JaxLayerLumos: A JAX-based Differentiable Simulator for Multilayer Optical/RF Structures

19 May 2025

Summary

JaxLayerLumos is an open-source Python software package for simulating electromagnetic wave interaction with multilayer structures using the transfer-matrix method (TMM). Designed for researchers and engineers working with applications from optics and photonics to radio frequencies, it provides a computationally efficient and intuitive tool for calculating properties like reflection and transmission across a broad spectral range. A key feature of JaxLayerLumos is its implementation in JAX (Bradbury et al. 2018), which enables automatic differentiation with respect to any input parameter (e.g., layer thickness, refractive index, permeability) and allows for efficient execution on hardware accelerators like GPUs and TPUs. This differentiability is particularly valuable for gradient-based optimization in device design and for integrating simulations into machine learning workflows, facilitating the discovery of novel materials and devices.

Statement of Need

The design of multilayer structures is fundamental to numerous applications, including optical filters, advanced solar cells, engineered structural coloration, and radar-absorbing materials. The transfer-matrix method (Born and Wolf 1999) is a cornerstone analytical technique for modeling such systems. While several TMM implementations exist (e.g., (Byrnes 2012-2024), (Niklas Wulff and Martin F. Schubert 2020)), many primarily focus on the optical (UV-Vis-IR) spectrum and may not readily support simulations involving magnetic materials or the lower frequencies typical of radio frequency (RF) and microwave applications. There is a growing need for tools that are not only fast and flexible across traditional optical ranges but also explicitly cater to these lower frequencies while seamlessly integrating with modern computational paradigms.

JaxLayerLumos addresses this need by offering a JAX-based TMM framework. Its core advantages include:

Differentiability: Automatic computation of gradients for any simulation parameter, crucial for inverse design and sensitivity analysis.

Performance: Leveraging JAX's JIT compilation and hardware acceleration (CPU, GPU, TPU) for rapid calculations, especially beneficial for large parameter sweeps or training machine learning models.

Broad Spectral Applicability & Flexibility: Support for complex refractive indices *and permeabilities* (essential for RF/microwave and magnetic materials), customizable layer stacks, varying incidence angles, and both TE and TM polarizations, enabling simulations across a wide range of frequencies from optical to RF.

Integration: Easy integration with the Python scientific computing ecosystem, including optimization libraries and machine learning frameworks.

These features make JaxLayerLumos particularly suited for research at the interface of computational electromagnetics and machine learning. This includes training neural networks for inverse design problems (i.e., predicting layer configurations from target spectral responses) across diverse spectral regions or performing large-scale optimization of device performance for both optical and RF applications. The software provides a lightweight, open-source alternative to some commercial packages, with a focus on speed, differentiability, and ease of use for advanced research.

Mathematics

The core of JaxLayerLumos implements the transfer-matrix method (TMM) for calculating the reflection R and transmission T of electromagnetic waves through a stack of L planar layers. Each layer j is defined by its thickness d_j, complex relative permittivity $\varepsilon_{r,j}$, and complex relative permeability $\mu_{r,j}$. The complex refractive index is $n_j = \sqrt{\varepsilon_{r,j}\mu_{r,j}}$.

For a given wavelength λ (or angular frequency ω) and incidence angle θ_0, the propagation of light is described by interface matrices $\mathbf{D}_j$ (capturing Fresnel coefficients at the boundary between layer j and $j+1$) and propagation matrices $\mathbf{P}_j$ (capturing phase accumulation within layer j). The total transfer matrix $\mathbf{M}$ for the stack is the product of these individual matrices:

$$\mathbf{M} = (\mathbf{P}_0\mathbf{D}_0)(\mathbf{P}_1\mathbf{D}_1)\cdots(\mathbf{P}_L\mathbf{D}_L)\mathbf{P}_{L+1}$$

From the elements of $\mathbf{M}$, the complex reflection r and transmission t amplitudes are calculated, from which $R = |r|^2$ and $T = |t|^2 \times$ factor (where factor accounts for impedance and angles of incident/exit media) are derived. JaxLayerLumos uses `lax.associative_scan` in JAX for efficient parallel computation of the matrix product.

Mention of Use

JaxLayerLumos is designed for a range of applications in optical and RF science and engineering. Example use cases provided with the software include:

- Design of radar-absorbing materials and frequency-selective surfaces by simulating spectra in microwave and radio frequency ranges, leveraging its support for magnetic material properties.
- Optimization of thin-film structures for targeted spectral responses using Bayesian optimization or gradient-based methods across optical and RF spectra.
- Analysis and design of solar cells, including multi-junction configurations.
- Exploration of structural coloration in nature and for novel material design.
- Inverse design of optical coatings and RF devices by training Transformer-based models on datasets generated by JaxLayerLumos.

The software's differentiability and performance make it a valuable tool for researchers exploring complex electromagnetic systems and for educators teaching computational photonics and applied electromagnetics.

Acknowledgements

References

Born, Max, and Emil Wolf. 1999. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light.* 7th ed. Cambridge University Press.

Bradbury, James, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, et al. 2018. "JAX: Composable Transformations of Python+NumPy Programs." http://github.com/jax-ml/jax.

Byrnes, Steven. 2012-2024. "Tmm: A Transfer-Matrix Method (TMM) Optics Calculator." *GitHub Repository.* https://github.com/sbyrnes321/tmm; GitHub.

Niklas Wulff and Martin F. Schubert. 2020. "MLResearchAtOSRAM/tmm_fast: v0.1.3." Zenodo. https://doi.org/10.5281/zenodo.3713950.