The Journal of Open Source Software

FDTDX: High-Performance Open-Source FDTD Simulation with Automatic Differentiation

Yannik Mahlau [1¶], **Frederik Schubert** [1], **Lukas Berg** [1], **and Bodo Rosenhahn** [1]

1 Institute of Information Processing, Leibniz University Hannover, Germany ¶ Corresponding author

DOI: 10.21105/joss.08912

Software

- Review
- Repository
- Archive

Editor: Fruzsina Agocs

Reviewers:

- @andrewgiuliani
- @victorapm

Submitted: 10 July 2025
Published: 13 January 2026

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

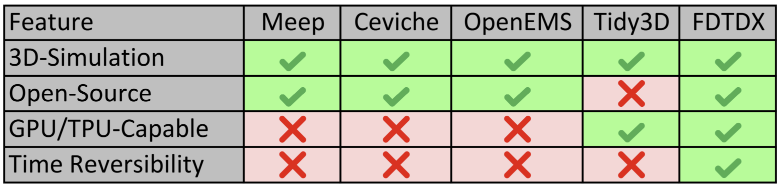
The ability to precisely manipulate the propagation of light is an increasingly critical factor in many modern applications. Aside from analytical solutions for simple geometries, the majority of practical problems require robust numerical methods such as the finite-difference time-domain (FDTD) method for discretizing Maxwell's equations in both space and time. We present FDTDX, an efficient implementation of the FDTD method with GPU acceleration through the JAX framework. It provides a simple user interface for specifying a simulation scene as well as a suite of tools for inverse design. Using the time reversibility of Maxwell's equations, gradients for optimizing the geometry of a design can be computated efficiently within FDTDX.

Statement of Need

FDTDX implements the FDTD algorithm, which aims to simulate Maxwell's equations $\frac{\partial H}{\partial t} = -\frac{1}{\mu}\nabla \times E$ and $\frac{\partial E}{\partial t} = \frac{1}{\epsilon}\nabla \times H$, where E and H are the electric and magnetic fields. This algorithm has been used in a number of research applications, for example in the field of photonic integrated circuits (Augenstein & Rockstuhl, 2020), optical computing (Mahlau, Schier, et al., 2025) or quantum computing (Larsen et al., 2025).

The FDTD algorithm has been well known for a long time and a number of open-source packages already implement it. However, most previous packages implement the algorithm only for CPU, thereby missing out on significant speedups through GPU acceleration. Additionally, the implementation of the FDTD algorithm in JAX allows for automatic differentiation using a specialized algorithm based on the time reversibility of Maxwell's equations (Schubert et al., 2025). In contrast to the adjoint method (Pontryagin, 2018), our custom gradient algorithm can calculate a gradient in the time-domain without the need to save the electric and magnetic field after every time step. This enables memory efficient inverse design, i.e. topological optimization of optical components using gradient descent. The simulation capabilities of FDTDX are useful for physicists from all kinds of backgrounds, while the inverse design capabilities are targeted more towards computational scientists in particular.

A non-exhaustive list of FDTD implementations includes the popular Meep (Oskooi et al., 2010), which was developed almost 20 years ago for execution on CPU and is still widely used today. Other frameworks for CPU only include OpenEMS (Liebig, 2024), fdtd (Laporte, 2024) and Ceviche (Hughes et al., 2019). Existing open-source packages that support execution on GPU are Khronos (Hammond, 2024) and FDTD-Z (Lu & Vučković, 2024), but neither of these packages are maintained. Additionally, various commercial implementations of FDTD exist. Notably, Tidy3D (Flexcompute, 2022) is an extremely fast commercial software due to its GPU acceleration. A comparison between the different software frameworks can be seen in Figure 1.



Feature	Meep	Ceviche	OpenEMS	Tidy3D	FDTDX
3D-Simulation	✓	✓	✓	✓	✓
Open-Source	✓	✓	✓	✗	✓
GPU/TPU-Capable	✗	✗	✗	✓	✓
Time Reversibility	✗	✗	✗	✗	✓

Figure 1: Feature comparison between different FDTD software frameworks.

Implementation

As the name suggests, the Finite-Difference Time-Domain (FDTD) algorithm discretizes Maxwell's equations in space and time. To compute the curl operation efficiently using only a single finite difference, the electric and magnetic fields are staggered in both space and time according to the Yee grid (Yee, 1966). The initial electric and magnetic fields are updated in a leapfrog pattern. First, the electric field is updated based on the magnetic field. Afterwards, the newly computed electric field is the basis for updating the magnetic field. The staggering through the Yee grids makes these updates very efficient, but as a consequence fields need to be interpolated for accurate measurements. In FDTDX, physical values can be measured through different detectors, which automatically perform interpolation.

To inject light into the simulation, the Total-Field Scattered-Field (TFSF) (Taflove & Hagness, 2005) formulation of a source is used in FDTDX. This formulation allows injecting light in a single direction into the simulation. In contrast, a naive additive source implementation would emit light in both directions perpendicular to the injection plane.

Two different boundary objects can be used to prevent unwanted reflections within the simulation. A periodic boundary wraps the fields around the simulation volume and automatically injects them on the other side. This is useful for simulating large repeating areas through a single unit cell, for example in metamaterials (Yadav & Chowdhury, 2024). The other boundary object is a perfectly matched layer (PML), which absorbs incoming light. Specifically, the PML in FDTDX is implemented in the form of a convolutional PML (Roden & Gedney, 2000).

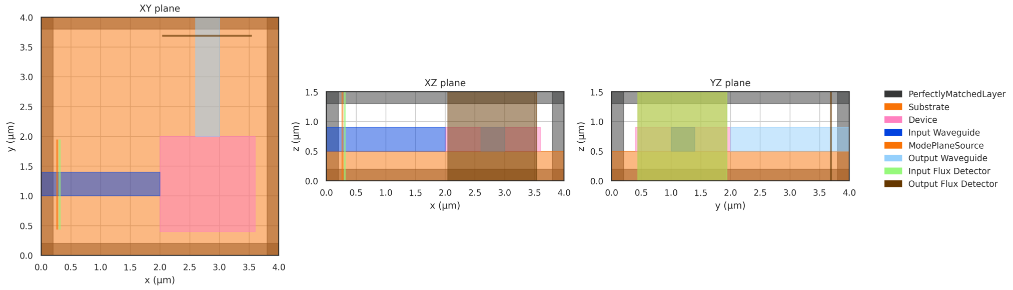


Figure 2: Visualization of a simulation scene using the `fdtdx.plot_setup` function.

In FDTDX, the specification of a simulation is simplified by the implementation of a constraint system. The position and size of sources, detectors or any other simulation objects can be specified using relative constraints. For example, it might make sense to position a detector next to a source for measuring the input energy in the simulation. If both detector and source are placed independently, then moving one of the objects also requires moving the other. With only two objects this is manageable, but with more objects such adaptations quickly become a burden. In contrast, in FDTDX the position between objects can be specified relative to each other. Consequently, if one of the objects is moved, the other object automatically moves as well. Additionally, FDTDX implements utility functions for easily plotting a visualization of the simulation scene. Such a visualization can be seen in Figure 2. Similarly, plotting functions are implemented for detectors to visualize the results of a simulation in the form of an image or video. The E_z for the same simulation scene is visualized in Figure 3.

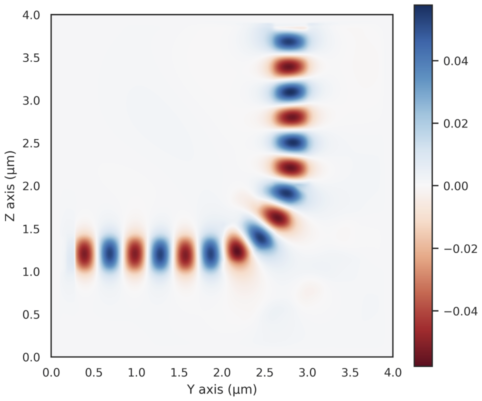


Figure 3: Visualization of the E_z field in a simulation as output of an `fdtdx.FieldDetector`.

Limitations and Future Work

At the time of publication, FDTDX only supports simulation of linear, non-dispersive materials. Additionally, lossy materials are currently only partially supported. In the future, an implementation of dispersive material models (Taflove & Hagness, 2005) is planned. Simulating non-linear materials is a difficult task due to the necessary theoretical physical knowledge as well as the capability to experimentally verify the simulation results in the lab. We are determined to tackle this, but acknowledge that this will require significant effort and time.

Further Information

The full API and tutorials can be found at the FDTDX documentation. The source code is publicly available via the corresponding Github repository. Additionally, our conference paper on large-scale FDTD simulations (Mahlau, Schubert, et al., 2025) provides a good introduction to FDTDX.

Acknowledgements

We thank Antonio Calà Lesina, Reinhard Caspary and Konrad Bethmann for helping us understand the physics behind Maxwell's equations and how to implement them within FDTD. Additionally, we acknowledge Fabian Hartmann for the initial idea of implementing a GPU accelerated FDTD algorithm. Moreover, community contributions from Marko Simic, Tianxiang Dai, Vatsal Limbachia, Agustín Galante, Rachel Smith, Quentin Wach and Robin Giesecke improved features of FDTDX.

This work was supported by the Federal Ministry of Education and Research (BMBF), Germany under the AI service center KISSKI (grant no. 01IS22093C), the European Union within the Horizon Europe research and innovation programme under grant agreement no. 101136006 – XTREME, the Lower Saxony Ministry of Science and Culture (MWK) through the zukunft.niedersachsen program of the Volkswagen Foundation and the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122) and (RO2497/17-1). Additionally, this was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 517733257.

References

Augenstein, Y., & Rockstuhl, C. (2020). Inverse design of nanophotonic devices with structural integrity. *ACS Photonics*, *7*(8), 2190–2196. https://doi.org/10.1021/acsphotonics.0c00699

Flexcompute. (2022). *Tidy3D: Hardware-accelerated electromagnetic solver for fast simulations at scale*. https://www.flexcompute.com/download-whitepaper/.

Hammond, A. (2024). *Khronos*. GitHub repository. https://github.com/facebookresearch/Khronos.jl

Hughes, T. W., Williamson, I. A., Minkov, M., & Fan, S. (2019). Forward-mode differentiation of Maxwell's equations. *ACS Photonics*, *6*(11), 3010–3016. https://doi.org/10.1021/acsphotonics.9b01238

Laporte, F. (2024). *Python 3D FDTD simulator*. GitHub repository. https://github.com/flaport/fdtd

Larsen, M., Bourassa, J., Kocsis, S., Tasker, J., Chadwick, R., González-Arciniegas, C., Hastrup, J., Lopetegui-González, C., Miatto, F., Motamedi, A., & others. (2025). Integrated photonic source of Gottesman–Kitaev–Preskill qubits. *Nature*, 1–5. https://doi.org/10.1038/s41586-025-09044-5

Liebig, T. (2024). *openEMS - open electromagnetic field solver*. General; Theoretical Electrical Engineering (ATE), University of Duisburg-Essen; https://www.openEMS.de. https://www.openEMS.de

Lu, J., & Vučković, J. (2024). *Fdtd-z: A systolic scheme for GPU-accelerated nanophotonic simulation*. GitHub repository. https://github.com/spinsphotonics/fdtdz

Mahlau, Y., Schier, M., Reinders, C., Schubert, F., Bügling, M., & Rosenhahn, B. (2025). Multi-agent reinforcement learning for inverse design in photonic integrated circuits. *Reinforcement Learning Journal*, *6*, 1794–1815. https://doi.org/10.48550/arXiv.2506.18627

Mahlau, Y., Schubert, F., Bethmann, K., Caspary, R., Lesina, A. C., Munderloh, M., Ostermann, J., & Rosenhahn, B. (2025). A flexible framework for large-scale FDTD simulations: Open-source inverse design for 3D nanostructures. *Photonic and Phononic Properties of Engineered Nanostructures XV*, *13377*, 40–52. https://doi.org/10.1117/12.3052639

Oskooi, A. F., Roundy, D., Ibanescu, M., Bermel, P., Joannopoulos, J. D., & Johnson, S. G. (2010). Meep: A flexible free-software package for electromagnetic simulations by the FDTD method. *Computer Physics Communications*, *181*(3), 687–702. https://doi.org/10.1016/j.cpc.2009.11.008

Pontryagin, L. S. (2018). *Mathematical theory of optimal processes*. Routledge. https://doi.org/10.1201/9780203749319

Roden, J. A., & Gedney, S. D. (2000). Convolution PML (CPML): An efficient FDTD implementation of the CFS–PML for arbitrary media. *Microwave and Optical Technology Letters*, *27*(5), 334–339. https://doi.org/10.1002/1098-2760(20001205)27:5%3C334::AID-MOP14%3E3.0.CO;2-A

Schubert, F., Mahlau, Y., Bethmann, K., Hartmann, F., Caspary, R., Munderloh, M., Ostermann, J., & Rosenhahn, B. (2025). Quantized inverse design for photonic integrated circuits. *ACS Omega*, *10*(5), 5080–5086. https://doi.org/10.1021/acsomega.4c10958

Taflove, A., & Hagness, S. C. (2005). *Computational electrodynamics: The finite-difference time-domain method* (3rd ed.). Artech House. https://doi.org/10.1016/B978-012170960-0/50046-3

Yadav, R., & Chowdhury, R. (2024). Impact of unit cell variation on visible spectrum multiband metamaterial absorbers. *2024 IEEE Microwaves, Antennas, and Propagation Conference*

(MAPCON), 1–5. https://doi.org/10.1109/MAPCON61407.2024.10923310

Yee, K. (1966). Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, *14*(3), 302–307. https://doi.org/10.1109/TAP.1966.1138693