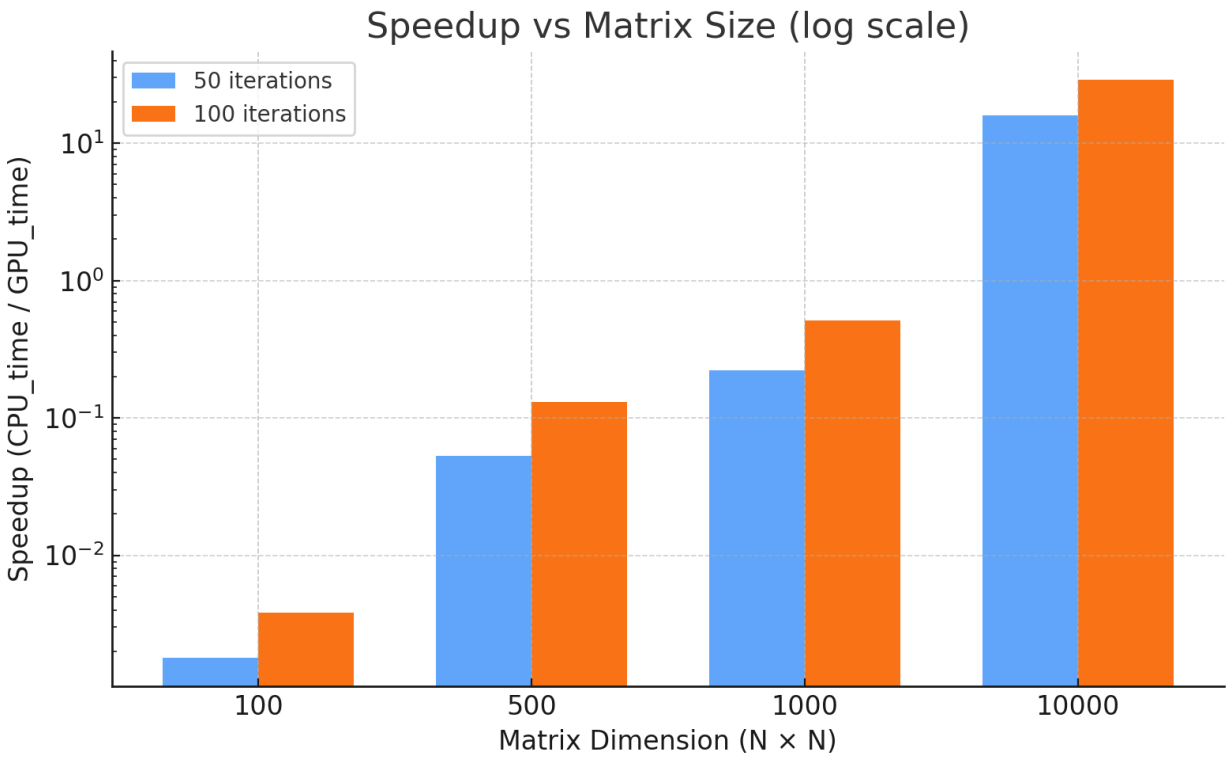

5. a. GPU more beneficial at N=10000, because the problem size is large enough to fully utilize GPU parallelism and amortize memory transfer + launch overhead.
5. b. Speedup lowest at small N (N=100, 500, 1000) because the fixed GPU overhead (kernel launch & memory transfers) dominates, and the CPU can compute small grids faster.
5. c. Speedup highest at large N (N=10000) because more computations per iteration allow the GPU to run thousands of threads in parallel, making overhead negligible compared to compute time.

6. a. GPU more beneficial at N =10000 (and grows with more iterations).
6. b. Speedup lowest at small N (N=100500), same reason: overhead dominates.
6. c. Speedup highest at large N (N=10000); doubling iterations improved the top speedup from ~16× -> ~29× in my data.

7. Increasing the number of iterations increases GPU speedup (all else equal).