

Survey of Open-Source Simulation Environments for Machine Learning in Analysing Environmental Impacts on Acoustic and Electromagnetic Wave Propagation*

Maciej KACZYŃSKI and Zbigniew PIOTROWSKI

Military University of Technology, Warsaw, Poland

Correspondence should be addressed to: Maciej KACZYŃSKI, maciej.kaczynski@wat.edu.pl

* Presented at the 46th IBIMA International Conference, 26-27 November 2025, Ronda, Spain

Abstract

The simulation of acoustic and electromagnetic wave propagation is essential for understanding environmental impacts, advancing wireless communication, and optimizing architectural and sensing systems. Open-source simulators provide reproducible, accessible, and extensible platforms that can be readily integrated with machine learning (ML) frameworks. This integration enables data-driven optimization, inverse problem solving, and synthetic dataset generation, which are increasingly important for both scientific research and business applications such as smart city planning, environmental monitoring, and next-generation communication networks. This paper surveys state-of-the-art open-source simulation environments, focusing on their programming foundations, dimensional capabilities, and support for ML integration. The evaluation considers both advantages and limitations, with emphasis on computational requirements, simulation fidelity, scalability to large-scale problems, and integration with ML.

Keywords: Acoustic propagation, electromagnetic wave propagation, open-source simulation, machine learning.

Introduction

The accurate simulation of acoustic and electromagnetic wave propagation is essential for advancing environmental analysis, wireless communications, and architectural acoustics. Recent developments in open-source simulation tools have enabled reproducible research, high-fidelity modelling, and seamless integration with machine learning (ML) frameworks. Such environments facilitate data-driven optimization, inverse problem solving, and large-scale scenario analysis. This survey reviews state-of-the-art open-source simulators, highlighting their programming foundations, dimensional capabilities, ML compatibility, and trade-offs, with emphasis on their applicability to analysing environmental impacts on wave propagation.

Open-Source Acoustic Wave Simulators

The accurate modelling of acoustic wave propagation is essential for applications spanning architectural acoustics, ultrasound imaging, and audio signal processing. Open-source simulation environments provide accessible, reproducible, and extensible platforms that can be readily coupled with ML frameworks to advance these domains. Table 1 summarizes representative open-source simulators for acoustic wave propagation.

Table 1: Overview of open-source simulators for acoustic wave propagation

Name	Related article	Repository link
CHORAS	(Willemsen et al., 2025)	https://github.com/Building-acoustics-TU-Eindhoven/CHORAS
ElasticMatrix	(Ramasawmy et al., 2020)	https://github.com/dannyramasawmy/ElasticMatrix
j-Wave	(Stanziola et al., 2023)	https://github.com/ucl-bug/jwave
mSOUND	(Gu et al., 2021)	https://github.com/m-SOUND/mSOUND
OptimUS	(Wout et al., 2024)	https://github.com/optimuslib/optimus
PyAWD	(Tribel et al., 2025)	https://github.com/pascaltribel/pyawd
Pyroomacoustics	(Scheibler et al., 2018)	https://github.com/LCAV/pyroomacoustics
simwave	(Souza et al., 2022)	https://github.com/HPCSys-Lab/simwave

CHORAS is an open-source simulator written in Python, aimed at modelling 3D acoustic wave propagation in built environments, especially in architectural acoustics. While it does not include native ML tools, its Python base allows seamless integration with ML frameworks such as PyTorch or TensorFlow for tasks like optimization or surrogate modelling. The main advantages are its flexibility, open-source nature, and reproducibility of experiments, but it can be computationally intensive for large 3D domains and lacks dedicated ML modules.

ElasticMatrix is implemented in MATLAB and focuses on 2D elastic and acoustic wave propagation. Although not inherently ML-oriented, MATLAB's ML toolboxes can extend its capabilities for optimization or inverse modelling. Its strengths are simplicity, accessibility for MATLAB users, and efficiency for 2D tasks; however, it lacks 3D support and is constrained by MATLAB's proprietary licensing.

J-Wave is written in Julia and supports both 2D and 3D acoustic simulations, benefiting from Julia's high-performance just-in-time compilation. The simulator can integrate with Julia's ML libraries such as Flux to address inverse problems or data-driven optimization. It offers modern performance advantages and flexibility but suffers from a smaller user community and limited documentation relative to established libraries.

mSOUND is a C++-based simulator with Python bindings, designed for high-fidelity 3D sound field simulations. It indirectly supports ML workflows by providing accurate simulation data suitable for ML training or testing. Its primary advantages are speed, precision, and Python interoperability, though installation is complex and its user base relatively small.

OptimUS is developed in Python and tailored for 3D ultrasound simulations. It supports ML integration via ML-based optimization for probe design and image reconstruction. Its advantage lies in specialization for ultrasound and ease of use within ML pipelines, but it has a narrow focus and an evolving ecosystem compared to broader libraries.

PyAWD is a Python-based simulator for 2D acoustic wave diffraction and propagation. While not ML-native, it can be extended to ML-driven inverse modelling and parameter estimation through Python's ecosystem. Its advantages are ease of use and lightweight design, but it is limited to 2D and lacks optimization for large-scale, high-resolution problems.

Pyroomacoustics is a Python library specialized in 3D room acoustics and sound field simulations, with emphasis on audio and speech processing. Unlike many simulators, it provides utilities for generating datasets for ML tasks such as speech enhancement and source localization. It is widely used, actively maintained, and ML-friendly, however simplifies physics relative to full wave-based solvers and is less precise at high frequencies.

Simwave is implemented in C++ with a focus on high-performance computing for 2D and 3D acoustic wave propagation. While lacking built-in ML support, its ability to generate large, high-resolution datasets makes it valuable for ML workflows. Its strengths are scalability and efficiency for cluster-based simulations, but it has a steep learning curve and is harder to deploy than lightweight frameworks.

Open-Source Electromagnetic Wave Simulators

Electromagnetic wave propagation is fundamental to wireless communication, sensing, and optical technologies. Open-source simulation tools have recently advanced to support complex propagation modelling and integration with ML for optimization and data-driven analysis. Table 2 presents an overview of open-source simulators for electromagnetic wave propagation.

Table 2: Overview of open-source simulators for electromagnetic wave propagation

Name	Related article	Repository link
DeepMIMO	(Alkhateeb, 2019)	https://github.com/DeepMIMO/DeepMIMO
DiffeRT2d	(Eertmans et al., 2024)	https://github.com/jeertmans/DiffeRT2d
Sionna / Sionna RT	(Hoydis et al., 2023)	https://github.com/NVlabs/sionna
OpenAirLink	(Deshpande et al., 2024)	https://github.com/n3martix/OpenAirLink
OpenGERT	(Tadik et al., 2025)	https://github.com/serhatadik/OpenGERT
PyWaveProp	(Lytaev, 2023)	https://github.com/mikelytaev/wave-propagation
Veneris / Opal	(Egea-Lopez et al., 2019)	https://gitlab.com/esteban.egea/veneris
WaveSim	(Osnabrugge et al., 2016)	https://github.com/IvoVellekoop/wavesim

DeepMIMO is written in Python and MATLAB, providing large-scale 3D wireless channel datasets rather than direct wave solvers. It is designed with ML support at its core, enabling training and evaluation of data-driven communication models. Its advantage is strong ML integration and realistic scenarios, but it is limited to predefined datasets rather than flexible physics-based simulations.

DiffeRT2d is implemented in Python with differentiable rendering for 2D electromagnetic wave propagation. It is inherently ML-compatible since it is designed for gradient-based optimization and inverse problems. Its main advantage is native differentiability for ML workflows, while the disadvantage is the restriction to 2D and early-stage community adoption.

Sionna / Sionna RT is developed in Python and integrated with TensorFlow, supporting 3D wireless communication and ray-tracing simulations. It is explicitly built for end-to-end ML integration, making it a strong tool for AI-driven communication system design. Advantages are tight ML compatibility and modern architecture, but disadvantages include heavy GPU requirements and relative newness compared to classical solvers.

OpenAirLink is written in Python and C++, targeting 3D wave propagation for wireless communication systems. It includes ML interfaces for adaptive communication research. Its main advantage is modern design for next-generation networks, but disadvantages include smaller community support and limited documentation.

OpenGERT is implemented in Python and C++, designed for 3D ray-tracing and wave propagation in complex urban environments. It is compatible with ML workflows by generating synthetic training datasets for wireless channel prediction. Advantages include realistic environment modelling and open-source access, while disadvantages are limited maturity and fewer ML-ready tools compared to DeepMIMO or Sionna.

PyWaveProp is written in Python and supports both 2D and 3D scalar and vector wave propagation. It is ML-friendly due to Python interoperability, enabling integration with ML frameworks for inverse design or optimization. Advantages include simplicity and accessibility, but disadvantages are limited performance on large-scale 3D simulations and a smaller community.

Veneris / Opal is developed in C++ and Java, offering 3D vehicular communication and electromagnetic propagation simulation. It can be combined with ML frameworks for intelligent transportation systems and channel prediction. Its advantages are realism in vehicular contexts and open-source access, while disadvantages are complexity of deployment and limited general-purpose electromagnetic simulation capabilities.

WaveSim is implemented in C++ with a Python interface, focusing on 2D and 3D optical/electromagnetic wave propagation using advanced numerical solvers. It is compatible with ML frameworks through dataset generation and adjoint-based optimization. Its advantages are high accuracy and performance, however disadvantages include high computational cost.

Summary

This survey reviewed open-source simulators for acoustic and electromagnetic wave propagation with emphasis on their integration with ML. The tools differ in programming languages, dimensional capabilities, and suitability for ML workflows. Open-source accessibility fosters reproducibility, yet trade-offs remain between computational cost, fidelity, and ML readiness. Future developments should integrate high-fidelity solvers with native ML support to enable scalable, data-driven environmental impact analysis.

Acknowledgment

This research work was funded by the project of the Military University of Technology titled: “New Neural Network Architectures for Signal and Data Processing in Radiocommunications and Multimedia.” Project No. UGB 22-054/2025.

Bibliography

- Alkhateeb, A. (2019) “DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications.” arXiv. Available at: <https://doi.org/10.48550/arXiv.1902.06435>.
- Deshpande, Y., Wang, X. and Kellerer, W. (2024) “OpenAirLink: Reproducible Wireless Channel Emulation using Software Defined Radios.” arXiv. Available at: <https://doi.org/10.48550/arXiv.2404.09660>.
- Eertmans, J., Oestges, C. and Jacques, L. (2024) “DiffeRT2d: A Differentiable Ray Tracing Python Framework for Radio Propagation,” *Journal of Open Source Software*, 9(98), p. 6915. Available at: <https://doi.org/10.21105/joss.06915>.
- Egea-Lopez, E. et al. (2019) “Vehicular Networks Simulation With Realistic Physics,” *IEEE Access*, 7, pp. 44021–44036. Available at: <https://doi.org/10.1109/ACCESS.2019.2908651>.
- Gu, J. and Jing, Y. (2021) “mSOUND: An Open Source Toolbox for Modeling Acoustic Wave Propagation in Heterogeneous Media,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 68(5), pp. 1476–1486. Available at: <https://doi.org/10.1109/TUFFC.2021.3051729>.
- Hoydis, J. et al. (2023) “Sionna: An Open-Source Library for Next-Generation Physical Layer Research.” arXiv. Available at: <https://doi.org/10.48550/arXiv.2203.11854>.
- Lytaev, M.S. (2023) “Tropospheric radio wave propagation modeling in Python 3 using PyWaveProp,” in 2023 IEEE 11th Asia-Pacific Conference on Antennas and Propagation (APCAP). 2023 IEEE 11th Asia-Pacific Conference on Antennas and Propagation (APCAP), pp. 1–2. Available at: <https://doi.org/10.1109/APCAP59480.2023.10470193>.
- Osnabrugge, G., Leedumrongwatthanakun, S. and Vellekoop, I.M. (2016) “A convergent Born series for solving the inhomogeneous Helmholtz equation in arbitrarily large media,” *Journal of Computational Physics*, 322, pp. 113–124. Available at: <https://doi.org/10.1016/j.jcp.2016.06.034>.
- Ramasawmy, D.R., Cox, B.T. and Treeby, B.E. (2020) “ElasticMatrix: A MATLAB toolbox for anisotropic elastic wave propagation in layered media,” *SoftwareX*, 11, p. 100397. Available at: <https://doi.org/10.1016/j.softx.2020.100397>.
- Scheibler, R., Bezzam, E. and Dokmanić, I. (2018) “Pyroomacoustics: A Python package for audio room simulations and array processing algorithms,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 351–355. Available at: <http://arxiv.org/abs/1710.04196> (Accessed: September 22, 2025).
- Souza, J.F. de et al. (2022) “simwave -- A Finite Difference Simulator for Acoustic Waves Propagation.” arXiv. Available at: <https://doi.org/10.48550/arXiv.2201.05278>.
- Stanziola, A. et al. (2023) “j-Wave: An open-source differentiable wave simulator,” *SoftwareX*, 22, p. 101338. Available at: <https://doi.org/10.1016/j.softx.2023.101338>.
- Tadik, S. et al. (2025) “OpenGERT: Open Source Automated Geometry Extraction with Geometric and Electromagnetic Sensitivity Analyses for Ray-Tracing Propagation Models.” arXiv. Available at: <https://doi.org/10.48550/arXiv.2501.06945>.
- Tribel, P. and Bontempi, G. (2025) “PyAWD: A Library for Generating Large Synthetic Datasets of Acoustic Wave Propagation.” arXiv. Available at: <https://doi.org/10.48550/arXiv.2411.12636>.
- Willemsen, S. et al. (2025) “Towards a Community Hub for Open-source Room Acoustics Software,” in. Available at: https://www.researchgate.net/publication/392763947_Towards_a_Community_Hub_for_Open-source_Room_Acoustics_Software.

- Wout, E., Haqshenas, R. and G elat, P. (2024) “Simulating Focused Ultrasound with the Boundary Element Method,” in 2024 IEEE UFFC Latin America Ultrasonics Symposium (LAUS). 2024 IEEE UFFC Latin America Ultrasonics Symposium (LAUS), pp. 1–4. Available at: <https://doi.org/10.1109/LAUS60931.2024.10553133>.