Numerical Methods: Direct and Iterative Solvers in Python

This repository consists of python implementation of Gaussian Elimination, Jacobi, Preconditioned Gradient Descent and Generalised Minimum Residual Solvers.

1. All the Solvers' python scripts are made as modular as possible with proper comments.
2. The matrix A and RHS vector b are generated using Random Number Generator functions separately for both respectively in each solver's python file inside the corresponding directory.
3. Solution time measuring snippet is also included in each solver's python file.
4. The results (data file) is generated inside the python code (as a separate function). Each linear system of equations (LSE) (10 x 10, 100 x 100, 1000 x 1000 etc) result file is separate for each solver. They are generated in the script and are placed under the *Results* directory with the following name convention:

`<Solver Name>_LSE_<row>_by_<col>`

e.g. for Gauss solver LSE 10 x 10:

`GAUSS_LSE_10_by_10`

This file consists of the following details:

1. Solution vector
2. Residual Norm Error
3. Run time
4. Iterations (if required)

NOTE:

For all the solvers, my computer could take only up to 1000 x 1000 LSE except PCG Solver which could compute for 10000 x 10000 LSE as well (Results can be checked in *Results* directory). I could observe for at most 2 hours for 10000 x 10000 LSE. But the solvers current implementation couldn't complete.

Steps for executing the scripts

Gaussian Elimination Solver

```
cd Gauss
python gauss_elim_prog.py <row> <col>
```

You can see the results data file in *Results* directory.

Jacobi Solver

```
cd Jacobi
python jacobi_prog.py <M> <N> <No. of iterations>
```

Preconditioned Conjugate Gradient (PCG) Solver

```
cd PCG
python PCG_prog.py <N> <No. of iterations>
```

Generalized Minimal Residual Method (GMRES) Solver

```
cd GMRes
python GMRES_prog.py <M> <N> <No. of iterations>
```