FDTD Acceleration using Matlab and Parallel Computing Toolbox on GPU Cards

Joseph E. Diener*, Atef Z. Elsherbeni
Electrical Engineering and Computer Science Department, Colorado School of Mines

Recent developments in GPU computing have shown that finite difference time domain (FDTD) based solvers can achieve much higher speeds when implemented on GPU's as opposed to traditional CPU based computing. Optimal performance for FDTD-GPU based computing requires some specialized programming ability in order to achieve the impressive speedups that have been reported in the literature. The Matlab Parallel Computing Toolbox (PCT) allows users to store and manipulate arrays directly on a GPU, without any specialized programming ability beyond general Matlab programming. We present a Matlab based FDTD code using development the PCT to solve electromagnetic problems on GPUs, and demonstrate significant speedups when compared with the same Matlab code executed on the CPU. Only relatively small modifications to the original Matlab code are required to achieve this improved performance. As there is the possibility that different Matlab releases will have different performance for similar code as a result of compiler changes, modifications are presented iteratively and profiled on different NVIDIA GPUs. The code presented is kept extremely general, and so is applicable to very general FDTD setup. Performance is benchmarked for a near field source - a dielectric sphere excited by an antenna nearby - as well as a TF/SF problem where the sphere is excited by an incident plane wave. The solver speed as measured in millions of cells per second (MCPS) is reported over increasingly finely discretized CPML-terminated FDTD grid. Results show that there are optimal memory loads for peak performance when executed using the Matlab PCT, after which performance begins to drop off.