COMP-605 Homework 0

Dr. J Corbino, SDSU

Geneva Porter, February 24 2020

For this assignment, a native Ubuntu 64-bit environment was used on a machine with 16 GB memory and 2.7 GHz CPU. I used C++ as my coding language. The solutions are written to output-solutions.txt upon compiling and running the code, Porter-hw00.cpp.

For measuring run times for each scheme, I ran each algorithm 1000 times and summed the resulting time measurements. From this I was able to take the average and provide a reasonably accurate estimate of the optimization effects. One odd occurrence is that the matrix multiplication time would occasionally output a negative value; in this case I ignored that value and ran the loop an additional time. This may be due to a clock resolution issue on my machine.

	ijk-form	jki-form
No optimization (-O0 flag)	0.0023799	0.00275436
With optimization (-O3 flag)	0.00051495	0.000476634

Table 1: Run times for 50×50 matrix

	ijk-form	jki-form
No optimization (-O0 flag)	0.0209508	0.0225209
With optimization (-O3 flag)	0.00483495	0.0042901

Table 2: Run times for 100×100 matrix

	ijk-form	jki-form
No optimization (-O0 flag)	0.12632	0.139251
With optimization (-O3 flag)	0.0285915	0.0363679

Table 3: Run times for 200×300 matrix

Since I used C++, the ijk row algorithm is predictably more efficient for the non-optimized run. This is because C++ accesses arrays through storing values in rows, which is more intuitive (for myself, as an applied math student). However, after testing several times, the optimized run shows that the jki column algorithm is more time efficient for the 50×50 and 100×100 matrices, but less efficient for the 200×300 matrix. I'm not sure why this switch happened.

Memory leaks were checked, and although valgrind threw some errors regarding initialization, all heap blocks were freed and no leaks were possible. While the instructions

stipulated that 40,000 32-bit real values should be stored, since I allocated a 2D array, I limited the matrix dimensions to 2000, which allowed for 4,000,000 entries. Hopefully this is sufficient for the assignment requirements and not wasteful.