted and compiled languages. Your program shall NOT do partial pivoting. If you use partial pivoting, you will have 50% off this project (this will prevent you from just copying solutions on the internet).

CHANGE OF REQUIREMENT. Python+NUMPY can use pivoting as LU decomposition does this by default. Your other 2 implementations should not have pivoting.

Also, FORTRAN on your domain is invoked with "gfortran"

Useful LINKS

https://labmathdu.wordpress.com/gaussian-elimination-without-pivoting/ (https://labmathdu.wordpress.com/gaussian-elimination-without-pivoting/)

Languages: FORTRAN, Python

Setup: Each program will take a single input, the size of the Matrix, N. Your program will allocate and populate the matrix using random numbers. Your program will then start the clock. Run Gaussian Elimination and back subsitution. And then take the stop time. Your program will output the time.

Task: Create Gaussian elimination with back substitution.

Input: Size of square matrix. Size should be 250, 500, 1000, 1500, 2000

Internals: Explicitly or implicitly allocate sufficient memory to a Nx(N+1) floating point Matrix, (or NxN) + N matrices for NUMPY
using a random number generator -- populate the Matrix.
Perform Gaussian elimination and back subsitution on the Matrix
Your routine should have no output other than the runtime

Your python implementation MUST be with NUMPY and also without NUMPY.

Externals: You MUST run each experiment 5 times for each size. This means , you should have 5 runs x 5 sizes x 3 implementations =75 data points.