

Problem

Wireless security is a challenge involving software and hardware development and integration. This project focuses on developing a GPU accelerated finite-difference time-domain (FDTD) solver capable of injecting spatio-temporally-variant waveforms (STVW) including Gaussian, Hermite-Gaussian, and Laguerre-Gaussian modes. The developed code enables fast analysis of radiation from phased array antennas and radar cross section of different complex targets under illumination using STVW.

Key Ideas

- Finite-difference time-domain (FDTD) method
- GPU acceleration using CUDA and C++
- Spatio-temporally-variant waveform injection using scattered field formulation

Relevance

The FDTD solver provides fast and accurate electromagnetic solutions for the complex geometries of military platforms. This work aims to identify regimes where the radar cross section under illumination by STVW outperforms the radar cross section from a plane wave excitation. This supports the development of next-generation radar systems and target detection technologies.

Technical Approach/Methodology

- The FDTD solver implements central difference approximations to solve Maxwell's Equations in the time-domain.
- The electric and magnetic fields are computed at every point in a Yee grid each time step.
- The total field is decomposed into an incident and scattered field.
- The incident field injects a plane wave or spatio-temporally-variant waveform using the analytical expression of the wave.
- CPML boundary conditions to absorb outgoing radiation
- Results are obtained from the scattered field by near-field sampling or far-field transformation.

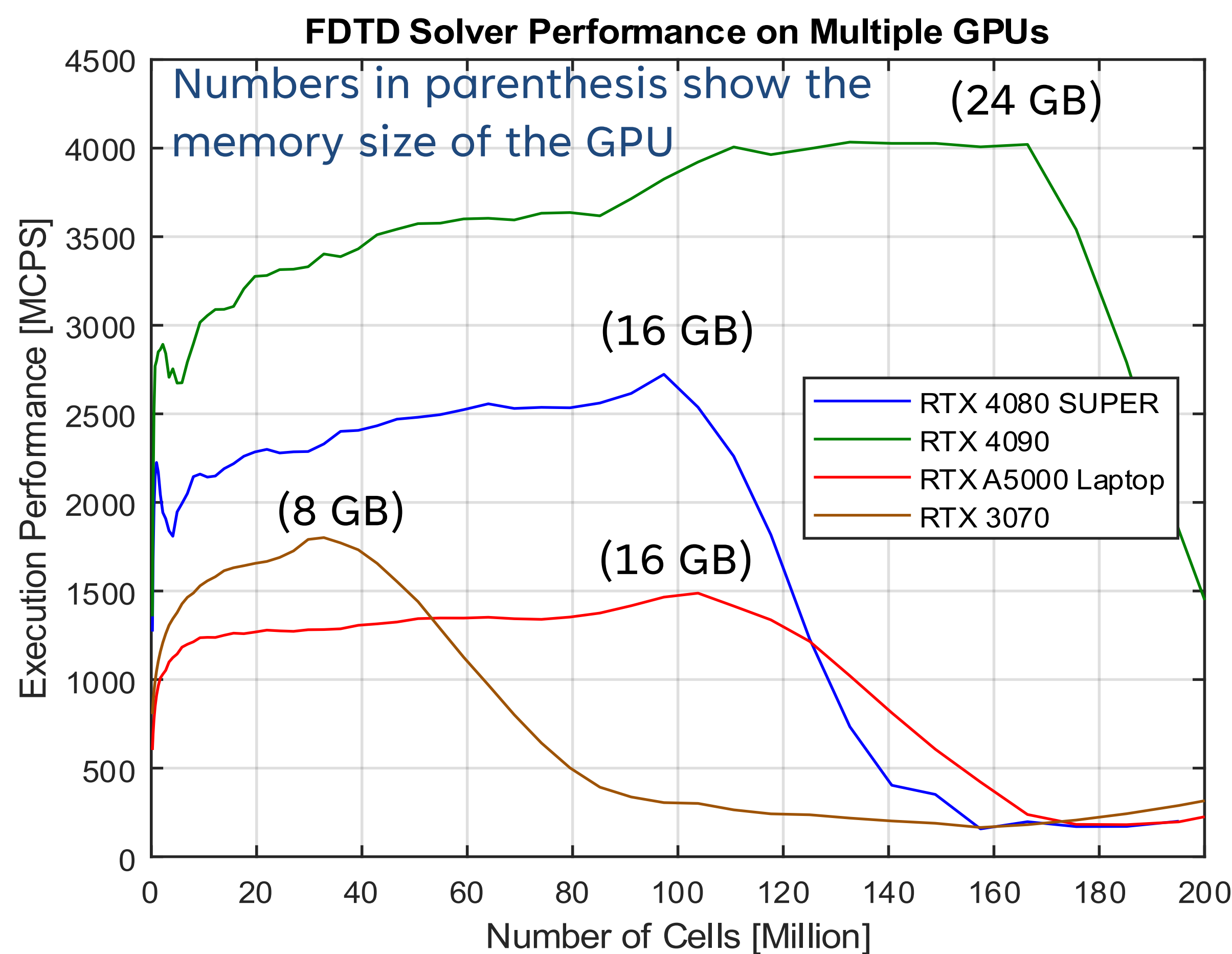


Figure 1: Performance test of FDTD solver measuring throughput in millions of cells per second (MCPS) for various domain sizes measured in millions of cells.

Results

The solvers speed was measured by simulating domains of increasing size on multiple GPUs. The benchmark simulation consisted of:

- Incident Gaussian beam
- Dielectric sphere positioned at the origin with $\epsilon_r = 4$ and varying radius to control domain size
- Cells of 2.5 mm length in all dimensions

For each simulation, the performance was measured by calculating the millions of cells updated per second (MCPS).

$$MCPS = \frac{(\# \text{ of Cells}) (\# \text{ of Time Steps})}{10^6 (\text{Simulation Time})}$$

The results of the performance evaluation is shown in Figure 1.

Next Steps

- Increase solver efficiency by optimizing memory access patterns.
- Develop user interface to quickly configure simulations and process results.
- Implementation and benchmarking of Multi-GPU simulations.
- Extend accuracy from single to double precision floating-points for the stability of long simulations.

Team

Colorado School of Mines

Student: Andrew Smith

Advisor: Dr. Atef Elsherbeni