Finite-Difference Time-Domain Method

Andrew Hayman[1, *]

[1] *Department of Physics, Engineering Physics and Astronomy, Queen's University, Kingston, ON K7L 3N6, Canada*

(Dated: April 5, 2022)

Here, we investigate the famous finite-difference time-domain (FDTD) method for solving Maxwell's equations. First, we consider a 1D vacuum and demonstrate simple absorbing boundary conditions that result in only 0.015% reflection of the injected field, as well as the total-field scattered-field approach that results in only 0.0010% backwards injection. We then analyze the behaviour of a dielectric slab, showing that the transmission and reflection closely matches the analytical solution. Then, we implement frequency-dependent permittivities with the Drude and Lorentz dispersion models through the usage of Z-transforms. Again, the numerical behaviour closely matches the analytical solutions. Finally, we extend the basic dielectric slab case to 2D for a dielectric box. We compare different methods and identify Numba's row-major method as the dominant strategy with a five-fold improvement over the next best method of Numba's vectorization. Finally, we parallelize the code with MPI and demonstrate near linear scaling with the number of processes as expected since the FDTD method is embarrassingly parallelizable.

I. RESULTS

A. Simple Absorbing Boundary Conditions and Total-Field Scattered-Field

First, we implemented the simple absorbing boundary condition (ABC) the total-field scattered-field (TFSF) approach with snapshots shown in Fig. 1, showing the pulse getting injected forwards due to the TFSF approach and then out of the slab due to the simple ABC. To quantify the backwards injection from the TFSF approach and reflection from the simple ABC, we tracked and plotted the field just behind the source point shown in Fig. 2. Note that the injected field has a maximum amplitude of 0.65 as shown in Fig. 3. The field due to backwards injection is 6.6×10^{-6}, or 0.0010%. The reflected field from the ABC is 0.00010 or 0.015% of the injected field. This indicates that the simple ABC and TFSF approach are working as expected with negligible reflections and backwards injections.

B. Dielectric Slab

A thin film of $1\mu m$ with $\epsilon = 9$ was added $3\mu m$ into the slab with snapshots shown in Fig. 4. A running fast Fourier transform (FFT) was implemented for the incident, transmitted, and reflected electric fields which allowed us to compute the transmission and reflection coefficients for a select set of frequencies. We show the fields over time, their FFTs, and the resulting reflection and transmission compared against the analytical solution in Fig. 5. As expected, the analytical and numerical methods match closely for the reflection and transmission.

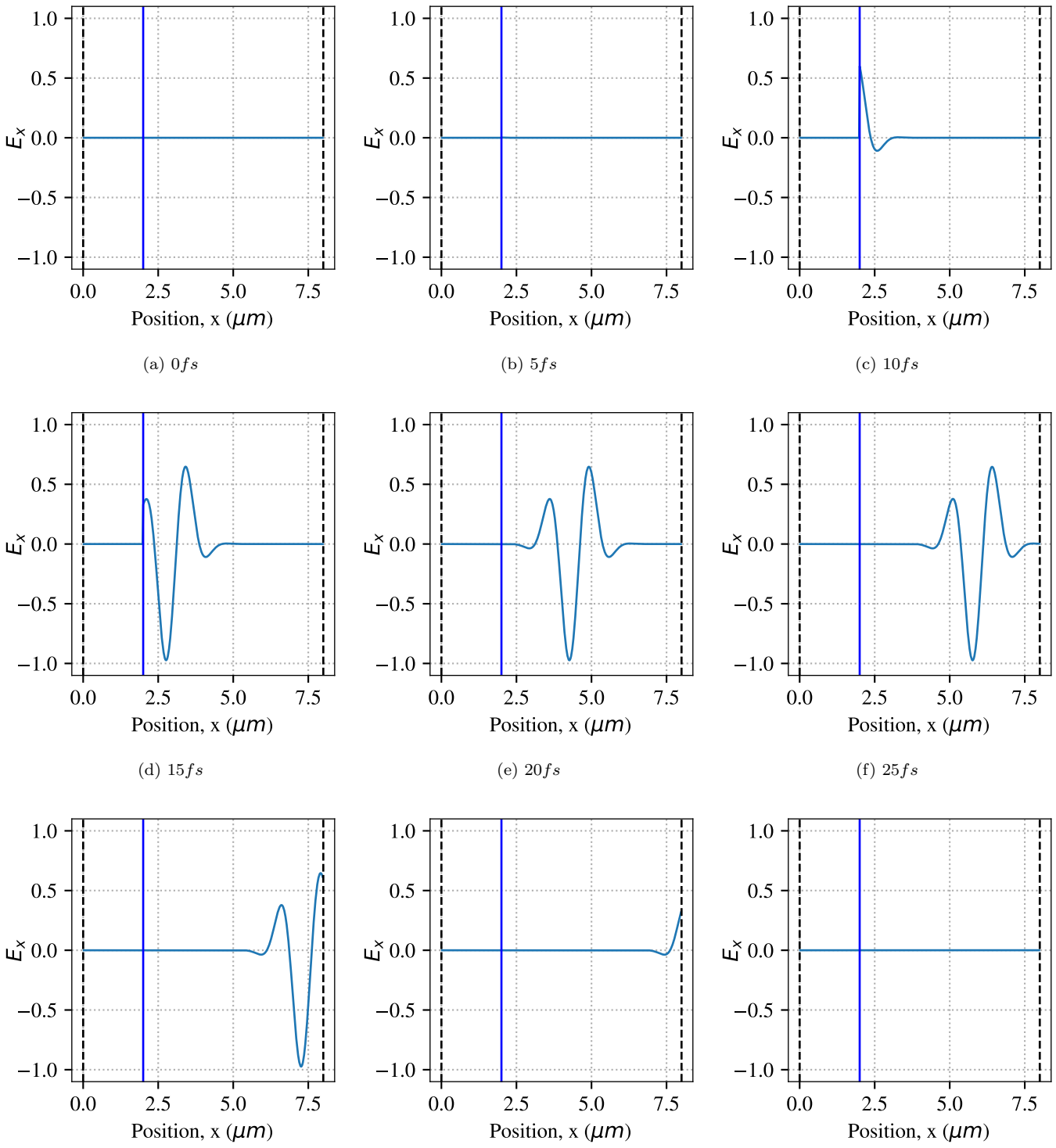
C. Drude and Lorentz Dispersion Model Slabs

Then, we implemented a frequency-dependent permittivity, $\epsilon(\omega)$ with the Drude dispersion model and the Lorentz dispersion model. This was accomplished through the usage of Z Transforms of the original frequency functions. For both cases, we examined slabs of $200nm$ and $800nm$ with $1fs$ Gaussian pulses. For the Drude model, we use parameters $\alpha = 140 \times 10^{12} rad/s$ and $\omega_p = 1.26 \times 10^{15} rad/s$ with results shown in Fig. 6. For the Lorentz model, we use parameters $\alpha = 4\pi \times 10^{12} rad/s$, $\omega_0 = 2\pi \times 200 \times 10^{12} rad/s$, and $f_0 = 0.05$ with results shown in Fig. 7. Both numerical solutions closely match their corresponding analytical solutions as expected.

D. 2D FDTD

The basic 1D FDTD was extended to 2D where grids of points are now considered. We simulate the same 2fs Gaussian envelope pulse as before with a dielectric box of $\epsilon = 9$ to the top right of the pulse. A set of sample plots are shown in Fig. 8. We examine different speed-up techniques using Numba, and timing results for different methods can be seen in Tab. I. The fastest technique is a Numba row-major looping method. Using row-major form results in around a 6x speedup for Numba due to cache usage and spatial locality of reference. The looping methods without Numba are nearly equal and drastically slower than all other methods due to a lack of optimizations for NumPy looping. We can also observe that vectorizing the code gives almost indistinguishable results, indicating that Numba may not be good for slicing optimization. For the fastest method, we remove the permittivity matrix and instead scale the fields in the dielectric box by the permittivity which is more memory and computationally efficient. We also use Numba's cache feature.

* 18arth@queensu.ca



(a) $0fs$ (b) $5fs$ (c) $10fs$

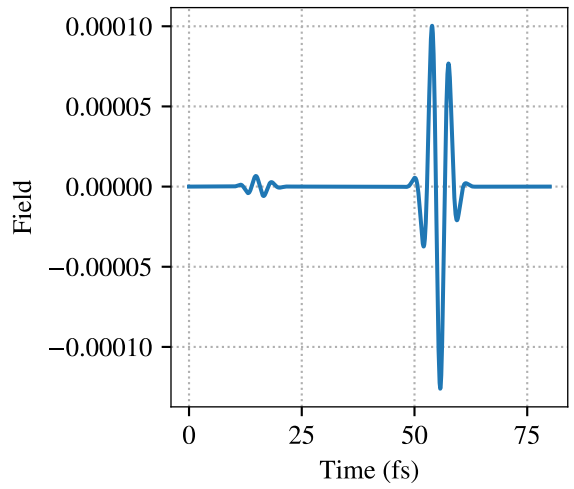
(d) $15fs$ (e) $20fs$ (f) $25fs$

(g) $30fs$ (h) $35fs$ (i) $40fs$

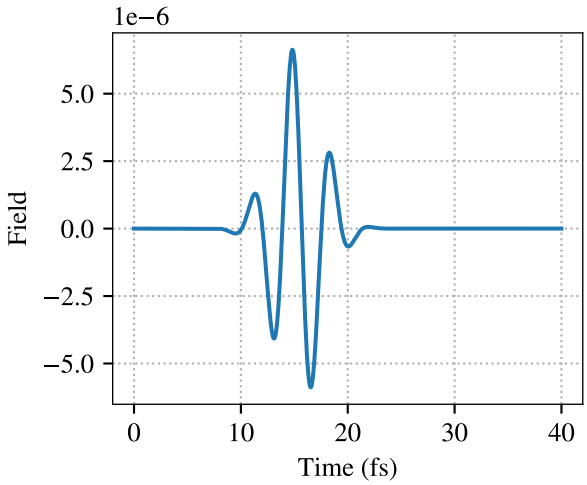
Figure 1. Snapshots of pulse over $40fs$, showing the implementation of simple absorbing boundary conditions, and the total-field scattered-field approach. As expected, the pulse only propagates to the right, and gets absorbed at the end.

E. 2D FDTD with MPI

We then parallelized the code for MPI by dividing the workload into parallel processes along the x-axis. A future test could be splitting over the y-axis, but it should make little difference for the row-major looping technique. The only two elements that each parallel process require from adjacent processes are the first Ez column from the right process and the final Hy column from the left process. Timing results can be seen in Tab. II and Fig. 9 with sample plots shown in Fig. 10 for a 3008x3008 grid with 1000



(a) (0-80)fs



(b) (0-40)fs

Figure 2. Diagram of the field over time before the source point at $1\mu m$. The first ripple at around 15fs is due to backwards injection of the TFSF approach and is 6.6×10^{-6}. The second ripple at around 55fs is due to backwards reflection from the simple ABC and is 0.00010.

time steps. The time scaling is nearly linear, but cannot be confirmed due to substantial variations in times for the 8 and 16 parallel process cases.

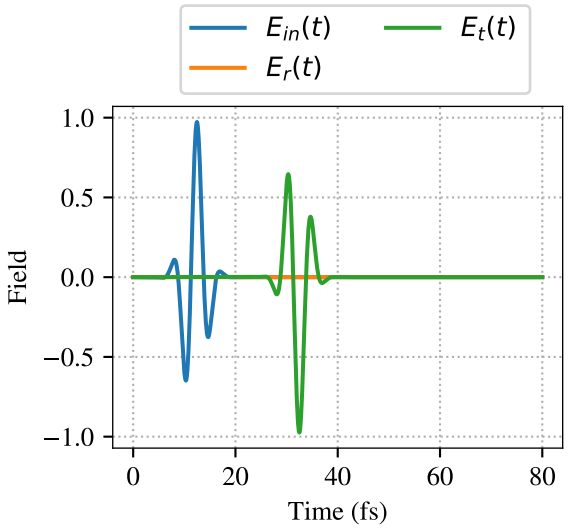
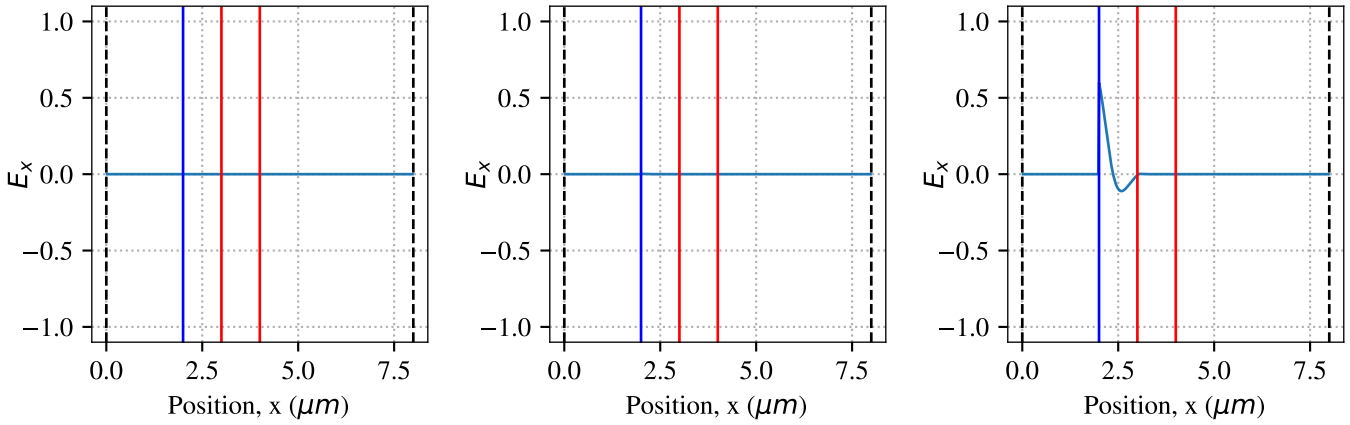


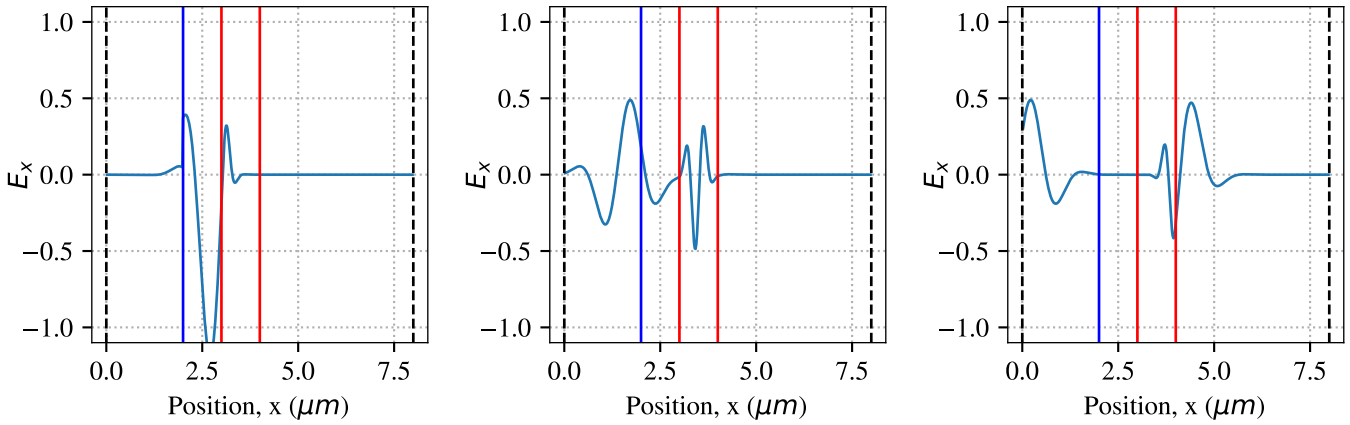
Figure 3. Reflection, transmission and pulse fields over time, showing a maximum injected field amplitude of 0.65. As expected, there is negligible reflection.

Table I. Timing results for a grid size of 500x500 and 1000x1000 over 1000 times steps averaged over three trials. With the exception of the "Fastest" method, all methods use the same format for consistency. The non-Numba loop methods are run for 10 times steps and multiplied by 100 to reduce the time to get results.

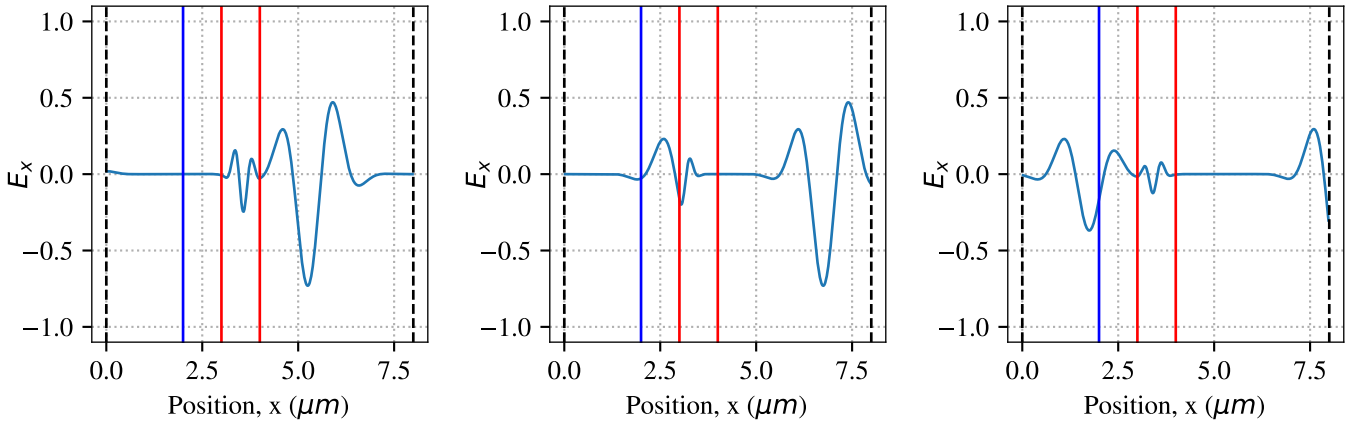
Method	**500x500 Time [s]**	**1000x1000 Time [s]**
Loop Row-Major	599	2408
Loop Col-Major	586	2488
Vectorized	5.6	25.2
Numba Loop Row-Major	1.4	5.5
Numba Loop Col-Major	5.7	31.7
Numba Vectorized	6.8	25.4
Fastest	1.1	4.4



(a) $0fs$ (b) $5fs$ (c) $10fs$

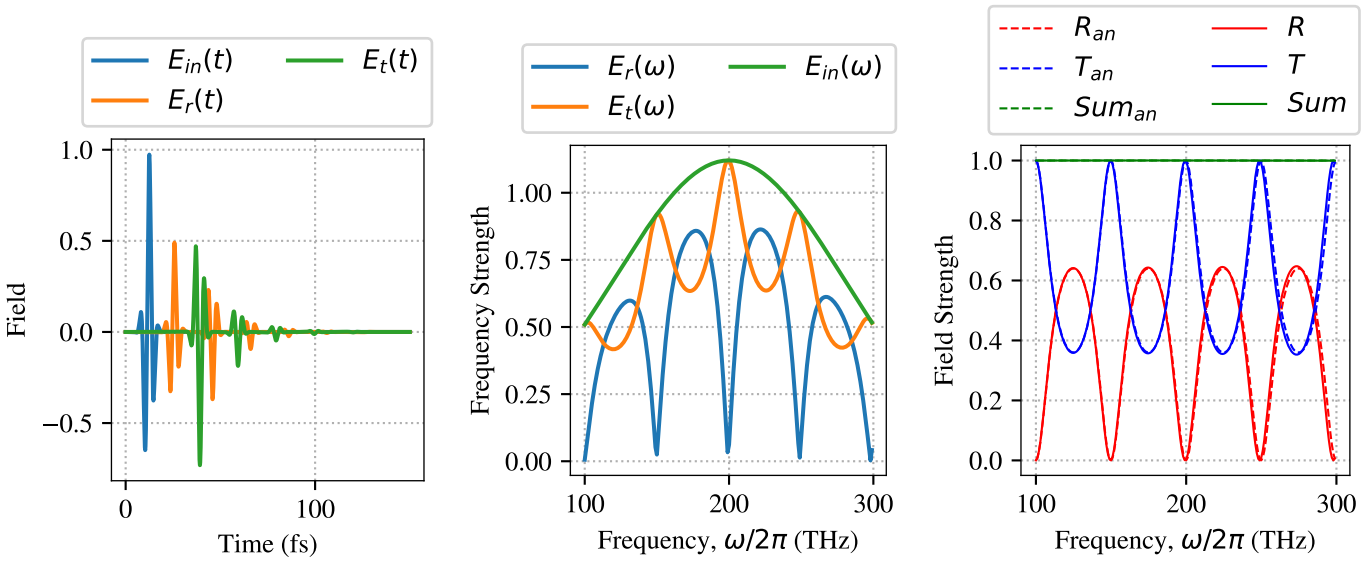


(d) $15fs$ (e) $20fs$ (f) $25fs$



(g) $30fs$ (h) $35fs$ (i) $40fs$

Figure 4. Snapshots of dielectric film shown by the red lines. As expected, the field oscillates inside the film, generating transmitted and reflected fields.

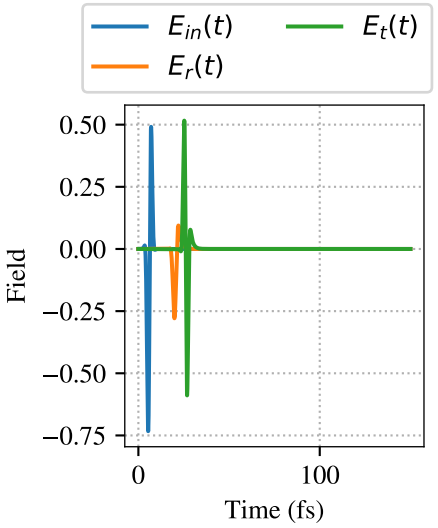


(a) Field over Time (b) Fourier Transform (c) Transmission and Reflection

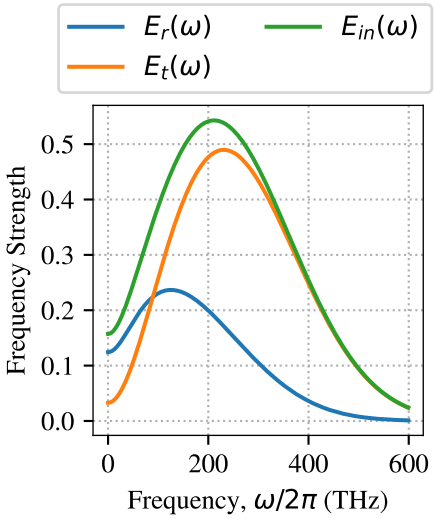
Figure 5. Fourier Transform of numerical solution and the corresponding transmission and reflection coefficients compared with the analytical solution.

Table II. Timing results for a grid size of 3008x3008 over 1000 times steps averaged over three trials. Values are rounded due to the correct significant digits to reflect variations between the three trials. The 8 and 16 processes times varied widely between (9-14)s for 8 processes and (8-11)s for 16 processes, likely due to cluster availability or communication bottlenecks.

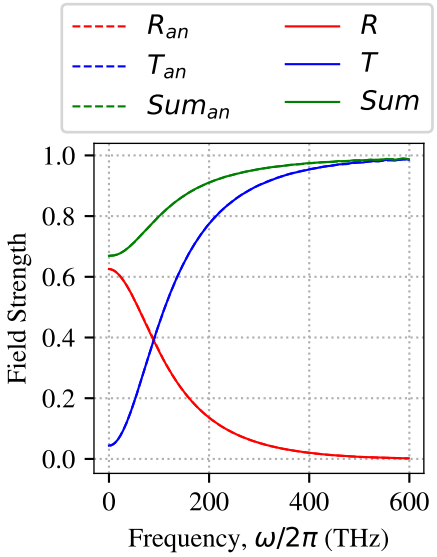
N Tasks	**Loop Time** [s]	**User+Sys Time** [s]
1	66	68
2	37	39
4	23	25
8	12	14
16	10	12



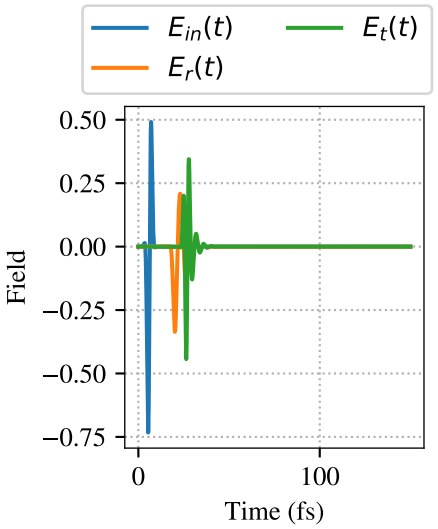
(a) Field over Time: 200nm slab



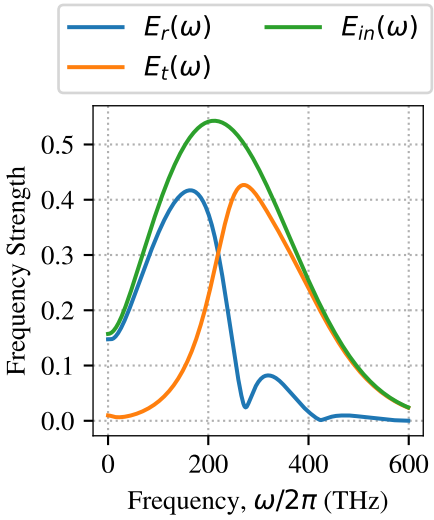
(b) Fourier Transform: 200nm slab



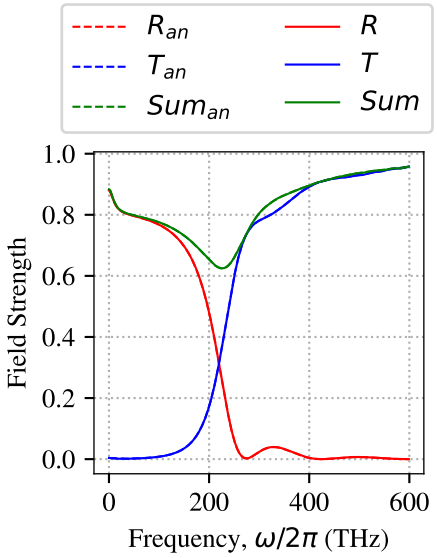
(c) Transmission and Reflection: 200nm slab



(d) Field over Time: 800nm slab

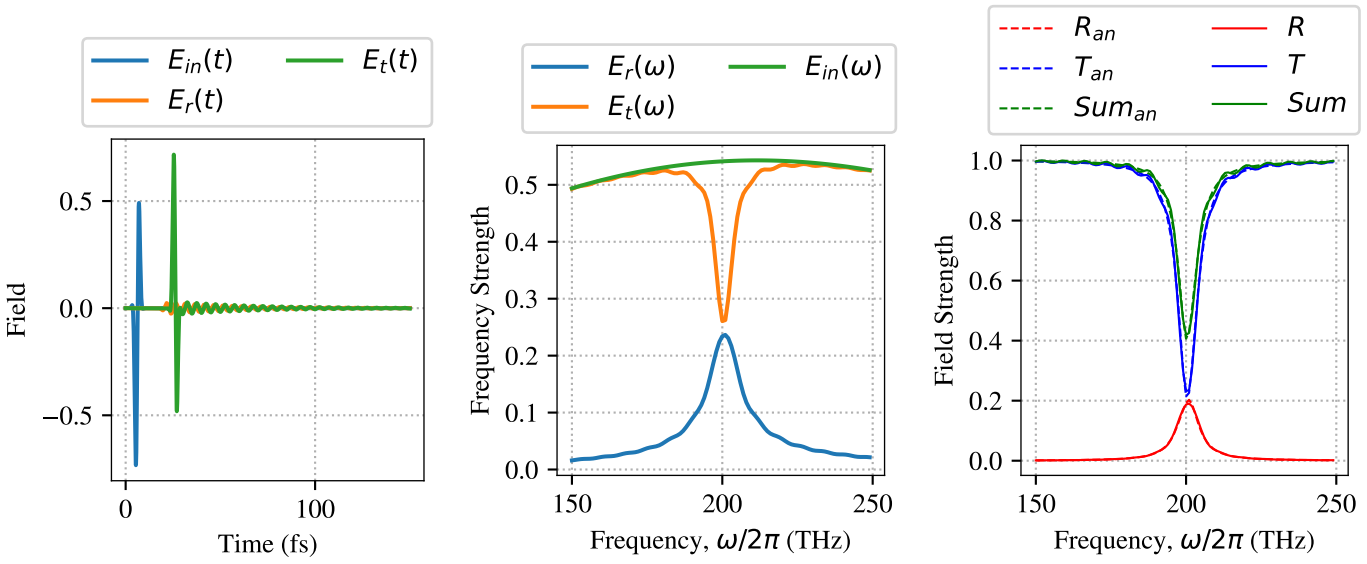


(e) Fourier Transform: 800nm slab



(f) Transmission and Reflection: 800nm slab

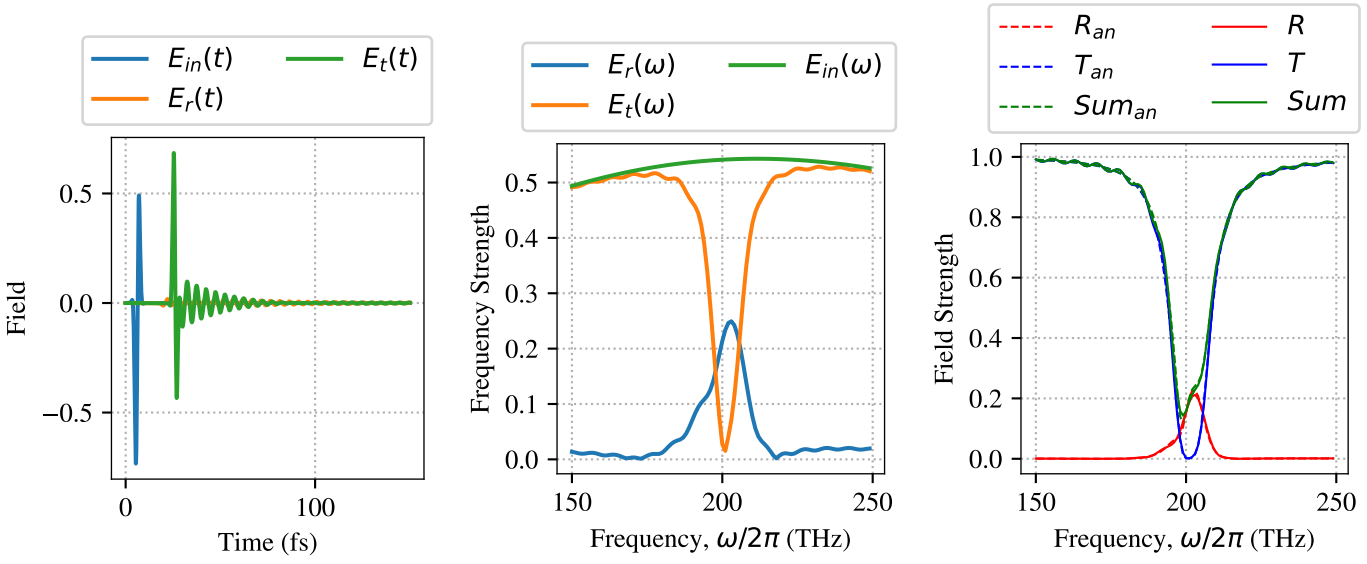
Figure 6. Field over time, Fourier transform and the corresponding transmission and reflection coefficients compared with the analytical solution for the Drude model. As expected, the numerical solution closely matches the analytical solution.



(a) Field over Time: 200nm slab

(b) Fourier Transform: 200nm slab

(c) Transmission and Reflection: 200nm slab

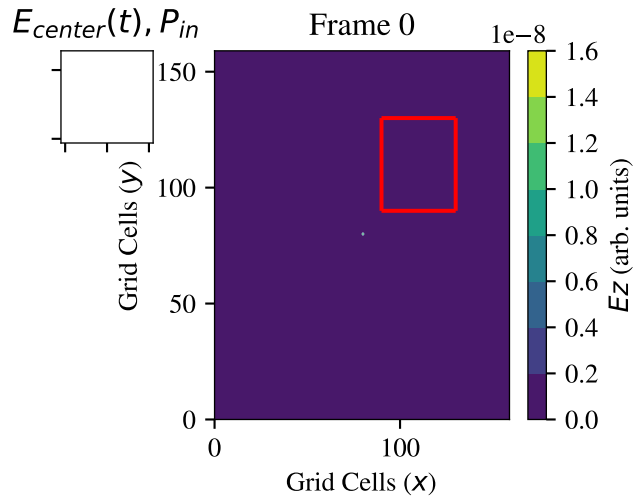


(d) Field over Time: 800nm slab

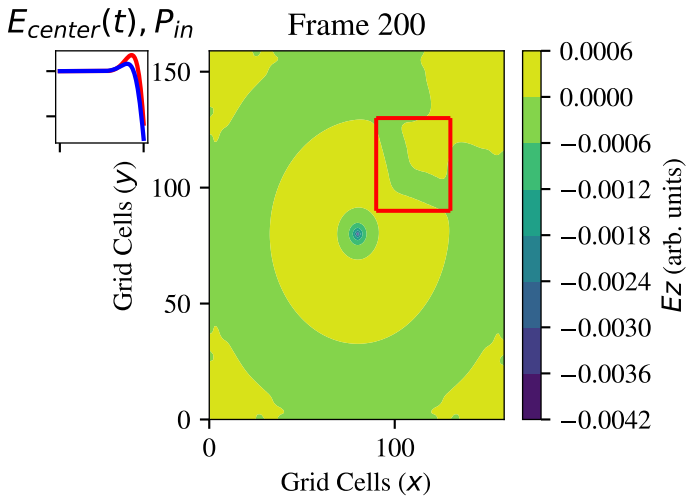
(e) Fourier Transform: 800nm slab

(f) Transmission and Reflection: 800nm slab

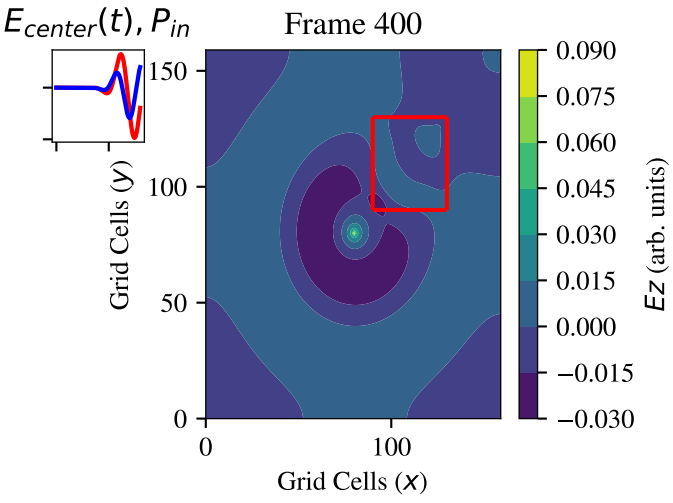
Figure 7. Field over time, Fourier transform and the corresponding transmission and reflection coefficients compared with the analytical solution for the Lorentz model. As expected, the numerical solution closely matches the analytical solution.



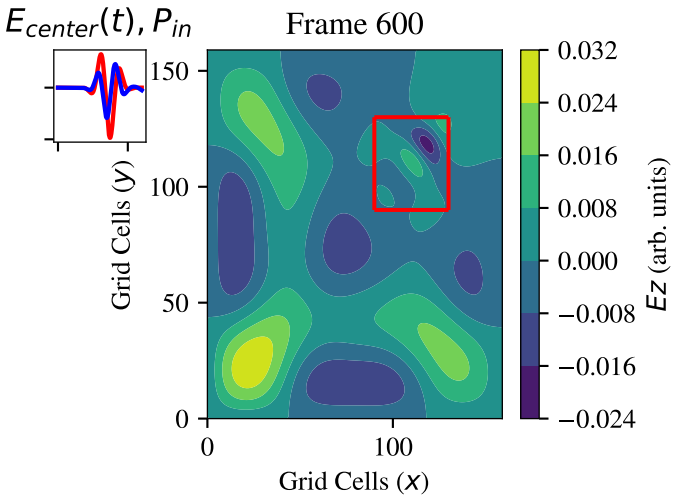
(a) Frame 0



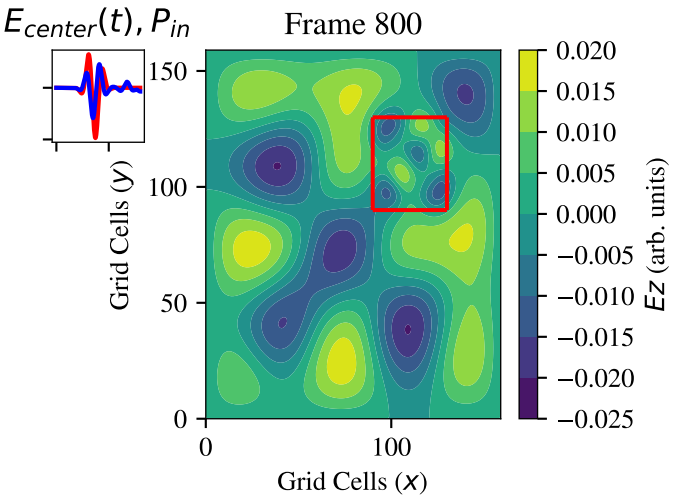
(b) Frame 200



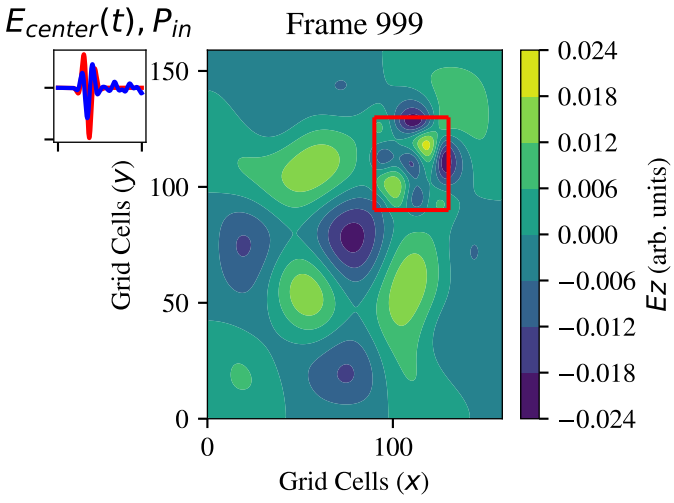
(c) Frame 400



(d) Frame 600

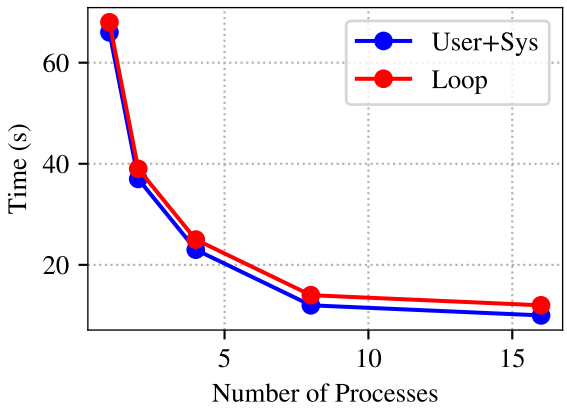


(e) Frame 800

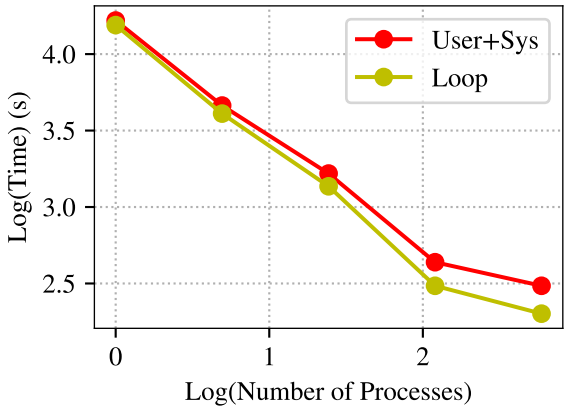


(f) Frame 999

Figure 8. Snapshots of 160x160 grid field over 1000 time steps with a dielectric box of $\epsilon = 9$. As expected, the box delays the field in the surrounding region compared with the rest of the field. We can also observe the development of nodes and anti-nodes.

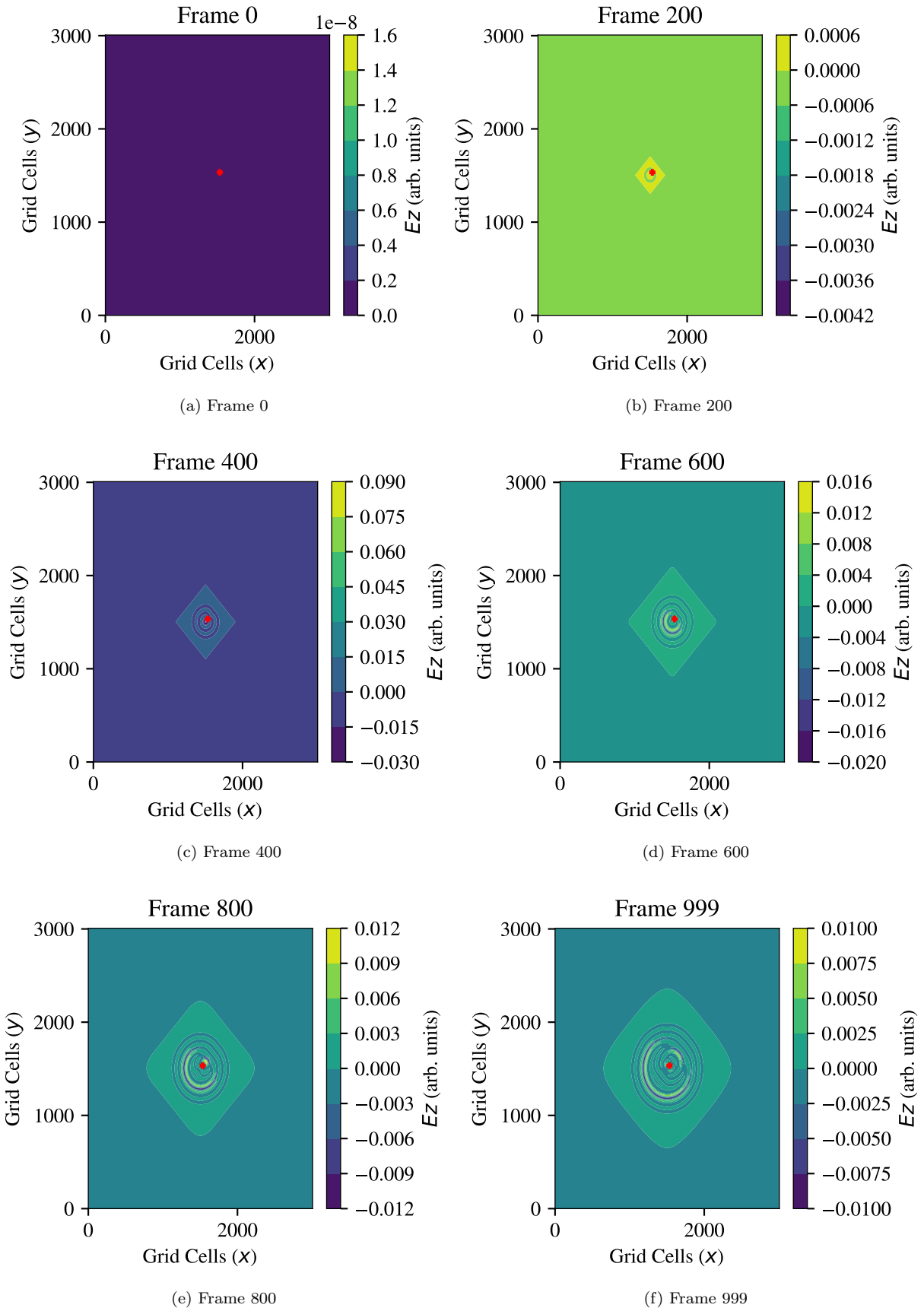


(a) Regular Plot



(b) Log-Log Plot

Figure 9. Timing as a function of parallel processes from the times in Tab. II, showing a reasonably linear trend, which may be off due to the notes in Tab. II



(a) Frame 0

(b) Frame 200

(c) Frame 400

(d) Frame 600

(e) Frame 800

(f) Frame 999

Figure 10. Snapshots of 3008x3008 grid field over 1000 time steps with a dielectric box of $\epsilon = 9$. As expected, the box delays the field in the surrounding region compared with the rest of the field. This was generated with 8 parallel processes. Note that the small pulse plots like the ones from Fig. 8 are omitted to avoid having to pass an extra message.