

chombo-discharge: An adaptive code for gas discharge simulations in complex geometries

Robert Marskar¹

¹ SINTEF Energy Research, Norway

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Open Journals](#)

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

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

In partnership with



AMERICAN
ASTRONOMICAL
SOCIETY

This article and software are linked with research article DOI [10.3847/xxxxx](https://doi.org/10.3847/xxxxx) <- [update this with the DOI from AAS once you know it.](#), published in the *Astrophysical Journal* <- The name of the AAS journal..

Summary

chombo-discharge is a 2D and 3D adaptive code for simulating low-temperature gas discharges in complex geometries. Such discharges occur when electrons accelerate in strong electric fields and ionize the gas, and further evolution is also determined by residual space charge. Streamers, for example, are filamentary plasma dominated by space charge effects. They are the natural precursors to leader, sparks, and lightning.

Gas discharge modeling involves simulations over multiple scales in time and space. chombo-discharge reduces the cost of such simulations by using Cartesian Adaptive Mesh Refinement (AMR). It also provides support for multi-material complex geometries (gas phase, electrodes, and solid dielectrics) through an embedded boundary (EB) formulation. Geometries can be constructed using either constructive solid geometry (CSG) or imported as surface meshes. The code uses a solver-centered modular design where larger applications are developed by strongly or weakly coupling numerical solvers shipped with the chombo-discharge base code, using existing C++ interfaces. Several such applications are included in the chombo-discharge code. Depending on their needs, users can therefore enter the framework at several levels. E.g. they only need to learn textual interfaces interfaces when using existing applications; use solver C++ APIs when developing new applications, or use the EB AMR infrastructure when contributing entirely new solvers.

Several solvers exist in chombo-discharge, all of which are parallelized and compatible with EBs and AMR:

- Advection-diffusion-reaction solvers.
- Helmholtz equation solvers, using geometric multigrid.
- Electrostatic solvers (with support for discontinuous coefficients).
- Kinetic Monte Carlo solvers.
- Radiative transfer solvers (continuum and Monte Carlo solvers).
- Various particle solvers, e.g. for Monte Carlo radiative transfer, tracer particles, Brownian walkers, or kinetic Particle-In-Cell.
- Volumetric and cut-cell ODE solvers.

All solvers exist as stand-alone applications, and many of them are also coupled through more complex physics applications that aim at resolving different types of discharge phenomena (e.g. statistical inception models or streamer discharges). The interaction of these solvers occurs through a common AMR core, which can also use dual grids where e.g. fluid and particle kernels are load-balanced separately. chombo-discharge uses the Chombo infrastructure for the AMR and EB infrastructure, and is parallelized using MPI. However, chombo-discharge supplies most solver and infrastructure discretizations.

Statement of need

There is already a number of discharge simulation codes currently available. Commercial codes used for simulating discharges include COMSOL, ANSYS Fluent, PLASIMO, VSIM/VPIC, and Vizglow. Several non-commercial codes also exist, such as Afivo-streamer (Teunissen & Ebert, 2017), which is also open source and uses Cartesian AMR.

While chombo-discharge is not the only open-source discharge simulation code, it has a number of unique features. In particular, chombo-discharge supports complex geometries, and is therefore useful in many discharge-related science applications. Lightning initiation investigations (from hydrometeors), high-voltage technology, plasma medicine, and plasma-assisted combustion are typical examples. The code is also quite performant, and its design pattern permits a flexible and extensible approach to numerically solving various discharge problems, even when these require many thousands of CPU cores.

Originally, the code was written for studying pre-breakdown phenomena in high-voltage equipment (Marskar, 2019), but over time it has been adapted to fit a wider category of discharge problems. Some examples science examples are given in Figure 1 (HV technology), Figure 2 (nanosecond pulsed discharges), and Figure 3 (upper atmospheric lightning).

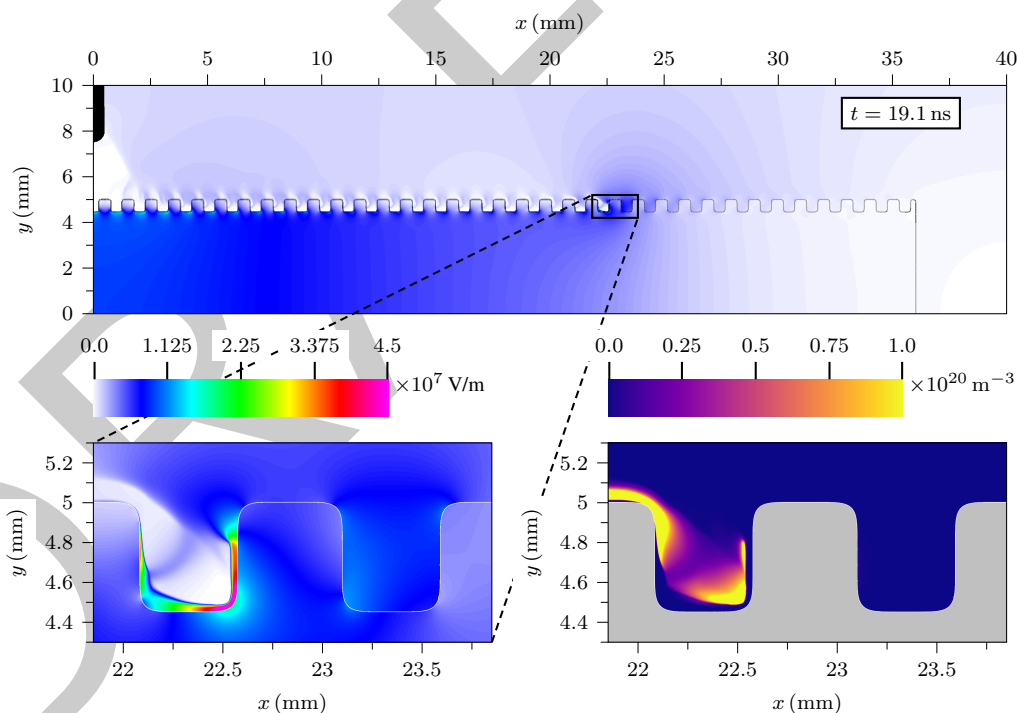


Figure 1: 2D Surface discharges over complex surfaces (Meyer et al., 2022). Top: Electrode (shaded region) and dielectric (profiled surface). Bottom left panel: Electric field distribution. Bottom right panel: Plasma density.

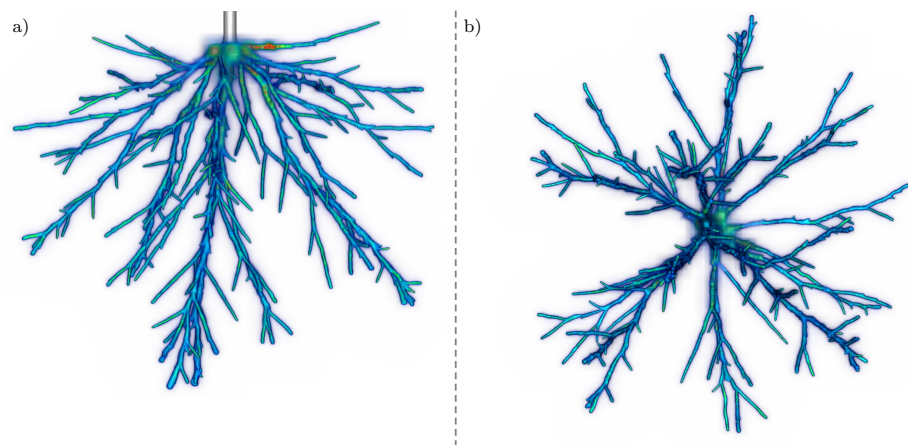


Figure 2: Streamer corona in a nanosecond pulsed discharge in atmospheric air. The two figures show the same discharge, viewed from the side and from the bottom.

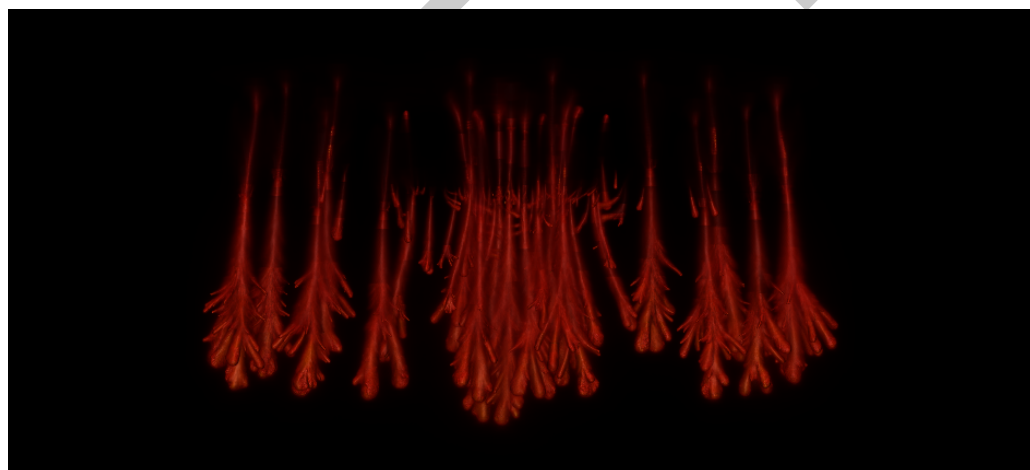


Figure 3: Sprite discharges in the upper atmosphere.

Acknowledgements

The development of chombo-discharge was partially achieved through funding by the Research Council of Norway through grants 245422, 319930/E20, and 321449.

References

- Marskar, R. (2019). An adaptive cartesian embedded boundary approach for fluid simulations of two- and three-dimensional low temperature plasma filaments in complex geometries. *Journal of Computational Physics*. <https://doi.org/10.1016/j.jcp.2019.03.036>
- Meyer, H. K. H., Marskar, R., & Mauseth, F. (2022). Evolution of positive streamers in air over non-planar dielectrics: Experiments and simulations. *Plasma Sources Science and Technology*, 31, 114006. <https://doi.org/10.1088/1361-6595/aca0be>
- Teunissen, J., & Ebert, E. (2017). Simulating streamer discharges in 3D with the parallel adaptive afivo framework. *Journal of Physics D: Applied Physics*, 50, 474001. <https://doi.org/10.1088/1361-6463/aa8faf>