magnetic resonance integral equation

Marie

Coil modeling:
Numerical overview
File formats
Data structures

Beta Version, Spring 2015

www.rle.mit.edu/cpg

MARIE - magnetic resonance integral equation suite.
A prototype MATLAB open source software for the fast electromagnetic analysis of MRI systems

RLE Computational Prototyping Group, Massachusetts Institute of Technology
contact: jvillena@mit.edu / thanos_p@mit.edu

- **This document introduces**
  - the information related to the coil models in MARIE
  - file formats for coil definition
    - step-by-step of how to generate valid MARIE geometries
  - structures with coil information

contents

- **Coil modeling**
  - brief numerical overview

- **Surface based coil models**
  - how to generate a surface coil model (.smm)
  - how to use a surface coil model

- **Wire based coil models**
  - how to generate a wire coil model (.wmm)
  - how to use a wire coil model

- **Main reference**

[1] Villena et al., MRGFs paper

- **Coil analysis functionalities are based on**

[1] Tai, “Dyadic Green functions in electromagnetic theory”. 1994

[2] Harrigton, Thin wire approximations / Book

[3] Rao et al., “Electromagnetic scattering by surfaces of arbitrary shape”, TAP 1982

[4] Mosig et al., Delta gap paper

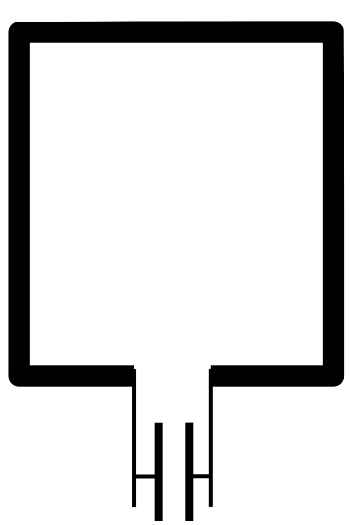
[5] Chew et al., Delta gap paper

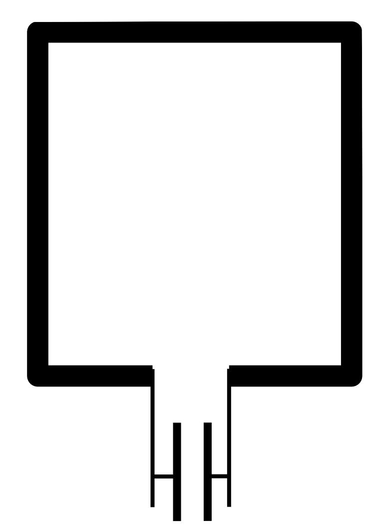
[5] **C. Geuzaine and J.-F. Remacle.**“Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities”. International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331, 2009.
more details and download in http://geuz.org/gmsh/

brief numerical overview

- **Why integral equation methods for coil analysis?**
  - integral equations: mature technology
    - widely used on antenna community for decades
  - are based on applying the Green function
    - gives the fundamental solution at any given point in space
    - satisfies (by definition) the radiation conditions
  - consequently
    - only need to discretize the surface of the conductors
      - smaller (but dense) systems
    - do not require to define a bounded domain or boundary conditions
      - simplifies things for the user

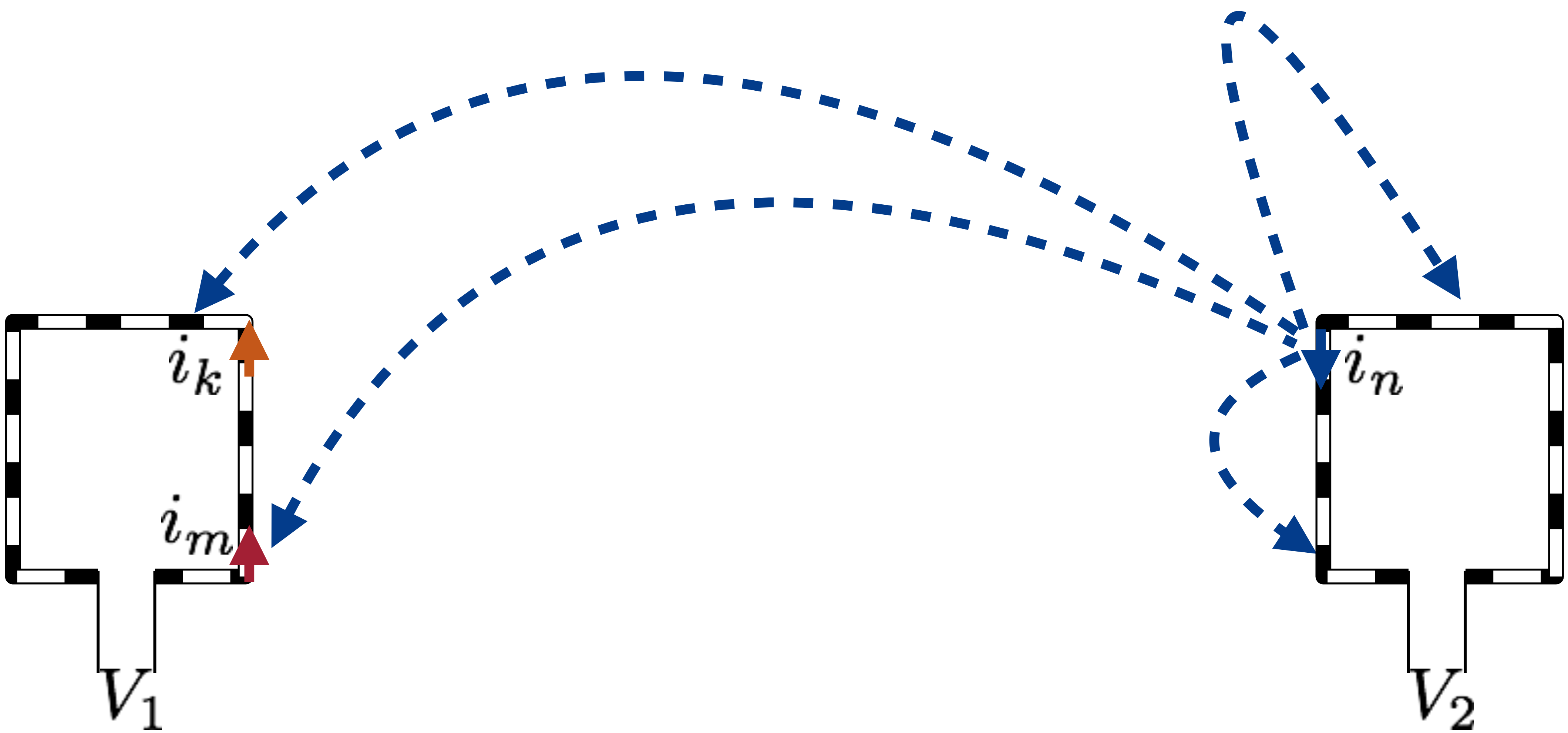
- **Define the geometry of the coils**
  - coil materials are homogeneous conductors
    - shield can be included as part of the coil array
  - coils are surrounded by air
    - or any other non-electric non-magnetic material





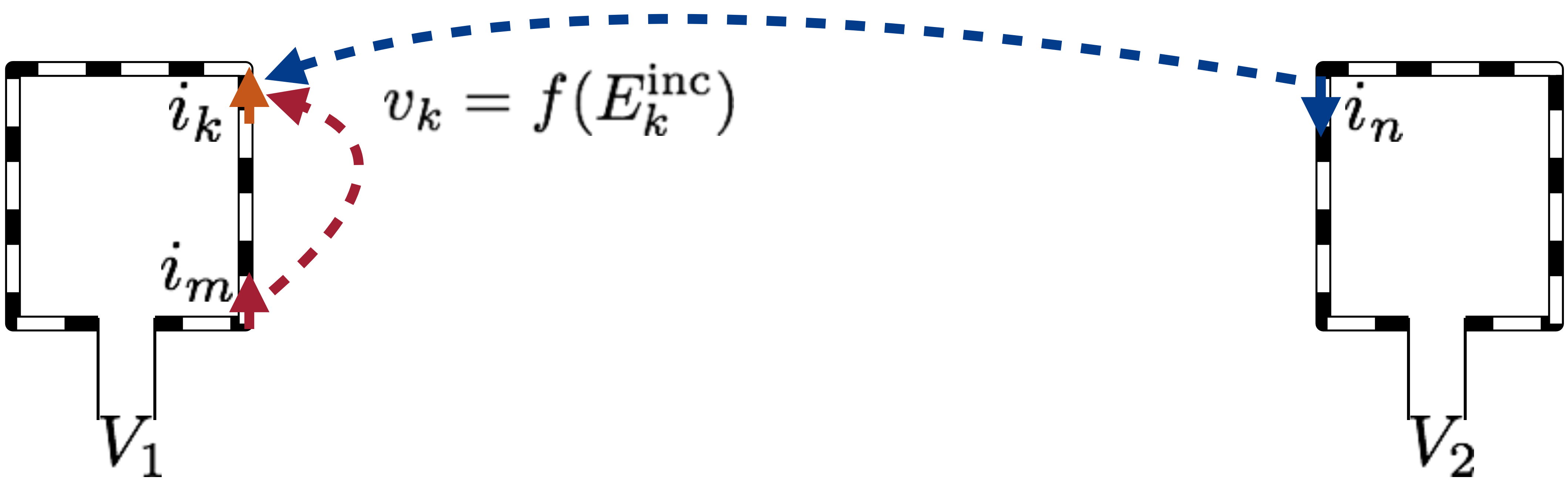
- **Select the discretization, basis functions, and excitation**
  - only the conductor surface needs to be tessellated (not the air)
  - assume constant current at each tessellated element
  - elements connected to external circuitry define the ports

- **Apply the corresponding numerical engine**
  - each element radiates EM fields due to its associated constant current
  - electric fields are imprinted in the conductor surface



- **Apply the corresponding numerical engine**
  - each element radiates EM fields due to its associated constant current
  - electric fields are imprinted in the conductor surface
    - field at each element surface is the contribution of all elements

$$\ldots + \underline{G(k,m)i_m} + \underline{G(k,k)i_k} + \underline{G(k,n)i_n} + \ldots = E_k^{\mathrm{inc}}$$

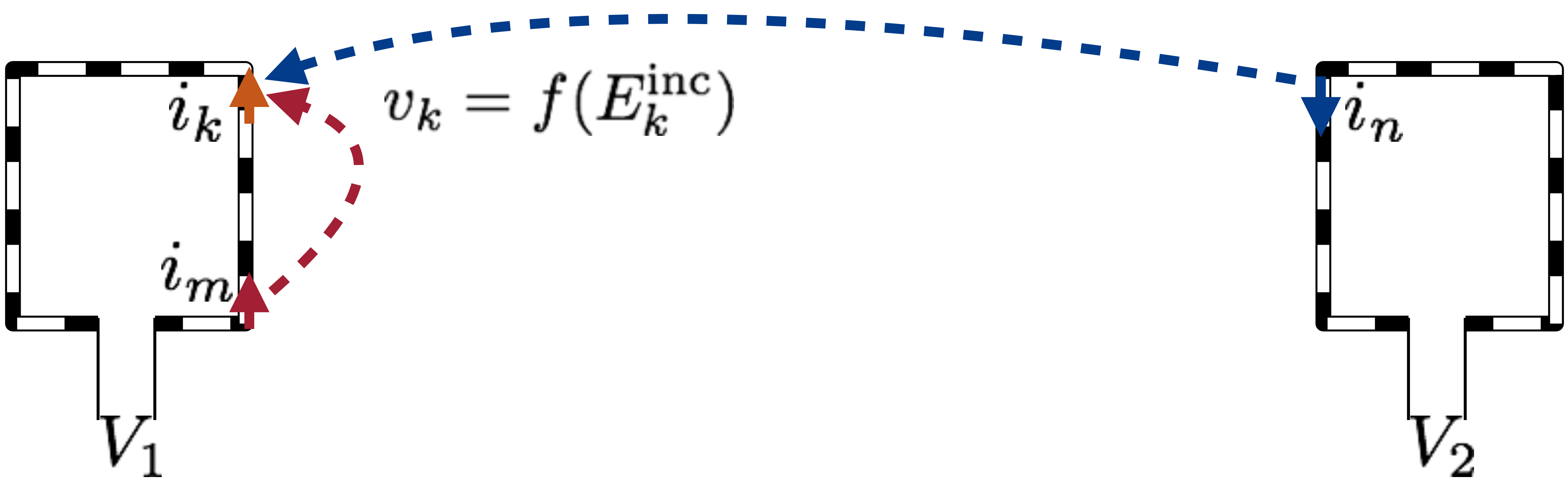


- **Apply the corresponding numerical engine**
  - form a system that satisfies Maxwell's equations at each element
    - unknowns are the coefficients of the basis functions (related to currrents)
    - excitations are external fields imprinted in the ports

$$\mathbf{Z}^c\mathbf{J} = \mathbf{F}$$

$$z_{k,m} \qquad z_{k,k} \qquad z_{k,n}$$

$$\ldots + \underline{G(k,m)i_m} + \underline{G(k,k)i_k} + \underline{G(k,n)i_n} + \ldots = E_k^{\mathrm{inc}}$$



- **Solve the system to obtain**
    - port parameters: Z, Y or S port parameters
        - from the basis coefficients and port excitations
    - current distribution within the coil
        - from the basis coefficients
    - electromagnetic fields at any position outside the coil conductors
        - from the basis coefficients

surface coil models

- **What is a surface coil model in MARIE?**
  - a structure based on planar conductors (e.g. copper tape)
    - negligible thickness in comparison with the width of conductor
  - surface integral equation (SIE) methods are applied to model it

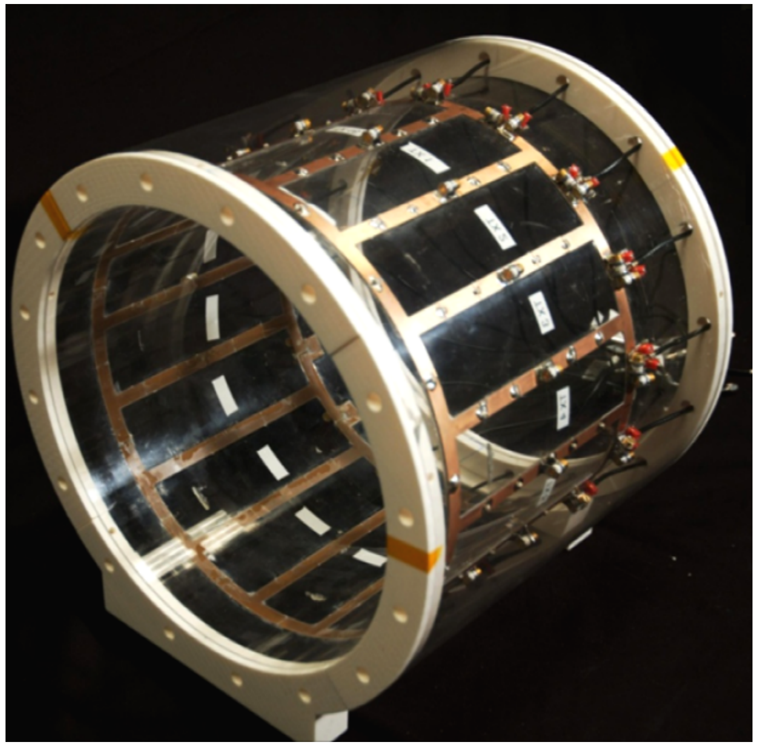
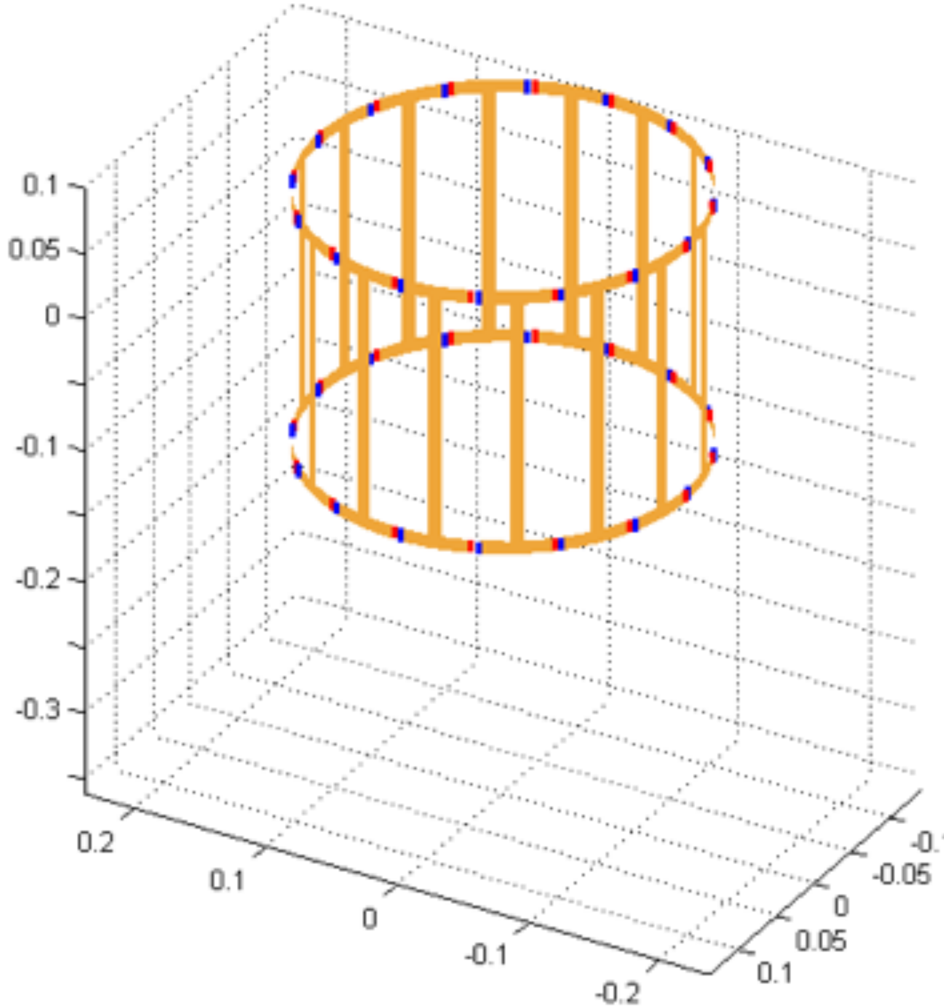


image courtesy of Wei Zhao
Martinos Center, MGH



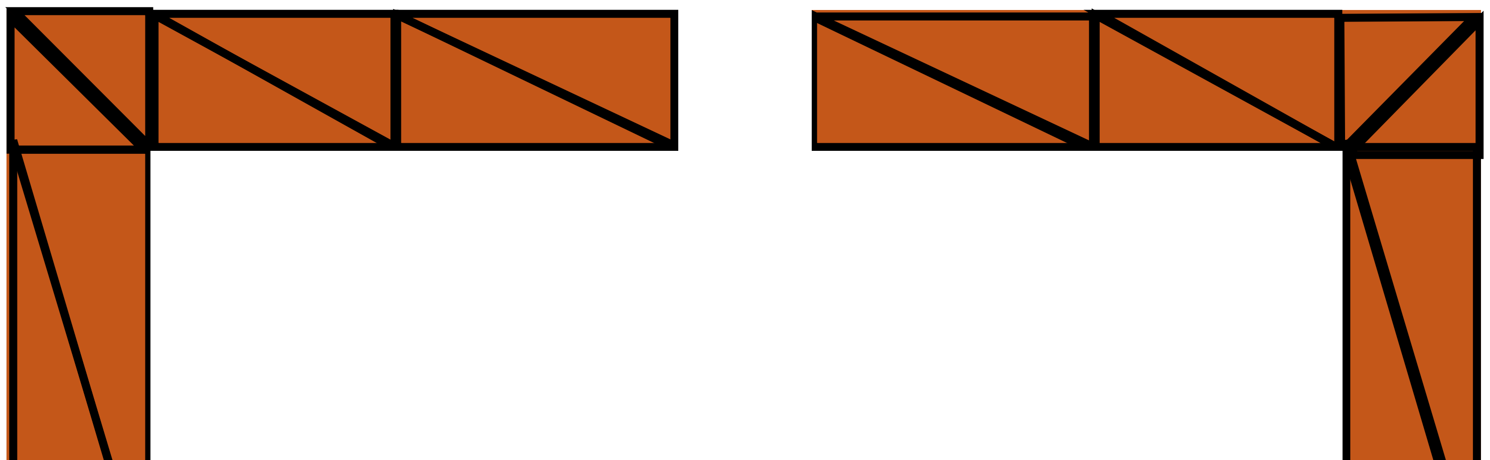
- **Discretization and modeling methodology**
  - triangular tessellation (standard in antenna's community)
    - allows to capture conformal and complicated geometries
    - roust and available engines for mesh generation*
  - RWG basis functions (the de facto standard in SIE approaches)
    - well established, good accuracy
  - Machine precision numerical integration
    - DEMCEM

*MARIE surface coil model generation and format is based on gmsh

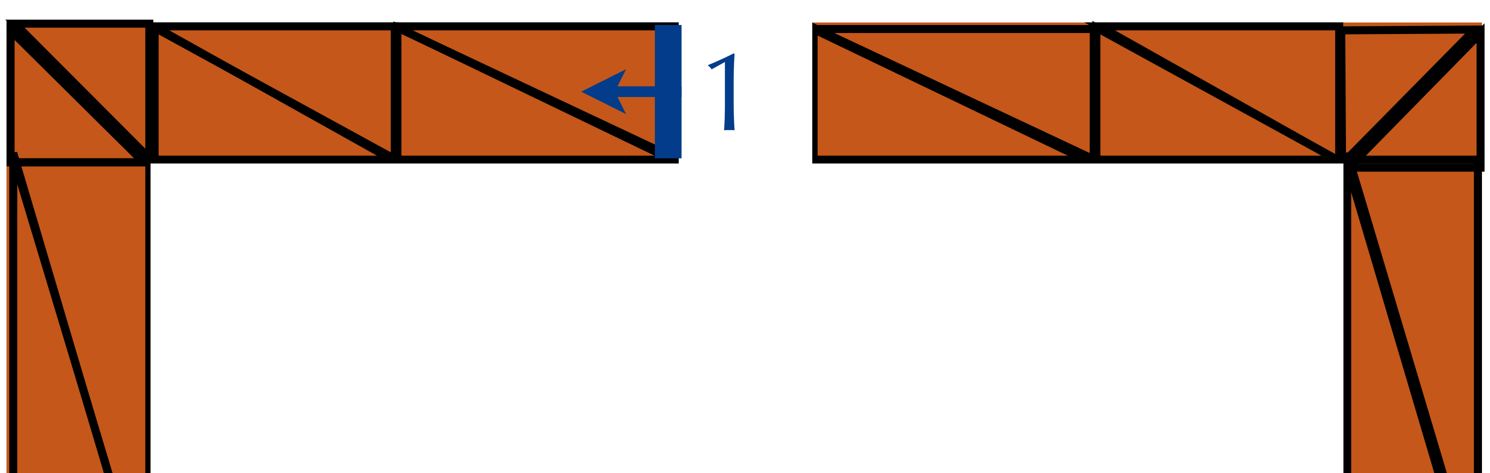
C. Geuzaine and J.-F. Remacle."Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities". International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331, 2009. more details and download in http://geuz.org/gmsh/)

- **Formulation, port and basis definitions**
  - given some planar coil

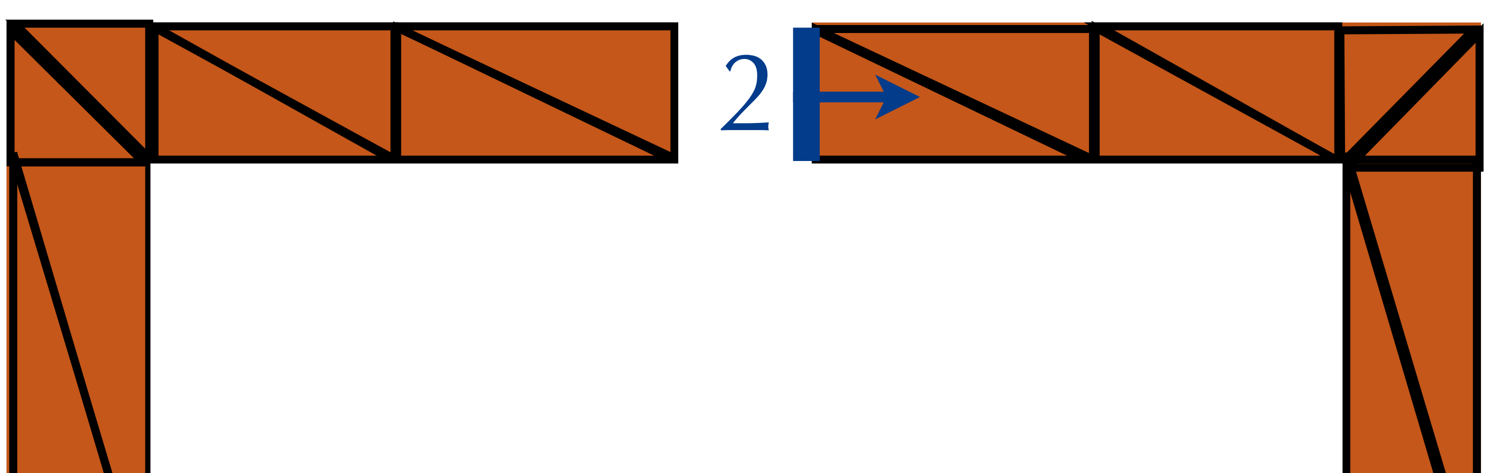
- **Formulation, port and basis definitions**
    - tesselate the surface of conductors into small triangles
    - combine with desired basis functions: RWG in our case



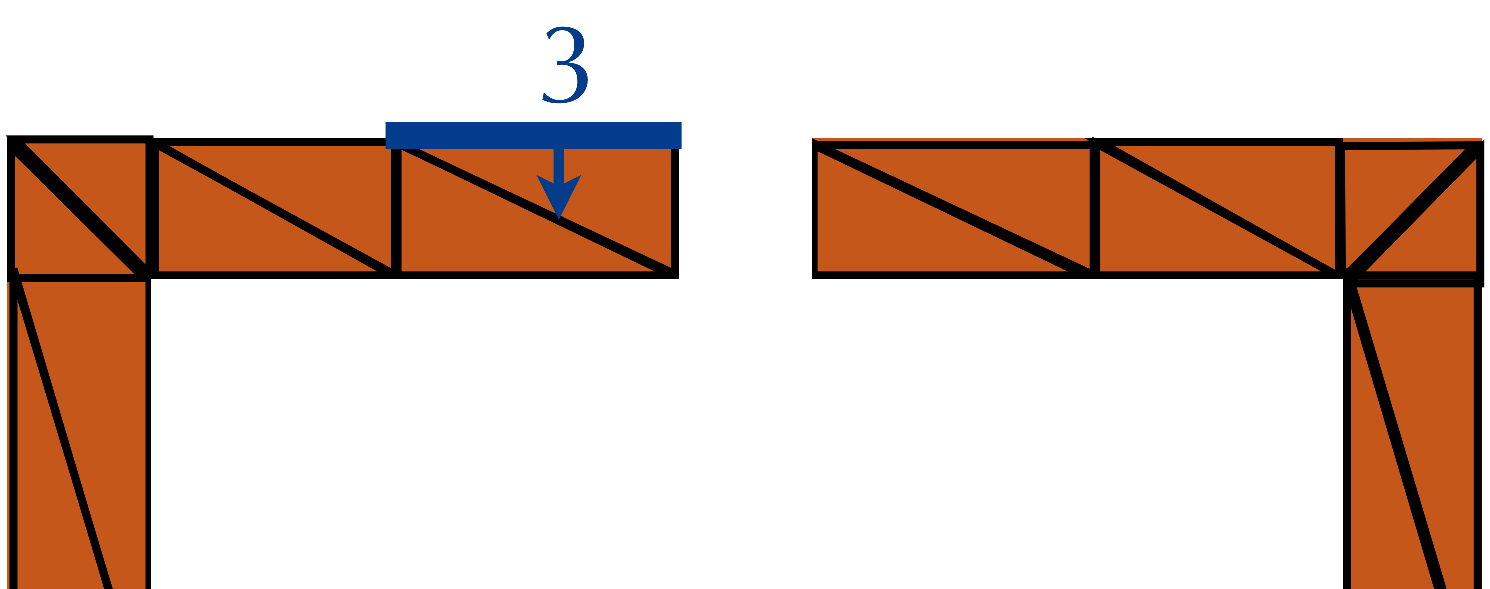
- **Formulation, port and basis definitions**
  - tesselate the surface of conductors into small triangles
  - combine with desired basis functions: RWG in our case
    - each triangle edge is associated with at unknown
    - unknowns: coefficients of the basis functions, related to currents



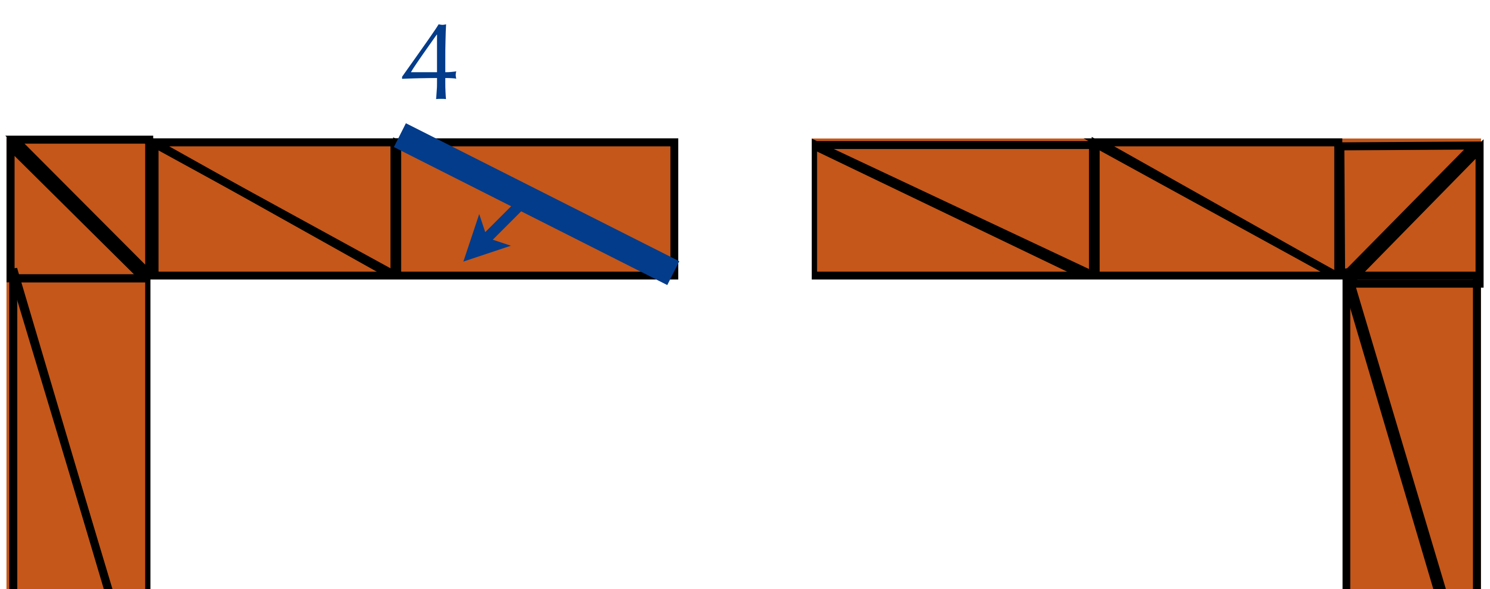
- **Formulation, port and basis definitions**
  - tesselate the surface of conductors into small triangles
  - combine with desired basis functions: RWG in our case
    - each triangle edge is associated with at unknown
    - unknowns: coefficients of the basis functions, related to currents



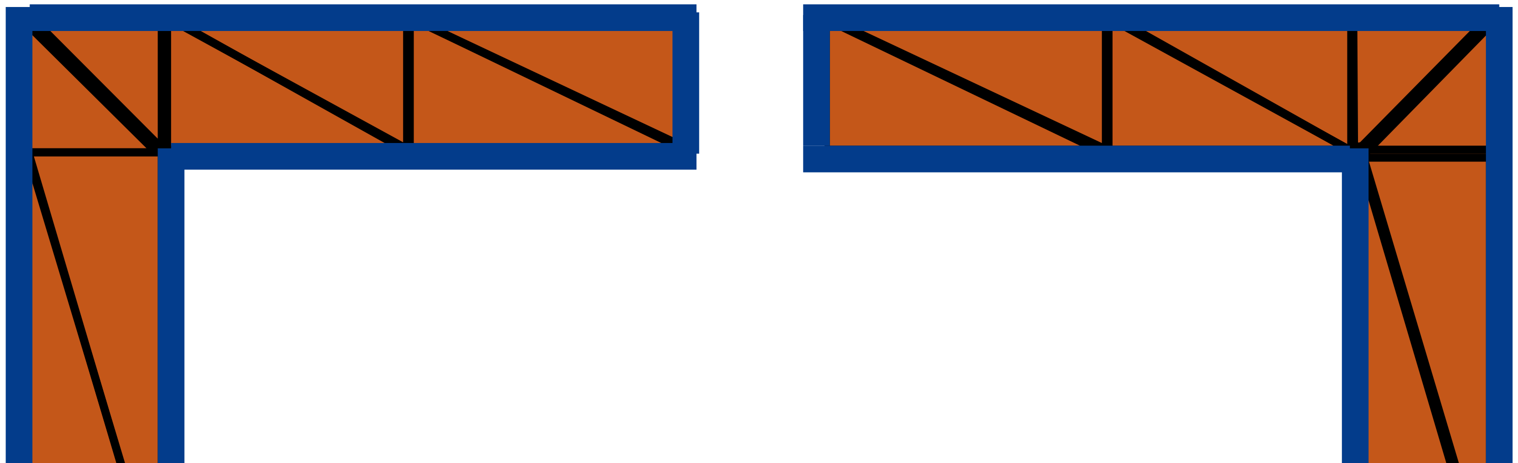
- **Formulation, port and basis definitions**
  - tesselate the surface of conductors into small triangles
  - combine with desired basis functions: RWG in our case
    - each triangle edge is associated with at unknown
    - unknowns: coefficients of the basis functions, related to currents



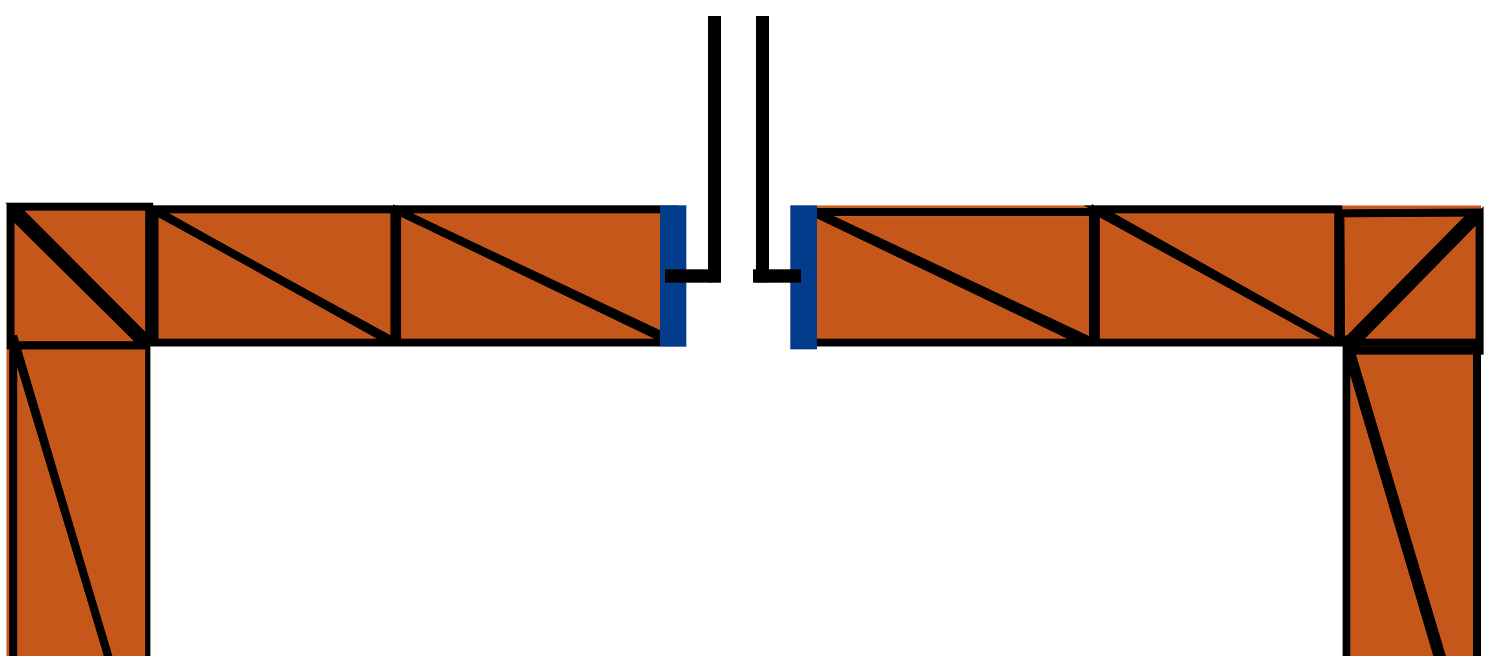
- **Formulation, port and basis definitions**
  - tesselate the surface of conductors into small triangles
  - combine with desired basis functions: RWG in our case
    - each triangle edge is associated with at unknown
    - unknowns: coefficients of the basis functions, related to currents



- **Formulation, port and basis definitions**
  - tesselate the surface of conductors into small triangles
  - combine with desired basis functions: RWG in our case
    - each triangle edge is associated with at unknown
    - unknowns: coefficients of the basis functions, related to currents
  - external edges are not taken into account
    - current does not flows outside the conductor



- **Formulation, port and basis definitions**
  - tesselate the surface of conductors into small triangles
  - combine with desired basis functions: RWG in our case
    - each triangle edge is associated with at unknown
    - unknowns: coefficients of the basis functions, related to currents
  - external edges are not taken into account
    - current does not flows outside the conductor
    - except for ports: pair of external edges connected to circuitry



- **Formulation, port and basis definitions**
  - apply the corresponding numerical computations to form the system

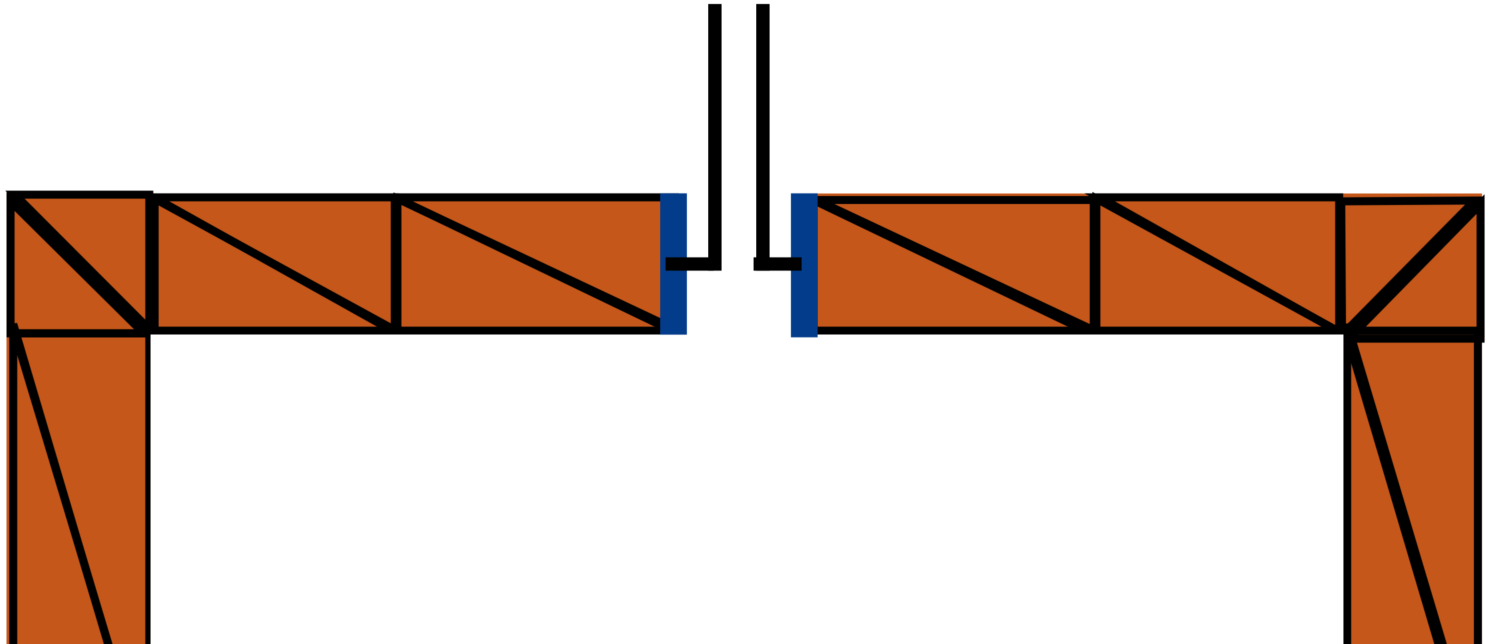
    [Zc,F]=Assembly_SIE(COIL,freq)

  - solve the system to obtain the port impedance and basis

    [Zparameters,J]=SIE_Solver(COIL,freq)

  - obtain the radiated fields in any point outside the coils

    [E,H]=SIE_Radiate(COIL,J,freq,r)



- **MARIE uses .smm (surface marie model) file defining surface coil**
  - these are just text files with 3 lines

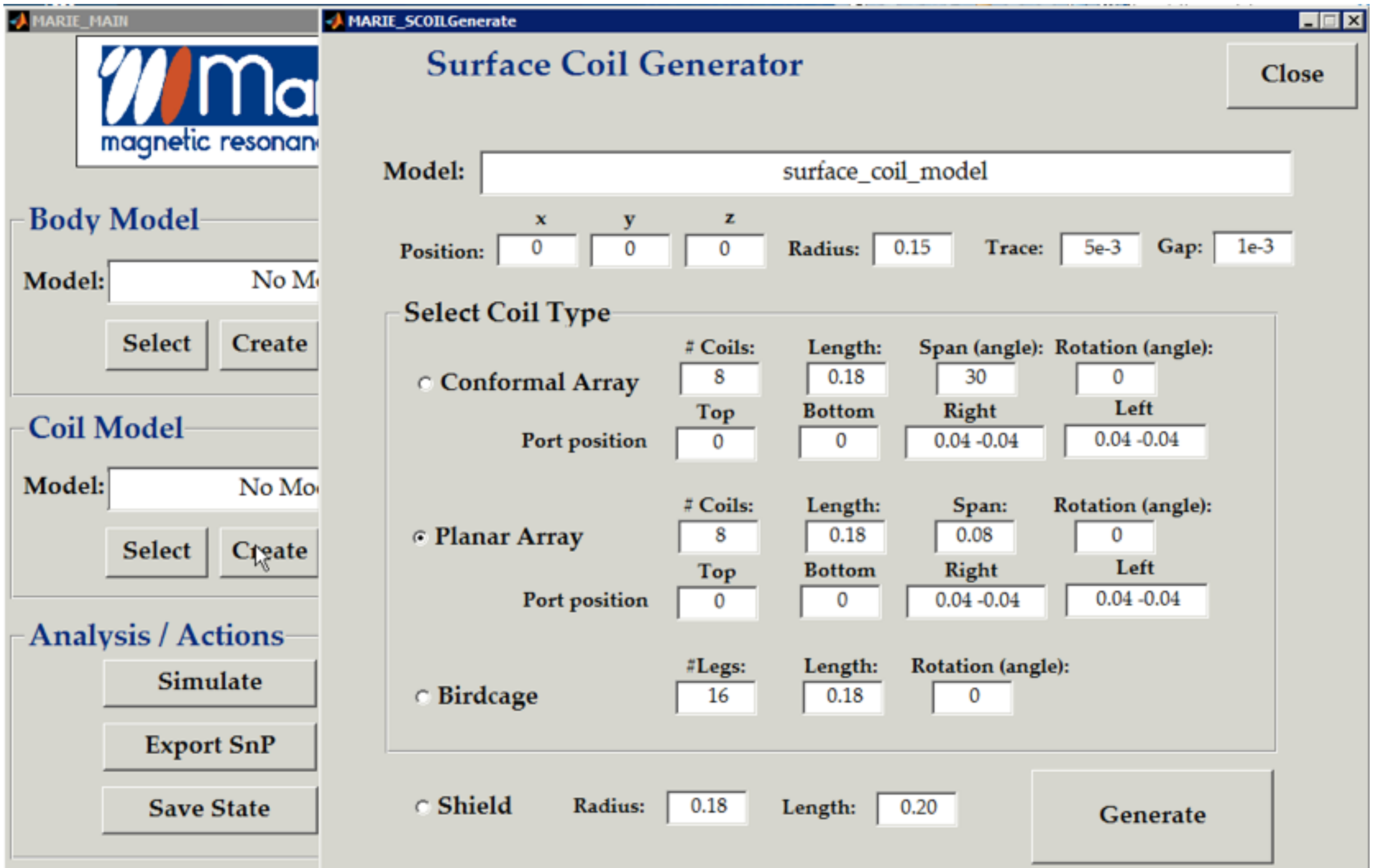
circularloop_planar.smm

```
Circular Loop, R 40mm, Dist 120mm
1.2579e-08 0.0005
CircularLoop.msh
```

- first line is the model name and/or description
- second line is two numbers
  - first the resistivity of the material
  - second the thickness of the planar material
- third line is the name of the .msh file with the geometry tessellation
  - Next info will illustrate how the .msh can be generated

- **Creating the surface coil model**
  - MARIE needs a triangular tessellation of the surface and port info
  - uses gmsh, a free open-source code for polygon mesh
    - http://geuz.org/gmsh/
    - robust and simple to use, very well documented
    - reads a .geo file and meshes the geometry to generate a .msh file
  - MARIE incorporates a interface for typical coil array generation
    - automatically generates the geometry files (.geo) for gmsh
    - and functions to read standard .msh files generated by gmsh

- **MARIE GUI incorporate an automatic coil generator**
  - For common geometries
  - Other geometries can be easily defined by creating the .geo file



- **.geo files used as gmsh to define the geometry**

surface_coil_model.geo — Edited

```
// ------------------------------------------------------------------
// CONFORMAL COIL Radius 0.15m, geo created by MARIE coil generator
// ------------------------------------------------------------------

RES = 0.015; // Resolution

Point(1101) = { 0,  0, 0.09, RES};
Point(1102) = { 0,  0, 0.085, RES};
Point(1103) = { 0,  0, -0.085, RES};
.
.
.
Point(1128) = {-0.0005, 0.15, -0.09, RES};

Line(1501) = {1105, 1106};
Circle(1502) = {1106, 1101, 1107};
.
.
.
Line(1532) = {1108, 1119};

Line Loop(1701) = { 1501, 1502, 1503, 1504 };
.
.
.
Line Loop(1710) = { -1514, 1531, -1504, 1532 };

Physical Line(1001) = {1518};
Physical Line(1002) = {1520};
Physical Line(1003) = {1526};
Physical Line(1004) = {1528};

Ruled Surface(1801) = {1701};
.
.
.
Ruled Surface(1810) = {1710};

Physical Surface(1) = {S1801, S1802, S1803, S1804, S1805, S1806, S1807, S1808, S1809, S1810};

// ---------------------------------------------------------------
```

- **.geo files used as gmsh to define the geometry**

surface_coil_model.geo — Edited

```
// ------------------------------------------------------------------
// CONFORMAL COIL Radius 0.15m, geo created by MARIE coil generator
// ------------------------------------------------------------------

RES = 0.015; // Resolution

Point(1101) = { 0,  0, 0.09, RES};
Point(1102) = { 0,  0, 0.085, RES};
Point(1103) = { 0,  0, -0.085, RES};
.
.
.
Point(1128) = {-0.0005, 0.15, -0.09, RES};

Line(1501) = {1105, 1106};
Circle(1502) = {1106, 1101, 1107};
.
.
.
Line(1532) = {1108, 1119};

Line Loop(1701) = { 1501, 1502, 1503, 1504 };
.
.
.
Line Loop(1710) = { -1514, 1531, -1504, 1532 };

Physical Line(1001) = {1518};
Physical Line(1002) = {1520};
Physical Line(1003) = {1526};
Physical Line(1004) = {1528};

Ruled Surface(1801) = {1701};
.
.
.
Ruled Surface(1810) = {1710};

Physical Surface(1) = {S1801, S1802, S1803, S1804, S1805, S1806, S1807, S1808, S1809, S1810};

// --------------------------------------------------------------
```

Resolution of the discretization
- All dimensions are in meters

- **.geo files used as gmsh to define the geometry**

surface_coil_model.geo — Edited

```
// ---------------------------------------------------------------
// CONFORMAL COIL Radius 0.15m, geo created by MARIE coil generator
// ---------------------------------------------------------------

RES = 0.015; // Resolution

Point(1101) = { 0,  0, 0.09, RES};
Point(1102) = { 0,  0, 0.085, RES};
Point(1103) = { 0,  0, -0.085, RES};
.
.
.
Point(1128) = {-0.0005, 0.15, -0.09, RES};

Line(1501) = {1105, 1106};
Circle(1502) = {1106, 1101, 1107};
.
.
.
Line(1532) = {1108, 1119};

Line Loop(1701) = { 1501, 1502, 1503, 1504 };
.
.
.
Line Loop(1710) = { -1514, 1531, -1504, 1532 };

Physical Line(1001) = {1518};
Physical Line(1002) = {1520};
Physical Line(1003) = {1526};
Physical Line(1004) = {1528};

Ruled Surface(1801) = {1701};
.
.
.
Ruled Surface(1810) = {1710};

Physical Surface(1) = {S1801, S1802, S1803, S1804, S1805, S1806, S1807, S1808, S1809, S1810};

// ------------------------------------------------------------
```

Points used to define the geometry
- Corners, beginning or end of lines
- Points used as centers to define curves

- **.geo files used as gmsh to define the geometry**

surface_coil_model.geo — Edited

```
// ------------------------------------------------------------------
// CONFORMAL COIL Radius 0.15m, geo created by MARIE coil generator
// ------------------------------------------------------------------

RES = 0.015; // Resolution

Point(1101) = { 0,  0, 0.09, RES};
Point(1102) = { 0,  0, 0.085, RES};
Point(1103) = { 0,  0, -0.085, RES};
.
.
.
Point(1128) = {-0.0005, 0.15, -0.09, RES};

Line(1501) = {1105, 1106};
Circle(1502) = {1106, 1101, 1107};
.
.
.
Line(1532) = {1108, 1119};

Line Loop(1701) = { 1501, 1502, 1503, 1504 };
.
.
.
Line Loop(1710) = { -1514, 1531, -1504, 1532 };

Physical Line(1001) = {1518};
Physical Line(1002) = {1520};
Physical Line(1003) = {1526};
Physical Line(1004) = {1528};

Ruled Surface(1801) = {1701};
.
.
.
Ruled Surface(1810) = {1710};

Physical Surface(1) = {S1801, S1802, S1803, S1804, S1805, S1806, S1807, S1808, S1809, S1810};

// ---------------------------------------------------------------
```

Lines built from the points
- Either straight line between 2 points
- Or curves, between 2 points with centre

- **.geo files used as gmsh to define the geometry**

surface_coil_model.geo — Edited

```
// -----------------------------------------------------------------
// CONFORMAL COIL Radius 0.15m, geo created by MARIE coil generator
// -----------------------------------------------------------------

RES = 0.015; // Resolution

Point(1101) = { 0,   0, 0.09, RES};
Point(1102) = { 0,   0, 0.085, RES};
Point(1103) = { 0,   0, -0.085, RES};
.
.
.
Point(1128) = {-0.0005, 0.15, -0.09, RES};

Line(1501) = {1105, 1106};
Circle(1502) = {1106, 1101, 1107};
.
.
.
Line(1532) = {1108, 1119};

Line Loop(1701) = { 1501, 1502, 1503, 1504 };
.
.
.
Line Loop(1710) = { -1514, 1531, -1504, 1532 };

Physical Line(1001) = {1518};
Physical Line(1002) = {1520};
Physical Line(1003) = {1526};
Physical Line(1004) = {1528};

Ruled Surface(1801) = {1701};
.
.
.
Ruled Surface(1810) = {1710};

Physical Surface(1) = {S1801, S1802, S1803, S1804, S1805, S1806, S1807, S1808, S1809, S1810};

// -------------------------------------------------------------
```

Line loops
- Sets of lines that will define a surface
- Ordering is important to enclose surface

surface coil models

- **.geo files used as gmsh to define the geometry**

surface_coil_model.geo — Edited

```
// ------------------------------------------------------------------
// CONFORMAL COIL Radius 0.15m, geo created by MARIE coil generator
// ------------------------------------------------------------------

RES = 0.015; // Resolution

Point(1101) = { 0,  0, 0.09, RES};
Point(1102) = { 0,  0, 0.085, RES};
Point(1103) = { 0,  0, -0.085, RES};
.
.
.
Point(1128) = {-0.0005, 0.15, -0.09, RES};

Line(1501) = {1105, 1106};
Circle(1502) = {1106, 1101, 1107};
.
.
.
Line(1532) = {1108, 1119};

Line Loop(1701) = { 1501, 1502, 1503, 1504 };
.
.
.
Line Loop(1710) = { -1514, 1531, -1504, 1532 };

Physical Line(1001) = {1518};
Physical Line(1002) = {1520};
Physical Line(1003) = {1526};
Physical Line(1004) = {1528};

Ruled Surface(1801) = {1701};
.
.
.
Ruled Surface(1810) = {1710};

Physical Surface(1) = {S1801, S1802, S1803, S1804, S1805, S1806, S1807, S1808, S1809, S1810};

// ------------------------------------------------------------
```

Physical lines

- Lines that are labeled in mesh
- Used to define the ports
- Each port defined by a pair of physical lines
    - Odd number (e.g. 1001) is positive end
    - Even number (e.g. 1002) is negative end
- Numbering of ports must be consecutive

- **.geo files used as gmsh to define the geometry**

surface_coil_model.geo — Edited

```
// ------------------------------------------------------------------
// CONFORMAL COIL Radius 0.15m, geo created by MARIE coil generator
// ------------------------------------------------------------------

RES = 0.015; // Resolution

Point(1101) = { 0,   0, 0.09, RES};
Point(1102) = { 0,   0, 0.085, RES};
Point(1103) = { 0,   0, -0.085, RES};
.
.
.
Point(1128) = {-0.0005, 0.15, -0.09, RES};

Line(1501) = {1105, 1106};
Circle(1502) = {1106, 1101, 1107};
.
.
.
Line(1532) = {1108, 1119};

Line Loop(1701) = { 1501, 1502, 1503, 1504 };
.
.
.
Line Loop(1710) = { -1514, 1531, -1504, 1532 };

Physical Line(1001) = {1518};
Physical Line(1002) = {1520};
Physical Line(1003) = {1526};
Physical Line(1004) = {1528};

Ruled Surface(1801) = {1701};
.
.
.
Ruled Surface(1810) = {1710};

Physical Surface(1) = {S1801, S1802, S1803, S1804, S1805, S1806, S1807, S1808, S1809, S1810};

// ------------------------------------------------------------
```

Ruled surfaces
- Surfaces generated from loops
- These surfaces will be discretized

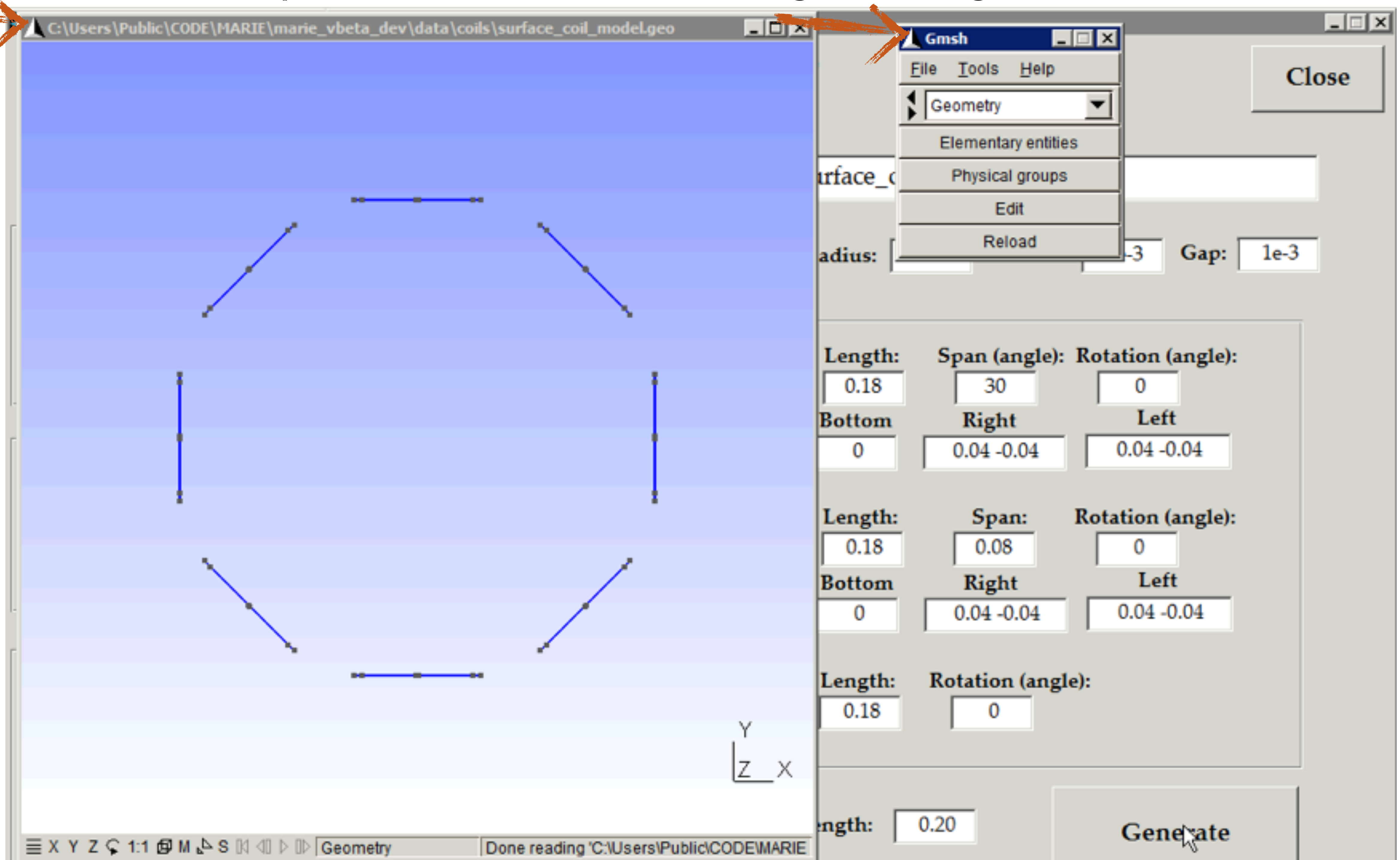
- **.geo files used as gmsh to define the geometry**

surface_coil_model.geo — Edited

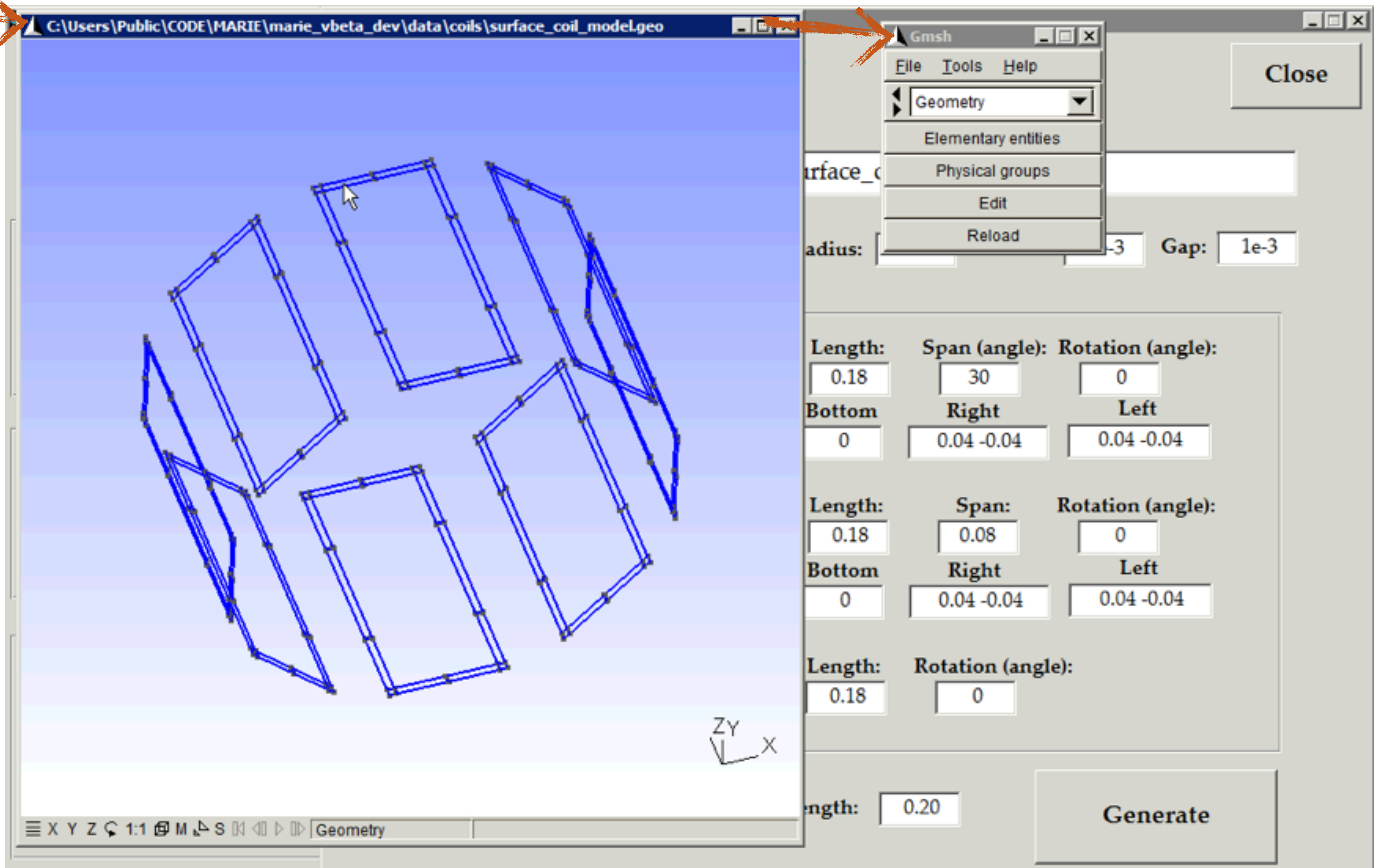
```
// ----------------------------------------------------------------
// CONFORMAL COIL Radius 0.15m, geo created by MARIE coil generator
// ----------------------------------------------------------------

RES = 0.015; // Resolution

Point(1101) = { 0,  0, 0.09, RES};
Point(1102) = { 0,  0, 0.085, RES};
Point(1103) = { 0,  0, -0.085, RES};
.
.
.
Point(1128) = {-0.0005, 0.15, -0.09, RES};

Line(1501) = {1105, 1106};
Circle(1502) = {1106, 1101, 1107};
.
.
.
Line(1532) = {1108, 1119};

Line Loop(1701) = { 1501, 1502, 1503, 1504 };
.
.
.
Line Loop(1710) = { -1514, 1531, -1504, 1532 };

Physical Line(1001) = {1518};
Physical Line(1002) = {1520};
Physical Line(1003) = {1526};
Physical Line(1004) = {1528};

Ruled Surface(1801) = {1701};
.
.
.
Ruled Surface(1810) = {1710};

Physical Surface(1) = {S1801, S1802, S1803, S1804, S1805, S1806, S1807, S1808, S1809, S1810};

// ------------------------------------------------------------
```

Physical surface
- Set of surfaces that form an entity (e.g. a coil)
- These surfaces will be labeled in the mesh (in this case as surface 1)

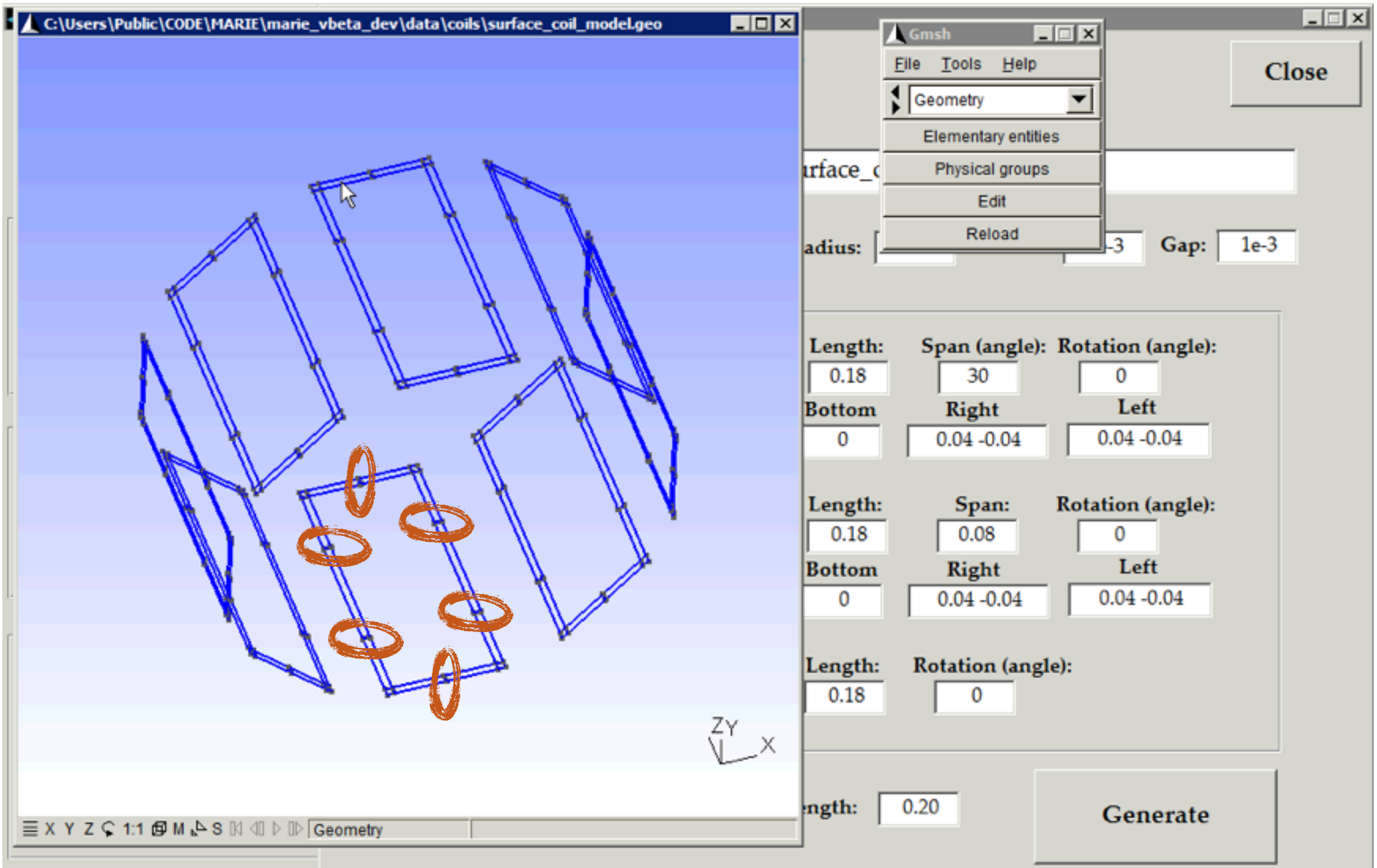
- **.geo files used as gmsh to define the geometry**
  - gmsh has a nice graphic user interface to handle the .geo file
  - MARIE incorporates an interface when generating coils



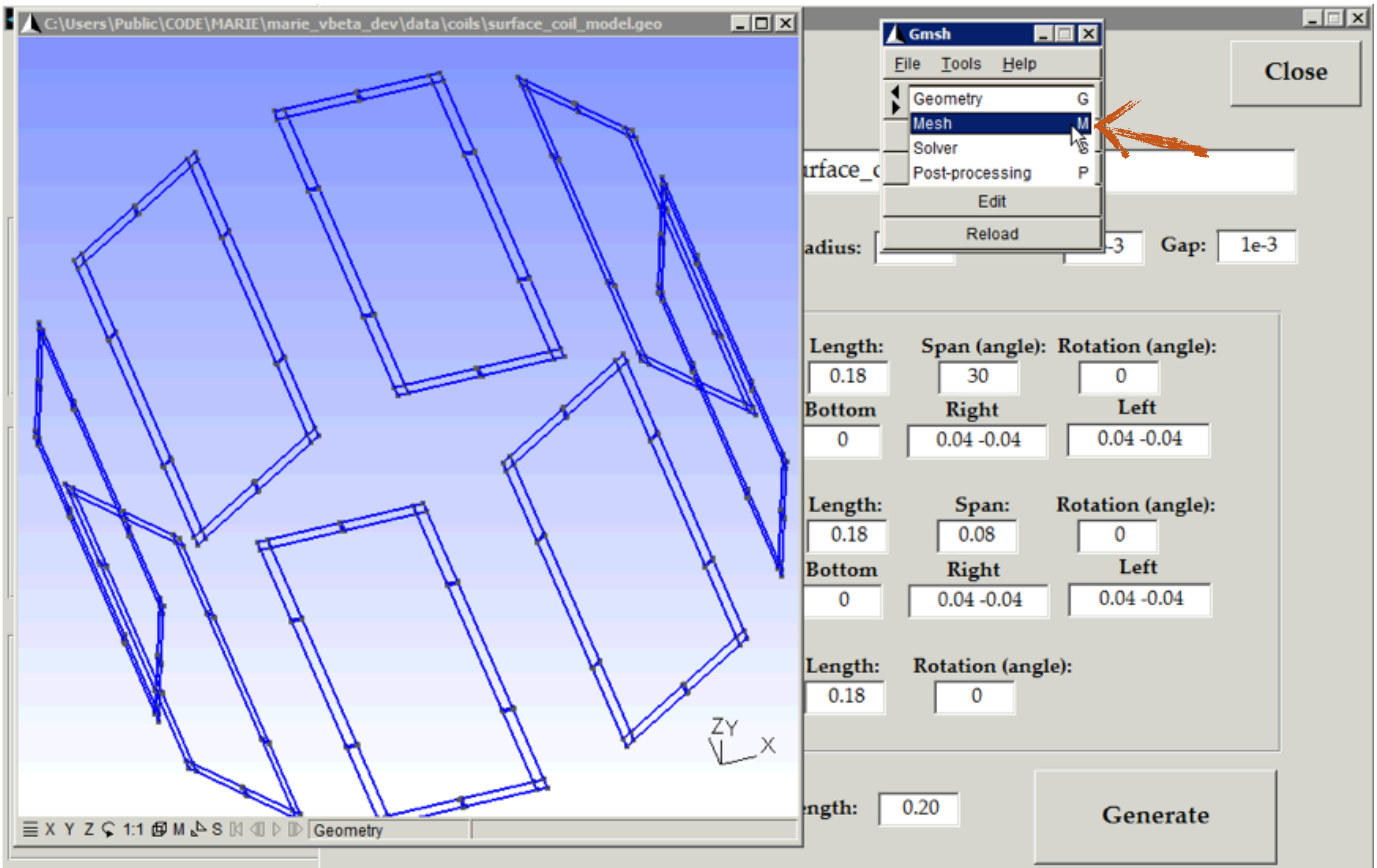
- **.geo files used as gmsh to define the geometry**
  - 3D visual environment to visualize the geometry



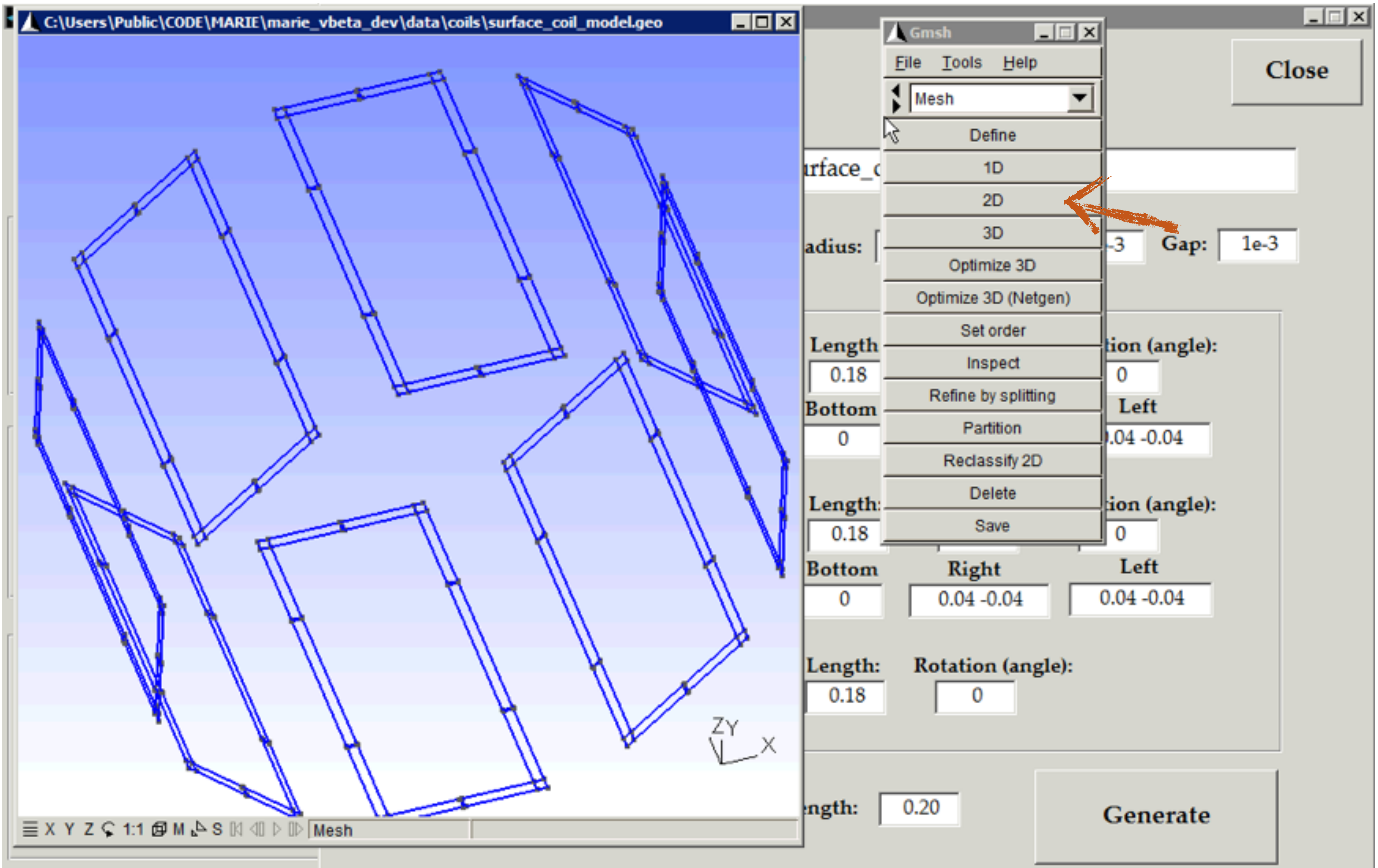
- **.geo files used as gmsh to define the geometry**
  - note ports (gaps) positions in each coil: these lines are physical



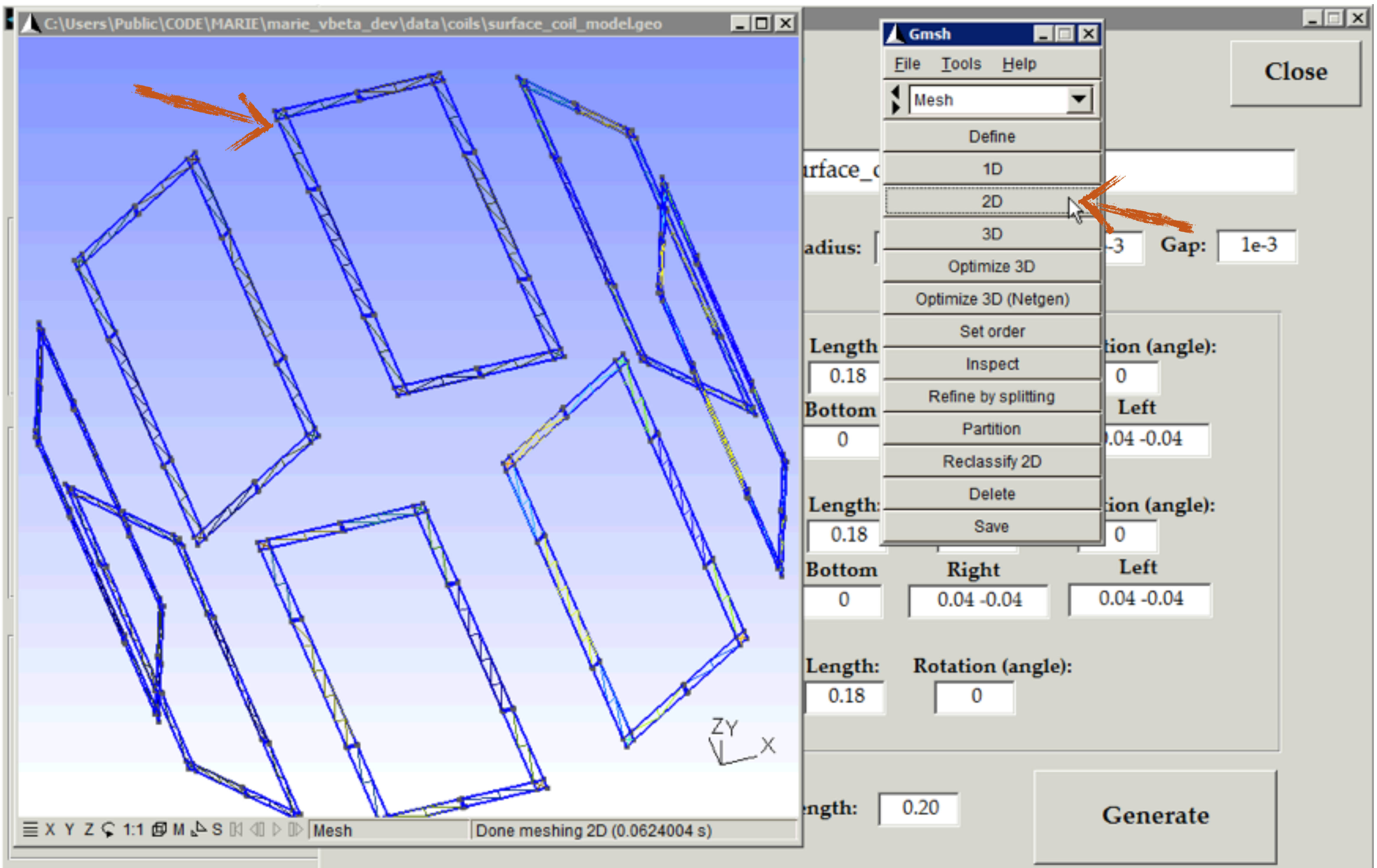
- **.geo files used as gmsh to define the geometry**
  - allows you to automatically generate a triangular mesh from .geo



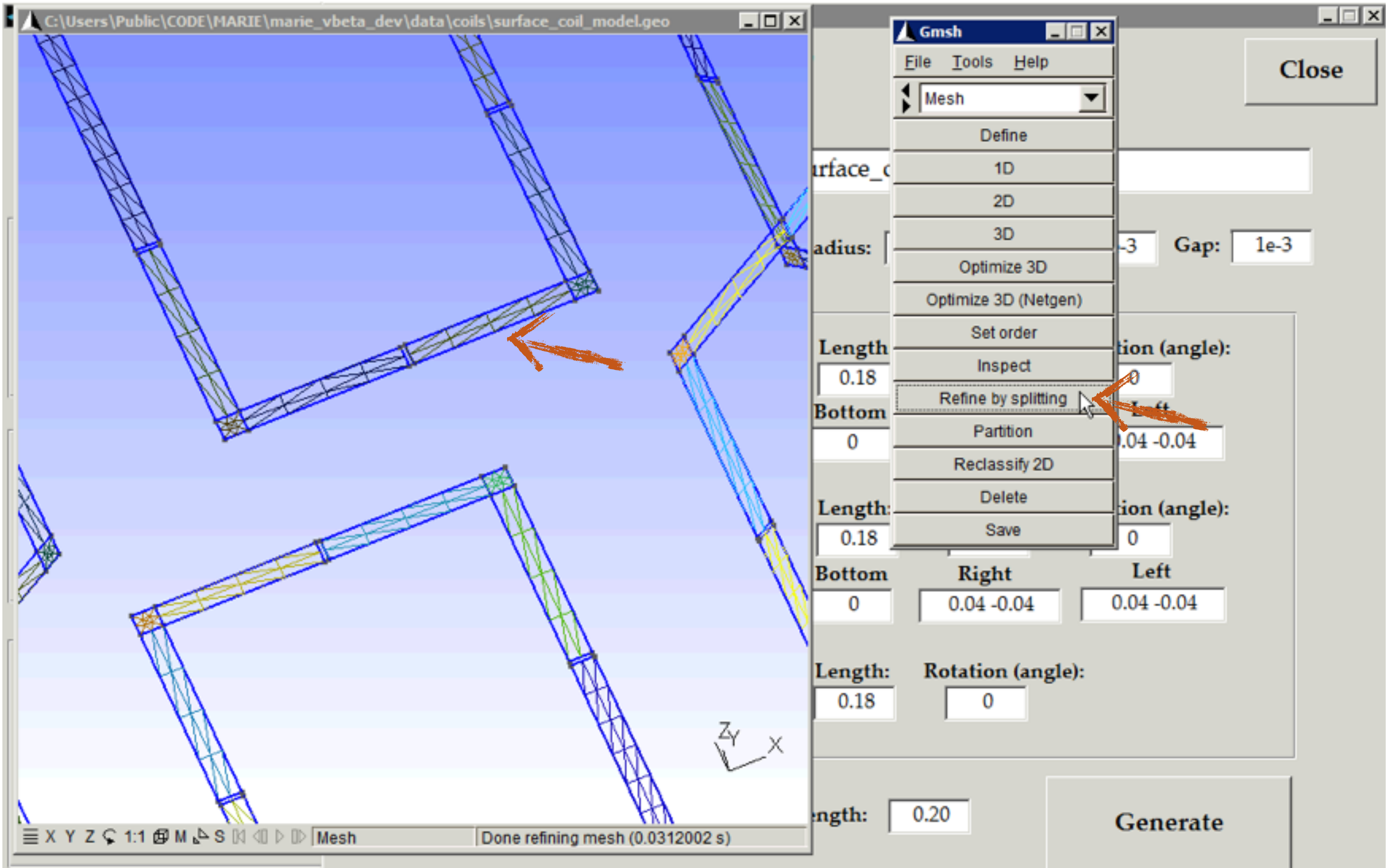
- **.geo files used as gmsh to define the geometry**
  - go to 2D mesh to perform the tessellation



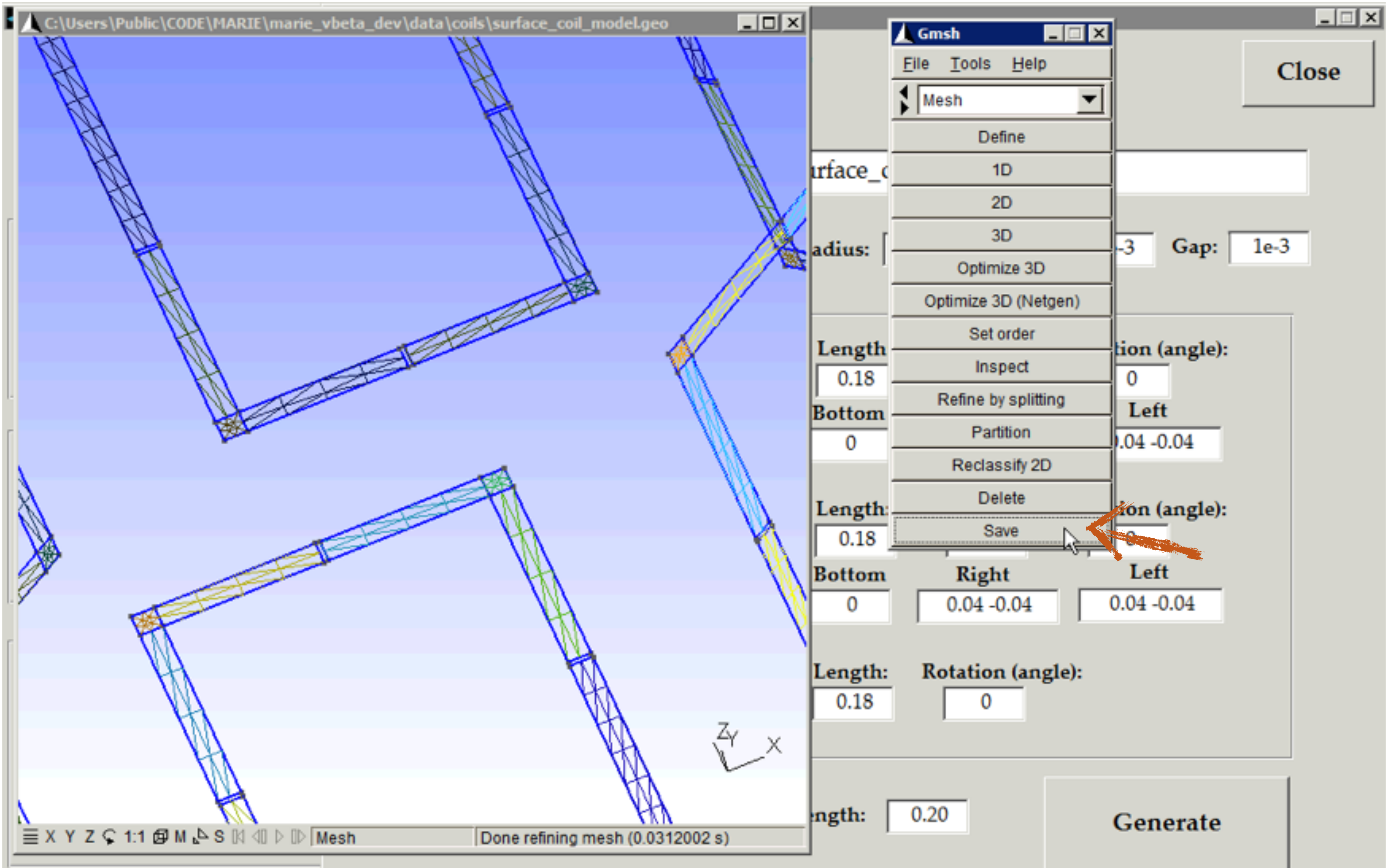
- **.geo files used as gmsh to define the geometry**
  - the ruled surfaces are discretized



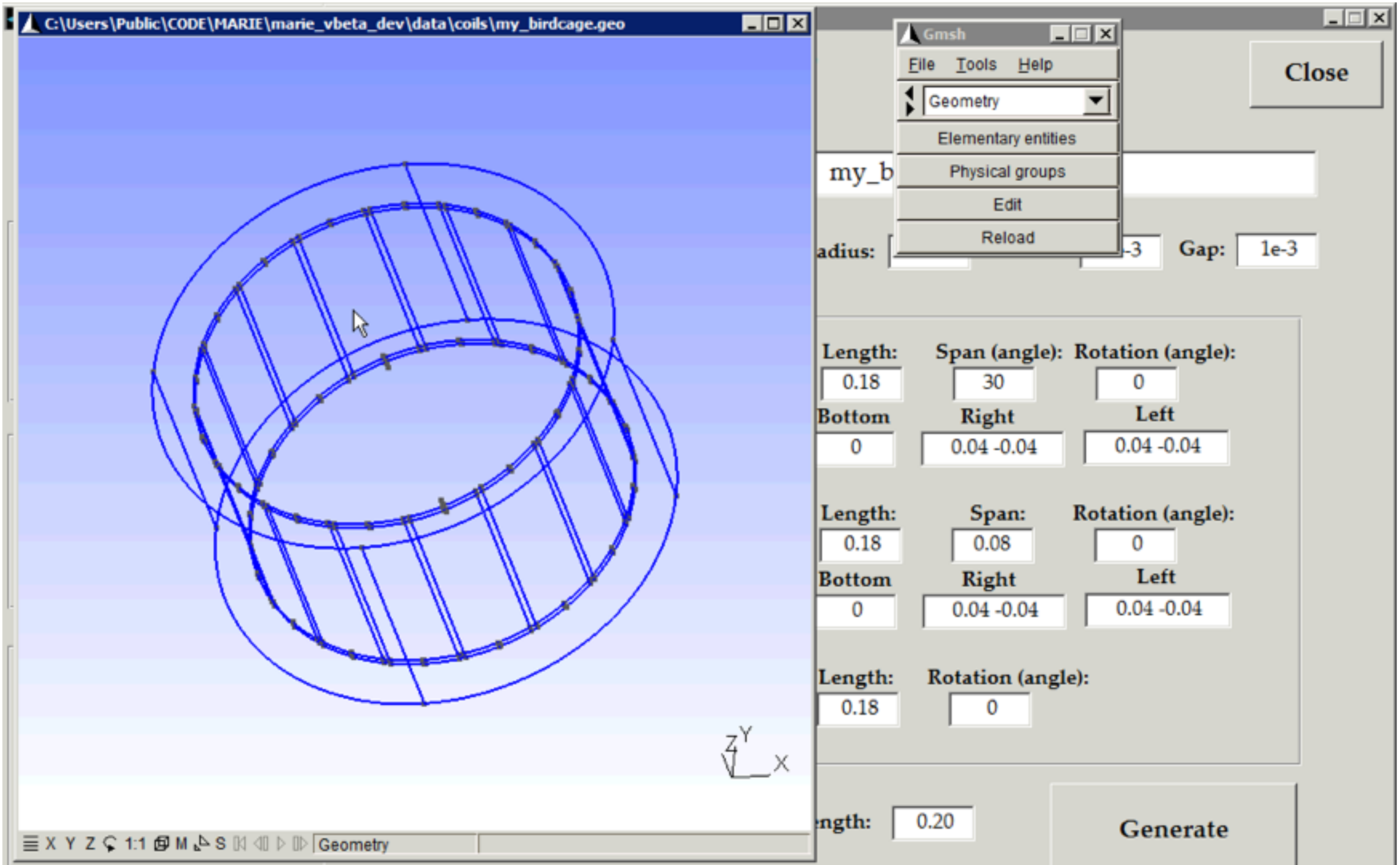
- **.geo files used as gmsh to define the geometry**
  - .geo is prepared to generate a coarse discretization: refine by splitting
    - to generate a finer discretization (better accuracy at computational cost)



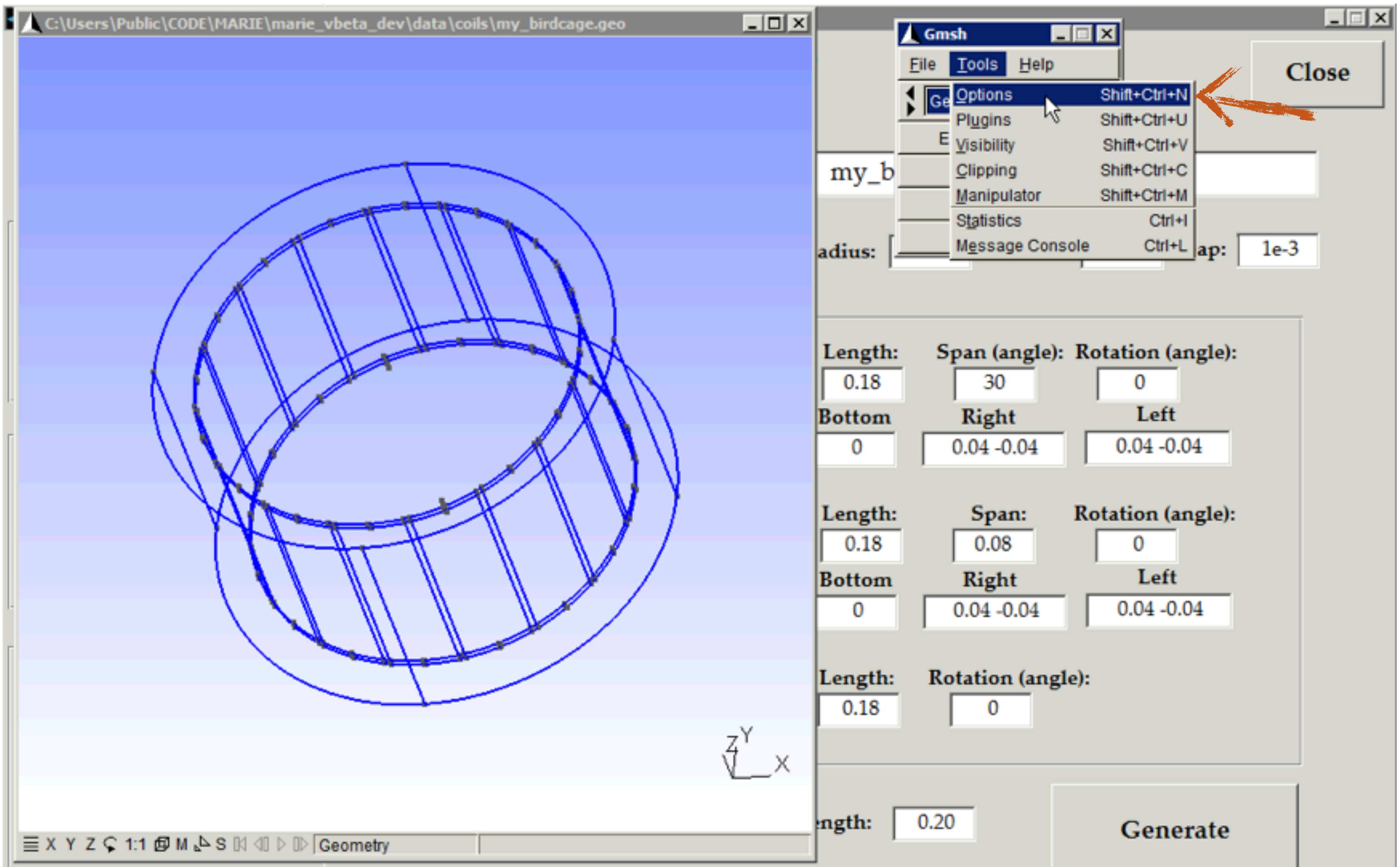
- **.geo files used as gmsh to define the geometry**
  - push save to save the corresponding files
    - it will generate the .msh file, along with the .geo and .smm



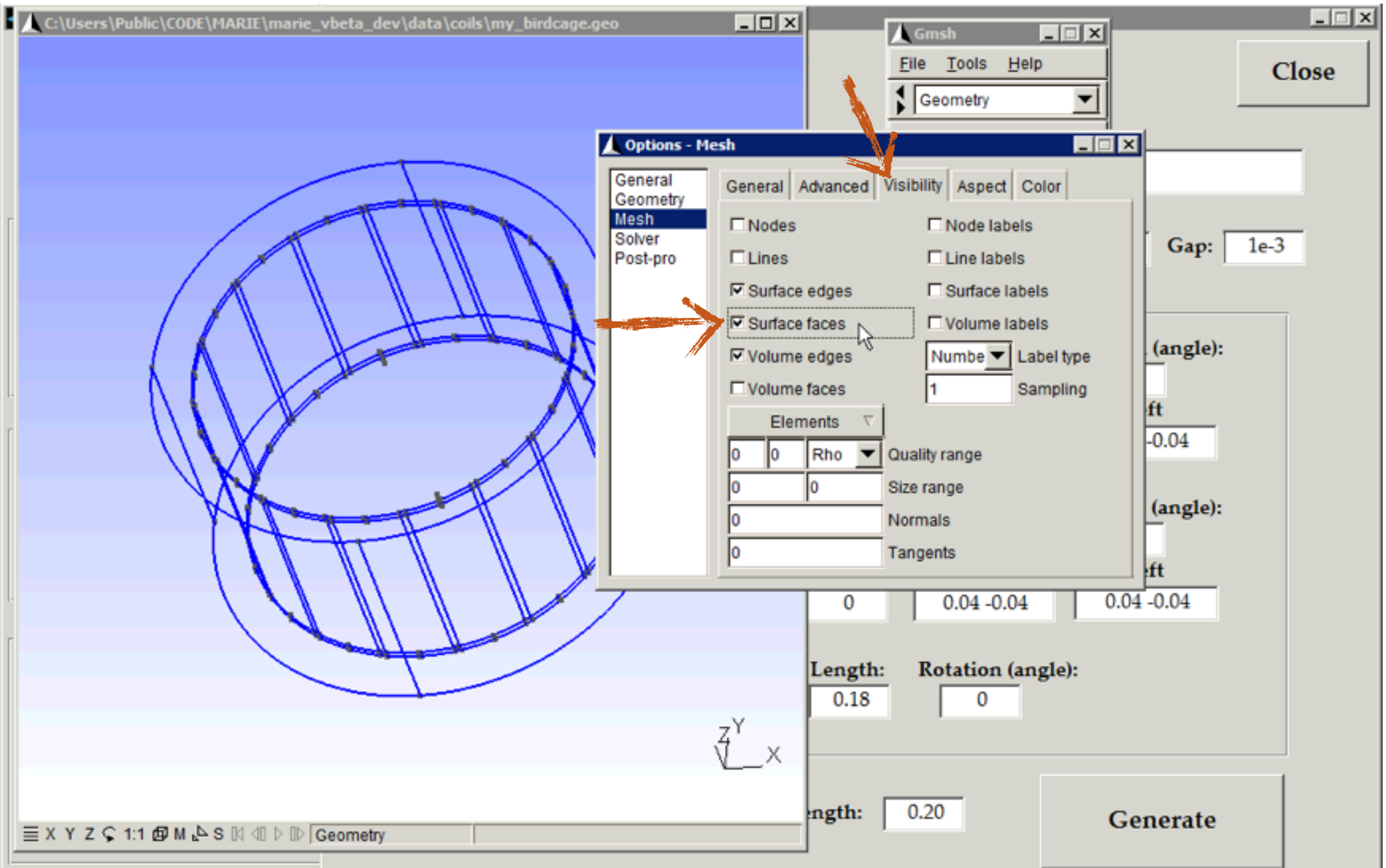
- **.geo files used as gmsh to define the geometry**
  - can generate more complex structures
    - can also merge .geo or .msh files



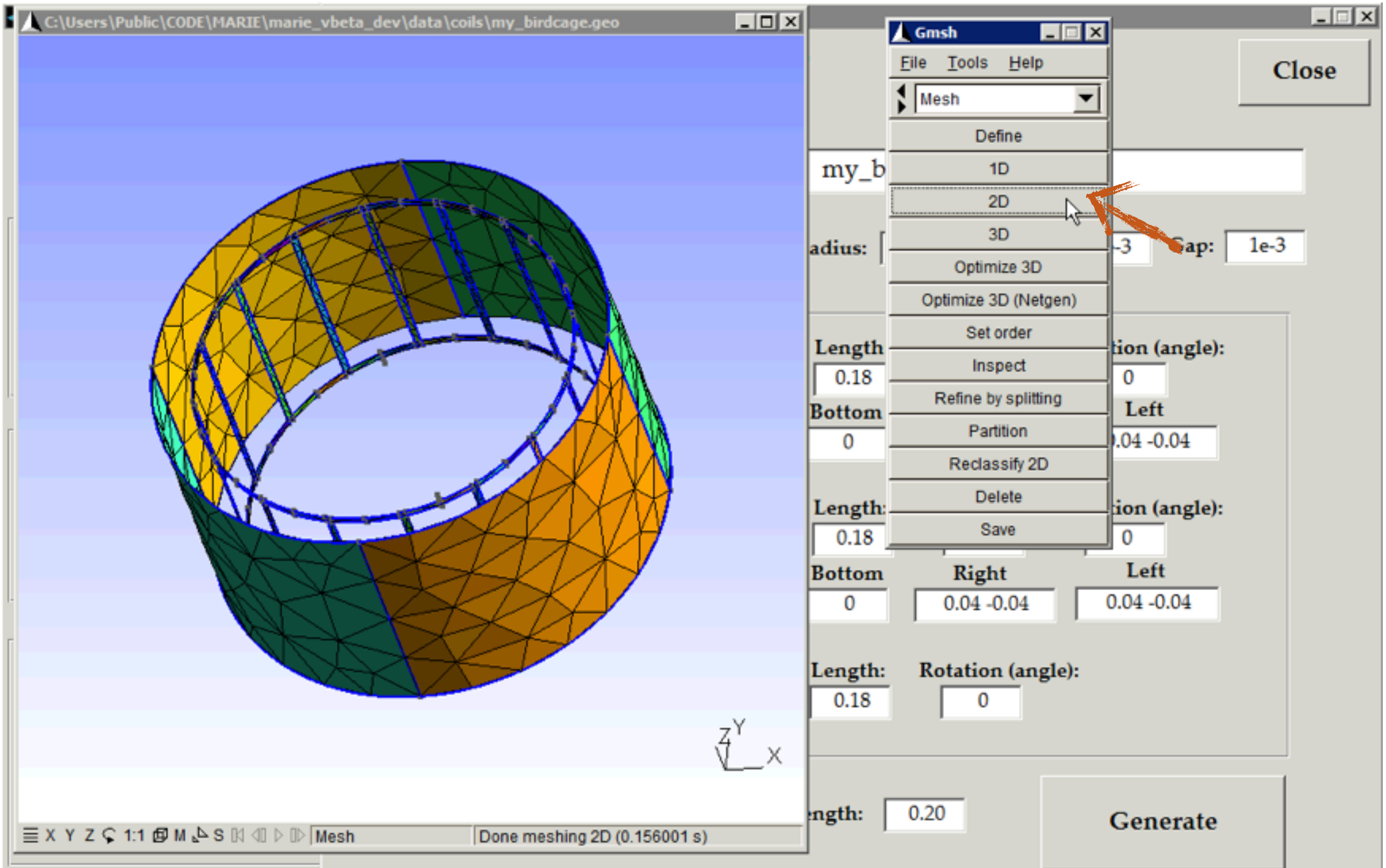
- **.geo files used as gmsh to define the geometry**
  - we can modify the visibility options to show the geometry clearer



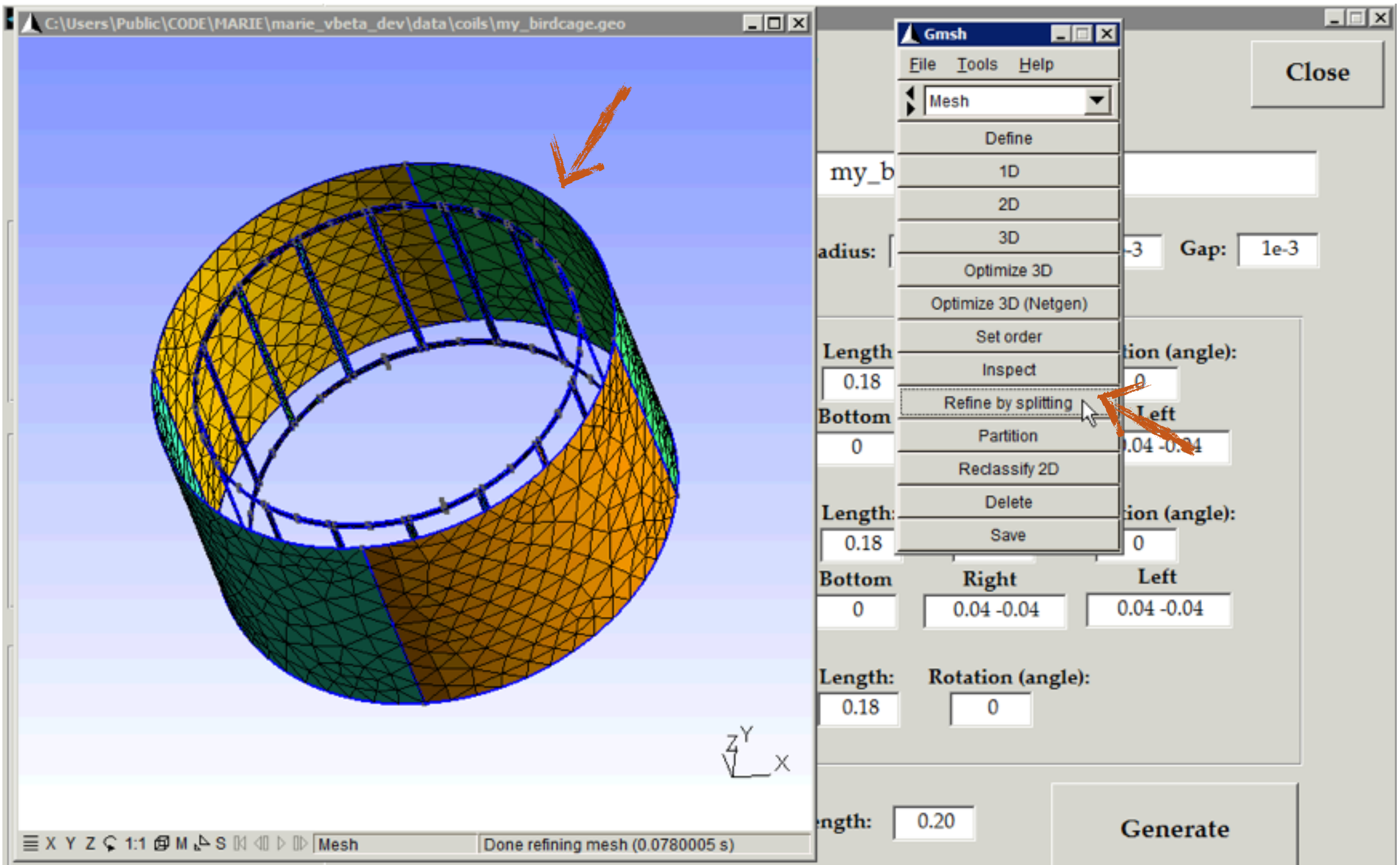
- **.geo files used as gmsh to define the geometry**
  - we can modify the visibility options to show the geometry clearer



- **.geo files used as gmsh to define the geometry**
  - and the discretization



- **.geo files used as gmsh to define the geometry**
  - again, refine by splitting to generate finer regular tessellation



surface coil models

- **.msh files, generated by gmsh, with geometry info of tessellation**

surface_coil_model.msh — Edited

```
$MeshFormat
2.2 0 8
$EndMeshFormat
$Nodes
76
1 -0.0388228567653781 0.1448888739433603 0.09
2 -0.03397165338566188 0.1461024529781908 0.09
.
.
.
75 0.03640231270882265 0.145515881021451 -0.08749999999999999
76 -0.03640231270882265 0.145515881021451 -0.08749999999999999
$EndNodes
$Elements
80
1 1 2 1001 1518 17 18
2 1 2 1002 1520 19 20
3 1 2 1003 1526 21 22
4 1 2 1004 1528 23 24
5 2 2 1000 1801 2 73 1
6 2 2 1000 1801 4 73 3
.
.
.
80 2 2 1000 1810 58 67 68
$EndElements
```

Mesh format and info
- $ is comment

- **.msh files, generated by gmsh, with geometry info of tessellation**

surface_coil_model.msh — Edited

```
$MeshFormat
2.2 0 8
$EndMeshFormat
$Nodes
76
1 -0.03882285676537811 0.1448888739433603 0.09
2 -0.03397165338566188 0.1461024529781908 0.09
.
.
.
75 0.03640231270882265 0.145515881021451 -0.08749999999999999
76 -0.03640231270882265 0.145515881021451 -0.08749999999999999
$EndNodes
$Elements
80
1 1 2 1001 1518 17 18
2 1 2 1002 1520 19 20
3 1 2 1003 1526 21 22
4 1 2 1004 1528 23 24
5 2 2 1000 1801 2 73 1
6 2 2 1000 1801 4 73 3
.
.
.
80 2 2 1000 1810 58 67 68
$EndElements
```

Begin and number of nodes

- **.msh files, generated by gmsh, with geometry info of tessellation**

surface_coil_model.msh — Edited

```
$MeshFormat
2.2 0 8
$EndMeshFormat
$Nodes
76
1 -0.03882285676537811 0.1448888739433603 0.09
2 -0.03397165338566188 0.1461024529781908 0.09
.
.
.
75 0.03640231270882265 0.145515881021451 -0.08749999999999999
76 -0.03640231270882265 0.145515881021451 -0.08749999999999999
$EndNodes
$Elements
80
1 1 2 1001 1518 17 18
2 1 2 1002 1520 19 20
3 1 2 1003 1526 21 22
4 1 2 1004 1528 23 24
5 2 2 1000 1801 2 73 1
6 2 2 1000 1801 4 73 3
.
.
.
80 2 2 1000 1810 58 67 68
$EndElements
```

For each node

- x, y and z position of node

- **.msh files, generated by gmsh, with geometry info of tessellation**

surface_coil_model.msh — Edited

```
$MeshFormat
2.2 0 8
$EndMeshFormat
$Nodes
76
1 -0.03882285676537811 0.1448888739433603 0.09
2 -0.03397165338566188 0.1461024529781908 0.09
.
.
.
75 0.03640231270882265 0.145515881021451 -0.08749999999999999
76 -0.03640231270882265 0.145515881021451 -0.08749999999999999
$EndNodes
$Elements
80
1 1 2 1001 1518 17 18
2 1 2 1002 1520 19 20
3 1 2 1003 1526 21 22
4 1 2 1004 1528 23 24
5 2 2 1000 1801 2 73 1
6 2 2 1000 1801 4 73 3
.
.
.
80 2 2 1000 1810 58 67 68
$EndElements
```

End nodes and begin elements

- Elements in our case: lines and triangles

- gmsh also generates other types

Number of elements in tessellation

- **.msh files, generated by gmsh, with geometry info of tessellation**

surface_coil_model.msh — Edited

```
$MeshFormat
2.2 0 8
$EndMeshFormat
$Nodes
76
1 -0.03882285676537811 0.1448888739433603 0.09
2 -0.03397165338566188 0.1461024529781908 0.09
.
.
.
75 0.03640231270882265 0.145515881021451 -0.08749999999999999
76 -0.03640231270882265 0.145515881021451 -0.08749999999999999
$EndNodes
$Elements
80
1 1 2 1001 1518 17 18
2 1 2 1002 1520 19 20
3 1 2 1003 1526 21 22
4 1 2 1004 1528 23 24
5 2 2 1000 1801 2 73 1
6 2 2 1000 1801 4 73 3
.
.
.
80 2 2 1000 1810 58 67 68
$EndElements
```

For each element:
- **Element number**
- **Element type code (1 lines, 2 triangles)**
- **Element physical number (to which set)**
  - **For lines, the physical lines: ports**
  - **For triangles, physical surface to which belong**
- **Element parent**
  - **Number of line**
  - **Ruled surface to which triangle belongs**
- **Nodes defining the element**
  - **2 for line**
  - **3 for triangle**

- **.msh files, generated by gmsh, with geometry info of tessellation**

surface_coil_model.msh — Edited

```
$MeshFormat
2.2 0 8
$EndMeshFormat
$Nodes
76
1 -0.03882285676537811 0.1448888739433603 0.09
2 -0.03397165338566188 0.1461024529781908 0.09
.
.
.
75 0.03640231270882265 0.145515881021451 -0.08749999999999999
76 -0.03640231270882265 0.145515881021451 -0.08749999999999999
$EndNodes
$Elements
80
1 1 2 1001 1518 17 18
2 1 2 1002 1520 19 20
3 1 2 1003 1526 21 22
4 1 2 1004 1528 23 24
5 2 2 1000 1801 2 73 1
6 2 2 1000 1801 4 73 3
.
.
.
80 2 2 1000 1810 58 67 68
$EndElements
```

End the elements and of file

surface coil models

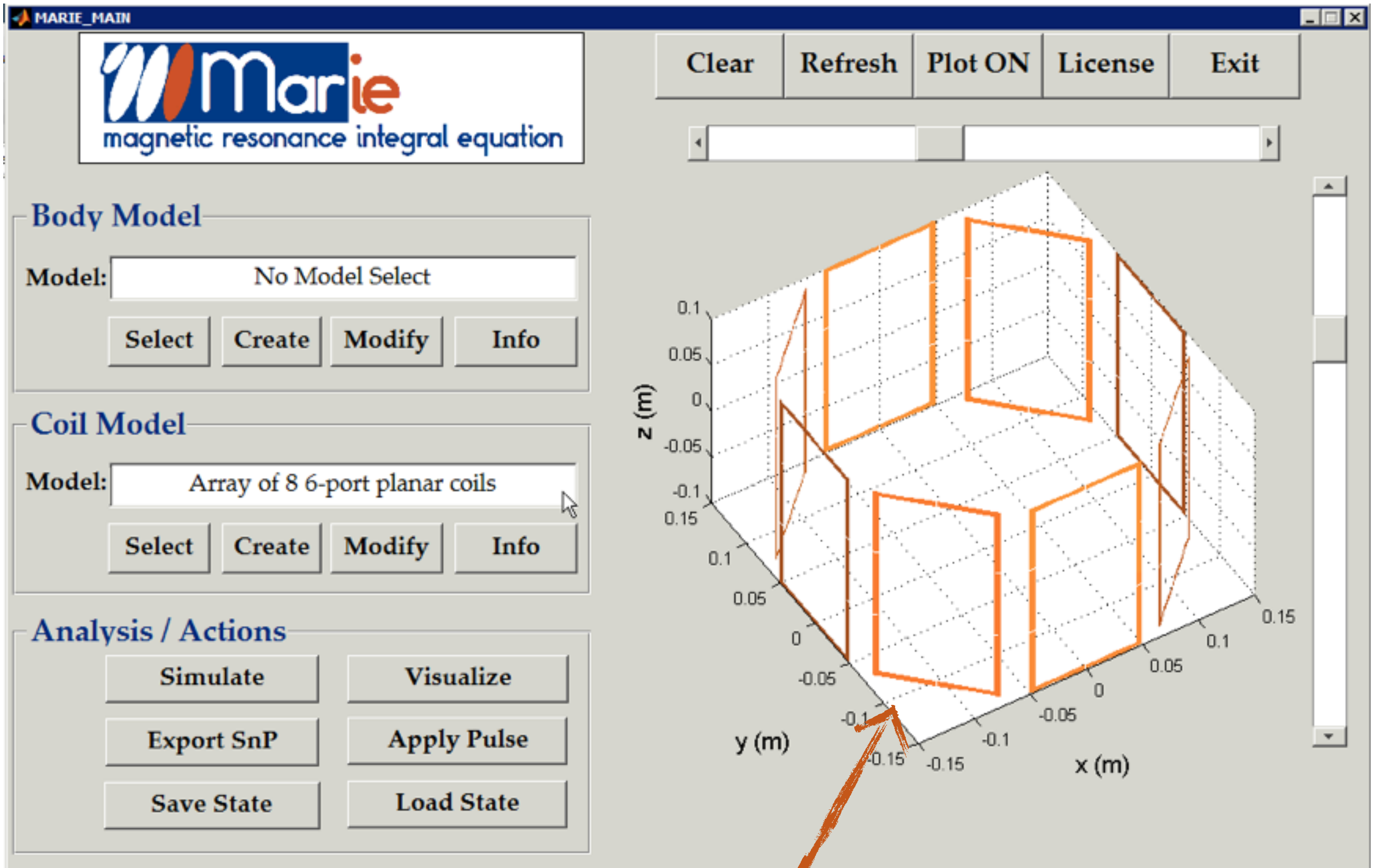
- **Reminder: .smm (surface marie model) file defining surface coil**
  - these are just text files with 3 lines

circularloop_planar.smm

```
Circular Loop, R 40mm, Dist 120mm
1.2579e-08 0.0005
CircularLoop.msh
```

  - first line is the model name and/or description
  - second line is two numbers
    - first the resistivity of the material
    - second the thickness of the planar material
  - third line is the name of the .msh file with the geometry tessellation

- **Marie reads this file and parses the .msh file**



wire coil models

- **What is a wire coil model in MARIE?**
  - a structure based on wire conductors (e.g. copper wire)
    - cylindrical cross section, thin in comparison with length
  - surface integral equation (SIE) methods are applied to model it

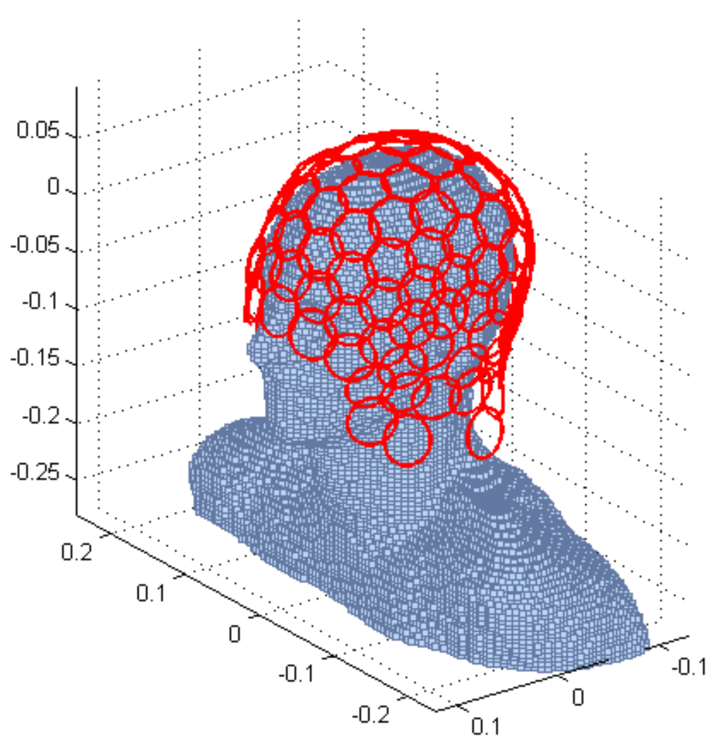
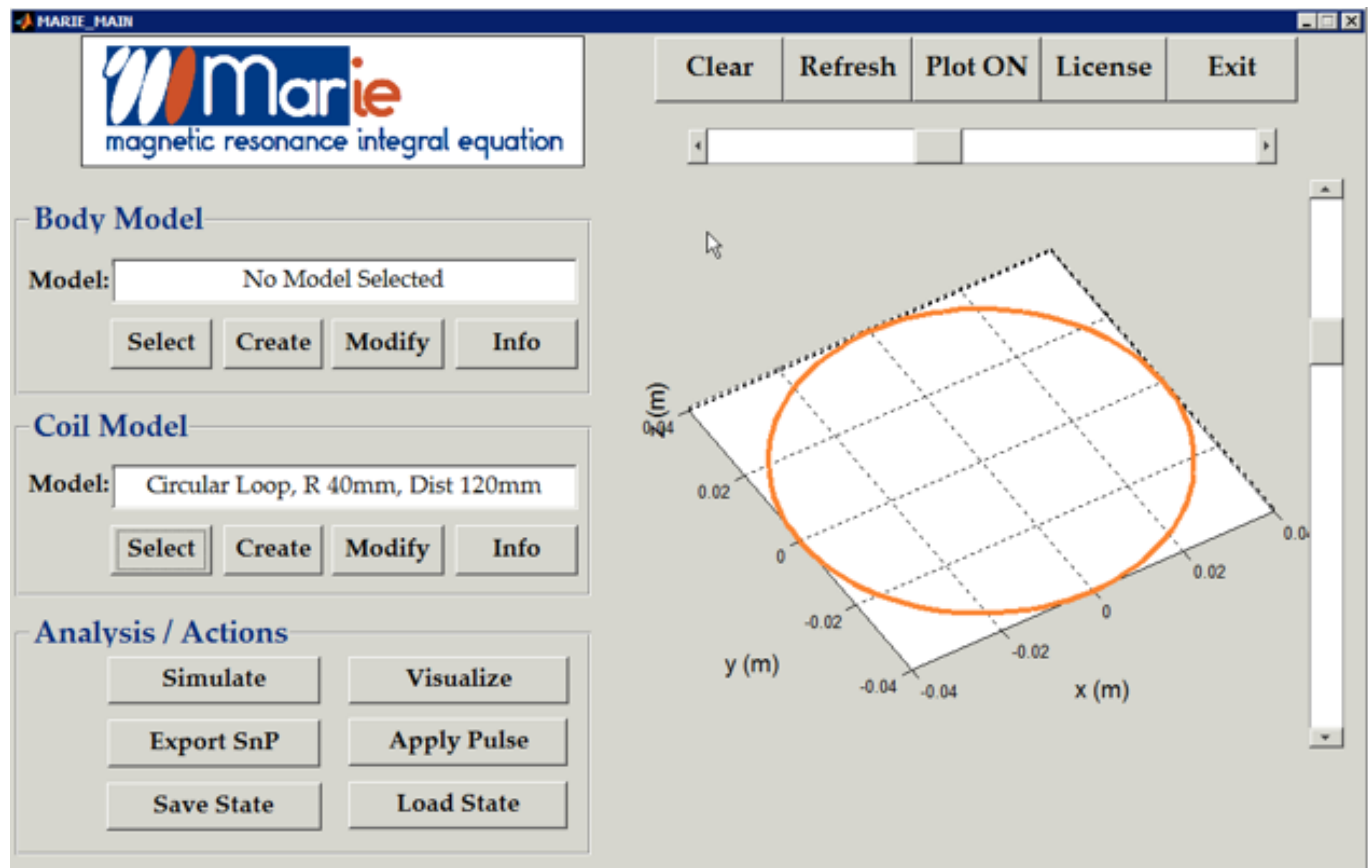
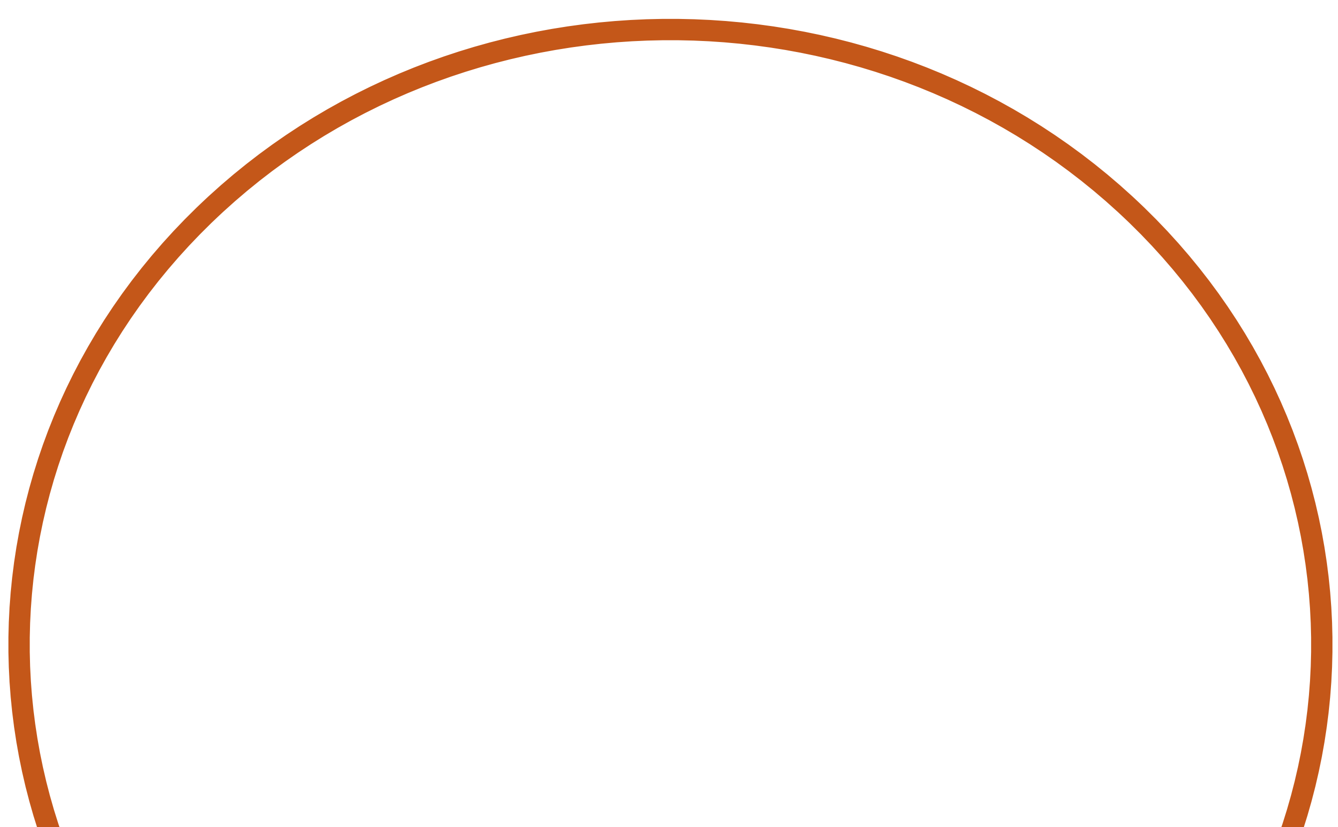


image courtesy of
Martinos Center, MGH



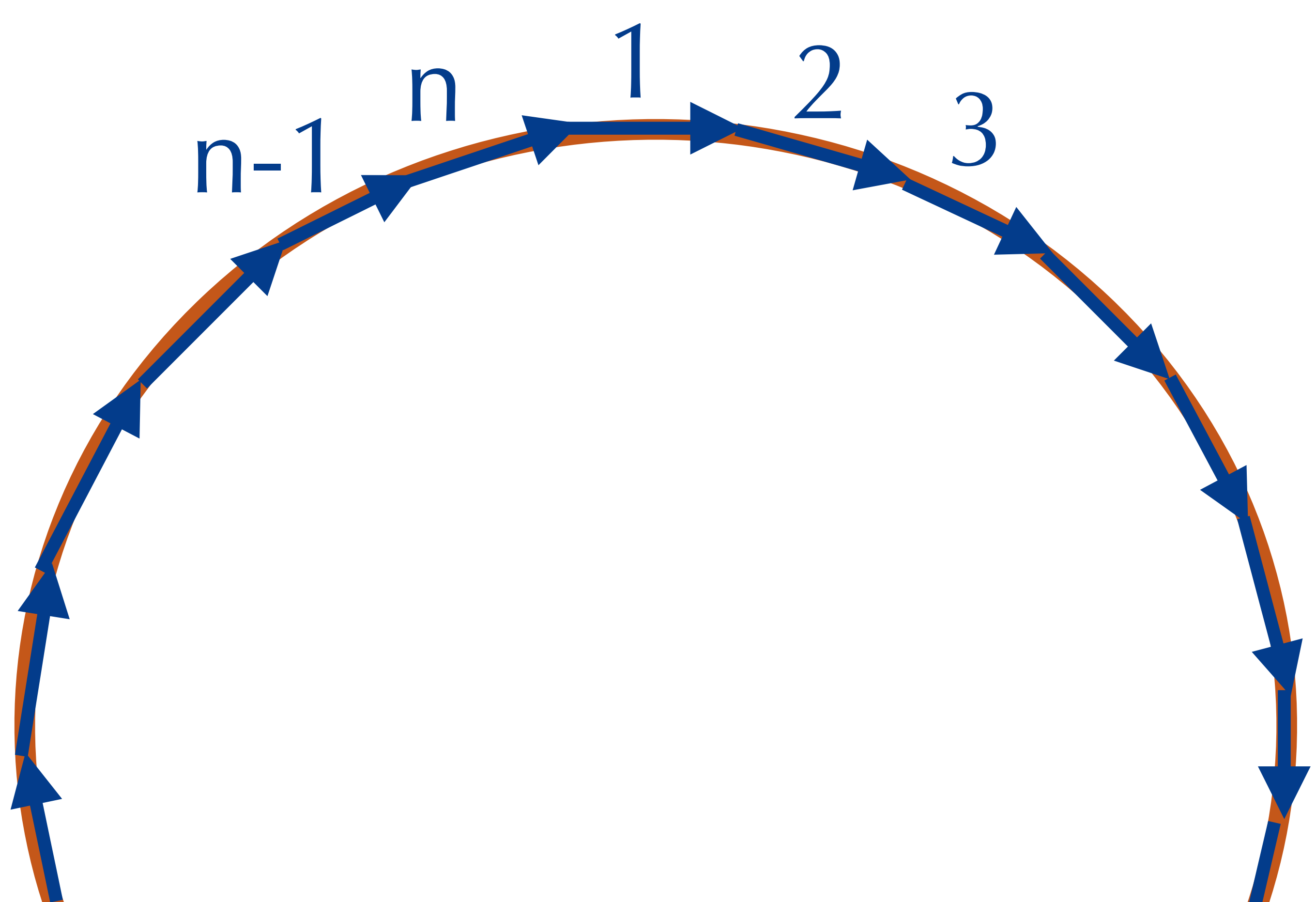
- **Discretization and modeling methodology**
  - simple wire segments
  - current constant in each segment
  - integration is done on each segment, point matching approaches
  - Simple to generate:
    - initial and end point of each segment
  - Restrictions:
    - each segment must be a thin wire: length times larger than cross section

- **Formulation, port and basis definitions**
  - given some wire coil

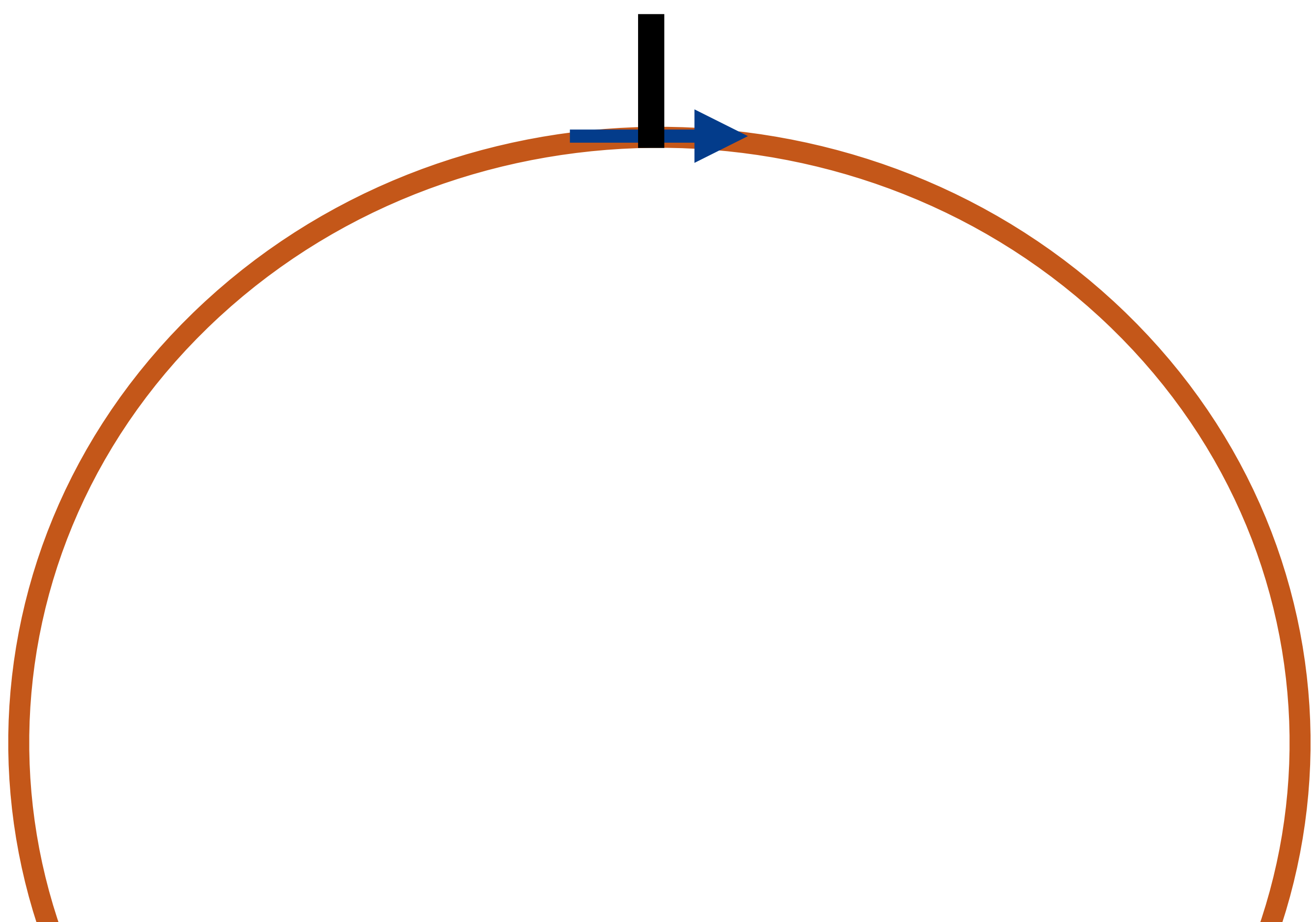


- **Formulation, port and basis definitions**
  - discretize the conductors into small segments

- **Formulation, port and basis definitions**
  - discretize the conductors into small segments
  - combine with desired basis functions
    - each segment is associated with at unknown
    - unknowns: coefficients of the basis functions, related to currents



- **Formulation, port and basis definitions**
  - discretize the conductors into small segments
  - combine with desired basis functions
    - each segment is associated with at unknown
    - unknowns: coefficients of the basis functions, related to currents
  - Some segments are defined as ports
    - delta gap excitation: constant external excitation in the segment



- **Formulation, port and basis definitions**
  - apply the corresponding numerical computations to form the system

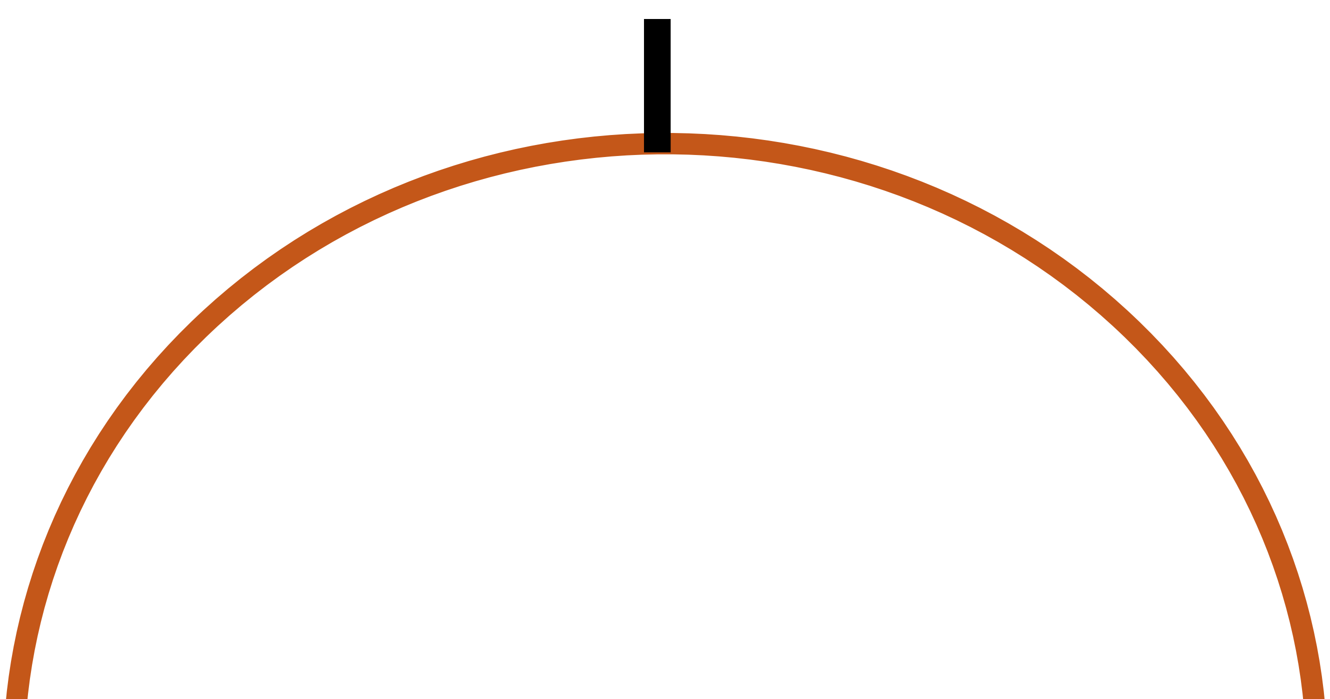
[Zc,F]=Assembly_WIE(COIL,freq)

  - solve the system to obtain the port impedance and basis

[Zparameters,J]=WIE_Solver(COIL,freq)

  - obtain the radiated fields in any point outside the coils

[E,H]=WIE_Radiate(COIL,J,freq,r)



wire coil models

- **MARIE uses .wmm (wire marie model) file defining wire coil**
  - these are just text files with 3 lines

CircularLoop_Wire.wmm

```
Circular Loop, R 40mm, Dist 120mm
1.2579e-08 0.0005
CircularLoop.wsd
```

- first line is the model name and/or description
- second line is two numbers
  - first the resistivity of the material
  - second the thickness of the planar material
- third line is the name of the .wsd (wire segment discretization)
  - file with the wire discretization

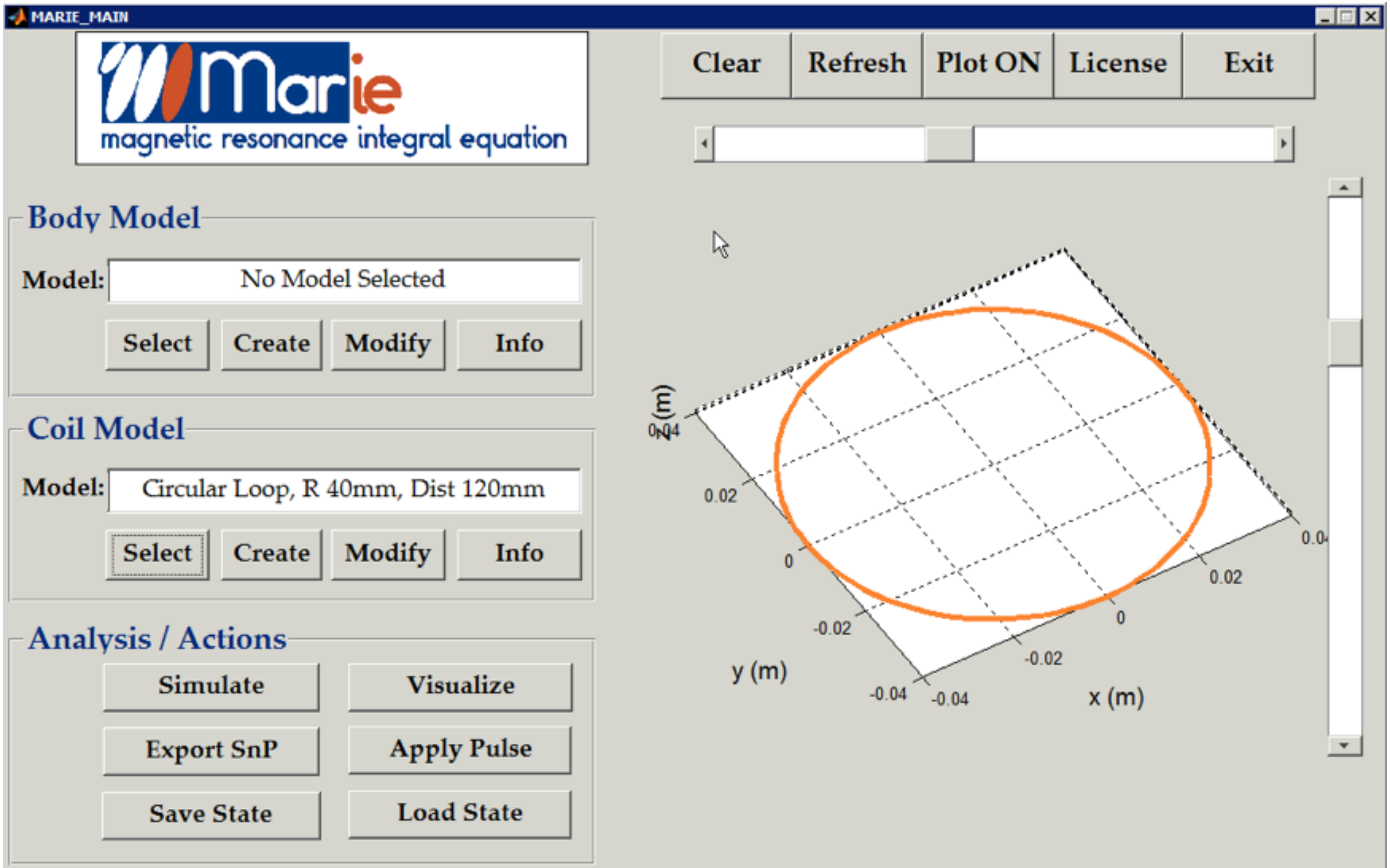
- **.wsd files, each line:**
  - coordinates of pair of points defining a segment (begin and end)
  - flag indicating the port number, or zero if internal segment
  - last end point should match first begin point for closed loop coils

CircularLoop.wsd

```
4.0000000e-02  0.0000000e+00   1.2000000e-01   3.9921069e-02   2.5116208e-03   1.2000000e-01   1.0000000e+00
3.9921069e-02  2.5116208e-03   1.2000000e-01   3.9684588e-02   5.0133293e-03   1.2000000e-01   0.0000000e+00
3.9684588e-02  5.0133293e-03   1.2000000e-01   3.9291490e-02   7.4952526e-03   1.2000000e-01   0.0000000e+00
3.9291490e-02  7.4952526e-03   1.2000000e-01   3.8743326e-02   9.9475955e-03   1.2000000e-01   0.0000000e+00

   ...

3.9921069e-02  -2.5116208e-03  1.2000000e-01   4.0000000e-02   0.0000000e+00   1.2000000e-01   0.0000000e+00
```

x y z coordinates of begin point **x y z coordinates of end point** **port flag**

- **Marie reads the .wmm file and parses the .wsd file**



coil data structures

- **COIL data structure**
  - has geometric information of the coil array
  - two different kind of coil models are used by MARIE

  - surface coil models
    - planar metal structures, small trace width and negligible thickness
    - typical TX coil arrays, including shield
    - modeled by Surface Integral Equation approach

  - wire coil models
    - wire antennas, a thin cylindric wire
    - typical RX arrays are usually wire loops
    - modeled by Wire Integral Equation approach

- **COIL data structure**

Command Window

```
>> whos
  Name         Size                 Bytes  Class     Attributes

  COIL         1x1                    386  struct    global
  FIGIDX       1x1                      8  double    global
  PULSE        1x1                   1760  struct    global
  RHBM         1x1                   1090  struct    global
  SOL          1x1                   1408  struct    global
  p            1x6341               12682  char

>> COIL

COIL =

    name: 'No Model Selected'
    type: []

>> 
```

- **COIL structure fields (common to both cases)**

  .name - string with name of description of the coil

  .type - character that identifies the type of coil

  - 'N' if no coil is loaded yet
  - 'S' for surface coil model
  - 'W' for wire coil model

  .Rhocoil - resistivity of the coil material

  - 0 indicates perfect electric conductor (PEC)
  - note: coil losses in the surface coil model are not yet included

- **COIL data structure: surface coils**

Command Window

```
  COIL        1x1                2223924  struct    global
  FIGIDX      1x1                      8  double    global
  PULSE       1x1                   1760  struct    global
  RHBM        1x1                   1090  struct    global
  SOL         1x1                   1408  struct    global
  p           1x6341               12682  char

>> COIL

COIL =

          name: 'Circular Loop, R 40mm, Dist 120mm'
          type: 'S'
       Rhocoil: 1.257900000000000e-08
     Thickness: 5.000000000000000e-04
         index: [840x1 double]
          etod: [3x512 double]
          node: [3x330 double]
          edge: [2x840 double]
          elem: [4x512 double]
    index_elem: [1x1 struct]
            Ct: [3x512 double]
            Ln: [3x512 double]
            Pn: [3x3x512 double]
          port: [1x2 struct]
         Pcoil: []
         Ncoil: []
         Dwire: []

>>
```

- **COIL structure fields (for surface coil model)**

  .thickness - thickness of the planar coil

  .index - indexes of the tessellation

  .etod - orientation of the edges of the tessellation

  .node - nodes of the tessellation

  .edge - edges of the tessellation

  .elem - element of the tessellation

  .index_elem - index of each element

  .Ct - baricenters of the triangles

  .Ln - length of the triangles

  .Pn - perpendiculars of the triangles

  .port - port information

  - this information is used by the Surface Integral Equation modeling

- **COIL data structure: wire coils**

```
Command Window
  COIL         1x1                  7892  struct     global
  FIGIDX       1x1                     8  double     global
  PULSE        1x1                  1760  struct     global
  RHBM         1x1                  1090  struct     global
  SOL          1x1                  1408  struct     global
  p            1x6341              12682  char

>> COIL

COIL =

          name: 'Circular Loop, R 40mm, Dist 120mm'
          type: 'W'
       Rhocoil: 1.257900000000000e-08
     Thickness: []
         index: []
          etod: []
          node: []
          edge: []
          elem: []
    index_elem: []
            Ct: []
            Ln: []
            Pn: []
         Pcoil: [100x3 double]
         Ncoil: [100x3 double]
         Dwire: 5.000000000000000e-04
          port: [2x1 double]

fx >>
```

- **COIL structure fields (for wire coil model)**

  .Pcoil - positive edges of segments

  .Ncoil - negative edges of segments

  .Dwire - diameter of the wire

  .port - port information

- this information is used by the Wire Integral Equation modeling enjoy Marie