Approximating the Capacitance of Square Plates; A Method of Moments Convergence Study

Samuel J. Wyss[†]
[†]School of Nuclear Engineering
Purdue University
West Lafayette, Indiana 47907
E-mail: wysss@purdue.edu

***Abstract*—The Method of Moments (MoM) is applied to simulate the capacitance of an arbitrarily sized square metallic plate. To model this system *in silico*, the Electrostatic Integral Equation (EIE) is used to simulate the charge distribution for a fixed potential on the metallic plate. The charge is then determined by integrating it element-wise over the surface which is then used to solve for capacitance. This procedure is used to evaluate various forms of the underlying integral equations from approximations of square elements, exact forms, to subdomain collocation. Capacitance values from these methods are first compared to data found in the literature as a verification and validation step and then compared to each other in terms of the mesh size required for these methods to achieve the same level of precision.**

I. Introduction

Contrary to many other methods in Computational Electromagnetics (CEM), the Method of Moments (MoM) is purely based on the integral form of Maxwell's Equations and underlying Green's Functions [1]. In the context of the Electrostatic Integral Equation (EIE), MoM uses the potential created by individual point charges on surfaces to solve for charge density as a function of surface location. This charge distribution can trivially be integrated over surface elements to solve for the net charge on an surface for an arbitrary geometry and applied potential. From this, the capacitance of said structure can easily be obtained.

In matrix form, the EIE relates the point source response of charges in a given surface element to all other surface elements and the applied potential. The use of Green's Functions in the problem formulation allows for these problems to be solved without the need to terminate models spatially which is a common source of error in other CEM methods. However, the use if Green's Functions results in fully dense matrices, thus requiring the computational complexities and memory footprint that comes with solving them which at first may be seen as a downside of MoM. However, As these matrices only require the discretization of surfaces [1], [2] thereby resulting in much smaller system matrices overall which helps to compensate for the typical $O(n^3)$ computation complexity required to solve these dense systems.

The development and results of this work are laid out as follows. Section II contains a short derivation of the EIE from Gauss's Law followed by the setup of the EIE matrix equations for a circular approximation of square surface elements, an exact form for surface elements, and finally the subdomain collocation. Section III contains a verification of the model with a capacitance range found in the literature, an explanation of the plate charge distribution, an analysis of the asymptotic number of operations required to determine the plate capacitance for all three methods. Finally, Section IV contains closing remarks regarding the analysis and potential future work.

II. Mathematical Model

To model these systems *in silico*, an appropriate mathematical model must first be derived from Maxwell's Equations. The development of said model is arranged as follows. Section II-1 contains the derivation of the EIE from Maxwell's Equations followed by surface element discretization into a simultaneous set of equations. Section II-2 outlines three formulations of the Systems matrix A starting with elements approximated as circles, an exact solution, and finally subdomain collocation.

1) The Electrostatic Integral Equation (EIE):

From basic electrostatics, the integral form of Gauss's Equation over some surface S is as follows

$$\iint_S \epsilon_0^{-1} g(\mathbf{r},\mathbf{r'})\rho_s(\mathbf{r'})dS' = \Phi(r) \tag{1}$$

where ϵ_0 is the permittivity of free space, $\rho_s(\mathbf{r'})$ is the surface charge density, $\Phi(r)$ is the electrostatic potential, and $g(\mathbf{r},\mathbf{r'})$ is the corresponding Green's function (integration kernel)

$$g(\mathbf{r},\mathbf{r'}) = \frac{1}{4\pi|\mathbf{r}-\mathbf{r'}|}. \tag{2}$$

Basic electrostatic theory is also used to enforce that for a sufficiently conductive metal (as assumed in this problem) the surface of the plate must be at an equipotential, thus $\Phi(r) \rightarrow \Phi_0$ where Φ_0 is an arbitrary electric equipotential. Under these stipulations, the only unknown in this problem is the charge density on the plate surface. To solve, we write the charge density in terms of a series of linearly independent basis functions, $v_n(\mathbf{r'})$, and expansion coefficients, c_n, defined over individual elements as follows [1], [2]

$$\rho_s(\mathbf{r'}) = \sum_{n=1}^{N} c_n v_n(\mathbf{r'}). \tag{3}$$

With the surface charge density discretized, it is now necessary to sample the underlying integral equations in a similar fashion using testing functions $w_m(\mathbf{r})$. With these definitions, 1 can now be expressed as the EIE which is defined as follows

$$\sum_{n=1}^{N} c_n \iint_S \iint_S w_m(\mathbf{r})\epsilon_0^{-1} g(\mathbf{r},\mathbf{r'}) v_n(\mathbf{r'}) dS' dS = \iint_S w_m(\mathbf{r})\Phi_0 dS. \tag{4}$$

The EIE, (4), can be expressed as the following matrix equation

$$[A]\{c\} = \{b\} \tag{5}$$

where A is the system matrix defined as

$$[A]_{mn} = \iint_S \iint_S w_m(\mathbf{r})\epsilon_0^{-1} g(\mathbf{r},\mathbf{r'}) v_n(\mathbf{r'}) dS' dS, \tag{6}$$

b contains the equipotential boundary condition

$$\{b\}_m = \iint_S w_m(\mathbf{r})\Phi_0 dS, \tag{7}$$

and c is an array of charge density scalars corresponding to their indexed element which is solved for. The following section will go over several methodologies for defining the system matrix and equipotential boundary condition array.

2) System Matrix Formulations: For simplicity only the following zeroth order basis function will be used for all analyses

$$v_n(\mathbf{r'}) = \begin{cases} 1, & r' \in S_n \\ 0, & \text{else} \end{cases} \tag{8}$$

where S_n is defined as the area of the square surface element n.

The first method that will be investigated involves using a Dirac delta function as the testing function

$$w_m(\mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}_m) \tag{9}$$

where $\mathbf{r}_m$ denotes the center of element m. This testing function can be interpreted as testing each element at it's respective center point only [1] This testing function will be used for the first two methods discussed.

Using (9), (6)-(7) can be re-written as

$$[A]_{mn} = \iint_{S_n} \epsilon_0^{-1} g(\mathbf{r},\mathbf{r'}) dS', \tag{10}$$

and

$$\{b\}_m = \Phi_0. \tag{11}$$

This leads to the first two formulations of the system matrix A. In the first case, (10) can be approximated using midpoint integration for all $m \neq n$. The case containing $m = n$ contains a numerical singularity. To alleviate this, the element can be approximated as a circular element possessing the same area as S_n. This results in the following formulation of the system matrix,

$$[A]_{mn} = \begin{cases} \frac{S_n}{4\pi\epsilon_0 |\mathbf{r}_m - \mathbf{r}_n|}, & m \neq n \\ \frac{1}{2\epsilon_0}\sqrt{\frac{S_n}{\pi}}, & m = n. \end{cases} \tag{12}$$

For the duration of this paper, this method will be referred to as the circular element approximation.

The next methodology directly integrates (10) for square elements thus eliminating the need to use an approximation. The closed form of (10) for square elements is

$$[A]_{mn} = \frac{1}{4\pi\epsilon_0}\big((x_m - x')\ln((y_m - y') + R) + (y_m - y')\ln((x_m - x') + R)\big) \tag{13}$$

where $R = \sqrt{(x_m - x')^2 + (y_m - y')^2}$ evaluated over $x' = x_n \pm \Delta x/2$, $y' = y_n \pm \Delta y/2$ [2]. This method will be referred to as the exact square element method.

For the third and final method studied here, the pulse function is used as a basis function as opposed to the Dirac delta function, (9), used in (10). This is physically equivalent to averaging the testing function over the entire element as opposed to measuring only at the element center point as in the first two methods [1], [2]. This methodology gives rise to the following system matrix

$$[A]_{mn} = \iint_{S_m} \iint_{S_n} \epsilon_0^{-1} g(\mathbf{r},\mathbf{r'}) dS' dS, \tag{14}$$

and boundary condition array

$$\{b\}_m = S_m \Phi_0. \tag{15}$$

Under this formulation, (14) can be exactly solved resulting in

$$[A]_{mn} = \frac{1}{4\pi\epsilon}\Big(\frac{(x-x')^2(y-y')}{2}\ln((y-y')+R) + \frac{(x-x')(y-y')^2}{2}\ln((x-x')+R) - \frac{(x-x')(y-y')}{4}((x-x')+(y-y')) - \frac{R^3}{6}\Big) \tag{16}$$

where $R = \sqrt{(x-x')^2 + (y-y')^2}$ evaluated over $x' = x_n \pm \Delta x/2$, $y' = y_n \pm \Delta y/2 x = x_m \pm \Delta x/2$, $y = y_m \pm \Delta y/2$ [2]. This method is referred to as subdomain collocation. Unlike the previous set of equations, this method results in symmetric matrices thereby allowing a halving of the total memory footprint (assuming appropriate data structure and solver choice) as only the upper diagonal matrix needs to be stored. Despite this, solving these systems is still asymptotically $O(n^3)$.

One major issue with this formulation is the fact that indeterminate terms ($\ln(0) \times 0$) arise when $x = x'$ or $y = y'$. A trivial means of resolving these indeterminate forms is to add a very small constant c to each $\ln(x + c)$ call such that c is much less than x for any value of x. Firstly, this prevents indeterminate forms from arising as the natural log of a small number multiplied by zero evaluates to zero. Secondly, provided that $c << x$, the error introduced by including c can be reduced to machine precision for sufficiently small c. Since the introduced error can be made arbitrarily small, there is no need to conditionally check for problematic arguments to $\ln()$ calls as this c term can always be included. This increases assembly efficiency as the matrix assembler does not need to conditionally check for problematic arguments 16 times for all n^2 elements in the system matrix. For the purposes of this work, $c = 1 \times 10^{-100}$ was chosen as it satisfies $c << x$.

III. Numerical Results

With the mathematical model now fully established, the implementation of this model is documented in Section III-A. From here Sections III-B verifies the model against exact dispersion relations for multiple modes in rectangular waveguides. Section **??** performs an analysis of Dispersion characteristics in circular waveguides and addresses the advantages of using FEA for this task. Finally, Section **??** compares the dispersion characteristics of the rectangular and ridged rectangular waveguides for multiple modes and discusses practical applications of ridged waveguides.

A. Implementation

meshing
code implementation

B. Verification and Validation

present data for 10,20,30 (maps,a nd data)

Table 1: Predicted Capacitances [pF]

Method	10×10	20×20	30×30
Circular Element Approximation	0.397	0.403	0.405
Exact Square Elements	0.394	0.401	0.403
Subdomain Collocation	0.399	0.403	0.405

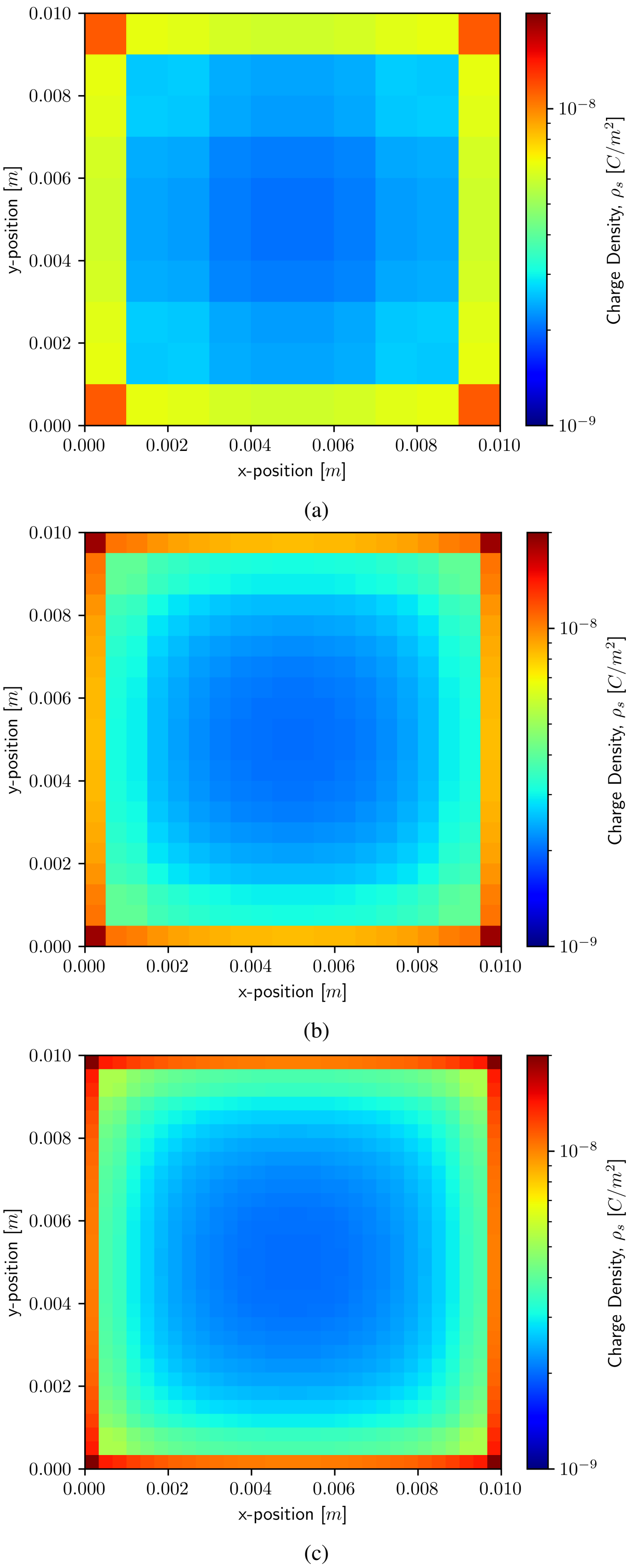
random walk solutions
show agreement

C. Convergence Study

plot and talk

IV. Conclusion

A 2-dimensional finite element method program was developed from Maxwell's Equations allowing for field distribution and dispersion analysis of arbitrarily shaped homogenous waveguides. The model was first validated against the analytic field profiles and dispersion curves for a rectangular waveguide. From here, additional verification was performed on the field profiles of circular waveguides. Next, the model was used to predict the dispersion characteristics of said circular guide which can be difficult to do analytically without tabulated Bessel function data. Finally, the model was used to compare ridged waveguides to their non-ridged counterparts. It was shown that the inclusion of ridges significantly expands the bandwidth available to the dominant TE_{10} mode in alignment with theory. This phenomenon was explained using the change in field buckling caused by complex modes requiring higher frequencies to achieve the same field profiles where as the buckling working in the favor of simpler field profiles resulting in a lowering of the cutoff wave number.



(a)

(b)

(c)

Fig. 1: Use of wave ports to analyze a cylindrical cavity resonator. (a) The geometry analyzed and (b) the comparison of numerical and measured results (from [2]).

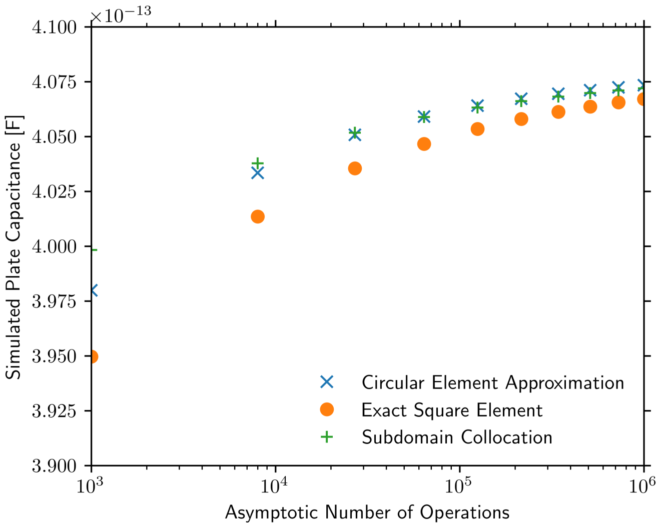


Fig. 2: Example of caption text.

While relatively general in the sense that the model works for any `.inp` input waveguide mesh, the model would benefit from increased work regarding the identification of modes which currently is done manually by the user in order to produce dispersion curves and field profile plots. The model would also benefit from using sparse matrix types as defined in SciPy as opposed to the dense NumPy arrays used in this program. In addition to this, it would be interesting to combine this model with the FDTD model developed in the last project in order to generate the field profiles of more complex waveguides for use in the TF/SF source condition. This would allow for temporal analysis of more sophisticated waveguides than the simple rectangular model used in the former project.

References

[1] T. E. Roth, *ECE 61800 Lecture Notes*. Purdue University, 2024.

[2] J.-M. Jin, *Theory and Computation of Electromagnetic Fields*. John Wiley & Sons, 2011.