Program #4

Joey Troyer

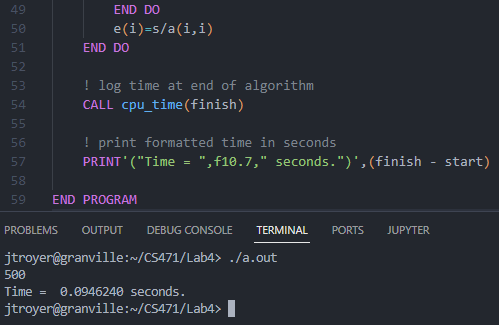
09/26/22

Problem description:

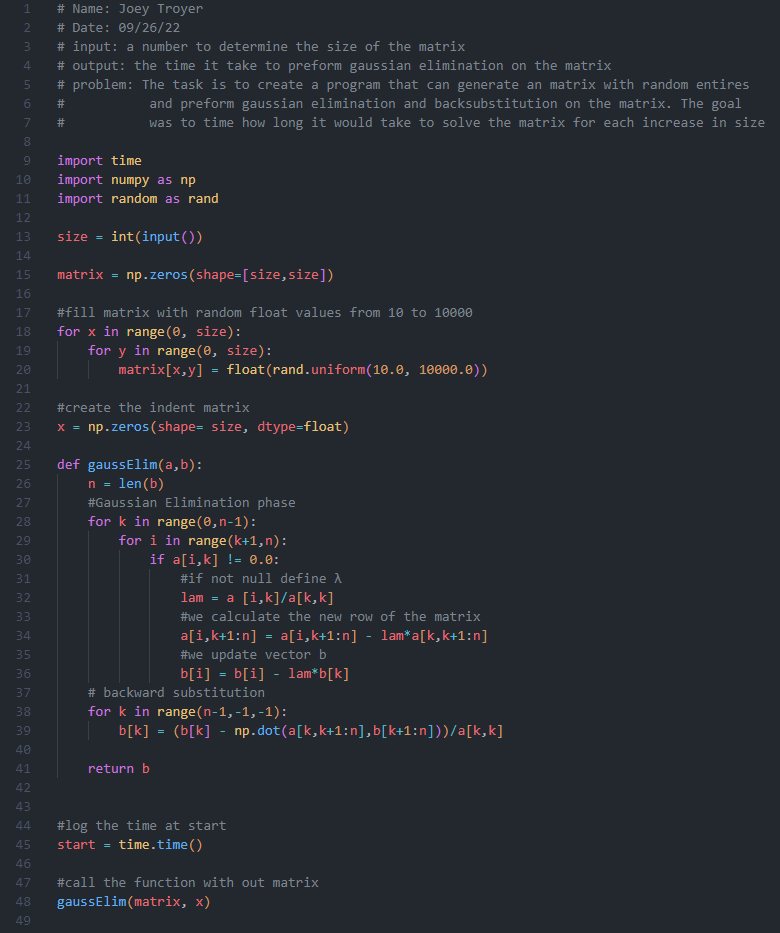
The task of this programming assignment was to test and compare the speed of interpreted and compiled languages. I was tasked with creating three different programs that would perform gaussian elimination with back substitution. The programs to develop were in python, python with NumPy, and Fortran.  At the start of the program, you should be able to enter a number that will determine the size of the matrix and fill that matrix with randomly generated values. We are then to time how long each program takes to perform gaussian elimination on each matrix for every size.

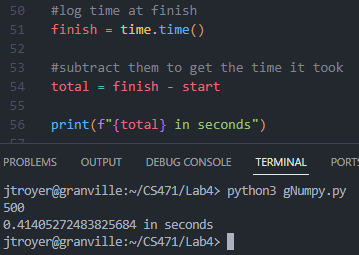
FORTRAN:





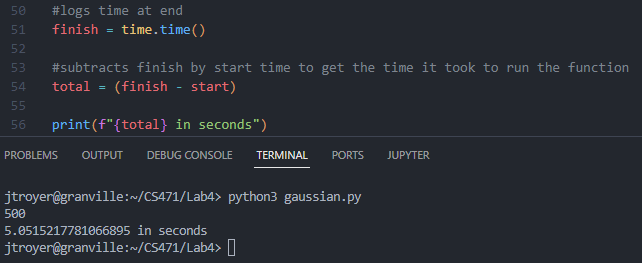
PYTHON WITH NUMPY:





PYTHON:



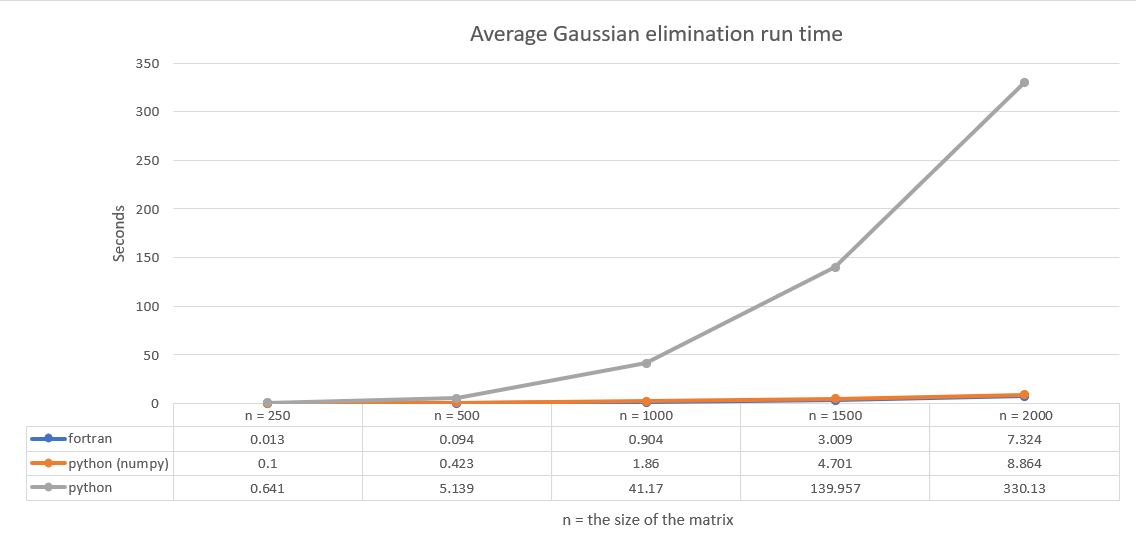


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FORTRAN  In seconds | 250 | 500 | 1000 | 1500 | 2000 |
| TEST 1 | 0.013 | 0.095 | 0.905 | 3.071 | 7.296 |
| TEST 2 | 0.013 | 0.093 | 0.903 | 3.045 | 7.263 |
| TEST 3 | 0.013 | 0.096 | 0.906 | 2.839 | 7.254 |
| TEST 4 | 0.013 | 0.093 | 0.910 | 3.057 | 7.259 |
| TEST 5 | 0.013 | 0.094 | 0.897 | 3.032 | 7.549 |
| AVERAGE | 0.013 | 0.094 | 0.904 | 3.009 | 7.324 |
| STD | 0.000 | 0.001 | 0.004 | 0.096 | 0.126 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Python  (Numpy)  in seconds | 250 | 500 | 1000 | 1500 | 2000 |
| TEST 1 | 0.100 | 0.423 | 1.870 | 4.686 | 9.017 |
| TEST 2 | 0.099 | 0.421 | 1.858 | 4.712 | 8.850 |
| TEST 3 | 0.100 | 0.425 | 1.846 | 4.697 | 8.819 |
| TEST 4 | 0.099 | 0.425 | 1.852 | 4.707 | 8.808 |
| TEST 5 | 0.103 | 0.422 | 1.873 | 4.704 | 8.827 |
| AVERAGE | 0.1 | 0.423 | 1.86 | 4.701 | 8.864 |
| STD | 0.001 | 0.001 | 0.011 | 0.010 | 0.086 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Python in seconds | 250 | 500 | 1000 | 1500 | 2000 |
| TEST 1 | 0.668 | 5.136 | 40.957 | 138.679 | 327.127 |
| TEST 2 | 0.634 | 5.124 | 41.016 | 138.855 | 328.394 |
| TEST 3 | 0.635 | 5.135 | 41.036 | 139.840 | 329.918 |
| TEST 4 | 0.635 | 5.134 | 41.333 | 140.576 | 332.361 |
| TEST 5 | 0.635 | 5.166 | 41.509 | 141.837 | 332.850 |
| AVERAGE | 0.641 | 5.139 | 41.17 | 139.957 | 330.13 |
| STD | 0.014 | 0.015 | 0.239 | 1.301 | 2.472 |

Graph shows the average time of each program



Python without NumPy performed the slowest by far. It was relatively close in speed compared to python, with NumPy and Fortran up to 500. After 500, it took a wildly longer time than the other two. Rendering it as almost entirely impractical for this use case. Python with NumPy had impressively fast speed, cutting its time down substantially. Fortran performed the best of the three, with each of its times fastest. In conclusion these results are to be expected because python is an interpreted language known to be slower. However, I was pleasantly surprises by how much NumPy improved the runtime. Of course, the language with the fastest time is Fortran because it is a compiled language.

Sources:

<https://labmathdu.wordpress.com/gaussian-elimination-without-pivoting/>

<https://youtu.be/r89yH82OAFw>

<https://techgoggler.com/computer-engineering/linear-equations-python-gauss-elimination-method/>