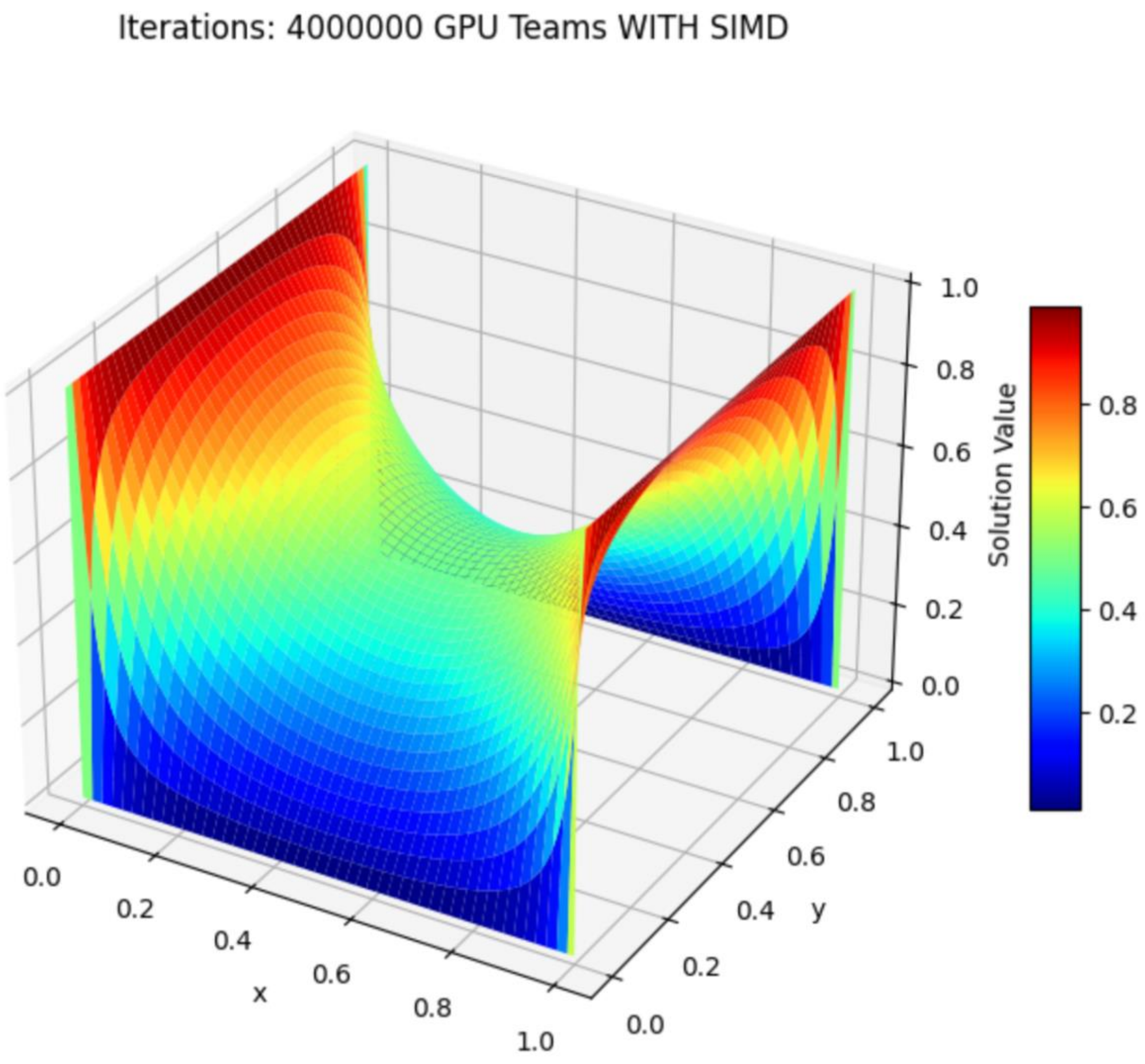
3D Surface Plots

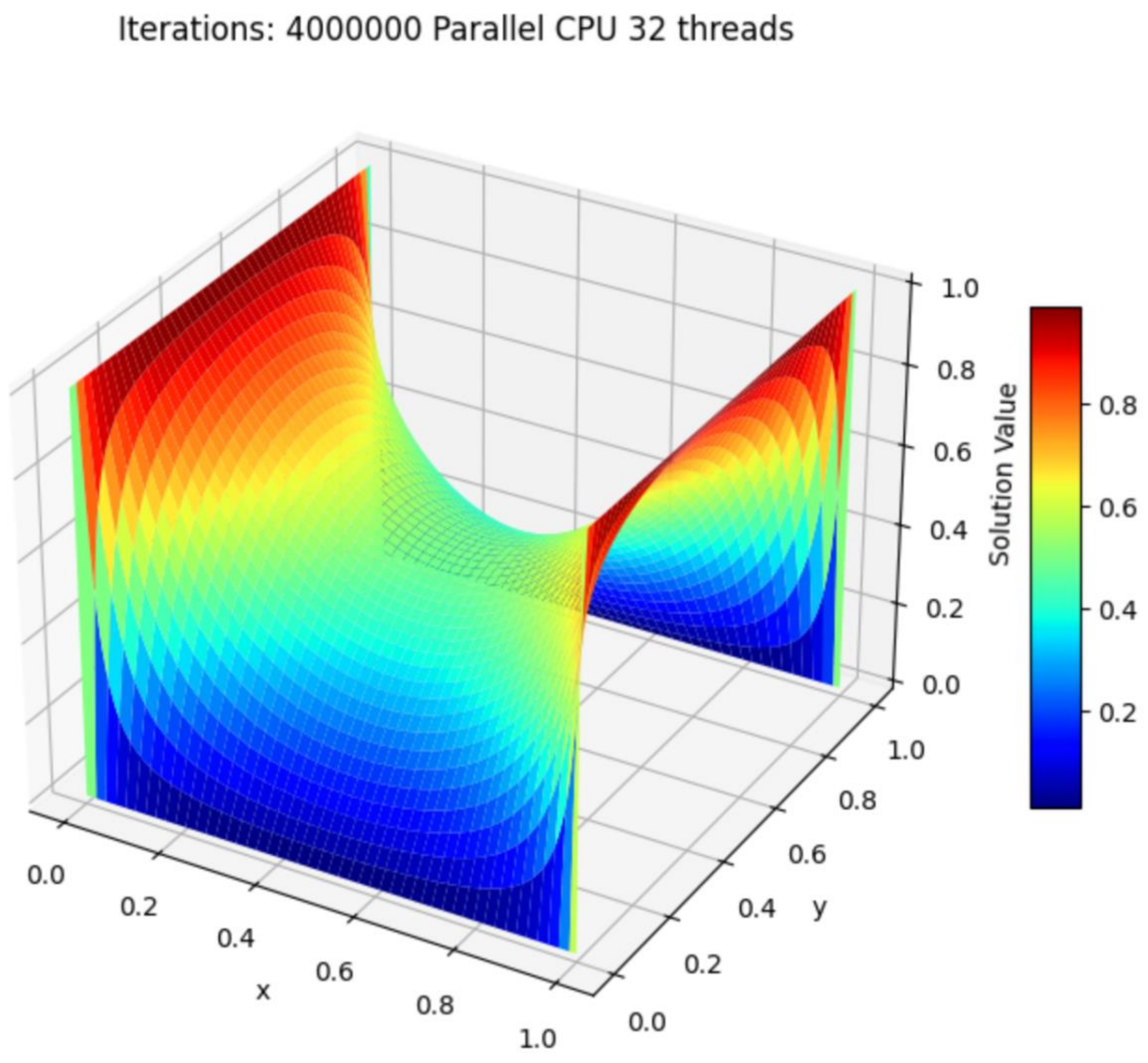
Part 1 – Jacobi 2D Serial Execution 400000 iterations

Final Residual: 2.06701e-09



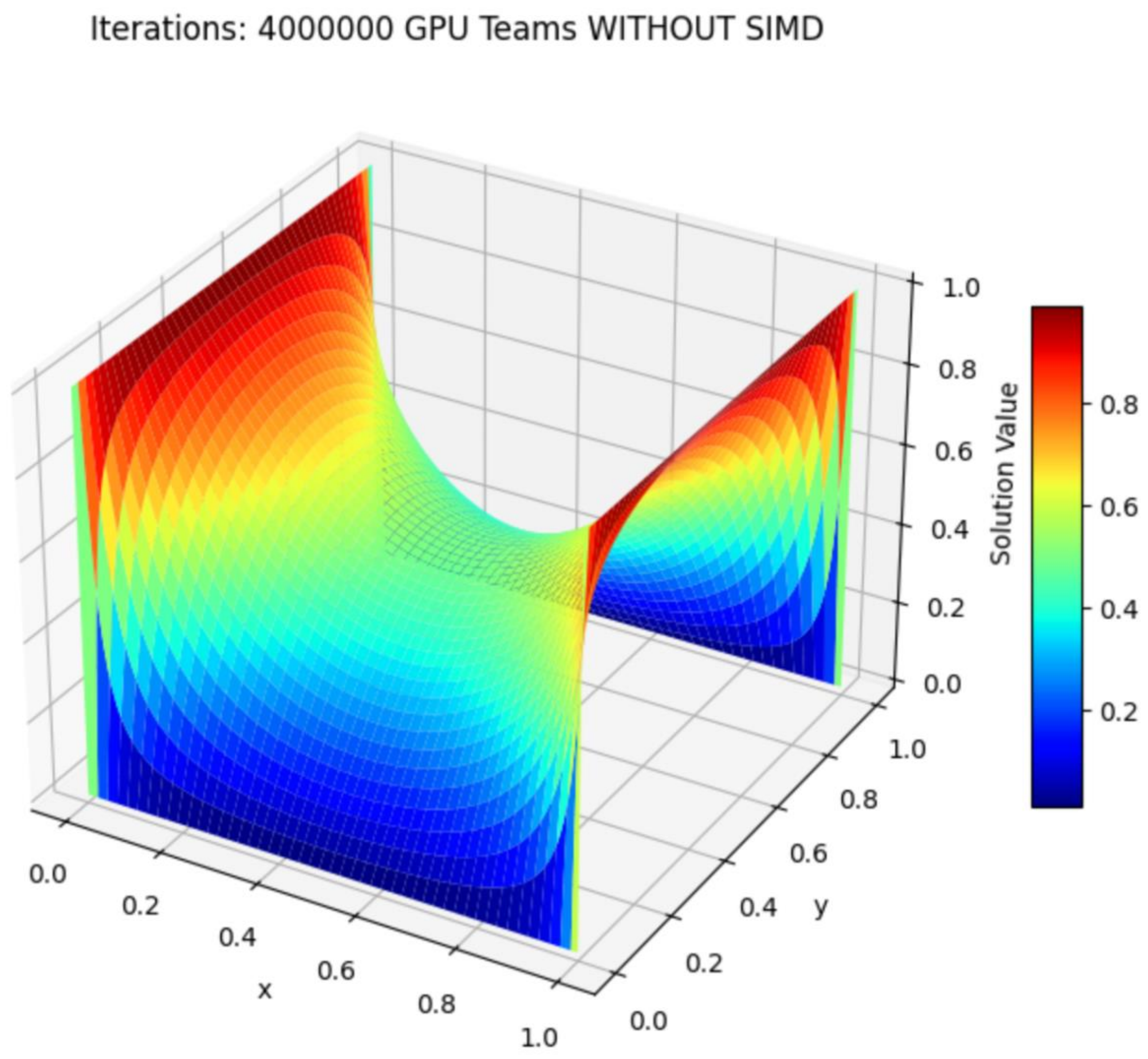
Part 2 - OpenMP solution using CPU hardware threads.

Final Residual: 1.767781e-09

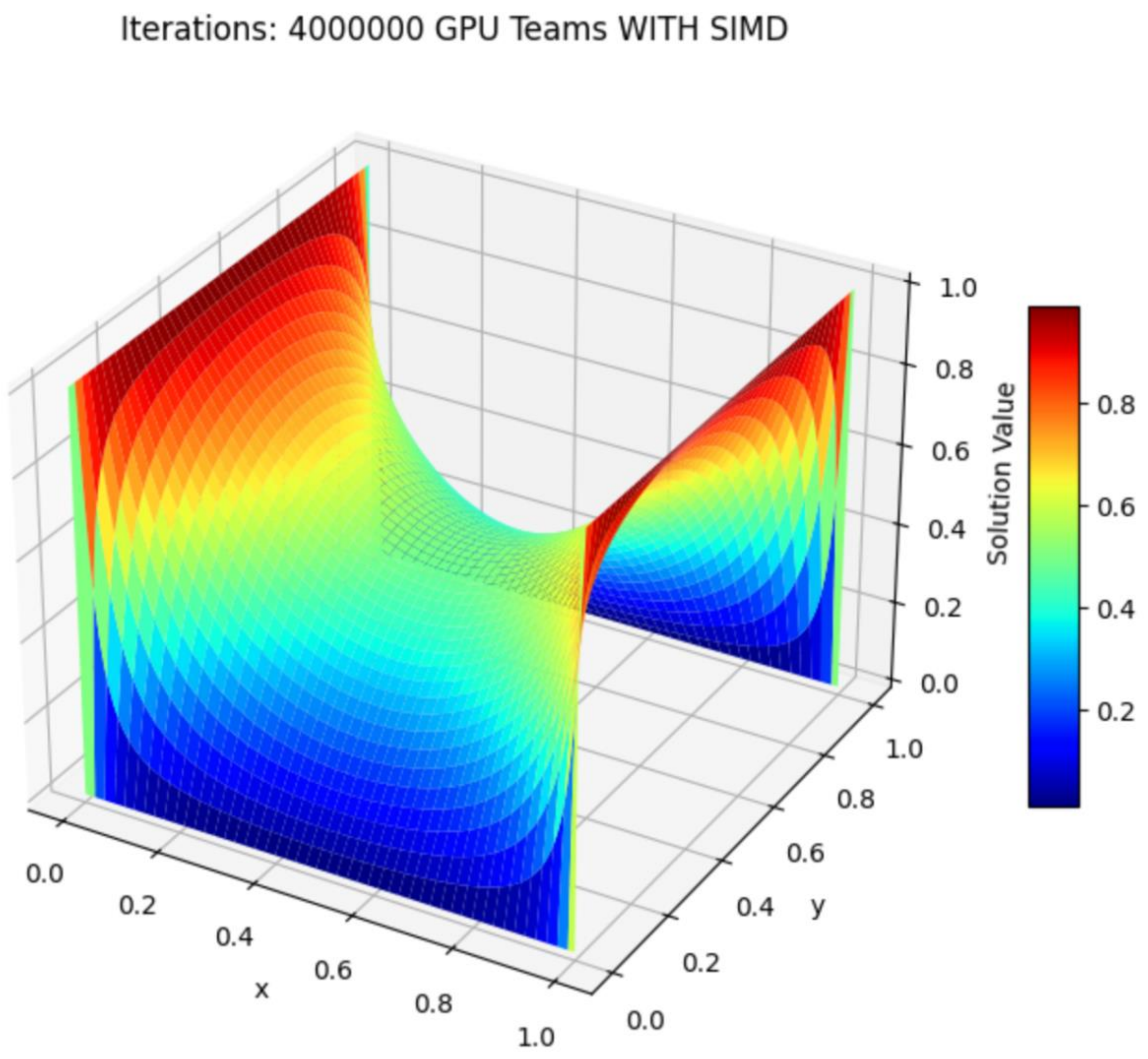


Part 3 – GPU With Teams without SIMD

Final Residual: 6.816744e-10



Part 4 - GPU With Teams with SIMD



Brief Discussion

Through my experiments with OpenMP GPU offloading, I compared four different implementations of the Jacobi method: serial execution, CPU parallel execution with 16,32,64,128 threads, GPU offloading using teams, and GPU offloading with teams and SIMD. The serial version was the slowest due to its lack of parallelization, while the CPU parallel version showed significant speedup, with 32 threads providing the optimal performance. However, GPU offloading further improved efficiency, as the GPU with teams outperformed the CPU parallel approach, highlighting the benefits of massive parallelism. The best performance was achieved using GPU offloading with teams and SIMD, as vectorization further optimized computation within each thread. The key takeaway is that while CPU parallelization is effective, leveraging OpenMP offloading with proper memory management and vectorization on the GPU yields the highest speedup for computationally intensive iterative solvers like Jacobi.