

Automating the FEMM-analysis of a coil

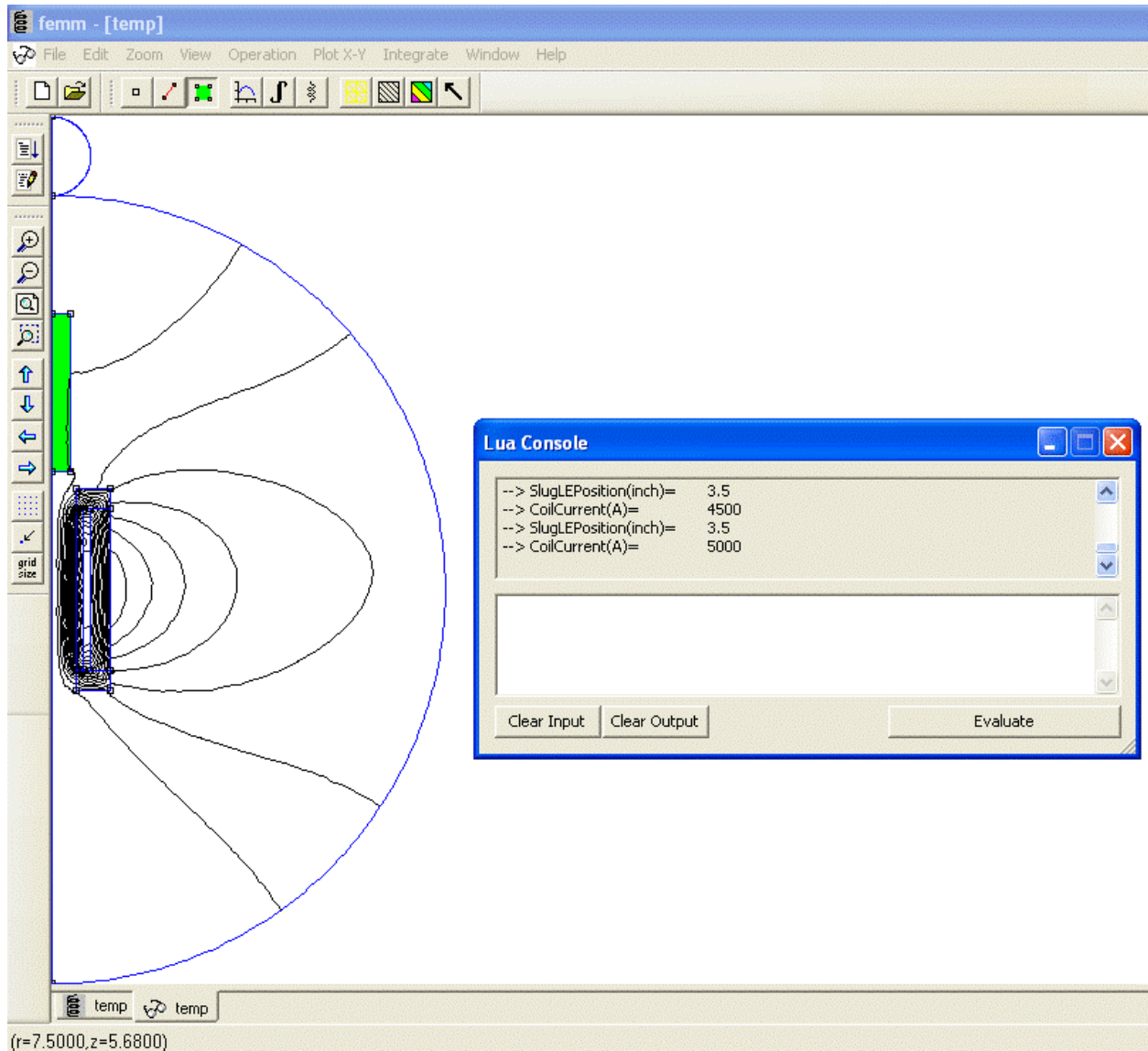
The Lua script described in an earlier paper titled *Using FEMM to analyze a coil from a coil gun* calculates the coil resistance, coil inductance and axial force exerted on a slug. That is a very useful calculation, but the calculation is done only for one particular configuration. The script would be even more useful if it could be enhanced to make the same calculations for a range of one or more variables. The purpose of this paper is to describe the enhanced script which repeats the calculation for a range of slug positions and, simultaneously, a range of coil currents. Once a coil and slug have been specified, these two variables – slug position and coil current – are the most important ones which will vary during a firing even.

The enhanced Lua script is listed in Appendix "A" below. Let me describe briefly the changes from the original script listed in the earlier paper.

1. The script uses two nested Do-loops to step through the variable ranges. The outer loop steps through various slug positions. The inner loop, which is executed for each separate slug position, steps through various coil currents.
2. In the sample script listed below, I have used 51 slug positions. In the first position, the leading edge (nose) of the slug is 1½ inches below the center of the coil. Since the coil has a length of two inches, this means that the slug is one-half inch away from entering the coil. In the last position, the leading edge of the slug is 3½ inches above the center of the coil. Since the slug is two inches long, this means that the slug has passed completely through the coil and a further one-half inch as well. From the first position to the last, the slug has effectively "travelled" by a distance of five inches. Since there are 51 slug positions, they are separated by a distance of $5 / 50 = 0.1$ inch.
3. In the sample script listed below, I have used ten coil currents, ranging from a low of 500 Amperes to a high of 5,000 Amperes, in steps of 500 Amperes.
4. The FEMM analysis is carried out $51 \times 10 = 510$ times, once for each pair of slug position and coil currents. The same three quantities as before – the ohmic resistance, inductance and force – are calculated for each pair. This is a little bit redundant since the ohmic resistance of the coil is always the same, but repeating the calculation makes it a little bit easier to process the results.
5. The results of the calculations are saved in a text file, rather than being written in the Lua display window. The Lua display window is still used, but only to keep the user up-to-date with the slug position and coil current pair which FEMM is currently processing. In the sample script, the output text file is named `FEMM_Analysis_Results.txt` and is created in the same directory as the Lua script. The file can be opened using any text editor, such as Notepad or Word.
6. The results of each calculation are saved on one line of the output text file. When the analysis is done the output text file will therefore contain 510 lines. Each line will consist of five numbers: (i) the position of the slug's leading edge, in inches, (ii) the coil current, in Amperes, (iii) the coil's ohmic resistance, in milli-Ohms, (iv) the coil's inductance, in micro-Henries and (v) the axial force exerted on the slug, in Newtons. Axial forces which are algebraically positive represent forces which accelerate the slug in the positive "z" direction, which is to say, upwards. But, the coil can also decelerate the slug, tending to pull it back into the coil after it has passed through. Such decelerating forces will be algebraically negative.

- An interesting bit of the script is the part which re-positions the slug. The physical configuration is constructed with the leading edge of the slug three-quarters of an inch below the center of the coil. (This is the same position of the slug as in the previous paper.) The Lua script defines everything associated with the slug, being its vertices, lines and label, as objects in Group(2). All of these objects can be moved using Lua's translation command. When it is necessary to re-position the slug, the script translates Group(2) up or down the "z" axis.

Executing the script takes about two hours on my IBM ThinkPad laptop. The following extract from a screenshot shows the FEMM display when the process is complete. The green rectangle is the slug, which has passed upwards and through the coil.



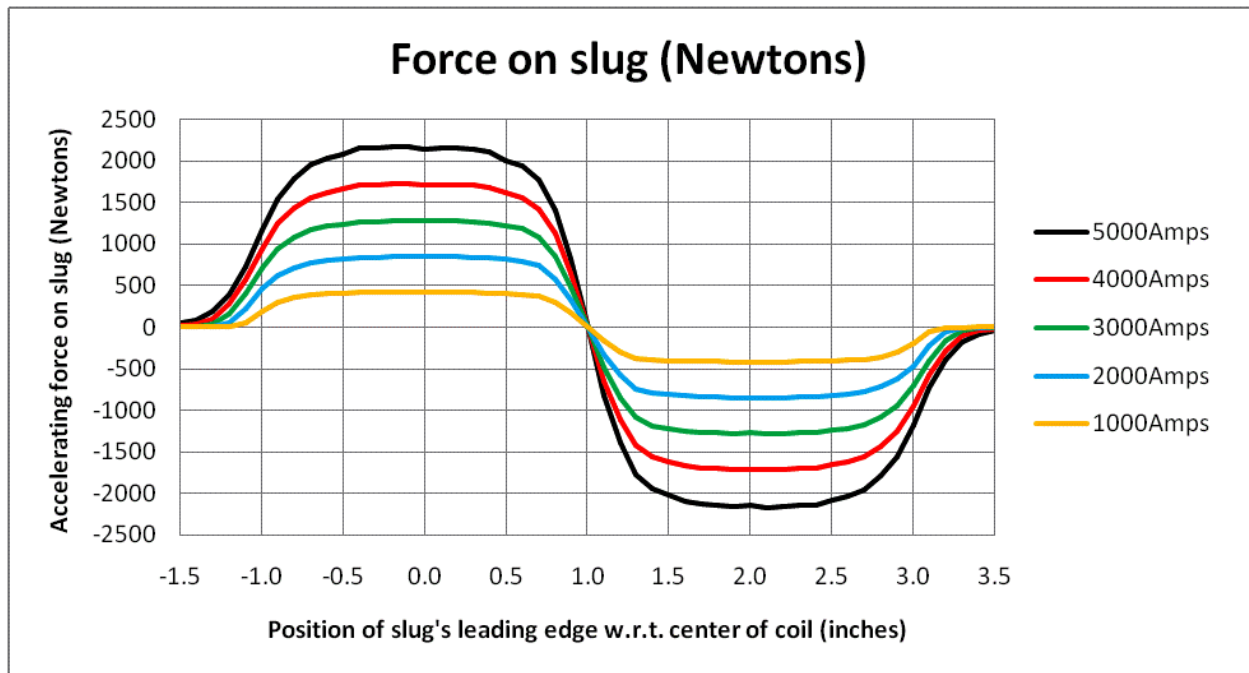
The following text is the first few lines in the output file.

```
Wire gauge = 16
Number of layers = 2
Z (inch) Current (A) Resistance (mO) Inductance (uH) ForceOnSlug (N)
-1.5 500 59.55534 33.54567 0.04245
```

-1.5	1000	59.55534	33.48010	0.17735
-1.5	1500	59.55534	33.36262	0.43546
-1.5	2000	59.55534	33.22720	0.90780
-1.5	2500	59.55534	33.05089	2.02760
-1.5	3000	59.55534	32.84791	4.61317
-1.5	3500	59.55534	32.64714	9.65051
-1.5	4000	59.55534	32.46824	17.91282
-1.5	4500	59.55534	32.31661	29.62454
-1.5	5000	59.55534	32.18879	44.71660
-1.4	500	59.55534	33.54724	0.16707
-1.4	1000	59.55534	33.48218	0.69619
-1.4	1500	59.55534	33.36365	1.70076
-1.4	2000	59.55534	33.22910	3.40734

The Lua script formats the output in a way which makes it suitable for import into an Excel spreadsheet. Open a new Excel spreadsheet. Click on the "Data" tab and then on the "From Text" sub-tab. Browse through your file system to find the FEMM_Analysis_Results.txt file. Highlight that file and click on the "Import" button. A short dialogue will follow in which you can set the division between columns to capture the five columns of numbers in the text file. The data will be imported into the spreadsheet, ready for manipulation or plotting.

I have plotted the force on the slug from just such a spreadsheet.



The horizontal axis is the location of the slug's leading edge from the center of the coil. Locations are measured upwards from the center of the coil. For example, when the location is -1.0 inch, the slug is below the coil and just entering it, with its leading edge right at the downward end of the coil, which is two inches long. As a second example, when the location is 3.0 inches, the slug is just leaving the coil, and its trailing edge is right at the upward end of the coil.

Coil guns would be easy if we could simply select a nice high current, start it up with the slug near the entrance to the coil, and then stop the current as soon as the slug starts to exit from the coil. But that is

not realistic. Inductive circuits are fairly easy to start, but they can be very difficult to stop. The practical difficulties also increase tremendously when the currents are as large as this.

The Lua script which I used to produce this graph is listed in Appendix "A".

Jim Hawley

© December 2014

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',0,0,0,0,0,0,0,4)

-- 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")

_*****
--** All of the code up to this point is exactly the same as it was in the Lua script **
--** listed in Appendix "A" of the earlier paper titled "Using FEMM to analyze a coil **
--** from a coil gun". Absolutely no changes were made to the code above.          **
_*****

-- Define the range of positions for the leading edge of the slug
FirstPosition=-1.5
LastPosition=3.5
NumberOfPositions=51
DeltaPosition=(LastPosition-FirstPosition)/(NumberOfPositions-1)

-- Define the range of coil currents
FirstCoilCurrent=500
LastCoilCurrent=5000
NumberOfCoilCurrents=10
DeltaCurrent=(LastCoilCurrent-FirstCoilCurrent)/(NumberOfCoilCurrents-1)

-- Open the output text file
-- (The user can record in the output file as much as desired about the configuration.)
handle = openfile("./FEMM_Analysis_Results.txt","a")
write(handle,"Wire gauge = ",WireGauge,"\n")
write(handle,"Number of layers = ",NumLayers,"\n")

-- Write headers for the columns of numbers in the output text file
write(handle,"Z(inch) ")
write(handle,"Current(A) ")
write(handle,"Resistance(mO) ")
write(handle,"Inductance(uH) ")
write(handle,"ForceOnSlug(N)\n")

-----
-- Start the analysis
-----
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_seteditmode("group")      -- Make edits such as translation by Group

-- Loop through slug positions
for Iposition=1,NumberOfPositions do

    -- Calculate the desired position of the slug's leading edge (nose)
    DesiredSlugLeadingEdgeZ=FirstPosition+((Iposition-1)*DeltaPosition)
    RequiredSlugTranslation=DesiredSlugLeadingEdgeZ-SlugLeadingEdgeZ
    -- Translate the slug to the desired position
    mi_clearselected()
    mi_selectgroup(2)
    mi_movetranslate(0,RequiredSlugTranslation)
    -- Update the recorded position of the slug
    SlugLeadingEdgeZ=DesiredSlugLeadingEdgeZ

    -- Loop through the range of currents
    for Icurrent=1,NumberOfCoilCurrents do
        Current=FirstCoilCurrent+((Icurrent-1)*DeltaCurrent)
        mi_clearselected()
        mi_selectgroup(1)
        mi_modifycircprop('Coil',1,Current)

        -- Display the current slug position and coil current in the Lua window
        print("SlugLEPosition(inch)=",SlugLeadingEdgeZ)
        print("CoilCurrent(A)=",Current)

        -- FEMM re-calculation for this pair of variables
        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

        -- Write the results to the output text file
        write(handle,format("%.1f",SlugLeadingEdgeZ))
        write(handle,format("    %4d",Current))
        write(handle,format("    %10.5f",CoilResistance*1000))
        write(handle,format("    %10.5f",CoilInductance*1000000))
        write(handle,format("    %10.5f\n",ForceOnSlug))
    end
end

-- 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.")

```