Closed Form Solutions for Potential and Field of a Charged Ring

Kurt Nalty

July 18, 2013

Abstract

This note provides a step by step calculation for the voltage potential and and electric field associated with a charged filament. At the end of the note is a listing of a quick numerical verification of closed form and discrete sum models.

Summary of Results

The source is a ring of charge q and radius R in the XY plane at $z = 0$. The observation point in cylindrical coordinates is (ρ, ϕ, z).

$$
\begin{aligned}
k &= \sqrt{\frac{4\rho R}{(\rho + R)^2 + z^2}} \\
V(\rho, \phi, z) &= \frac{q}{2\pi^2 \epsilon} \frac{1}{\sqrt{(\rho + R)^2 + z^2}} K(k) \\
E_\rho &= \frac{q}{4\pi\epsilon\rho\sqrt{(R+\rho)^2 + z^2}} \left(\frac{K(k)}{\pi} - \left[\frac{E(k)}{\pi} \frac{z^2 + R^2 - \rho^2}{(R-\rho)^2 + z^2} \right] \right) \\
E_\phi &= 0 \\
E_z &= \frac{q}{4\pi\epsilon\rho\sqrt{(\rho + R)^2 + z^2}} \left(\frac{E(k)}{\pi} \frac{2\rho z}{(R-\rho)^2 + z^2} \right)
\end{aligned}
$$

Potential of the Charged Ring

In this section, we will calculate the electrostatic potential associated with the charged ring. Just as a mathematical point charge has an infinity associated with the field at the charge point, we will likewise have an infinity for the field at the ring radius as well. Traditionally, we don't try to measure the voltage at the ring, itself.

For these calculations, I will be using MKS units. I will put a ring of charge q and radius R in the XY plane at $z = 0$. I will place my observation point in cylindrical coordinates at (ρ, ϕ, z), where $\rho = \sqrt{x^2 + y^2}$. Due to cylindrical symmetry, this same potential will be found all around the circle at height z and radial distance same as our test point x coordinate.

My path of integration around the ring will be parameterized by angle θ, starting at the x axis and going counter-clockwise as seen from above. The potential V at an observation point (ρ, ϕ, z) is given by

$$
\begin{aligned}
q' &= \frac{q}{2\pi R} \quad \text{charge density} \\
r &= \sqrt{(x - R\cos\theta)^2 + (y - R\sin\theta)^2 + z^2} \quad \text{distance from ring to observation point} \\
&= \sqrt{x^2 + y^2 + z^2 + R^2 - 2x\cos\theta - 2y\sin\theta} \quad \text{expanded} \\
&= \sqrt{\rho^2 + z^2 + R^2 - 2\rho R\cos\theta} \quad \text{aligning } \theta \text{ and } \phi \\
V &= \frac{1}{4\pi\epsilon} \oint \left(\frac{q}{2\pi R}\right) \frac{1}{r} R d\theta \\
&= \frac{q}{8\pi^2\epsilon} \oint \frac{1}{r} d\theta \\
&= \frac{q}{8\pi^2\epsilon} \oint \frac{d\theta}{\sqrt{\rho^2 + z^2 + R^2 - 2\rho R\cos\theta}} \\
&= \frac{q}{8\pi^2\epsilon} \frac{1}{\sqrt{\rho^2 + z^2 + R^2}} \oint \frac{d\theta}{\sqrt{1 - b^2\cos\theta}} \\
b &= \sqrt{\frac{2\rho R}{\rho^2 + z^2 + R^2}}
\end{aligned}
$$

We look at this, and recognize our $K(k)$ integral in disguise.

$$
\begin{aligned}
K(k) &= \frac{1}{4}\sqrt{1+b^2} \oint \frac{d\theta}{\sqrt{1-b^2\cos\theta}} \\
\oint \frac{d\theta}{\sqrt{1-b^2\cos\theta}} &= \frac{4}{\sqrt{1+b^2}} K(k) \\
&= \frac{4}{\sqrt{1+b^2}} K\left(\sqrt{\frac{2b^2}{1+b^2}}\right)
\end{aligned}
$$

$$
\begin{aligned}
b^2 &= \frac{2\rho R}{\rho^2+z^2+R^2} \\
1+b^2 &= \frac{2\rho R+\rho^2+z^2+R^2}{\rho^2+z^2+R^2} \\
V &= \frac{q}{8\pi^2\epsilon}\frac{1}{\sqrt{\rho^2+z^2+R^2}} \oint \frac{d\theta}{\sqrt{1-b^2\cos\theta}} \\
&= \frac{q}{8\pi^2\epsilon}\frac{1}{\sqrt{\rho^2+z^2+R^2}}\frac{4}{\sqrt{1+b^2}} K\left(\sqrt{\frac{2b^2}{1+b^2}}\right) \\
&= \frac{q}{2\pi^2\epsilon}\frac{1}{\sqrt{2\rho r+\rho^2+z^2+R^2}} K\left(\sqrt{\frac{2b^2}{1+b^2}}\right)
\end{aligned}
$$

We now substitute for b^2 to get a nice, standalone expression.

$$
\begin{aligned}
V(\rho,\phi,z) &= \frac{q}{2\pi^2\epsilon}\frac{1}{\sqrt{2\rho R+\rho^2+z^2+R^2}} K\left(\sqrt{\frac{4\rho R}{2\rho R+\rho^2+z^2+R^2}}\right) \\
&= \frac{q}{2\pi^2\epsilon}\frac{1}{\sqrt{(\rho+R)^2+z^2}} K\left(\sqrt{\frac{4\rho R}{(\rho+R)^2+z^2}}\right) \\
V(\rho,\phi,z) &= \frac{q}{2\pi^2\epsilon}\frac{1}{\sqrt{(\rho+R)^2+z^2}} K(k) \\
k &= \sqrt{\frac{4\rho R}{(\rho+R)^2+z^2}}
\end{aligned}
$$

The voltage along the X axis is plotted in Figure 1. The voltage along the Z axis is plotted in Figure 2.

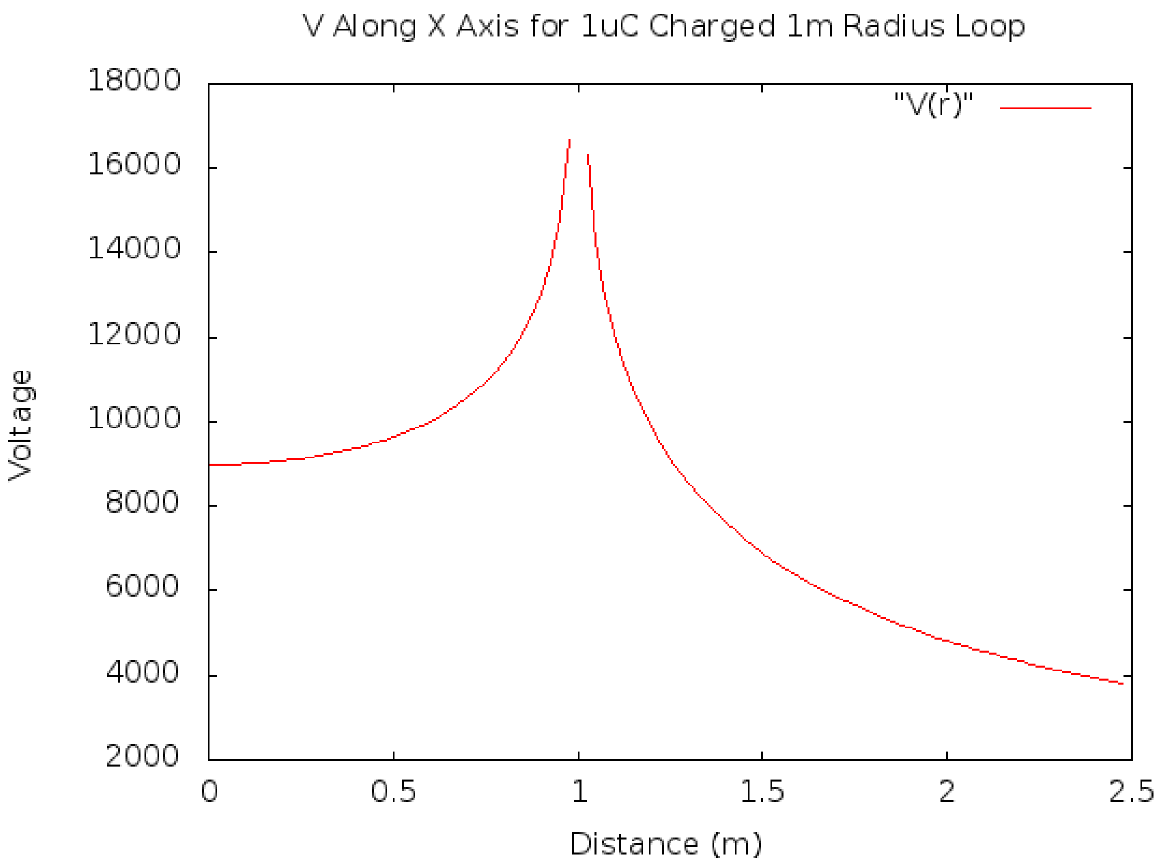

Figure 1: Voltage Along X Axis for a 1 μC, 1 m Radius Loop

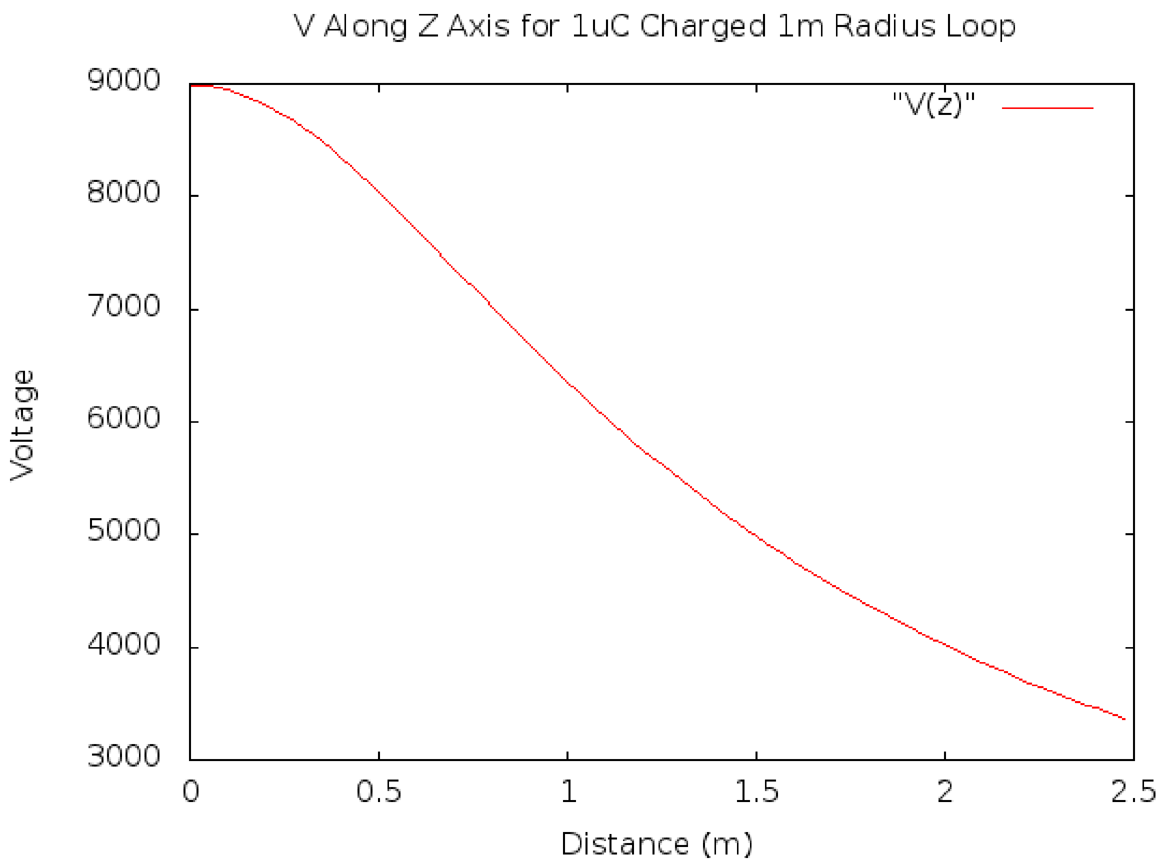

Figure 2: Voltage Along Z Axis for a 1 μC, 1 m Radius Loop

Static Electric Field

We are working in cylindrical coordinates. The gradient components are

$$\begin{aligned}
\vec{E} &= -\vec{\nabla} V \\
\vec{\nabla} V &= \vec{a}_\rho \frac{\partial V}{\partial \rho} + \vec{a}_\phi \frac{1}{\rho} \frac{\partial V}{\partial \phi} + \vec{a}_z \frac{\partial V}{\partial z}
\end{aligned}$$

Repeating our potentials, we see V is independent of ϕ.

$$\begin{aligned}
k &= \sqrt{\frac{4R\rho}{z^2 + (R+\rho)^2}} \\
V(\rho, \phi, z) &= \frac{q}{2\pi^2 \epsilon} \frac{1}{\sqrt{(R+\rho)^2 + z^2}} K(k)
\end{aligned}$$

The electric field components are then

$$\begin{aligned}
E_\rho &= - \frac{\partial V}{\partial \rho} \\
E_\phi &= 0 \\
E_z &= - \frac{\partial V}{\partial z}
\end{aligned}$$

We will have a number of terms in the derivative expression. We should list some of these before we get started.

$$\begin{aligned}
\frac{\partial K(k)}{\partial \rho} &= \left[\frac{E(k)}{k(1-k^2)} - \frac{K(k)}{k} \right] \frac{\partial k}{\partial \rho} \\
\frac{\partial K(k)}{\partial z} &= \left[\frac{E(k)}{k(1-k^2)} - \frac{K(k)}{k} \right] \frac{\partial k}{\partial z}
\end{aligned}$$

$$
\begin{aligned}
k &= \sqrt{\frac{4R\rho}{z^2 + (R+\rho)^2}} \\
\frac{\partial k}{\partial \rho} &= \frac{1}{2}\sqrt{\frac{4R}{\rho(z^2 + (\rho+R)^2)}} - \frac{1}{2}\frac{\sqrt{4R\rho}(2(R+\rho))}{\left(z^2 + (R+\rho)^2\right)^{3/2}} \\
&= \frac{k}{2}\left[\frac{1}{\rho} - \frac{2(R+\rho)}{z^2 + (R+\rho)^2}\right] \\
&= \frac{k}{2}\left[\frac{z^2 + (R+\rho)^2 - 2(R+\rho)\rho}{\rho(z^2 + (R+\rho)^2)}\right] \\
&= \frac{k}{2\rho}\left[\frac{z^2 + R^2 - \rho^2}{z^2 + (R+\rho)^2}\right]
\end{aligned}
$$

$$
\begin{aligned}
k &= \sqrt{\frac{4R\rho}{z^2 + (R+\rho)^2}} \\
\frac{\partial k}{\partial z} &= \frac{\sqrt{4R\rho}}{\left(z^2 + (R+\rho)^2\right)^{3/2}}\left(-\frac{1}{2}\right)2z \\
\frac{\partial k}{\partial z} &= -\frac{zk}{z^2 + (R+\rho)^2}
\end{aligned}
$$

Cylindrical Radial Field Component

We start with E_ρ.

$$
\begin{aligned}
V(\rho, \phi, z) &= \frac{q}{2\pi^2\epsilon}\frac{1}{\sqrt{(R+\rho)^2 + z^2}}K(k) \\
E_\rho &= -\frac{\partial V}{\partial \rho} \\
&= -\frac{q}{2\pi^2\epsilon}\frac{\partial}{\partial \rho}\left(\frac{1}{\sqrt{(R+\rho)^2 + z^2}}K(k)\right)
\end{aligned}
$$

$$
\begin{aligned}
E_\rho &= -\frac{q}{2\pi^2\epsilon}\left[\frac{\partial}{\partial\rho}\left(\frac{1}{\sqrt{(R+\rho)^2+z^2}}\right)K(k)+\frac{1}{\sqrt{(R+\rho)^2+z^2}}\frac{\partial K(k)}{\partial\rho}\right] \\
&= -\frac{q}{2\pi^2\epsilon}\left[\left(\frac{-(R+\rho)}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)K(k)+\frac{1}{\sqrt{(R+\rho)^2+z^2}}\frac{\partial K(k)}{\partial\rho}\right]
\end{aligned}
$$

Typesetting requires I re-arrange the leading constants to allow line breaks

$$
\begin{aligned}
-\frac{2\pi^2\epsilon}{q}E_\rho &= \left(\frac{-(R+\rho)}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)K(k)+ \\
&\quad \frac{1}{\sqrt{(R+\rho)^2+z^2}}\left[\frac{E(k)}{k(1-k^2)}-\frac{K(k)}{k}\right]\frac{k}{2\rho}\left[\frac{z^2+R^2-\rho^2}{z^2+(R+\rho)^2}\right] \\
&= \left(\frac{-(R+\rho)}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)K(k)+ \\
&\quad \frac{1}{\sqrt{(R+\rho)^2+z^2}}\left[\frac{E(k)}{(1-k^2)}-K(k)\right]\frac{1}{2\rho}\left[\frac{z^2+R^2-\rho^2}{z^2+(R+\rho)^2}\right] \\
&= \left(\frac{-(R+\rho)}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)K(k)- \\
&\quad \left(\frac{-(R+\rho)}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)\left[\frac{E(k)}{(1-k^2)}-K(k)\right]\frac{1}{2\rho}\left[\frac{z^2+R^2-\rho^2}{(R+\rho)}\right]
\end{aligned}
$$

Re-arranging our new common factor for typesetting, we have

$$
-E_\rho\left(\frac{2\pi^2\epsilon}{q}\right)\left(\frac{\left((R+\rho)^2+z^2\right)^{3/2}}{(R+\rho)}\right) = -K(k)+\left[\frac{E(k)}{(1-k^2)}-K(k)\right]\frac{1}{2\rho}\left[\frac{z^2+R^2-\rho^2}{(R+\rho)}\right]
$$

We are going to work on the right hand side for a little bit.

$$\begin{aligned}
\text{RHS} &= -K(k) + \left[\frac{E(k)}{(1-k^2)} - K(k)\right]\frac{1}{2\rho}\left[\frac{z^2+R^2-\rho^2}{(R+\rho)}\right] \\
k^2 &= \frac{4\rho R}{(R+\rho)^2+z^2} \\
1-k^2 &= \frac{(R-\rho)^2+z^2}{(R+\rho)^2+z^2} \\
\frac{1}{1-k^2} &= \frac{(R+\rho)^2+z^2}{(R-\rho)^2+z^2} \\
\text{RHS} &= -K(k) + \left[E(k)\frac{(R+\rho)^2+z^2}{(R-\rho)^2+z^2} - K(k)\right]\frac{1}{2\rho}\left[\frac{z^2+R^2-\rho^2}{(R+\rho)}\right]
\end{aligned}$$

$$\begin{aligned}
\text{RHS} &= -K(k) + \left[E(k)\frac{(R+\rho)^2+z^2}{(R-\rho)^2+z^2} - K(k)\right]\frac{1}{2\rho}\left[\frac{z^2+R^2-\rho^2}{(R+\rho)}\right] \\
&= -K(k) + \left[E(k)\frac{(R+\rho)^2+z^2}{(R-\rho)^2+z^2} - K(k)\right]\left[\frac{z^2+R^2-\rho^2}{(2\rho R+2\rho^2)}\right] \\
&= -K(k)\left[1+\frac{z^2+R^2-\rho^2}{(2\rho R+2\rho^2)}\right] + \left[E(k)\frac{(R+\rho)^2+z^2}{(R-\rho)^2+z^2}\right]\left[\frac{z^2+R^2-\rho^2}{(2\rho R+2\rho^2)}\right] \\
&= -K(k)\left[\frac{(R+\rho)^2+z^2}{(2\rho R+2\rho^2)}\right] + \left[E(k)\frac{(R+\rho)^2+z^2}{(R-\rho)^2+z^2}\right]\left[\frac{z^2+R^2-\rho^2}{(2\rho R+2\rho^2)}\right] \\
\text{RHS} &= \left[\frac{(R+\rho)^2+z^2}{2\rho(R+\rho)}\right]\left(-K(k) + \left[E(k)\frac{z^2+R^2-\rho^2}{(R-\rho)^2+z^2}\right]\right) \\
\text{LHS} &= -E_\rho\left(\frac{2\pi^2\epsilon}{q}\right)\left(\frac{\left((R+\rho)^2+z^2\right)^{3/2}}{(R+\rho)}\right)
\end{aligned}$$

Re-uniting, and clearing common factors, we have

$$\begin{aligned}
E_\rho &= \frac{q}{2\pi\rho}\frac{1}{2\pi\epsilon}\frac{1}{\sqrt{(R+\rho)^2+z^2}}\left(K(k) - \left[E(k)\frac{z^2+R^2-\rho^2}{(R-\rho)^2+z^2}\right]\right) \\
&= \frac{q}{4\pi\epsilon\rho\sqrt{(R+\rho)^2+z^2}}\left(\frac{K(k)}{\pi} - \left[\frac{E(k)}{\pi}\frac{z^2+R^2-\rho^2}{(R-\rho)^2+z^2}\right]\right)
\end{aligned}$$

Axial Field Component

$$
\begin{aligned}
E_z &= -\frac{\partial V}{\partial z} \\
&= -\left(\frac{q}{2\pi^2\epsilon}\right)\frac{\partial}{\partial z}\left(\frac{1}{\sqrt{(R+\rho)^2+z^2}}K(k)\right) \\
-\left(\frac{2\pi^2\epsilon}{q}\right)E_z &= \left(\frac{\partial}{\partial z}\frac{1}{\sqrt{(R+\rho)^2+z^2}}\right)K(k)+\frac{1}{\sqrt{(R+\rho)^2+z^2}}\frac{\partial K(k)}{\partial z}
\end{aligned}
$$

As before, we will work on the right hand side.

$$
\begin{aligned}
-\left(\frac{2\pi^2\epsilon}{q}\right)E_z &= \left(\frac{\partial}{\partial z}\frac{1}{\sqrt{(R+\rho)^2+z^2}}\right)K(k)+\frac{1}{\sqrt{(R+\rho)^2+z^2}}\frac{\partial K(k)}{\partial z} \\
&= \left(\frac{-z}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)K(k)+\frac{1}{\sqrt{(R+\rho)^2+z^2}}\frac{\partial K(k)}{\partial z} \\
&= \left(\frac{-z}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)K(k) \\
&\quad +\frac{1}{\sqrt{(R+\rho)^2+z^2}}\left[\frac{E(k)}{k(1-k^2)}-\frac{K(k)}{k}\right]\frac{\partial k}{\partial z} \\
&= \left(\frac{-z}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)K(k) \\
&\quad +\frac{1}{\sqrt{(R+\rho)^2+z^2}}\left[\frac{E(k)}{k(1-k^2)}-\frac{K(k)}{k}\right]\left(-\frac{zk}{z^2+(R+\rho)^2}\right) \\
&= \left(\frac{-z}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)K(k) \\
&\quad -\left(\frac{z}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)\left[\frac{E(k)}{(1-k^2)}-K(k)\right] \\
&= -\left(\frac{z}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)\left[\frac{E(k)}{(1-k^2)}\right]
\end{aligned}
$$

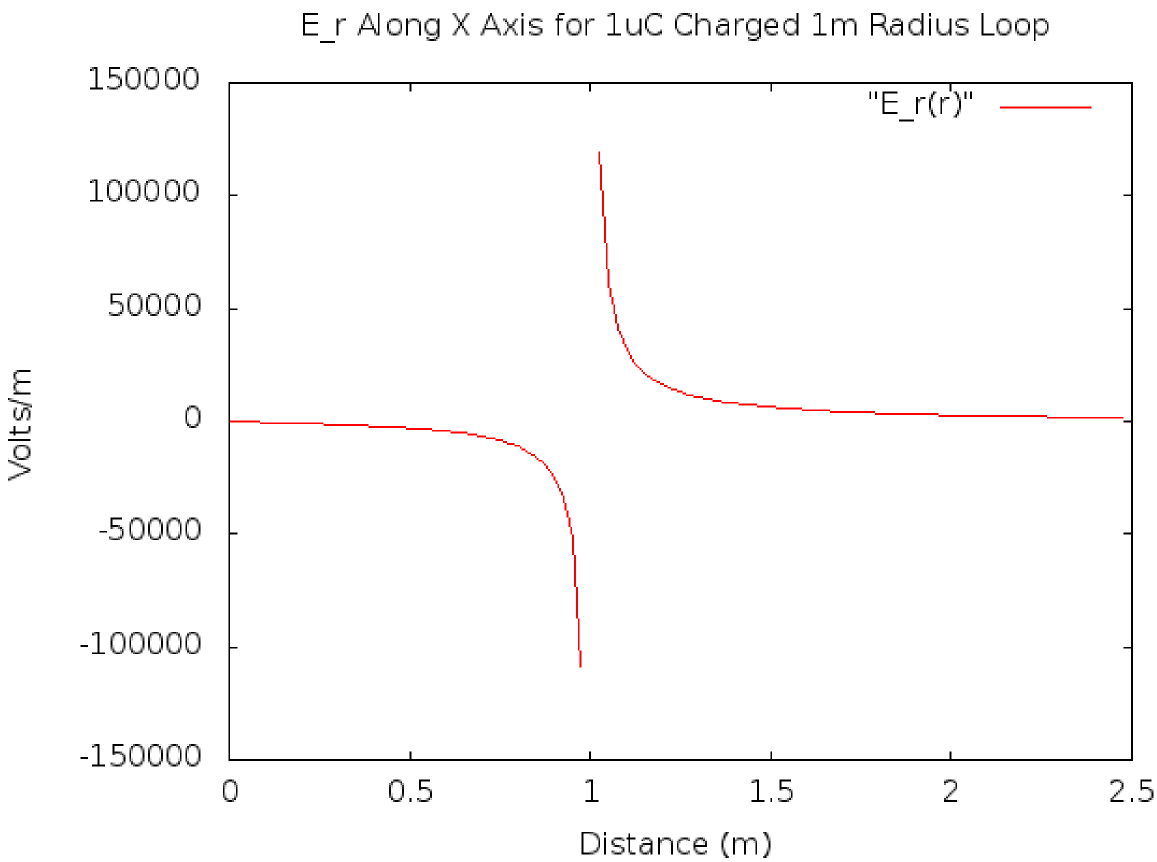

Figure 3: Radial Electric Field Along X Axis for a 1 μC, 1 m Radius Loop

So,

$$
\begin{aligned}
E_z &= \left(\frac{q}{2\pi^2\epsilon}\right)\left(\frac{z}{\left((R+\rho)^2+z^2\right)^{3/2}}\right)\left[E(k)\frac{(R+\rho)^2+z^2}{(R-\rho)^2+z^2}\right] \\
&= \frac{q}{4\pi\epsilon\rho\sqrt{(\rho+R)^2+z^2}}\left(\frac{E(k)}{\pi}\frac{2\rho z}{(R-\rho)^2+z^2}\right)
\end{aligned}
$$

The radial electric field along the X axis is plotted in Figure 3. The axial electric field along the Z axis is plotted in Figure 4.

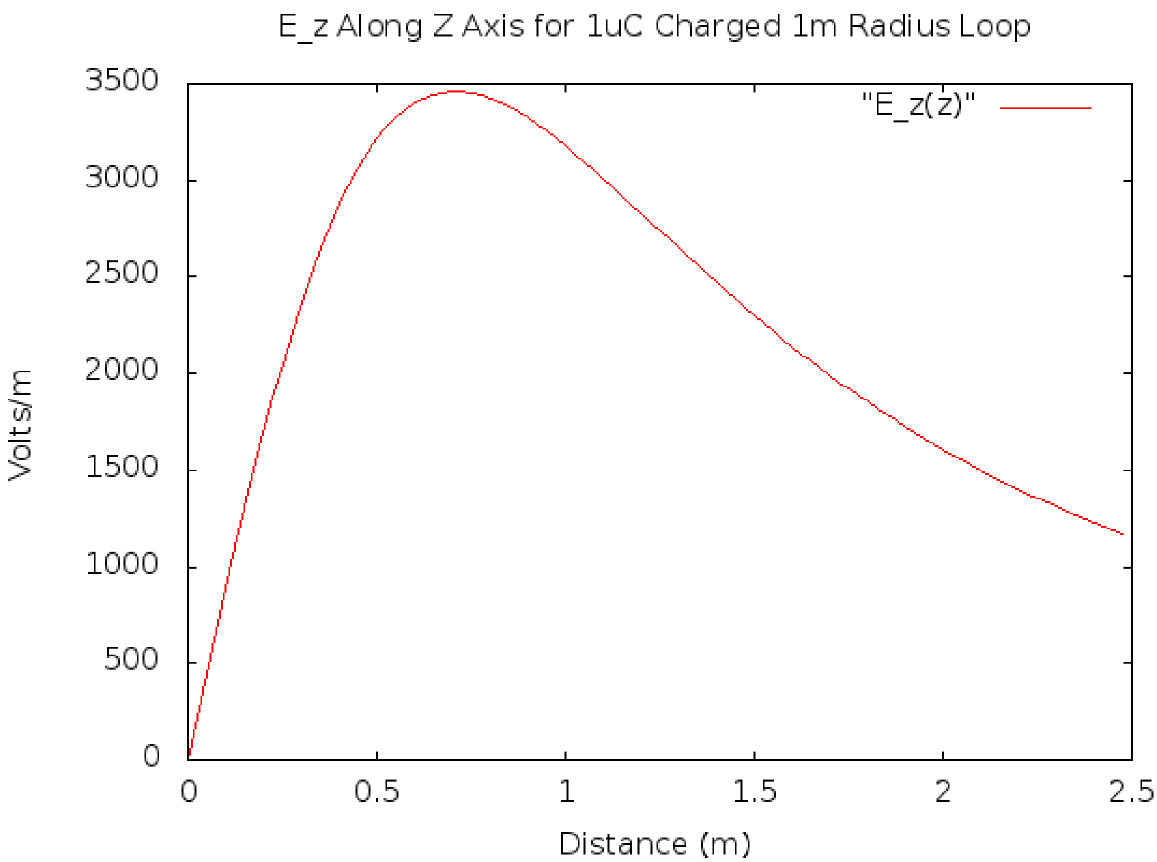

Figure 4: Axial Electric Field Along Z Axis for a 1 μC, 1 m Radius Loop

Verification Code

As a check against careless mistake, a simple C program is provided. This program calculates the E(k) and K(k) functions for comparison against published tables, then calculates the potential and electric field of a charged ring numerically by simple integration using MKS units. The program then calculates the same potential and fields using the elliptic integral formulas, and present both results. This program can be downloaded from `http://www.kurtnalty.com/Charged_Ring.c`.

```
/*

This is a program to calculate the voltage potential due
to a charged ring with charge q at radius R in the x-y plane.

The voltage is calculated at the point (x,0,z) using two methods.
The first method is a sum over 360 segments, approximating each segment
as a charged point.
The second method is using the complete elliptic integral of the first kind K.

*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

typedef struct
{
    double x;
    double y;
    double z;
} Vector;

Vector Cross(Vector A, Vector B)
{
    Vector C;
    C.x = A.y*B.z - A.z*B.y;
    C.y = A.z*B.x - A.x*B.z;
```

```
    C.z = A.x*B.y - A.y*B.x;
    return (C);
};

void PrintVector(Vector A)
{
    printf("( %e, %e, %e )",A.x,A.y,A.z);
}

Vector AddVector(Vector A, Vector B)
{
    Vector C;
    C.x = A.x + B.x;
    C.y = A.y + B.y;
    C.z = A.z + B.z;
    return C;
}

Vector SubVector(Vector A, Vector B)
{
    Vector C;
    C.x = A.x - B.x;
    C.y = A.y - B.y;
    C.z = A.z - B.z;
    return C;
}

Vector ScaleVector(Vector A, double scale)
{
    Vector B;
    B.x = A.x*scale;
    B.y = A.y*scale;
    B.z = A.z*scale;
    return B;
}

float Dot(Vector A, Vector B)
{
```

```
    return (A.x*B.x + A.y*B.y + A.z*B.z);
}


int Complete_Elliptic_K_and_E
        (char arg, double parameter, double* K, double* E)
{
/*    Usage: the character arg is 'k', 'm', or 'a' to inform this
             routine of the type of parameter being used. Pointers
             are used for K and E so that we  can return multiple
     values without a struct.

       Example Call: err = Complete_Elliptic_K_and_E('k',k,&K,&E);

        License: Freeware by Kurt Nalty, 2011.

        Credits: Algorithm based upon
         Handbook of Mathematical Functions,
                Abramowitz and Stegun, National Bureau of Standards,
 pp 589-599

        Credit Coding Based After:

   http://mymathlib.webtrellis.net/functions/elliptic_integrals.html

                  (I very much liked his use of a letter parameter for
     function selection)

    returned code:  0 -> no problem
                      -1 -> bad parameter specification
                      -2 -> K sent to infinity by k=1, m=1, or
             alpha = 90 degrees

    Accuracy - seems to be within 5 digits easily.

*/

    double a[10],b[10],c[10];
```

```
    int i;
    double k, m, alpha;
    double pi = 3.141592653589793;
    double S, scale;
    double tol = 1.0e-10,err;    // initial tolerance

// Check arguments

    switch (arg) {
        case 'k':
            k = parameter;
            m = k*k;
            alpha = asin(k);
            break;
        case 'm':
            k = sqrt(parameter);
            m = parameter;
            alpha = asin(k);
            break;
        case 'a':
            k = sin(parameter);
            m = k*k;
            alpha = parameter;
            break;
                default:
            *K = 1.0E30;    // infinity approximated
            *E = 1.0;
            return (-2);    // bad argument
        }

//        if (m > 0.99999999) {
//            *K = 1.0E6;    // infinity approximated
//            *E = 1.0;
//            return (-2);    // bad argument
//        }

/*
    K(m)=/int^{\pi/2}_{0} \left(1-m \sin^2 \theta \right)^{-1/2} d \theta
```

```
    E(m)=/int^{\pi/2}_{0} \left(1-m \sin^2 \theta \right)^{1/2}  d \theta

    For the AGM process

    a[0] = 1.0     b[0] = cos(alpha) = sqrt(1-k*k)
                         c[0] = sin(alpha) = k

    a[i]=0.5*(a[i-1]+b[i-1])  b[i]=sqrt(a[i-1]*b[i-1])
                         c[i]=0.5*(a[i-1]-b[i-1])

    when c[i] is small enough,

    K(alpha) = pi/(2 a[n])

    S = 0.5*Sigma_i (2^i c_i^2) = Sigma_i (2^(i-1) c_i^2)
*/

    a[0] = 1.0;  b[0] = cos(alpha);   c[0] = sin(alpha);
    S = 0.5*c[0]*c[0];  scale = 1.0;

    for (i=1;i<9;i++) {  // max interation depth is 10.
                         // Break if within tolerance

        a[i] = 0.5*(a[i-1] + b[i-1]);
        b[i] = sqrt(a[i-1]*b[i-1]);
        c[i] = 0.5*(a[i-1] - b[i-1]);
        S += scale*c[i]*c[i];
        scale *= 2.0;
        err = (c[i]);
        if (err < 0.0) err *= -1.0;    // take magnitude
        if(err <= tol) break;

    }
    *K = pi/(2.0*a[i]);
    *E = *K - *K*S;
    if (i==9) return(-2);    // no convergence

    return 0;
```

```
}



int main(void)
{
    int i,j;

    Vector deltaR, Sensor;

    Vector SourceCharge[360];
            // current source as vector segments,
            // 1A, tangential direction, offset 0.5 degree
    Vector SourcePosition[360];
            // current location, centered in segment,
            // offset 0.5 degree

    Vector SensorCurrent;

    double theta, dtheta, phi, a, r, R, R2, x, y, z, q, q_prime, dq;
    double pi = 3.141592653589793;
    double k, rho, I, E, K, K_arg;
    double V, dV, epsilon, V_elliptic;
    double C,Alpha, Beta, Gamma;
    double E_x, E_y, E_z, EE_rho, EE_z;
    double scale;
    double rprhoz, rmrhoz;
double E2_rho, E2_z;


    FILE* OutputWireFrame;
    FILE* EFile;
    FILE* KFile;

// First, we provide a table of value for E(k) and K(k)
// for comparison against references.
```

```
    EFile = fopen("E(k)","w");
    KFile = fopen("K(k)","w");
    fprintf(EFile,"k         E(k)  \n\n");
    fprintf(KFile,"k         K(k)  \n\n");
    for (i=0;i<100;i++) {
        k = i/100.0;
        Complete_Elliptic_K_and_E('k',k,&K, &E);
        fprintf(EFile,"%g   %g  \n",k,E);
        fprintf(KFile,"%g   %g  \n",k,K);
    }
    fclose(EFile);
    fclose(KFile);



// Define the sensor position

    Sensor.x = 0.01;
    Sensor.y = 0.0;
    Sensor.z = 3.0;

//  Define the charged ring
    r = 1.0; // 1 meter radius
    q = 1.0e-6; // 1 micro-coulomb
    q_prime = q/(2.0*pi*r); // charge density
    epsilon = 8.854e-12; // permittivity of free space

// Create a wireframe model for the current source,
// and indicate the sensor position

    OutputWireFrame = fopen("Path.xyz","w");

    // Draw a vector from zero to the Sensor
    fprintf(OutputWireFrame,"0 0 0 0 \n");    // moveto
    fprintf(OutputWireFrame,"%f %f %f 90 \n",Sensor.x,
                        Sensor.y,Sensor.z);
// lineto
```

```
   // Draw a vector from zero to the Sensor height
   fprintf(OutputWireFrame,"0 0 0 0 \n");    // moveto
   fprintf(OutputWireFrame,"0 0 %f 120 \n",Sensor.z);
                     // lineto

   // Draw a vector from Sensor.z to the Sensor
   fprintf(OutputWireFrame,"0 0 %f 0 \n",Sensor.z);
                     // moveto
   fprintf(OutputWireFrame,"%f %f %f 150 \n",Sensor.x,
                        Sensor.y,Sensor.z);
   // lineto

   // Draw a vector from zero to a SourcePosition
   fprintf(OutputWireFrame,"0 0 0 0 \n");    // moveto
   fprintf(OutputWireFrame,"1 0 0 180 \n");    // lineto

   // Draw a vector from SourcePosition to the Sensor
   fprintf(OutputWireFrame,"%f %f %f 90 \n",Sensor.x,
                        Sensor.y,Sensor.z);
// lineto


   // make a source current clockwise in a unit circle
   // in the XY plane

   for (i=0;i<360;i++) {
            //draw a wire frame for geometry sanity check
       theta = (i+0.5)*pi/180.0;
       SourcePosition[i].x = r*cos(theta);
       SourcePosition[i].y = r*sin(theta);
       SourcePosition[i].z = 0.0;

       phi = i*pi/180.0;
       x = r*cos(phi);
       y = r*sin(phi);
       z = 0.0;
       fprintf(OutputWireFrame,"%f %f %f 0 \n",x,y,z);
             // moveto
```

```
        phi = (i+1.0)*pi/180.0;
        x = r*cos(phi);
        y = r*sin(phi);
        z = 0.0;
        fprintf(OutputWireFrame,"%f %f %f 30 \n",x,y,z);
            // lineto

    }

    fclose(OutputWireFrame);

// First calculate V, E_x, E_y and E_z by piecewise integration for reference

    V = 0.0;
    E_x = E_y = E_z = 0.0;
    dtheta = 2.0*pi/360.00;
    dq = q/360.0; // charge per segment

//  Formula for dE
//
// dE = (1/(4 pi epsilon)) (dq/R^3) \vec deltaR
//

    for (i=0;i<360;i++) {
        deltaR = SubVector(Sensor,SourcePosition[i]);
          // sensor - source
        R2 = Dot(deltaR,deltaR);
        R = sqrt(R2); // distance between source and sensor
dV = (1.0/(4.0*pi*epsilon))*(dq/R);
V += dV;

scale = (1.0/(4.0*pi*epsilon))*(dq/(R*R*R));
E_x += scale*deltaR.x;
E_y += scale*deltaR.y;
E_z += scale*deltaR.z;

    }
```

```
// print V for comparison to integral calculation
    printf("V calculated by summing 360 one degree segments (Cartesian format)\n")
    printf("\nV = %f \n\n", V);

//    Now we calculate V using a charged loop and Elliptic Integrals

/*
        Charged loop in XY plane. Charge q, radius r

    k = sqrt[(2.0*Sensor.x*r)/(Sensor.x*Sensor.x + Sensor.z*Sensor.z + r*r)]

    V = (q/(8.0*pi*pi*epsilon))*(1.0/sqrt(Sensor.x*Sensor.x + Sensor.z*Sensor.z +
         (4.0/sqrt(1.0*k*k))*K(sqrt((2.0*k*k)/(1.0 + k*k)));

*/

    k = sqrt((2.0*Sensor.x*r)/(Sensor.x*Sensor.x + Sensor.z*Sensor.z + r*r));

    K_arg = sqrt((2.0*k*k)/(1.0 + k*k));

    Complete_Elliptic_K_and_E('k',K_arg, &K, &E);

    V_elliptic = (q/(8.0*pi*pi*epsilon))*(1.0/sqrt(Sensor.x*Sensor.x + Sensor.z*Se
         (4.0/sqrt(1.0+k*k))*K;


    printf("(Numerical Summation Method) V = %g \n", V);
    printf("(Elliptic Integral Method)   V_elliptic = %g  \n",V_elliptic);
    printf("\n");

// now we calculate V, EE_rho and EE_z by elliptic integrals

printf("\n\nCalculate V, E_rho and E_z via elliptic integrals using k");

rho = sqrt(Sensor.x*Sensor.x + Sensor.y*Sensor.y);
k = sqrt((4.0*rho*r)/( (r + rho)*(r + rho) + Sensor.z*Sensor.z));
        Complete_Elliptic_K_and_E('k',k, &K, &E);
```

```
V_elliptic = (q/(2.0*pi*pi*epsilon))*(1.0/sqrt((r+rho)*(r+rho) + Sensor.z*Sensor.z
    printf("\n(Elliptic Integral Method, second formula)   V_elliptic = %g  \n",V_
    printf("\n");

rprhoz = sqrt((r + rho)*(r + rho) + Sensor.z*Sensor.z);
rmrhoz = sqrt((r - rho)*(r - rho) + Sensor.z*Sensor.z);
EE_rho = (q/(2.0*pi*rho))*((1.0)/(2.0*pi*epsilon))*((1.0)/(rprhoz))*
         (K - E*( (Sensor.z*Sensor.z + r*r - rho*rho)/( (r-rho)*(r-rho) + Sensor.z

EE_z = ((q)/(2.0*pi*pi*epsilon))*((Sensor.z)/(rprhoz*rprhoz*rprhoz))*
            (E*((rprhoz*rprhoz)/(rmrhoz*rmrhoz)) );

printf("\nE_z = %g   EE_z = %g\n",E_z, EE_z);

printf("\nE_x = %g    E_y = %g    EE_rho = %g \n", E_x, E_y, EE_rho);

// now we use the cleaner formulas

z = Sensor.z;
R = r;
E2_rho = ((q)/(4.0*pi*epsilon*rho*rprhoz))*((K/pi) - (E/pi)*((z*z + R*R - rho*rho)
E2_z   = ((q)/(4.0*pi*epsilon*rho*rprhoz))*((E*2.0*rho*z)/(pi*rmrhoz*rmrhoz));
printf("E2_rho = %g   E2_z = %g  \n", E2_rho, E2_z);

// ************** create four plots - V along r, V along z, E_r, E_z ************

FILE* V_r_File;
FILE* V_z_File;
FILE* E_r_File;
FILE* E_z_File;

V_r_File = fopen("V(r)","w");
V_z_File = fopen("V(z)","w");
E_r_File = fopen("E_r(r)","w");
E_z_File = fopen("E_z(z)","w");

//  Define the charged ring
```

```
    r = 1.0; // 1 meter radius
    q = 1.0e-6; // 1 micro-coulomb
    q_prime = q/(2.0*pi*r); // charge density
    epsilon = 8.854e-12; // permittivity of free space


for (i=0;i<100;i++) { // do 100 points along axis
Sensor.x = 2.5*((i+0.0)/100.0);
Sensor.y = 0.0;
Sensor.z = 0.0;

rho = sqrt(Sensor.x*Sensor.x + Sensor.y*Sensor.y);
k = sqrt((4.0*rho*r)/( (r + rho)*(r + rho) + Sensor.z*Sensor.z));
Complete_Elliptic_K_and_E('k',k, &K, &E);

V_elliptic = (q/(2.0*pi*pi*epsilon))*(1.0/sqrt((r+rho)*(r+rho) + Sensor.z*Sensor.z

rprhoz = sqrt((r + rho)*(r + rho) + Sensor.z*Sensor.z);
rmrhoz = sqrt((r - rho)*(r - rho) + Sensor.z*Sensor.z);
EE_rho = (q/(2.0*pi*rho))*((1.0)/(2.0*pi*epsilon))*((1.0)/(rprhoz))*
 (K - E*( (Sensor.z*Sensor.z + r*r - rho*rho)/( (r-rho)*(r-rho) + Sensor.z*Sensor.

EE_z = ((q)/(2.0*pi*pi*epsilon))*((Sensor.z)/(rprhoz*rprhoz*rprhoz))*
(E*((rprhoz*rprhoz)/(rmrhoz*rmrhoz)) );

if (i != 40) {
fprintf(V_r_File,"%g  %g  \n", Sensor.x, V_elliptic);
fprintf(E_r_File,"%g  %g  \n", Sensor.x, EE_rho);
}
else {
fprintf(V_r_File," \n"); // skip infinity at rho = 1.0
fprintf(E_r_File," \n");
}

// now do a march along the z axis

Sensor.x = 0.0;
Sensor.y = 0.0;
```

```
Sensor.z = 2.5*((i+0.0)/100.0);

rho = sqrt(Sensor.x*Sensor.x + Sensor.y*Sensor.y);
k = sqrt((4.0*rho*r)/( (r + rho)*(r + rho) + Sensor.z*Sensor.z));
Complete_Elliptic_K_and_E('k',k, &K, &E);

V_elliptic = (q/(2.0*pi*pi*epsilon))*(1.0/sqrt((r+rho)*(r+rho) + Sensor.z*Sensor.z

rprhoz = sqrt((r + rho)*(r + rho) + Sensor.z*Sensor.z);
rmrhoz = sqrt((r - rho)*(r - rho) + Sensor.z*Sensor.z);
EE_rho = (q/(2.0*pi*rho))*((1.0)/(2.0*pi*epsilon))*((1.0)/(rprhoz))*
 (K - E*( (Sensor.z*Sensor.z + r*r - rho*rho)/( (r-rho)*(r-rho) + Sensor.z*Sensor.

EE_z = ((q)/(2.0*pi*pi*epsilon))*((Sensor.z)/(rprhoz*rprhoz*rprhoz))*
(E*((rprhoz*rprhoz)/(rmrhoz*rmrhoz)) );

fprintf(V_z_File,"%g  %g  \n", Sensor.z, V_elliptic);
fprintf(E_z_File,"%g  %g  \n", Sensor.z, EE_z);




}




fclose(V_r_File);
fclose(V_z_File);
fclose(E_r_File);
fclose(E_z_File);

}

/************* Results ***************************
V calculated by summing 360 one degree segments (Cartesian format)

V = 862.357542
```

```
(Numerical Summation Method) V = 862.358
(Elliptic Integral Method)   V_elliptic = 862.358



Calculate V, E_rho and E_z via elliptic integrals using k
(Elliptic Integral Method, second formula)   V_elliptic = 862.358


E_z = 24.12   EE_z = 24.12

E_x = 79.2979    E_y = -9.45424e-17    EE_rho = 79.2979

*/
```

References

[1] J. C. Maxwell, *Electricity and Magnetism, Vol II, Article 701*

[2] Edward B. Rosa and Louis Cohen, "Formulae and Tables for the Calculation of Mutual and Self-Inductance", Bulletin of the Bureau of Standards, Vol 5, No. 1.

[3] Alexander Russell, "The Magnetic Field and Inductance Coefficients of Circular, Cylindrical and Helical Currents", Procedings of the Royal Academy of Science.

[4] William H. Beyer, *CRC Standard Math Tables, 27th Edition*, CRC Press, Boca Raton Florida, 1984.

[5] I. Gradshteyn and I. Ryzhik,*Table of Integrals, Series and Products*, Academic Press, New York, 1980

[6] Milton Abramowitz and Irene A. Stegun, *Handbook of Mathematical Functions with Formulas. Graphs and Mathematical Tables*, National Bureau of Standards, Washington D. C. 1964.

[7] Frederick W. Grover, *Inductance Calculations Working Formulas and Tables*, Instrument Society of America, Research Triangle Park, N. C., 1973

[8] George Arfken, *Mathematical Methods for Physicists*, Third Edition, Academic Press, Orlando Florida, 1985 pp 321-326.

[9] Arthur Erdelyi, *Higher Transcendal Functions* , Krieger Publishing, Malabar Florida, 1985 Ch. 13 in Vol II, especially.