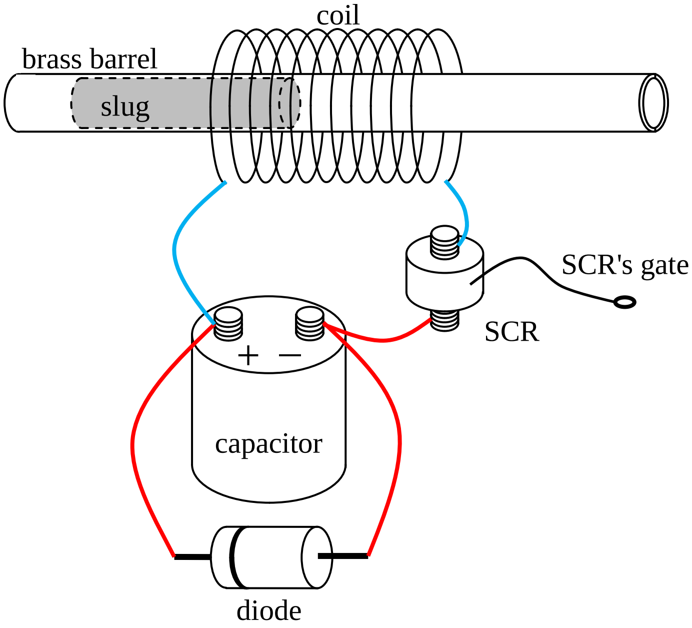
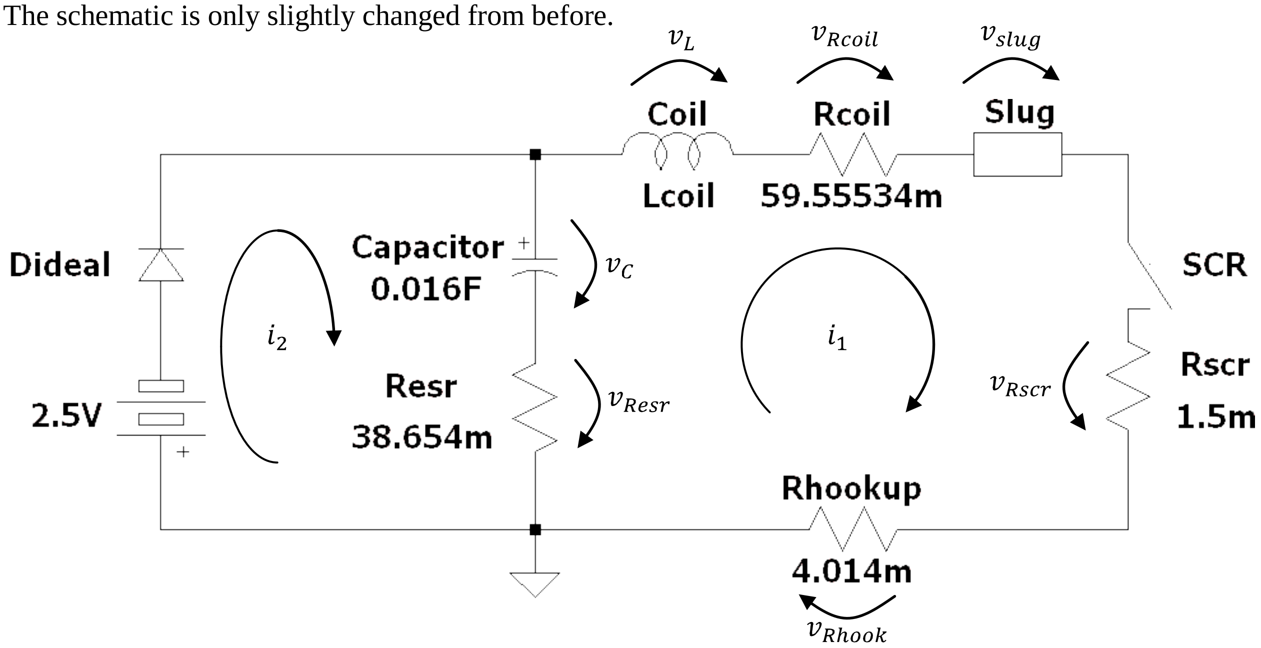
Using FEMM to simulate the firing of a coil gun

I described in an earlier paper, titled *Using FEMM to integrate the circuit equations when the slug is fixed in place*, a Lua script which simulates the circuit equations of a coil from a coil gun with the slug fixed in place. The script uses the FEMM program to calculate inductance. In this paper, I will allow the slug to move.

With one small exception, I will use the circuit from the previous paper. The exception is that I have added a protection diode across the terminals of the capacitor. It will prevent the destruction of the capacitor if current begins to flow the wrong way



The schematic is only slightly changed from before.



The protection diode is modeled as an ideal diode reverse-biased by a constant voltage of 2.5V. If the forward-voltage over the real diode exceeds 2.5V, the ideal diode will begin to conduct.

The circuit equations when the protection diode is not conducting

When the protection diode is not conducting, the circuit reduces to the one in the earlier paper. The Voltage-Current characteristics are very nearly the same. The only change is to the characteristic of the ideal inductance. And, of course, since two currents are identified in the schematic, the current flowing in the main loop is now given by i_1 and not just by i.

$$\left.\begin{array}{ll} \text{Ideal capacitor:} & i_1 = -C\dfrac{dv_C}{dt} \quad \rightarrow \quad v_C = v_{C0} - \dfrac{1}{C}\displaystyle\int_{t=0}^{t} i_1 dt \\ \text{Capacitor's resistance:} & v_{Resr} = -R_{esr} i_1 \\ \text{Ideal inductance:} & v_{coil} + v_{slug-I} + v_{slug-Z} = \dfrac{d(L_{total} i_1)}{dt} \\ \text{Coil's resistance:} & v_{Rcoil} = R_{coil} i_1 \\ \text{SCR's resistance:} & v_{Rscr} = R_{scr} i_1 \\ \text{Hookup wire:} & v_{Rhook} = R_{hook} i_1 \end{array}\right\} \quad (1)$$

As before, I have used the symbol L_{total} for the total inductance of the combination of the ideal coil and the slug. The product of this total inductance and the current i_1 is the flux which connects the individual turns in the coil with each other and with the slug. I am going to break down the voltage drop over this combination into three parts and its total inductance into two parts, as follows:

$$\begin{array}{ll} v_{total} = v_{coil} + v_{slug-I} + v_{slug-Z} & (2A) \\ L_{total} = L_{coil} + L_{slug} & (2B) \end{array}$$

My intention is this. L_{coil} will represent the inductance of the turns of copper wire when the slug is not within the sphere of influence of the coil. In other words, L_{coil} is the (constant) inductance of the coil when it has an air core. L_{slug} will represent the changes to the total inductance which are caused by changes in the current and in the slug's location. The separate effects of the changes in current and location are allocated to separated voltage drops. The "I" and "Z" subscripts denote the dependence on current and location, respectively.

As before, one can expand the rate-of-change of the inductance-times-current product using the product rule for derivatives. We get:

$$\begin{aligned} v_{coil} + v_{slug-I} + v_{slug-Z} &= \frac{d\left(L_{coil} i_1 + L_{slug} i_1\right)}{dt} \\ &= L_{coil}\frac{di_1}{dt} + L_{slug}\frac{di_1}{dt} + i_1\frac{dL_{slug}}{dt} \quad (3) \end{aligned}$$

It is true that L_{slug} will change with time. But, it is not the passage of time *per se* that causes the change in this inductance, it is the change in the current and the change in the location. We can express the rate-of-change of L_{slug} with respect to time in terms of its rates-of-change with respect to current and location. This is done using partial derivatives, as follows.

$$\frac{dL_{slug}}{dt} = \frac{\partial L_{slug}}{\partial i_1}\frac{di_1}{dt} + \frac{\partial L_{slug}}{\partial z}\frac{dz}{dt} \quad (4)$$

We can substitute Equation (4) into Equation (3) to get:

$$v_{coil} + v_{slug-I} + v_{slug-Z} = L_{coil}\frac{di_1}{dt} + L_{slug}\frac{di_1}{dt} + i_1\frac{\partial L_{slug}}{\partial i_1}\frac{di_1}{dt} + i_1\frac{\partial L_{slug}}{\partial z}\frac{dz}{dt} \quad (6)$$

It makes sense to assign the terms to the voltages in the following way.

$$\left.\begin{aligned} v_{coil} &= L_{coil}\frac{di_1}{dt} \\ v_{slug-I} &= \left(L_{slug} + i_1\frac{\partial L_{slug}}{\partial i}\right)\frac{di_1}{dt} \\ v_{slug-Z} &= i_1\frac{\partial L_{slug}}{\partial z}\frac{dz}{dt} \end{aligned}\right\} \quad (7)$$

Now that we have sorted out the voltage-current characteristics, we can move on to the equation that governs the operation of the circuit. The circuit equation is based on the observation that the sum of the voltage drops all the way around the loop must be zero. Proceeding clockwise from the upper left, we add up the voltage drops as follows:

$$v_{coil} + v_{Rcoil} + v_{slug-I} + v_{slug-Z} + v_{Rscr} + v_{Rhook} - v_{Resr} - v_C = 0 \quad (8)$$

Note that v_{Resr} and v_C are "added" with minus signs because I happened to draw the little voltage drop arrows for them in the reverse direction from the others. Substituting the voltage-current relationships for the components, and then collecting terms, gives:

$$\left(L_{coil} + L_{slug} + i_1\frac{\partial L_{slug}}{\partial i}\right)\frac{di_1}{dt} + \left(R_{fixed} + \frac{\partial L_{slug}}{\partial z}\frac{dz}{dt}\right)i_1 - v_{C0} + \frac{1}{C}\int_{t=0}^{t} i_1 dt = 0 \quad (9)$$

As before, I have grouped the four fixed resistances into one, as follows:

$$R_{fixed} = R_{coil} + R_{scr} + R_{hook} + R_{esr} \quad (10)$$

I am going to introduce a new variable q, which is the electrical charge stored in the capacitor. By definition, the current i_1 is the rate at which electrical charge flows out of the capacitor. Therefore:

$$i_1 = -\frac{dq}{dt} \quad (11)$$

The minus sign is present because the stored charge is reduced, and dq/dt is algebraically negative, when "positive" current i_1 is flowing. We can use this definition of current to simplify the integral in Equation (9).

$$\int_{t=0}^{t} i_1 dt = -\int_{t=0}^{t}\frac{dq}{dt}dt = -\int_{t=0}^{t} dq = q(0) - q(t) \quad (12)$$

At any instant in time, the charge stored in the capacitor is equal to the product of the capacitance and the voltage. Using this relationship, we can write the initial charge stored in the capacitor as:

$$q(0) = Cv_{C0} \quad (13)$$

Substituting Equations (12) and (13) into the circuit equation (9) gives:

$$\left(L_{coil} + L_{slug} + i_1\frac{\partial L_{slug}}{\partial i}\right)\frac{di_1}{dt} + \left(R_{fixed} + \frac{\partial L_{slug}}{\partial z}\frac{dz}{dt}\right)i_1 - \frac{q}{C} = 0$$

$$\rightarrow \qquad \Big(L_{total} + \boxed{i_1\frac{\partial L_{total}}{\partial i}\Big)\frac{di_1}{dt}} + \Big(R_{fixed}\boxed{+\frac{\partial L_{total}}{\partial z}\frac{dz}{dt}\Big)i_1} - \frac{q}{C} = 0 \qquad (14)$$

With the exception of one new term, this is the same differential equation we used in the earlier paper. The ellipse identifies the new term. It depends on the slug's speed dz/dt and on the rate-of-change of the total inductance with respect to the slug's position. There is something else to see as well.

The term which represents the effect of the slug due to changes in current (which is identified by the rectangle) enters into Equation (14) in exactly the same way as a typical inductance, like L_{total}. The term which represents the effect of the slug due to changes in location (which is identified by the ellipse) is quite different. It enters into the equation like the standard resistance R_{fixed}. Inductances remove energy from the circuit, store it but eventually return it to the circuit. Resistances remove energy from the circuit, convert it into another form and do not return it to the circuit. Resistors convert the energy into heat; the slug will convert it into kinetic energy.

Relating the mechanical power added to the slug to the electrical power dissipated

In the schematic diagram above, the slug is depicted as a "black box". Its instantaneous voltage drop is v_{slug} when the instantaneous current flowing through it is i_1. I then divided the voltage drop into two components. I called the one related to the slug's location v_{slug-Z}.

The electrical power p_e absorbed by this hypothetical component is, as always, the product of the instantaneous voltage and the instananeous current, thus:

$$p_e = v_{slug-Z}i_1 \quad (15)$$

There is another way to look at this power. One can look at the mechanical work done on the slug. If the slug is subjected to an instantaneous force F which tends to accelerate it axially down the barrel, then the mechanical work done when pushing the slug through a small distance dz is equal to the force-multiplied-by-distance product Fdz. The rate at which this mechanical work is being done is the amount of work divided by the time dt during which it is done. This quotient is the mechanical power (p_m) being delivered to the slug.

$$p_m = \frac{Fdz}{dt} \quad (16)$$

Assuming that there is no loss of energy during the conversion of electrical energy into mechanical energy, the two statements of power must be the same. Setting $p_e = p_m$, we get:

$$v_{slug-Z} = \frac{F}{i_1}\frac{dz}{dt} \quad (17)$$

Compare Equation (17) with the third of Equations (7). They are both expressions for the instantaneous voltage drop over the slug which arise from changes in the slug's location. We can set them equal, to get:

$$i_1 \frac{\partial L_{total}}{\partial z}\frac{dz}{dt} = \frac{F}{i_1}\frac{dz}{dt}$$

$$\rightarrow \qquad \frac{\partial L_{total}}{\partial z} = \frac{F}{i_1^2} \qquad (18)$$

Making the substitution allows us to re-write Equation (14) as follows:

$$\left(L_{total} + i_1 \frac{\partial L_{total}}{\partial i}\right)\frac{di_1}{dt} + \left(R_{fixed} + \frac{F}{i_1^2}\frac{dz}{dt}\right) i_1 - \frac{q}{C} = 0 \quad (19)$$

This gives us two different ways to handle the numerical integration, using either Equation (14) or Equation (19). We are going to integrate it numerically, in almost exactly the same way we did in the earlier paper. We will first re-arrange the equation so that the key dependent variable – the first derivative of the current i_1 – is isolated on the left-hand side, as follows:

$$Using\ (14){:}\quad \frac{di_1}{dt} = \frac{-\left(R_{fixed} + \frac{\partial L_{total}}{\partial z}\frac{dz}{dt}\right) i_1 + \frac{q}{C}}{L_{total} + i_1 \frac{\partial L_{total}}{\partial i}} \quad (20A)$$

$$Using\ (19){:}\quad \frac{di_1}{dt} = \frac{-\left(R_{fixed} + \frac{F}{i_1^2}\frac{dz}{dt}\right) i_1 + \frac{q}{C}}{L_{total} + i_1 \frac{\partial L_{total}}{\partial i}} \quad (20B)$$

With the exception of the new term in the numerator, this is the same as Equation (12) in the earlier paper, whose evaluation is the most important step in the integration process.

Since we will be using FEMM to calculate the inductance, we can use either version of Equation (20).

Using Equation $(20A)$

When with dealt with this equation in the earlier paper, we included a little side-calculation to estimate the value of the partial derivative $\partial L_{total}/\partial i$. At the start of every time step, we called the FEMM program to calculate the inductance at the then-current. Then, we called the FEMM program a second time, to calculate the inductance at a slightly different current. Dividing the change in inductance by the change in current gives a direct approximation of the partial derivative. I am going to continue to use that side-calculation to estimate $\partial L_{total}/\partial i$. We can use the same approach to estimate the partial derivative $\partial L_{total}/\partial z$ in the new term. We would call the FEMM program a third time at the start of every time step. This time, we would joggle the slug to a slightly different location (but with the same starting current), and calculate another inductance. Dividing the change in inductance from the starting value by the joggle distance would give a direct approximation of the partial derivative $\partial L_{total}/\partial z$.

Using Equation $(20B)$

There is a second way. When FEMM calculates the inductance at the start of every time step, it can calculate the force exerted on the slug at the same time. We can substitute that force in Equation $(20B)$.

In the analysis below, I am going to use the second approach, based on Equation $(20B)$. Equation $(20A)$ has the disadvantage of requiring three FEMM calculations per time step, while Equation $(20B)$ requires only two. For comparison purposes, however, I am going to calculate the partial derivative $\partial L_{total}/\partial z$ and multiply it by i_1^2. The resulting product is a force. If everything was ideal, the value would be equal to the force calculated directly by FEMM.

Integration of the circuit equations if the diode starts conducting

The procedure that carries out the numerical integration of Equation $(20B)$ is so similar to the one descibed in the earlier paper that I will not go to the trouble of describing the steps. What is different from the earlier paper is that the slug moves, so the equations of the slug's dynamics need to be integrated as well. I am going to kill two birds with one stone.

The addition of the protection diode to the circuit causes there to be two current loops. The circuit analysis described above only applies when the diode is not conducting. If the diode starts to conduct, things are different. We need to make a second analysis of the circuit to deal with those periods of time when the protection diode conducts and bypasses current around the capacitor. This section works through that second analysis. It also describes the procedures needed to accelerate the slug (see Step #8). The slug-dynamics procedures are the same whether the diode is conducting or not. They depend only on the force the coil exerts on the slug.

The place to begin this analysis is to be precise about when the protection diode will begin to conduct. Note that the diode is wired in parallel with the real capacitor, not the ideal capacitance C. It will conduct whenever:

$$v_C + v_{Resr} \leq -2.5V \quad (21)$$

The sum on the left-hand side $(v_C + v_{ESR})$ is the voltage drop over the protected branch of the circuit, measured from "top" to "bottom" in the schematic. This is the voltage drop over the real capacitor, not just the ideal capacitance C. If this voltage drop is such that it exceeds the threshold voltage for diode conduction, which is $-2.5V$ from "top" to "bottom", the diode will conduct. The second current I labelled in the schematic, i_2, will be non-zero when the diode conducts. So long as the diode remains forward-biased, the voltage drop over the real capacitor will remain <u>at</u> $-2.5V$ (neither higher nor lower). The net current flowing "upwards" through the real capacitor will be the difference $i_1 - i_2$. In this configuration, the Voltage-Current characteristics for all of the components in the circuit can be written as:

$$\left.\begin{array}{rl}
\text{Ideal capacitor:} & i_1 - i_2 = -C\dfrac{dv_C}{dt} \quad \rightarrow \quad v_C = v_{C*} - \dfrac{1}{C}\displaystyle\int_{t=t_*}^{t}(i_1 - i_2)dt \\
\text{ESR:} & v_{Resr} = -R_{esr}(i_1 - i_2) \\
\text{Ideal inductance:} & v_{coil} + v_{slug-I} + v_{slug-Z} = \dfrac{d(L_{total}i_1)}{dt} \\
\text{Coil's resistance:} & v_{Rcoil} = R_{coil}i_1 \\
\text{SCR's resistance:} & v_{Rscr} = R_{scr}i_1 \\
\text{Hookup wire:} & v_{Rhook} = R_{hook}i_1
\end{array}\right\} \quad (22)$$

The symbol v_{C*} in this set of circuit equations is the voltage drop over the ideal capacitor at the moment the diode begins to conduct. Let's say, arbitrarily, that the diode start to conduct at some time $t = t_*$. Since we are dealing with two currents, we expect there to be two circuit equations. The first one is the same as before, namely, that the sum of the voltage drops all around the right-hand loop must be zero.

The second expresses the fact that the voltage drop over the real capacitor is known and constant. The two circuit equations can be written as:

$$\left.\begin{array}{c} v_{coil} + v_{Rcoil} + v_{slug-I} + v_{slug-Z} + v_{Rscr} + v_{Rhook} - v_{Resr} - v_C = 0 \\ -v_{Resr} - v_C = 2.5V \end{array}\right\} \quad (23)$$

We can combine these two equations, and substitute the V-I characteristics, to obtain:

$$\left.\begin{array}{c} \left(L_{total} + i_1 \frac{\partial L_{total}}{\partial i_1}\right)\frac{di_1}{dt} + \left(R_{coil} + R_{hook} + R_{scr} + \frac{\partial L_{total}}{\partial z}\frac{dz}{dt}\right) i_1 + 2.5V = 0 \\ R_{esr}(i_1 - i_2) - v_{C*} + \frac{1}{C}\int_{t=t_*}^{t}(i_1 - i_2)dt = 2.5V \end{array}\right\} \quad (24)$$

The first circuit equation is a first-order differential equation in the independent variable i_1. It includes the same spatial partial derivative $\partial L_{total}/\partial z$ and, as above, I will substitute the mechanical quantity F/i_1^2 for the partial derivative. We are going to integrate it numerically in the following form:

$$\frac{di_1}{dt} = -\frac{\left(R_{coil} + R_{hook} + R_{scr} + \frac{F}{i_1^2}\frac{dz}{dt}\right) i_1 + 2.5V}{L_{total} + i_1 \frac{\partial L_{total}}{\partial i_1}} \quad (25)$$

Now consider the second circuit equation. The net rate at which charge leaves the capacitor is now the difference between currents $i_1 - i_2$, and not just the current i_1, as before. That is:

$$i_1 - i_2 = -\frac{dq}{dt} \quad (26)$$

Substituting this into the second circuit equation makes clear that it, too, is a first-order differential equation, in the independent variable q:

$$-R_{esr}\frac{dq}{dt} - \frac{q}{C} - v_{C*} = 2.5V \quad (27)$$

which we will integrate numerically in the following form:

$$\frac{dq}{dt} = -\frac{\frac{q}{C} + v_{C*} + 2.5}{R_{esr}} \quad (28)$$

Let's assume that we have just completed the n^{th} time step. We will have updated all of the circuit variables (and the location of the slug, too). I will use the same notation as before for quantities which we know at the end of the n^{th} time step, which is coincident with the start of the $n + 1^{\text{st}}$ time step. Here is how we will proceed.

Step #1 - FEMM analysis
As before, we will run a FEMM analysis. The FEMM analysis will give us the coil's inductance $L_{total}|_{start}$ and the force $F|_{start}$ exerted on the slug. We can calculate the part of this total inductance

which is attributable to the slug by subtracting the inductance of the ideal coil alone, which we will have calculated in a separate FEMM analysis done before starting the integration.

$$L_{slug}\Big|_{start} = L_{total}|_{start} - L_{coil} \quad (29)$$

Step #2 - Second FEMM analysis

As before, we will run a second FEMM analysis. I am going to re-run the FEMM analysis with a current which is 10 Amperes larger than $i|_{start}$. This will give a revised total inductance, for which I will use the symbol $L_{total}|_{start+10A}$. The slug's inductance which corresponds to this revised total inductance is:

$$L_{slug}\Big|_{start+10A} = L_{total}|_{start+10A} - L_{coil} \quad (30)$$

An approximation for the partial derivative $\partial L_{slug}/\partial i$ is:

$$\left.\frac{\partial L_{slug}}{\partial i_1}\right|_{start} \cong \frac{L_{slug}\big|_{start+10A} - L_{slug}\big|_{start}}{10\ \text{Amperes}} \quad (31)$$

Step #3 - Calculate the time-rate-of-change of the current i_1 using Equation (25)

We will evaluate the right-hand side of Equation (25) using the values of all the variables at the start of this n^{th} time step. That set of variables now includes the speed of the slug dz/dt at the start of the time step. The differential equation must apply at every instant in time, including this particular instant. Therefore:

$$\left.\frac{di_1}{dt}\right|_{start} = -\frac{\left(R_{coil} + R_{hook} + R_{scr} + \frac{F|_{start}}{i_1^2|_{start}}\left.\frac{dz}{dt}\right|_{start}\right) i_1|_{start} + 2.5V}{L_{slug}\big|_{start} + i_1|_{start}\left.\frac{\partial L_{slug}}{\partial i_1}\right|_{start}} \quad (32)$$

Step #4 - Integrate to find the current i_1 at any time

As before, we will assume that the time step is so short that this first derivative is constant during the whole of the time step. It is a simple matter to integrate as we did before, to calculate the current i_1 at any time t during this time step.

$$i_1|_{\substack{any\ time\ t \\ in\ time\ step}} = i_1|_{start} + \left[(t - t_{start})\left.\frac{di_1}{dt}\right|_{constant}\right] \quad (33)$$

Of particular interest, of course, is the value at the end of this time step. Since the time step is ΔT long, the value at the end of the time step will be:

$$i_1|_{end} = i_1|_{start} + \Delta T\left.\frac{di_1}{dt}\right|_{constant} \quad (34)$$

Step #5 - Calculate the time-rate-of-change of the charge q using Equation (28)

We will now calculate the first time-derivative of the stored charge q at the start of this time step. We will use the following restated form of Equation (28) for this purpose:

$$\left.\frac{dq}{dt}\right|_{constant} = -\frac{\frac{q|_{start}}{C} + v_{C*} + 2.5}{R_{esr}} \quad (35)$$

All of the quantities on the right-hand side of this Equation (35) are known, so the first derivative on the left-hand side can be computed.

Step #6 - Integrate to find the charge q at any time

We will assume that the time step is so short that this derivative, too, remains constant during the whole of the time step. It is a simple matter to integrate as we have done before, to calculate the charge q at any time t during this time step.

$$q|_{\substack{any\ time\ t \\ in\ time\ step}} = q|_{start} + \left[(t - t_{start})\left.\frac{dq}{dt}\right|_{constant}\right] \quad (36)$$

Of particular interest, of course, is the value at the end of this time step. Since the time step is ΔT long, the value at the end of the time step will be:

$$q|_{end} = q|_{start} + \Delta T\left.\frac{dq}{dt}\right|_{constant} \quad (37)$$

Step #7 - The current i_2 in the protection circuit

As a loose end, we should calculate the current i_2 flowing in the protection circuit. This is a matter of subtraction. At any time t during the time step, we have the following function of time:

$$\begin{aligned} & i_2 = i_1 + \frac{dq}{dt} \\ \rightarrow \quad & i_2|_{\substack{any\ time\ t \\ in\ time\ step}} = \left(i_1|_{start} + \left.\frac{dq}{dt}\right|_{constant}\right) + \left[(t - t_{start})\left.\frac{di_1}{dt}\right|_{constant}\right] \quad (38) \end{aligned}$$

and, at the end of the time step:

$$i_2|_{end} = \left(i_1|_{start} + \left.\frac{dq}{dt}\right|_{constant}\right) + \Delta T\left.\frac{di_1}{dt}\right|_{constant} \quad (39)$$

Step #8 - Slug dynamics

Newton's Third Law is the basis for the slug's dynamics. The slug exerts a force F on the slug, which is directed axially through the coil. If the slug has mass M, then its acceleration (the second time-derivative of its location z) is given by:

$$\frac{d^2z}{dt^2} = \frac{F}{M} \quad (40)$$

At the start of any time step, the force $F|_{start}$ will have been calculated in Step #1 above. The slug's acceleration at the start of the time step will be:

$$\left.\frac{d^2z}{dt^2}\right|_{start} = \frac{F|_{start}}{M} \quad (41)$$

We will assume that the duration of the time steps are short enough that one can assume that this acceleration remains constant throughout the whole time step.

By definition, the slug's acceleration is the time-derivative of its speed, as in:

$$\frac{d^2z}{dt^2} = \frac{d}{dt}\left(\frac{dz}{dt}\right) \qquad (42)$$

so the speed is the integral of the acceleration:

$$\frac{dz}{dt} = \int \frac{d^2z}{dt^2} dt + Constant \qquad (43)$$

If we start the integration at the start of the time step $t|_{start}$, at which time the slug's speed is $dz/dt|_{start}$, and if we recognize that the slug's acceleration has been assumed to be constant during the time step, then the speed of the slug at any time t during the time step is given by:

$$\begin{aligned}
\left.\frac{dz}{dt}\right|_{\substack{any\ time\ t \\ in\ time\ step}} &= \left.\frac{dz}{dt}\right|_{start} + \int_{t|_{start}}^{time\ t} \left.\frac{d^2z}{dt^2}\right|_{constant} dt \\
&= \left.\frac{dz}{dt}\right|_{start} + \frac{F|_{start}}{M} \int_{t|_{start}}^{time\ t} dt \\
&= \left.\frac{dz}{dt}\right|_{start} + \left[(t - t_{start})\frac{F|_{start}}{M}\right] \qquad (44)
\end{aligned}$$

At the end of the time step, this evaluates to:

$$\left.\frac{dz}{dt}\right|_{end} = \left.\frac{dz}{dt}\right|_{start} + \Delta T \frac{F|_{start}}{M} \qquad (45)$$

This is an important speed. It is the speed at the end of this time step, so it is also the speed at the start of the next time step. It is this speed which will be substituted into Equation (32) when we solve the circuit equations during the next time step.

Now, by definition, the slug's speed is the time-derivative of its location z. Inversely, the location is the integral of the speed.

$$\frac{dz}{dt} = \frac{d}{dt}(z) \quad \rightarrow \quad z = \int \frac{dz}{dt} dt + Constant \qquad (46)$$

If we start the integration at the start of the time step $t|_{start}$, at which time the slug's location is $z|_{start}$, and if we substitute the slug's speed from Equation (44), then the location of the slug at any time t during the time step is given by:

$$
\begin{aligned}
z|_{\substack{any\ time\ t\\ in\ time\ step}} &= z|_{start} + \int_{t|_{start}}^{time\ t} \frac{dz}{dt}dt \\
&= z|_{start} + \int_{t|_{start}}^{time\ t} \left\{ \frac{dz}{dt}\bigg|_{start} + \left[(t - t_{start}) \frac{F|_{start}}{M} \right] \right\} dt \\
&= z|_{start} + \left[(t - t_{start}) \frac{dz}{dt}\bigg|_{start} \right] + \left[\frac{1}{2} (t - t_{start})^2 \frac{F|_{start}}{M} \right] \quad (47)
\end{aligned}
$$

At the end of the time step, this evaluates to:

$$z|_{end} = z|_{start} + \Delta T \frac{dz}{dt}\bigg|_{start} + \frac{1}{2} \Delta T^2 \frac{F|_{start}}{M} \quad (48)$$

This is the location of the slug at the end of the time step. Once this location is known, FEMM can be instructed to move the slug to its new location in the physical model. The only pitfall to watch out for when doing this is to ensure that the measurement units are consistent. In the code listed below, the slug's dynamics are stated in S.I. units (Newtons, meters and seconds), but the FEMM physical model was constructed in inches.

Step #9 - Calculate the voltage drops over the resistances

$$
\begin{aligned}
\text{ESR:} \quad & v_{Resr}|_{end} = -R_{esr}(i_1|_{end} - i_2|_{end}) \\
\text{Coil's resistance:} \quad & v_{Rcoil}|_{end} = R_{coil} i_1|_{end} \\
\text{Hookup wire:} \quad & v_{Rhook}|_{end} = R_{hook} i_1|_{end} \\
\text{SCR's resistance:} \quad & v_{Rscr}|_{end} = R_{scr} i_1|_{end}
\end{aligned}
$$

We have to keep our eye on one thing – the circuit operates in the way described only when the diode is forward-biased. Mathematically, if the equations in this configuration result in a negative protection current ($i_2 < 0$), then the diode has stopped conducting and the equations in the other configuration must be used in the following time steps.

When the capacitor and inductance are reasonably well-balanced for coil gun purposes, the diode will remain forward-biased once it begins to conduct. But, if the capacitance and inductance are too incompatible, the circuit may oscillate several times. The Lua script uses a Boolean-type variable `IsDiodeConducting` to keep track of which set of circuit equations it should use during the upcoming time step.

Other features of the Lua script

Since the slug is allowed to move, the user needs to set three parameters.

- `LELocation_start` establishes the starting location of the leading edge of the slug with respect to the center of the coil, with the positive direction being the direction of travel. In all of the runs described below, the leading edge was set to -1 inch (negative one inch), so the leading edge of the slug is placed at the face of the coil.
- `Speed_0` is the initial speed of the slug, with the positive direction being the direction of travel. In all of the runs described below, the slug starts out from rest, with zero speed.

- LELocation_Stop is the location of the leading edge of the slug at which the user wants to stop simulating. The default value used in the Lua script is 3.25 inches, at which time the slug's leading edge is 3.25 inches past the center of the coil. In this position, the slug's trailing edge is one-quarter inch past the last turn of the coil. This termination condition can be over-ridden by the other two termination conditions. There is another user-set parameter called MaxNumTimeSteps (default 100,000) which stops execution after that number of time steps. And, of course, the simulation will end if the main current i_1 falls to zero and causes the SCR to turn off.

The only parameter left to specify is the mass of the slug. Its physical dimensions were described in the earlier papers. Its volume can be calculated from these. The slug is 0.5", or 0.0127 m, in diameter with a length of 2", or 0.0508 m. Its volume is $\pi R^2 L = \pi \times 0.00635^2 \times 0.0508 = 6.435 \times 10^{-6}$ m^3. The density of iron is approximately 7,874 kg/m^3, so the mass of the slug is approximately 0.05067 kg, or 50.67 grams.

The energy balance

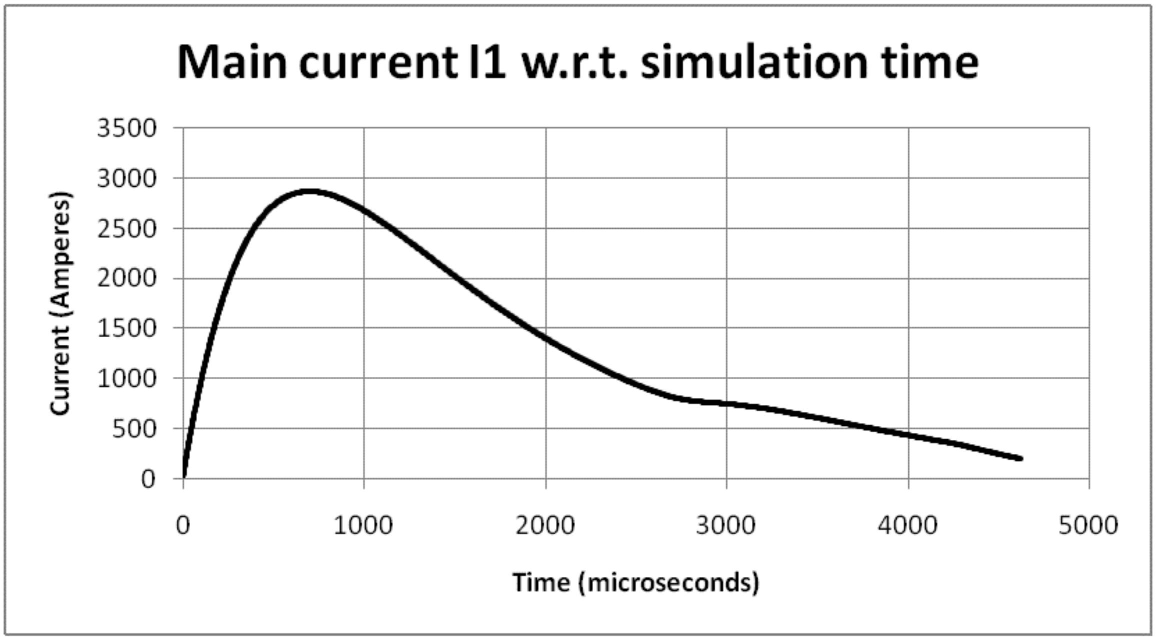
In the earlier paper, there were four sources and uses of energy: (i) electrostatic energy due to the charge stored in the capacitor, (ii) magnetostatic energy in the magnetic field set up by the current flowing through the copper wires of the coil, (iii) an additional magnetostatic energy contribution made by the presence of the slug, and (iv) the cumulative energy burned off as heat by the fixed resistors. Now that the slug can move, a fifth component of the system's total energy is the kinetic energy possessed by the slug when it is moving. The presence of the protection diode also adds another bit to the cumulative heat. If current flows through the real diode, the power consumed is the product of that current (i_2) muliplied by the voltage drop over the capacitor ($2.5V$ has been assumed above).

The kinetic energy possessed by the slug is a "state" variable. It depends only on the instantaneous mass and speed of the slug; it does not depend on the history of the slug's previous movement. At the end of any time step, the slug's kinetic energy can be computed as:

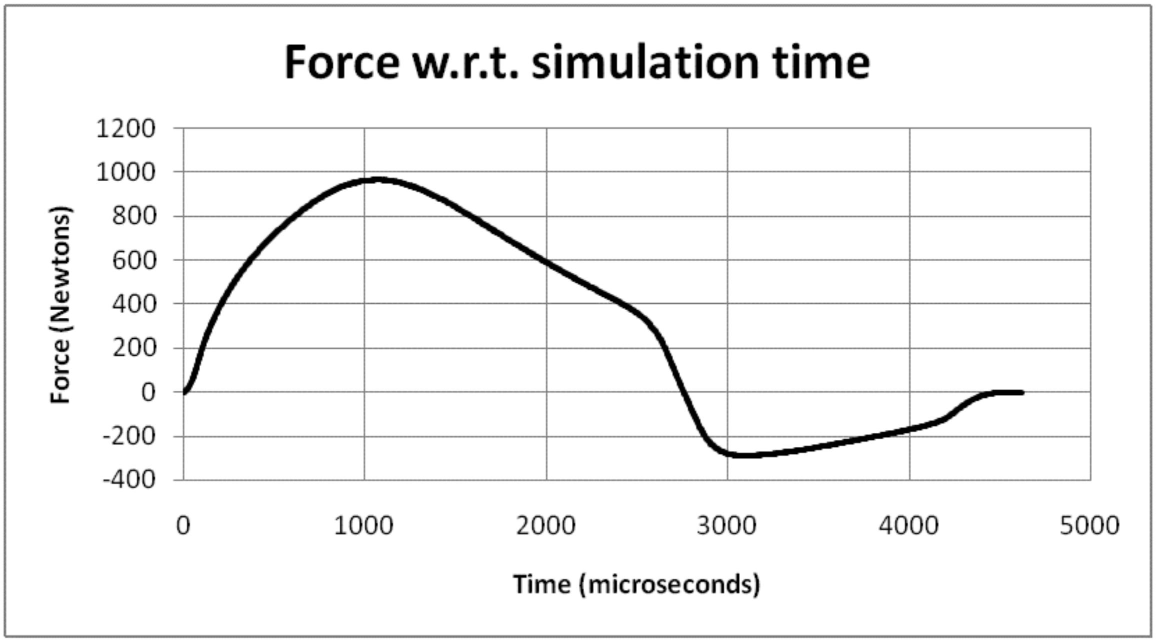
$$KE|_{end} = ½M\left(\frac{dz}{dt}\bigg|_{end}\right)^2 \qquad (49)$$

Results of the simulation

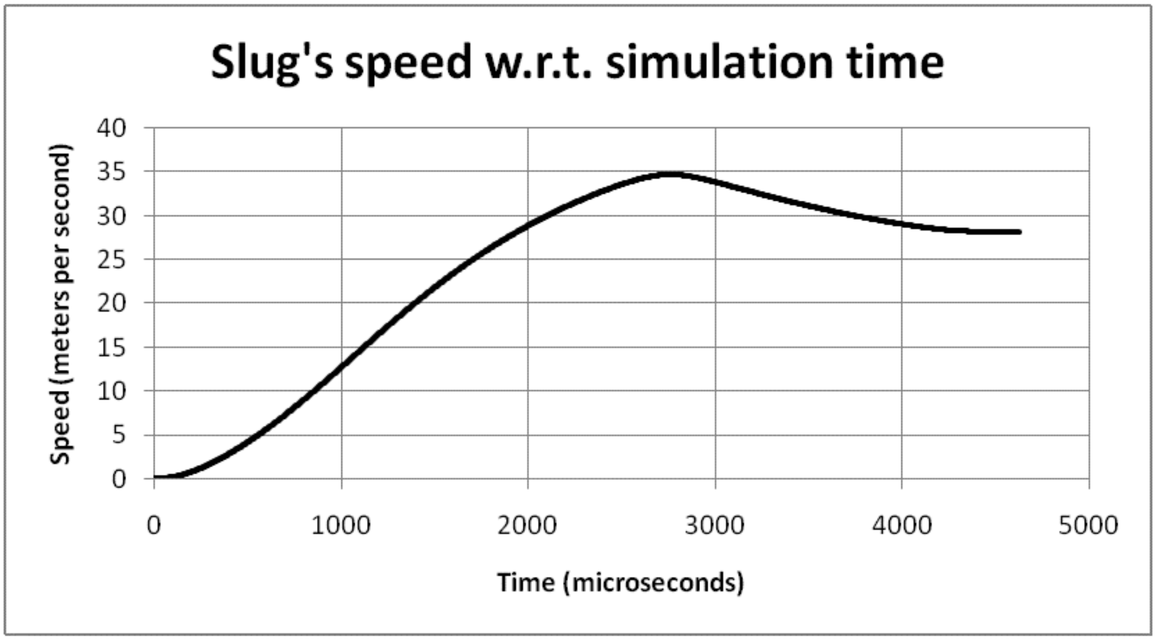
The following graph shows the main current i_1 as a function of simulation time.



The following graph shows the force exerted by the coil on the slug as a function of time.

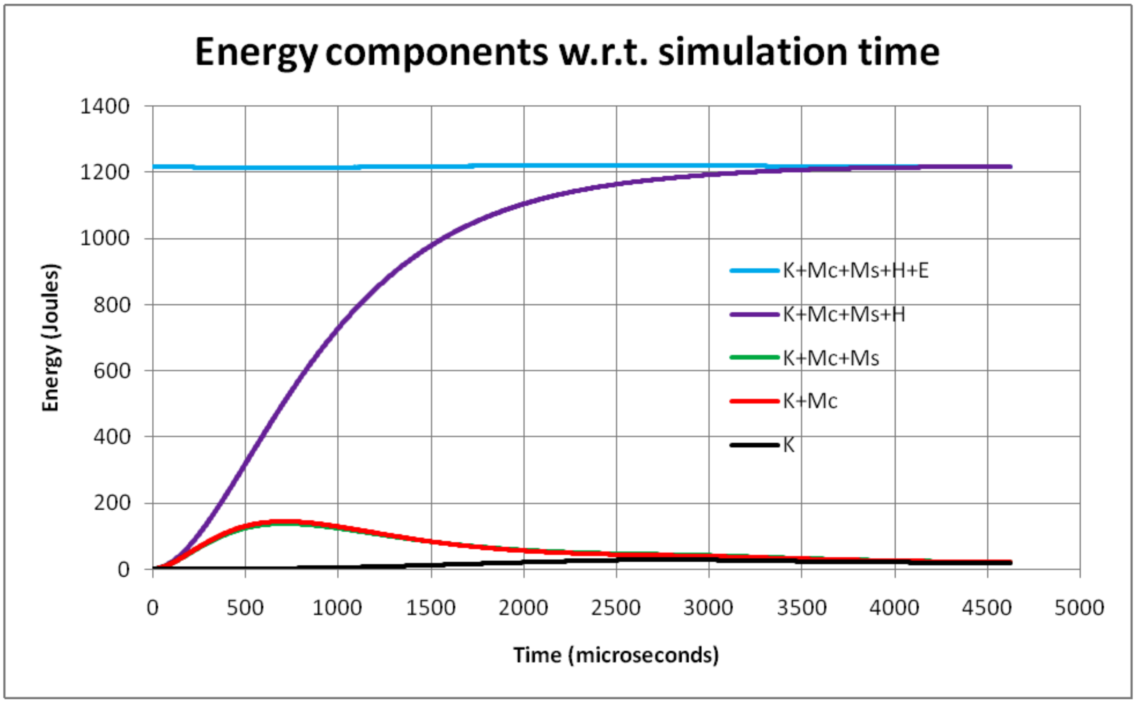


The following graph shows the speed of the slug as a function of time.

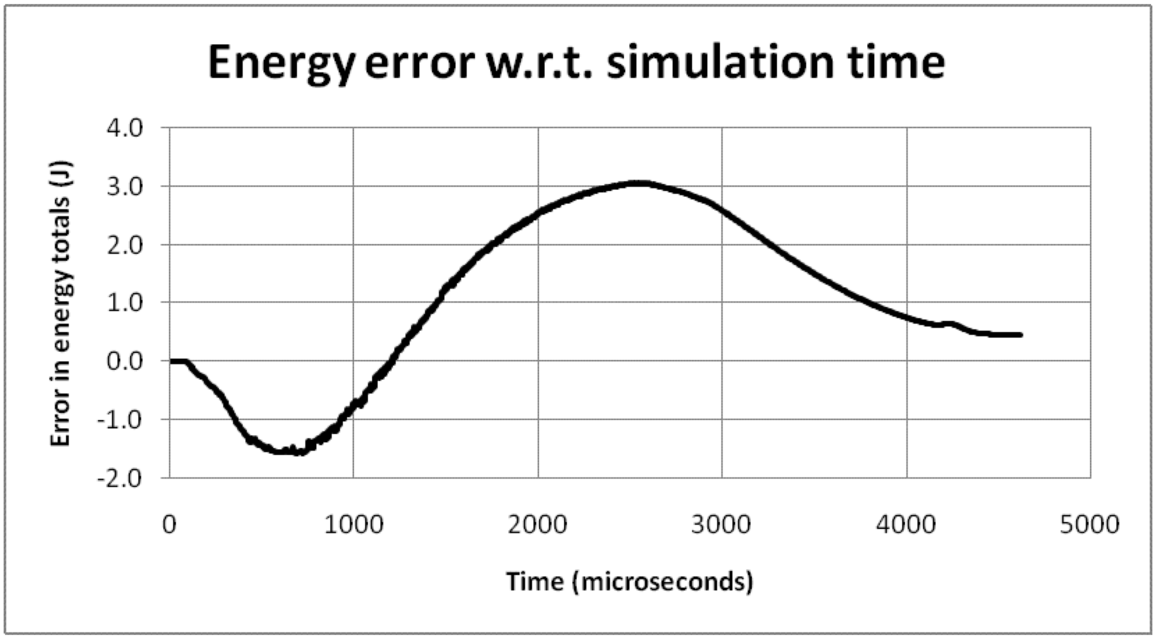


The slug reaches its maximum speed, about 35 meters per second, at the instant it is centered inside the coil. Unfortunately, since the main current continues to flow thereafter, the force (which now changes direction) begins to decelerate the slug. It leaves the coil at about 28 meters per second. Without any means to turn the SCR off, this lingering current and deceleration is problem which is not going to go away. The best we will be able to do is to juggle the slug's starting location and the capacitor's starting voltage (and, in the bigger picture, the size of the capacitance and the configuration of the coil) to get the most from whatever energy is stored inside the capacitor.

The following graph shows the components of the system's total energy as a function of time. The components are layered, as cumulative values added consecutively on top of each other. The slug's kinetic energy (the "K" series in the graph) is the black line near the bottom. The slug leaves the coil with 20.13 Joules of kinetic energy, only 1.65% of the energy initially stored in the capacitor. Adding the magnetostatic energy stored in the coil (the "Mc" label in the graph) gives the red curve. Adding the magnetostatic energy stored in the slug (the "Ms" label in the graph) gives the green curve. Just as we found in the earlier paper, the slug consumes part of the coil's magnetic energy, with the result that it reduces the total magnetic energy in the system. That is why the green curve lies below the red curve. Most of the initial energy is burned off as heat (the "H" label in the graph). Lastly, adding the electrostatic energy stored in the capacitor (the "E" label in the graph) gives the total system energy.



A careful look at the graph above shows that our calculation of the total system energy is slightly off. The following graph shows in more detail the difference between the sum of the energy components as I calculated them and the system's initial energy.



At the worst, our calculations are off by three Joules, or 0.25%, of the 1217 Joules initially stored in the capacitor.

I should point out that the diode protection circuit was never activated during this simulation.

How well does the partial derivative $\partial L_{total}/\partial z$ represent the force F?

An outstanding question is how well the force F tracks the partial derivative $\partial L_{total}/\partial z$ (or is it the other way around)? There should be an equivalence between the mechanical power delivered to the slug and the electrical power absorbed as the inductance changes due to the slug's location. The relationship is given by Equation (18).

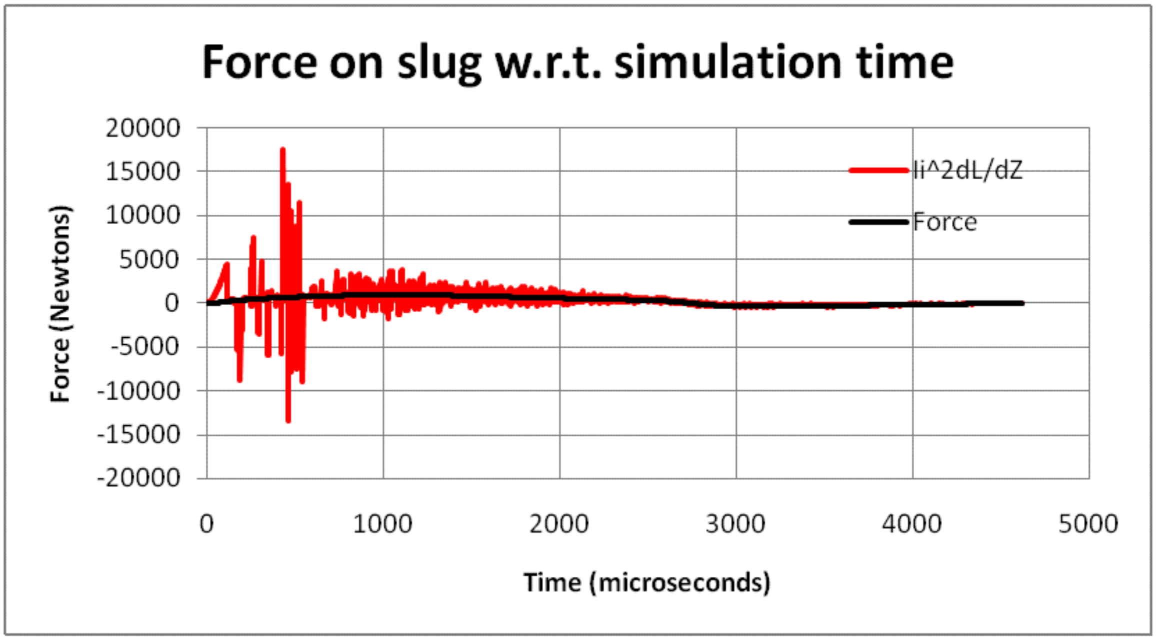
$$\frac{\partial L_{total}}{\partial z} = \frac{F}{i_1^2} \qquad (18)$$

Since this equality represents the interface between the world of mechanical things and the world of electrical things, we should be able to confirm it.

What I did is this. At the start of every time step, I calculated both the force $F|_{start}$ and the partial derivative $\partial L_{total}/\partial z|_{start}$. Multiplying the partial derivative by the square of the starting current should give the equality:

$$F|_{start} = (i_1|_{start})^2 \left.\frac{\partial L_{total}}{\partial z}\right|_{start} \qquad (19)$$

The following graph shows the left-hand side and right-hand side values for my first attempt.



The black line is the force which was plotted above in the graph titled *Force w.r.t. simulation time*. The other calculation of the force – the partial derivative multiplied by the square of the current – is plotted in red. The partial derivative calculation is highly variable, although centered on the directly-calculated force, and swamps the latter on the scale of this graph.

What I have not disclosed yet is the difference in location on which I based the changes used to estimate the partial derivative. When calculating the differences in inductance, I used the following changes in location:

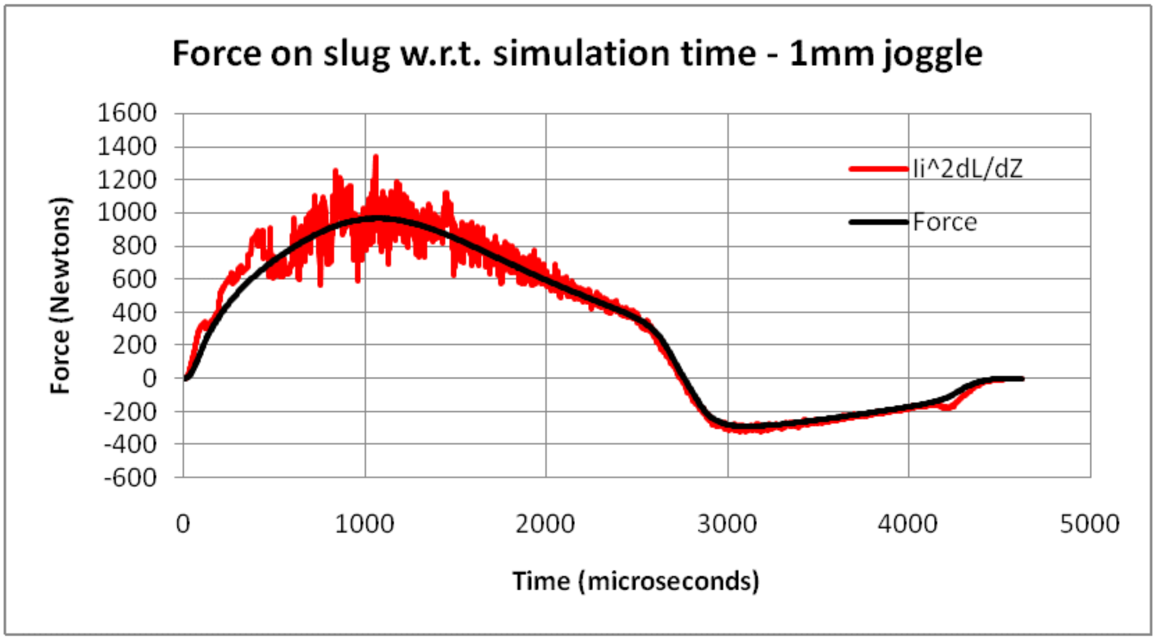
- If the change in the slug's location during the previous 1µs time step was less than one millimeter, then the joggle in the slug's location was set to 0.01 millimeter, or
- If the change in the slug's location during the previous 1µs time step was more than one millimeter, then the joggle in the slug's location was set of 0.1 millimeter.

As an example of the former usage,

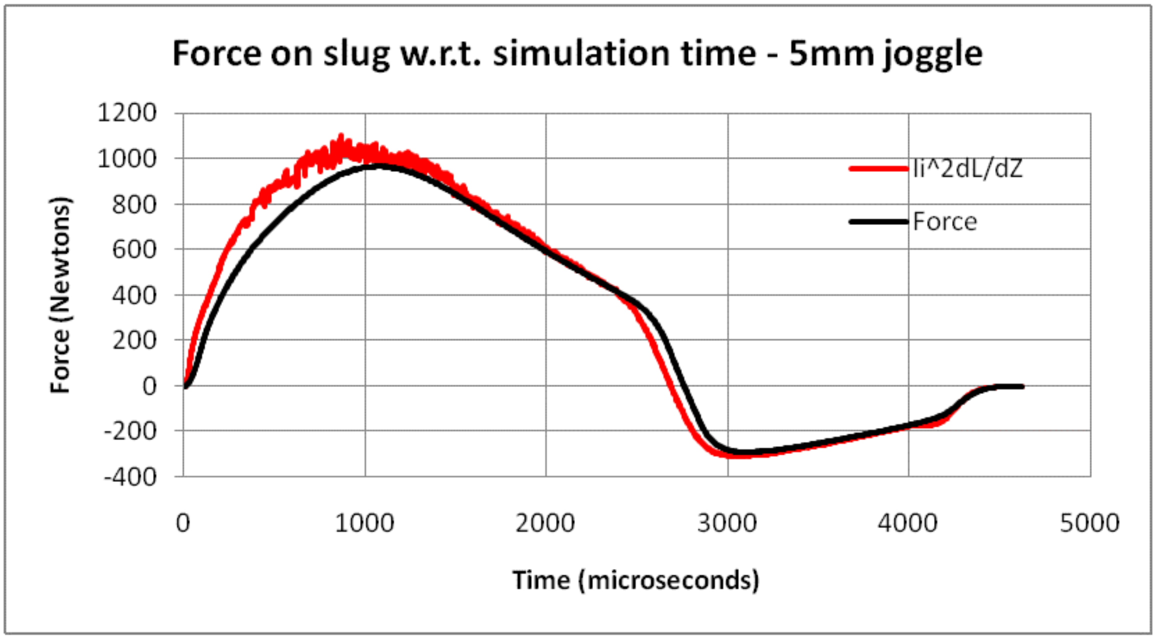
$$\left.\frac{\partial L_{total}}{\partial z}\right|_{start} = \frac{L_{total}(at\ z|_{start}) - L_{total}(at\ z|_{start} + 0.01mm)}{0.01mm}$$

It would seem that the calculation of the partial derivative is sensitive to the magnitude of the joggle distance. This is similar to the phenomenon we encountered when we used this approach to estimate the partial derivative of the inductance with respect to current. It made a difference whether the joggle in the current was one, ten or 100 Amperes. We concluded in the earlier paper that ten Amperes was a suitable compromise.

As a test, I changed the joggle distance. The following graph shows the two sides of Equation (19) when the joggle distance was set to one millimeter.



This is quite a bit better. As a further test, I increased the joggle distance to five millimeters. The result is shown in the following graph. Although the variability is reduced further, a different problem is becoming apparent. Five millimeters is too much of a joggle, when compared with the actual change in the slug's location during one time step. It causes a consistent overestimate of the partial derivative.



I should point out that all of the inductances, including those on which the partial derivatives are founded, were calculated using FEMM's default precision (MaxResidual of 1×10^{-8}) and a rather large maximum triangle size (MaxMeshSize of 0.01 inches). It could be that increasing the precision of FEMM's calculations would bring about a closer agreement between the two approaches to calculating the force. I have not explored this. Based on the analysis so far, I am comfortable that mechanical-electrical interface is being handled satisfactorily.

For the sake of completeness, I have attached as Appendix "A" a listing of the Lua script which I used for this paper.

The next step, which I intend to address in a subsequent paper, is to streamline this process so that it can be used for optimization. On my computer, one of these runs takes about six hours, which is too long for practical use. Removing the sections of code which calculate the spatial partial derivative $\partial L_{total}/\partial z$ reduces runtime by about an hour. Removing the sections of code which confirm conservation of energy makes only a small reduction in runtime. Both of these sections exist only to confirm that the equations are internally consistent, and they seem to have done their job.

Jim Hawley
© March 2015

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 FEMM solver parameters
MaxMeshSize = 0.01
MaxResidual = 1E-8

-- 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',MaxResidual,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,AirInnerRadiu
s)

-- 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,MaxMeshSize,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,MaxMeshSize,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,MaxMeshSize,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,MaxMeshSize,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 an earlier paper titled "Using FEMM to analyze a coil  **
--** from a coil gun", except setting the maximum mesh size and precision.           **
--**************************************************************************************

-- Set the component values
Cap = 0.016            -- Ideal capacitance, in Farads
Resr = 0.038654        -- Capacitor's ESR, in Ohms
Rcoil = 0.05955534     -- Ohmic resistance of coil, in Ohms
Rhook = 0.004014       -- Resistance of hookup wire, in Ohms
Rscr = 0.0015          -- On-resistance of SCR, in Ohms
Vdiode = 2.5           -- Threshold voltage of ideal diode, in Volts
Mass = 0.05067         -- Mass of slug, in kilograms
Rfixed = Rcoil + Rhook + Rscr + Resr

-- Set the initial conditions; leave the slug where it was when we built the geometry
VCap_0 = 390           -- Capacitor is charged up to 390V
I1_0 = 0               -- No current flowing in primary loop
I2_0 = 0               -- No current flowing in protection loop
LELocation_0 = -1      -- The slug is poised at the near lip of the coil
Speed_0 = 0            -- In the base case, the slug is at rest
Distance_0 = 0          -- The slug has not started moving yet

-- Definition of electrical variables at the start and end of a time step
-- QCap_start          -- Charge stored in the ideal capacitor, in Coulombs
-- QCap_end
-- I1_start            -- Current flowing in primary circuit, in Amperes
-- I1_end
-- I2_start            -- Current flowing in protection circuit, in Amperes
-- I2_end
-- dI1dt_constant      -- First derivative of current at start of time step
-- dQCap_constant      -- First derivative of charge ar start of time step
-- LTotal_start        -- Total inductance at start of time step, in Henries
-- LCoil               -- (Constant) inductance of the coil, in Henries
-- LSlug_start         -- Additional inductance due to the slug, in Henries
-- dLSlugdI1_start     -- Partial derivative of LSlug w.r.t. current
-- dLSlugdZ_start      -- Partial derivative of LSlug w.r.t. location
-- VCap_start          -- Voltage drop over the ideal capacitor
-- VCap_end
-- VCap_Real_end       -- Voltage drop over real capacitor
-- VCap_Asterisk       -- Voltage drop over ideal capacitor at start of protection

-- Definition of fixed resistance voltage drops, in Volts, at the end of time step
-- V_Resr              -- Voltage drop over capacitor's ESR
-- V_Rcoil             -- Voltage drop over coil's Ohmic resistance
-- V_Rhook             -- Voltage drop over the hookup wire
-- V_Rscr              -- Voltage drop over the SCR

-- Definition of kinematic variables at the start and end of a time step
-- Force_start         -- Force on slug, in Newtons, at start of time step
-- Accel_constant      -- Acceleration of slug, in m/s^2, constant in time step
-- Speed_start         -- Speed of slug, in m/s
-- Speed_end
-- Distance_start      -- Distance travelled by slug from start, in meters
-- Distance_end
-- LELocation_start    -- Location of leading edge of slug, in inches
-- LELocation_end

-- Circuit configuration control flags
IsSCRConducting = "Yes"  -- When the SCR stops conducting, this will be " No"
IsDiodeConducting = "No" -- When the diode starts conducting, this will be "Yes"

-- Definition of energies, in Joules
-- deltaH                -- Heat burned off by fixed resistors during one time step
-- deltaD                -- Heat burned off by protection diode in one time step
-- ERheat_end            -- Cumulative heat burned off by resistors, at end of time step
-- EDheat_end            -- Cumulative heat burned off by diode, at end of time step
-- ECap_end              -- Electrical energy stored in ideal capacitor, at end of time step
-- ECoil_end             -- Magnetic energy stored in ideal coil, at end of time step
-- EMslug_end            -- Magnetic energy stored in slug, at end of time step
-- EKslug_end            -- Kinetic energy of slug, at end of time step
-- EnergyTotal           -- Total system energy, at end of time step
-- EnergyError           -- Difference between total system energy and initial energy

-- Variables related to the simulation process
deltaT = 0.000001        -- The duration of one time step, in seconds
LELocation_Stop = 3.25   -- Stop simulating when slug reaches here, in inches
SaveInterval = 1         -- Save results to file after this number of time steps
SaveCounter = 0          -- Counter of time steps between save events
MaxNumTimeSteps = 100000 -- Maximum number of time steps permitted

-- As a preliminary step, move the slug in the FEMM construction to the
-- location where we want it to be at the start of the simulation.  We
-- have to do this because we built the physical configuration with the
-- slug placed in a rather arbitrary spot.
mi_seteditmode("group")                 -- Make edits such as translation by Group
CurrentLELocation = SlugLeadingEdgeZ    -- This is the spot where we built the slug
DesiredLELocation = LELocation_0        -- This is where we want the slug to be
RequiredTranslation = DesiredLELocation - CurrentLELocation
mi_clearselected()
mi_selectgroup(2)                       -- The slug was defined as Group(2)
mi_movetranslate(0,RequiredTranslation)
CurrentLELocation = DesiredLELocation   -- Update the slug's location variable
```

```
-- As a second preliminary step, let's calculate the inductance of the coil while
-- the slug is in its starting position.  Note that the value for current specified
-- in function modifycuircuitprop() cannot be zero.   We will set it to something small.
mi_modifycircprop('Coil',1,0.000000000001)
mi_analyze()
mi_loadsolution()
val1,val2,val3 = mo_getcircuitproperties('Coil')
LCoil = val3 / val1

-- Set all variables to their values at the end of time step #0, i.e., at Time=0
VCap_end = VCap_0
QCap_end = Cap * VCap_0
I1_end = I1_0
I2_end = I2_0
Speed_end = Speed_0
Distance_end = Distance_0
LELocation_end = LELocation_0
ERheat_end = 0
EDheat_end = 0
ECap_end = 0.5 * Cap * VCap_0 * VCap_0
ECoil_end = 0
EMslug_end = 0
EKslug_end = 0.5 * Mass * Speed_0 * Speed_0
EnergyTotal_0 = ECap_end

-- Open the text file for output and write a short header
handle = openfile("./Coil_Gun_Simulation.txt","a")
write(handle," ,Firing a coil gun with:\n")
write(handle,"   ,TimeStep = ",deltaT * 1000000," micro-seconds\n")
write(handle,"   ,Capacitor voltage = ",VCap_0," Volts\n")
write(handle,"   ,Slug's LE location = ",SlugLeadingEdgeZ," inches\n")
write(handle,"   ,Slug's initial speed = ",Speed_0," m/s\n")
write(handle,"   ,Wire gauge = ",WireGauge,"\n")
write(handle,"   ,Number of layers = ",NumLayers,"\n")
write(handle,"   ,FEMM maximum mesh size = ",MaxMeshSize,"\n")
write(handle,"   ,FEMM maximum residual = ",MaxResidual,"\n")

-- Write a skipped-column header, to help with importing into Excel
write(handle," ,Time(us), ,LCoil(uH), ,LSlug(uH), ,dLSlug/dI(uH), ,dLSlug/dZ(uH)")
write(handle,", ,QCap(C), ,VCap(V), ,I1_end(A), ,I2_end, ,SlugLE(inch), ,Distance(mm)")
write(handle,", ,Speed(m/s), ,Force(N), ,ECap(J), ,ECoil(J), ,EMslug(J), ,EKslug(J)")
write(handle,", ,ERheat(J), ,EDheat(J), ,ETotal(J), ,EError(J)")
write(handle,", ,Force#2(N), ,F2-F1(N)\n")

-- Write the starting values
-- In this version, output is in "csv" format
write(handle,"Time(us)=,",0,",")
write(handle,"LCoil(uH)=,",LCoil * 1000000,",")
write(handle,"LSlug(uH)=,",0,",")
write(handle,"dLSlug/dI_start(uH)=,",0,",")
write(handle,"dLSlug/dZ_start(uH)=,",0,",")
write(handle,"QCap_end(C)=,",QCap_end,",")
write(handle,"VCap_end(V)=,",VCap_end,",")
write(handle,"I1_end(A)=,",I1_end,",")
write(handle,"I2_end(A)=,",I2_end,",")
write(handle,"SlugLE_end(inch)=,",LELocation_end,",")
write(handle,"Distance_end(mm)=,",0,",")
write(handle,"Speed_end(m/s)=,",Speed_end,",")
write(handle,"Force_start(N)=,",0,",")
write(handle,"ECap_end(J)=,",ECap_end,",")
write(handle,"ECoil_end(J)=,",ECoil_end,",")
write(handle,"EMslug_end(J)=,",EMslug_end,",")
write(handle,"EKslug_end(J)=,",EKslug_end,",")
write(handle,"ERheat_end(J)=,",ERheat_end,",")
write(handle,"EDheat_end(J)=,",EDheat_end,",")
write(handle,"ETotal_end(J)=,",EnergyTotal_0,",")
write(handle,"EError_end(J)=,",0,",")
write(handle,"Force#2_start(N)=,",0,",")
write(handle,"ForceError(N)=,",0,"\n")
SaveCounter = 0

-- Set the screen display as desired
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

--------------------------
-- Begin the simulation --
--------------------------

-- Main loop through time steps
for TimeStep = 1,MaxNumTimeSteps do
    Time = TimeStep * deltaT

    -- Step #1: Bring forward the values from the end of the previous time step
    QCap_start = QCap_end
    I1_start = I1_end
    I2_start = I2_end
    VCap_start = VCap_end
    Speed_start = Speed_end
    LELocation_start = LELocation_end
    Distance_start = Distance_end

    -- Step #2A: Use FEMM to calculate the inductance and the force
    -- Note 1: The slug will already have been moved to LELocation_start.
    -- Note 2: The function modifycircuit() will not accept zero currents so,
    --         if necessary, use a small value instead.
    -- Note 3: The inductance and force do not depend on the direction of flow
    --         of current i1.
    if (abs(I1_start) < 0.000000000001) then
        mi_modifycircprop('Coil',1,0.000000000001)
    else
        mi_modifycircprop('Coil',1,abs(I1_start))
    end
    mi_analyze()
    mi_loadsolution()
    val1,val2,val3 = mo_getcircuitproperties('Coil')
    LTotal_start = val3 / val1
    LSlug_start = LTotal_start - LCoil
    mo_groupselectblock(2)
    Force_start = mo_blockintegral(19)

    -- Step #2B: Use FEMM to calculate the partial derivative dLSlug/dI1
    -- In this version, the change in current is 10 Amperes
    mi_modifycircprop('Coil',1,abs(I1_start + 10))
    mi_analyze()
    mi_loadsolution()
    val1,val2,val3 = mo_getcircuitproperties('Coil')
    LTotal_differentCurrent = val3 / val1
    dLSlugdI1_start = (LTotal_differentCurrent - LTotal_start) / 10
```

```
-- Step #2C: Use FEMM to calculate the partial derivative dLSlug/dZ
-- In this version, two joggle distances are used
if (Distance_start <= 0.001) then
    JoggleDistanceMeters = 0.00001
else
    JoggleDistanceMeters = 0.0001
end
JoggleDistanceInches = JoggleDistanceMeters * 100 / 2.54
-- Return the current to its Step #2A value
if (abs(I1_start) < 0.000000000001) then
    mi_modifycircprop('Coil',1,0.000000000001)
else
    mi_modifycircprop('Coil',1,abs(I1_start))
end
mi_seteditmode("group")                 -- Make edits such as translation by Group
DesiredLELocation = CurrentLELocation + JoggleDistanceInches
RequiredTranslation = DesiredLELocation - CurrentLELocation
mi_clearselected()
mi_selectgroup(2)                       -- The slug was defined as Group(2)
mi_movetranslate(0,RequiredTranslation)
CurrentLELocation = DesiredLELocation   -- Update the slug's location variable
-- Do the analysis
mi_analyze()
mi_loadsolution()
val1,val2,val3 = mo_getcircuitproperties('Coil')
LTotal_differentLocation = val3 / val1
dLSlugdZ_start = (LTotal_differentLocation - LTotal_start) / JoggleDistanceMeters

--//////////////////////////////////////////////////////////////////
--///// First set of circuit equations /////////////////////////////
--//////////////////////////////////////////////////////////////////
if ((IsSCRConducting == "Yes") and (IsDiodeConducting == "No")) then

    -- Step #3: Calculate the derivative dI1/dt_start
    Denominator = LTotal_start + (I1_start * dLSlugdI1_start)
    if (TimeStep == 1) then
        Numerator = QCap_start / Cap
    else
        Numerator = (QCap_start / Cap) +
            (-(Rfixed + (Force_start * Speed_start / (I1_start^2))) * I1_start)
    end
    dI1dt_constant = Numerator / Denominator

    -- Step #4: Integrate twice
    I1_end = I1_start + (dI1dt_constant * deltaT)
    QCap_end = QCap_start +
        (-I1_start * deltaT) +
        (-0.5 * dI1dt_constant * deltaT * deltaT)
    I2_end = 0
end

--//////////////////////////////////////////////////////////////////
--///// Second set of circuit equations ////////////////////////////
--//////////////////////////////////////////////////////////////////
if ((IsSCRConducting == "Yes") and (IsDiodeConducting == "Yes")) then

    -- Step #3B: Calculate the first derivatives dI1dt_start and dQCap/dt_start
    Denominator = LTotal_start + (I1_start * dLSlugdI1_start)
    Rtemp3of4 = Rfixed - Resr
    Numerator =  -Vdiode +
        (-(Rtemp3of4 + (Force_start * Speed_start / (I1_start^2))) * I1_start)
    dI1dt_constant = Numerator / Denominator
    Denominator = Resr
    Numerator = -((QCap_start / Cap) + VCapAsterisk + Vdiode)
    dQCapdt_constant = Numerator / Deniminator

    -- Step #4B: Integrate once, and calculate I2_end
    I1_end = I1_start + (dI1dt_constant * deltaT)
    QCap_end = QCap_start + (dQCapdt_constant * deltaT)
    I2_start = I1_start + dQCapdt_constant
    I2_end = I1_end + dQCapdt_constant
end

--//////////////////////////////////////////////////////////////////
--///// SCR is not conducting.  Continue with dynamics only. ///////
--//////////////////////////////////////////////////////////////////
if (IsSCRConducting == "No") then
    QCap_end = QCap_start
    dQCapdt_end = dQCapdt_start
    I1_end = I1_start
    I2_end = I2_start
    VCap_end = VCap_start
    VCap_Real_end = VCap_end + V_Resr
end

--//////////////////////////////////////////////////////////////////
--///// Rejoin the separate electrical threads /////////////////////
--//////////////////////////////////////////////////////////////////

-- Step #5: Calculate the slug's dynamics
Accel_constant = Force_start / Mass
Speed_end = Speed_start  + (Accel_constant * deltaT)
Distance_end = Distance_start +
    (Speed_start * deltaT) +
    (0.5 * Accel_constant * deltaT * deltaT)

-- Step #6: Convert the change in distance (meters) to location (inches)
LELocation_end = LELocation_start +
    ((Distance_end - Distance_start) * 100 / 2.54)

-- Step #7: Calculate the ending voltage drop over the ideal capacitor
VCap_end = QCap_end / Cap

-- Step #8: Calculate the voltage drop over the real capacitor
V_Resr_end = -Resr * (I1_end - I2_end)
VCap_Real_end = VCap_end + V_Resr_end

-- Step #9: Calculate the heat burned off by the fixed resistors
if ((IsSCRConducting == "Yes") and (IsDiodeConducting == "No")) then
    deltaH = Rfixed * deltaT * (
        (I1_start * I1_start) +
        (I1_start * dI1dt_constant * deltaT) +
        ((1/3) * dI1dt_constant * dI1dt_constant * deltaT * deltaT)  )
end
if ((IsSCRConducting == "Yes") and (IsDiodeConducting == "Yes")) then
    deltaH =
        (Rtemp3of4 * deltaT * (
            (I1_start * I1_start) +
            (I1_start * dI1dt_constant * deltaT) +
            ((1/3) * dI1dt_constant * dI1dt_constant * deltaT * deltaT)  )) +
        (Resr * deltaT * dQCapdt_constant * dQCapdt_constant)
```

```
end
if (IsSCRConducting == "No") then
    deltaH = 0
end
ERheat_end = ERheat_end + deltaH

-- Step #10: Calculate the heat burned off by the protection diode
if ((IsSCRConducting == "Yes") and (IsDiodeConducting == "Yes")) then
    deltaD = Vdiode * deltaT * (
        (I1_start + dQCapdt_constant) +
        (0.5 * dI1dt_constant * deltaT)  )
else
    deltaD = 0
end
EDheat_end = EDheat_end + deltaD

-- Step #11: Calculate the electrostatic energy stored in the ideal capacitor
ECap_end = 0.5 * Cap * VCap_end * VCap_end

-- Step #12: Calculate the magnetic energy stored in the ideal coil
ECoil_end = 0.5 * LCoil * I1_end * I1_end

-- Step #13: Calculate the magnetic energy stored in the slug
EMslug_end =
    (0.5 * LSlug_start * I1_end * I1_end) +
    ((1/3) * dLSlugdI1_start * I1_end * I1_end * I1_end)

-- Step #14: Calculate the kinetic energy of the slug
EKslug_end = 0.5 * Mass * Speed_end * Speed_end

-- Step #15: Calculate the system total energies
EnergyTotal = ERheat_end + EDheat_end +
    ECap_end + ECoil_end + EMslug_end + EKslug_end
EnergyError = EnergyTotal - EnergyTotal_0

-- Step #16: Calculate the force using the second method
Force2_start = I1_start * I1_start * dLSlugdZ_start
ForceError = Force2_start - Force_start

-- Step #17: Physically translate the slug to its new location
mi_seteditmode("group")                 -- Make edits such as translation by Group
DesiredLELocation = LELocation_end
RequiredTranslation = DesiredLELocation - CurrentLELocation
mi_clearselected()
mi_selectgroup(2)                       -- The slug was defined as Group(2)
mi_movetranslate(0,RequiredTranslation)
CurrentLELocation = DesiredLELocation   -- Update the slug's location variable

-- Step #18: Write the results to the output text file
SaveCounter = SaveCounter + 1
if (SaveCounter >= SaveInterval) then
    SaveCounter = 0
    write(handle,"Time(us)=,",Time * 1000000,",")
    write(handle,"LCoil(uH)=,",LCoil * 1000000,",")
    write(handle,"LSlug(uH)=,",LSlug_start * 1000000,",")
    write(handle,"dLSlug/dI_start(uH)=,",dLSlugdI1_start * 1000000,",")
    write(handle,"dLSlug/dZ_start(uH)=,",dLSlugdZ_start * 1000000,",")
    write(handle,"QCap_end(C)=,",QCap_end,",")
    write(handle,"VCap_end(V)=,",VCap_end,",")
    write(handle,"I1_end(A)=,",I1_end,",")
    write(handle,"I2_end(A)=,",I2_end,",")
```

```
        write(handle,"SlugLE_end(inch)=,",LELocation_end,",")
        write(handle,"Distance_end(mm)=,",Distance_end * 1000,",")
        write(handle,"Speed_end(m/s)=,",Speed_end,",")
        write(handle,"Force_start(N)=,",Force_start,",")
        write(handle,"ECap_end(J)=,",ECap_end,",")
        write(handle,"ECoil_end(J)=,",ECoil_end,",")
        write(handle,"EMslug_end(J)=,",EMslug_end,",")
        write(handle,"EKslug_end(J)=,",EKslug_end,",")
        write(handle,"ERheat_end(J)=,",ERheat_end,",")
        write(handle,"EDheat_end(J)=,",EDheat_end,",")
        write(handle,"ETotal_end(J)=,",EnergyTotal,",")
        write(handle,"EError_end(J)=,",EnergyError,",")
        write(handle,"Force#2_start(N)=,",Force2_start,",")
        write(handle,"ForceError(N)=,",ForceError,"\n")
     end

    -- Step #19: Will the SCR be conducting during the next time step?
    if (I1_end < 0) then
        if (IsSCRConducting == "Yes") then
            write(handle,"\nI1 is less than zero, so SCR stops conducting.\n")
            IsSCRConducting = "No"
        end
    end

    -- Step #20: Will the diode be conducting during the next time step?
    if (VCap_Real_end <= -Vdiode) then
        if (IsDiodeConducting == "No") then
            write(handle,"\nVCap_Real is less than zero, so diode starts conducting.\n")
            IsDiodeConducting = "Yes"
            VCapAsterisk = VCap_end
        end
    end
    if ((IsDiodeConducting == "Yes") and (I2_end < -0.0001)) then
        write(handle,"\nI2 has turned negative, so diode stops conducting.\n")
        IsDiodeConducting = "No"
    end

    -- Step #21: Terminate the simulation if the slug has passed through the coil
    if (LELocation_end >= LELocation_Stop) then
        TimeStep = MaxNumTimeSteps + 1
    end

    -- Step #22: Display a message to the user in the Lua window
    print("Time (us) = ", Time * 1000000)
    print("LSlug_start (uH) = ", LSlug_start * 1000000)
    print("dLSlug/dI1_start (uH/A) = ", dLSlugdI1_start * 1000000)
    print("I1 (A) = ",I1_start)
end

-- It is not necessary to remove the temporary wire 5material 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
closefile(handle)
mo_close()
mi_close()

messagebox("All done.")
```