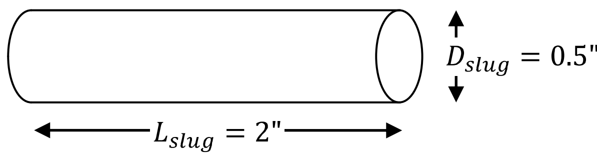
Using FEMM to analyze a coil from a coil gun

David Meeker, who works for the IEEE, has developed and maintains a wonderful program which simulates magnetics, electrostatics, heat flow and current flow. It is called FEMM, which is short for Finite Element Method Magnetics. Not only is it a free download from the internet (www.femm.info), but it is easy to learn and use. FEMM can be used in interactive mode for simple configurations. More complex configurations are better analyzed if the details are described using the scripting language Lua.

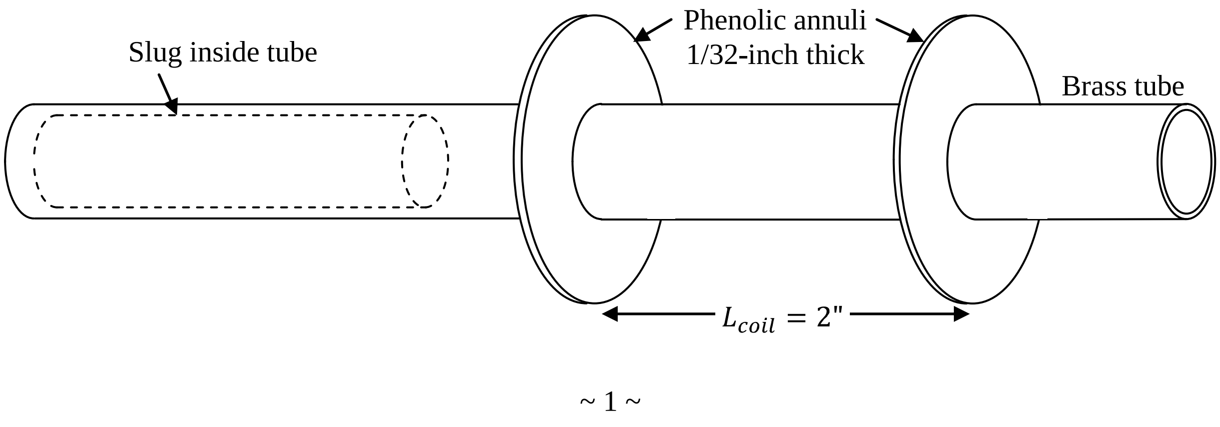
In this paper, I will describe a Lua script which constructs and analyzes a typical coil from an axial coil gun. The complete script is listed in Appendix "A". In the following paragraphs, I will describe the configuration I have in mind.

I intend to shoot a 50-caliber projectile, or slug. It is a round cylinder one-half inch in diameter and two inches long. For the purposes of this paper, I will assume it is a simple cylinder without a rounded nose. I will also assume that it is made of pure iron, which I can machine to size from a casting of melted iron filings.



The slug will be fired through a barrel made from a brass tube. Brass is one of the slipperiest metals and should reduce the friction between the barrel and slug as well as any other material. With the appropriate longitudinal slits cut into it, in order to reduce eddy currents, the brass tube should have only a small effect on the magnetics. When constructing the physical model for FEMM, I will assume that it is a non-magnetic material. The length of the brass tube is not important, but both its inner and outer diameters are. The outside diameter of the tube should be as small as possible – which maximizes the strength of the magnetic field of the coils wound around it – while still leaving a bit of clearance between the slug and the inside wall. An internet search shows that www.OnlineMetals.com has in stock brass tubes with an outside diameter of 0.625 inches, a wall thickness of 0.05 inches and an inside diameter of 0.525 inches. This could be perfect. It leaves clearance of 12.5 thousandths of an inch between the slug and the inside wall. This tube can be bought in various lengths up to eight feet long at a modest cost.

The coil of copper wire will be wound directly onto the brass tube. The experience of other coil-gun builders as reported on the internet is that the length of the coil should be the same as the length of the projectile for good results. For this reason, the coil I will look at in the so-called "base case" will be two inches long, the same as the slug. To make it easier to wind the coil, I will assume that the two ends of the coil are physically set by thin annuli of non-magnetic material glued to the barrel. Unclad phenolic printed circuit board 1/32-inch thick would be ideal. Before winding the coil, then, the configuration is as follows.

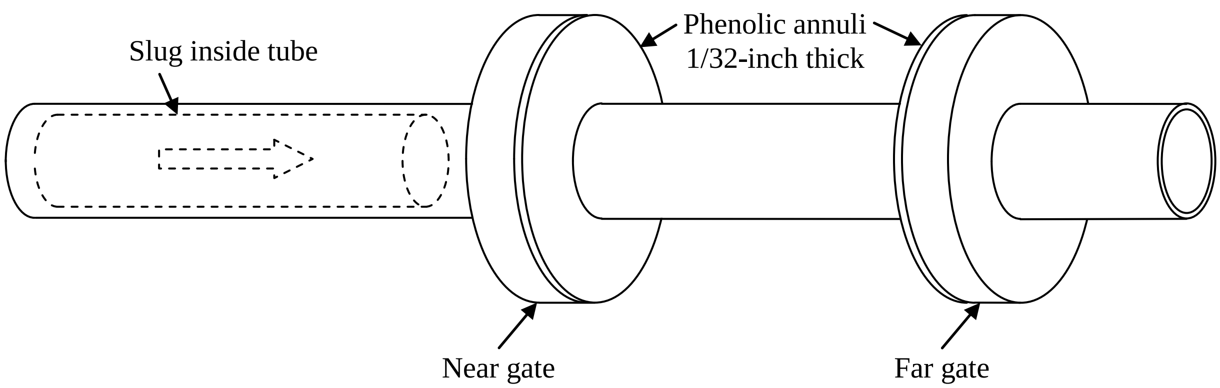


The next step is to wind a coil around the brass tube and between the two phenolic annuli. The size of wire used determines the maximum number of turns which can be put into a single layer. In the base case, I will use AWG#16 wire (American Wire Gauge #16). This wire has an outer diameter of 1.291183 millimeters, or 1.291183 / 25.4 = 0.0508 inches. The two inches between the annuli can hold 2 / 0.0508 = 39.4 turns. I will round down to the lowest integral number of turns, 39. In the base case, I will wind two layers. Two layers of this wire have a thickness of 2 × 0.0508 inches = 0.1016 inches. Therefore, the radius to the outside of the coil will be equal to the outer radius of the brass tube (0.3125 inches) plus the thickness of the coil (0.1016 inches), or 0.4141 inches, a little less than 7/16 inch.

Many experimenters with coil guns have reported success enclosing the coil in a magnetic housing. This confines the lines of magnetic flux, increasing the number which interact with the slug when it is near or inside the coil. The best material to use for such a housing is pure iron. Very mild steel, like 1018, have a high permeability and could also be used. But steel has a disadvantage – it is has some magnetic memory. After the coil gun has been fired a few times, the steel will not return to its original demagnetized state, thus reducing its effectiveness. This is also an issue for the slug, particularly if the same slug is used repeatedly or if there are multiple coils in the gun. Any residual magnetic field in the slug will reduce the magnetic force which can be exerted on it.

To allow for tests of different configurations, I have divided the pure iron housing into three separate parts. The parts are separate only for the purpose of describing the geometry. They can still touch each other. The three parts are: (i) a cylindrical shield around the coil, which I will call the "sheath", (ii) an annular ring at the breach-end of the coil, which I will call the "near gate" and (iii) another annular ring at the barrel-end of the coil, which I will call the "far gate". "Near" and "far" makes sense to the slug as it moves from the breach of the gun out through the barrel.

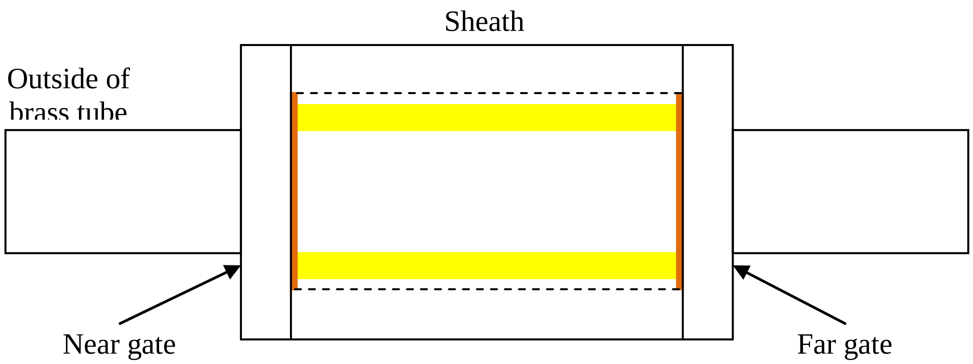
The following figure shows typical near and far gates, but not the sheath, or the actual wire coil either.



I have not said anything yet about the thickness of the gates. In the base case, I have set the thickness to one-quarter of an inch. My simulations show that increasing the thickness of the gates is subject to diminishing returns and thickness greater than 1/4-inch adds very little.

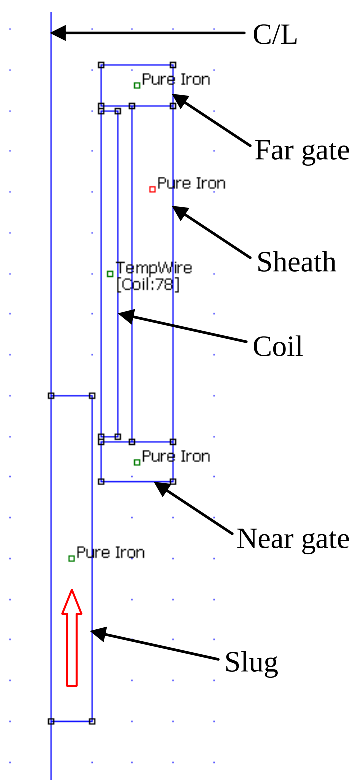
Nor have I said anything about the position of the gates with respect to the phenolic annuli. In the figure, I have shown the gates nestled right up against the annuli. My simulations show that leaving a gap between the gates and the annuli reduces the effectiveness of the coil. Even so, the Lua script in Appendix "A" includes the possibility of such gaps and lets the user set non-zero gaps.

Now, let me deal with the sheath. The inner radius of the sheath cannot be less than the outer radius of the coil, which we calculated above as 0.4141 inches. In the base case, I will give the sheath an inner radius of one-half inch and a thickness of one-quarter inch, but the user can set different parameters in the script. I set the length of the sheath so that it would meet seamlessly with the two gates, as is shown in the following 1:1 scale cross-sectional view of the coil. The copper wire of the coil is rendered in yellow; the two phenolic annuli are rendered in brown. Just to be clear, I have defined variables in the script in such a way that the sheath lies between the two gates. (That makes it easier to use the Lua script to simulate sheaths which are longer than the coil when there are no gates.)



Note that the whole configuration is rotationally symmetric around the long axis of the brass tube. FEMM is ideally suited to such rotational symmetry. The magnetic equations are much simpler when only two variables are needed: (i) one variable (say, "r") to measure the radial distance from the central axis, and (ii) another variable (say "z") to measure distances along the tube from some station we will define. It is not necessary to have a third variable to track an angle around the central axis of the brass tube since the cross-section at every such angle is the same.

The default settings of FEMM's user interface is such that it displays only one-half of the cross-section, and that it displays the "z" axis vertically. The following extract from a screenshot shows the FEMM display of the cross-sections of the slug, the coil and the coil's housing.



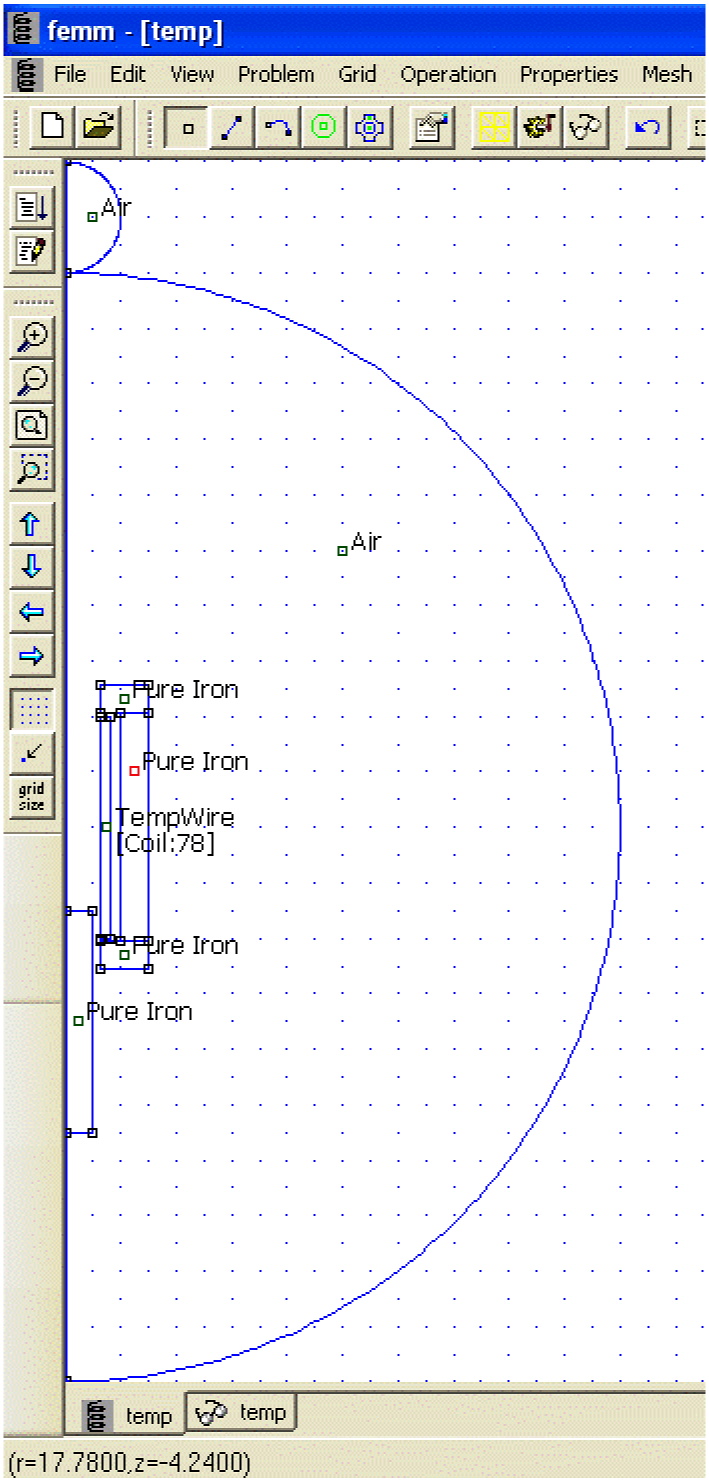
The centerline (C/L) is the central axis of the barrel. The slug is assumed to be fired upwards, in the direction shown by the red arrow. In the base case shown, I have positioned the slug so that its leading edge is 3/4-inch from the center of the coil.

The brass tube is not shown, nor are the phenolic annuli, because they are both assumed to be nonmagnetic and are simply not described in the Lua script.

Each rectangle has somewhere near its middle a label describing the material from which it is made. The coil deserves special mention. It is made of "TempWire". This is a material defined in the Lua script to describe standard AWG gauge 16 wire. I have written the Lua script to allow the user to use wire gauges which are not included in FEMM's default materials library. 16 gauge wire happens to be included in the default library, but others are not. The sublabel for the coil indicates that it contains 78 turns of wire, just as we calculated above.

In the Lua script, each of the five rectangles shown here is defined as a separate "Group". That makes it convenient to refer to the whole of an object without having to specify the individual vertices and lines in it.

The extract shown above is not the whole of the screenshot of FEMM's display. The complete screen appears like this.

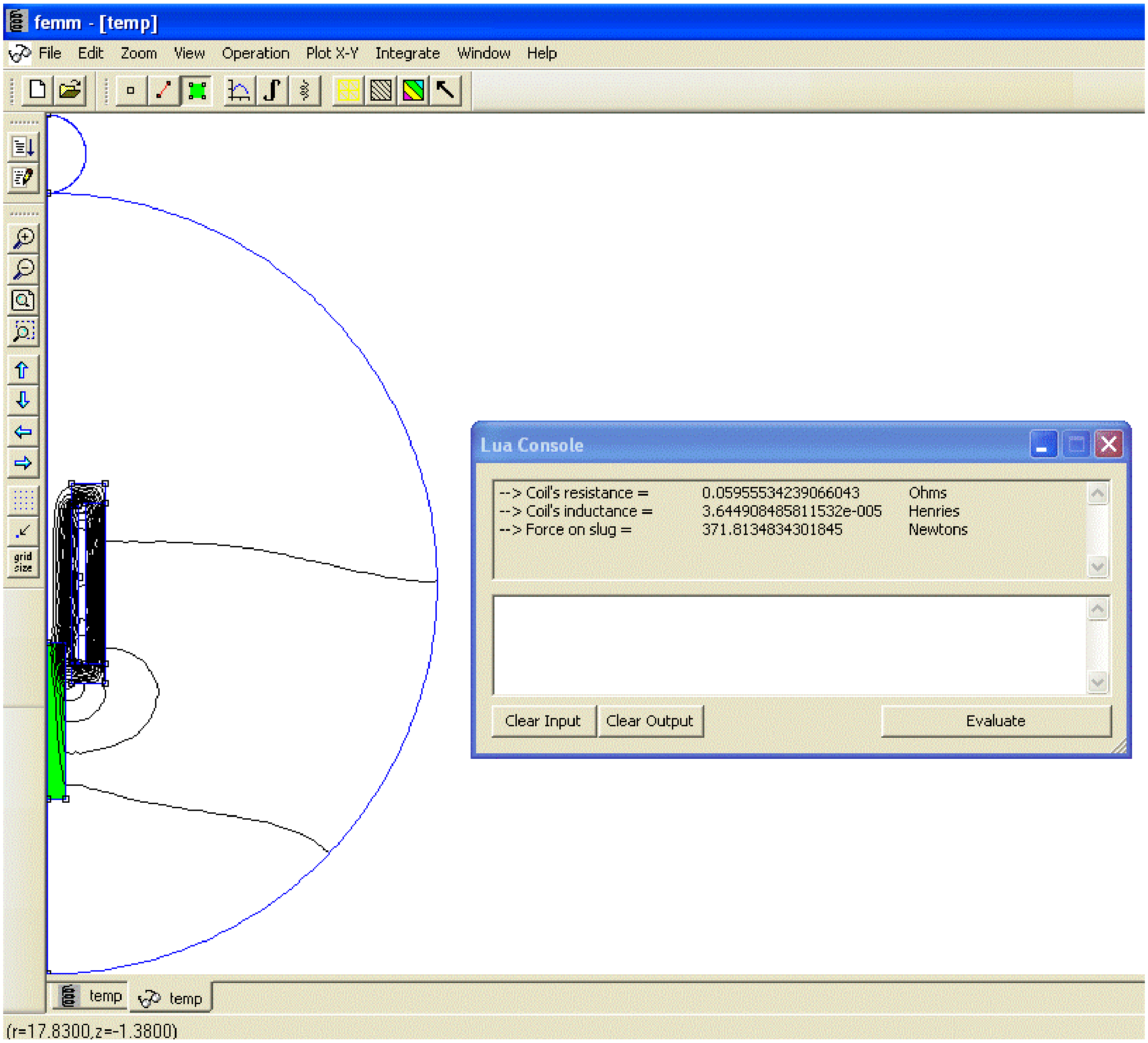


What was not shown in the detail above are two spheres of air, which appear as semi-circles in the half cross-section.

The larger sphere happens to have a radius of five inches. It represents the air immediately surrounding the objects which are of interest to us. I will call this the "local sphere". The smaller sphere, which I will call the "external sphere", has been included to simplify the solution of the magnetic equations. A complete analysis of the magnetic field lines generated by the coil requires that they be tracked everywhere in space, all the way out to an infinite distance away. That, of course, is simply not practical. There are various ways that such an infinite problem can be made finite. The one used here is based on the so-called Kelvin transformation. In that transformation, the whole of the universe outside of the local sphere is mapped into the interior of the external sphere. Fortunately for the user, FEMM takes care of all the details so long as the user properly links the local sphere to its complementary external sphere. That linkage makes sure that the boundary conditions on the surface of the local sphere match the boundary conditions on the external sphere.

Just one side note: I mentioned above that distances in the axial "z" direction are measured from some arbitrary point. The point I selected to use as the origin is the geometric center of the coil.

We need also to specify a current flowing through the coil. In the base case, I used 1000 Amperes. The following screenshot shows the resulting magnetic field lines.



The screenshot includes Lua's output window with the three quantities the script calculates: (i) the Ohmic resistance of the copper wire in the coil (59.56 mΩ), (ii) the inductance of the coil (36.45 μH) and (iii) the net force exerted on the slug (371.8 N, in the positive direction, being upwards).

The Lua script is listed in Appendix "A". The script can be created and edited using any text editor program, such as Notepad or Word. The procedure to execute the script is this. Open a new FEMM window and click on the "File" tab. Click on the item at the very bottom of the drop-down menu, which is "Open Lua Script". Then, using the popup window, browse through your file system to find the file containing the Lua script and highlight it.

Jim Hawley

If you found this description helpful, please let me know. If you spot any errors or omissions, please send an e-mail. Thank you.

Appendix "A"

Listing of the Lua script

```
-- Definition of Groups
-- Group(0) is the air
-- Group(1) is the coil
-- Group(2) is the projectile, or slug
-- Group(3) is the far gate annulus
-- Group(4) is the near gate annulus
-- Group(5) is the surrounding sheath

-- Definition of slug parameters, in inches
SlugLength=2
SlugRadius=0.5/2
SlugMaterial='Pure Iron'
SlugLeadingEdgeZ=-0.75

-- Definition of coil parameters, in inches
CoilLength=2
CoilInnerRadius=0.625/2
WireGauge=16
NumLayers=2

-- Definition of current flowing through the coil
CoilCurrent=1000

-- Thickness of phenolic annuli, in inches
PhenolicThickness=1/32

-- Definition of far gate, in inches
-- (Set IncludeFarGate=0 to eliminate the far gate entirely.)
IncludeFarGate=1
FarGateThickness=1/4
FarGateStandoff=PhenolicThickness
FarGateMaterial='Pure Iron'

-- Definition of near gate, in inches
-- (Set IncludeNearGate=0 to eliminate the far gate entirely.)
IncludeNearGate=1
NearGateThickness=1/4
NearGateStandoff=PhenolicThickness
NearGateMaterial='Pure Iron'

-- Definition of sheath, in inches
-- (Set IncludeSheath=0 to eliminate the sheath entirely.)
IncludeSheath=1
SheathThickness=1/4
SheathInnerRadius=1/2
SheathOuterRadius=SheathInnerRadius+SheathThickness
SheathMaterial='Pure Iron'

-- Extension of sheath beyond the limits of the phenolic annuli, in inches,
-- to be used to determine the length of the sheath if IncludeFarGate=0 and/or
-- IncludeNearGate=0.
SheathFarExtension=0
SheathNearExtension=0

-- Radius of bounding air spheres, in inches
AirInnerRadius=5
```

```
AirExternalDiameter=1

-- Table to look up the wire diameter, in inches, from the specified gauge
if(WireGauge==26) then
    WireDiameter=0.4050189/25.4
end
if(WireGauge==24) then
    WireDiameter=0.5107140/25.4
end
if(WireGauge==22) then
    WireDiameter=0.6439917/25.4
end
if(WireGauge==20) then
    WireDiameter=0.8120499/25.4
end
if (WireGauge==18) then
    WireDiameter=1.023965/25.4
end
if (WireGauge==16) then
    WireDiameter=1.291183/25.4
end
if (WireGauge==14) then
    WireDiameter=1.628134/25.4
end
if (WireGauge==12) then
    WireDiameter=2.053018/25.4
end
if (WireGauge==10) then
    WireDiameter=2.588780/25.4
end
if (WireGauge==8) then
    WireDiameter=3.2639/25.4
end
if (WireGauge==6) then
    WireDiameter=4.1148/25.4
end
if (WireGauge==4) then
    WireDiameter=5.18922/25.4
end
if (WireGauge==2) then
    WireDiameter=6.54304/25.4
end

-- Calculate the other parameters of the coil
NumTurns=NumLayers*floor(CoilLength/WireDiameter)
CoilOuterRadius=CoilInnerRadius+(NumLayers*WireDiameter)

--------------------------------
-- Start laying out the geometry
--------------------------------
newdocument(0)                                 -- Create a new magnetics problem
mi_probdef(0,'inches','axi',1E-8,0,30)         -- The geometry is axisymmetric, in inches
mi_grid_snap('off')                            -- Do not snap points to any grid

-- Define all nodes (i.e., points) of the surrounding air, which comprise Group(0)
mi_addnode(0,AirInnerRadius)                   -- Top of inner sphere
mi_addnode(0,-AirInnerRadius)                  -- Bottom of inner sphere
mi_addnode(0,AirInnerRadius+AirExternalDiameter) -- Top of external sphere
mi_clearselected()
mi_selectnode(0,AirInnerRadius)
mi_selectnode(0,-AirInnerRadius)
```

```
mi_selectnode(0,AirInnerRadius+AirExternalDiameter)
mi_setnodeprop('',0)

-- Define all nodes of the coil, which comprise Group(1)
mi_addnode(CoilInnerRadius,CoilLength/2)         -- Top inside corner of coil
mi_addnode(CoilOuterRadius,CoilLength/2)         -- Top outside corner of coil
mi_addnode(CoilInnerRadius,-CoilLength/2)        -- Bottom inside corner of coil
mi_addnode(CoilOuterRadius,-CoilLength/2)        -- Bottom outside corner of coil
mi_clearselected()
mi_selectnode(CoilInnerRadius,CoilLength/2)
mi_selectnode(CoilOuterRadius,CoilLength/2)
mi_selectnode(CoilInnerRadius,-CoilLength/2)
mi_selectnode(CoilOuterRadius,-CoilLength/2)
mi_setnodeprop('',1)

-- Define all nodes of the slug, which comprise Group(2)
SlugTopZ=SlugLeadingEdgeZ
SlugBottomZ=SlugTopZ-SlugLength
mi_addnode(0,SlugTopZ)                           -- Top center corner of slug
mi_addnode(SlugRadius,SlugTopZ)                  -- Top outside corner of slug
mi_addnode(0,SlugBottomZ)                        -- Bottom center corner of slug
mi_addnode(SlugRadius,SlugBottomZ)               -- Bottom outside corner of slug
mi_clearselected()
mi_selectnode(0,SlugTopZ)
mi_selectnode(SlugRadius,SlugTopZ)
mi_selectnode(0,SlugBottomZ)
mi_selectnode(SlugRadius,SlugBottomZ)
mi_setnodeprop('',2)

-- Define all nodes of the far gate, which comprise Group(3)
if (IncludeFarGate==1) then
    FarGateBottomZ=(CoilLength/2)+FarGateStandoff
    FarGateTopZ=FarGateBottomZ+FarGateThickness
    FarGateInnerR=CoilInnerRadius
    FarGateOuterR=SheathOuterRadius
    mi_addnode(FarGateInnerR,FarGateTopZ)        -- Top inside corner of far gate
    mi_addnode(FarGateOuterR,FarGateTopZ)        -- Top outside corner of far gate
    mi_addnode(FarGateInnerR,FarGateBottomZ)     -- Bottom inside corner of far gate
    mi_addnode(FarGateOuterR,FarGateBottomZ)     -- Bottom outside corner of far gate
    mi_clearselected()
    mi_selectnode(FarGateInnerR,FarGateTopZ)
    mi_selectnode(FarGateOuterR,FarGateTopZ)
    mi_selectnode(FarGateInnerR,FarGateBottomZ)
    mi_selectnode(FarGateOuterR,FarGateBottomZ)
    mi_setnodeprop('',3)
end

-- Define all nodes of the near gate, which comprise Group(4)
if (IncludeNearGate==1) then
    NearGateTopZ=(-CoilLength/2)-NearGateStandoff
    NearGateBottomZ=NearGateTopZ-NearGateThickness
    NearGateInnerR=CoilInnerRadius
    NearGateOuterR=SheathOuterRadius
    mi_addnode(NearGateInnerR,NearGateBottomZ)   -- Bottom inside corner of near gate
    mi_addnode(NearGateOuterR,NearGateBottomZ)   -- Bottom outside corner of near gate
    mi_addnode(NearGateInnerR,NearGateTopZ)      -- Top inside corner of near gate
    mi_addnode(NearGateOuterR,NearGateTopZ)      -- Top outside corner of near gate
    mi_clearselected()
    mi_selectnode(NearGateInnerR,NearGateBottomZ)
    mi_selectnode(NearGateOuterR,NearGateBottomZ)
    mi_selectnode(NearGateInnerR,NearGateTopZ)
```

```
    mi_selectnode(NearGateOuterR,NearGateTopZ)
    mi_setnodeprop('',4)
end

-- Define all nodes of the sheath, which comprise Group(5)
-- Take care to select the nodes by referring to very close neighbouring points.
if (IncludeSheath==1) then
    if (IncludeFarGate==1) then
        SheathTopZ=FarGateBottomZ
    end
    if (IncludeFarGate==0) then
        SheathTopZ=(CoilLength/2)+PhenolicThickness+SheathFarExtension
    end
    mi_addnode(SheathInnerRadius,SheathTopZ)     -- Top inside corner of sheath
    mi_addnode(SheathOuterRadius,SheathTopZ)     -- Top outside corner of sheath
    if (IncludeNearGate==1) then
        SheathBottomZ=NearGateTopZ
    end
    if (IncludeNearGate==0) then
        SheathBottomZ=-((CoilLength/2)+PhenolicThickness+SheathNearExtension)
    end
    mi_addnode(SheathInnerRadius,SheathBottomZ)  -- Bottom inside corner of sheath
    mi_addnode(SheathOuterRadius,SheathBottomZ)  -- Bottom outside corner of sheath
    mi_clearselected()
    mi_selectnode(SheathInnerRadius,SheathTopZ-0.00001)
    mi_selectnode(SheathOuterRadius,SheathTopZ-0.00001)
    mi_selectnode(SheathInnerRadius,SheathBottomZ+0.00001)
    mi_selectnode(SheathOuterRadius,SheathBottomZ+0.00001)
    mi_setnodeprop('',5)
end

-- Define all segments (i.e., lines) of the surrounding air, which must be put into Group(0)
-- There are three line segments:
-- 1. From the bottom of the local sphere to the trailing edge of the slug.
-- 2. From the leading edge of the slug to the top of the local sphere.
-- 3. The diameter line across the external sphere.
mi_addsegment(0,-AirInnerRadius,0,SlugBottomZ)
mi_addsegment(0,SlugTopZ,0,AirInnerRadius)
mi_addsegment(0,AirInnerRadius,0,AirInnerRadius+AirExternalDiameter)
mi_clearselected()
mi_selectsegment(0,-AirInnerRadius+0.00001)
mi_selectsegment(0,AirInnerRadius-0.00001)
mi_selectsegment(0,AirInnerRadius+0.00001)
mi_setsegmentprop('',0,0,0,0)

-- Define a periodic boundary condition
mi_addboundprop('PeriodicBC',0,0,0,0,0,0,0,0,4)

-- Define all arcs of the surrounding air, which must be put into Group(0)
-- There are two arcs:
-- 1. Enclosing the local sphere.
-- 2. Enclosing the external sphere.
mi_addarc(0,-AirInnerRadius,0,AirInnerRadius,180,1)
mi_addarc(0,AirInnerRadius,0,AirInnerRadius+AirExternalDiameter,180,1)
mi_clearselected()
mi_selectarcsegment(0,AirInnerRadius-0.00001)
mi_selectarcsegment(0,AirInnerRadius+0.00001)
mi_setarcsegmentprop(1,'PeriodicBC',0,0)

-- Define all segments of the coil, which are in Group(1)
-- The coil's cross-section is a simple rectangle, with four sides.
```

```
mi_addsegment(CoilInnerRadius,-CoilLength/2,CoilInnerRadius,CoilLength/2)
mi_addsegment(CoilInnerRadius,CoilLength/2,CoilOuterRadius,CoilLength/2)
mi_addsegment(CoilOuterRadius,CoilLength/2,CoilOuterRadius,-CoilLength/2)
mi_addsegment(CoilOuterRadius,-CoilLength/2,CoilInnerRadius,-CoilLength/2)
mi_clearselected()
mi_selectsegment((CoilInnerRadius+CoilOuterRadius)/2,CoilLength/2)
mi_selectsegment((CoilInnerRadius+CoilOuterRadius)/2,-CoilLength/2)
mi_selectsegment(CoilInnerRadius,0)
mi_selectsegment(CoilOuterRadius,0)
mi_setsegmentprop('',0,0,0,1)

-- Define all segments of the slug, which are in Group(2)
-- The slug's cross-section is a simple rectangle, with four sides.
mi_addsegment(0,SlugBottomZ,0,SlugTopZ)
mi_addsegment(0,SlugTopZ,SlugRadius,SlugTopZ)
mi_addsegment(SlugRadius,SlugTopZ,SlugRadius,SlugBottomZ)
mi_addsegment(SlugRadius,SlugBottomZ,0,SlugBottomZ)
mi_clearselected()
mi_selectsegment(SlugRadius/2,SlugTopZ)
mi_selectsegment(SlugRadius/2,SlugBottomZ)
mi_selectsegment(0,(SlugTopZ+SlugBottomZ)/2)
mi_selectsegment(SlugRadius,(SlugTopZ+SlugBottomZ)/2)
mi_setsegmentprop('',0,0,0,2)

-- Define all segments of the far gate, which are in Group(3)
-- The far gate's cross-section is a simple rectangle, with four sides.
if (IncludeFarGate==1) then
    mi_addsegment(FarGateInnerR,FarGateBottomZ,FarGateInnerR,FarGateTopZ)
    mi_addsegment(FarGateInnerR,FarGateTopZ,FarGateOuterR,FarGateTopZ)
    mi_addsegment(FarGateOuterR,FarGateTopZ,FarGateOuterR,FarGateBottomZ)
    mi_addsegment(FarGateOuterR,FarGateBottomZ,FarGateInnerR,FarGateBottomZ)
    mi_clearselected()
    mi_selectsegment(FarGateInnerR+0.00001,FarGateTopZ)
    mi_selectsegment(FarGateInnerR+0.00001,FarGateBottomZ)
    mi_selectsegment(FarGateInnerR,(FarGateTopZ+FarGateBottomZ)/2)
    mi_selectsegment(FarGateOuterR,(FarGateTopZ+FarGateBottomZ)/2)
    mi_setsegmentprop('',0,0,0,3)
end

-- Define all segments of the near gate, which are in Group(4)
-- The near gate's cross-section is a simple rectangle, with four sides.
if (IncludeNearGate==1) then
    mi_addsegment(NearGateInnerR,NearGateBottomZ,NearGateInnerR,NearGateTopZ)
    mi_addsegment(NearGateInnerR,NearGateTopZ,NearGateOuterR,NearGateTopZ)
    mi_addsegment(NearGateOuterR,NearGateTopZ,NearGateOuterR,NearGateBottomZ)
    mi_addsegment(NearGateOuterR,NearGateBottomZ,NearGateInnerR,NearGateBottomZ)
    mi_clearselected()
    mi_selectsegment(NearGateInnerR+0.00001,NearGateTopZ)
    mi_selectsegment(NearGateInnerR+0.00001,NearGateBottomZ)
    mi_selectsegment(NearGateInnerR,(NearGateTopZ+NearGateBottomZ)/2)
    mi_selectsegment(NearGateOuterR,(NearGateTopZ+NearGateBottomZ)/2)
    mi_setsegmentprop('',0,0,0,4)
end

-- Define all segments of the sheath, which must be put into Group(5)
-- The sheath's cross-section is a simple rectangle, with four sides.
if (IncludeSheath==1) then
    mi_addsegment(SheathInnerRadius,SheathBottomZ,SheathInnerRadius,SheathTopZ)
    mi_addsegment(SheathInnerRadius,SheathTopZ,SheathOuterRadius,SheathTopZ)
    mi_addsegment(SheathOuterRadius,SheathTopZ,SheathOuterRadius,SheathBottomZ)
    mi_addsegment(SheathOuterRadius,SheathBottomZ,SheathInnerRadius,SheathBottomZ)
```

```
    mi_clearselected()
    mi_selectsegment((SheathInnerRadius+SheathOuterRadius)/2,SheathTopZ-0.00001)
    mi_selectsegment((SheathInnerRadius+SheathOuterRadius)/2,SheathBottomZ+0.00001)
    mi_selectsegment(SheathInnerRadius,0)
    mi_selectsegment(SheathOuterRadius,0)
    mi_setsegmentprop('',0,0,0,5)
end

-- Define all block labels of the air, which must be put into Group(0)
mi_addblocklabel(AirInnerRadius/2,AirInnerRadius/2)
mi_addblocklabel(AirExternalDiameter/4,AirInnerRadius+(AirExternalDiameter/2))
mi_clearselected()
mi_selectlabel(AirInnerRadius/2,AirInnerRadius/2)
mi_selectlabel(AirExternalDiameter/4,AirInnerRadius+(AirExternalDiameter/2))
mi_getmaterial('Air')
mi_setblockprop('Air',0,0,0,0,0,0)

-- Describe the external region as a Kelvin transformation
mi_defineouterspace(AirInnerRadius+(AirExternalDiameter/2),AirExternalDiameter/2,AirInnerRadius)

-- Temporarily define a new material for the wire being used.  This is done so that the user
-- does not have to manually add a wire to the FEMM's default materials library.
-- Name = TempWire
-- Relative permeability in r-direction = 1
-- Relative permeability in z-direction = 1
-- Permanent magnet coercivity = 0
-- Applied source current density = 0
-- Electrical conductivity = 58 MS/m
-- Lamination thickness = 0
-- Hysteresis lag angle = 0
-- Lamination fill fraction = 1 (Used
-- Lamination type = 3 (This code identifies magnet wire)
-- Hysteresis lag angle in the x-direction
-- Hysteresis lag angle in the y-direction
-- Number of strands in wire = 1
-- Diameter of wire (in millimeters) has been selected by the user above.
mi_addmaterial('TempWire',1,1,0,0,58,0,0,1,3,0,0,1,WireDiameter*25.4)

-- Define all block labels of the coil, which must be put into Group(1)
-- The "1" argument in mi_addcircuitprop puts the coil into series with
-- whatever circuit is being examined.  It's not relevant in this analysis.
mi_addcircprop('Coil', CoilCurrent,1)
mi_addblocklabel((CoilInnerRadius+CoilOuterRadius)/2,0)
mi_clearselected()
mi_selectlabel((CoilInnerRadius+CoilOuterRadius)/2,0)
mi_setblockprop('TempWire',0,0,'Coil',0,1,NumTurns)

-- Define all block labels of the slug, which must be put into Group(2)
mi_addblocklabel(SlugRadius/2,SlugLeadingEdgeZ-(SlugLength/2))
mi_clearselected()
mi_selectlabel(SlugRadius/2,SlugLeadingEdgeZ-(SlugLength/2))
mi_getmaterial('Pure Iron')
mi_setblockprop('Pure Iron',0,0,0,0,2,0)

-- Define all block labels of the far gate, which must be put into Group(3)
if (IncludeFarGate==1) then
    mi_addblocklabel((FarGateInnerR+FarGateOuterR)/2,(FarGateTopZ+FarGateBottomZ)/2)
    mi_clearselected()
    mi_selectlabel((FarGateInnerR+FarGateOuterR)/2,(FarGateTopZ+FarGateBottomZ)/2)
    mi_getmaterial('Pure Iron')
    mi_setblockprop('Pure Iron',0,0,0,0,3,0)
```

```
end

-- Define all block labels of the near gate, which must be put into Group(4)
if (IncludeNearGate==1) then
    mi_addblocklabel((NearGateInnerR+NearGateOuterR)/2,(NearGateTopZ+NearGateBottomZ)/2)
    mi_clearselected()
    mi_selectlabel((NearGateInnerR+NearGateOuterR)/2,(NearGateTopZ+NearGateBottomZ)/2)
    mi_getmaterial('Pure Iron')
    mi_setblockprop('Pure Iron',0,0,0,0,4,0)
end

-- Define all block labels of the sheath, which must be put into Group(5)
if (IncludeSheath==1) then
    mi_addblocklabel((SheathInnerRadius+SheathOuterRadius)/2,(SheathBottomZ+(3*SheathTopZ))/4)
    mi_clearselected()
    mi_selectlabel((SheathInnerRadius+SheathOuterRadius)/2,(SheathBottomZ+(3*SheathTopZ))/4)
    mi_getmaterial('Pure Iron')
    mi_setblockprop('Pure Iron',0,0,0,0,5,0)
end

-- Save the construction in a temporary file which is a sister file to this Lua script.
mi_saveas("./temp.fem")

-------------------
-- Analyze the coil
-------------------
main_maximize()                            -- Maximize the main FEMM window
mi_zoomnatural()                           -- Scale the display to show the complete
configuration
showconsole()                              -- Show the Lua output window
clearconsole()                             -- Clear the Lua output window
mi_analyze()                               -- Execute the FEMM analysis and ...
mi_loadsolution()                          -- ... load the solution
val1,val2,val3=mo_getcircuitproperties('Coil')
CoilResistance=val2/val1                   -- Calculate the ohmic resistance of the coil
CoilInductance=val3/val1                   -- Calculate the inductance of the coil
mo_groupselectblock(2)                     -- Select Group(2), which is the slug
ForceOnSlug=mo_blockintegral(19)           -- Integrate around the surface of the slug

-- Print the results in the Lua output window
print("Coil's resistance = ",CoilResistance," Ohms")
print("Coil's inductance = ",CoilInductance," Henries")
print("Force on slug = ",ForceOnSlug," Newtons")

-- It is not necessary to remove the temporary material from the materials library, but
-- removal can be done by uncommenting the following statement by removing the leading "--".
-- mi_deletematerial('TempWire')

-- Close out the Lua program
-- mo_close()
-- mi_close()
messagebox("All done.")
```