PHYS 511: Computational Modeling and Simulation - Fall 2018
Assignment #1, due Friday October 26, before class

Getting familiar with Fortran, LAPACK, and plotting software

Write a Fortran program that solves a system of linear equations

$$Ax = b, \tag{1}$$

where A is a real random matrix of size $n \times n$ and b is a real random matrix of size $n \times 1$ (i.e. b is a vector). The random values should be taken from the interval $[0, 1]$. You will need to find an appropriate LAPACK subroutine that computes the solution of the system of linear equations. There may be several options for this, please think which one is the most optimal from the computational viewpoint. Use double precision for all matrices/vectors in your program.

Make a loop in your program over the following set of n values: 100, 200, 300, 400, . . . , 3000. Measure the time it takes to solve the system of linear equation for each n. Print this time on the screen vs matrix size, e.g. make the output in the column format:

```
100  <time>
100  <time>
  ⋮
3000 <time>
```

Make sure that in your loop over the given values of n you measure *only* the time that is spend on actually solving the system of linear equations, and not on generating random matrices. You will need to find out how to measure time in Fortran. There may be built-in functions or subroutines that return current astronomical (or system) time. So what you can do is to call that function/subroutine just before and after your LAPACK calls.

Find a way to make sure that the solution you obtain is correct. For example, before your main loop, where you measure time, you can have a single test call of LAPACK for some small matrix size. Then you can substitute the solution into eq. (1) and estimate $||Ax - b||/||b||$, where $||...||$ is the norm of a vector. If the solution happens to be incorrect you can stop the program with a warning message.

When you are sure your program works fine and does not require any further debugging, compile it using the flag corresponding to the highest optimization level, `-O3`. That is, the command line to compile your code should look like this:

```
gfortran myprogram.f90 -O3 -o myprogram -llapack
```

Lastly, execute your program, take time measurements and plot them using the plotting software of your choice (Gnuplot recommended). Generate an image of this plot in the png or jpg format with high enough resolution (720×720 pixels or higher). Analyze the plot and determine how the computational cost scales with n. This should be a power law (roughly), i.e. time $\propto n^p$, with some possible fluctuations. What is the value of the power, p? If the dependence does not look like a power law or if the fluctuations are substantial, try to explain why it may take place.

Put the source file of your program (e.g. `myprogram.f90`), the text file containing time measurements (`time.txt`), plot (`time.png`), and your report (`report.txt`) with a brief discussion of your findings in subdirectory called `as1` in your google-drive directory that is shared with the instructor.

Found an error or need a clarification? Email the instructor at sergiy.bubin@nu.edu.kz