

pyMoM3d

An Open-Source 3D Method of Moments Solver for Electromagnetic Analysis in Free Space and Layered Media

Austin Rothschild and Owen Ryan

Stanford University, Department of Electrical Engineering

March 19, 2026

Abstract

pyMoM3d is an open-source Python library implementing a surface integral equation solver for three-dimensional electromagnetic scattering and antenna radiation problems. The solver discretizes the Electric Field Integral Equation (EFIE) using Rao–Wilton–Glisson (RWG) basis functions on triangular surface meshes generated by **Gmsh**. The full mesh-to-basis-function pipeline is exposed in Python. The impedance matrix is assembled with Dunavant symmetric Gauss quadrature on triangles and analytic singularity extraction (Wilton–Graglia decomposition) for self- and near-patch interactions. Direct LU factorization or preconditioned GMRES solves the resulting system. Rudimentary multilayer problems are also supported using an interface to **Strata** – a C++ based open-source tool for computing multilayered dyadic Green’s functions using the Discrete Complex Image Method (DCIM). This report documents the theoretical and implementation details of each core numerical component. Multiple examples and solver benchmarks are highlighted. Detailed code can be found on the **pyMoM3d GitHub Repository**.

Contents

1	Introduction	3
2	The Method of Moments	4
2.1	MoM Formulation	4
2.1.1	Problem Formulation	4
2.2	Electric Field Integral Equation (EFIE)	4
2.3	Galerkin Discretization and The MoM Matrix System	5
2.4	RWG Basis Functions	6
2.4.1	Definition and Mathematical Properties	6
2.5	Singular and Near-Singular Integrals	7
2.6	The $\mathbf{Z}$ Matrix	8
2.6.1	Formulating the $\mathbf{Z}$ Matrix	8
2.6.2	Gaussian Quadrature on Triangles	10
3	Multilayer Problems	12
3.1	Green's Function in Layered Media	12
3.2	The Discrete Complex Image Method	14
4	pyMoM3d Implementation	14
4.1	Code Architecture	14
4.2	Surface Meshing and Basis Assignment	16
4.3	Excitation and Port Modeling	18
4.4	Post-Processing	18
4.4.1	Surface Currents	18
4.4.2	Far-Fields	19
4.4.3	Radar Cross Sections	19
5	Validation and Examples	20
5.1	Spherical Plate Scattering	20
5.2	Dipole Antenna	20
5.3	On-Chip Inductor	21
5.4	PEC Solver Performance	22
6	Future Work	24
A	Use of AI Tools	25
B	Details on Source Excitations	27
B.1	Plane-Wave Sources	27
B.2	Delta-Gap Sources	27
C	Multi-Antenna Modeling	29
C.1	Friis Transmission Validation	29
D	Final Project Timeline	31

1 Introduction

The accurate and efficient simulation of three-dimensional electromagnetic structures is a foundational problem in computational electromagnetics, with applications spanning antenna design, scattering analysis [1], and on-chip passive component modeling. As operating frequencies increase and geometries become electrically large or highly complex, full-wave solutions of Maxwell’s equations are required to capture distributed electromagnetic effects that are not adequately modeled by lumped or quasi-static approximations.

Among numerical techniques for full-wave electromagnetic analysis, the Method of Moments (MoM) [2] occupies a central role for problems involving conducting surfaces. By formulating Maxwell’s equations as boundary integral equations, MoM reduces the dimensionality of the problem by one and inherently satisfies radiation conditions at infinity. This makes MoM particularly well suited for open-region problems such as antennas, scatterers, and planar conductors, where volumetric discretization methods can become computationally expensive.

In this report, a three-dimensional frequency-domain MoM solver—`pyMoM3d` [3]—is developed for the analysis of arbitrarily shaped perfectly electrically conducting (PEC) structures. The solver employs the Electric Field Integral Equation (EFIE) in conjunction with Rao–Wilton–Glisson (RWG) basis functions defined on triangular surface meshes. Surface meshing is carried out using `Gmsh` [4], a powerful open-source 3D mesh generator.

The core numerical challenge in MoM lies in the accurate evaluation of the dense impedance matrix arising from the discretized integral equations. This matrix contains singular and near-singular integrals associated with the Green’s function kernel, which must be treated carefully to ensure numerical stability and convergence. In this implementation, Gaussian quadrature techniques are used to evaluate the required surface integrals, with special consideration given to self-terms and closely spaced basis function interaction.

Beyond theoretical formulation, this report emphasizes implementation details that are often omitted in analytical treatments but are critical in practice. These include geometry import and surface meshing, basis function indexing, matrix assembly strategies, and solution of the resulting dense linear system. The solver is implemented in Python with a modular architecture designed to facilitate experimentation, extension, and validation against known reference problems.

To verify correctness and assess performance, the solver is validated using a series of canonical electromagnetic structures, such as PEC spheres and planar dipole antennas. The framework is further applied to the full-wave simulation of an on-chip spiral inductor, demonstrating the relevance of surface MoM techniques to modern integrated circuit passive component analysis. These examples highlight both the strengths and limitations of surface-based MoM approaches when applied to electrically small and moderately complex structures.

The remainder of this report is organized as follows. Section 2 is the most comprehensive and presents the theoretical foundations of the Method of Moments and formulation used in this work, along with relevant implementation details. Section 3 describes multilayer modeling and the `Strata` interface pipeline. Section 4 describes the architecture and implementation of the `pyMoM3d` solver. Section 5 presents validation results and example simulations, and Section 6 discusses directions for future work. The appendix expands on some details omitted from the main report.

2 The Method of Moments

2.1 MoM Formulation

The Method of Moments (MoM) is a boundary-element technique for solving electromagnetic integral equations on the surfaces of conducting objects. The MoM reduces a volumetric PDE governed by Maxwell's equations to a problem involving two-dimensional surface integrals, allowing for efficient and accurate solutions to problems with homogeneous exteriors where an understanding of scattering or radiation is desired. MoM finds widespread usage in problems naturally formulated as surface integral equations. These arise frequently in the field of antennas, passive RF structure design, radar, and electromagnetic device modeling in open-regions.

2.1.1 Problem Formulation

Consider a closed PEC surface Ω immersed in free space and illuminated by an incident field $\mathbf{E}_i(\mathbf{r})$. By the surface equivalence principle, the original scattering problem — a dielectric body with permittivity $\epsilon(\mathbf{r})$ inside volume Ω illuminated by $\mathbf{E}_i$ — is exactly equivalent to a distribution of surface currents $\mathbf{J}(\mathbf{r})$ on the boundary $\partial\Omega$ radiating into unbounded free space [1]. The exterior scattered field $\mathbf{E}_s$ is identical in both formulations, allowing the volumetric problem to be reduced to a surface integral equation for $\mathbf{J}(\mathbf{r})$.

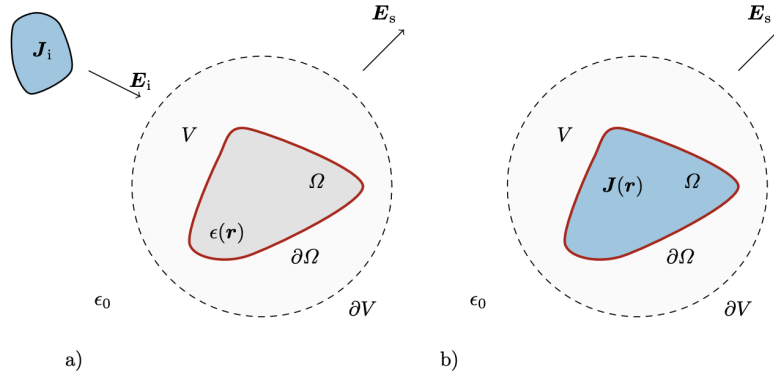


Figure 1: The surface equivalence principle: the original scattering problem (left), in which a plane wave $\mathbf{E}_i$ illuminates a dielectric body with permittivity $\epsilon(\mathbf{r})$ inside volume Ω , is exactly equivalent to an impressed surface current distribution $\mathbf{J}(\mathbf{r})$ on the boundary $\partial\Omega$ radiating into free-space (right) [1].

Given a surface current distribution $\mathbf{J}(\mathbf{r}')$ on $\partial\Omega$, the scattered field is related to $\mathbf{J}(\mathbf{r}')$ via the dyadic Green's function $\mathbf{G}(\mathbf{r}, \mathbf{r}')$ through the radiation integral:

$$\mathbf{E}_s(\mathbf{r}) = \int_{\partial\Omega} \mathbf{G}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}') dS'. \quad (1)$$

The Green's function $\mathbf{G}(\mathbf{r}, \mathbf{r}')$ is known; $\mathbf{J}(\mathbf{r}')$ is the unknown to be determined.

2.2 Electric Field Integral Equation (EFIE)

Expressing the total field as the sum of incident and scattered contributions,

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}_i(\mathbf{r}) + \mathbf{E}_s(\mathbf{r}), \quad (2)$$

and enforcing the PEC boundary condition (the tangential electric field vanishes on $\partial\Omega$),

$$\hat{\mathbf{n}} \times \mathbf{E}(\mathbf{r}) = \mathbf{0}, \quad \mathbf{r} \in \partial\Omega, \quad (3)$$

gives

$$\hat{\mathbf{n}} \times \mathbf{E}_i(\mathbf{r}) = -\hat{\mathbf{n}} \times \mathbf{E}_s(\mathbf{r}), \quad \mathbf{r} \in \partial\Omega. \quad (4)$$

Substituting the radiation integral (1) and using the free-space scalar Green's function

$$G_0(\mathbf{r}, \mathbf{r}') = \frac{e^{-jk_0 R}}{4\pi R}, \quad R = |\mathbf{r} - \mathbf{r}'|, \quad (5)$$

yields the vector/scalar potential representation:

$$\mathbf{E}_s(\mathbf{r}) = -j\omega\mu_0 \int_{\partial\Omega} G_0(\mathbf{r}, \mathbf{r}') \mathbf{J}(\mathbf{r}') dS' + \frac{1}{k_0^2} \nabla \int_{\partial\Omega} G_0(\mathbf{r}, \mathbf{r}') \nabla' \cdot \mathbf{J}(\mathbf{r}') dS'. \quad (6)$$

Substituting into the boundary condition and defining the linear operator $\mathcal{L}\{\cdot\}$ gives the well-known operator form of the EFIE:

$$\hat{\mathbf{n}} \times \mathbf{E}_i(\mathbf{r}) = -\hat{\mathbf{n}} \times \mathcal{L}\{\mathbf{J}(\mathbf{r}')\}, \quad \mathbf{r} \in \partial\Omega. \quad (7)$$

2.3 Galerkin Discretization and The MoM Matrix System

The unknown current density is expanded in a finite set of N basis functions $\{\mathbf{f}_n\}$:

$$\mathbf{J}(\mathbf{r}') \approx \sum_{n=1}^N I_n \mathbf{f}_n(\mathbf{r}'), \quad (8)$$

where $\mathbf{f}_n(\mathbf{r}')$ are known and I_n are unknown coefficients. Applying Galerkin testing with the same basis functions $\{\mathbf{f}_m\}$,

$$\langle \mathbf{f}_m, \mathbf{E}_i \rangle = \int_{\partial\Omega} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}_i(\mathbf{r}) dS, \quad (9)$$

and substituting (8) yields:

$$\int_{\partial\Omega} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}_i(\mathbf{r}) dS = \sum_{n=1}^N I_n Z_{mn}. \quad (10)$$

The excitation vector and impedance matrix entries are:

$$V_m = \int_{\partial\Omega} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}_i(\mathbf{r}) dS, \quad (11)$$

$$\begin{aligned} Z_{mn} = & jk_0 Z_0 \int_{\partial\Omega} \int_{\partial\Omega} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}') G_0(\mathbf{r}, \mathbf{r}') dS' dS \\ & - \frac{Z_0}{k_0} \int_{\partial\Omega} \int_{\partial\Omega} (\nabla \cdot \mathbf{f}_m(\mathbf{r})) (\nabla' \cdot \mathbf{f}_n(\mathbf{r}')) G_0(\mathbf{r}, \mathbf{r}') dS' dS. \end{aligned} \quad (12)$$

Collecting all equations (i.e. Galerkin testing over N basis functions) yields the MoM system:

$$\boxed{\mathbf{V} = \mathbf{Z} \mathbf{I}}. \quad (13)$$

Here $\mathbf{V} \in C^N$ is the known excitation vector, $\mathbf{Z} \in C^{N \times N}$ is the dense impedance matrix, and $\mathbf{I} \in C^N$ contains the unknown current coefficients. Computation of $\mathbf{Z}$ incurs a large numerical overhead as the number of basis functions increases, since the formation of $\mathbf{Z}$ scales as $\mathcal{O}(N^2 Q^2)$.

Solving (13) (via LU or GMRES) for $\mathbf{I}$ completes the MoM solution. For small to moderate problem sizes, a direct LU factorization (via NumPy) is used, while for larger systems an iterative solver such as GMRES (via SciPy) is employed to reduce memory and computational cost. With $\mathbf{I}$ known, all quantities of interest can be computed by modal relations and far-field expansions.

2.4 RWG Basis Functions

2.4.1 Definition and Mathematical Properties

Rao, Wilton, and Glisson [5] introduced a set of vector basis functions defined on pairs of adjacent triangles sharing an interior edge. Each basis function $\mathbf{f}_n$ is associated with an interior edge ℓ_n of length l_n . The edge is shared by two triangles, designated T_n^+ and T_n^- . Each triangle has a free vertex—the vertex not lying on the shared edge—denoted $\mathbf{r}_n^{+, \text{free}}$ and $\mathbf{r}_n^{-, \text{free}}$ respectively.

The RWG function is defined piecewise:

$$\mathbf{f}_n(\mathbf{r}) = \begin{cases} \frac{l_n}{2A_n^+} (\mathbf{r} - \mathbf{r}_n^{+, \text{free}}) & \mathbf{r} \in T_n^+, \\ \frac{l_n}{2A_n^-} (\mathbf{r}_n^{-, \text{free}} - \mathbf{r}) & \mathbf{r} \in T_n^-, \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

where A_n^+ and A_n^- are the areas of T_n^+ and T_n^- .

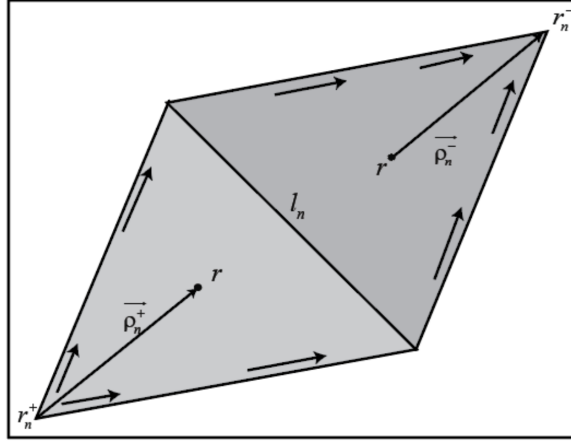


Figure 2: An RWG basis function [6].

Physical Interpretation

The vector $\boldsymbol{\rho}_n^+ = \mathbf{r} - \mathbf{r}_n^{+, \text{free}}$ points radially outward from the free vertex of T_n^+ . On T_n^+ the function therefore drives current *away* from the free vertex and across the shared edge. On T_n^- , the vector $\boldsymbol{\rho}_n^- = \mathbf{r}_n^{-, \text{free}} - \mathbf{r}$ drives current *toward* the free vertex of T_n^- . The resulting flow is continuous across the shared edge.

Surface Divergence

The surface divergence of $\mathbf{f}_n$ is piecewise constant:

$$\nabla_S \cdot \mathbf{f}_n(\mathbf{r}) = \begin{cases} \frac{l_n}{A_n^+} & \mathbf{r} \in T_n^+, \\ -\frac{l_n}{A_n^-} & \mathbf{r} \in T_n^-. \end{cases} \quad (15)$$

This is the property exploited in the scalar potential term Φ_{mn} : the surface divergence replaces the normal flux and converts a volume integral to a surface one via the divergence theorem.

Current Continuity

At the shared edge, the normal components of $\mathbf{f}_n$ from T_n^+ and T_n^- are equal and opposite, ensuring that no line charge accumulates. Explicitly, the flux of $\mathbf{f}_n$ across the shared edge is:

$$\int_{\ell_n} \mathbf{f}_n \cdot \hat{\ell}_\perp d\ell = \frac{l_n}{2A_n^+} \cdot A_n^+/l_n + \frac{l_n}{2A_n^-} \cdot A_n^-/l_n = 1, \quad (16)$$

a unit positive flux from T_n^+ to T_n^- . Boundary edges (those belonging to only one triangle) are excluded from the basis set entirely; the $\hat{n} \times \mathbf{J} = 0$ condition on boundary edges is enforced implicitly by the vector nature of the RWG basis.

Both the current continuity and surface divergence properties of RWG basis functions are two of their most important features and naturally lend to physically accurate surface current modeling. For all N interior edges the following arrays of length N are collected and stored in an `RWGBasis` dataclass:

Array	Type	Content
<code>edge_index</code>	<code>int32</code>	Index into <code>mesh.edges</code>
<code>edge_length</code>	<code>float64</code>	$l_n = \ \ell_n\ $
<code>t_plus</code>	<code>int32</code>	Triangle index of T_n^+
<code>t_minus</code>	<code>int32</code>	Triangle index of T_n^-
<code>free_vertex_plus</code>	<code>int32</code>	Vertex index of free vertex in T_n^+
<code>free_vertex_minus</code>	<code>int32</code>	Vertex index of free vertex in T_n^-
<code>area_plus</code>	<code>float64</code>	A_n^+
<code>area_minus</code>	<code>float64</code>	A_n^-

More details on surface meshing and basis function assignment are provided in Section 4.

2.5 Singular and Near-Singular Integrals

Considering the closed form Green's function in free space defined in (3), we observe that $G(\mathbf{r}, \mathbf{r}')$ tends to infinity as R becomes zero. For these self-terms, we must take a different approach to evaluating $G(\mathbf{r}, \mathbf{r}')$.

$$G(\mathbf{r}, \mathbf{r}') = \frac{e^{-jkR}}{4\pi R}, \quad R = |\mathbf{r} - \mathbf{r}'| \quad (17)$$

Following the methodology in [7], we begin by adding and subtracting the below analytic term from the general free space Green's function. Because the analytic term has a closed form general solution, we may evaluate it individually.

$$G(\mathbf{r}, \mathbf{r}') = \frac{1}{4\pi R} + \left[\frac{e^{-jkR}}{4\pi R} - \frac{1}{4\pi R} \right] \quad (18)$$

By the Taylor expansion of the free space Green's function, we show that our remainder term is smooth (i.e. the singularity is removed). Only the first term of the expansion contains a $1/R$ factor which is then immediately canceled by the complement of our analytic term, included in the integral. What remains is a smooth term that can be numerically integrated.

$$\frac{e^{-jkR}}{4\pi R} - \frac{1}{4\pi R} = \frac{1}{4\pi R} - \frac{jk}{4\pi} - \frac{k^2 R}{8\pi} + O(R^2) - \frac{1}{4\pi R} \quad (19)$$

$$\int_T G(\mathbf{r}, \mathbf{r}') dS' = \int_T \frac{1}{4\pi R} dS' + \int_T \left[\frac{e^{-jkR}}{4\pi R} - \frac{1}{4\pi R} \right] dS' \quad (20)$$

An expression for the analytic term is derived in [7] which is computed according to integration tables formulating terms in (7).

$$I_1 = \int_T \frac{1}{R} dT' = \sum_{i=1}^3 (t_i^0 f_{2i} - |w_0| \beta_i) \quad (21)$$

Where t_i^0 , f_{2i} , and β_i are geometric quantities derived from edge vectors and projected distances of the source triangle. t_i^0 is the signed projection of the observation point onto the i -th edge. f_{2i} represents the normal distance from the observation point to the edge, and β_i is the angle subtended. The closed form of (21) presents a numerical solution avoiding integration of the singularity entirely.

In **pyMoM3d**, impedance matrix elements are selected for singularity extraction by the observation point to nearest vertex distance, flooring against a threshold defined relative to the mean edge length of the source RWG basis.

2.6 The Z Matrix

2.6.1 Formulating the Z Matrix

Expanding **Z** explicitly, the (m, n) entry couples the m -th test function to the n -th source function via two bilinear forms:

$$Z_{mn} = jk_0\eta_0 A_{mn} - \frac{j\eta_0}{k_0} \Phi_{mn}, \quad (22)$$

where the *vector potential coupling*:

$$A_{mn} = \iint_{S_m \times S_n} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') dS' dS, \quad (23)$$

and the *scalar potential coupling*:

$$\Phi_{mn} = \iint_{S_m \times S_n} (\nabla_S \cdot \mathbf{f}_m(\mathbf{r})) (\nabla'_S \cdot \mathbf{f}_n(\mathbf{r}')) G(\mathbf{r}, \mathbf{r}') dS' dS. \quad (24)$$

These arise from the EFIE tested with $\mathbf{f}_m$ and the identity $\mathbf{E}_s = -j\omega\mu_0\mathbf{A} + (j/\omega\epsilon_0)\nabla\Phi$, where $\mathbf{A}$ and Φ are the magnetic vector potential and electric scalar potential generated by **J**.

Substituting the RWG divergence into Eq.24 expands the scalar potential integral into:

$$\Phi_{mn} = \sum_{\alpha \in \{+, -\}} \sum_{\beta \in \{+, -\}} \underbrace{\frac{\sigma_\alpha^m l_m}{A_m^\alpha}}_{\nabla \cdot \mathbf{f}_m \text{ on } T_m^\alpha} \cdot \underbrace{\frac{\sigma_\beta^n l_n}{A_n^\beta}}_{\nabla \cdot \mathbf{f}_n \text{ on } T_n^\beta} \cdot \iint_{T_m^\alpha \times T_n^\beta} G(\mathbf{r}, \mathbf{r}') dS' dS, \quad (25)$$

where $\sigma_+ = +1$, $\sigma_- = -1$.

Equivalently, substituting the RWG into A_{mn} :

$$A_{mn} = \sum_{\alpha \in \{+, -\}} \sum_{\beta \in \{+, -\}} \frac{\sigma_\alpha^m l_m}{2A_m^\alpha} \cdot \frac{\sigma_\beta^n l_n}{2A_n^\beta} \cdot \iint_{T_m^\alpha \times T_n^\beta} \boldsymbol{\rho}_m^\alpha(\mathbf{r}) \cdot \boldsymbol{\rho}_n^\beta(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') dS' dS. \quad (26)$$

We see that for each entry of impedance matrix Z_{mn} , there are four corresponding entries (in terms of the scalar and vector potentials) that must be computed.

$$Z_{mn} = Z_{mn}^{++} + Z_{mn}^{\pm} + Z_{mn}^{\mp} + Z_{mn}^{--} \quad (27)$$

Fig. 3 highlights all combinations of basis function interactions that must be considered per matrix element computation.

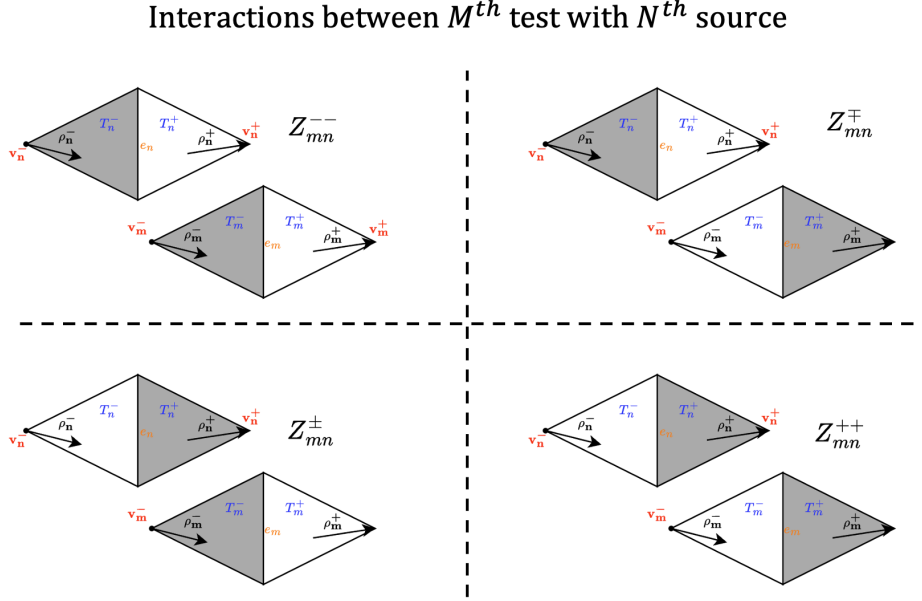


Figure 3: Combinations of quadrature to carry out for each computation of Z_{mn} involving all interactions between the m^{th} testing function with the n^{th} source function.

Both expressions in Eq. 25 and Eq. 26 share the same four triangle-pair integrals; they differ only in the prefactor and in whether the integrand contains $\boldsymbol{\rho}_m \cdot \boldsymbol{\rho}_n$ (vector potential) or just 1 (scalar potential). Numerically, this means each iteration of the inner 2×2 loop computes both I_A and I_Φ simultaneously at no extra quadrature cost. The details on quadrature and singularity extraction that are implemented for the Z_{mn} computation are discussed in the following section.

The Green's function is reciprocal: $G(\mathbf{r}, \mathbf{r}') = G(\mathbf{r}', \mathbf{r})$, and the dot product is commutative. Therefore both A_{mn} and Φ_{mn} are symmetric, and $Z_{mn} = Z_{nm}$. The structure of $\mathbf{Z}$ is highlighted below in Fig. 4.

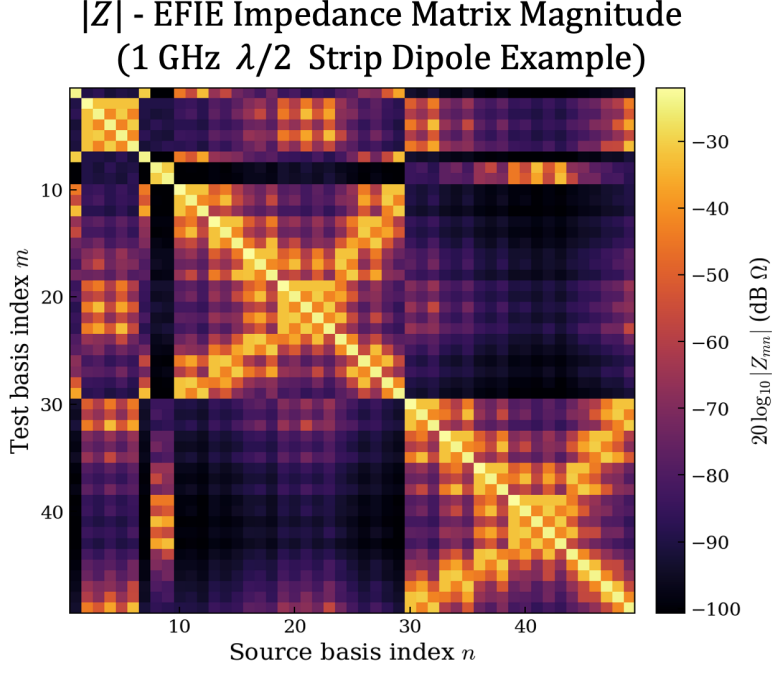


Figure 4: Structure of the EFIE $\mathbf{Z}$ matrix taken from the dipole antenna example covered in Section 5.

2.6.2 Gaussian Quadrature on Triangles

Gaussian quadrature is implemented on an element-by-element basis to populate the $\mathbf{Z}$ matrix. The Green's function is integrated across each source-observation RWG basis function according to (8) and (9) to compute the scalar and vector potentials respectively.

$$A_{mn} = jkZ_0 \int_{T_m} \int_{T_n} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}') G_0(\mathbf{r}, \mathbf{r}') dS' dS \quad (28)$$

$$\Phi_{mn} = \int_{T_m} \int_{T_n} \nabla \cdot \mathbf{f}_m(\mathbf{r}) \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G_0(\mathbf{r}, \mathbf{r}') dS' dS \quad (29)$$

These potentials are combined, yielding Z_{mn} in accordance with (10).

$$Z_{mn} = jkZ_0 \left(A_{mn} - \frac{1}{k^2} \Phi_{mn} \right) \quad (30)$$

To demonstrate integration of the Green's function over RWG basis functions, we first consider the simplified problem of integrating over a single RWG triangle. We begin by performing a transformation into Barycentric space and append the Jacobian. Using Barycentric coordinates rather than a cartesian grid allows us to develop an immediate area-based representation of internal points, preserve problem symmetry, and avoid integrating over points outside our basis functions. More importantly it allows us to cleanly implement Dunavant rule integration.

$$\int_T f(\mathbf{r}') dS' \approx \sum_{i=1}^N w_i \cdot f(\mathbf{r}'_i) \cdot 2A \quad (31)$$

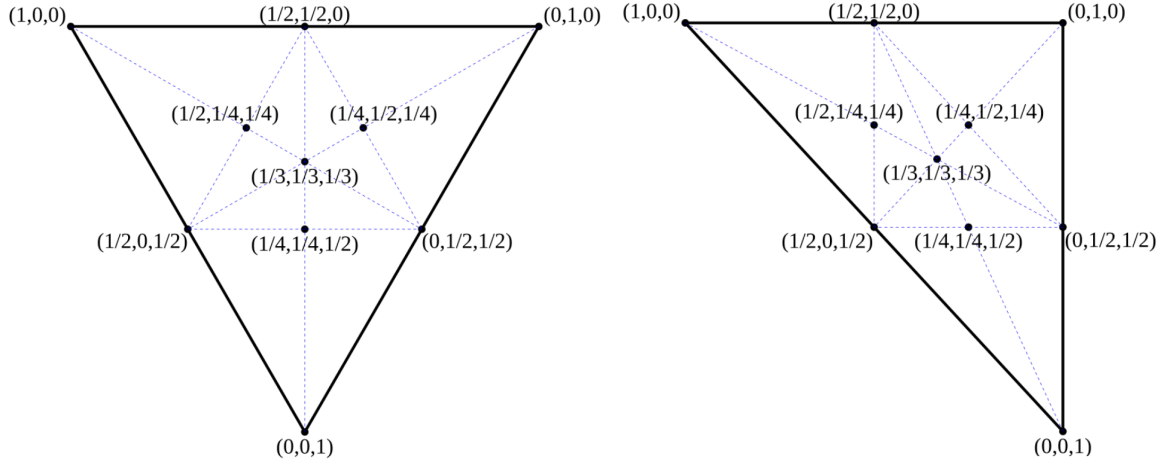


Figure 5: Internal coordinates on two different triangles represented in Barycentric coordinates

In a barycentric coordinate system, a point is represented by three separate bases essentially corresponding to the normalized area of a triangle composed up of two complementary vertices and the point represented. Each of these areas, λ_x and the vertex, $\mathbf{v}$ completes an area-based representation of coordinates on a RWG basis function [8].

$$\mathbf{r} = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3 \quad (32)$$

To discretize and numerically evaluate the integral we implement the approach proposed in [8]. For some order, $\mathbf{n}$, the Dunavant rule provides an approximation for integration over a single basis function by evaluating the integrand at $\mathbf{n}$ points in barycentric space. As the Dunavant rule order increases, the accuracy of the numerical integration does as well. These points for a variety of Dunavant orders are represented as lookup tables (LUTs) within `pyMOM3d`. In the literature as well as commercial solvers the $n = 4$ rule is predominantly implemented as it strikes a good balance between accuracy and efficiency. To symbolically demonstrate this approach, we first consider integration over just the first basis function alone.

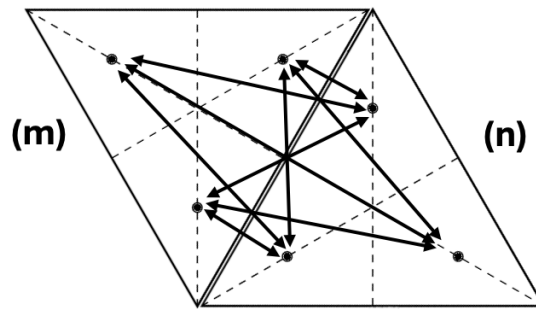


Figure 6: Source and observation pairs for Green's function sampling according to the third order Dunavant rule. Each arrow corresponds to a term in the Dunavant rule approximation to the closed form integral presented in (28) and (29)

For an arbitrary function $\mathbf{F}$ integrated over a single RWG basis function, Dunavant rule integration is demonstrated for order $n = 3$ in the following formulation:

$$\int_T F(\mathbf{r}') dS' \quad (33)$$

$$\approx 2A_n \sum_{i=1}^3 w_i F(\mathbf{r}'_i) = \frac{2A_n}{6} [F(\mathbf{r}'_1) + F(\mathbf{r}'_2) + F(\mathbf{r}'_3)] \quad (34)$$

Extending this approach to a third order integration of the Green's function sampled over the source and observation RWG basis functions m and n corresponding to a single Z_{mn} , we can obtain expressions for the scalar and vector potentials in (35) and (36) respectively [8].

$$\Phi_{mn} \approx \left(\pm \frac{\ell_m}{A_m} \right) \left(\pm \frac{\ell_n}{A_n} \right) \frac{2A_m}{6} \frac{2A_n}{6} \sum_{j=1}^3 \sum_{i=1}^3 G_0(\mathbf{r}_j, \mathbf{r}'_i) \quad (35)$$

$$A_{mn} \approx jkZ_0 \frac{2A_m}{6} \frac{2A_n}{6} \sum_{j=1}^3 \sum_{i=1}^3 [\mathbf{f}_m(\mathbf{r}_j) \cdot \mathbf{f}_n(\mathbf{r}'_i) G_0(\mathbf{r}_j, \mathbf{r}'_i)] \quad (36)$$

Summing contributions of A_{mn} and Φ_{mn} we may obtain the impedance element, Z_{mn}

$$Z_{mn} \approx \frac{2A_m}{6} \frac{2A_n}{6} \left[jkZ_0 \sum_{j=1}^3 \sum_{i=1}^3 \mathbf{f}_m(\mathbf{r}_j) \cdot \mathbf{f}_n(\mathbf{r}'_i) G_0(\mathbf{r}_j, \mathbf{r}'_i) - \frac{1}{jkZ_0^{-1}} \left(\pm \frac{\ell_m}{A_m} \right) \left(\pm \frac{\ell_n}{A_n} \right) \sum_{j=1}^3 \sum_{i=1}^3 G_0(\mathbf{r}_j, \mathbf{r}'_i) \right] \quad (37)$$

3 Multilayer Problems

Strata, an open source C++ library for computing Multilayer Green's functions is instantiated within **pyMOM3d** as a dependency to efficiently implement the discrete complex image method (DCIM). In the following section, we outline the basic methodology behind DCIM and the general evaluation of Multilayer Green's Functions.

3.1 Green's Function in Layered Media

In MoM implementations handling planarly layered media such as circuit boards (PCBs) or integrated circuits (ICs), the form of the Green's function must change to accomodate reflections from adjacent interfaces. This adaptation of the Green's function for multilayer problems represents the primary difference between a free space and layered media MoM solver.

While a handful of advanced commercial solvers mesh geometries on all layers of a stackup to accurately model multilayer PCBs and ICs, the problem may be simplified considerably if the layer stackup is taken to be uniform in the XY plane and layered in the Z direction, which is taken to be normal to XY, yielding translational invariance in XY. Multilayer problems in **pyMOM3D** and its multilayer solver **Strata** are evaluated strictly under this assumption of translational invariance [9].

The fundamental benefit of translational invariance in the XY plane is demonstrated formally by representing the multilayer Green's function (MGF) in a cylindrical coordinate system.

$$G(\mathbf{r}, \mathbf{r}') = G(x - x', y - y', z, z') = G(\rho, z, z') \quad (38)$$

For arbitrary $\mathbf{r}, \mathbf{r}'$ in cylindrical coordinates, the green's function in an XY translationally invariant space is independent of θ , reducing the dimensionality of the problem and enabling a representation of the Green's function in terms of only the source and observation Z-heights and their distance, ρ , by translating $\mathbf{r}$ to the origin [10; 11].

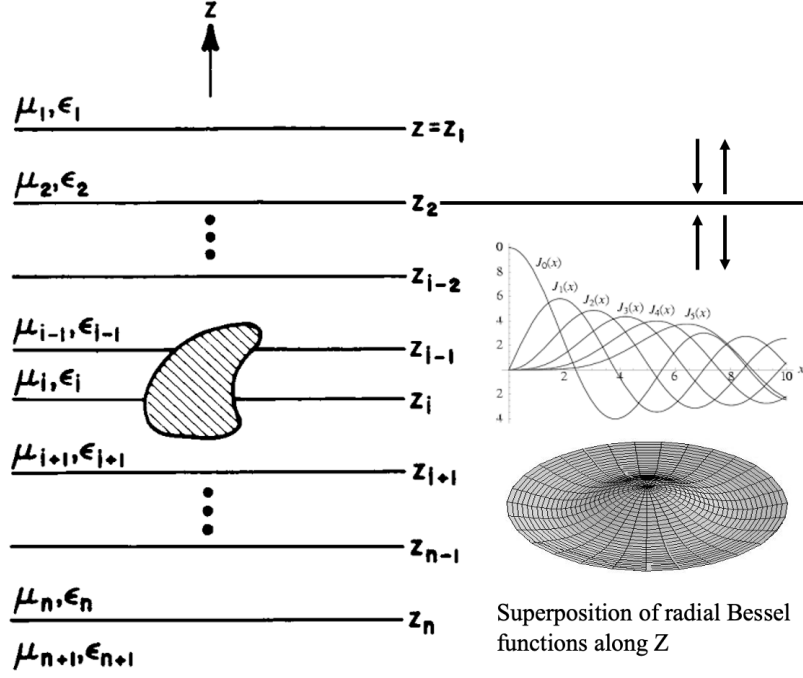


Figure 7: The Multilayer Green's Function (MGF) as a one dimensional scattering problem in the Z direction, with the total wave at each position represented by an infinite summation of Bessel functions of the first kind at each Z -position in the stackup.

We may further reduce the dimensionality by representing the behavior of the Green's function in the planar ρ space as a superposition of radial Bessel functions of the first kind. This is known as the Hankel transform, and represents the two dimensional problem as an infinite sum of one dimensional scattering problems in the Z direction. At any given height, the total J_0 is constructed from the integral of an infinite number of Bessel functions as formalized in the definition of the Hankel transform below.

$$\tilde{G}(k_\rho, z, z') = \int_0^\infty G(\rho, z, z') J_0(k_\rho \rho) \rho d\rho \quad (39)$$

Where J_0 is the superposition of an infinite number of first order Bessel functions.

$$J_0(k_\rho \rho) = \frac{1}{2\pi} \int_0^{2\pi} e^{jk_\rho \rho \cos(\phi - \phi')} d\phi \quad (40)$$

With our problem transformed as an infinite number of propagating waves in the Z -direction, we may express the spectral domain Green's function in the familiar form of a transmission line problem. We include a reflection term at each adjacent interface as well as a propagation term.

$$\tilde{G}(k_\rho, z, z') = \frac{1}{2jk_{z,i}} \left(e^{-jk_{z,i}|z-z'|} + \tilde{\Gamma}^+ e^{jk_{z,i}(z+z'-2z_i^-)} + \tilde{\Gamma}^- e^{-jk_{z,i}(z+z'-2z_i^+)} \right) \quad (41)$$

Where

$$k_{z,i} = \sqrt{k_i^2 - k_\rho^2}, \quad k_i = \omega \sqrt{\mu_i \epsilon_i} \quad (42)$$

The reflection coefficients, in a two-interface problem are calculated as follows, however must be computed separately for the TE and TM modes. Depending on the source and observation basis functions,

a superposition of TE and TM reflection coefficients may be used to compute the MGF, transforming the vector potential into a tensor quantity.

$$\Gamma_{i,i+1}^{TE} = \frac{k_{z,i}/\mu_i - k_{z,i+1}/\mu_{i+1}}{k_{z,i}/\mu_i + k_{z,i+1}/\mu_{i+1}} \quad (43)$$

$$\Gamma_{i,i+1}^{TM} = \frac{k_{z,i}/\epsilon_i - k_{z,i+1}/\epsilon_{i+1}}{k_{z,i}/\epsilon_i + k_{z,i+1}/\epsilon_{i+1}} \quad (44)$$

Additionally, for problems in many-layered media, reflection terms from all interfaces must be collapsed into the adjacent interface terms. This is typically handled via a recursive expression for Γ which is defined in its general form:

$$\tilde{\Gamma}_i^+ = \frac{\Gamma_{i,i+1} + \tilde{\Gamma}_{i+1}^+ e^{2jk_{z,i+1}(z_{i+1}-z_i)}}{1 + \Gamma_{i,i+1} \tilde{\Gamma}_{i+1}^+ e^{2jk_{z,i+1}(z_{i+1}-z_i)}} \quad (45)$$

Evaluation of these coefficients provides a complete expression for the XY-spectral MGF. To obtain the MGF in a usable form, the Hankel transform of $\tilde{G}(k_\rho, z, z')$ is inverted to yield what is known as the Sommerfeld integral.

$$G(\rho, z, z') = \frac{1}{2\pi} \int_0^\infty \tilde{G}(k_\rho, z, z') J_0(k_\rho \rho) k_\rho dk_\rho \quad (46)$$

The cartesian form of the MGF is then evaluated via direct numerical integration or approximation of the spectral MGF [11].

3.2 The Discrete Complex Image Method

While (46) provides an exact solution for the spatial MGF, direct numerical integration of the Sommerfeld Integral is computationally expensive. In general, the integrand is highly oscillatory requiring dense quadrature for accurate evaluation. Ultimately, this approach becomes impractical for large meshes and is generally avoided in favor of the discrete complex image method. Rather than directly evaluate the spatial domain integral, a sum of complex exponentials with constant k_z appearing at adjacent interfaces are fit to approximate the spectral MGF before inverting the Hankel transform.

$$\tilde{G}(k_\rho) \approx \sum_n f_n(k_\rho), e^{-jk_{z,n}d_n} \approx \sum_{n=1}^N a_n e^{-b_n k_z} \quad (47)$$

The Sommerfeld Identity can then be used to obtain an approximate analytic form of the MGF, considerably reducing compute overhead relative to direct integration.

$$\frac{e^{-jkr}}{r} = \int_0^\infty \frac{k_\rho}{jk_z} e^{-jk_z|z|} J_0(k_\rho \rho) dk_\rho \quad (48)$$

Typically, techniques like the matrix pencil method (MPM) are used to fit the sum of complex images to the true spectral MGF.

4 pyMoM3d Implementation

4.1 Code Architecture

A block diagram depicting the code architecture of `pyMoM3d` is depicted in Fig. 8. The solver pipeline has dependencies that move from left-to-right, top-to-bottom. To facilitate modularity, object oriented abstractions are used wherever possible.

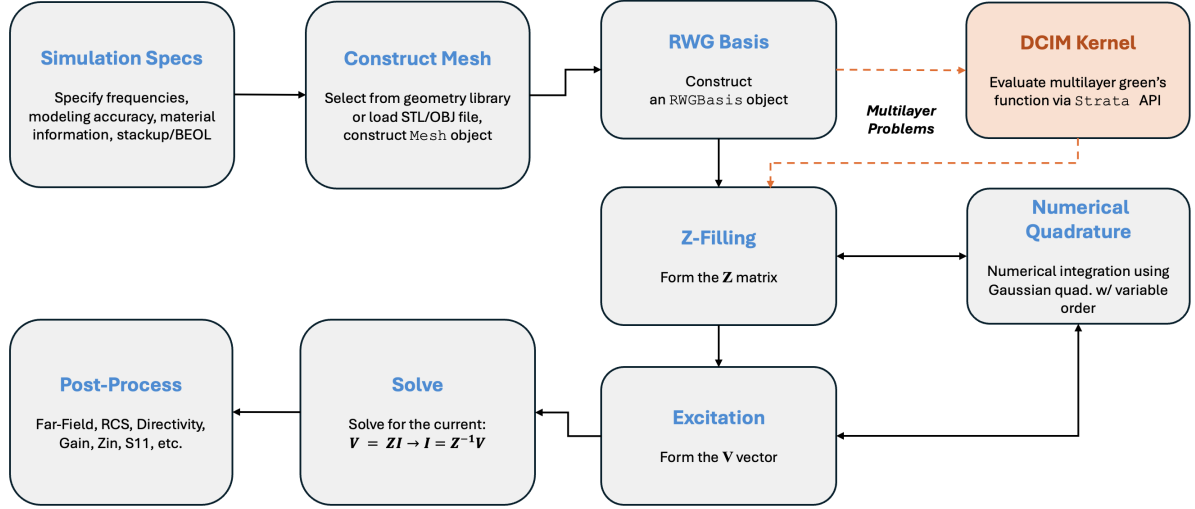


Figure 8: pyMoM3d code architecture.

Table 1: Core pyMoM3d modules.

Module	Key Class / Function	Responsibility
geometry/primitives.py	Sphere, Plate, ...	Parametric CAD geometry
mesh/gmsh_mesher.py	GmshMesher	Gmsh-based triangulation
mesh/mesh_data.py	Mesh	Vertices, triangles, edge topology
mesh/rwg_connectivity.py	compute_rwg_connectivity	RWG basis assembly
greens/quadrature.py	triangle_quad_rule	Dunavant quadrature rules
greens/singularity.py	integrate_green_singular	Singularity extraction
mom/impedance.py	fill_impedance_matrix	EFIE matrix assembly
mom/excitation.py	PlaneWaveExcitation, ...	Excitation vector $\mathbf{V}$
mom/solver.py	solve_direct, solve_gmres	Linear system solution
fields/far_field.py	compute_far_field	Far-field radiation integral
fields/rcs.py	compute_rcs	RCS in dBsm
visualization/	plot_mesh_3d, ...	Mesh and current plots
multilayer/strata_iface.py	strata_init	Configure Strata C API with layer geometry
multilayer/strata_iface.py	strata_compute_mgf_full	Evaluate G_ϕ and dyadic $\mathbf{G}^A$ via DCIM
greens/kernel.py	DCIMKernel	Kernel wrapper for layered media

4.2 Surface Meshing and Basis Assignment

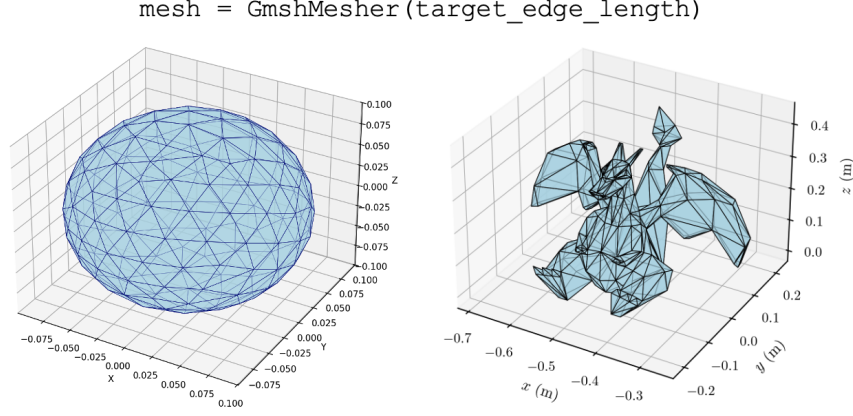


Figure 9: Visualization of surface meshes generated by `Gmsh`: canonical sphere (left) and arbitrary STL file (right).

The following arrays are populated directly by `GmshMesher` from `Gmsh` API outputs:

Attribute	Shape	dtype	Description
<code>vertices</code>	$(N_v, 3)$	<code>float64</code>	Vertex coordinates (x, y, z)
<code>triangles</code>	$(N_t, 3)$	<code>int32</code>	Vertex indices per triangle

The derived quantities after meshing are shown below:

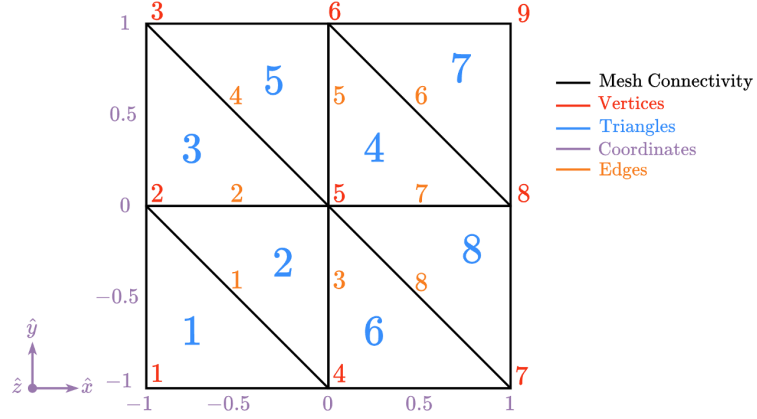
Attribute	Shape	dtype/type	Description
<code>edges</code>	$(N_e, 2)$	<code>int32</code>	Edge vertex indices
<code>triangle_normals</code>	$(N_t, 3)$	<code>float64</code>	Outward unit normals
<code>triangle_areas</code>	$(N_t,)$	<code>float64</code>	Triangle areas
<code>edge_lengths</code>	$(N_e,)$	<code>float64</code>	Edge lengths ℓ_n
<code>edge_to_triangles</code>	<code>dict</code>	—	Edge index $\rightarrow$ list of triangle indices

The algorithm for identifying RWG basis functions relies on counting shared vertices between pairs of triangles:

1. **3 shared vertices** $\Rightarrow$ same triangle (self-pair, skip).
2. **2 shared vertices** $\Rightarrow$ **shared edge** — defines one RWG basis function.
3. **1 shared vertex** $\Rightarrow$ common vertex only (no basis function).
4. **0 shared vertices** $\Rightarrow$ completely separated (no basis function).

Only condition 2 (a shared interior edge) generates an RWG basis function. Boundary edges (belonging to only one triangle) do not produce basis functions and represent open-circuit boundaries. These points are stored in the `edges` attribute of a mesh object.

An example meshed 2D plate [6; 12] highlighting the vertex and facet mapping returned by `Gmsh` is shown below.



N_v Array									
x	-1	-1	-1	0	0	0	1	1	1
y	-1	0	1	-1	0	1	-1	0	1
z	0	0	0	0	0	0	0	0	0
Point #	P1	P2	P3	P4	P5	P6	P7	P8	P9
N_t Array									
P1	1	4	2	5	3	4	6	5	-
P2	4	5	5	8	5	7	8	7	-
P3	2	2	3	6	6	5	9	8	-
Triangle #	1	2	3	4	5	6	7	8	-

Figure 10: 2D meshed plate showing N_v and N_t mapping to the surface mesh structure. Generalizes to 3D objects.

`Gmsh` stores triangle vertices in counter-clockwise (CCW) order when viewed from outside the surface. By the right-hand rule, CCW winding defines the outward unit normal $\hat{\mathbf{n}} = (\mathbf{v}_1 - \mathbf{v}_0) \times (\mathbf{v}_2 - \mathbf{v}_0)$.

A direct consequence is the opposite-traversal property: two triangles sharing an interior edge always traverse that edge in *opposite* directions. If T_a visits the shared edge as $v_0 \rightarrow v_1$ (CCW), then T_b visits it as $v_1 \rightarrow v_0$. `mesh.edges` stores each edge as the ordered pair (v_0, v_1) — this is the *canonical direction*. For an interior edge shared by triangles t_1 and t_2 , the code examines only t_1 :

1. Find the positions `idx_v0`, `idx_v1` of v_0 and v_1 within t_1 's vertex list
2. Check whether v_0 immediately precedes v_1 in cyclic order:

$$(\text{idx_v0} + 1) \bmod 3 = \text{idx_v1}.$$

3. **If true:** t_1 traverses the edge as $v_0 \rightarrow v_1$ (matching the canonical direction) $\Rightarrow t_1 = T^+$, $t_2 = T^-$.
4. **If false:** t_1 traverses the edge as $v_1 \rightarrow v_0$ (reversed) $\Rightarrow t_1 = T^-$, $t_2 = T^+$.

Assignment is arbitrary but must be consistent across the structure in order to generate a valid `RWGBasis` object to pass to subsequent stages in the MoM solver pipeline. The returned object from `compute_rwg_connectivity(mesh)` exposes:

Attribute	Shape / type	Description
<code>num_basis</code>	scalar	Total number of RWG basis functions N_{RWG}
<code>edge_index</code>	$(N_{\text{RWG}},)$	Index into <code>mesh.edges</code> for each basis
<code>edge_length</code>	$(N_{\text{RWG}},)$	ℓ_n for each basis
<code>t_plus</code>	$(N_{\text{RWG}},)$	Index of T_n^+ triangle
<code>t_minus</code>	$(N_{\text{RWG}},)$	Index of T_n^- triangle
<code>free_vertex_plus</code>	$(N_{\text{RWG}},)$	Vertex index of $\mathbf{r}_n^+$
<code>free_vertex_minus</code>	$(N_{\text{RWG}},)$	Vertex index of $\mathbf{r}_n^-$
<code>area_plus</code>	$(N_{\text{RWG}},)$	Area of T_n^+
<code>area_minus</code>	$(N_{\text{RWG}},)$	Area of T_n^-

Quadrature routines and visualization all act on these `RWGBasis` objects.

4.3 Excitation and Port Modeling

Once the impedance matrix $\mathbf{Z}$ is assembled, the only remaining ingredient of the linear system $\mathbf{Z}\mathbf{I} = \mathbf{V}$ is the excitation vector $\mathbf{V} \in C^N$. Its m -th entry is the Galerkin inner product of the m -th RWG test function with the incident or applied electric field:

$$V_m = \int_{\Omega} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}^{\text{src}}(\mathbf{r}) dS, \quad (49)$$

where $\mathbf{E}^{\text{src}}$ is the source field appropriate to the problem type (incident plane wave for scattering, or a localized applied field for antenna radiation). Substituting the piecewise RWG definition (equation (14)) into (49) splits the integral over the support of $\mathbf{f}_m$ into contributions from T_m^+ and T_m^- :

$$V_m = \frac{l_m}{2A_m^+} \int_{T_m^+} \boldsymbol{\rho}_m^+(\mathbf{r}) \cdot \mathbf{E}^{\text{src}}(\mathbf{r}) dS - \frac{l_m}{2A_m^-} \int_{T_m^-} \boldsymbol{\rho}_m^-(\mathbf{r}) \cdot \mathbf{E}^{\text{src}}(\mathbf{r}) dS, \quad (50)$$

where $\boldsymbol{\rho}_m^{\pm}(\mathbf{r}) = \mathbf{r} - \mathbf{r}_m^{\pm, \text{free}}$ is the vector from the free vertex to the field point. All excitation classes in `pyMoM3d` evaluate (50) numerically; only the definition of $\mathbf{E}^{\text{src}}$ differs between them. For details on plane-wave and delta-gap source excitations, refer to Appendix B.

4.4 Post-Processing

4.4.1 Surface Currents

Once the linear system $\mathbf{Z}\mathbf{I} = \mathbf{V}$ has been solved for the RWG coefficient vector $\mathbf{I} \in C^N$, the physical surface current density, radiated far field, and radar cross section are all derived quantities. Each I_n is complex (amplitude and phase) and carries units of amperes (A), while $\mathbf{f}_n$ has units of m^{-1} , so $\mathbf{J}$ has units of A/m (surface current density). The physical current at any point on the surface is the real part of the phasor field $\mathbf{J}(\mathbf{r}) e^{j\omega t}$. Substituting the definition of the RWG basis function into the current expansion expression, the current density at a point $\mathbf{r}$ in triangle T_t accumulates contributions from all the basis functions that involve T_t :

$$\mathbf{J}(\mathbf{r})|_{\mathbf{r} \in T_t} = \sum_{\{n: T_t \in \{T_n^+, T_n^-\}\}} I_n \frac{\sigma_n^t l_n}{2A_n^t} \boldsymbol{\rho}_n^t(\mathbf{r}), \quad (51)$$

where $\sigma_n^t = +1$ if $T_t = T_n^+$ and $\sigma_n^t = -1$ if $T_t = T_n^-$. For visualization, the magnitude of $\mathbf{J}$ is evaluated at the centroid of each triangle, which is simply the point $\bar{\mathbf{r}}_t = \frac{1}{3}(v_0 + v_1 + v_2)$, and surface current heatmaps of the structure are directly generated.

4.4.2 Far-Fields

In the radiation zone ($r \rightarrow \infty$) with the convention $e^{-j\omega t}$, the electric field radiated by the surface current $\mathbf{J}$ on a PEC body reduces to:

$$\mathbf{E}(\mathbf{r}) \xrightarrow{r \rightarrow \infty} -\frac{jk_0\eta_0}{4\pi} \frac{e^{-jk_0r}}{r} \hat{\mathbf{r}} \times \hat{\mathbf{r}} \times \mathbf{N}(\hat{\mathbf{r}}), \quad (52)$$

where the radiation vector $\mathbf{N}$ is:

$$\mathbf{N}(\hat{\mathbf{r}}) = \int_{\Omega} \mathbf{J}(\mathbf{r}') e^{+jk_0\hat{\mathbf{r}} \cdot \mathbf{r}'} dS'. \quad (53)$$

The positive sign in the exponent is opposite to the Green's function convention e^{-jk_0R} . It arises because in the far field $R = r - \hat{\mathbf{r}} \cdot \mathbf{r}' + O(1/r)$, so $e^{-jk_0R} \approx e^{-jk_0r} e^{+jk_0\hat{\mathbf{r}} \cdot \mathbf{r}'}$, and the e^{-jk_0r}/r factor is pulled out of the expression [6].

The double cross product projects $\mathbf{N}$ onto the transverse plane, enforcing the TEM behavior of the far field: $\hat{\mathbf{r}} \times \hat{\mathbf{r}} \times \mathbf{N} = \mathbf{N} - (\mathbf{N} \cdot \hat{\mathbf{r}})\hat{\mathbf{r}} = N_\theta \hat{\boldsymbol{\theta}} + N_\phi \hat{\boldsymbol{\phi}}$.

Projecting onto spherical unit vectors:

$$E_\theta = -\frac{jk_0\eta_0}{4\pi} \frac{e^{-jk_0r}}{r} N_\theta, \quad (54)$$

$$E_\phi = -\frac{jk_0\eta_0}{4\pi} \frac{e^{-jk_0r}}{r} N_\phi, \quad (55)$$

where the spherical basis vectors are:

$$\hat{\mathbf{r}} = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta),$$

$$\hat{\boldsymbol{\theta}} = (\cos \theta \cos \phi, \cos \theta \sin \phi, -\sin \theta),$$

$$\hat{\boldsymbol{\phi}} = (-\sin \phi, \cos \phi, 0).$$

Substituting $\mathbf{J} = \sum_n I_n \mathbf{f}_n$ and applying the Dunavant quadrature rule to each triangle integral:

$$\mathbf{N}(\hat{\mathbf{r}}) \approx \sum_{n=1}^N I_n \sum_{\alpha \in \{+, -\}} 2A_n^\alpha \sum_{i=1}^Q w_i \frac{\sigma_\alpha l_n}{2A_n^\alpha} \boldsymbol{\rho}_n^\alpha(\mathbf{r}'_i) e^{+jk_0\hat{\mathbf{r}} \cdot \mathbf{r}'_i}. \quad (56)$$

4.4.3 Radar Cross Sections

The bistatic RCS [6] $\sigma(\hat{\mathbf{r}})$ is the ratio of the scattered power density in direction $\hat{\mathbf{r}}$ to the incident power density, in the limit $r \rightarrow \infty$:

$$\sigma(\hat{\mathbf{r}}) = \lim_{r \rightarrow \infty} 4\pi r^2 \frac{|\mathbf{E}_s(\mathbf{r})|^2}{|\mathbf{E}_i|^2}. \quad (57)$$

Substituting the far-field expression (52) and cancelling the r^2 in numerator and denominator:

$$\sigma = \frac{4\pi}{|\mathbf{E}_0|^2} \left(\frac{k_0\eta_0}{4\pi} \right)^2 (|N_\theta|^2 + |N_\phi|^2). \quad (58)$$

In terms of the pattern functions returned by `compute_far_field`:

$$\sigma = 4\pi \frac{|E_\theta|^2 + |E_\phi|^2}{|\mathbf{E}_0|^2}, \quad (59)$$

where the prefactor $(k_0\eta_0/4\pi)^2$ is already embedded in E_θ and E_ϕ . The monostatic RCS (backscatter) direction is $\hat{\mathbf{r}} = -\hat{\mathbf{k}}$. For a $-z$ propagating wave ($\hat{\mathbf{k}} = -\hat{\mathbf{z}}$), the backscatter is at $+z$, which corresponds to $\theta = 0$ in spherical coordinates. In practice, $\theta = 0$ is a coordinate singularity, so $\theta = 0.001$ is used.

5 Validation and Examples

5.1 Spherical Plate Scattering

A PEC sphere of radius $a = 0.1$ m is meshed at $h = 0.02$ m $\approx \lambda/10$ at 1.5 GHz, producing $N \approx 600$ basis functions. The script sweeps 16 frequencies from 0.5 to 2.0 GHz, solving for monostatic RCS at each and comparing against the Mie series [13; 14]. At a fixed bistatic frequency ($f = 1.0$ GHz), the full bistatic pattern vs. elevation is also compared. Results agree within ± 0.5 dB across the sweep.

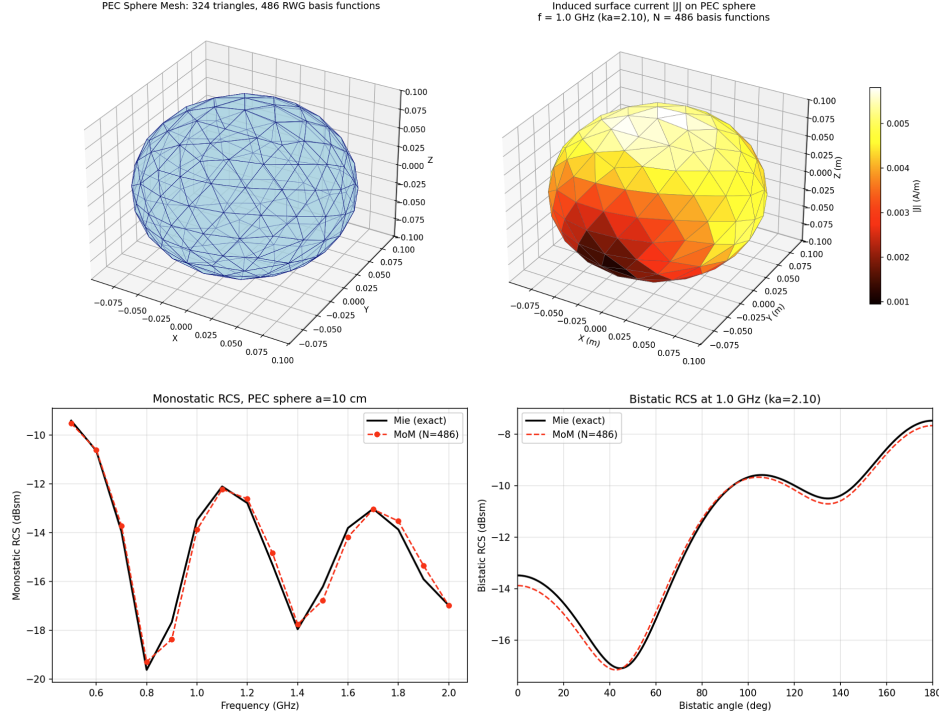


Figure 11: Comparison of PEC spherical scattering: MoM vs. Mie Series.

5.2 Dipole Antenna

A $15\text{ cm} \times 10\text{ mm}$ strip dipole is meshed with transverse edges at the feed plane and excited with a delta-gap source. The simulation computes $Z_{\text{in}}(f)$ at 21 frequencies. Resonance is identified by the $X_{\text{in}} = 0$ crossing. The real input impedance at resonance, R_{in} , at resonance is found to be $Z_{\text{in}} = 72.8 + j \cdot 0 \Omega$. This aligns with well-known results from [15], where the antenna input impedance for a finite-length half-wave dipole with non-zero thickness is between $67 - 75 \Omega$ depending on the antenna length-to-width ratio. Additionally, the resonant frequency is found to be $0.46\lambda \cdot \frac{c_0}{\ell}$ as opposed to the ideal of $0.5\lambda \cdot \frac{c_0}{\ell}$. This is due to fringing effects, which cause the antenna to look electrically longer than it really is. Lastly, the peak directivity ($D \approx 2.15$ dBi) is computed from far-field patterns, and is aligned with expectations.

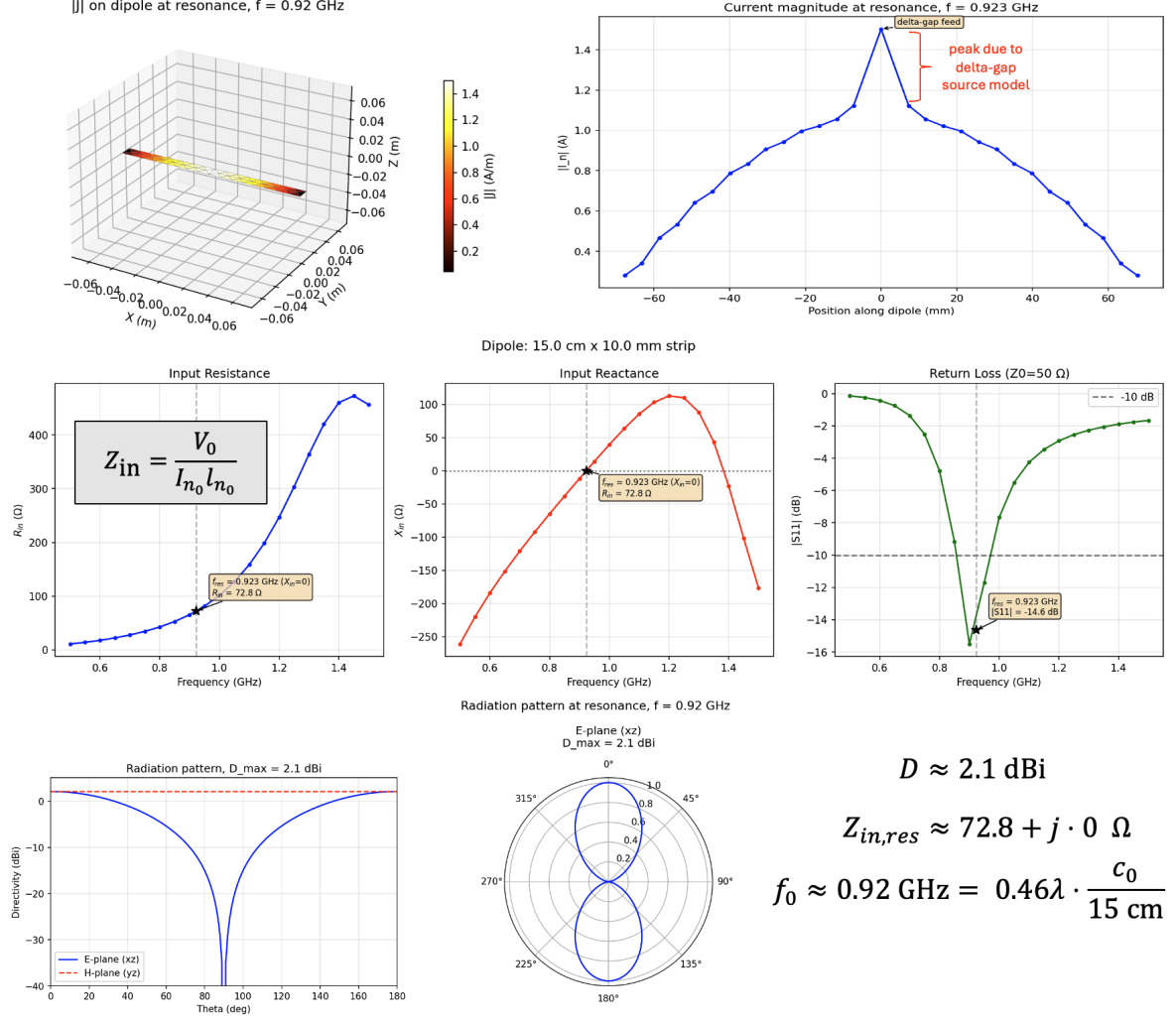
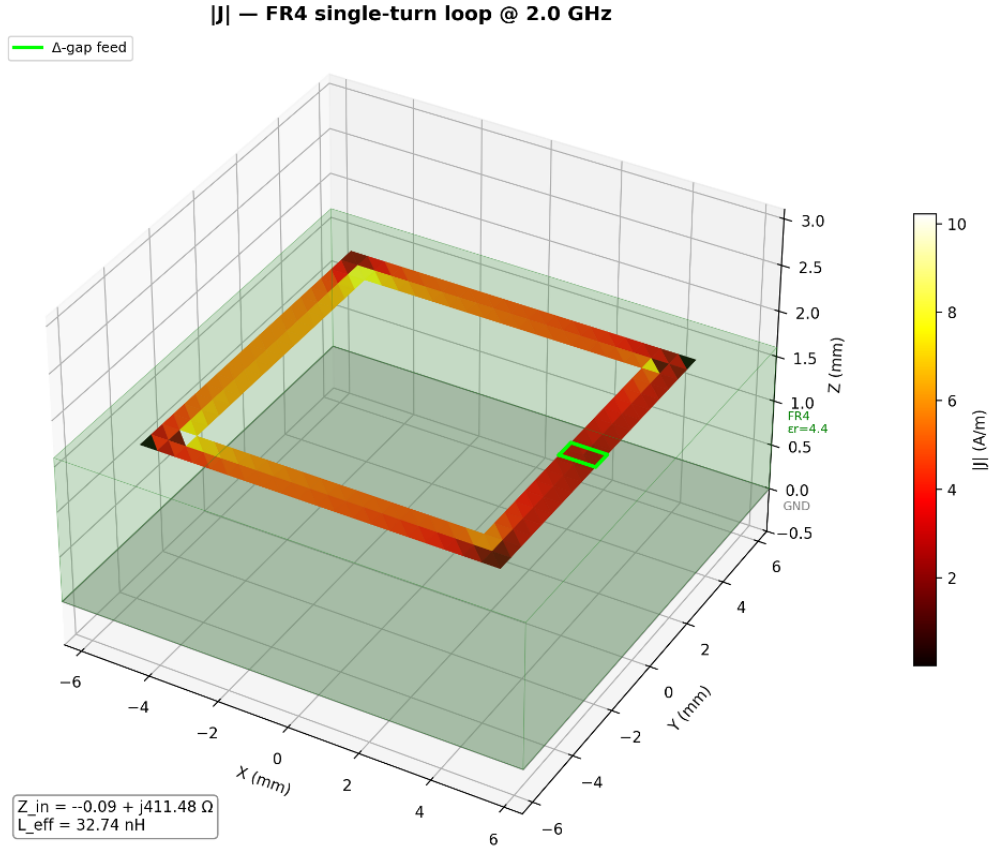


Figure 12: pyMoM3d simulation of a half-wave dipole antenna using a delta-gap excitation.

5.3 On-Chip Inductor

A single loop 10mm by 10mm planar rectangular inductor with a trace width of 1mm is meshed above a 1.5mm thick FR4 substrate above an infinite PEC conductor plane. It is excited with a delta-gap source and current density is plotted across the mesh. Impedance is evaluated using the applied delta-gap voltage and resulting current densities at $f_0 = 2.0\text{GHz}$ yielding a resistance of -0.09 Ohms and reactance of $+411.48\text{ jOhms}$.

A positive reactance of 411 jOhms at 2.0GHz yields an inductance of approximately 32nH which is consistent with the modified wheeler estimate for square planar inductors of 29.39 nH presented in [16]. While the Modified Wheeler estimate is framed in [16] towards SiO_2 dielectrics it does not consider C_{si} or the π model directly, making it equally applicable to FR4 and on-chip designs. It is generally considered the most accurate for discrete inductors, with an observed 15% error bound. While the negative real resistance is nonphysical, it's low magnitude and the PEC material supports its existence as a simulation artifact due to the coarseness and directionality of the inductor's mesh. Only two basis functions span the inductor width, limiting accuracy of the overall surface current modeling.



Integrated Spiral Inductor Calculator

Input Geometric Parameters

No. turns, $n =$ 1	Turn spacing, $s =$ 1 μm
Turn width, $w =$ 1 μm	Outer diameter, $d_{out} =$ 10000 μm

Results

	Square	Hexagonal	Octagonal
Modified Wheeler	29.394 nH	29.266 nH	28.261 nH
Current Sheet	79.292 nH	68.564 nH	67.484 nH
Monomial Fit	93.199 nH	106.439 nH	100.866 nH

Figure 13: Current density obtained from a multilayer simulation of a single loop inductor over an FR4 dielectric and PEC substrate and modified wheeler formula estimate from [16].

5.4 PEC Solver Performance

This section details a quantification of the PEC solver performance current supported in `pyMoM3d`. All measurements were taken on a single MacBook Pro CPU core using the `solver_performance.py`

benchmark: a PEC sphere of radius $a = 0.1$ m at $f = 1$ GHz ($ka = 2.09$), meshed at eight target edge lengths ranging from $h = 0.10$ m to $h = 0.01$ m, sweeping N from 70 to 3400 basis functions.

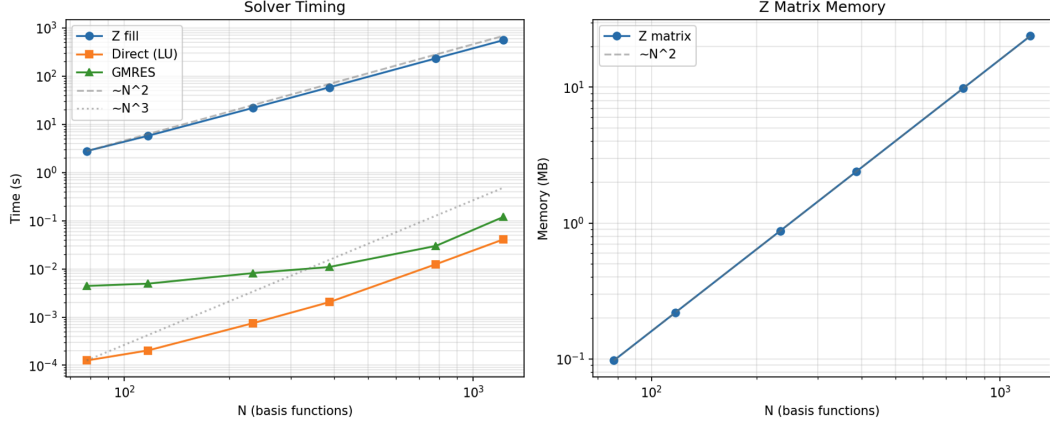


Figure 14: PEC solver performance profiling using the PEC sphere scattering example for an increasingly larger number of basis functions N .

Operation	Time Complexity	Memory
Mesh generation	$O(N_t)$	$O(N_t)$
RWG connectivity	$O(N_e)$	$O(N)$
Z-matrix fill	$O(N^2Q^2)$	$O(N^2)$
Direct solve	$O(N^3)$	$O(N^2)$
GMRES solve	$O(kN^2)$	$O(kN)$
Far-field	$O(NMQ)$	$O(M)$

Table 2: Computational complexity and memory requirements for MoM operations.

N	Z-fill time	Solve time	Memory
100	~ 1 s	< 0.1 s	~ 1 MB
500	~ 30 s	~ 1 s	~ 4 MB
1000	~ 2 min	~ 5 s	~ 16 MB
2000	~ 10 min	~ 40 s	~ 64 MB
5000	~ 1 hr	~ 10 min	~ 400 MB

Table 3: Empirical runtime and memory scaling with number of basis functions N .

6 Future Work

pyMoM3d establishes a complete, validated MoM solver pipeline in Python — from geometry and meshing through impedance matrix assembly, excitation, solving, and post-processing — with support for both free-space and layered media problems. The PEC solver has been validated against analytical references including the Mie series, half-wave dipole theory, and Friis transmission equation, demonstrating accuracy across scattering, antenna, and array problems. The multilayer support has been validated against simple planar microstrip transmission lines and multi-turn inductor geometries, demonstrating results within an acceptable error level. The most immediate priority for future development is performance. The current single-threaded Python implementation becomes the bottleneck at large mesh sizes, with Z-matrix fill scaling as $\mathcal{O}(N^2Q^2)$ and taking on the order of one hour at $N = 5000$ basis functions. Rewriting the inner fill loop in C++, adding multi-threading or GPU acceleration, and ultimately implementing Fast Multipole Methods (FMM) to reduce complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ would make the solver competitive with commercial tools for large-scale problems. The Combined Field Integral Equation (CFIE) would also address spurious interior resonances that currently affect closed-body simulations with an EFIE only based solver.

On the application side, the layered media capability opens a direct path toward IC and PCB passive component analysis. A polished layout meshing pipeline connecting a GDSII or Gerber import to the existing Gmsh mesher and DCIM kernel would enable on-chip inductor synthesis and topology optimization for multi-layer passive design. Adding finite conductivity and metal thickness effects would further close the gap between simulation and fabricated device performance.

The long-term vision for pyMoM3d is to grow into a professional open-source MoM solver, complementing existing tools such as Palace [17] (FEM, Amazon Web Services Labs) and openEMS [18] (FDTD), with a particular focus on rapid algorithmic experimentation and educational transparency that commercial solvers cannot offer.

References

- [1] M. Gustafsson, K. Schab, L. Jelinek, and M. Capek, “Upper bounds on absorption and scattering,” *New Journal of Physics*, vol. 22, no. 7, p. 073013, Jul. 2020. [Online]. Available: <http://dx.doi.org/10.1088/1367-2630/ab83d3>
- [2] R. F. Harrington, *Field Computation by Moment Methods*, ser. IEEE Press Series on Electromagnetic Wave Theory. Piscataway, NJ, USA: IEEE Press, 1993, reprint of the 1968 edition.
- [3] A. Rothschild and O. Ryan, “pymom3d: A python method of moments solver for 3d electromagnetic problems,” <https://github.com/austinrosh/pyMoM3d>, 2026, accessed: 2026-03-19.
- [4] C. Geuzaine and J.-F. Remacle, “Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities,” *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579>
- [5] S. Rao, D. Wilton, and A. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 3, pp. 409–418, 1982.
- [6] W. C. Gibson, *The Method of Moments in Electromagnetics*, 3rd ed. London, U.K. and Boca Raton, FL, USA: Chapman & Hall/CRC, 2021.

- [7] R. Graglia, “On the numerical integration of the linear shape functions times the 3-d green’s function or its gradient on a plane triangle,” *IEEE Transactions on Antennas and Propagation*, vol. 41, no. 10, pp. 1448–1455, 1993.
- [8] D. A. Dunavant, “High degree efficient symmetrical gaussian quadrature rules for the triangle,” *International Journal for Numerical Methods in Engineering*, vol. 21, no. 6, pp. 1129–1148, 1985. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620210612>
- [9] S. Sharma and P. Triverio, “Strata: Green’s functions for layered media,” 2021, c++ library. [Online]. Available: <https://github.com/modelics/strata>
- [10] K. A. Michalski and D. Zheng, “Electromagnetic scattering and radiation by surfaces of arbitrary shape in layered media, Part I: Theory,” *IEEE Transactions on Antennas and Propagation*, vol. 38, no. 3, pp. 335–344, 1990.
- [11] K. A. Michalski and J. R. Mosig, “Multilayered media Green’s functions in integral equation formulations,” *IEEE Transactions on Antennas and Propagation*, 1997.
- [12] J. Budhu, “Lecture 14: Setting up 3d method of moments problems,” <https://www.youtube.com/watch?v=08zBIPpFqCs>, n.d., youTube video, Advanced Engineering Electromagnetics II, Virginia Tech.
- [13] G. Mie, “Beiträge zur optik trüber medien, speziell kolloidaler metallösungen,” *Annalen der Physik*, vol. 330, no. 3, pp. 377–445, 1908.
- [14] C. F. Bohren and D. R. Huffman, *Absorption and Scattering of Light by Small Particles*. New York: Wiley, 1983.
- [15] C. A. Balanis, *Antenna Theory: Analysis and Design*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.
- [16] S. S. Mohan, M. d. M. Hershenson, S. P. Boyd, and T. H. Lee, “Simple accurate expressions for planar spiral inductances,” *IEEE Journal of Solid-State Circuits*, vol. 34, no. 10, pp. 1419–1424, October 1999.
- [17] AWS Center for Quantum Computing, “Palace: 3d finite element solver for computational electromagnetics,” <https://github.com/awslabs/palace>, 2023–, open-source parallel finite element solver for full-wave electromagnetic simulations.
- [18] T. Liebig, “openems – open electromagnetic field solver,” University of Duisburg-Essen, ATE, 2010–, free and open-source FDTD electromagnetic solver. [Online]. Available: <https://www.openEMS.de>

A Use of AI Tools

Claude (Anthropic) and ChatGPT (openAI) were used as development aids in this project. Its primary roles were code review and debugging, generating boilerplate and documentation, and answering questions about numerical methods and electromagnetic theory during development. All core solver components — including the EFIE formulation, RWG connectivity, impedance matrix assembly, singularity extraction, DCIM integration, excitation sources, and validation examples — were implemented by the

authors. AI assistance was used to accelerate development and improve code quality (i.e. documentation, comments, linting, stylistic plotting adjustments).

The development of `pyMoM3d` reflects a broader shift in how software engineering is practiced. AI coding assistants have become standard tools in professional development, analogous to how compilers, debuggers, and IDEs changed programming in prior decades. They are most effective when the developer has deep domain expertise — in this case, a thorough understanding of computational electromagnetics was necessary to direct, evaluate, and validate every piece of AI-assisted output. Used this way, AI assistance compresses implementation time without replacing the intellectual contribution of the engineer, and allows for professionally polished code-bases to be developed with much higher throughput.

B Details on Source Excitations

B.1 Plane-Wave Sources

A time-harmonic plane wave propagating in direction $\hat{k}$ with polarization $\mathbf{E}_0$ is:

$$\mathbf{E}_i(\mathbf{r}) = \mathbf{E}_0 e^{-jk \hat{k} \cdot \mathbf{r}}, \quad (60)$$

where $\mathbf{E}_0 \perp \hat{k}$ (transversality) and $k = \omega \sqrt{\mu_0 \varepsilon_0}$ is the free-space wavenumber. Inserting (60) into (50) gives the element-level integral:

$$V_m^\pm = \pm \frac{l_m}{2A_m^\pm} \int_{T_m^\pm} \boldsymbol{\rho}_m^\pm(\mathbf{r}) \cdot \mathbf{E}_0 e^{-jk \hat{k} \cdot \mathbf{r}} dS. \quad (61)$$

Because the integrand is smooth (no singularity), standard Gauss quadrature on triangles is sufficient. Using the Q -point Dunavant rule with weights w_i and barycentric quadrature points $\mathbf{r}_i \in T^\pm$:

$$V_m^\pm \approx \pm \frac{l_m}{2A_m^\pm} (2A_m^\pm) \sum_{i=1}^Q w_i (\mathbf{r}_i - \mathbf{r}_m^{\pm, \text{free}}) \cdot \mathbf{E}_0 e^{-jk \hat{k} \cdot \mathbf{r}_i}, \quad (62)$$

where the factor $2A_m^\pm$ is the Jacobian of the barycentric map and the normalized weights satisfy $\sum w_i = 1/2$. The total entry is $V_m = V_m^+ + V_m^-$.

B.2 Delta-Gap Sources

The delta-gap model is the standard idealized feed for wire and strip antennas [?]. A voltage V_0 is applied across an infinitesimally thin gap at a single mesh edge. The corresponding applied electric field is:

$$\mathbf{E}^{\text{gap}}(\mathbf{r}) = V_0 \delta(\xi - \xi_{\text{feed}}) \hat{\xi}, \quad (63)$$

where ξ is the coordinate along the antenna axis and ξ_{feed} is the feed location. Taking the Galerkin inner product with basis function $\mathbf{f}_m$ and using the sifting property of the delta function:

$$V_m = V_0 \int_{\ell_m} \mathbf{f}_m \cdot \hat{\xi} d\ell. \quad (64)$$

For an RWG basis function, the flux through the shared edge ℓ_m is a known identity [5]: $\int_{\ell_m} \mathbf{f}_m \cdot \hat{n} d\ell = l_m$, where $\hat{n}$ is the edge normal pointing from T_m^- to T_m^+ . When the edge is oriented along the feed direction ($\hat{n} \parallel \hat{\xi}$), this gives:

$$V_m = \begin{cases} V_0 l_m & \text{if } m = \text{feed basis index,} \\ 0 & \text{otherwise.} \end{cases} \quad (65)$$

Thus, delta-gap sources do not require numerical quadrature and are thus very computationally efficient.

In practice, artificially injecting current at only one feed-point can learn to sharp peaks in the surface current distribution as local basis around the feed see a roll-off in current. To combat this, delta-gaps can be applied to finer meshes or tapered across multiple basis elements at a feedpoint. The *strip delta-gap* model distributes the applied voltage uniformly across all N_{feed} transverse feed edges:

$$V_m = \begin{cases} V_0 l_m & \text{if } m \in \mathcal{F}, \\ 0 & \text{otherwise,} \end{cases} \quad (66)$$

where $\mathcal{F}$ is the set of feed basis indices. The terminal current is the sum of the individual edge currents weighted by edge length:

$$I_{\text{term}} = \sum_{m \in \mathcal{F}} I_m l_m, \quad (67)$$

so the input impedance is $Z_{\text{in}} = V_0/I_{\text{term}}$. This formulation is equivalent to connecting N_{feed} ideal voltage sources in parallel across the width of the strip, which faithfully models a coaxial or SMA feed whose outer conductor spans the full strip width.

C Multi-Antenna Modeling

Solving multi-antenna problems involves combination of independently generated meshes for each substructure. Each structure is independently meshed by `GmshMesher`, producing separate `Mesh` objects with their own vertex index spaces. To solve a single system of equations coupling all structures, the meshes must be combined into one `Mesh` with globally consistent vertex indices, while remaining electromagnetically disjoint (no shared edges between antennas). After combination of meshes into one super-mesh, the usage of the solver pipeline remains the same. The corresponding impedance matrix becomes block-diagonal; every basis function n on the TX structure generates a scattered field at every basis function m on the RX structure; this is captured in the off-diagonal block $Z_{mn}^{\text{RX},\text{TX}}$ of the combined impedance matrix:

$$\begin{bmatrix} Z^{\text{TX},\text{TX}} & Z^{\text{TX},\text{RX}} \\ Z^{\text{RX},\text{TX}} & Z^{\text{RX},\text{RX}} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{\text{TX}} \\ \mathbf{I}_{\text{RX}} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{\text{TX}} \\ \mathbf{0} \end{bmatrix}. \quad (68)$$

C.1 Friis Transmission Validation

The Friis equation [15] gives the received power between two antennas in free space in terms of their gains and the propagation path:

$$\frac{P_{\text{rx}}}{P_{\text{tx}}} = G_{\text{tx}} G_{\text{rx}} \left(\frac{\lambda}{4\pi R} \right)^2 |\hat{\mathbf{e}}_{\text{tx}} \cdot \hat{\mathbf{e}}_{\text{rx}}^*|^2, \quad (69)$$

where:

- $G_{\text{tx}}, G_{\text{rx}}$ — gain of the transmit and receive antennas in the link direction;
- $\lambda/4\pi R$ — free-space path loss;
- $|\hat{\mathbf{e}}_{\text{tx}} \cdot \hat{\mathbf{e}}_{\text{rx}}^*|^2$ — polarization loss factor (PLF).

For identical co-polarized antennas with broadside gain G and a link along the broadside direction, Eq. (69) simplifies to:

$$\frac{P_{\text{rx}}}{P_{\text{tx}}} = G^2 \left(\frac{\lambda}{4\pi R} \right)^2. \quad (70)$$

With a polarization rotation ψ about the link axis, the PLF becomes $\cos^2(\psi)$, so:

$$\frac{P_{\text{rx}}}{P_{\text{tx}}} = G^2 \left(\frac{\lambda}{4\pi R} \right)^2 \cos^2(\psi). \quad (71)$$

A half-wave dipole has a theoretical broadside gain of $G \approx 2.15$ dBi, with a donut-shaped radiation pattern. Two $\lambda/2$ strip dipoles at 5 GHz are combined into a two-element system. A single impedance matrix is filled for the combined mesh; the received power ratio $P_{\text{rx}}/P_{\text{tx}}$ is extracted via the Thevenin equivalent and compared against the Friis formula. Agreement is within 1 dB over $R/\lambda \in [5, 15]$ and the $\cos^2 \psi$ polarization factor is recovered within 1.5 dB.

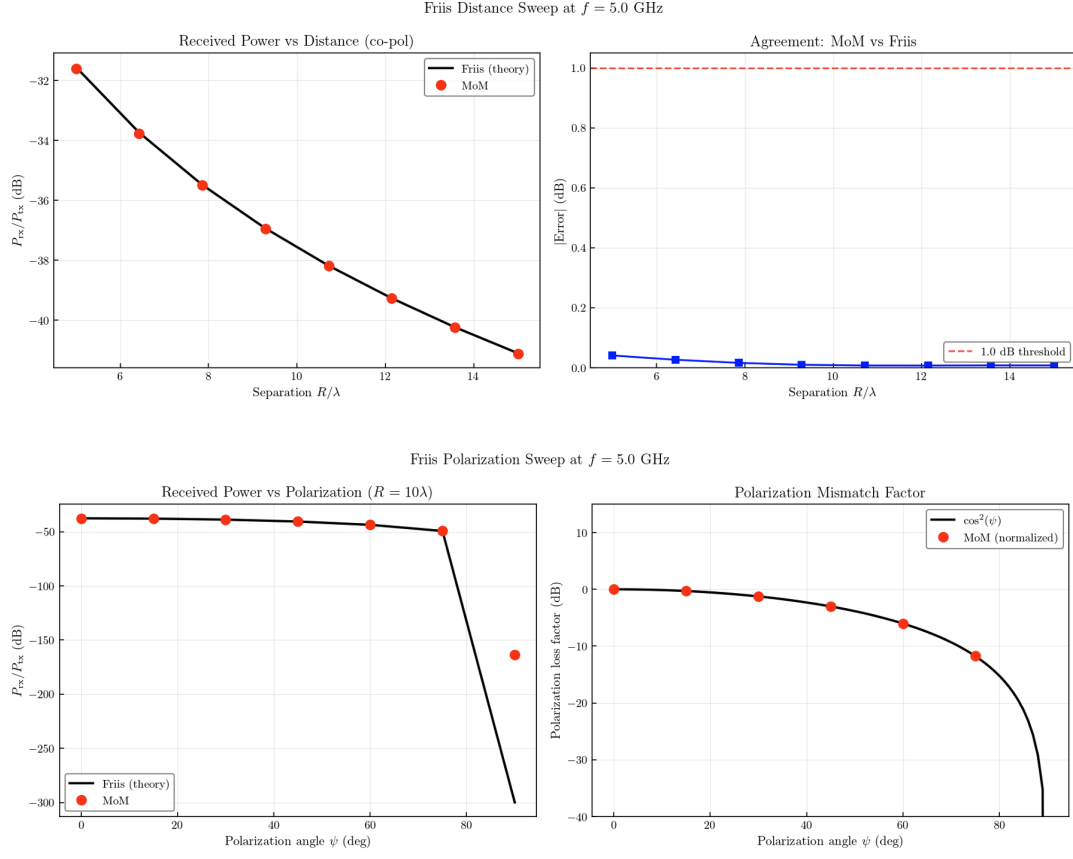


Figure 15: Simulation of LoS TX/RX 5 GHz half-wave dipole antenna link vs. expected performance governed by Friis theory.

D Final Project Timeline

Feature	Target Date	Ownership	Status
Geometry handling and Gmsh pipeline	2/6	Austin & Owen	Complete
RWG basis function support	2/9	Austin	Complete
PEC EFIE $\mathbf{Z}$ -matrix evaluation w/ Galerkin testing and Gaussian quadrature	2/14	Austin	Complete
Plane wave and delta-gap excitation support	2/20	Austin	Complete
PEC EFIE solver analytical benchmarking (spherical Mie series, dipole antenna, etc.)	2/25	Austin	Complete
Numerical formulation for Green's functions in layered media	2/20	Owen	Complete
Multilayer structure support in pyMoM3d backend	3/01	Owen	Complete
Multi-antenna scattering (linear arrays, LoS TX/RX link)	3/01	Austin	Complete
Multilayer passive design simulation (transmission lines, inductors)	3/09	Owen	Complete
Project presentation slides done	3/17	Austin & Owen	Complete
Final report done	3/19	Austin & Owen	Complete
Complete pyMoM3d GitHub documentation	3/20	Austin & Owen	Complete

Table 4: pyMoM3d Feature Support Timeline and Ownership