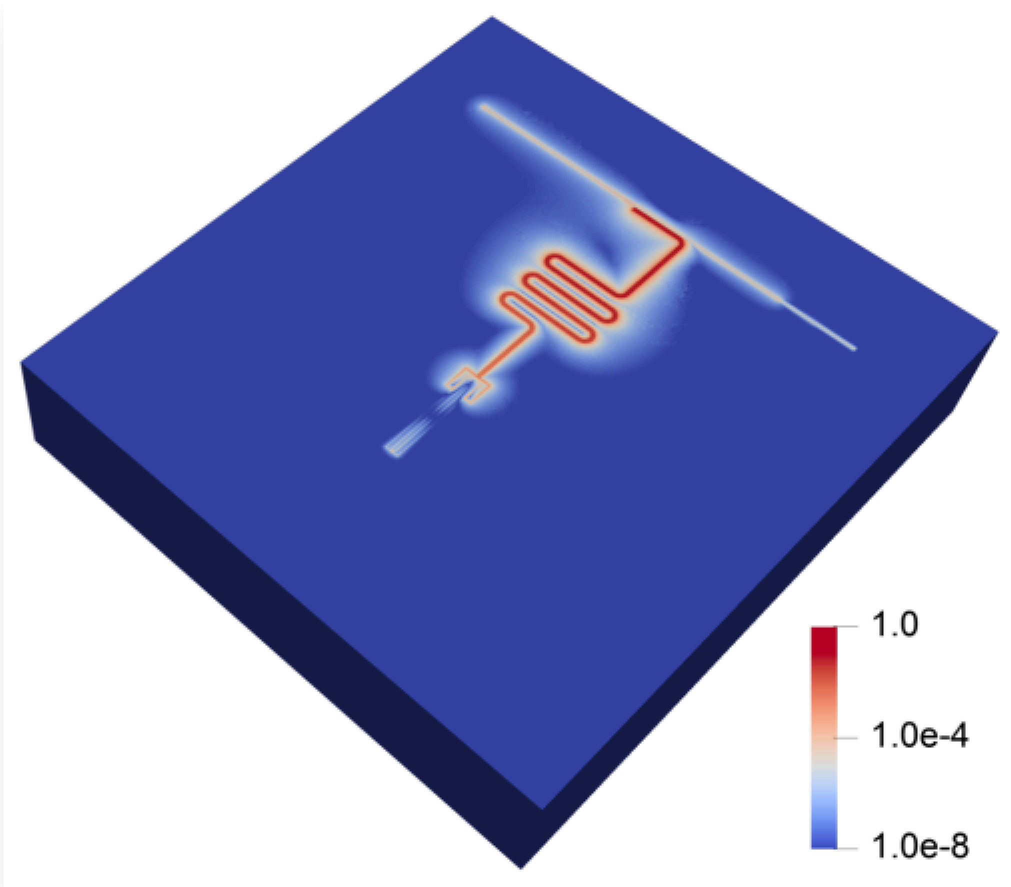


Mar 19, 2025

Open-Source Electromagnetic Simulation: FDTD, FEM, MoM

An up-to-date review of the best open-source projects in Computational Electromagnetics, including FDTD, FEM and BEM/MoM methods.

Authors: Martin D. Maas **Categories:** software **Tags:** open-source

Computational Electromagnetics encompasses a wide range of application areas, including antennas, nano-photonics, solar cells, metamaterials, lasers, and many more. There are also many ways to approach the computations required in these various areas.

There are **three major family of numerical methods** for Electromagnetic simulations: FDTD, FEM and MoM. Each of these approaches is best suited for certain cases, and has advantages and disadvantages.

In this post we discuss the basic trade-offs between these families of methods, and we give some references to some of the best open-source implementations out there.

Difference between FDTD, FEM and MoM.

There are three basic numerical approaches for electromagnetic simulation, namely:

- Finite-differences-time domain (FDTD)
- Finite Element Method (FEM)
- Method of Moments (MoM), or equivalently, Boundary Element Method (BEM)

Before digging deeper into each of these options, let’s provide a brief summary of the three alternatives:

Method	**Solver type**	**Discretization**	**Material type**
FDTD	Differential equation	Volumetric domain	Non-linear, anisotropic
FEM	Variational form	Volumetric domain	Non-linear, anisotropic, multi-physics
MoM/BEM	Integral equations	Surface currents	Linear, piecewise homogeneous

What is the FDTD method?

FDTD is the application of finite differences to Maxwell's equations, in a second order, stable, staggered-grid approach for electric and magnetic fields. The method was introduced in a seminal 1966 paper by Kane Yee, and the term "FDTD" was later popularized by Taflove during the 1980s.

Advantages of the FDTD Method

- Being based on the time domain, the FDTD method supports a **wide range of frequencies**, making it valuable for broadband analysis.
- The incorporation of **non-linear materials** is straightforward.
- It's explicit time-stepping and simplicity leads to **straightforward parallelization** and implementation in various computational environments.

Disadvantages of FDTD

The flexibility of time-domain discretization comes at a certain price, tough.

- **It's Hard to accomodate complex geometries.** As the FDTD uses a global grid-like mesh, when applied to complex geometries it ends up with high resolution even in regions where it is not required. As such, we have to expect that the mesh-size scales like $(\lambda/dx)^3$, where λ is the wavelength, and dx is the the finest geometric resolution required. This can be compensated with the use of graded meshes, but these are still global in nature, limiting their effectiveness. In view of the CFL condition, according to which the time-step Δt scales like Δx, this leads to a total computing time of $(\lambda/dx)^4$

- **The error accumulates over time.** As with any finite difference method, the propagation of a wave in the discrete grid doesn't obey the exact dispersion relations of Maxwell's equations, but rather an approximate version of them. This is called numerical dispersion error, and it can quickly become one of the main accuracy limitations of the FDTD.

The combined effect of requiring a global grid and numerical dispersion can make high frequency propagation very challenging to compute with FDTD, as the initial error (and thus the grid size) have to be really small to compensate the effect of error accumulation. See for example, this paper by Taflove where they report to have used a **discretization finer than 1/100th wavelengths** in order to achieve a 1.5 dB accuracy.

Best open-source FDTD codes

Software	License	Written in	Interface	Parallelization
Meep	GPL	C++	Python, Scheme, C++	MPI
gprMax	GPL	Python+Cython	Python	CUDA, MPI
OpenEMS	GPL	C++	Matlab, Python	MPI

- **Meep**. Developed at MIT, Meep is a highly efficient FDTD package, scriptable in Python, Scheme or callable from C++ APIs. It is parallelized with MPI, and it includes a library with support for a variety of material types. Recommended for optics and photonics applications.
- **OpenEMS**. Developed at the University of Duisburg-Essen, and parallelized with MPI. Matlab or Octave are mainly used as a scripting interfaces. The related project pyEMS provides a high-level Python interface. Recommended for RF applications.
- **gprMax**. Developed at the University of Edinburgh, gprMax was designed for modelling Ground Penetrating Radar (GPR) but can also be used to model electromagnetic wave propagation for many other applications. gprMax is command-line driven software written in Python, with performance-critical parts written in Cython/OpenMP.

FEM for Computational Eletromagnetics

The Finite Element Method (FEM) is a widely used approach for solving partial differential equations, particularly effective in computational electromagnetics due to its **flexibility in handling complex geometries**. The method is based on a weak integral formulations obtained via multiplication of the PDE by test functions and integration-by-parts arguments. The unknowns and the test functions are then restricted into suitable discrete spaces, which usually leads to sparse linear algebra problems.

The FEM, as it is usually employed on the frequency domain, **doesn't suffer from dispersion error as the FDTD**. It is also possible to use the FEM as a part of a time-stepping algorithm, in which case the same concerns about numerical dispersion will apply.

The FEM is also **very well suited for multi-physics problems** (i.e. where electromagnetism is coupled with elasticity, heat transfer, fluid dynamics), as substantial existing software can be reused.

Disadvantages of FEM

The flexibility of FEM comes at a cost, as outlined below:

- **Challenging Volumetric Meshing.** Generating accurate volumetric meshes is a complex, time-consuming task that often requires significant manual effort. Fully automated meshing methods exist, but they have limitations and may not always produce the necessary quality.
- **Steeper Learning Curve.** Despite its great flexibility, FEM tends to be harder to use than FDTD, demanding more intricate setup, deeper understanding, and a higher level of expertise.
- **Less Efficient for Homogeneous Materials and Large Open Regions.** FEM's volumetric discretization requires meshing the entire domain—even areas with uniform linear materials or vast open spaces—making it

Best open-source FEM codes for electromagnetism

Software	**License**	**Written in**	**Interface**	**Parallelization**
Elmer FEM	LGPL	Fortran	GUI, config file	MPI
FEniCS	LGPL	C++	Python, C++	MPI
Palace	Apache-2	C++	config file	MPI and GPU

- Elmer FEM. An open-source Finite Element Solver, dealing with multiphysical simulations. Built-in Electromagnetics Solvers include magnetostatic, electrostatic and wave-equation solvers. See the Elmer Models Manual for more information. Elmer resorts to a configuration file, which can be generated with a GUI. The related project pyelmer provides an alternative method for generating the required configuration files.

- FEniCS. FEniCS is a popular open-source LGPLv3-licenced software package for solving partial differential equations (PDEs). It features high-level Python and C++ interfaces, and can be run in high-performance clusters. To get started, visit the FEniCS Tutorial which includes an example in magnetostatics, or check out the official Discourse forum.

- Palace. Developed by AWS and targeting quantum-computing hardware simulations, Palace is built on MFEM and libCEED. It is a high-performance FEM solver optimized for large-scale electromagnetic simulations on HPC systems. Featuring advanced electromagnetic solvers, it is notably the most challenging to use, with a steep learning curve that reflects its HPC focus. Check out the docs, but buckle up—it's a wild ride.

The Method of Moments (MOM) or BEM in Electromagnetics

The method of moments (or boundary element method) is a frequency-domain method for performing electromagnetic simulations. The MoM enforces radiation boundary conditions automatically, without requiring to discretize a large volume of air around a given geometry of interest.

The main advantage of the MoM is that only requires a surface surfaces, giving the MoM a big computational advantage over the FDTD and the FEM. As a frequency domain method, it does not suffer from numerical dispersion.

Its main drawback is that is mainly suited to deal with linear problems and piecewise homogeneous materials, and it is harder to parallelize.

Best open-source MOM codes

Software	License	Written in	Interface	Input meshes	Parallelizatio
Bempp	MIT	Python	Python, C++	Gmsh, meshio	Shared memory
PumaEM	GPLv3	C++, Fortran	Python	Gmsh, GiD, Ansys, VRML	MPI
NEC-2	GPLv2	C++	C++,Python,Ruby	Antenna parameters	None
Traceon	Open-core	Python + C	Python	Own geometry module	Shared memory

- Bempp. Bempp is an open-source, MIT licenced, computational boundary element platform to solve electrostatic, acoustic and electromagnetic problems. Bempp uses just-in-time compiled OpenCL or Numba kernels to assemble BEM operators in CPUs or GPUs. Features include a Python interface, Fast Multipole Method acceleration via Exafmm-t, and coupled FEM/BEM computations via interfaces to FEniCS.
- PumaEM is an open-source (GPL v3 licensed) Method of Moments implementation for Electromagnetics, accelerated with the Multilevel Fast Multipole Method, and parallelized via MPI.

- NEC-2. A classical code by LLNL rewritten in C++, targeted at wire and surface antenna simulation. For a Mac OS GUI, check out CocoaNEC.
- Traceon. An open-core project focused on electron optics simulations using Boundary Element Methods and particle tracing. The project consists of an open-source version, with some features only available in a commercial version. For documentation, see the API docs.

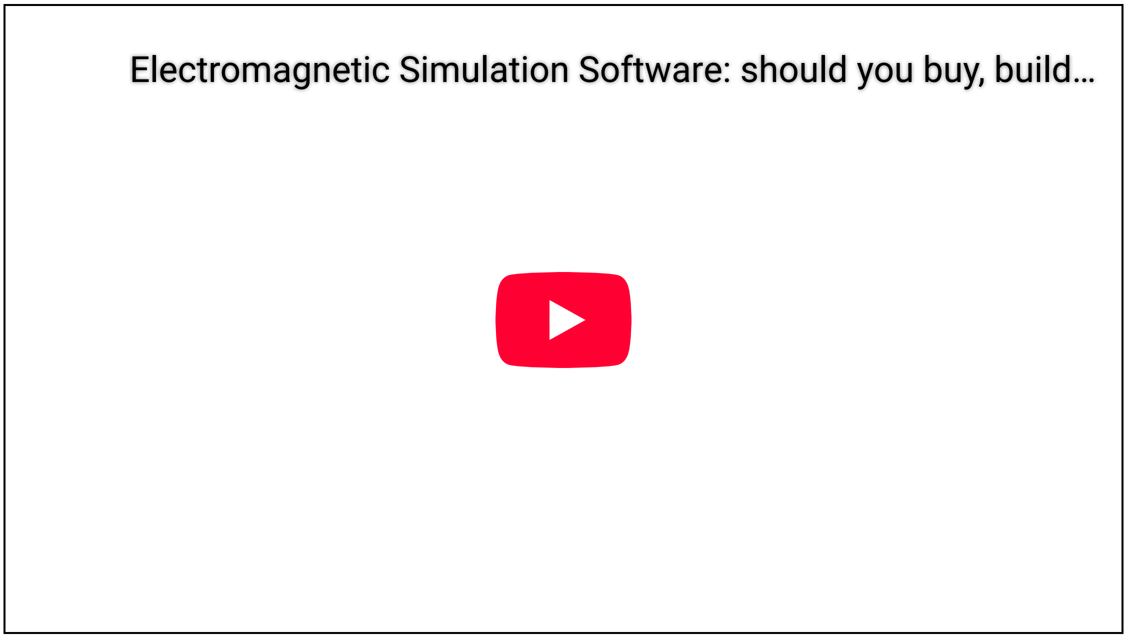
A Streamlined CAD & Simulation Service

Managing every aspect of an EM simulation project—from CAD modeling and mesh generation to pre- and post-processing—requires specialized expertise and significant time. If you're looking to streamline your workflow, outsourcing some (or all) of your CAD and simulation tasks can be a powerful solution.

At EpsilonForge, we provide comprehensive CAD and simulation services, taking full ownership of your projects to deliver accurate, actionable results without the overhead of an in-house team. Let's discuss how we can support your goals and make your simulation process seamless.

Let's Work Together

Learn More



Epsilon Forge SAS

Your On-Demand EM Simulation Partner. We help engineers and businesses solve complex electromagnetic challenges to accelerate innovation and ensure product performance.

Quick Links

Our Services

Pricing

About Us

Blog

Contact Information

Email: info@epsilonforge.com

Phone: +54 (911) 6551-3876

Address: Tronador 4122, Ciudad Autónoma de Buenos Aires, Argentina.

Get Started

Ready to accelerate R&D? Let's discuss your project.

Let's Work Together

Follow Us

Serving clients globally in advanced EM simulation and design.