A 2D FINITE ELEMENT SOLVER FOR ELECTROMAGNETIC FIELDS WITH M-FOLD AZIMUTHAL SYMMETRY

A. Vrielink*, M. Nasr, S. Tantawi, SLAC, Menlo Park, US

Abstract

Radiofrequency (RF) cavities for use in accelerators, from RF sources to accelerating and transverse cavities, often exhibit m-fold azimuthal symmetry. For cases where m>0, commercially available finite element codes used to simulate the beam-wave interaction typically require a full 3D simulation. We have derived a finite element formulation which accounts for the known azimuthal dependence of the electromagnetic fields, allowing us to solve for these problems on a 2D mesh and reducing simulation times significantly. The theory, including the construction of the local finite element matrices and the selection of appropriate basis functions, will be presented in addition to numerical results.

INTRODUCTION AND MOTIVATION

The ability to rapidly simulate the fields and beam-wave interactions in increasingly more complicated RF structures has enabled significant advances in the field of particle accelerator engineering. As we move to higher frequency structures which must be highly overmoded to accommodate reasonable beam current, simulation codes are being tested to their limits in terms of memory and computation time. For structures such as these, simulations can take on the order of hours to days, rendering the design and optimization of such systems challenging. Fortunately, in accelerator physics, most RF fields of interest have some degree of azimuthal symmetry of the form $e^{im\theta}$.

While it is common to find axisymmetric solvers ($m = 0$), we are interested in developing a finite element formulation that would allow us to account for the known azimuthal dependence beforehand, for arbitrary m, reducing the problem to one that can be solved on a 2D mesh. Herein we present this formulation, first for the electrostatic case and then in the context of an eigenmode solver. We present some details of our computational implementation and conclude with plans for future work.

ELECTROSTATIC CASE

We start in Eq.(1) with the variational formulation of Poisson's equation. This is a special case of the general Lagrangian formulation for the electromagnetic potentials (steady state, $\vec{A} = 0$).

$$\mathcal{L} = \int \frac{\epsilon}{2}|\nabla\phi|^2 + \rho\phi \, dv \tag{1}$$

We can substitute $\phi = \phi(r, z)e^{im\theta}$ and evaluate this integral over $\theta = (0, 2\pi)$. The resulting expression is given by Eq. (2).

$$\mathcal{L} = \int \frac{\epsilon}{2}[m^2\frac{\phi(r,z)^2}{r} + r(\frac{\partial\phi(r,z)}{\partial z})^2 + r(\frac{\partial\phi(r,z)}{\partial r})^2] + r\rho(r,z)\phi(r,z)\, dr dz \tag{2}$$

As the next step in the finite element method, we discretize our solution space and expand the potential into a linear combination of basis functions which span this discretized space: $\phi(r, z) = \sum_i \Phi_i \xi_i(r, z)$ where $\xi_i(r, z)$ are the well defined basis functions and Φ_i are the unknown coefficients that we will solve for. In our implementation we used a triangular mesh with quadratic basis functions. A natural choice for the basis functions are thus the 6 nodal basis functions, $\eta_i(r_j, z_j) = \delta_{ij}$ where (r_j, z_j) are the six sample points on the triangle (3 vertices and 3 midpoints).

However, this choice of basis function is not necessarily the best choice and can lead to very poor convergence of the solution depending on the value of m. For all $m \neq 0$, $E_z = 0 \implies \partial_z\phi = 0$ on axis, so a much better choice for the basis function is $\xi_i(r, z) = r^x\eta_i(r, z)$ for some power, x. A basis function of this type is essentially made necessary anyway by the singularity at $r = 0$ that arises in the integral in Eq.(2) from the $\frac{m}{r}$ term.

Having discretized the Lagrangian, we can write it as the matrix equation in Eq.(3). The variation is then done by taking $\partial\underline{\Phi} = [\frac{\partial}{\partial\Phi_1}, \frac{\partial}{\partial\Phi_2}, ...]$ such that we can obtain the approximate solution, Φ by solving Eq.(4).

$$\mathcal{L} = \underline{\Phi}\,\mathbf{L}\,\underline{\Phi}^{\mathrm{T}} \tag{3}$$

$$\partial_\Phi\mathcal{L} = \mathbf{L}\,\underline{\Phi}^{\mathrm{T}} = 0 \tag{4}$$

$$\mathbf{L}_{ij} = \epsilon\int m^2\frac{\xi_i(r,z)\xi_j(r,z)}{r} + r\Big(\frac{\partial\xi_i(r,z)}{\partial z}\frac{\partial\xi_j(r,z)}{\partial z} + \frac{\partial\xi_i(r,z)}{\partial r}\frac{\partial\xi_j(r,z)}{\partial r} + \rho(r,z)\Big)dS \tag{5}$$

Note the integral in Eq.(5) is taken over the surface of each triangle element. $\mathbf{L}$ can be calculated either numerically or analytically for each element invidividually, creating a local matrix. These local matrices are added to the global matrix for all nodes in the mesh and the resulting linear system is then solved numerically.

Calculating the E field

To obtain the electric field, $\vec{E} = -\nabla\phi$, one could apply a numerical gradient to the solution vector, Φ. However, through this process, the order of the solution is then reduced (for example, we work with second order basis functions, so

* vrielink@stanford.edu

ISBN 978-3-95450-182-3

$\vec{E}$ calculated this way would then be piecewise linear). To avoid this, a weak form of $\vec{E} = -\nabla\phi$ is used and a second variation performed, producing $\vec{E}$ to the same order as ϕ. This step does not add significant overhead in terms of computation time as the matrices associated with this step can be calculated once, after meshing, and reused for various problems on the same mesh.

Sample output from our electrostatic solver is shown in Fig. 1 for a typical DC gun. While this gun is of course azimuthally symmetric ($m = 0$), we show the case where voltages with $m = 2$ azimuthal symmetry are applied as well to demonstrate how the potential and fields change.

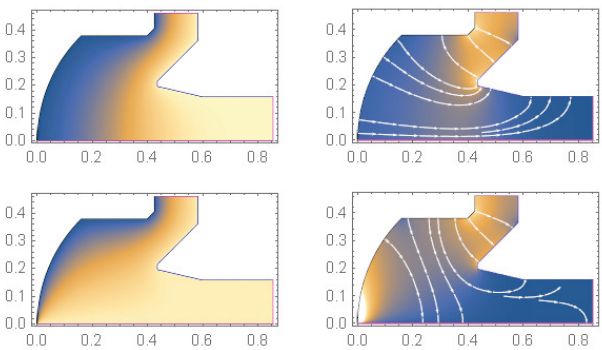

Figure 1: The electrostatic potential, ϕ (left) and the electric field $\vec{E}$ (right) for a DC gun, as obtained from our solver. The $m = 0$ case is on top and the $m = 2$ case is on the bottom.

EIGENMODE SOLVER

In a similar manner to the electrostatic solver, our steady state, source free eigenmode solver uses the Lagrangian formulation for the electromagnetic potentials except we now include $\vec{A}$, such that our variational form is given by Eq.(6). Note this analysis is conducted in the frequency domain, so ϕ and $\vec{A}$ are complex spectral amplitudes here. The second term is needed to impose arbitrary boundary conditions (Y is a dyadic related to the conductance on the boundary). Without it, only the natural boundary condition of a magnetic wall is enforced.

$$\mathcal{L} = \frac{\epsilon}{2}\int |\nabla\phi + i\omega\vec{A}|^2 - c^2|\nabla\times\vec{A}|^2\, dv + \frac{1}{i\omega}\int (-\nabla\phi - i\omega\vec{A})\cdot Y(-\nabla\phi^* + i\omega\vec{A}^*)\, dS \quad (6)$$

In implementing the eigenmode solver, we considered first the case where $m = 0$. While too lengthy to include here, when the $e^{im\theta}$ dependence is substituted into the Lagrangian, it splits into terms that depend only on A_θ and terms where A_r, A_z and ϕ are coupled to each other. This split corresponds to transverse electric (TE) modes when $A_r = A_z = \phi = 0$ and transverse magnetic (TM) modes when $A_\theta = 0$. While it is possible to solve for the TE modes first and then derive the TM modes from these, we implemented both as solving for the TM case directly allowed us to develop a better understanding of the issue of gauge fixing and spurious modes [1]. This was crucial before we could move to $m > 0$. The TE modes do not require any gauge fixing as the additional degrees of freedom are all set to zero.

In either case, the Lagrangian could once again be written in the form of Eq.(3), except now our discretized Lagrangian matrix operator, $\mathbf{L}$ depends also on ω:

$$\mathbf{L} = \mathbf{M}\omega^2 + \mathbf{C}\omega^2 + \mathbf{K} \quad (7)$$

The resulting variation gives a quadratic generalized eigenvalue problem, however if the Coulomb or Lorenz gauge are enforced and appropriate basis functions used (see [1]), it can be reduced to a linear generalized eigenvalue problem (GEP). Either can be solved numerically, however the quadratic problem is essentially twice as costly in terms of computation time and memory. It can also be less numerically stable than the linear GEP, hence the motivation to work with the latter when possible.

Sample output from our eigenmode solver is shown in Fig. 2 for a 1mm x 1mm pillbox cavity.

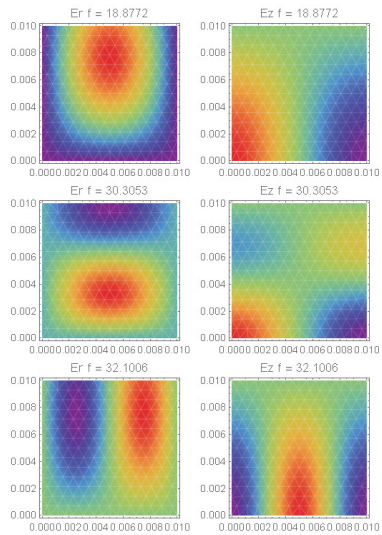

Figure 2: Results of the eigenmode solver showing the $\vec{E}$ field for select TM modes of 1mm x 1mm pillbox cavity.

IMPLEMENTATION

The computational implementation of this FEM formulation was done in C/C++. Included in the current implementation is a user interface through mathematica, a custom mesher, functions to calculate the local matrices for the electrostatic solver (m arbitrary) and the eigenmode solver ($m = 0$ only), and a function to assemble and solve the global matrix. The $\vec{E}$ and $\vec{B}$ field solvers that solve the weak formulation of the differential equations with respect to the potentials ($\vec{E} = -\nabla\phi - i\omega\vec{A}$ and $\vec{B} = \nabla\times\vec{A}$) are in the process of being implemented. Below we discuss some of these features in more detail.

Mathematica Interface

The user interface is currently through Mathematica. The outer hull geometry can be constructed in Mathematica and the boundary conditions defined. Through the WSLINK feature in Mathematica a connection between a C program and the Mathematica run time can be established such that function calls and data can be transferred to and from a Mathematica notebook.

Mesher

A custom mesher is used which maps the user defined outer hull to a grid over which triangular elements are created to fill the hull. These triangles are iteratively transformed via rotations, splitting, and inverting end points of adjacent triangles until the minimum angle is above a user defined tolerance. The end result is a fairly uniform mesh with no triangles having any extreme aspect ratios, necessary for finite element analysis.

The final mesh is not an optimal mesh in the sense of a Delaunay triangulation, but it does get close and in most cases is much faster, requiring usually only 4-5 iterations to achieve a minimum angle of 20°, for example. That being said, the number of iterations will depend heavily on the initial grid size used. A sample mesh demonstrating a few iterations is shown in Fig. 3.

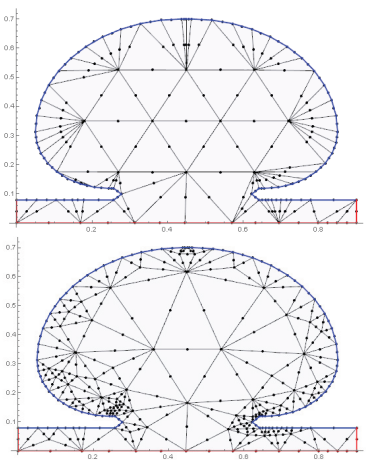

Figure 3: The inital mesh vs final mesh after triangle perturbations. The grid size was set exceptionally large to demonstrate the ability to handle high aspect ratio triangles.

Local Matrix Calculation

The integrals for the local matrices are calculated analytically in Mathematica for our 2nd order triangular nodal basis functions. The result is a $4d \times 4d$ matrix with expressions that are functions of the nodal coordinates, (r_i, z_i). Here d is the number of nodes per element (6 in our case) and the 4 comes from the fact that we are solving for ϕ and three components of $\vec{A}$. In the electrostatics solver and the TE mode solver, the matrix size is reduced to a $d \times d$ matrix.

Some of the local matrix entries are quite complex, particularly for $m \neq 0$, and can reach up to 60-100 operations per entry if left in the original output form in Mathematica. We have thus spent effort devising an optimization algorithm that reduces the number of operations via substitutions of commonly encountered terms in the matrix entries. These terms can be calculated at the beginning and reused, reducing the total number of operations significantly. The local matrix entries are then ported to the c code.

Global Matrix Assembly and Solution

After creating an adjacency matrix from the mesh, the global matrix is assembled by traversing each triangle, calculating the local matrix for that element, and adding the values to the appropriate entry in the global matrix according to the adjacency matrix.

The solution is then calculated using the Intel MKL libraries. For the linear system in the electrostatic problem, we use the Paradiso solver. For the eigenmode solver, the FEAST algorithm is used to solve the generalized eigenvalue problem. The FEAST algorithm computes a subset of the eigenvalues using a contour integration in the complex plane [2]. We extended this algorithm to work for quadratic eigenvalue problems as well, however this is only implemented in Mathematica and not in the *c* code.

CONCLUSION AND FUTURE WORK

The theoretical formulation and initial groundwork for a 2D finite element solver for electromagnetic fields with m-fold azimuthal symmetry has been presented. We have described the theory and some implementation details for an electrostatics solver and an $m = 0$ eigenmode solver based on this formulation and some sample results presented.

The project is a work in progress and there is much to be done still. The immediate next step is the implementation of the $m \neq 0$ eigenmode solver. The next step will then be to add the source terms to the formulation allowing us to solve driven problems. In conjunction with this work, we are developing a particle tracking code to be used with this EM solver, allowing us to iteratively solve for the fields and the particle trajectories until we converge on the correct solution for both.

Finally, an in-depth study of the performance of the solver is planned for the future, including convergence characteristics, timing and memory usage.

REFERENCES

[1] A. Vrielink, M. Nasr and S. Tantawi, "Derivation of a Finite Element Formulation From a Lagrangian for the Electromagnetic Potentials," presented at the 8th Int. Particle Accelerator Conf. (IPAC'17), Copenhagen, Denmark, May 2017, paper WEPIK111, this conference.

[2] E. Polizzi, "Density-Matrix-Based Algorithms for Solving Eigenvalue Problems," *Phys. Rev. B.* , vol. 79, p. 115112, 2009.

ISBN 978-3-95450-182-3