Abstract.

In this project, I implement an algorithm named Gaussian Elimination with Pivoting for solving Linear Systems. There are two parts of the program, one of which is implemented by Fortran codes whose purpose is to perform the core calculation to solve linear systems; the other part is implemented by Python codes which prepares the content of linear systems and drive the Fortran codes to produce solution, and then compare the results with those produced by some numpy method.

1. Gaussian Elimination explained

Consider a linear system as follows:

\[
\begin{bmatrix}
2 & 1 & 1 & 0 \\
4 & 3 & 3 & 1 \\
8 & 7 & 9 & 5 \\
6 & 7 & 9 & 8 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
\end{bmatrix}
= 
\begin{bmatrix}
3 \\
6 \\
10 \\
1 \\
\end{bmatrix}
= b.
\]

The basic idea of Gaussian elimination is to transfer the Matrix A above to have a form of upper triangular as below:

\[
\begin{bmatrix}
2 & 1 & 1 & 0 \\
1 & 1 & 1 & \\
1 & 1 & 2 & 2 \\
1 & 1 & 2 & 2 \\
\end{bmatrix}
\]

The way of solving this transformed system has now become as easy as doing a backward substitution.

2. Programming logic explained

Directory structure of source codes, as in the directory ./project/: 

.. LinAlg/linear_solve.F90
   —— read_inputs_module.F90
   ——— guassian_elimination_module.F90
   —— output_module.F90

1
The Python scripts are located at PyRun directory. Program pyRun.py would produce necessary \( A_i.dat \) and \( b_i.dat \) files to directory LinAlg/inputs, and then drive Fortran codes in directory LinAlg to solve it. The resulting set of solution \( x \) would be printed to screen as well as saved to files \( x_i.dat \) in directory LinAlg/outputs.

Then the python program would solve same linear systems using numpy module. The result is considered as in good precision and be further used for checking the correctness of Fortran results. Then, numpy results are printed out and the results of the comparison would be shown on screen as Pass/Fail indicating if Fortran has correctly solved for each linear systems.

3. Results reproduction

To test out my codes, please make sure your are now working in directory PyRun. Enter command below in Unix console:

```
python pyRun.py
```

Subsequently, the Fortran codes would be automatically compiled, linked and executed. The computation results from Fortran codes would be shown on screen and also be saved to file \( x_i.dat \) where \( i=1,2,3 \). Then, an alternate solutions provided by numpy module would be shown on screen for comparison. Furthermore, the comparison results would be shown on screen by either Pass or Fail, indicating the result from Fortran codes is correct or incorrect, respectively.

To restore what you have done so far, please make sure your are now working in directory PyRun. Following command would delete every intermediate files and every .dat files during previous execution. In Unix console:

```
python pyRun.py clean
```

4. Results

The solutions from Fortran codes:
The subsequent python solutions and comparisons:

```
# solutions from numpy
x_1= [ 2.  2.  3.]
x_2= [  0.000000000000  1.000000000000  -1.000000000000 -2.05485921e-17]
x_3= [ 1.  1. -3. -3.]

# Compare results
The error tolerance is now set to: 1e-06
Check correctness for system 1: Pass
Check correctness for system 2: Pass
Check correctness for system 3: Pass
```

The visualization for linear system 1:
The visualization for linear system 2:

The visualization for linear system 3:
5. Conclusion

The results from fortran codes is extremely close to the numpy module's result that we assumed is more accurate and used for correctness checking. In fact, the results from fortran is only 1e-16 differ from numpy results. Therefore it can easily pass the correctness check using a 1e-06 error tolerance. This project helps me to get hands-on experience to implement a scientific algorithm, though not very sophisticated, from scratch.

6. Acknoledgement

I want to thank Professor Dongwook Lee for his help to my project and his dedication to present the AMS209 course. I also appreciate the Teaching Assistant Skylar Trigueiro for his scrutinious work grading our homeworks.