OpenParEM2D

Impedance Guide

Overview

- Characteristic impedance is a very important design target and a large reason for the existence of 2D electromagnetic (EM) simulators.
- However, characteristic impedance is not a fundamental quantity in EM.
  - There are many definitions combining voltage, current, and power.
- The user must use a definition for characteristic impedance that is useful for the problem at hand.
  - The best definition is the one that is most useful in quantifying reflections for a particular transmission line or waveguide used in a specific system.
- OpenParEM2D is a general full-wave 2D EM solver, so it is capable of calculating characteristic impedance using any combination of voltage, current, and power.
- This presentation covers the general setups in detail.

Characteristic Impedance Definitions

- Symbol definitions
  - V - voltage
  - I - current
  - P - average power propagation
  - Z_o - characteristic impedance
- Basic relationships
  - $Z_o=V/I$
  - $P=1/2VI^*$
- Combining these leads to 3 definitions for Z_o

  (a) $Z_o=V/I$

  (b) $Z_o=2P/|I|^2$

  (c) $Z_o=1/2|V|^2/P$
- It is up to the user to decide which definition is best to use for the problem at hand
  - In most cases, there is a conventional selection that is made for a given type of transmission line or waveguide.
- The selection is made in the setup file as
  - `solution.impedance.definition   string`
- The string is “VI” for definition (a), “PI” for (b), or “PV” for (c). “none” indicates that no Z_o calculation is to be made.
- With the Zo definition selected, the user must properly set up either the voltage or current calculation, or both.
  - Note that P is a fundamental EM quantity and is automatically calculated without user input.

Voltage Setup

- Voltage is calculated as a line integral of the electric field along a path.
  - OpenParEM2D calculates the line integral from the computed electric field given the integration path supplied by the user.
  - The details of adding paths are covered in the user manual and the specifications document in the section on “Boundary/Mode File Specification”.
- Generally, a path will start on one conductor and end on another.
- OpenParEM2D does not provide any checks on whether a voltage path is set up correctly.
  - A path can start anywhere and end anywhere.
  - It will warn if the path falls outside of the physical space of the problem.

Example

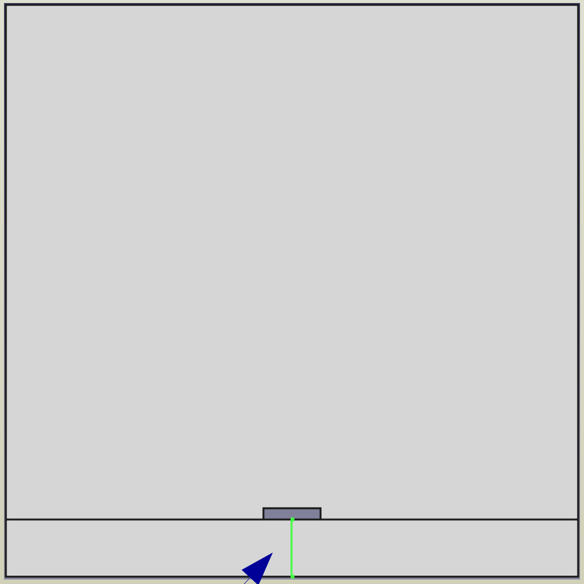
- The boundary/modes file for the example regression/Lee_microstrip is shown to the right.
- This is a microstrip, so the voltage line goes from the ground plane to the center conductor.
  - The middle of the center conductor is best since the fields are smoothest there.

```
#OpenParEMmodes 1.0

File
   name=Lee_microstrip.FCStd
EndFile

Path
   name=V1
   point=(0.00125,0.0)
   point=(0.00125,0.00025)
   closed=false
EndPath

Path
   name=line1
   point=(0.001125,0.00025)
   point=(0.001375,0.00025)
   point=(0.001375,0.0003)
   point=(0.001125,0.0003)
   closed=true
EndPath

Mode
   mode=1
   type=voltage
   path=V1
EndMode

Mode
   mode=1
   type=current
   path=line1
EndMode
```

path describing the voltage integration line

assigning the path to mode 1 to calculate the line integral

- OpenParEM2D will solve for the tangential electric fields then integrate along the path V1 to find the voltage. It will assign the voltage to mode 1 for calculating characteristic impedance.



voltage integration line

Current Setup

- Current is calculated directly from Ampere's law as a closed line integral around a conductor.
  - Similar to the voltage discussion, OpenParEM2D calculates the line integral from the solved magnetic field using a path provided by the user.
- Generally, a current path is set up to go around a conductor.
- OpenParEM2D checks for closed paths and warns if points fall outside of the physical cross section.

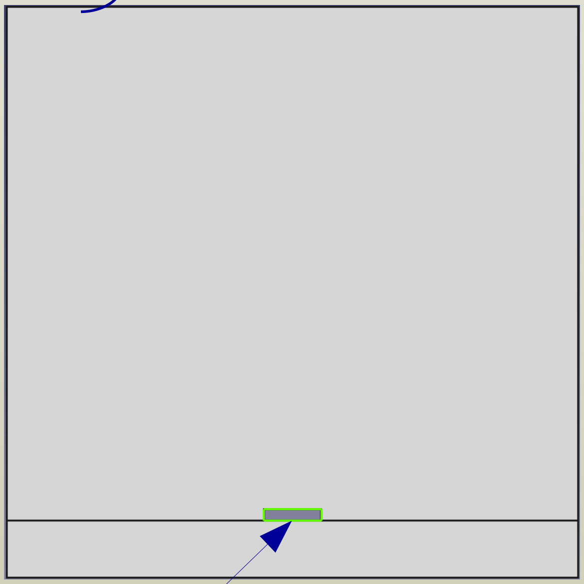
Example

- The boundary/modes file for the example regression/Lee_microstrip is shown to the right.
- This is a microstrip, so the closed current path goes around the center conductor.
  - The path crosses the corners where the fields are not smooth, so the current is slow to converge.
- OpenParEM2D will solve for the tangential magnetic fields then integrate along the closed path line 1 to find the current, then it will assign the current to mode 1 for calculating characteristic impedance.

```
#OpenParEMmodes 1.0

File
   name=Lee_microstrip.FCStd
EndFile

Path
   name=V1
   point=(0.00125,0.0)
   point=(0.00125,0.00025)
   closed=false
EndPath

Path
   name=line1
   point=(0.001125,0.00025)
   point=(0.001375,0.00025)
   point=(0.001375,0.0003)
   point=(0.001125,0.0003)
   closed=true
EndPath

Mode
   mode=1
   type=voltage
   path=V1
EndMode

Mode
   mode=1
   type=current
   path=line1
EndMode
```

path describing the current integration line

assigning the path to mode 1 to calculate the line integral



Multi-Mode Setup

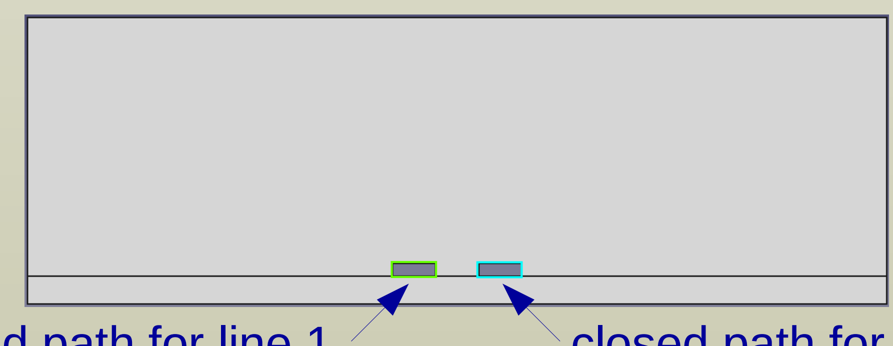
- When OpenParEM2D is solving for more than one mode, the user must make more decisions regarding the setup.
- Being a full-wave EM 2D simulator, fundamentally, the computed solutions are the modes of the transmission line or waveguide.
  - All voltage and current calculations are made on the modes.
  - Modes are orthogonal, so there is no coupling between them.
- OpenParEM2D provides two ways to set up impedance calculations.
  - modal
    - The voltage and/or current paths are set up to match the field configurations of the each mode.
    - Since the modes are orthogonal, this setup results in one impedance per mode.
    - This setup can be very challenging and may be of interest to those interested in exploring modes in detail.
  - line
    - The voltage and/or current paths are set up uniformly across conductors irrespective of field configurations.
    - The impedance calculation results in a matrix result.
    - This setup is relatively simple and likely the method of choice for most users.
- For a single conductor, the modal and line setups are identical.
- The two methodologies are covered in detail below.

Multi-Mode Setup: Modal

- The modal setup follows the pure electromagnetic definition for modes.
- When OpenParEM2D is asked to solve for N modes, the N modes are calculated and stored in N data structures.
  - The modes are orthogonal.
- The user can provide up to N Z_o definitions.
  - The setups are differentiated by mode number.
    - The user must align the definition to the correct mode number.
  - OpenParEM2D simply then uses the path and Z_o definition per mode to calculate Z_o for each mode.
  - If a definition for a mode is missing, then OpenParEM2D simply skips the impedance calculation.
- The modal calculation is called for by the following line in the setup file:
  - `solution.impedance.calculation  modal`

2-mode modal example - Z_o using current

- The boundary/modes file simplified for currents for the example regression/differential_pair/diff_pair_modal is shown to the right.
- This is a coupled microstrip, so there are two modes.
  - common mode - current travels down both conductors and returns on the ground plane
  - differential mode - current travels down one conductor and returns on the other
- The current paths must be set up to reflect how the modes use the conductors.
  - "Easy" for 2 conductors. Challenging, otherwise.
- Current paths are set up for each microstrip.
- Mode 1 definition - the common mode - current uses both conductors, so the total current path uses both paths on lines 1 and 2.
- Mode 2 definition - the differential mode - current travels down line 1, so a current path is defined. Current returns on line 2, so the path around line 2 is not included.
  - Lines 1 and 2 can be flipped with no change in the computed solution.




```
#OpenParEMmodes 1.0

File
   name=diffPair.FCStd
EndFile

Path
   name=line1
   point=(-0.0015,0.000635)
   point=(-0.0005,0.000635)
   point=(-0.0005,0.000935)
   point=(-0.0015,0.000935)
   closed=true
EndPath
```

closed path for line 1

```
Path
   name=line2
   point=(0.0005,0.000635)
   point=(0.0015,0.000635)
   point=(0.0015,0.000935)
   point=(0.0005,0.000935)
   closed=true
EndPath
```

closed path for line 2

```
Mode
   mode=1
   type=current
   path=line1
   path+=line2
EndMode
```

mode 1 current is on both line 1 and line 2

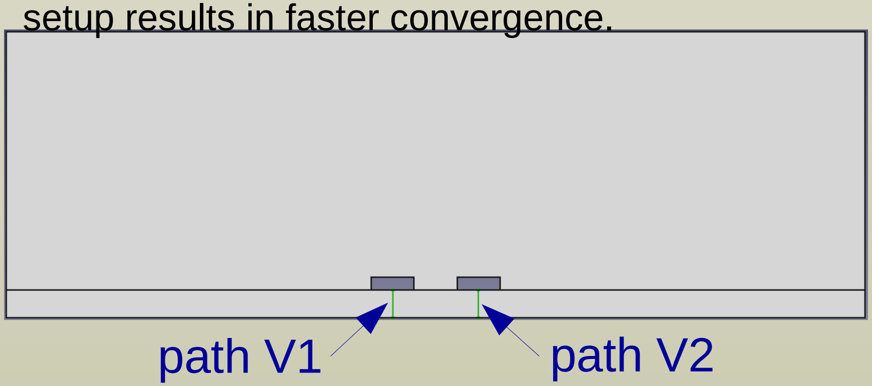
```
Mode
   mode=2
   type=current
   path=line1
EndMode
```

mode 2 current is on line 1, with return current on line 2

- Setup
  - `mesh.order` 5
  - `refinement.required.passes` 3
  - `refinement.tolerance` 0.001
  - `solution.impedance.definition` PI
  - `solution.impedance.calculation` modal
  - `frequency.plan.point` 52e9
  - `solution.modes` 2
  - `solution.tolerance` 1e-12
- Results
  - Mode 1 - $Z_{o,common}$=27.63 Ω
  - Mode 2 - $Z_{o,differential}$=96.92 Ω
- Even and odd Z_o
  - $Z_{oe}=2*Z_{o,common}=2*27.63=55.26$ Ω
  - $Z_{oo}=1/2*Z_{o,differential}=96.92/2=48.46$ Ω

2-mode modal example - Z_o using voltage

- The boundary/modes file simplified for currents for the example regression/differential_pair/diff_pair_modal is shown to the right.
- This is a coupled microstrip, so there are two modes.
  - common mode - The voltages on the lines have the same magnitude and sign.
  - differential mode - The voltages on the lines have the same magnitude but opposite signs.
- The voltage paths must be set up to reflect how the modes reference the voltage.
  - “Easy” for 2 conductors. Challenging, otherwise.
- Mode 1 definition - the common mode - Use one path from one of the conductors to ground.
- Mode 2 definition - the differential mode - Use a path from line 1 to line 2. Each line uses the other as the ground reference.
  - An integration line directly between the two can be used, but the fields are rapidly varying due to the corners on the conductors.
  - Here, the integration path goes to the centers of the conductors. This relies on the path independence of voltage when the path is small compared to the wavelength. This setup results in faster convergence.




```
#OpenParEMmodes 1.0

File
   name=diffPair.FCStd
EndFile

Path
   name=V1
   point=(-0.001,0.000635)
   point=(-0.001,0.0)
   closed=false
EndPath

Path
   name=V2
   point=(0.001,0.000635)
   point=(0.001,0.0)
   closed=false
EndPath

Mode
   mode=1
   type=voltage
   path=V1
EndMode

Mode
   mode=2
   type=voltage
   path=V1
   path-=V2
EndMode
```

path V1

path V2

mode 1 voltage goes from line 1 to ground

mode 2 voltage goes from line 1 to line 2

- Setup
  - `mesh.order` 5
  - `refinement.required.passes` 3
  - `refinement.tolerance` 0.001
  - `solution.impedance.definition` PV
  - `solution.impedance.calculation` modal
  - `frequency.plan.point` 52e9
  - `solution.modes` 2
  - `solution.tolerance` 1e-12
- Results
  - Mode 1 - $Z_{o,common}=29.43\ \Omega$
  - Mode 2 - $Z_{o,differential}=134.54\ \Omega$
- Even and odd Z_o
  - $Z_{oe}=2*Z_{o,common}=2*29.43=58.86\ \Omega$
  - $Z_{oo}=1/2*Z_{o,differential}=134.54/2=67.27\ \Omega$

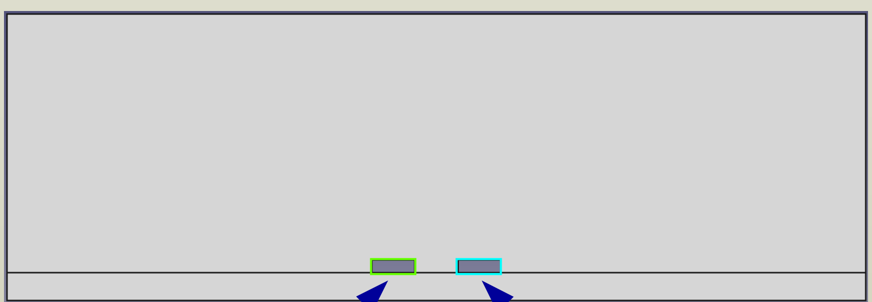
- Note that as expected, the impedance values are different between the PI and PV definitions. For microstrip and stripline transmission lines, the PI definition is generally preferred.

Multi-Mode Setup: Lines

- The lines setup defines voltages and currents based on the lines rather than on the modes.
- Since the setup does not follow the modes, an impedance matrix for the line impedances is produced.
- Note that OpenParEM2D is still solving for the modes. The basic EM solution is the same whether the impedance setup uses modal or lines definitions.
- The lines setup is far simpler than the modal setup since no understanding of the field structure or referencing is required.
- Setup
  - For currents, each line gets a closed loop around the conductor.
  - For voltages, each line gets a path from the conductor to ground.
- The line calculation is called for by the following line in the setup file:
  - `solution.impedance.calculation  line`

2-mode line example - Z_o using current

- The boundary/modes file simplified for currents for the example regression/differential_pair/diff_pair_line is shown to the right.
- This is a coupled microstrip, so there are two lines.
  - Each line gets a closed path around the conductor.
  - The Mode/EndMode block still uses the “mode=” keyword, but now the mode number is really referring to the line number. [This should probably be changed in the spec.]
- All of the lines are treated the same.




```
#OpenParEMmodes 1.0

File
   name=diffPair.FCStd
EndFile

Path
   name=line1
   point=(-0.0015,0.000635)
   point=(-0.0005,0.000635)
   point=(-0.0005,0.000935)
   point=(-0.0015,0.000935)
   closed=true
EndPath

Path
   name=line2
   point=(0.0005,0.000635)
   point=(0.0015,0.000635)
   point=(0.0015,0.000935)
   point=(0.0005,0.000935)
   closed=true
EndPath

Mode
   mode=1
   type=current
   path=line1
EndMode

Mode
   mode=2
   type=current
   path=line2
EndMode
```

closed path for line 1

closed path for line 2

mode 1 refers to line 1

mode 2 refers to line 2

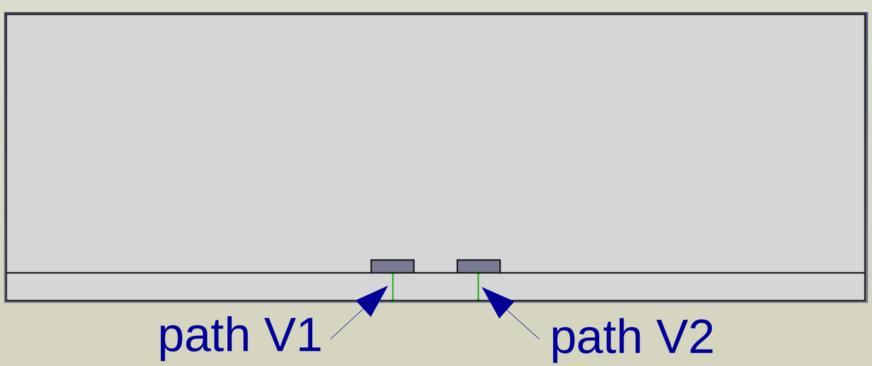
- Setup
  - `mesh.order` 5
  - `refinement.required.passes` 3
  - `refinement.tolerance` 0.001
  - `solution.impedance.definition` PI
  - `solution.impedance.calculation` line
  - `frequency.plan.point` 52e9
  - `solution.modes` 2
  - `solution.tolerance` 1e-12
- Results

$$Z = \begin{bmatrix} 55.25 & 0.00092 \\ 0.00092 & 48.46 \end{bmatrix}$$

- Diagonals are Z_{oe} and Z_{oo}
- Compare to the prior modal results
  - Z_{oe}=55.26 Ω
  - Z_{oo}=48.46 Ω
- Same answer

2-mode lines example - Z_o using voltage

- The boundary/modes file simplified for currents for the example regression/differential_pair/diff_pair_line is shown to the right.
- This is a coupled microstrip, so there are two lines.
  - Each line gets a closed path around the conductor.
  - The Mode/EndMode block still uses the “mode=” keyword, but now the mode number is really referring to the line number. [This should probably be changed in the spec.]
- All of the lines are treated the same.




```
#OpenParEMmodes 1.0

File
   name=diffPair.FCStd
EndFile

Path
   name=V1
   point=(-0.001,0.000635)
   point=(-0.001,0.0)
   closed=false
EndPath

Path
   name=V2
   point=(0.001,0.000635)
   point=(0.001,0.0)
   closed=false
EndPath

Mode
   mode=1
   type=voltage
   path=V1
EndMode

Mode
   mode=2
   type=voltage
   path=V2
EndMode
```

path V1

path V2

mode 1 voltage goes from line 1 to ground

mode 2 voltage goes from line 1 to ground

- Setup
  - `mesh.order` 5
  - `refinement.required.passes` 3
  - `refinement.tolerance` 0.001
  - `solution.impedance.definition` PV
  - `solution.impedance.calculation` line
  - `frequency.plan.point` 52e9
  - `solution.modes` 2
  - `solution.tolerance` 1e-12
- Results

$$Z = \begin{bmatrix} 58.84 & 0.0014 \\ 0.0014 & 67.23 \end{bmatrix}$$

- Diagonals are Z_{oe} and Z_{oo}
- Compare to the prior modal results
  - Z_{oe}=58.86 Ω
  - Z_{oo}=67.27 Ω
- Same answer