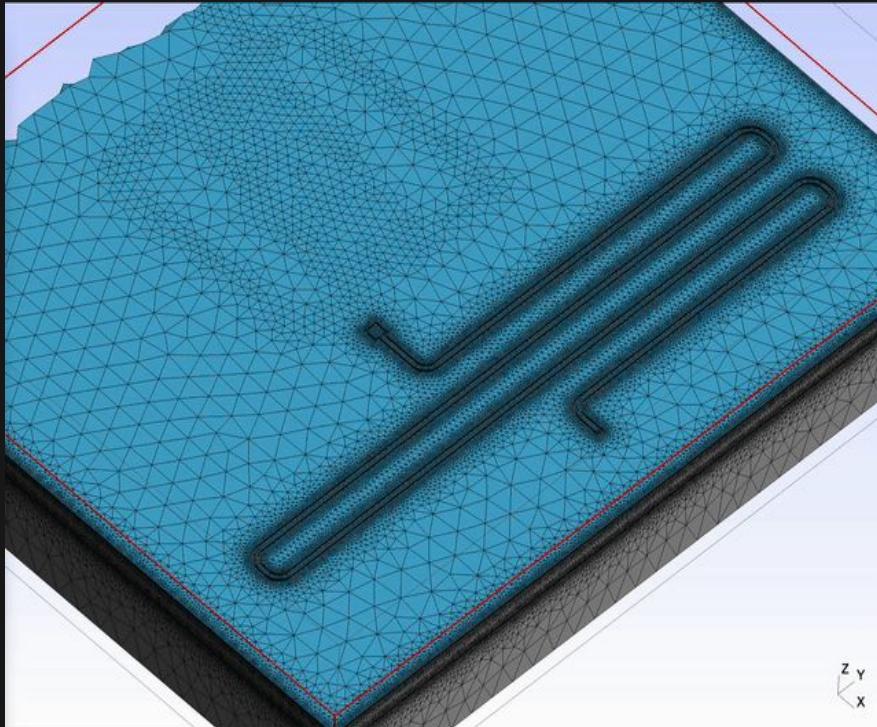


Open-Source Simulation Renderers for Metal — Gmsh & ElmerFEM

Presented By:

Diego Emilio Serrano
Qiskit Advocate (2020 -)
QAMP Fall 2022 Participant
MEMS dev, Panasonic

Abeer Vaishnav
Qiskit Advocate (2021 -)
Co-mentor @ QAMP Fall 2022
Graduate (Master's), Duke University



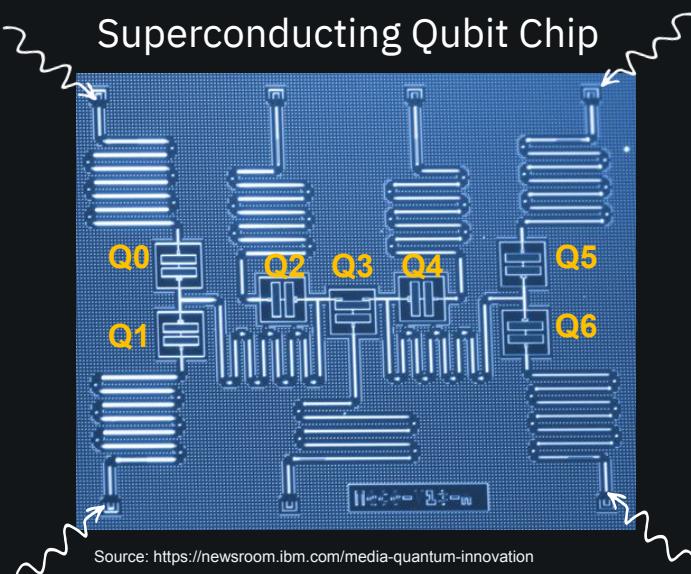
Superconducting Qubits



Circuit Abstraction



Superconducting Qubit Chip



Source: <https://newsroom.ibm.com/media-quantum-innovation>

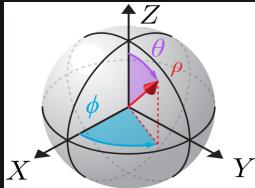
Superconducting Chip Design Flow



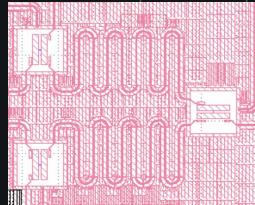
Concept



Hamiltonian



Layout



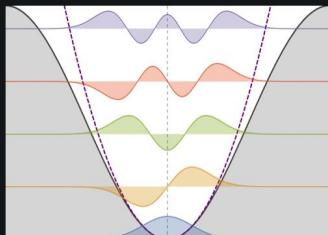
DRC
LVS



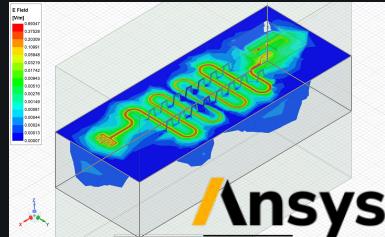
Fabrication



Quantum Analysis



Electromagnetic
Simulation



Ansys

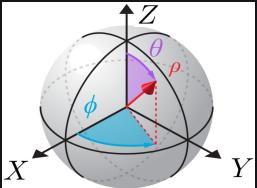
Superconducting Chip Design Flow



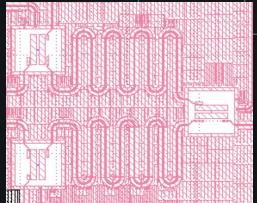
Concept



Hamiltonian



Layout



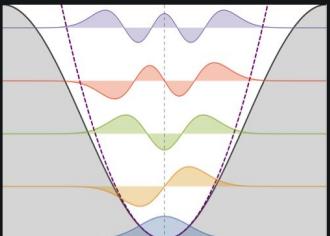
DRC
LVS



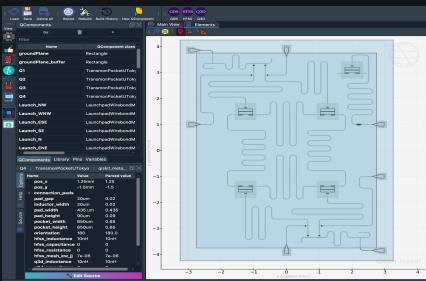
Fabrication



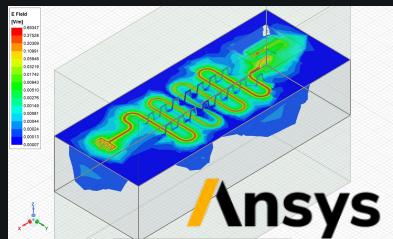
Quantum Analysis



Qiskit Metal¹ – Open Source



Electromagnetic
Simulation



Ansys

Limitations to ANSYS pipeline



High upfront costs



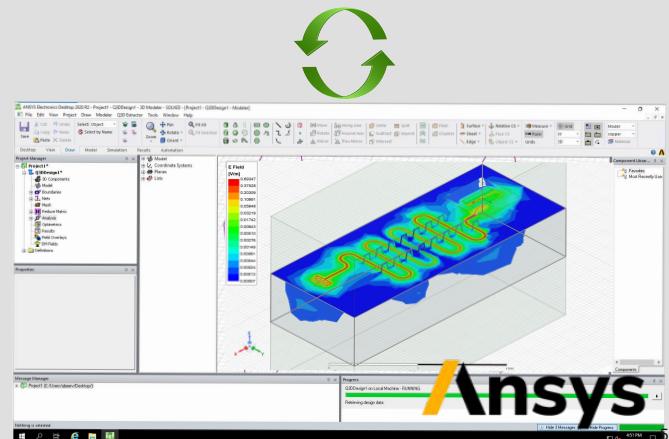
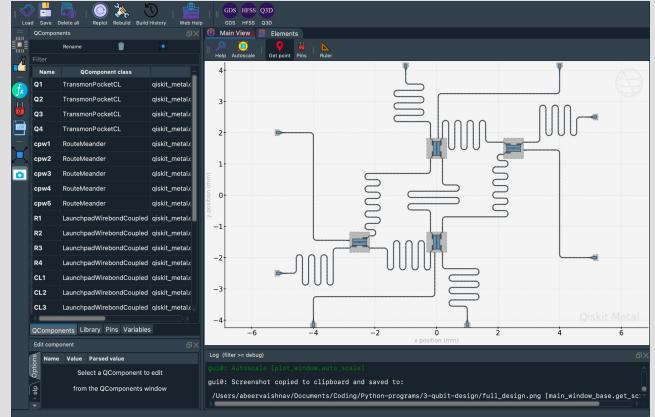
Limited access to academia



Closed source, limited availability



Extremely tight integration leading to inflexibility



Ansys

Different Open-source Solvers



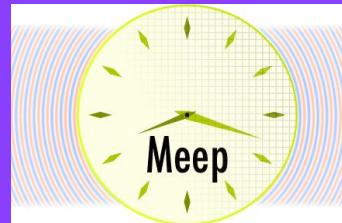
Solver
Type

Python

Non
Python

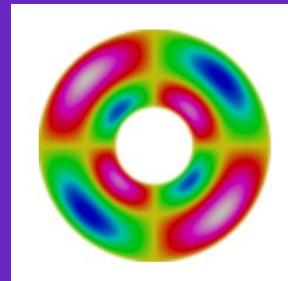
FDTD

(Finite Difference Time
Domain)



Multiphysics-FEM

(Finite Element Method)

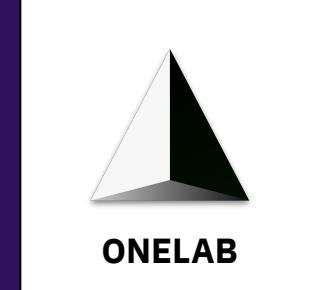


Elmer FEM



PDE-FEM

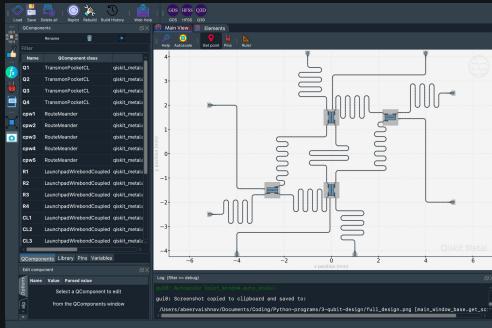
(Partial Differential
Equation-FEM)



Open-Source Renderer & Simulation Flow

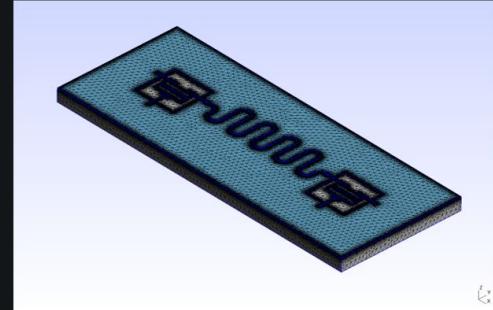


Qiskit | Metal



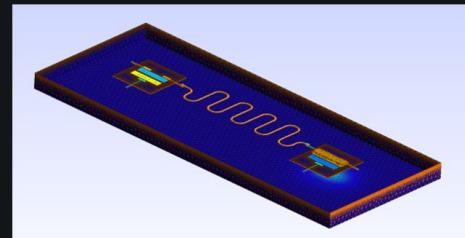
Render &
mesh

Gmsh



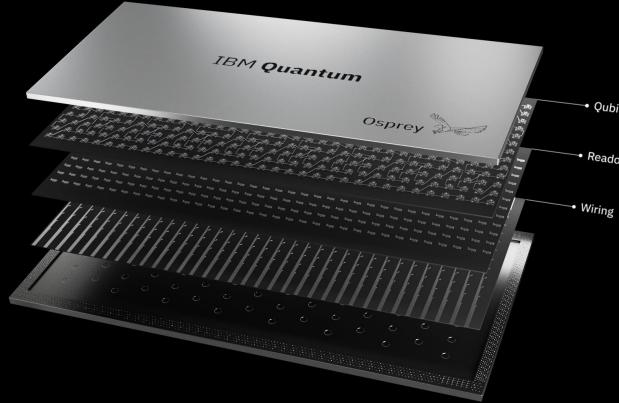
Quantum
Analysis

Elmer FEM
open source multiphysical simulation software

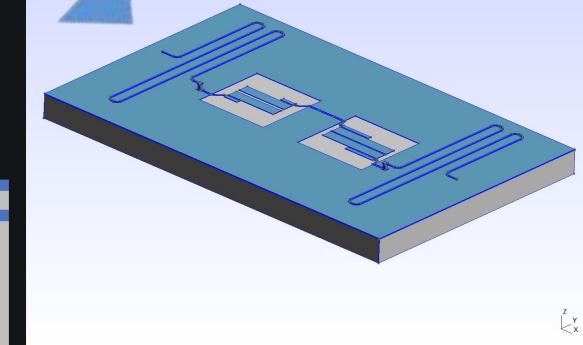
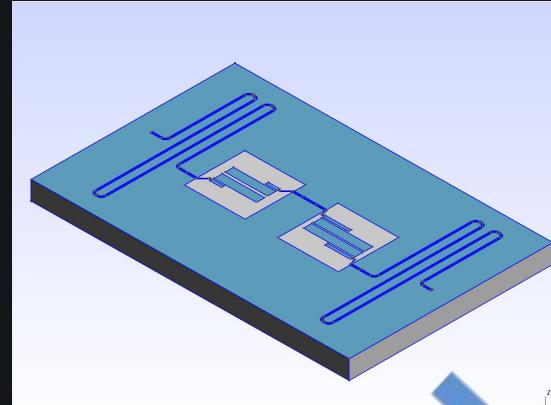


EM
Simulation

Making a Multi-planar design?



**IBMQ Osprey chip
(400+ qubits)**



“MultiPlanar” and “LayerStackHandler”



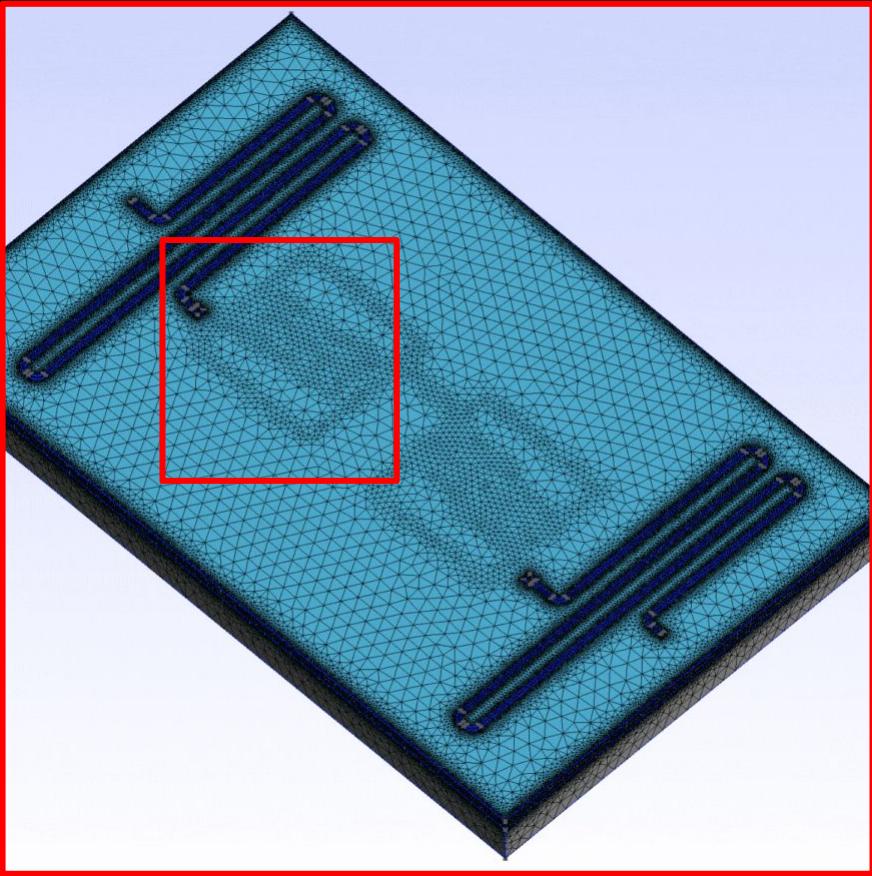
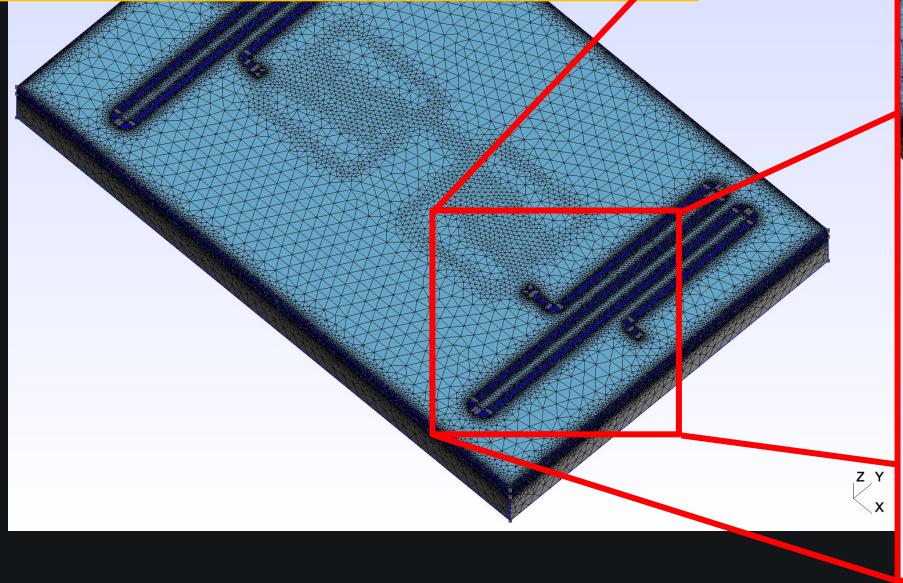
```
4
3 class MultiPlanar(QDesign):
2     """Metal class for a multiple planar design, consisting of either single or multiple chips.
1     Typically assumed to have some CPW geometries.
30 """
1 Inherits QDesign class.
"""
3
4 def __init__(self,
5         metadata: dict = None,
6         overwrite_enabled: bool = False,
7         enable_renderers: bool = True,
8         layer_stack_filename: str = None):
9     """Pass metadata to QDesign.
10
11     Args:
12         metadata (dict, optional): Pass to QDesign. Defaults to {}
13         overwrite_enabled (bool, optional): Passed to QDesign base
14         enable_renderers (bool, optional): Passed to QDesign base
15     """
16
```

```
5 class LayerStackHandler():
4     """Use DataFrame to hold information for multiple chips.
3
2     For all the chips, do NOT reuse the layer-number. However, within the dataframe,
1     we can repeat layer numbers since a layer can have multiple datatypes.
19 """
1
2
3     Col_Names = [
4         'chip_name', 'layer', 'datatype', 'material', 'thickness', 'z_coord',
5         'fill'
6     ]
```

	chip_name	layer	datatype	material	thickness	z_coord	fill
0	'main'	1	0	'pec'	'5um'	'0um'	'True'
1	'main'	2	0	'pec'	'50um'	'5um'	'True'
2	'main'	3	0	'pec'	'5um'	'55um'	'True'
3	'main'	4	0	'silicon'	'-200um'	'0um'	'True'
4	'main'	5	0	'silicon'	'50um'	'5um'	'True'

The Multi-planar design (layer-by-layer)

chip_name	layer	datatype	material	thickness	z_coord	fill	
0	'main'	1	0	'pec'	'5um'	'0um'	'True'
1	'main'	2	0	'pec'	'50um'	'5um'	'True'
2	'main'	3	0	'pec'	'5um'	'55um'	'True'
3	'main'	4	0	'silicon'	'-200um'	'0um'	'True'
4	'main'	5	0	'silicon'	'50um'	'5um'	'True'

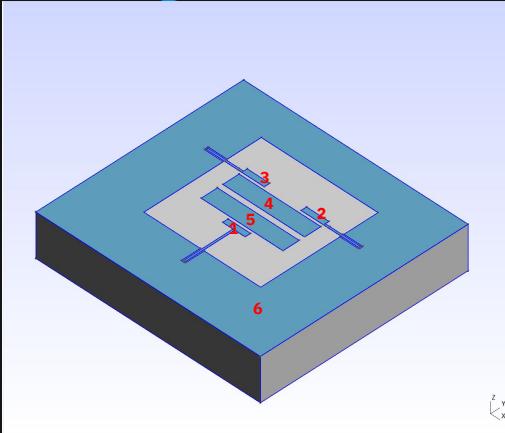


Changes to ElmerRenderer



- Several bugs fixed, such as correctly tying nodes to gnd planes
- Code within metal now identifies galvanically-connected elements across layers

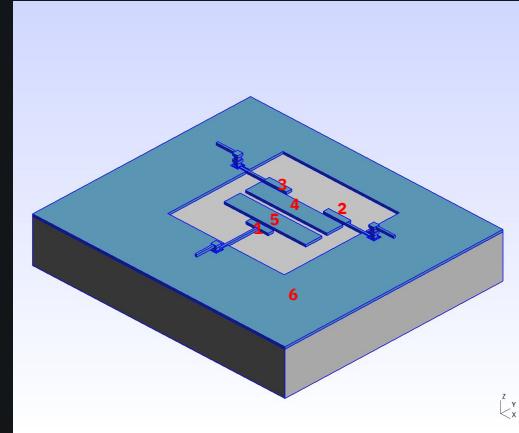
DesignPlanar (Old)



QAMP 2022



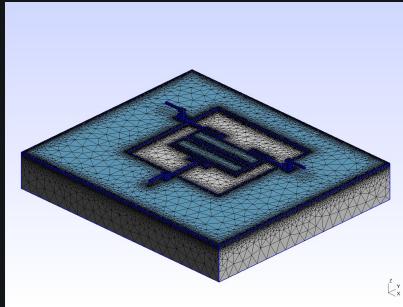
MultiPlanar (New)



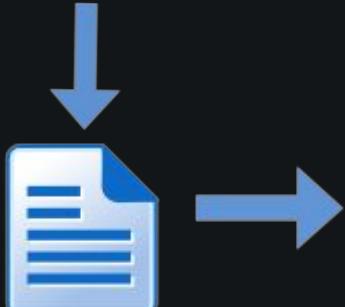
	1	2	3	4	5	6
Q1_readout_connector_pad	65.277768	-0.210593	-0.207456	-2.111670	-19.297574	-43.450474
Q1_coupler1_connector_pad	-0.210593	60.660334	-0.319431	-16.632976	-1.841543	-41.655791
Q1_coupler2_connector_pad	-0.207456	-0.319431	60.446323	-16.672141	-1.854419	-41.392876
Q1_pad_top	-2.111670	-16.632976	-16.672141	109.367816	-39.135447	-34.815582
Q1_pad_bot	-19.297574	-1.841543	-1.854419	-39.135447	102.522064	-40.393081
ground_plane	-43.450474	-41.655791	-41.392876	-34.815582	-40.393081	300.000000

	1	2	3	4	5	6
cpw1_rectangle	57.953225	-0.187749	-0.187077	-1.917556	-17.862119	-37.798723
via2_layer5_rectangle	-0.187749	63.533861	-0.294552	-15.389151	-1.714886	-45.947522
cpw3_rectangle	-0.187077	-0.294552	63.301340	-15.337806	100.791202	-35.968858
Q1_pad_top	-1.917556	-15.389151	-15.337806	-1.705651	-35.968858	-32.177831
Q1_pad_bot	-17.862119	-1.714886	-1.705651	-35.968858	94.620063	-37.368548
ground_plane	-37.798723	-45.947522	-45.776254	-32.177831	-37.368548	300.000000

Simulation Flow with Gmsh & ElmerFEM

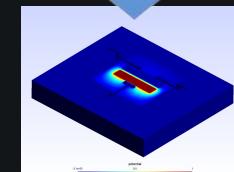
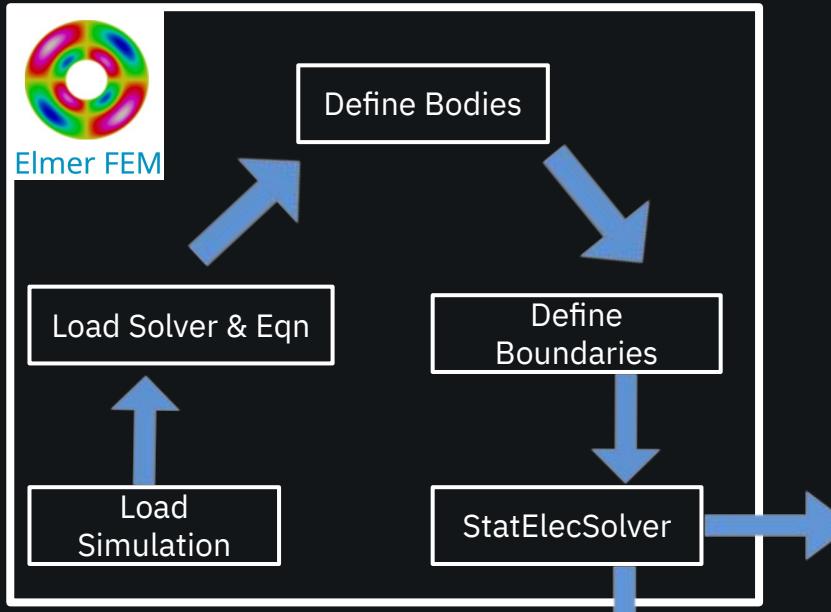


Gmsh geometry
and mesh

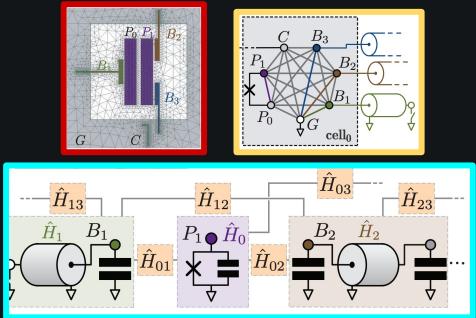


Mesh file from
Gmsh

Fully Automated !!!



Potential and
E-fields



	pad_left	pad_right	coup_right_top	coup_right_bot	coup_left	ground_plane
pad_left	88.734627	-39.284673	-1.943513	-1.943867	-7.820679	-37.741895
pad_right	-39.284673	97.906012	-12.554137	-12.565023	-1.563115	-31.939063
coup_right_top	-1.943513	-12.554137	40.012820	-0.217592	-0.205490	-25.092087
coup_right_bot	-1.943867	-12.565023	-0.217592	40.045262	-0.205291	-25.113488
coup_left	-7.820679	-1.563115	-0.205490	-0.205291	29.607223	-19.812647
ground_plane	-37.741895	-31.939063	-25.092087	-25.113488	-19.812647	300.000000

Capacitance
matrix

ElmerFEM Sample Code



```
# Import Qiskit Metal
from qiskit_metal import designs, MetalGUI

# Import previously made design from another file
from my_design import two_transmons_cpw_via

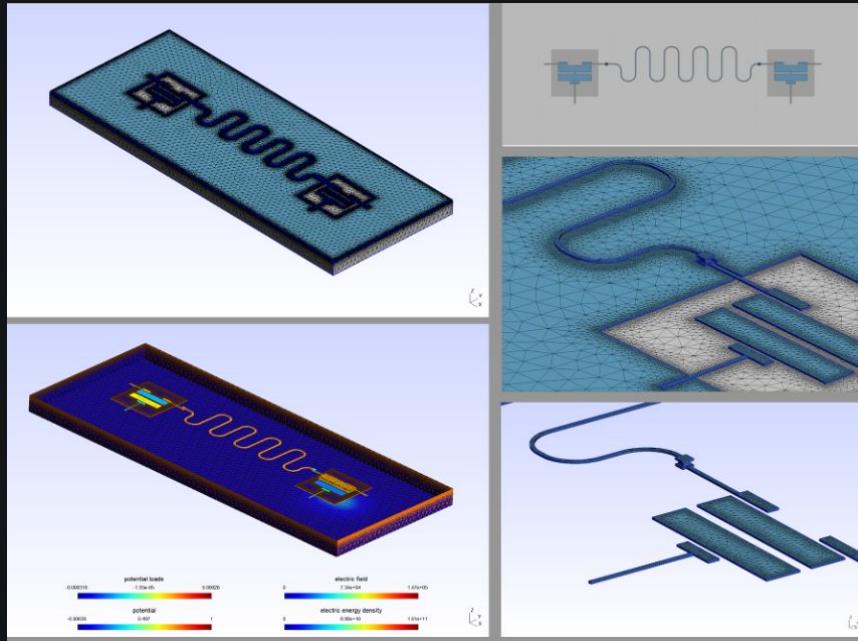
design = two_transmons_cpw_via()
gui = MetalGUI(design) # Initiate Metal GUI
gui.rebuild()

# Initiate Elmer and Gmsh renderers and render design
design.renderers.elmer.initialize_renderer()
design.renderers.elmer.render_design(mesh_geoms=True,
                                      skip_junctions=True,
                                      ignore_metal_volume=True)
design.renderers.elmer.launch_gmsh_gui()

# Export mesh
design.renderers.elmer.export_mesh()

# Add solution setup for Capacitance simulation
design.renderers.elmer.add_solution_setup("capacitance")

# Run the simulation and get the capacitance matrix
design.renderers.elmer.run("capacitance")
design.renderers.elmer.capacitance_matrix
```



	Q1.readout_connector_pad	via2_top.met_rectangle	via1_top.met_rectangle	Q2_coupler1.wire	Q2.readout_wire	Q2.pad_bot	Q2.pad_top	Q1.coupler2.wire	Q1.pad_bot	Q1.pad_top	ground_plane
Q1.readout_connector_pad	1.866081e+02	-2.196170e-08	-1.959564e-01	-6.677632e-09	-3.618326e-10	-1.523158e-09	-1.170250e-09	-1.939131e-01	-1.959269e+01	-1.954447e+00	-154.671073
via2_top.met_rectangle	-2.196170e-08	1.929452e+03	2.497471e+00	2.948392e-01	-1.944685e-01	-1.738548e-09	-1.710669e+01	-1.525556e-08	-5.405125e-08	-5.298342e-08	-1907.546402
via1_top.met_rectangle	-1.959564e-01	2.497471e+00	1.917029e-02	-1.817490e-08	1.890388e-08	1.432672e-08	-1.738838e-08	-2.950238e-01	-1.739856e-08	-1.705325e+01	-17.321361
Q2_coupler1.wire	-6.677632e-09	-2.948392e-01	-1.817490e-08	1.695470e-02	-1.884198e-01	-1.714790e-09	-1.694522e-01	-1.662721e-08	-1.694522e-01	-3.49811e-09	-1.662721e-08
Q2.readout_wire	-3.618326e-10	-3.618326e-09	-1.944685e-01	1.884198e-01	1.885919e-02	-1.944732e-01	-1.946803e+00	-9.911296e-10	-5.827352e+02	-3.685274e+01	1.551127e-09
Q2.pad_bot	-1.523158e-09	-1.738548e+00	1.432672e-08	-1.714780e+00	-1.944732e+01	5.827352e+02	-3.685274e+01	1.551127e-09	-5.799669e-09	-5.411305e-09	-52.981840
Q2.pad_top	-1.170250e-09	-1.738633e-08	-1.945221e-01	-1.946803e+00	-3.685274e+01	-5.897546e+02	-8.001452e-11	-5.168499e-09	-5.347795e-09	-4.759484e-09	-516.849939
Q1.coupler2.wire	-1.939131e-01	-1.525556e-08	-2.950238e-01	-3.648911e+00	-9.911296e-10	1.551127e-09	-8.001452e-11	1.680331e-02	-1.719450e+00	-1.704211e+01	-148.782558
Q1.pad_bot	-1.959269e+01	-5.405125e-08	-5.298342e-08	-1.632791e-08	-1.493772e-09	-6.579696e-09	-5.347795e-09	-1.719450e+00	-5.829981e+02	-3.670977e+01	-522.934297
Q1.pad_top	-1.954447e+00	-5.298342e-08	-1.705256e-09	-7.027731e-09	-1.161767e-09	-5.411305e-09	-4.759484e-09	-1.704211e+01	-3.670977e+01	-5.399533e+02	-517.195730
ground_plane	-1.646711e+02	-1.907546e+03	-1.753214e+02	-1.505637e+02	-1.648150e+02	-5.229818e+02	-5.168499e+02	-1.407829e+02	-5.229343e+02	-5.171957e+02	300.300000

Current Status

- All outstanding items in PR have been completed
- PR ready to be merged once reviewed by qiskit-metal team

What are the issues this pull addresses (issue numbers / links)?

Fixes #881

Summary

Adds ElmerFEM to available renderers in qiskit-metal.

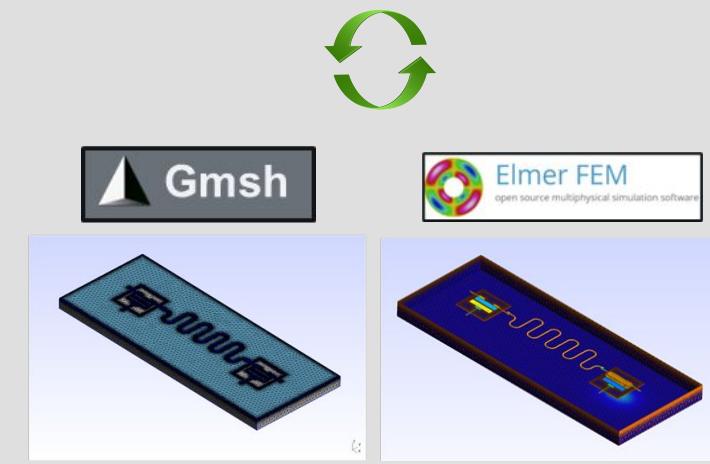
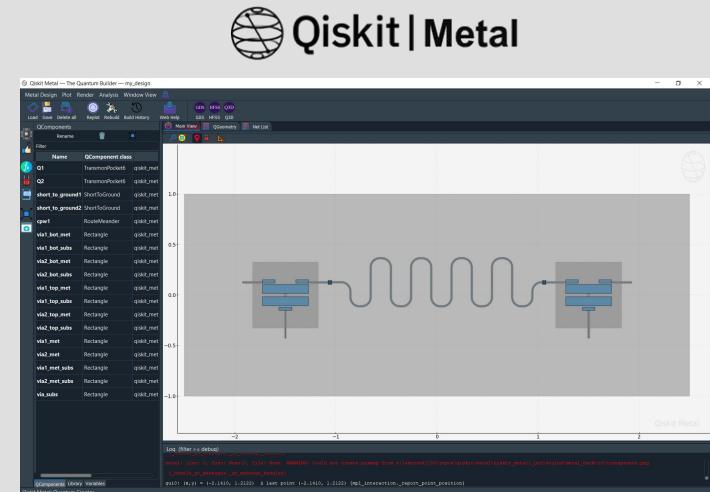
Details

Submitting this as a draft PR. The following items need to be completed before merging:

- Fix name of `BIGStab1 Polynomial Degree` parameter to avoid duplicate.
- Fix net assignment issue generated when a given node is in contact with ground plane.
- Add/update Docstrings to all functions/methods in `elmer_renderer.py`.
- Add/update Docstrings to all functions/methods in `elmer_runner.py`.
- Add error checking within `QGmshRenderer` to inform user if items in `open_pins` list don't exist.
- Add warning message if no cap bodies found in design (ElmerFEM will error out).
- Add unit tests for `QElderRender`.
- Update Gmsh/ElmerFEM installation instruction, and place them in the appropriate directory.
- Update ElmerFEM Tutorial to work with `Multiplanar` class.
- Add `ElmerRender` options to `Analysis` class

Additional Comments

- The current version of ElmerFEM will only support capacitance extraction analysis. Next steps will include adding more analyses, such as eigenfrequency extraction.
- The `assign_nets` function inside `QElderRender` works properly but probably needs refactoring since it inefficiently scans through different geometries assign nets and it has grown so much it makes the code difficult to read.
- It has been decided to leave out adding `QElderRender` to `Analysis` as part of this PR. This has to be done later on when LOM2.0 gets integrated.



Contribution Opportunities



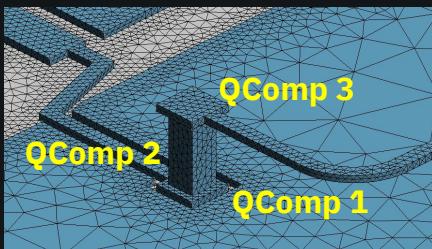
- **Metal intersection issue in gmsh**



<https://github.com/Qiskit/qiskit-metal/issues/871>

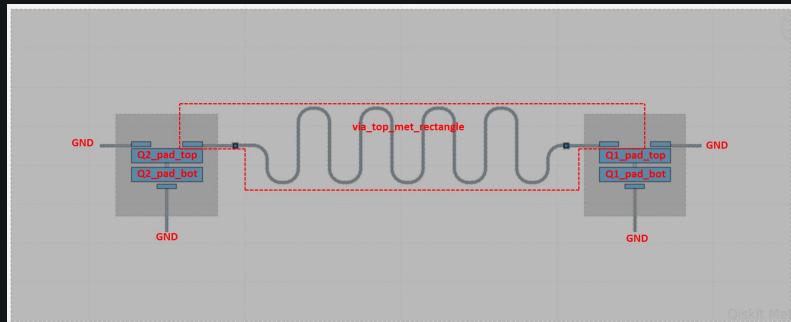
- **Defining multi-layer components**

- Qcomponents, only support single-layer definition
- Is it possible to expand that functionality?



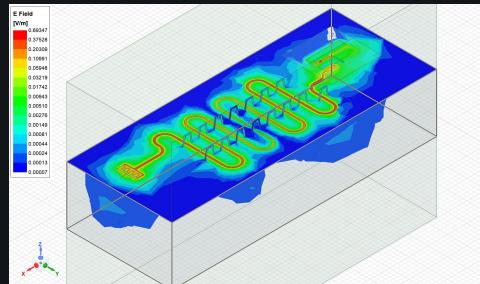
- **Better naming convention**

- ElemerRenderer assigns node name based on last element
- Names could be hard to track down



Qiskit Metal

- **Eigenmode simulation and field calculation**



Thank You!