Gaussian Elimination Runtime Analysis Report

# Time

## Setup

* System: 4x4 linear system
* Small initial pivot (A[1][1]) for computational imprecision
* Iterations: 1000 runs per configuration
* Configurations tested: 5 variants, including baseline (because the first loop consistently showed anomalously large times)

### Results

#### Performance Summary

| **Configuration** | **Mean Time (s)** | **Performance vs Baseline** |
| --- | --- | --- |
| Dummy | 0.000106 | Baseline |
| NGE\_Float | 0.000092 | 1.14× faster |
| SPP\_Float | 0.000100 | 1.05× faster |
| NGE\_Double | 0.000084 | 1.25× faster |
| SPP\_Double | 0.000088 | 1.20× faster |

### Findings

* NGE\_Double is the clear winner, about 25% faster than the baseline. This is unsurprising because though NGE and SPP have the same Big-O time, SPP has a larger coefficient.
* NGE operations are generally faster than their SPP counterparts. Similarly, Double operations are faster than their Float counterparts, which I find odd since there are more bits to divide—an expensive operation.

# Precision

### Results

#### Solutions Summary

| **Method** | **x1** | **x2** | **x3** | **x4** |
| --- | --- | --- | --- | --- |
| NGE\_Float | 0.211136 | -0.002448 | 0.639386 | 0.746613 |
| SPP\_Float | 0.216025 | -0.007915 | 0.635243 | 0.746174 |
| NGE\_Double | 0.216024767007660 | -0.007915106091192 | 0.635243326484800 | 0.746174276089684 |
| SPP\_Double | 0.216024767008412 | -0.007915106087778 | 0.635243326493106 | 0.746174276085716 |

### Findings

* SPP methods are more consistent and numerically stable than their NGE counterparts. This can be seen in the minimal difference between SPP\_Float and SPP\_Double as opposed to the difference in NGE\_Float and NGE\_Double