Solving the Wave Equation on GPUs in the Context of Lightning Physics

Contact: Andy.Martinez@cwi.nl

1 Topic Details

For this topic we are looking for a student who can adapt our current CPU based wave-equation solvers to GPU code. The wave equations being solved have the form:

$$\Delta\phi - \frac{1}{c^2}\frac{\partial^2\phi}{\partial t^2} = f \tag{1.0.1}$$

Discretizing this wave equation gives us an explicit formula which can be solved on GPUs efficiently. The question to be answered is when (and if) the GPU solver outperforms the CPU solver.

2 Scientific Context

This project is in the context of computational lightning physics. In the Multiscale Dynamics group we perform simulations of so called "streamers". These are small self propagating plasma channels which precede the big lightning flashes seen in thunderstorms.

Understanding how streamers work can improve our understanding of topics like protecting windmills from lightning strikes, extinguishing large arcs in power production facilities, and of course lightning itself.

These streamers can be simulated as a fluid, for this we would solve the drift-diffusion equation for the particle species inside the plasma channel:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\mathbf{J}(\mathbf{E}) = \text{sources} - \text{drains} \tag{2.0.1}$$

Where ρ is the density of the particle species, $\mathbf{J}$ is the current-density of that species, and $\mathbf{E}$ is the electric field in the simulation. Calculating the electric field is where this project is going to be about.

If we want to calculate the electric field we need to solve the Maxwell equations which can be formulated as a set of two wave equations:

$$\Delta\phi - \frac{1}{c^2}\frac{\partial^2\phi}{\partial t^2} = -\frac{\rho}{\epsilon_0} \tag{2.0.2}$$

$$\Delta\mathbf{A} - \frac{1}{c^2}\frac{\partial^2\mathbf{A}}{\partial t^2} = -\frac{\mathbf{J}}{\epsilon_0 c^2} \tag{2.0.3}$$

This project will consist of several parts:

1. Solve the wave equation in 1D on a CPU.
2. Literature study on solving the wave equation on a GPU.
3. Learning to work with CUDA.
4. Developing a 1D wave equation solver on a GPU.
5. Comparing the efficiency of the GPU solve with the CPU solver.
6. Extending to 2D if time permits.