OpenParEM2D

<u>Open</u> <u>Par</u>allel <u>E</u>lectro<u>m</u>agnetic <u>2D</u>

A free, open-source electromagnetic simulator for
2D waveguides and transmission lines.

Brian Young
Independent Software Developer

Motivation

- 3D full-wave electromagnetic (EM) simulators producing frequency-dependent scattering parameters (S-parameters)
  - General microwave engineering
  - High-speed digital signal integrity
  - Antenna design
- Expensive
  - ~$35K/year to *lease*
  - Hurts innovation
    - Individuals cannot explore ideas on their own
    - Startups struggle with funding [can lead to piracy]
    - Certain business models are impractical [ex. cloud-based modeling services]
- Commercial capabilities are stagnant
  - Leading tools are used the same way now as for the past 30 years. Maxwell's equations have not changed.
- An open-source alternative is needed.

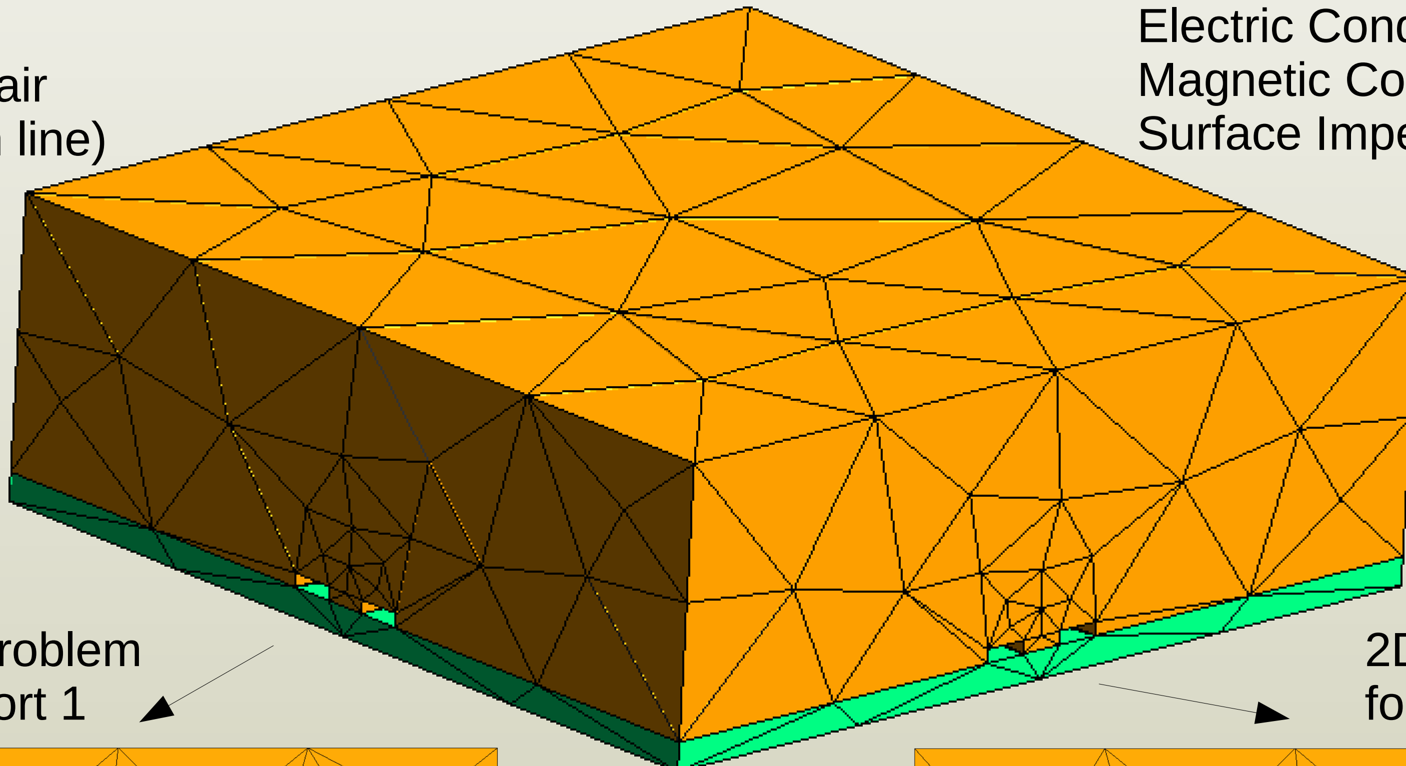
How?

- The building blocks needed for a fully-capable 3D full-wave EM simulator are available as free and open source.
- Just need to glue the pieces together
  - A manageable programming exercise for one person or a very small team.
- 1^{st} step
  - 2D solution of the ports to set up the boundary conditions for the 3D problem
  - This work.
- 2^{nd} step
  - Setup and solve the 3D boundary value problem
  - In development.

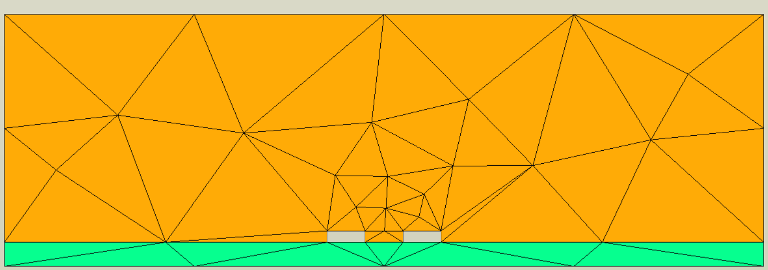
3D → 2D Breakdown

3D Problem
(90º bend of
differential pair
transmission line)

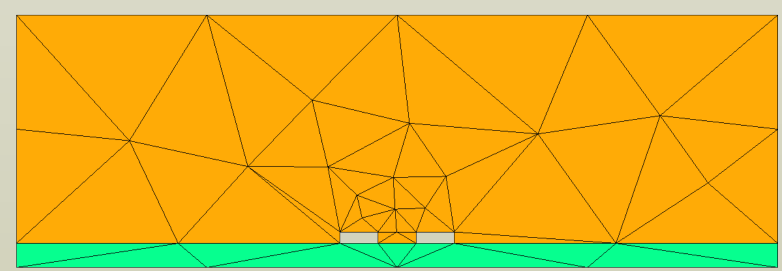
Other surfaces are Perfect
Electric Conductor, Perfect
Magnetic Conductor, or
Surface Impedance.



2D Problem
for Port 1



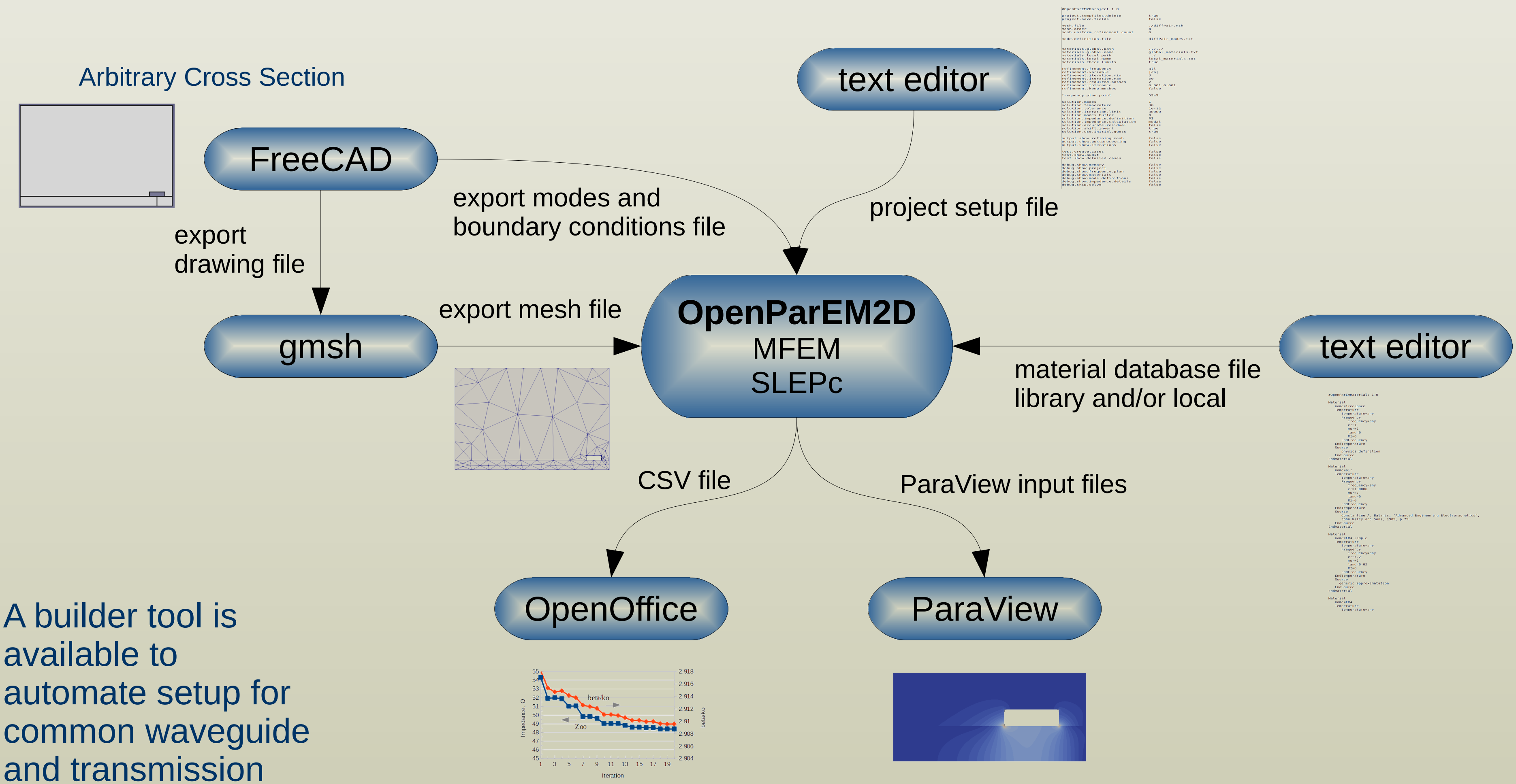
2D Problem
for Port 2



- 2D solutions solve for the EM fields, propagation constants, and characteristic impedances.

Workflow for 2D Ports

- OpenParEM2D is a command-line tool with text inputs and outputs
- Use with open source tools to create a complete workflow
- Workflow used for development:



A builder tool is available to automate setup for common waveguide and transmission line types.

OpenParEM2D

Open Parallel Electromagnetic 2D - A free, open-source electromagnetic simulator for 2D waveguides and transmission lines.

Features

MFEM

- Full-wave solver
  - simultaneously solves the electric and magnetic fields
- Advanced finite-element method (FEM) with arbitrary high-order elements
- Adaptive mesh refinement
- Parallel processing through the Message Passing Interface (MPI)
- Front-end input file builder for common transmission line and waveguide types.
- Licensed under GPLv3 or later.

MFEM, PETSc, SLEPc, OpenParEM2D

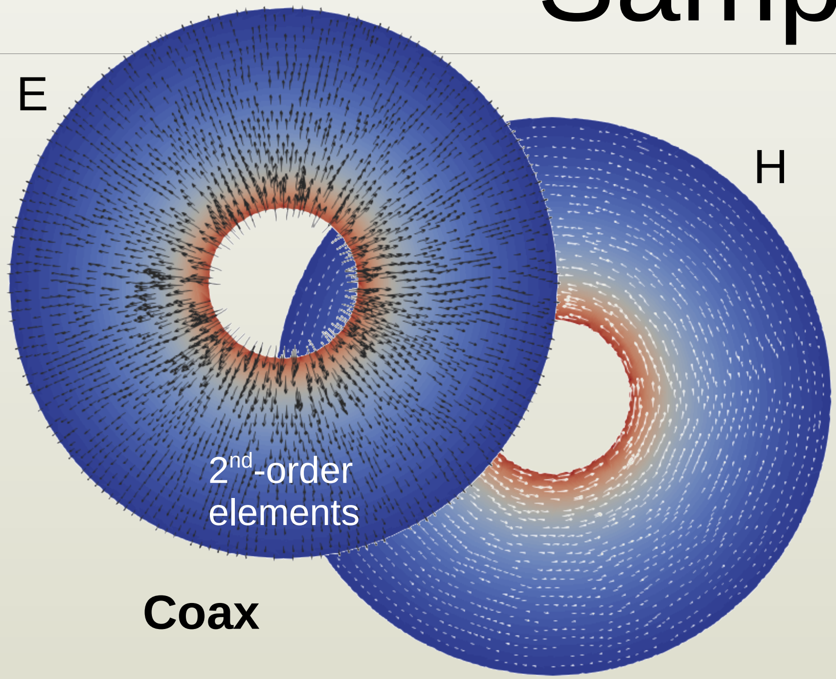
Capabilities

Field-based Calculations

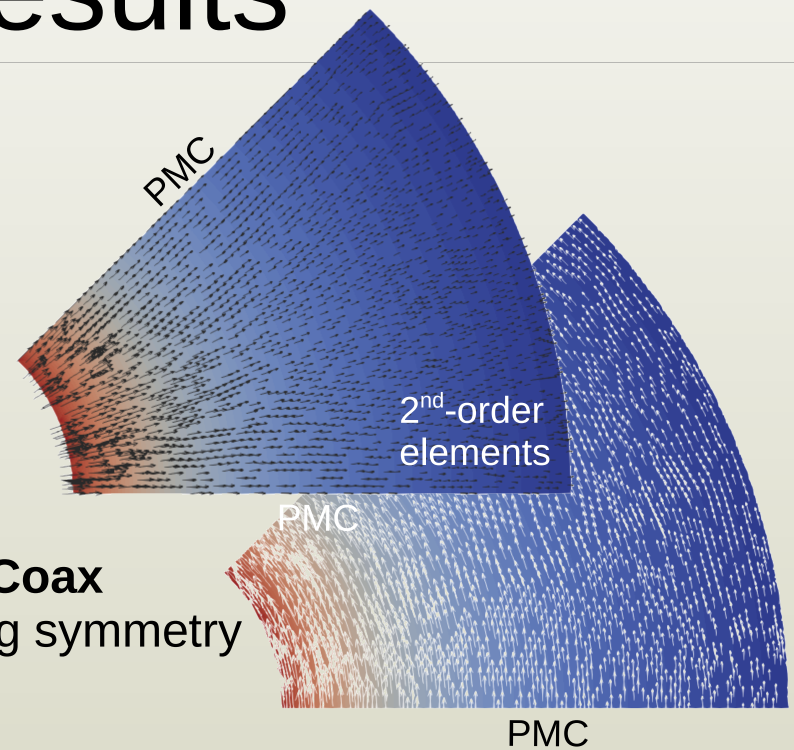
- Calculations
  - propagation constant
  - characteristic impedance **MFEM**
  - dielectric loss
  - conductor loss
  - surface roughness loss
  - field distributions

  MFEM
- Dominant and higher-order modes
- Arbitrary cross sections
- Arbitrary high frequencies

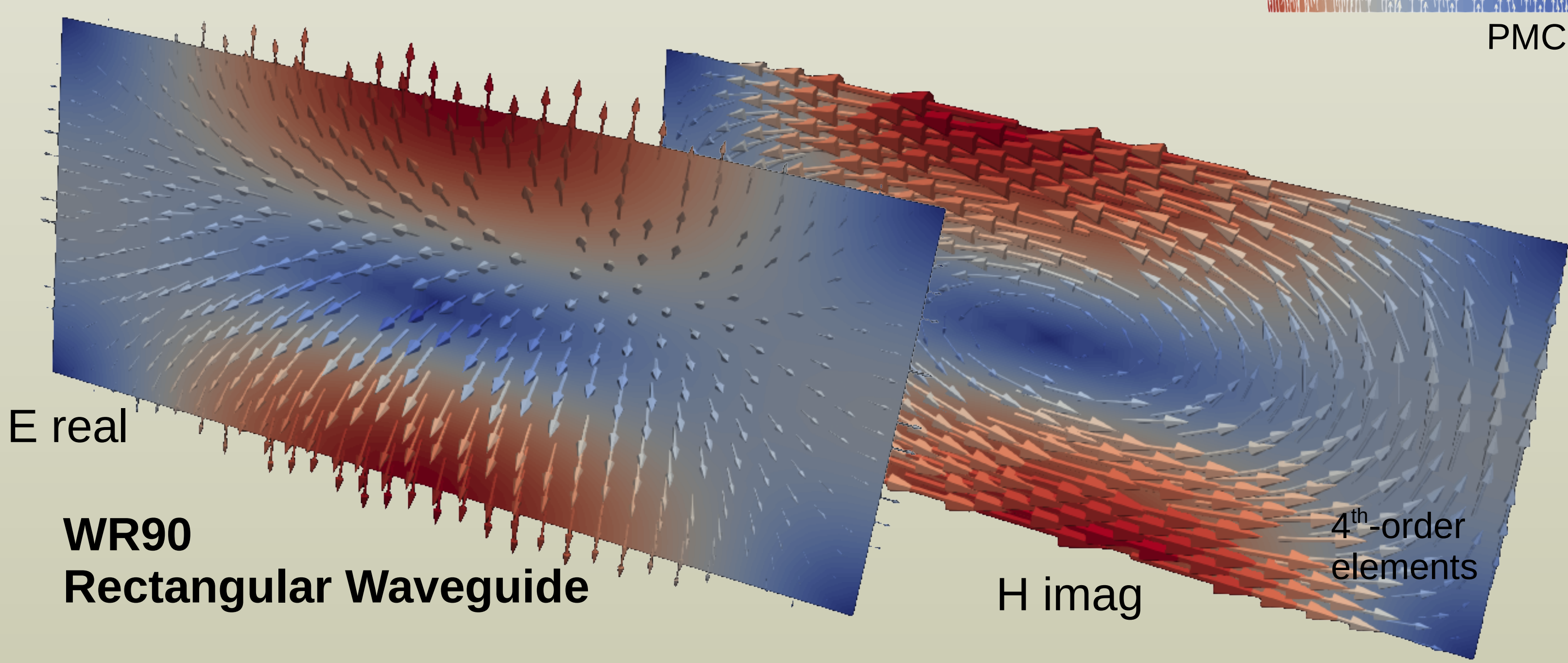
Sample Results



Coax

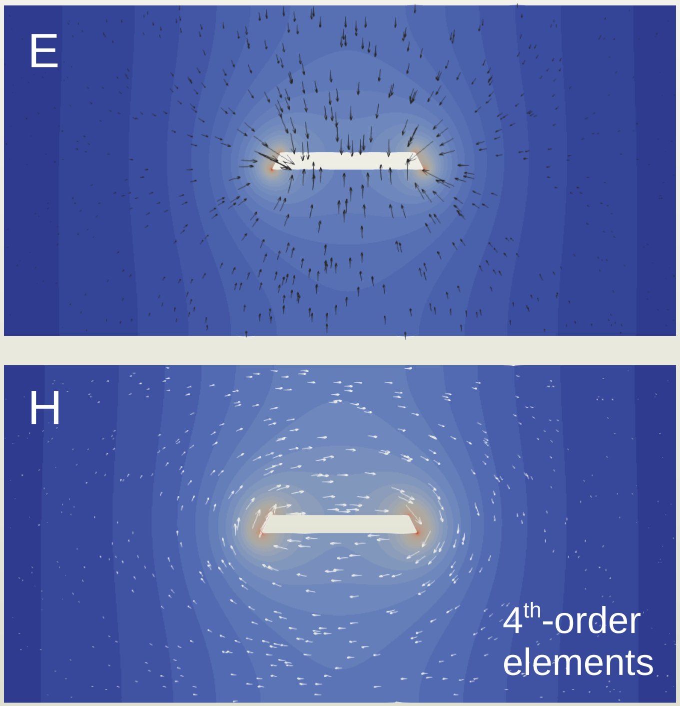


1/8 Coax
using symmetry

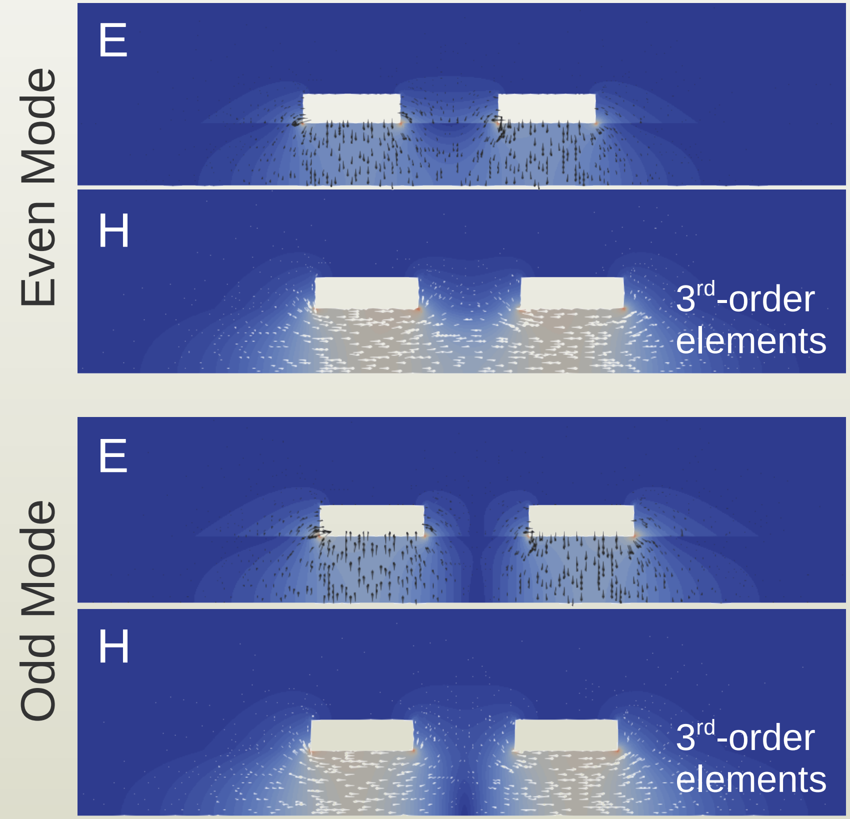


WR90
Rectangular Waveguide

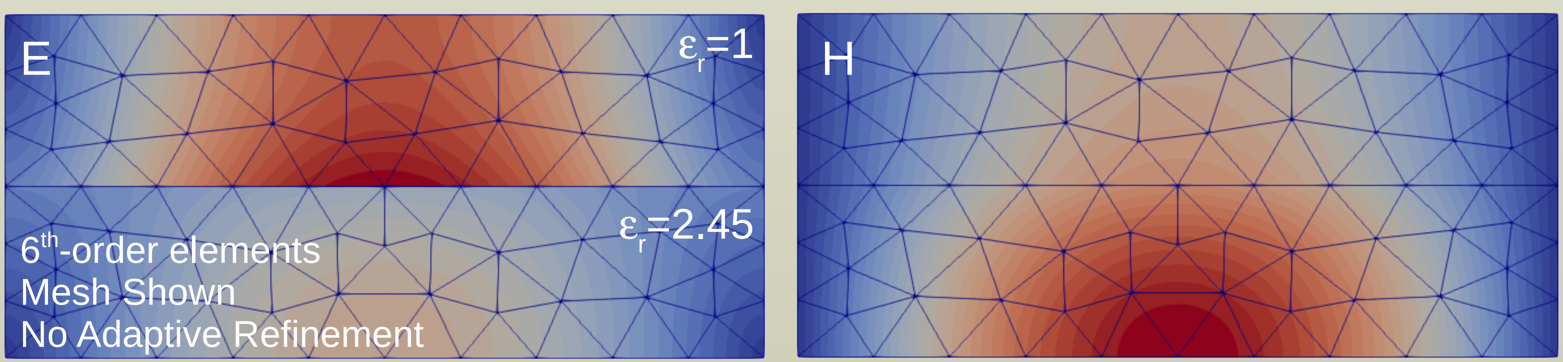
Cutoff
TE11 Mode



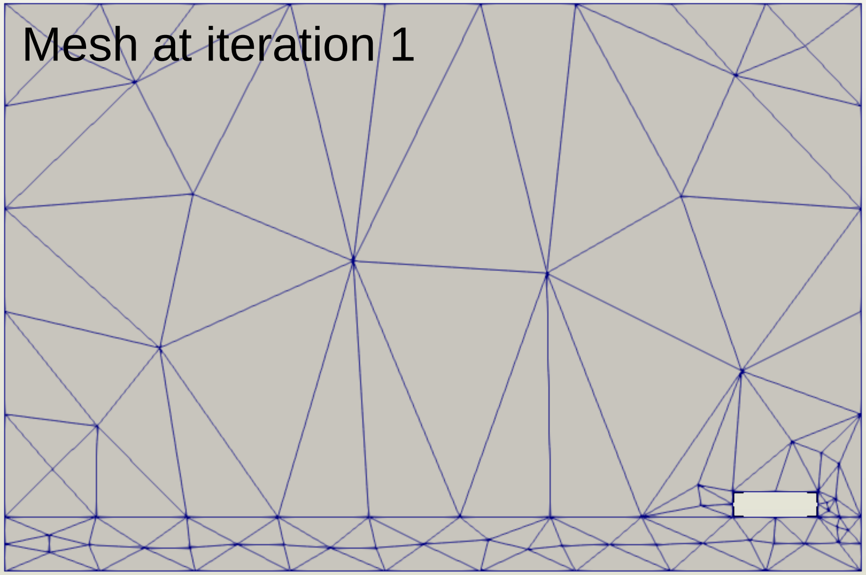
Stripline
with etch profile



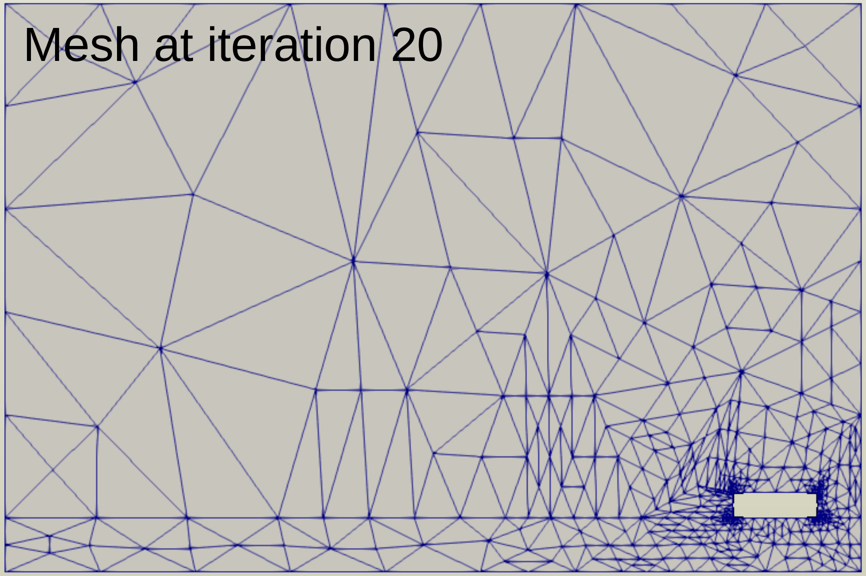
Coupled Microstrip



Partially-Filled Rectangular Waveguide

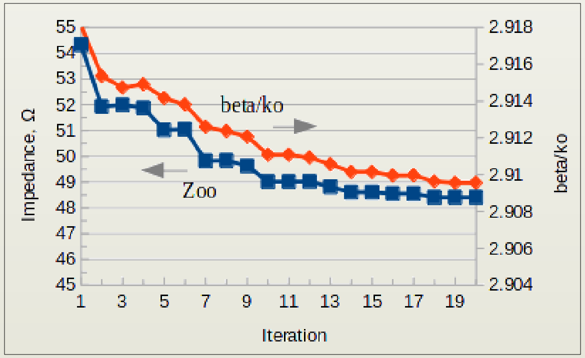


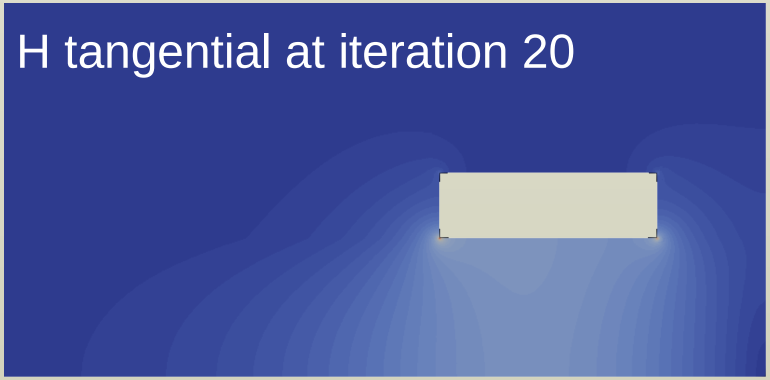
PEC



PEC

4th-order elements

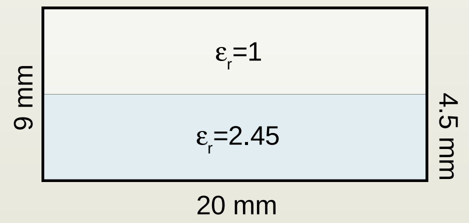


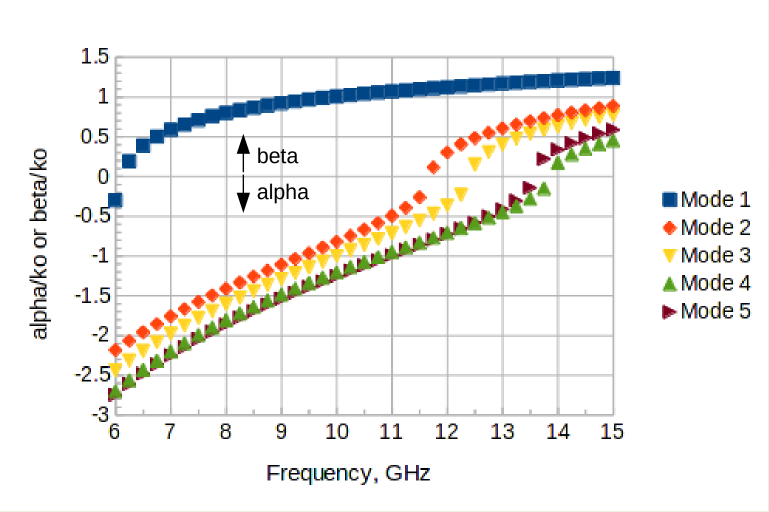


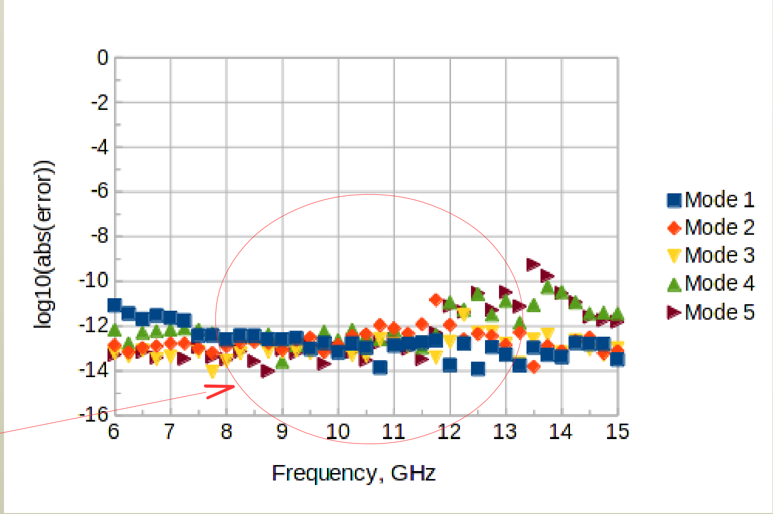
Odd Mode of Coupled Microstrip

Example Accuracy: Partially-Filled Waveguide

- Partially-filled rectangular waveguide
  - First 5 modes
  - Roger E. Harrington, *Time-Harmonic Electromagnetic Fields*, McGraw-Hill, 1961, p. 161, example from Fig. 4-7.
    - Dimensioned figure in upper right
    - Exact solution is from numerical solution of the transcendental equations in (4-56) and (4-58).
  - regression/partially_filled_rect_waveguide/PartFilled_order_6_refinement/PartFilled_accuracy_run.proj
- accuracy settings
  - `mesh.order` `6`
  - `mesh.uniform_refinement.count` `0`
  - `refinement.frequency` `all`
  - `refinement.variable` `|gamma|`
  - `refinement.required.passes` `2`
  - `refinement.tolerance` `1e-8`
  - `solution.tolerance` `1e-13`
  - `solution.shift.invert` `true`
  - `solution.shift.factor` `10`
- The propagation/attenuation plot in the middle right shows the exact analytical results. The computed results are not shown since they overlay the exact results.
- The error plot in the lower right shows the error between the analytical and computed results for the propagation/attenuation constants.
- The baseline error is about 10^{-12} with an expected increase in the error as the eigenvalue transitions through zero.







MFEM enables high accuracy near absolute numerical limits.

MFEM Challenges For This Project

1. MFEM is either vector 2D or 3D.
   - The full-wave EM waveguide/transmission line problem requires 3D vector fields restricted to a 2D plane.
   - The 3D vector problem is separated into transverse field components (vector in x-y) and longitudinal components (scalar in z).
     - Z-dependence assumes for $e^{-\gamma z}$ wave propagation, where γ is the complex propagation constant.
2. Frequency-domain solution requires complex numbers.
   - MFEM 4.3 has a start at complex support, but it was not used here.
3. Mesh de-refinement not supported in MFEM 4.3.
4. Steep learning curve.

1. Handling the 3D Vector in 2D

- 3D reduction to a 2D surface follows Lee's paper:
  - Jin-Fa Lee, “Finite Element Analysis of Lossy Dielectric Waveguides”, *IEEE Tran. Microwave Theory and Techniques*, vol. 42, no. 6, June 1994, pp. 1025-1031.
- Results in a generalized eigenvalue problem in complex E_t, E_z, and γ.

$$\sum_{\Omega_e}\left\{\begin{bmatrix}\frac{1}{\mu_r}[S_t]_e-k_o^2\bar{\epsilon}_r[T_t]_e & 0\\ 0 & 0\end{bmatrix}\right\}\begin{bmatrix}\bar{E}_t\\ \bar{E}_z\end{bmatrix}=\gamma^2\sum_{\Omega_e}\left\{\begin{bmatrix}\frac{1}{\mu_r}[T_t]_e & \frac{1}{\mu_r}[G]_e\\ \frac{1}{\mu_r}[G]_e^T & \frac{1}{\mu_r}[S_z]_e-k_o^2\bar{\epsilon}_r[T_z]_e\end{bmatrix}\right\}\begin{bmatrix}\bar{E}_t\\ \bar{E}_z\end{bmatrix} \qquad (12)$$

where

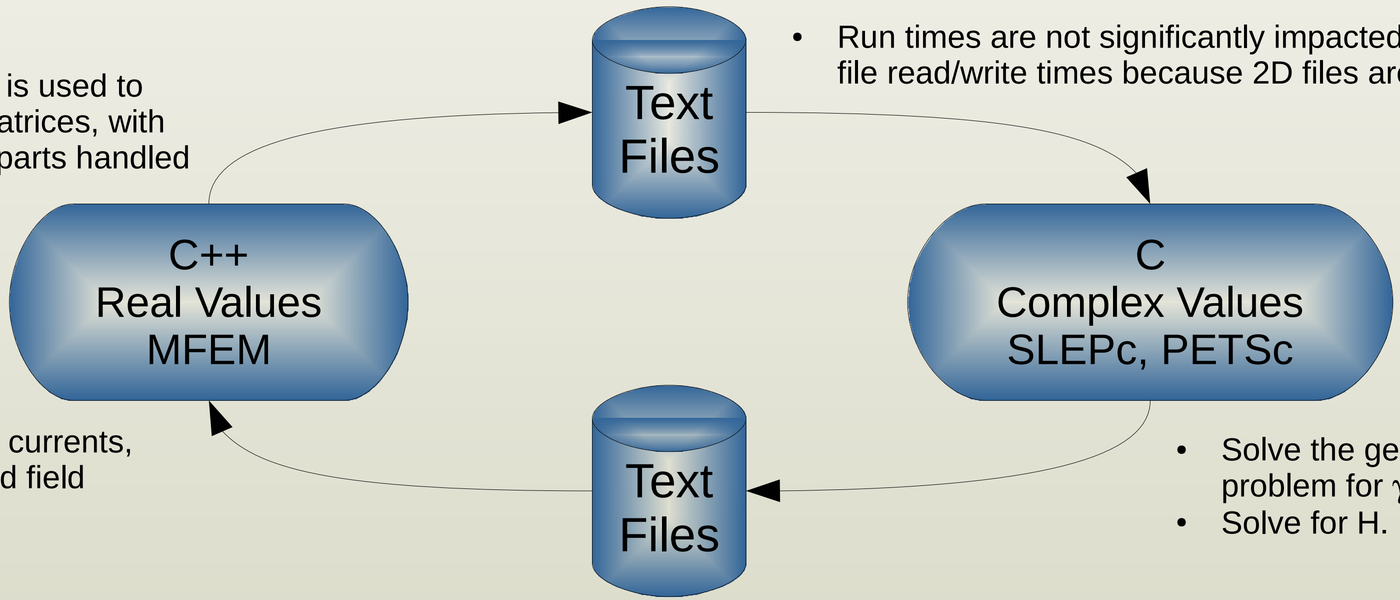
$$\begin{aligned}
\underline{W}_t[T_t]_e\bar{E}_t &= \int_{\Omega_e}\vec{w}_t\cdot\vec{e}_t\,d\Omega && \text{VectorFEMassIntegrator}\\
\underline{W}_t[G]_e\bar{E}_z &= \int_{\Omega_e}\vec{w}_t\cdot\nabla_t e_z\,d\Omega && \text{MixedVectorGradientIntegrator}\\
\underline{W}_z[S_z]_e\bar{E}_z &= \int_{\Omega_e}\nabla_t w_z\cdot\nabla_t e_z\,d\Omega && \text{DiffusionIntegrator}\\
\underline{W}_z[T_z]_e\bar{E}_z &= \int_{\Omega_e}w_z e_z\,d\Omega && \text{MassIntegrator}\\
\underline{W}_t[S_t]_e\bar{E}_t &= \int_{\Omega_e}(\nabla_t\times\vec{w}_t)\cdot(\nabla_t\times\vec{e}_t)\,d\Omega && \text{CurlCurlIntegrator}
\end{aligned} \qquad (14)$$

MFEM Integrators

- E_t is computed on a vector 2D finite element space using Nedelec finite elements.
- E_z is computed on a scalar 2D finite element space using L2 finite elements.
- H is similarly broken into H_t and H_z and solved from E using 2 complex Ax=b solves.
- Other calculations for currents, voltages, and propagating power use numerical integration or MFEM integrators with the real and imaginary parts separately handled.

2. Handling Complex Numbers

- Real-valued MFEM is used to build the needed matrices, with real and imaginary parts handled separately.
- Run times are not significantly impacted by the file read/write times because 2D files are small.



- Post-processing for currents, voltages, power, and field distribution.
- Solve the generalized eigenvalue problem for γ and E.
- Solve for H.

- Real math is handled in C++.
  - PETSc is compiled for real numbers.
- Complex math is handled in C.
  - PETSc is compiled for complex numbers.
- The C++ → C → C++ operation is transparent to the user.
  - C is called from C++ resulting in one executable.

4. Steep Learning Curve

- The examples are numerous and well done.
  - Can be tough to work out the intent of the code.
  - Examples in areas of physics outside of personal expertise can be tough to decipher.
- Many methods in Doxygen have no descriptions.
- What would have helped on this project:
  - General descriptions of basic ways to get things done without using specifically-coded examples or terminology specific to one branch of physics.
    - Examples:
      - How to multiply two vector fields.
      - How to integrate a scalar over a cross section.
    - LLNL provided the needed descriptions as support emails.
      - Thanks!

Open Source Project

- https://www.openparem.org
  - Full documentation
  - Link to github
- Current implementation is ~27,000 lines of code
- Regression suite
  - 21,586 individual tests across 27 cases
    - 4 cases compare with analytically exact results.
    - 2 cases compare to the literature.
    - Complex propagation constants, impedances, losses, and field values at points in the cross sections
  - Also serves as an extensive set of worked examples.

Conclusion

- A full-wave 2D EM simulator based on MFEM finite elements is reviewed.
  - A 2D simulator is useful for designing transmission lines and waveguides.
  - The expected primary use is in solving 3D EM problems.
- Highlights for this project
  - High accuracy on general full-wave waveguide and transmission line problems with MFEM finite elements.
  - Flexibility of MFEM to handle a 3D vector problem restricted to a 2D plane.
  - Details on solving a complex-valued problem with real-valued MFEM.
  - A documentation strategy that could potentially help with the learning curve.

Acknowledgement

- Big thanks to LLNL's Mark Stowell for many email exchanges breaking several logjams I encountered along the way.
- I doubt I would have successfully completed the 2D solver without his help.