A BOUNDARY ELEMENT AND FAST MULTIPOLE METHOD FOR ELECTRON CLOUD FIELD COMPUTATION

S. Feng[1], Z. Li, W. Liu[1], Y. Zhang[1], Y. Liu, T. Xin*,
Institute of High Energy Physics, Chinese Academy of Sciences,
Accelerator Division, Beijing, China
[1]also at University of Chinese Academy of Sciences,
School of Physical Sciences, Beijing, China

Abstract

Electron cloud build-up simulations rely on accurate self-consistent electric field computation to correctly model the secondary emission cascade. The PyECLOUD code solves this via a Particle-in-Cell (PIC) approach using the Shortley-Weller finite-difference (SW-FD) Poisson solver. This work presents a Boundary Element Method (BEM) formulation for the electrostatic space-charge field that discretizes only the chamber wall into panels and evaluates particle forces via direct Coulomb summation, entirely avoiding volumetric grids. The BEM solver is validated against the analytic image solution for a circular chamber (error $< 0.2\,\%$) and cross-validated with the existing SW-FD solver on the LHC Arc-Dipole chamber, showing sub-percent agreement over the chamber interior. The BEM module is integrated into the PyECLOUD simulation pipeline as a plug-in field solver. Build-up simulations comparing BEM ($N = 50$ and $N = 200$ panels) with the baseline PIC solver (0.3 mm grid) produce consistent electron cloud line densities, confirming that the BEM formulation correctly captures the physics of the original solver. The BEM panels simultaneously provide a unified geometry for impact detection and secondary emission. The Fast Multipole Method is introduced to accelerate the intrinsic $\mathcal{O}(N^2)$ particle-particle Coulomb sum, and GPU acceleration is proposed as a path toward a standalone BEM-FMM code scalable to $> 10^5$ particles.

INTRODUCTION

Electron clouds — accumulations of low-energy electrons within accelerator vacuum chambers — can cause transverse instabilities, emittance growth, and beam losses in high-intensity hadron and lepton storage rings [1]. The PyECLOUD code [2] is a 2D macro-particle simulation tool widely used at CERN to model electron cloud build-up. At each timestep, PyECLOUD computes the self-consistent electric field via a Particle-in-Cell (PIC) approach: macro-particle charges are scattered onto a structured grid, Poisson's equation is solved using the Shortley-Weller finite-difference method (SW-FD) [3], and the field is interpolated back to particle positions.

This work presents a Boundary Element Method (BEM) [4] for electrostatic field computation that replaces the grid-based Poisson solver. BEM discretizes only the chamber boundary, evaluating fields via direct Coulomb summation without volumetric grids. We validate the BEM solver against analytic solutions for a circular chamber and cross-check it with the existing SW-FD solver for the realistic LHC Arc-Dipole chamber geometry. The BEM module is then integrated into the PyECLOUD pipeline as a selectable field solver. Build-up simulations using BEM-PyECLOUD produce cloud density results consistent with the PIC-based PyECLOUD, confirming that the BEM formulation correctly captures the physics of the original solver.

The paper is organized as follows. Section describes the BEM formulation and its integration into PyECLOUD. Section presents validation against analytic benchmarks and cross-validation with the SW-FD solver. Section discusses the integration of the BEM module into the PyECLOUD pipeline. Section concludes and outlines future work.

* xintm@ihep.ac.cn

METHOD

Boundary Element Method for 2D Electrostatics

In a 2D chamber cross-section with grounded conducting walls ($\phi = 0$), the BEM reformulates Poisson's equation as an integral equation for the induced surface charge density $\sigma(\mathbf{x})$ on the wall [4]:

$$\oint_{\partial\Omega} G(|\mathbf{x}-\mathbf{x}'|)\,\sigma(\mathbf{x}')\,ds' = -\int_{\Omega} G(|\mathbf{x}-\mathbf{x}'|)\,\frac{\rho(\mathbf{x}')}{\varepsilon_0}\,d^2x' \quad (\mathbf{x}\in\partial\Omega), \quad (1)$$

where $G(r) = -\log(r)/(2\pi)$ is the 2D free-space Green's function. Discretizing the wall into N constant-element panels gives the linear system $\sum_{j=1}^{N} A_{ij}\,\sigma_j = b_i$, with $A_{ij} = \int_{\text{panel } j} G\,ds$ and $b_i = -\sum_k (q_k/\varepsilon_0)\,G(|\mathbf{x}_i - \mathbf{x}_k|)$. The diagonal $A_{ii} = L_i(1-\log(L_i/2))/(2\pi)$ handles the logarithmic singularity; off-diagonal elements use the midpoint approximation. $\mathbf{A}$ is assembled once, LU-factorized at initialization, and only $\mathbf{b}$ is recomputed per timestep: $\sigma = \mathbf{A}^{-1}\mathbf{b}$. A charge-conservation correction [5] removes the null-space ambiguity of the 2D single-layer potential. The electric field is then

$$\mathbf{E}(\mathbf{x}) = \sum_k \frac{q_k}{2\pi\varepsilon_0}\frac{\mathbf{x}-\mathbf{x}_k}{|\mathbf{x}-\mathbf{x}_k|^2} + \sum_j \frac{\sigma_j L_j}{2\pi}\frac{\mathbf{x}-\mathbf{x}_j}{|\mathbf{x}-\mathbf{x}_j|^2}, \quad (2)$$

evaluated analytically from known charges, giving a continuous field throughout the interior.

Fast Multipole Method for Particle-Particle Forces

Eq. (2) involves Coulomb interactions between every particle pair—an $\mathcal{O}(N^2)$ operation. The Fast Multipole Method (FMM) [6] reduces this to $\mathcal{O}((N_s+N_t)\log(N_s+N_t))$ by grouping distant sources into a quad-tree and using multipole expansions (Fig. 1). We use `pyfmmlib` [7], a Python wrapper for FMMLIB2, with the 2D Laplace kernel $G(r) = -\log(r)/(2\pi)$.

For future coupled beam-cloud simulations, the beam's self-field ($\propto 1/\gamma^2 \to 0$) must be excluded. A dual-evaluation $\mathbf{E}_{\text{cloud}} \propto \text{FMM}(q_{\text{beam}} + q_e + q_\sigma) - \text{FMM}(q_{\text{beam}})$ removes the beam self-field without modifying the FMM kernel.

A practical advantage of BEM is that the panel set also serves as the geometry for impact detection and secondary emission. Panel normals directly give impact angles for the SEY model, eliminating redundant geometry representations.

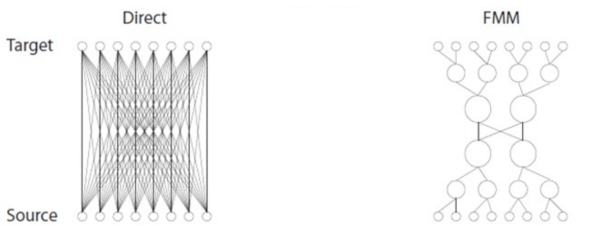

Figure 1: Schematic comparison of the all-to-all $\mathcal{O}(N^2)$ interaction model (left) versus the tree-based FMM (right). In FMM, distant sources are grouped and their far field is represented through multipole expansions [6].

VALIDATION

We validate the BEM field solver through two complementary approaches: (i) comparison against the analytic image solution for a circular chamber; and (ii) cross-validation with the SW-FD PIC solver for the LHC Arc-Dipole chamber geometry.

Circular Chamber: Analytic Benchmark

For a grounded circular chamber of radius $R = 10$ mm with a point charge q at position $(-8\text{ mm}, 0)$, the electric field on the x-axis can be computed analytically via the method of images:

$$E_x(x) = \frac{q}{2\pi\varepsilon_0}\left[\frac{1}{x + x_s} + \frac{x_s}{R^2 + x_s x}\right], \tag{3}$$

where $x_s = 8$ mm is the source distance from the origin.

Figure 2 compares the BEM-computed field with the analytic solution. Over the full evaluation range, the maximum BEM error is below 0.2 %, confirming the accuracy of the BEM formulation.

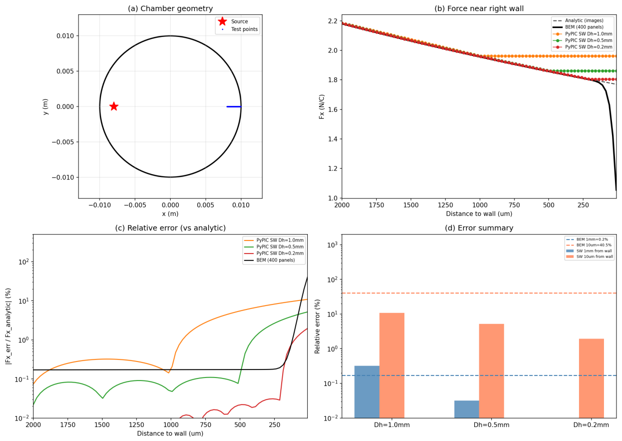

Figure 2: (a) Geometry of the circular chamber benchmark. (b) Electric field E_x along the x-axis, comparing BEM (400 panels) with the analytic solution. (c) Relative error vs. distance from wall. (d) Summary of maximum and average errors.

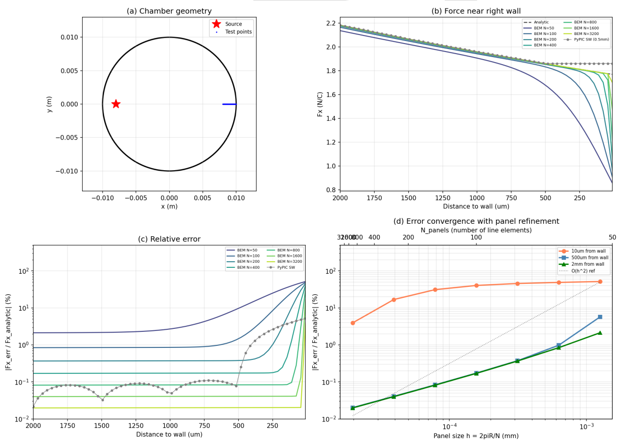

Figure 3: LHC Arc-Dipole chamber: BEM (250 panels) vs. SW-FD at three grid resolutions (1.0, 0.5, 0.2 mm). Source is a Gaussian beam profile ($\sigma = 1$ mm) at three positions (center, $(7\text{ mm}, 0)$, $(8\text{ mm}, 8\text{ mm})$).

LHC Arc-Dipole Chamber Cross-Validation

The LHC Arc-Dipole chamber is described by a 50-vertex polygon (Fig. 3). For evaluation points in the chamber interior, BEM and SW-FD agree to within 1 % at the finest grid resolution (0.2 mm), confirming that both solvers correctly compute the interior field.

Wall Charge Contribution

To quantify the importance of correctly modeling the wall charges, Fig. 4 compares the total field (beam + BEM-computed wall σ) with the free-space field (beam only, as would be computed without boundary conditions). Near the chamber wall, the induced surface charges contribute 20–33 % of the total field. This demonstrates that wall charge effects are significant for electrons near the boundary and cannot be neglected in accurate electron cloud simulations.

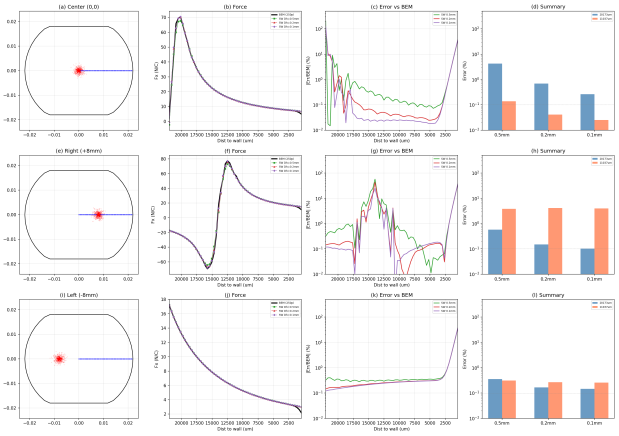

Figure 4: Comparison of the total electric field (beam + wall induced charges) with the free-space field (beam only) for the LHC chamber. Near the wall, induced charges contribute 20–33 % of the total field magnitude, demonstrating that wall charge effects cannot be neglected for accurate field computation near boundaries.

INTEGRATION INTO PYECLOUD

To verify the BEM solver in context, we integrated it into the PyECLOUD build-up pipeline. The BEM module exposes the same `scatter` / `solve` / `gather` interface as the PIC module, enabling direct comparison under identical beam, magnetic-field, and SEY conditions.

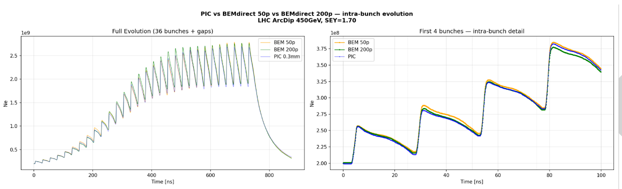

Figure 5: Electron cloud line density from PyECLOUD build-up simulations using three field solvers: baseline SW-FD PIC (0.3 mm grid), BEM with $N = 50$ panels, and BEM with $N = 200$ panels. All use identical LHC Arc-Dipole parameters at 450 GeV, $\delta_{\max} = 1.7$.

Figure 5 compares the electron cloud density evolution from the three solvers. All three produce consistent densities in both magnitude and temporal structure, confirming that the BEM formulation reproduces the PIC solver's physics. The per-timestep cost is comparable to PIC for the macro-particle counts used here ($N_p \sim 10^4$).

CONCLUSION

We have presented a BEM formulation for the electrostatic field in electron cloud simulations. Validation against the analytic circular chamber (error < 0.2 %) and cross-check with SW-FD PIC on the LHC Arc-Dipole (interior agreement < 1 %) confirm the accuracy. The BEM module integrates into PyECLOUD, produces cloud densities consistent with PIC, and unifies field computation with impact detection and SEY through a single panel geometry.

Future work includes deploying FMM for particle-particle forces, GPU acceleration of the Coulomb sums, and extending the method to 3D complex geometries and dielectric walls.

REFERENCES

[1] F. Zimmermann, "Electron-cloud effects in future accelerators," in *Proc. ECLOUD'04*, Napa, CA, USA, Apr. 2004.

[2] G. Iadarola and G. Rumolo, "Electron Cloud Simulations with PyECLOUD", in *Proc. ICAP'12*, Rostock-Warnemunde, Germany, Aug. 2012, paper WESAI4, pp. 138–142.

[3] G. Shortley and R. Weller, "The numerical solution of Laplace's equation," *J. Appl. Phys.*, vol. 9, p. 334, 1938. doi:10.1063/1.1712309

[4] C. Brebbia and J. Dominguez, *Boundary Elements: An Introductory Course*, McGraw-Hill, 1992.

[5] A. Hearn, "The calculation of electric fields in accelerator tubes," *Nucl. Instrum. Methods*, vol. 128, p. 85, 1975. doi:10.1016/0029-554X(75)90126-3

[6] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comput. Phys.*, vol. 73, p. 325, 1987. doi:10.1016/0021-9991(87)90140-9

[7] pyfmmlib: Python wrappers for FMM libraries, https://github.com/blackwer/pyfmmlib