

# Geometric Mean Distance

---

## Its Derivation and Application in Inductance Calculations

Robert Weaver, Saskatoon, Canada

2016-03-05

(Revised 2016-03-06<sup>1</sup>)

### Abstract

This article begins with an informal introduction to the use of Geometric Mean Distance (GMD) in inductance calculations, and why it is important. This is followed by brief discussion of the logarithm function. Because it pervades the subject of GMD calculation, it is important to have a good working knowledge of various logarithmic identities and integrals of formulae involving logarithms. The issue of integrating across logarithmic anomalies (i.e.,  $\log(0)$ ) is also treated.

The self GMD is derived in complete detail for some very simple shapes, in order to demonstrate the general analytical method of calculation. This is followed by a discussion of the use of numerical methods—The Monte Carlo Method, in particular—for the GMD calculation of more difficult shapes. Detailed examples are given, which result in empirical formulae for the self GMD of a triangular area, the mutual GMD of elliptical areas and the self GMD of elliptical loci.

The use of GMD in inductance calculations is often accompanied by the proviso, that the "*diameter of the conductor should be considerably smaller than the diameter of the winding.*" However, there is almost nothing available in the literature giving quantitative information about the limits of accuracy of the GMD method. This is treated for the cases of self inductance of a circular loop, and the mutual inductance of two parallel circular loops. Empirical functions are developed, which give the estimated error due to the approximation of the GMD method. These error formulae may be used to calculate correction factors. An example of this is given for a multi-turn solenoid coil.

The final part of the article deals with the calculation of the self GMD of groups of objects. In particular, closed form formulae are developed for the self GMD of linear arrays of circular conductors and linear arrays of thin strips of conductors. These are further developed into a new closed form formula for Rosa's round wire inductance corrections, and formulae for the inductance of short coils which account for conductor shape.

---

<sup>1</sup> For revision history, see the last page of this document. The latest version of this document may be downloaded, free of charge, from the author's website: <http://electronbunker.ca/DLpublic/GMD.pdf>

## Table of Contents

<b>Abstract</b>	<b>1</b>
<b>Part 1 - Basic Concepts</b>	<b>3</b>
1.1 Introduction.....	3
1.2 Conventions.....	3
1.3 Inductance Calculations using GMD.....	4
1.4 Why Geometric Mean and not Arithmetic Mean?.....	5
1.5 The Logarithm Function - A Primer.....	7
<b>Part 2 - Analytical Calculation of Simple Shapes</b>	<b>12</b>
2.1 GMD of two co-linear lines of equal length.....	12
2.2 GMD of a Line from Itself (Self GMD of a line).....	18
<b>Part 3 - Calculation of GMD using numerical methods</b>	<b>23</b>
3.1 The Monte Carlo Method.....	23
3.2 The GMD of a Triangle.....	24
3.3 The GMD of an Elliptical Area.....	27
3.4 The Self GMD of an Elliptical Line.....	37
<b>Part 4 - Limits of Accuracy of the GMD method</b>	<b>44</b>
4.1 Self Inductance.....	44
4.2 Mutual Inductance.....	49
4.3 Example Calculation.....	60
4.4 Summary.....	61
<b>Part 5 - Combined GMD of Multiple Objects</b>	<b>62</b>
5.1 Basic Principles.....	62
5.2 Formula for the GMD of a Linear Array of Circular Conductors (LACC).....	63
5.3 Formula for the GMD of a Linear Array of Co-linear Straight Line Segments (LALS).....	71
5.4 Inductance Calculation Based on Aggregate GMD.....	75
5.5 Rosa's Round Wire Inductance Corrections.....	77
5.6 Flat Conductor Corrections.....	87
<b>Afterword</b>	<b>89</b>
<b>References</b>	<b>90</b>
<b>Appendices</b>	<b>92</b>
A – Program Listing for calculation of the GMD of an Elliptical Locus.....	92
B – A Method for Finding and Eliminating Errors in Mathematical Derivations.....	96
<b>Revision History</b>	<b>99</b>

## Part 1 - Basic Concepts

### 1.1 Introduction

It is impossible to study the techniques of inductance calculation to any great extent without encountering the term *geometric mean distance*. Often a simple but inadequate explanation is given, resulting in more questions than answers. For that reason, this article has been prepared to address some of the basics and some of the common questions that arise. This is largely a non-rigorous treatment. However, in different sections, the degree of rigour will vary, according to what information could be readily located or calculated with reasonable effort. For some sections, the reader will require some knowledge of basic calculus.

### 1.2 Conventions

In this article, SI units will be used. Where non-SI formulae are quoted from various references they will be adjusted to SI. Hence inductance will be expressed in henries, and length, diameter and radius in metres.

Logarithms will be encountered frequently, and in all cases will be natural logarithms. The logarithm function will be indicated as *log()*. In other literature, *Ln()* is often used, but it means the same thing.

The following symbols will be used:

$\epsilon$	error
$\mu_0$	Permeability of free space, or vacuum, equal to $4\pi \times 10^{-7}$ henries / metre
$\mu_R$	Relative permeability of the medium under consideration
$\mu$	Absolute permeability of the medium under consideration ( $\mu = \mu_0 \times \mu_R$ )
$\gamma$	self GMD shape characteristic factor
$a, b, c...$	principal dimensions (length, width etc.) of a shape under study
$D$	Diameter of coil or loop of wire
$d$	Diameter of conductor
$g$	geometric mean distance (GMD) of one object from another
$g_s$	geometric mean distance of an object from itself (self GMD) <sup>2</sup>
$g_M$	mutual GMD, geometric mean distance between objects
$k$	Miscellaneous numeric constant, often distinguished by a numeric subscript
$\ell$	Length (of conductor or coil depending on subscript)
$\ell_W$	Conductor (wire) length

---

<sup>2</sup> In the technical literature, the term *geometric mean radius* or *GMR* is often used for the geometric mean distance of an object from itself. It is an inaccurate term, and hence, *self GMD* will be used in this discussion.

$\ell_C$	Coil length
$n$	Number of turns in coil, or number of conductors in array
$R$	Radius of coil or loop of wire
$r$	Radius of conductor (the subscript is $w$ for <i>wire</i> , since the subscript $c$ has already been used for <i>coil</i> )
$s$	length of line or line segment
$u$	Shape factor, aspect ratio, or miscellaneous dimensionless ratio
$u_c$	Coil shape factor $\ell_C/R$
$x$	distance separating two conductors or filaments
$x_{ij}$	distance separating conductor or filament $i$ from conductor or filament $j$

A word of warning should be given about the use of symbols in this document. Because many of the derivations will be developed in great detail, a consequence of this is that many numeric constants will appear during the derivation, and then later disappear as formulae become simplified. For example, a formula that employs constants  $k_0$ ,  $k_1$ , and  $k_2$  may end up having  $k_0$  and  $k_1$  drop out, leaving an orphan  $k_2$ . Leaving such a constant named  $k_2$  in the final result is guaranteed to invite confusion, especially if the formula is later presented out of context. For that reason, once a formula reaches its final form, some symbol names may change. Every attempt will be made to clearly point this out whenever it happens. In any event, the reader is hereby forewarned.

Similarly, the symbol  $u$  is used for ratios or shape factors such as length/diameter, length/width. Depending in which section it is used it may have different meanings. However, its definition will be given when it is introduced in a discussion.

### 1.3 Inductance Calculations using GMD

A formula for the **mutual** inductance between a pair parallel of identical infinitesimally thin filaments can be turned into a formula for **self** inductance of a finite conductor by one of two methods:

1. **Difficult:** Integrate the filamentary mutual inductance formula across the cross sectional area of the conductor (a double integral involving area, which devolves into a quadruple integral involving length);
2. **Moderately Easy:** Determine the self-geometric mean distance (self GMD) of the cross section of the conductor, and then use that value to do a single mutual inductance calculation.

In the first case we are treating the conductor as an infinite number of parallel infinitesimal filaments, and then calculating the mutual inductance of each filament paired with each other filament, multiplying it by a weighting factor  $dA$  which is the infinitesimal cross sectional area of the filament. These mutual inductance calculations are summed and then divided by the total cross sectional area  $A$  of the conductor to yield the self inductance.

In the second case, we are doing essentially the same thing, with one important difference: In each mutual inductance calculation, rather than using the actual distance,  $x_{ij}$ , (between the filaments  $i$  and  $j$ ), which varies with each filament pair, instead we use a *mean* of the distances separating the filament pairs. Using this single value allows for a massive simplification of the calculation—the integration becomes trivial—with virtually no loss of accuracy for most practical applications.

The GMD principle is strictly correct for an infinitely long straight conductor, but is approximately correct for a straight conductor of finite length, as long as the end effects are relatively small (i.e., the conductor diameter is significantly smaller than the conductor length). The GMD principle can also be applied to conductor geometries other than straight lines as long as the conductor diameter is small relative to the overall conductor length, and the radius of curvature is large compared to the cross sectional diameter of the conductor. This is because the GMD is really a two dimensional attribute related to the cross sectional geometry of the conductor (in, say, the X–Y plane), and assumes that what goes on in the third dimension (Z–axis) is constant (ideally), or changes very gradually and smoothly relative to what happens in the other dimensions. In this way the effect of what happens in the third dimension has minimal effect on the field lines in the other dimensions. For the remainder of this discussion we will assume that conductor size is small relative to these other parameters. An exception will be where these conditions are deliberately violated in order to find the limits of accuracy of the GMD method.

#### 1.4 Why Geometric Mean and not Arithmetic Mean?

Mutual inductance of two filaments is determined from the Neumann integral or one of several other equivalent formulae. The Neumann integral is:

$$M = \frac{\mu_0}{4\pi} \int \frac{ds_1 \cdot ds_2}{x} \tag{1.4.1}$$

where  $x$  is the distance separating infinitesimal segments of filaments  $ds_1$  and  $ds_2$ . When the integral is evaluated, the principle term will be  $\log(x)$ , because the integral  $\int \frac{1}{x} dx$  is equal to  $\log(x)$ . Depending on the geometric details of the filaments, there will also be other terms, but generally having a smaller contribution. To take this result and then calculate the self inductance of a conductor of finite cross section, it is necessary to integrate this log term across the cross sectional area of the conductor. Since integration is a linear operation, it is permissible to simplify the calculation by taking the arithmetic mean of the integrand and multiplying by the limits of the integral. But since the arithmetic mean of  $\log(x)$  is the same as the antilog of the geometric mean of  $x$ , we are essentially taking the geometric mean of  $x$ . (There are some cases where one or two of the secondary non logarithmic terms are significant enough that an arithmetic mean or mean square should be applied to those terms rather than a geometric mean.)

For a straight conductor, we can take the formula for the mutual inductance between two straight filaments and using the self GMD of the conductor cross section, combine the two into a formula

for the self inductance of a straight conductor. The formula for the mutual inductance of two parallel straight filaments [2] (page 31), adjusted to SI units, is:

$$M = \frac{\mu_0 \ell}{2\pi} \left[ \log \left( \frac{\ell}{x} + \sqrt{1 + \left( \frac{\ell}{x} \right)^2} \right) - \sqrt{1 + \left( \frac{x}{\ell} \right)^2} + \frac{x}{\ell} \right] \quad (1.4.2)$$

where  $\ell$  is the length of the filaments, and  $x$  is their separation distance.

Assuming the conductor has a round cross section, the self GMD is given by Maxwell [1] (Art. 691) to be simply:

$$e^{-1/4} \times r \text{ or } 0.77880 \times r \quad (1.4.3)$$

where  $r$  is the radius of the conductor.

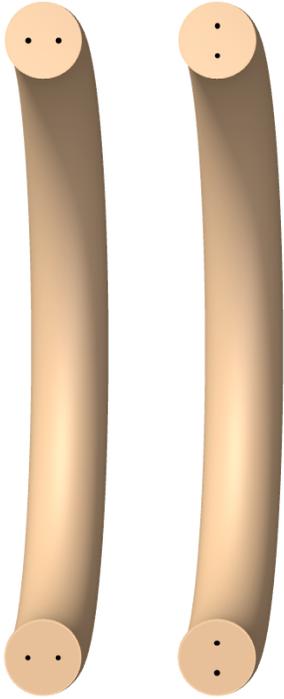
We replace the finite round conductor with two fictitious infinitesimal filaments separated by the conductor's self GMD, and use that self GMD value, as the separation distance, in the mutual inductance formula giving:

$$L = \frac{\mu_0 \ell}{2\pi} \left[ \log \left( \frac{\ell}{0.7788r} + \sqrt{1 + \left( \frac{\ell}{0.7788r} \right)^2} \right) - \sqrt{1 + \left( \frac{0.7788r}{\ell} \right)^2} + \frac{0.7788r}{\ell} \right] \quad (1.4.4)$$

In the case of a circular loop of round wire, we can use the mutual inductance formula for two circular filaments, in combination with the GMD of round wire to quickly come up with a formula for the self inductance of the loop. However, there's a bit of a catch. So far, nothing has been said about the direction of offset of the two fictitious filaments. In the case of the straight conductor it makes no difference. If the conductor cross section is in the X-Y plane, and the length of the conductor is in the Z dimension, then the two filaments can be separated in either the X or Y dimension or a combination of both, because the conductor is radially symmetric. However, in the case of a circular loop we have the choice of using two filaments which are offset axially or radially (see diagram below).

The axial offset would seem to make the most sense, as there is no question about what the radius of the filaments should be (i.e., the same as the radius to the centre of the conductor). Indeed, that is generally the approach that is taken.

In the loop cross section shown in the leftmost figure, the two black dots indicate the spacial relationship of the proxy filaments in true scale relative to the conductor cross section for which they are substituted. It turns out that we can also use radially offset filaments (rightmost figure) as long as the mean radius of the two filaments is the same as radius of the centre of the conductor. But what kind of mean? Arithmetic or Geometric? Not surprisingly, in the context of inductance calculations, it turns out that it is the geometric mean. Two radially offset circular filaments whose



geometric mean radius is the same as the radius of the centre of the conductor will give exactly the same inductance value as two axially offset filaments. (We are assuming conductors of round cross section.)

If the radius of the loop of conductor is  $R$  (measured from axis to centre of conductor), and the radii of the two proxy filaments are  $R_A$  (inner) and  $R_B$  (outer), then we are saying:

$$R = \sqrt{R_A R_B} \quad (1.4.5)$$

And since the filaments are separated by a distance equal to  $g_S$ , then:

$$g_S = R_B - R_A \quad (1.4.6)$$

Combining the two formulae, leads to a quadratic equation:

$$R_A^2 + g_S R_A - R^2 = 0 \quad (1.4.7)$$

With the solution:

$$R_A = (-g_S + \sqrt{g_S^2 + 4 R^2}) / 2 \quad (1.4.8)$$

$$R_B = R_A + g_S \quad (1.4.9)$$

In the above right diagram, the difference between arithmetic mean positioning and geometric mean positioning is too small to be seen, and in most cases the numerical difference is also nearly insignificant, but for the most accurate calculations, the geometric mean is the correct one. The above illustrations are scale drawings of loops with a radius of 8 (any units) and conductor radius of 1. For a geometric mean positioning,  $R_A$  and  $R_B$  are 7.62007 and 8.39887 respectively, while for an arithmetic mean positioning,  $R_A$  and  $R_B$  would be 7.61060 and 8.38940 respectively. The difference would be more significant if the ratio of conductor radius to loop radius were larger. However, with these larger ratios, the accuracy of the overall GMD approximation would deteriorate.

*As an important consequence from the above, it can be seen that there is an equivalence between axially offset circular filaments and radially offset circular filaments (or for that matter filaments that are offset both radially and axially). This means that it is possible to convert a pair of radially offset filaments into an equivalent pair of axially offset filaments having the same mutual inductance. This will prove useful in developing certain inductance formulae.*

### 1.5 The Logarithm Function - A Primer

Since the logarithm function figures prominently in GMD calculations, it is worthwhile to do a quick review of some of its properties. In calculus, the logarithm function,  $\log(x)$ , is defined as a definite integral with a variable upper limit:

$$\log x = \int_1^x \frac{dt}{t} \quad [x > 0] \quad (1.5.1)$$

Alert readers will note that when calculating geometric mean distances, the argument of the logarithm function is a distance, and not surprisingly, has the units of distance. It's often stated that the argument of a logarithm function must be dimensionless, and in many cases it is the ratio of two values which have the same dimensions so that they do cancel out and make for a dimensionless argument. In the case of GMD calculations we clearly see arguments which have dimensions of length. This leads to the question: What are the units of the result of the logarithm function?

From the above integral formula, it can be seen that if the variable  $x$  has dimensional units such as length, then both  $t$  and  $dt$  must take on the same units. Since  $dt$  is in the numerator and  $t$  is in the denominator, the units cancel out, and the integrand is therefore dimensionless. Because integration is a linear operation, the units of the integrand, if any, become the units of the overall integral. Therefore, the argument of the logarithm function may have dimensional units, but from a dimensional analysis point of view, the resulting value of the logarithm function is dimensionless.

Of course, this leads to the question of what happens when we take the anti-log of a dimensionless value. This seems to imply that a random dimension can magically appear as a result. However, common sense tells us that if we take the logarithm of an argument having some particular dimension, and then subsequently take the anti-log, as part of the same derivation, the result should have the same dimension as the original argument. Dimensional analysis does have its limits.

The integral definition of the log function has no analytical solution, which is why we simply refer to it as  $\log(x)$  and leave the dirty work to a calculator or a spreadsheet program. However, it may be evaluated numerically using one of several series expansions [3] such as:

$$\log x = \int_1^x \frac{dt}{t} = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \dots \quad (1.5.2)$$

Several different series may be found in the literature. Different expansions will provide different rates of convergence, depending on the value of argument.

Some fundamental logarithm function identities follow:

$$\log(ab) = \log(a) + \log(b) \quad (1.5.3)$$

$$\log(a/b) = \log(a) - \log(b) \quad (1.5.4)$$

$$\log(a^n) = n \log(a) \quad (1.5.5)$$

The calculation of GMD involves integrating the logarithm function over various ranges. Hence, it is important to remember that  $\log(r)$  is undefined for  $r \leq 0$ . Since the calculation is only concerned

with the magnitude of distances, we can write  $\log(|r|)$ . Often, the absolute value is obtained automatically such as in the case where  $r$  is calculated from cartesian coordinates as

$$r = \sqrt{x^2 + y^2}$$

In cases of integration, it may be necessary to split the integral into two parts:

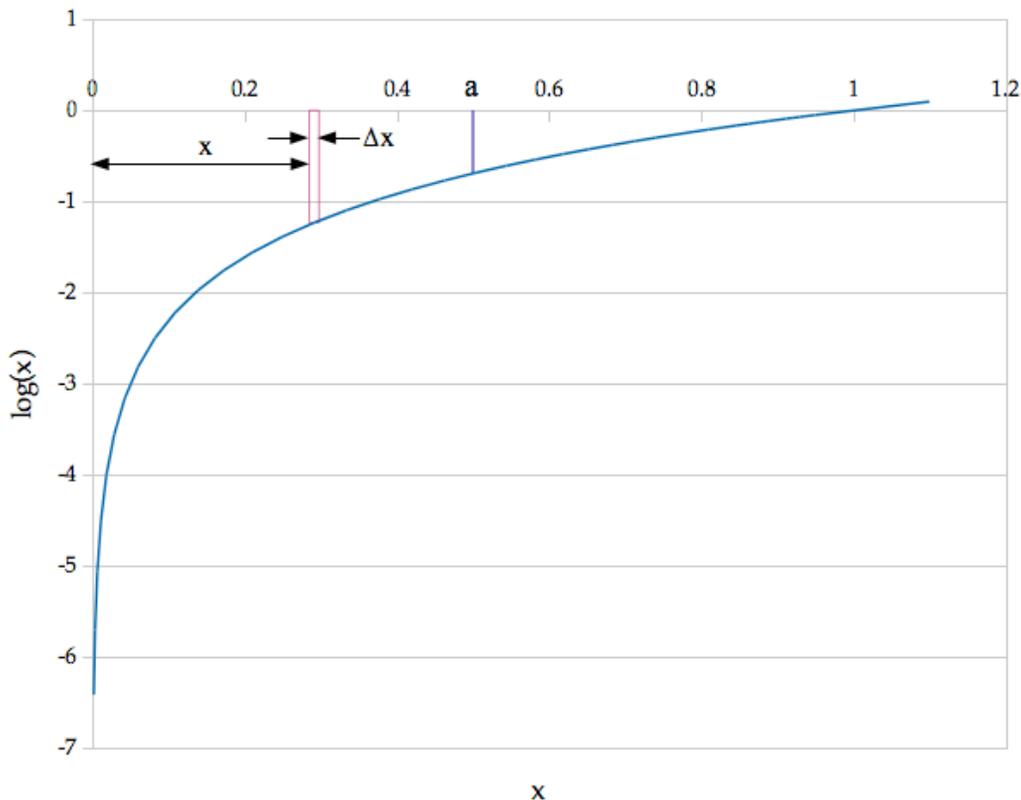
$$\int \log(r) dr + \int \log(-r) dr$$

in order to cover the range of  $r < 0$  separately from the range of  $r > 0$ .

While this resolves how to deal with negative values, it still leaves the situation of calculating a self GMD which will always involve integration across an area where  $r$  will, at some point, have a value of zero. How is this singularity resolved? To understand what happens in this situation, it is useful to look at a simpler but equivalent situation, and that is the following integral:

$$\int_0^a \log(x) dx$$

In graphical terms (see diagram below), we are taking the area bounded by the  $x$  and  $y$  axes, the function  $\log(x)$ , and the vertical line  $x = a$ .

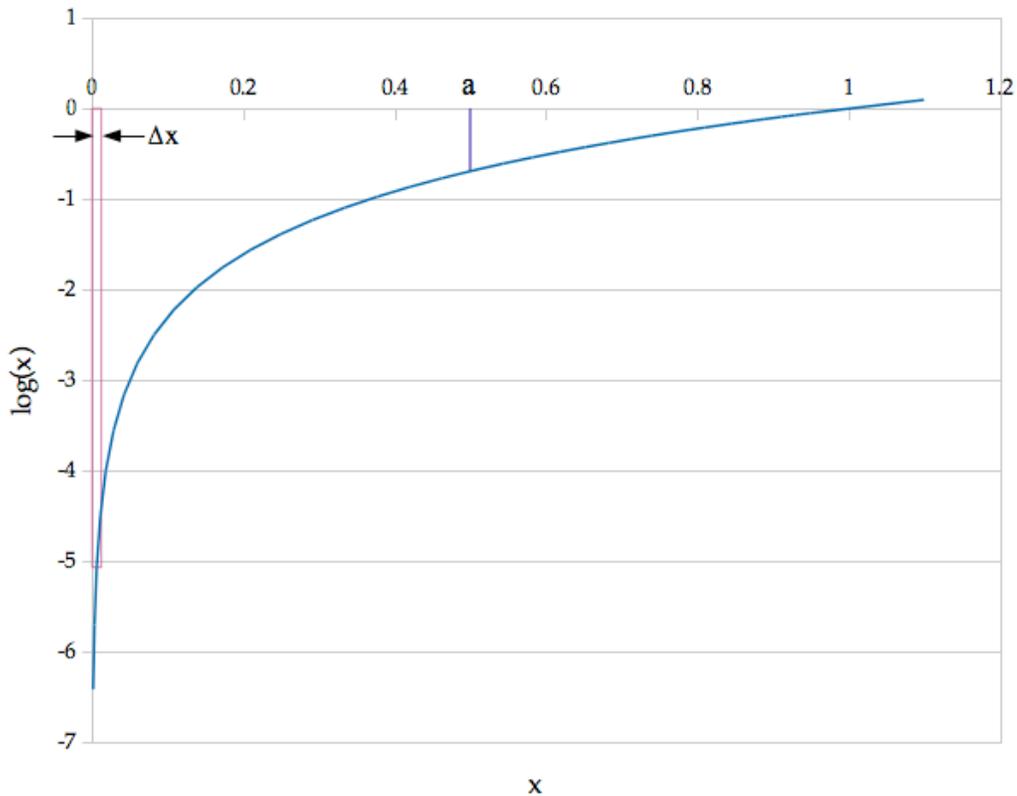


Remembering that an integral can be defined as a sum of areas  $f(x)\Delta x$  as  $\Delta x$  approaches zero, it is equivalent to:

$$\lim_{\Delta x \rightarrow 0} \sum_{x=0}^a \log(x) \Delta x$$

where we are taking the area from  $x = 0$  to  $x = a$ , and the areas are long skinny slices with height equal to the value of the function and width equal to  $\Delta x$ . The value  $\Delta x$  is made infinitely small.

The red rectangle denotes the area  $\log(x+\frac{1}{2}\Delta x) \times \Delta x$  and is summed as  $x$  varies from 0 to  $a-\Delta x$  in increments of  $\Delta x$ . And of course  $\Delta x$  is made infinitesimal. As  $x$  approaches zero, it can be seen that  $\log(x)$  tends towards minus infinity. The area we are concerned with is where  $x = 0$ , and thus the value of  $\log(\frac{1}{2}\Delta x) \times \Delta x$  as  $\Delta x$  becomes infinitesimal.



As  $\Delta x$  approaches zero the incremental area approaches  $0 \times -\infty$ , leaving us to wonder which factor wins out: the zero or the minus infinity. To resolve this, we can apply L'Hôpital's rule which states that in the case of a limit involving a quotient of two functions that approaches  $0/0$  or  $\infty/\infty$ , then the limit is equal to the quotient of the derivatives of the two functions. In this case it is possible to rewrite  $\log(\frac{1}{2}\Delta x) \times \Delta x$  as  $\log(\frac{1}{2}\Delta x) / (1/\Delta x)$  which yields the  $\infty/\infty$  form, and then:

$$\lim_{x \rightarrow 0^+} x \log\left(\frac{x}{2}\right) = \frac{\frac{d}{dx} \log\left(\frac{x}{2}\right)}{\frac{d}{dx} x^{-1}} = \frac{\frac{1}{2x}}{-x^{-2}} = \frac{-x^2}{2x} = \frac{-x}{2} \quad (1.5.6)$$

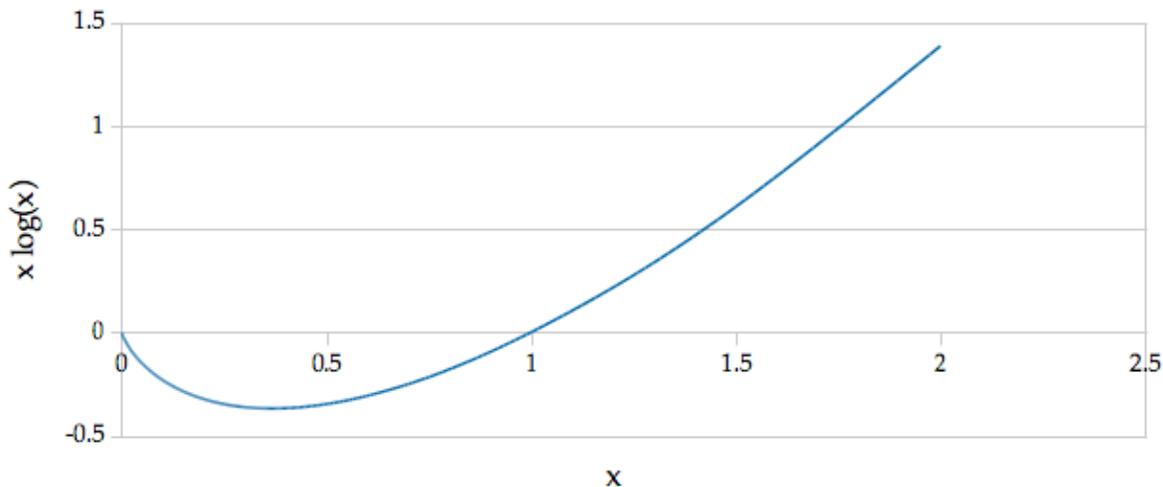
And so, it is clear that, as  $x$  goes to zero,  $-x/2$  also goes to zero. Therefore, there is no singularity in the integral due to  $\log(0)$ . This is perfectly logical in the context of GMD calculations, because it is

saying, essentially, that when two points have zero distance separating them, their mean distance is zero, not minus infinity. In fact using the same analysis, it can be shown that

$$\lim_{x \rightarrow 0^+} x^n \log(kx) = 0 \quad (1.5.7)$$

for any  $k > 0$  and  $n > 0$ , including fractional powers.

The zero limit can be illustrated by plotting the formula,  $x \log(x)$ :



In this graph, the smallest value of  $x$  that was calculated and plotted is  $10^{-8}$ , and it can readily be seen that as  $x$  gets closer to zero, the value of  $x \log(x)$  clearly tends to zero. Having resolved this issue, we can safely proceed to calculate GMD integrals with no further worries about  $\log(0)$  singularities.

Following are the solutions to several integrals [3] involving the logarithm function which will prove useful in the upcoming derivations:

$$\int x \log(x) dx = \frac{x^2}{2} \log(x) - \frac{x^2}{4} \quad (1.5.8)$$

$$\int \log(a + bx) dx = \frac{a + bx}{b} \log(a + bx) - x \quad (1.5.9)$$

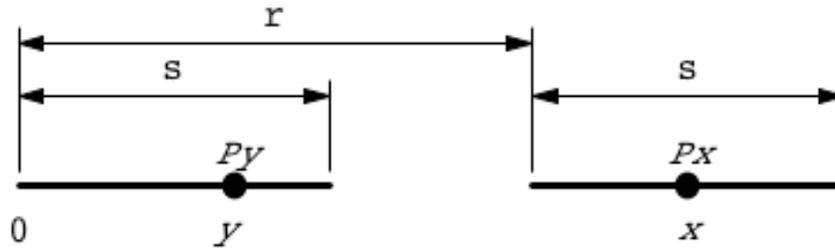
$$\int x \log(a + bx) dx = \frac{b^2 x^2 - a^2}{2b^2} \log(a + bx) + \frac{ax}{2b} - \frac{x^2}{4} \quad (1.5.10)$$

The next few sections will give several examples of the calculation of GMD for some simple shapes.

## Part 2 - Analytical Calculation of Simple Shapes

### 2.1 GMD of two co-linear lines of equal length

The following diagram shows two co-linear lines of length  $s$  with the first line's leftmost point at the origin (for convenience), and the second line offset by a distance  $r$ .



If  $P_y$  is a point on the first line located at a distance  $y$  from the origin, and  $P_x$  is a point on the second line located at a distance  $x$  from the origin, then the distance between these points is  $(x - y)$ . To calculate the GMD, we take the sum of the logarithms of the distance between every combination of points and divide by the length of the first line and then divide again by the length of the second line. The result is the logarithm of the GMD. Hence:

$$\log(g) = 1/s \times 1/s \times \sum \sum \log(x_i - y_j)$$

Because there is an infinite number of points on each line, the formula for the GMD between the two lines becomes a double integral:

$$\log(g) = \frac{1}{s^2} \int_r^{r+s} \int_0^s \log(x - y) dy dx \quad (2.1.1)$$

where  $x$  is a distance over the right line segment and  $y$  is a distance over the left line segment.

Evaluating the first integral with respect to  $y$ , using the integral formulae given in the previous section, we get:

$$\begin{aligned}
 \log(g) &= \frac{1}{s^2} \int_r^{r+s} \left[ (y-x) \log(x-y) - y \right]_0^s dx \\
 &= \frac{1}{s^2} \int_r^{r+s} \left[ (s-x) \log(x-s) - s \right] - \left[ -x \log(x) \right] dx \\
 &= \frac{1}{s^2} \int_r^{r+s} (s-x) \log(x-s) - s + x \log(x) dx
 \end{aligned} \tag{2.1.2}$$

Before doing the second integration, it's a good idea to split the terms into separate integrals rather than trying to deal with the entire integral, which would quickly become very messy. Therefore we now have:

$$\log(g) = \frac{1}{s^2} \left( \int_r^{r+s} A dx - \int_r^{r+s} B dx + \int_r^{r+s} C dx \right) \tag{2.1.3}$$

where:

$$\begin{aligned}
 A &= (s-x) \log(x-s) = s \log(x-s) - x \log(x-s) \\
 B &= s \\
 C &= x \log(x)
 \end{aligned}$$

Additionally, we will split the terms of  $A$  into sub-terms  $A_1$  and  $A_2$ :

$$\begin{aligned}
 A_1 &= s \log(x-s) \\
 A_2 &= x \log(x-s)
 \end{aligned}$$

Evaluating these integrals one by one:

$$\begin{aligned}
\int_r^{r+s} A_1 dx &= \int_r^{r+s} s \log(x-s) dx \\
&= \left[ s(x-s) \log(x-s) - sx \right]_r^{r+s} \\
&= \left[ s(r+s-s) \log(r+s-s) - s(r+s) \right] - \left[ s(r-s) \log(r-s) - sr \right] \\
&= sr \log(r) - sr - s^2 - (sr - s^2) \log(r-s) + sr \\
&= sr \log(r) - (sr - s^2) \log(r-s) - s^2
\end{aligned}$$

$$\begin{aligned}
\int_r^{r+s} A_2 dx &= \int_r^{r+s} x \log(x-s) dx \\
&= \left[ \frac{x^2 - s^2}{2} \log(x-s) - \frac{sx}{2} - \frac{x^2}{4} \right]_r^{r+s} \\
&= \left[ \frac{(r+s)^2 - s^2}{2} \log(r+s-s) - \frac{s(r+s)}{2} - \frac{(r+s)^2}{4} \right] \\
&\quad - \left[ \frac{r^2 - s^2}{2} \log(r-s) - \frac{sr}{2} - \frac{r^2}{4} \right] \\
&= \left( \frac{r^2}{2} + sr \right) \log(r) - sr - \frac{3s^2}{4} - \frac{r^2}{4} + \frac{(s^2 - r^2)}{2} \log(r-s) + \frac{sr}{2} + \frac{r^2}{4} \\
&= \left( \frac{r^2}{2} + sr \right) \log(r) - \frac{sr}{2} - \frac{3s^2}{4} + \frac{(s^2 - r^2)}{2} \log(r-s)
\end{aligned}$$

Combining the above  $A_1$  and  $A_2$  integrals:

$$\begin{aligned}
\int_r^{r+s} A dx &= \int_r^{r+s} A_1 - A_2 dx \\
&= sr \log(r) - (sr - s^2) \log(r - s) - s^2 \\
&\quad - \left[ \left( \frac{r^2}{2} + sr \right) \log(r) - \frac{sr}{2} - \frac{3s^2}{4} + \frac{(s^2 - r^2)}{2} \log(r - s) \right] \\
&= sr \log(r) - (sr - s^2) \log(r - s) - s^2 - \left( \frac{r^2}{2} + sr \right) \log(r) + \frac{sr}{2} + \frac{3s^2}{4} \\
&\quad - \frac{(s^2 - r^2)}{2} \log(r - s) \\
&= \frac{-r^2}{2} \log(r) - \left( sr - \frac{s^2 + r^2}{2} \right) \log(r - s) - \frac{s^2}{4} + \frac{sr}{2}
\end{aligned}$$

The  $B$  integral is mercifully simple:

$$\begin{aligned}
\int_r^{r+s} B dx &= \int_r^{r+s} s dx = [sx]_r^{r+s} \\
&= [s(r+s)] - [sr] = sr + s^2 - sr = s^2
\end{aligned}$$

And finally, the  $C$  integral:

$$\begin{aligned}
\int_r^{r+s} C dx &= \int_r^{r+s} x \log(x) dx \\
&= \left[ \frac{x^2}{2} \log(x) - \frac{x^2}{4} \right]_r^{r+s} \\
&= \left[ \frac{(r+s)^2}{2} \log(r+s) - \frac{(r+s)^2}{4} \right] - \left[ \frac{r^2}{2} \log(r) - \frac{r^2}{4} \right] \\
&= \left( sr + \frac{r^2 + s^2}{2} \right) \log(r+s) - \frac{r^2}{2} \log(r) - \frac{s^2}{4} - \frac{sr}{2}
\end{aligned}$$

At last, combining all the terms:

$$\begin{aligned}
\log(g) &= \frac{1}{s^2} \left[ \int_r^{r+s} A dx - \int_r^{r+s} B dx + \int_r^{r+s} C dx \right] \\
&= \frac{1}{s^2} \left[ \frac{-r^2}{2} \log(r) - \left( sr - \frac{s^2 + r^2}{2} \right) \log(r - s) - \frac{s^2}{4} + \frac{sr}{2} - s^2 \right. \\
&\quad \left. + \left( sr + \frac{r^2 + s^2}{2} \right) \log(r + s) - \frac{r^2}{2} \log(r) - \frac{s^2}{4} - \frac{sr}{2} \right] \\
&= \frac{1}{s^2} \left[ \left( sr + \frac{r^2 + s^2}{2} \right) \log(r + s) - \left( sr - \frac{r^2 + s^2}{2} \right) \log(r - s) - r^2 \log(r) - \frac{3s^2}{2} \right]
\end{aligned}$$

And so, we have derived the GMD formula for two equal co-linear lines:

$$\log(g) = \frac{1}{s^2} \left[ \left( sr + \frac{r^2 + s^2}{2} \right) \log(r + s) - \left( sr - \frac{r^2 + s^2}{2} \right) \log(r - s) - r^2 \log(r) - \frac{3s^2}{2} \right] \quad \dots(2.1.4)$$

This is equivalent to formula (130) given by Rosa and Grover [4](page 168).

We might be tempted to set  $r = 0$  in order to superimpose the two lines, and thus get the self GMD of a line. However, this fails, because the argument of the log function in the second term goes negative, resulting in an undefined value.

There is also a singularity in the case of  $s = r$ , i.e., the point where the end of the first line coincides with the start of the second line. In that case, the argument of the log function in the second term is equal to zero, and the log term goes to  $-\infty$ . However, the factor outside log function goes to zero, and it forces the term to zero (as with the discussion of limits in the previous section). Hence, the second term may be eliminated in the case where  $s = r$ , and for two equal co-linear lines just touching, we get:

$$\begin{aligned}
\log(g) &= \frac{1}{s^2} \left[ \left( sr + \frac{r^2 + s^2}{2} \right) \log(r + s) - r^2 \log(r) - \frac{3s^2}{2} \right] \\
&= \log(4s) - \frac{3}{2} \\
\therefore g &= 0.8925s \quad \dots(2.1.5)
\end{aligned}$$

One can also use the result of the first integration, above, to find the GMD from a line to a co-linear point.



It's simply a matter of substituting into the the result of the first integral, the distance  $a$  between the point and the line (and adjusting signs), which then gives:

$$\log (g) = \frac{1}{s} \left( (a + s) \log (a + s) - s - a \log (a) \right) \quad (2.1.6)$$

In the case of a point located exactly at the end of a line, i.e.,  $a = 0$ , then the last term disappears (as previously explained), and the right hand side of the formula simplifies to:

$$\log (g) = \log (s) - 1 \quad (2.1.7)$$

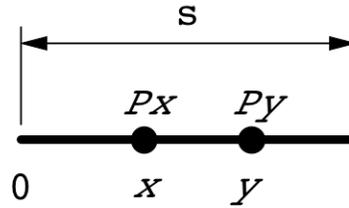
Hence, the GMD of a point at the end of a line to the line itself is:

$$g = se^{-1} = 0.36788s \quad (2.1.8)$$

Before moving on to the next example, something should be said about the practical difficulty in working through the above derivation. Even though this example has dealt with very simple shapes, and even though the integrals themselves have straightforward solutions, nevertheless, the number of terms that are generated during the derivation, make it a rather daunting task. The probability of committing errors during the evaluation and subsequent simplification, is enormous. (Software packages which perform symbolic mathematical operations are available. However, they can be costly, and therefore are unavailable to many people. There are also free symbolic math packages such as **Maxima**, but still, they can be rather unwieldy to use for various reasons.) The method which was used here, is described in **Appendix B**.

## 2.2 GMD of a Line from Itself (Self GMD of a line)

The two lines of length  $s$  from the previous example are now superimposed as shown in the following diagram. The line's leftmost point is again at the origin.



$P_x$  and  $P_y$  are two points both located on the line segment at distances  $x$  and  $y$ , respectively, from the origin. The distance between points  $P_x$  and  $P_y$  is  $|x-y|$ .

We will begin by calculating the GMD of point  $P_x$  to the entire line. Hence:

$$\log (g) = \frac{1}{s} \int_0^s \log |y - x| dy \quad (2.2.1)$$

The variable  $x$  is held constant, and the variable of integration is  $y$ . The requirement for the absolute value of the log function argument poses a minor problem. It is necessary to split the integral into two separate integrals: the first to handle the range  $0 \leq y \leq x$ , and the second to handle the range  $x \leq y \leq s$ :

$$\log (g) = \frac{1}{s} \int_0^x \log (x - y) dy + \frac{1}{s} \int_x^s \log (y - x) dy \quad (2.2.2)$$

This ensures that the argument of the log function will never be negative. These integrals can be solved using the integral formulae given previously.

$$\begin{aligned}
\log(G) &= \frac{1}{s} \left[ (y-x) \log(x-y) - y \right]_0^x + \frac{1}{s} \left[ (y-x) \log(y-x) - y \right]_x^s \\
&= \frac{1}{s} \left[ ((x-x) \log(x-x) - x) - (-x \log(x)) \right. \\
&\quad \left. + ((s-x) \log(s-x) - s) - ((x-x) \log(x-x) - x) \right] \\
&= \frac{1}{s} [(x \log(x) - x) + ((s-x) \log(s-x) - s + x)] \\
&= \frac{1}{s} [x \log(x) + (s-x) \log(s-x) - s]
\end{aligned} \tag{2.2.3}$$

which is the formula for the GMD of a point on a line to the line itself. The symmetry of the formula is immediately apparent: a point located at a distance  $x$  from either end of the line will have exactly same GMD, just as we would expect.

Now, integrating a second time from 0 to  $s$  with respect to  $x$  and multiplying by  $1/s$ , we will get the self GMD of the line.

$$\log(g) = \frac{1}{s} \times \frac{1}{s} \int_0^s x \log(x) + (s-x) \log(s-x) - s \, dx \tag{2.2.4}$$

Unfortunately, the second integration gets rather messy. So, as was done in the case of the **GMD** of two co-linear lines, we will split the formula into terms **A**, **B**, **C**, etc., integrate them separately, and then add them back together. Thus:

$$\begin{aligned}
A &= x \log(x) \\
B &= (s-x) \log(s-x) \\
C &= s
\end{aligned}$$

And term **B** will be split into **B<sub>1</sub>** and **B<sub>2</sub>**:

$$\begin{aligned}
B_1 &= s \log(s-x) \\
B_2 &= x \log(s-x)
\end{aligned}$$

Integrating the individual terms:

$$\begin{aligned}\int_0^s A \, dx &= \int_0^s x \log(x) \, dx = \left[ \frac{x^2}{2} \log(x) - \frac{x^2}{4} \right]_0^s \\ &= \frac{s^2}{2} \log(s) - \frac{s^2}{4}\end{aligned}$$

$$\begin{aligned}\int_0^s B_1 \, dx &= \int_0^s s \log(s-x) \, dx = s \int_0^s \log(s-x) \, dx \\ &= s \left[ (x-s) \log(s-x) - x \right]_0^s \\ &= s \left[ ((s-s) \log(s-s) - s) - ((0-s) \log(s-0) - 0) \right] \\ &= s [s \log(s) - s] \\ &= s^2 \log(s) - s^2\end{aligned}$$

$$\begin{aligned}\int_0^s B_2 \, dx &= \int_0^s x \log(s-x) \, dx \\ &= \left[ \frac{x^2 - s^2}{2} \log(s-x) - \frac{sx}{2} - \frac{x^2}{4} \right]_0^s \\ &= \left[ \left( \frac{s^2 - s^2}{2} \log(s-s) - \frac{s^2}{2} - \frac{s^2}{4} \right) - \left( \frac{-s^2}{2} \log(s) - \frac{0}{2} - \frac{0}{4} \right) \right] \\ &= \left[ \left( -\frac{s^2}{2} - \frac{s^2}{4} \right) - \left( \frac{-s^2}{2} \log(s) \right) \right] \\ &= \frac{s^2}{2} \log(s) - \frac{3}{4} s^2\end{aligned}$$

$$\begin{aligned}
\int_0^s B \, dx &= \int_0^s B_1 \, dx - \int_0^s B_2 \, dx \\
&= \left( s^2 \log(s) - s^2 \right) - \left( \frac{s^2}{2} \log(s) - \frac{3}{4} s^2 \right) \\
&= \frac{s^2}{2} \log(s) - \frac{1}{4} s^2
\end{aligned}$$

$$\begin{aligned}
\int_0^s C \, dx &= \int_0^s s \, dx = \left[ sx \right]_0^s = s^2 - 0 \\
&= s^2
\end{aligned}$$

Now recombining the terms and simplifying:

$$\begin{aligned}
\log(g) &= \frac{1}{s^2} \left[ \int_0^s A \, dx + \int_0^s B \, dx - \int_0^s C \, dx \right] \\
&= \frac{1}{s^2} \left[ \left( \frac{s^2}{2} \log(s) - \frac{s^2}{4} \right) + \left( \frac{s^2}{2} \log(s) - \frac{s^2}{4} \right) - \left( s^2 \right) \right] \\
&= \frac{1}{s^2} \left( s^2 \log(s) - \frac{3}{2} s^2 \right) \\
&= \log(s) - \frac{3}{2}
\end{aligned} \tag{2.2.5}$$

Taking the anti-log of both sides we get:

$$\begin{aligned}
g &= e^{\log(s) - \frac{3}{2}} \\
&= e^{\log(s)} \times e^{-\frac{3}{2}} \\
&= se^{-\frac{3}{2}} \\
&= 0.22313s
\end{aligned} \tag{2.2.6}$$

which is the well known formula [4](page 167) for the self GMD of a straight line of length  $s$ .

Further GMD formulae for straight lines are given by Gray [5], including the case of parallel lines of equal length. By means of additional integrations, he then expands this to find the GMD of rectangles (of which squares are a special case). Gray's derivation is presented in good detail and is recommended reading.

Derivations of the GMD for circles and circular areas are given by Gray [5] and in more detail by Seneff [6]

## Part 3 - Calculation of GMD using numerical methods

### 3.1 The Monte Carlo Method

From the previous section, it can be seen that the derivation of formulae for GMD for even the simplest shapes can get quite involved. For more complex shapes, it may be more practical to use numerical methods to determine GMD. The Monte Carlo method is suitable for this, and can find the GMD of complex shapes to useable accuracies with relatively little computing power. These methods can also be used to check analytical GMD derivations. The Monte Carlo method involves the generation of random data points in the object under consideration which are then used to calculate the desired result. This is repeated many times with more random data points, and the results are averaged to produce a good approximation of the true value.

For the case of the self GMD of a straight line, the method is almost trivial. Simply generate two random numbers in the range from 0 to 1 (representing random points on a line of length = 1), take the logarithm of the absolute value of their difference and then repeat this many times, keeping a running total of the result of each iteration. Finally, divide the total by the number of trials and take the exponential. The program code is even shorter than this brief explanation:

```
N = 1000000000
Sum = 0
For i = 1 to N
    Sum = Sum+log(abs(Rnd()-Rnd()))
next
GMD = exp(Sum/N)
Print GMD
```

One quick run of the program immediately verifies the value of 0.22313, as was derived in the previous section.

The accuracy of the Monte Carlo method is approximately proportional to the square root of the number of random samples. Therefore, to get an improvement in accuracy of one additional significant digit, the number of random samples must be increased by a factor of 100. So, although the method is usually very simple to implement, it may require a lot of computing power if more than five significant figures are required in the result.

### 3.2 The GMD of a Triangle

For a slightly more complicated example, suppose we wish to find the self GMD of the area enclosed by an equilateral triangle. (Doing a search of the technical literature, it appears that no one has investigated the GMD of a triangular area, making this exercise somewhat worthwhile.) In the physical world, an analogous methodology might be as follows:

1. Draw an equilateral triangle on the wall.
2. Throw darts at the wall randomly, until two of the darts fall within the bounds of the triangle.
3. Measure the distance between the points where the two darts landed. Take the logarithm of this distance, and record it.
4. Repeat steps 2 and 3 many many many times.
5. Calculate the total of the recorded  $\log(\text{distance})$  measurements, and then divide this total by the count of measurements. Take the anti-log of this value. The result will be, approximately, the self GMD.

There are two primary requirements for this method to produce acceptable results:

1. The darts must have an equal probability of hitting any spot on the triangular area. That is, the  $x$  and  $y$  coordinates of the dart positions must have a uniform random distribution. (This requires very special dart playing skills, which are, unfortunately, disadvantageous for tournament play.)
2. Sufficient trials must be conducted in order to average out the random noise.

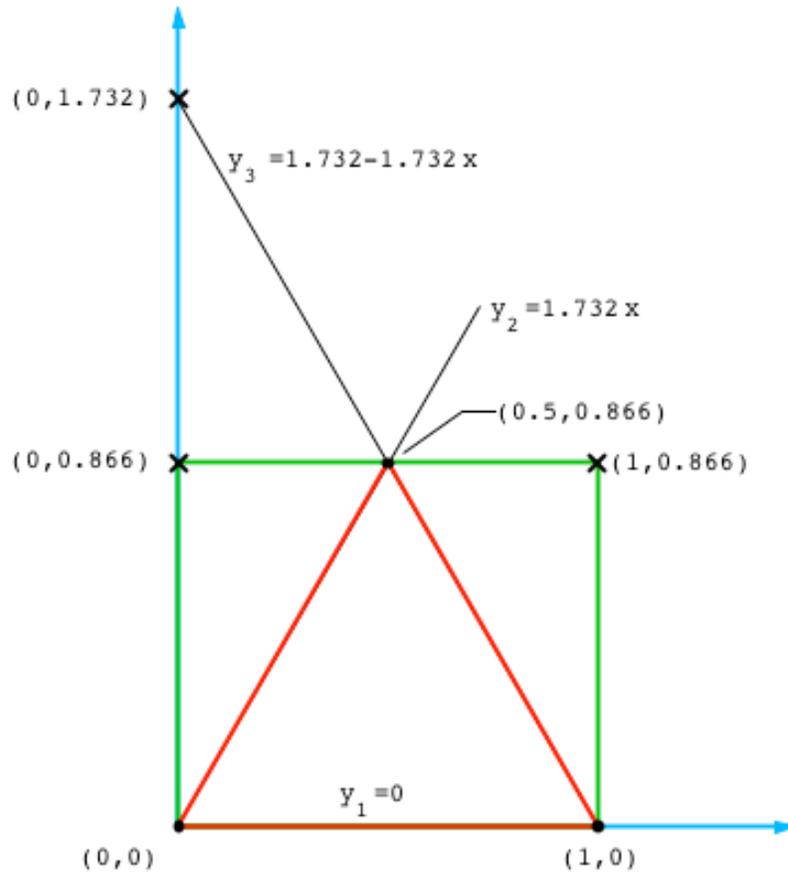
To get reasonable results, we likely will want to throw several million darts. Therefore, it may be more appealing to give this job to a computer.

Most computer languages provide a random number generating function, typically generating numbers in the range  $0 \leq N_R < 1$ , where  $N_R$  is the random number. Assuming that the random number function produces uniformly distributed random numbers<sup>3</sup>, we must make sure that we don't subsequently manipulate them in such a way that we render them non-uniform, while fitting them into the test shape that we are analyzing. A reliable method is to place the shape inside a bounding rectangle. Then, generate a random  $x$  coordinate scaled to the width of the bounding rectangle, and a random  $y$  coordinate scaled to the height of the bounding rectangle. The random point will then fall within the bounding rectangle. Then, it must be determined if the resulting random point also falls inside the bounds of the test shape. This is accomplished by applying the appropriate geometric formulae which define the test shape. If the point does not fall within the bounds of the test shape, then it is discarded and the process is repeated.

---

<sup>3</sup> Random number generators that are built into spreadsheets and various programming languages tend to be deficient, and begin to display non uniform behaviour for very large sample sizes. For this reason it is a good idea to implement an independent and well proven pseudo-random number generator routine. Eg., the Wichman and Hill algorithm [9].

Continuing with the example of an equilateral triangle, refer to the diagram below. Let's assume a side length of one, and one of the sides lying on the  $x$  axis, leftmost vertex at the origin, and the rightmost vertex at  $x = 1$ ,  $y = 0$ , and then it follows that the third vertex is at  $x = 0.5$ ,  $y = 0.866 = \sin(60^\circ)$ . The triangle is shown in red. The bounding rectangle, shown in green, will have a width of 1 and a height of 0.866.



The three lines forming the triangle are defined as:

$$y_1 = 0 \tag{3.2.1}$$

$$y_2 = 2x \sin(60^\circ) = 1.732x \tag{3.2.2}$$

$$y_3 = (2 - 2x) \sin(60^\circ) = 1.732 - 1.732x \tag{3.2.3}$$

To generate the random points, the  $x$  coordinate random number,  $n_x$ , is scaled to the range of 0..1, and the  $y$  coordinate random number  $n_y$ , is scaled to the range of 0..0.866. To determine whether the random point is within the bounds of the triangle, the following conditions must be true:

$$n_y \leq 1.732 n_x$$

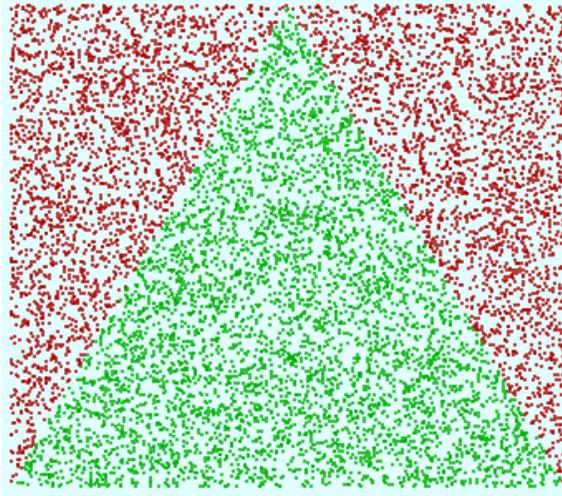
$$n_y \leq 1.732 - 1.732 n_x$$

There is no need to test that the point lies on or above the line  $y = 0$  because the  $y$  coordinate random number will always be greater than or equal to zero. Following, is the listing for a BASIC program which performs the calculation.

```
'calculate self GMD of equilateral triangle with side length equal to 1
a = sin(60/180*pi) 'Height (altitude) of triangle
b = 1 'Base of triangle
N = 1000000000 'Number of random pairs of points inside triangle
Sum = 0 'Sum of all log(distance) calculations
Npt = 0 'Count of all random points generated
For i = 1 to N
  'Generate test point T
  do
    'Generate point tx,ty that falls inside the circumscribed rectangle.
    tx = rnd*b
    ty = rnd*a
    tya = 2*a*tx 'eqn of first bounding line
    tyb = 2*a-2*a*tx 'eqn of second bounding line
    Npt = Npt+1
    'Test whether the point falls within the
    'bounds of the triangle, and repeat until true
  loop until (ty< = tya and ty< = tyb)
  'Now generate the second test point S
  do
    sx = rnd*b
    sy = rnd*a
    sya = 2*a*sx
    syb = 2*a-2*a*sx
    Npt = Npt+1
  Loop Until (sy< = sya and sy< = syb)
  'Calculate the distance between points T and S,
  ' then take the log and add to the sum
  Sum = Sum+log(Sqrt((sx-tx)^2+(sy-ty)^2))
next
'Calculate GMD from the sum of Log(distance)
GMD = exp(Sum/N)
'Accuracy check based on comparison of Monte Carlo
' area calculation compared to analytical area.
'Monte Carlo area is equal to inside/outside points
' ratio (2N/Npt) times bounding rectangle area.
'The bounding rectangle area is a*b
Aarea = a*b*2*N/Npt
'True area calculation
Tarea = 0.5*a*b
```

Referring to the last few lines of the program, to get an indication of the accuracy of the Monte Carlo calculation, we can keep count of the total number of points that fall inside the triangle compared to the total number of points generated. The ratio will approximately equal the ratio of the area of the triangle to the area of the rectangle—the approximation improving as the number of test points increases. The true ratio in the present example will be exactly  $\frac{1}{2}$ . By checking the number of significant digits that are in agreement with this value, we can get a reasonable estimate of the number of significant digits of the GMD.

The following figure shows a plot of points from a program run of  $10^8$  random points. The points falling inside the triangular area are green, and the remainder are red. (Only a fraction of the total points have been plotted.)



In a subsequent program run, the number of points falling inside the triangle was 20000000000 out of a total of 39999997361 generated points, suggesting an accuracy of about seven significant figures. The self GMD value produced in the same run, was 0.30838221. It should then be safe to assume a self GMD of perhaps six significant figures, or 0.308382. Hence, we can conclude that the self GMD of an equilateral triangular area with a side length  $s$ , will be  $0.308382s$ .

$$g = 0.308382 s \quad (3.2.4)$$

Naturally, having arrived at the self GMD value of 0.308382, we must immediately be curious whether this happens to be the anti-logarithm of a simple fraction, as the factor would be in the case of a circular area ( $-\frac{1}{4}$ ) or a straight line ( $-\frac{3}{2}$ ). Alas, the natural logarithm of 0.308382 is  $-1.17642$ , which doesn't suggest any obvious simple fraction. The nearest fraction with a reasonably small denominator is  $-\frac{20}{17} = -1.17647$ . It's likely unwise to draw any conclusions about a simple fraction in this case.

### 3.3 The GMD of an Elliptical Area

As a another example, we can consider the case of the self GMD of a circle or ellipse. The program code is virtually the same as the case of the triangle. The only difference being the formulae of the shape's bounding lines. For an ellipse with a horizontal axis of  $2b$  and vertical axis of  $2a$ , the formula for the upper half of the curve is:

$$y = a \sqrt{1 - (x/b)^2} \quad (3.3.1)$$

And the formula for the lower half of the curve is:

$$y = -a \sqrt{1 - (x/b)^2} \quad (3.3.2)$$

We will depart from the usual convention of referring to the ellipse axes as “major” and “minor” and use “horizontal” and “vertical” instead. The horizontal axis may be either the major or minor axis, and we will define the horizontal axis to be the line in which the ellipses will be offset.

If  $a = b$ , then the figure will be a circle. If  $a \neq b$ , then the figure will be an ellipse. The program code follows:

```
' Monte Carlo method for determining the Self or Mutual GMD of ellipses
a = 3 'vertical semi-axis
b = 1 'horizontal semi-axis
x0 = 0 'horizontal offset of the conductors (zero for self GMD)
y0 = 0 'vertical offset of the conductors (zero for self GMD)
N = 1000000000 'Number of random pairs of points inside ellipse to be evaluated
Npt = 0 'Count of all random points generated
Sum = 0 'Sum of all log(distance) calculations
For i = 1 to N
  'Generate test point T
  do
    'Generate point tx,ty that falls inside a circumscribed rectangle.
    tx = rnd*2*b-b
    ty = rnd*2*a-a
    tya = a*sqrt(1-(tx/b)^2) 'eqn of upper bound of ellipse
    tyb = -tya 'lower bound of ellipse
    Npt = Npt+1
    'Test whether the point falls within the
    'bounds of the ellipse; repeat until true
  loop until (ty< = tye and ty> = tyel)
  'Generate test point S
  do
    sx = rnd*2*b-b
    sy = rnd*2*a-a
    sya = a*sqrt(1-(sx/b)^2)
    syb = -sya
    Npt = Npt+1
  Loop Until (sy< = sye and sy> = syel)
  'Calculate the distance between points T and S,
  ' then take the log and add to the sum
  Sum = Sum+log(Sqrt((sx-tx+x0)^2+(sy-ty+y0)^2))
next
'Calculate GMD from the sum of Log(distance)
GMD = exp(Sum1/N)
'Accuracy check based on comparison of Monte Carlo
' area calculation compared to analytical area
'Monte Carlo area is equal to inside/outside points
' ratio (2N/Npt) times bounding rectangle area.
'The bounding rectangle area is 2a*2b
Aarea = 8*a*b*N/Npt
'True area of ellipse
Tarea = pi*a*b
End
```

The analytical formula for the self GMD of an ellipse is given by Grover [2] (page 21) as

$$g = 0.5(a+b) e^{-1/4} \quad (3.3.3)$$

A quick program run verifies agreement with this formula.

Grover is a bit ambiguous about whether this formula is supposed to apply to an elliptical area, or an elliptical line, or both. It is, in fact, the formula for an elliptical area as the above program run has demonstrated, and not for the GMD of an elliptical line. Since a circle is a degenerate form of an ellipse, we see that the formula will give the correct value for a circular area when  $a = b$ . But since the self GMD of a circular line is simply equal to its radius, Grover's formula is obviously not applicable in the case of the elliptical line. Alas, developing a numerical method to find the GMD of an elliptical line requires a more sophisticated Monte Carlo simulation than what has been discussed so far, but will be addressed in a later section.

In addition to the calculation of self GMD, the Monte Carlo method can of course be applied, with almost no additional effort, to the calculation of GMD between two shapes external to each other. In fact, it is as simple as setting non-zero values to the variables  $x0$  and  $y0$  in the above program code. These variables are the horizontal and vertical offsets of the shapes being analyzed.

Now, whereas the case of self GMD is dependent only upon the geometry of the shape, and it scales linearly with its size, the GMD between two objects is often a non linear function of the distance separating them, and also varies depending upon the relative orientation of the objects. The case of two circles is a notable exception, because the GMD is simply the distance between their centres, and a circle is identical with any orientation. But for non-circular shapes, the calculation must be performed for every different spacing and orientation of the objects. It may be possible to analyze the data generated by these calculations, and then derive an empirical formula that predicts the GMD, but there is no guarantee that this will be a simple formula. If we are using this method to determine the GMD of only a small group of conductors, then it doesn't matter too much, because there are only a few GMD values that need to be calculated, and we can calculate each of them directly using the Monte Carlo method. When dealing with a large number of evenly spaced conductors, the most practical solution may be to calculate the cases for adjacent conductors, and then for  $1 \times$ ,  $2 \times$ ,  $3 \times$  conductor pitch separations, and so on up to, perhaps,  $5 \times$  conductor pitch. These can be stored in a lookup table, and used as required in an inductance calculation. For any spacing greater than what is in the table, the centre to centre conductor spacing can be used, since the true GMD will quickly converge to this value as the spacing becomes significantly larger than the conductor mean diameter.

For example, the table to the right shows the results of calculating the GMD of ellipses with a horizontal semi-axis of 0.5 and a vertical semi-axis of 1.0, for offsets ranging from 0 to 20. The GMD for a zero offset is the self GMD of a single ellipse.

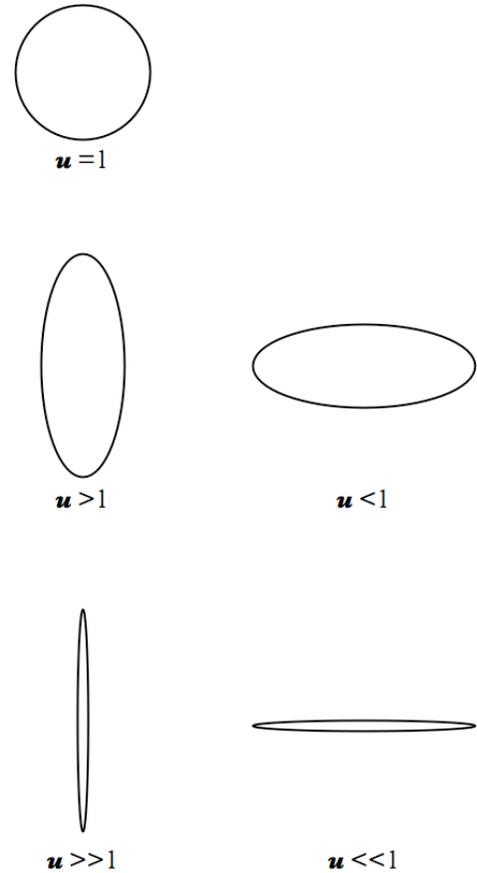
<b>a</b>	<b>b</b>	<b>offset</b>	<b>GMD</b>
0.5	1	0	0.5840898753
0.5	1	2	1.8946815473
0.5	1	4	3.9520103933
0.5	1	6	5.9684639214
0.5	1	8	7.9762391919
0.5	1	10	9.9810447569
0.5	1	12	11.9845009509
0.5	1	14	13.9863041625
0.5	1	16	15.9881743067
0.5	1	18	17.9895128001
0.5	1	20	19.9905701963

For each calculation, 200 million points were generated, and it is probably safe to rely on five significant figures of the GMD results.

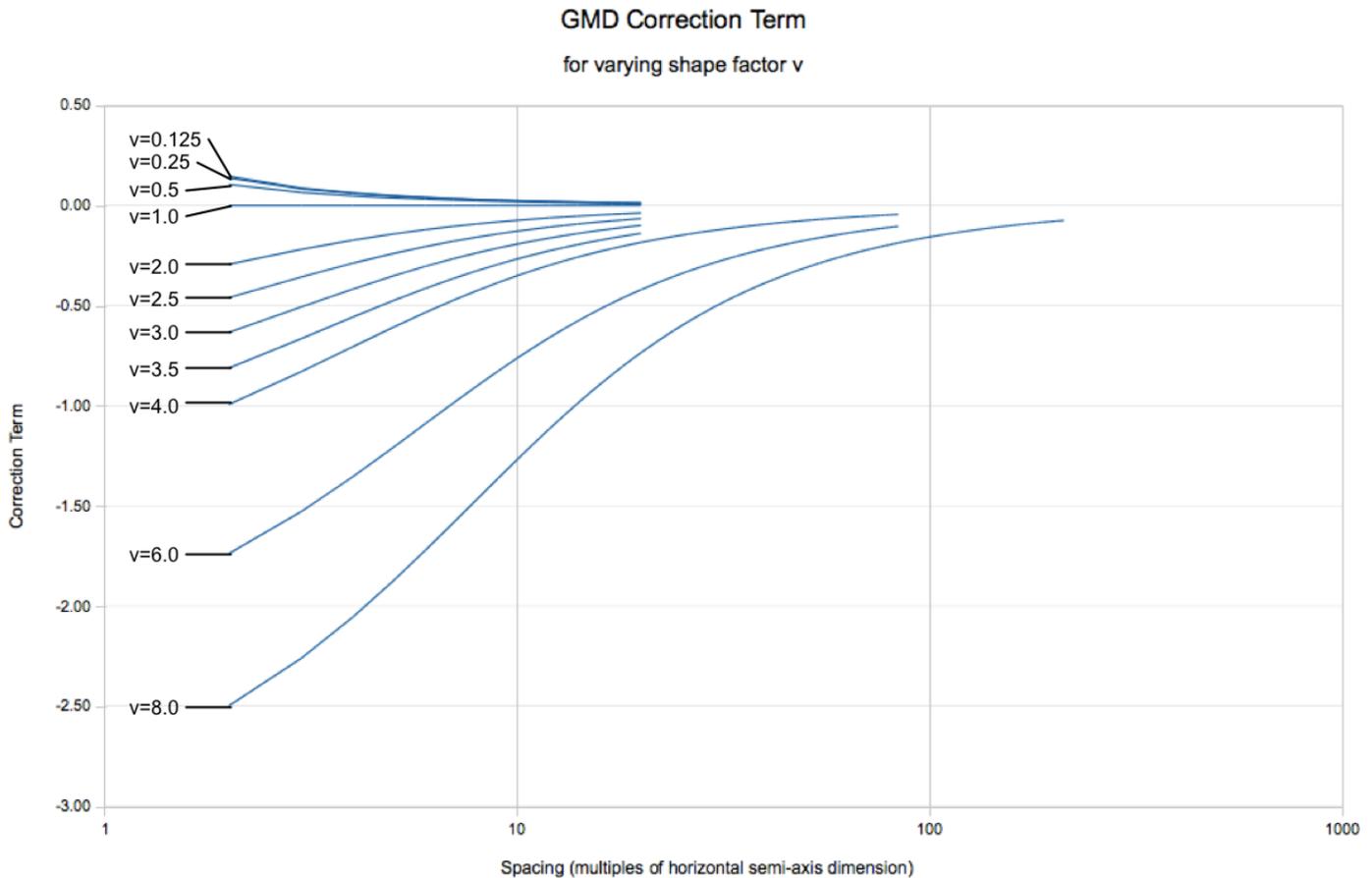
It can be seen that as the offset increases in value, the GMD quickly converges towards the offset value. If we are to come up with an empirical function, based on the offset, it seems reasonable to use the actual offset value as the principal term, and then determine an additional correction term to adjust the final value. We will use the function  $f(x,u)$  for the correction term, where  $x$  is the offset (which is measured from centre to centre of the shapes), and  $u$  is a shape factor (see diagram to the right, and description below). Hence  $GMD = x + f(x,u)$ . We must be careful, because we must find the correction term by taking the difference between the Monte Carlo GMD value and the offset. This difference becomes very small very quickly as the offset value increases, which can result in noisy data for large offsets. Fortunately, this is the part of the range where the correction term will tend towards zero and has little significance.

The above table lists the GMD only for ellipses with an  $a/b$  shape factor of 0.5. An empirical formula must be general enough to accommodate different shape factors.

We will define the variable  $u$  to be the shape factor and it will be the ratio of the vertical semi-axis to the horizontal semi-axis. Hence  $u = a/b$ .



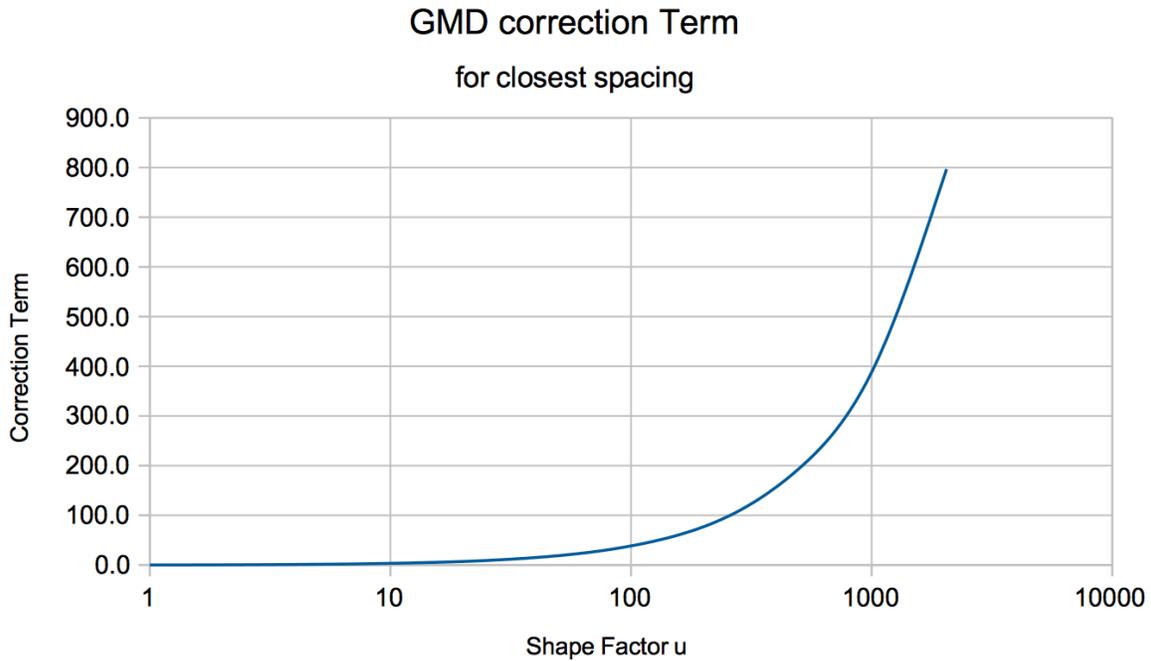
The Monte Carlo program was run for shape factors varying from 0.125 to 8.0, and the GMD correction factor is plotted in the following graph:



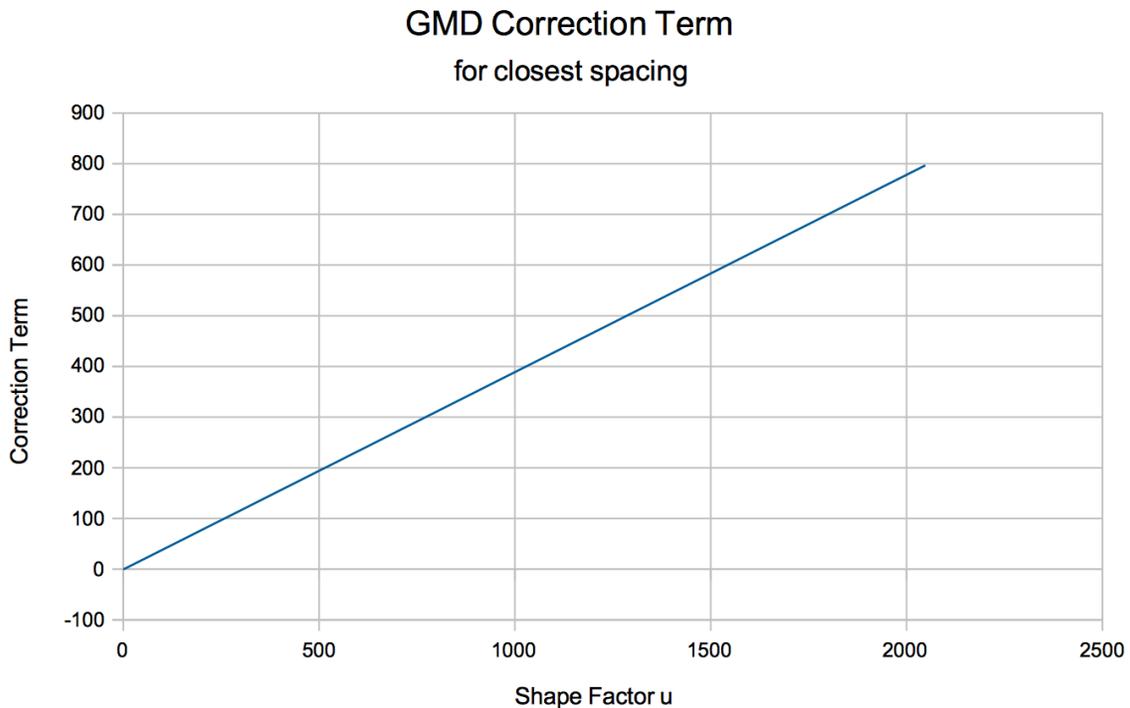
The leftmost point of each curve corresponds to two ellipses, of the specified shape factor, just touching each other. It can be seen that the correction term is largest at this point and then decays to zero as the spacing increases.

Developing a fitting function of more than a single argument is usually a major undertaking. It's worthwhile to look at developing a function of one variable while the other is held fixed. We can fix either the shape factor  $u$  or the offset  $x$ . Let us look at developing a function with a fixed offset  $x$  and variable shape factor  $u$ . The offset will be fixed at  $2b$  (double the horizontal semi-axis) so that the two ellipses will be just touching each other in every case. The rationale for proceeding in this order is that the most critical GMD values are the ones for the closest spacings, and if we develop a formula for closely spaced ellipses first, and then later develop a secondary *decay* function to correct for spacing, then it is much easier to ensure that the overall function is most accurate at the minimum offset value.

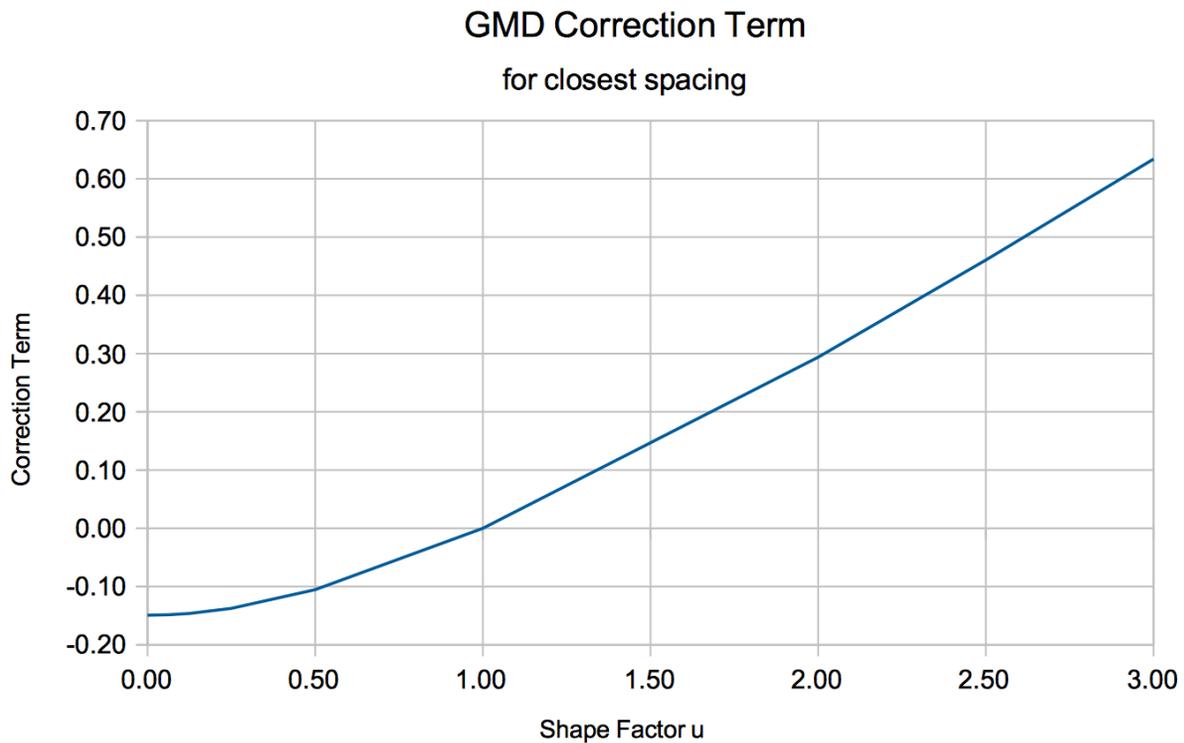
Therefore, let us re-plot the above data to show the required correction for different values of  $u$  for a fixed offset of  $2b$ . We will include a larger range of shape factors varying from  $10^{-13}$  to 2000. This larger range allows us to get a better idea of any possible asymptotic behaviour.



The graph shows what may be a horizontal asymptote near zero, and the curve appears to asymptotically approach an inclined straight line at large values of  $u$ . The inclined asymptote is much more obvious when the graph is redrawn with a linear horizontal axis:



Similarly the horizontal asymptote is more obvious on a linear plot restricted to small values of  $u$ :



Further analysis of the data confirms the asymptotes. The horizontal asymptote is, in fact, at a value of about  $-0.149261$ , and the inclined asymptote has a slope of  $0.38940$ . The existence of and the values of these asymptotes can be easily explained.



In the case of very large shape factor values (see figure at left), the horizontal component of the distance between points on the ellipses becomes insignificant compared to the vertical coordinate. The two ellipses begin to approximate vertical lines which are nearly superimposed. As such, the GMD begins to approximate the self GMD of an ellipse of the same shape factor. That is:

$$g = 0.5(a+b) e^{-1/4}$$

But because  $u = a/b$ , (hence,  $a = bu$ ):

$$g = 0.5(ub+b) e^{-1/4}$$

or

$$g = 0.5b(u+1) e^{-1/4}$$

Note when  $u$  is very large,  $u+1 \approx u$  and asymptotic nature is apparent. At large values of  $u$ , we have an asymptote with a slope of  $0.5e^{-1/4}$  or  $0.38940$ , just as the data show.

The horizontal asymptote for small values of  $u$  can be explained in a similar way. Very small values of  $u$ , indicate two extremely flat looking ellipses located end to end (see diagram below).



Because they are so flat, these appear much like two co-linear lines of unit length just touching. We might think therefore, that they would approach the value calculated for touching co-linear lines previously derived as  $2 \times 4e^{-3/2} = 1.7850$ . (The additional factor of 2 accounts for the fact that the closest spacing is  $2 \times b$ .) Instead they approach a limiting GMD value of 1.8507. The reason for this is that even though the ellipses appear to become straight horizontal lines, the distribution of points is not linear, as they would be in the case of lines. Instead, the points are more dense in the centres of the ellipses, and so this results in the slightly larger limiting value of GMD. However, the horizontal distribution of points remains the same regardless of the shape factor, and so a limiting value is reached at small values of  $u$ , when the vertical coordinate of the points becomes insignificant compared to the horizontal coordinate. Subtracting the fixed offset of 2 from the limiting GMD value of 1.8507, gives the asymptote value of  $-0.1493$ .

It is encouraging to find that we are dealing with a doubly asymptotic function. We can come up with a simple fitting function that gives these same asymptotes, and then we only have to worry about minimizing error in the intermediate region. A function which accomplishes this is:

$$f_v(u) = k_0u + k_1 + k_2 / (1 + k_3u)$$

$k_0$  is the slope of the inclined asymptote and has a value of 0.38940 as previously determined.

The sum of  $k_1 + k_2$  is equal to the horizontal asymptote of  $-0.1493$ .

The values of  $k_2$  and  $k_3$  determine the shape of the curve in the transition region. It is also possible, if necessary, to add additional terms to the denominator of the last term to help minimize error. It turns out that additional terms are unnecessary, because the existing terms can be optimized to give a peak error of no more than 0.6%. The optimized values are:

$$k_0 = 0.38940$$

$$k_1 = -0.71245$$

$$k_2 = 0.56397$$

$$k_3 = 0.73139$$

Thus, the correction term is:

$$f_v(u) = 0.38940u - 0.71245 + 0.56397 / (1 + 0.73139u)$$

And the complete GMD formula for touching ellipses is:

$$g = 2 + [0.3894u - 0.71245 + 0.56397 / (1 + 0.73139u)] \quad (3.3.2)$$

To get the complete correction term  $f(x, u)$ , we must combine  $f_v(u)$  with another function that is dependent on the offset  $x$ . We will now introduce a decay function,  $f_x(x)$ , to account for variable

offset values. This function must have a value of 1 at  $x = 2$  (the minimum offset) and then decrease asymptotically towards zero as  $x$  becomes large. A simple function meeting the requirements is:

$$f_x(x) = k_4 / (k_5 + k_6x)$$

which, when combined with  $f_v(u)$ , gives a complete formula for the correction term:

$$f(x,u) = [k_0v + k_1 + k_2 / (1 + k_3v)] \times [k_4 / (k_5 + k_6x)]$$

And ultimately, the complete normalized GMD formula:

$$g = x + [k_0v + k_1 + k_2 / (1 + k_3v)] \times [k_4 / (k_5 + k_6x)]$$

The function  $f_x(x)$  is known as a rational function because it is the ratio of two polynomials, although this is a fairly trivial example. The polynomial in the numerator is simply the constant  $k_4$ , and the polynomial in the denominator is  $k_5 + k_6x$ . Neither are overly complicated. However, rational functions can have a polynomial of any degree in both the numerator and denominator, and because they can be tailored to fit a wide variety of curves they are popular as fitting functions. (They will be employed several more times as fitting functions in later sections.) It turns out that the simple form given here will fit the data just as well as higher degree polynomials.

The optimum values for the constants are:

$$k_4 = 3.55610$$

$$k_5 = 2.54114$$

$$k_6 = 0.49887$$

Adding more terms of higher power gives no improvement beyond what this basic function gives. These constants give a peak error of 1.78% in the overall GMD value, which is fair, considering that  $f_v(u)$  and  $f_x(x)$  are independent of each other and the functions are very simple. Unfortunately, the worst case error occurs close to the minimum spacing where it does the most harm. Therefore, it's worth putting a bit more effort into the development of a decay function with better characteristics.

Looking at the original set of curves, it can be seen that there is a point of inflection for each curve which occurs at a different position. The varying location of this inflection point makes it difficult to apply a simple decay function based only on the offset distance. It appears that the inflection point occurs approximately at  $x = u$ . It may be beneficial therefore to shift the argument of the decay function by a factor of  $u$ . Hence  $f_x(x)$  now becomes  $f_x(x + k_7u)$ . This does give some further improvement. In addition, once the constants  $k_4$  through  $k_7$ , have been optimized, we can then do a final re-optimization all of the constants  $k_1$  through  $k_7$ , and during the optimization, weight the optimizer's error criteria so that errors at the lowest offsets have the most influence on the final result. During this optimization, it is also useful to rearrange the  $f_v(u)$  function as:

$$f_v(u) = k_0u + (k_1 + k_2u) / (1 + k_3u)$$

With suitable adjustment of the values  $k_1$  and  $k_2$ , this is completely equivalent to the previous form of  $f_v(u)$ , but allows for more convenient constant values. The constant  $k_1$  is now equal to the horizontal asymptote. However, rather than restricting its value to the known asymptote, we will

adjust it in order to improve the accuracy of the function over the more useful range of the function, thus sacrificing the correct asymptotic behaviour at very small values of  $u$ . However, this is somewhat compensated for by the simultaneous optimization of the decay function constants.

Doing the final optimization, we end up with a maximum peak error of about 0.5% and this peak error occurs at at more distant offsets than in the earlier optimization. The optimized function and the constants are:

$$g = x + [k_0u + (k_1 + k_2u)/(1+k_3u)] \times [k_4/(k_5+k_6(k_7u + x))] \quad (3.3.3)$$

$$k_0 = 0.38940$$

$$k_1 = -0.0025158$$

$$k_2 = -0.39042$$

$$k_3 = 0.0091452$$

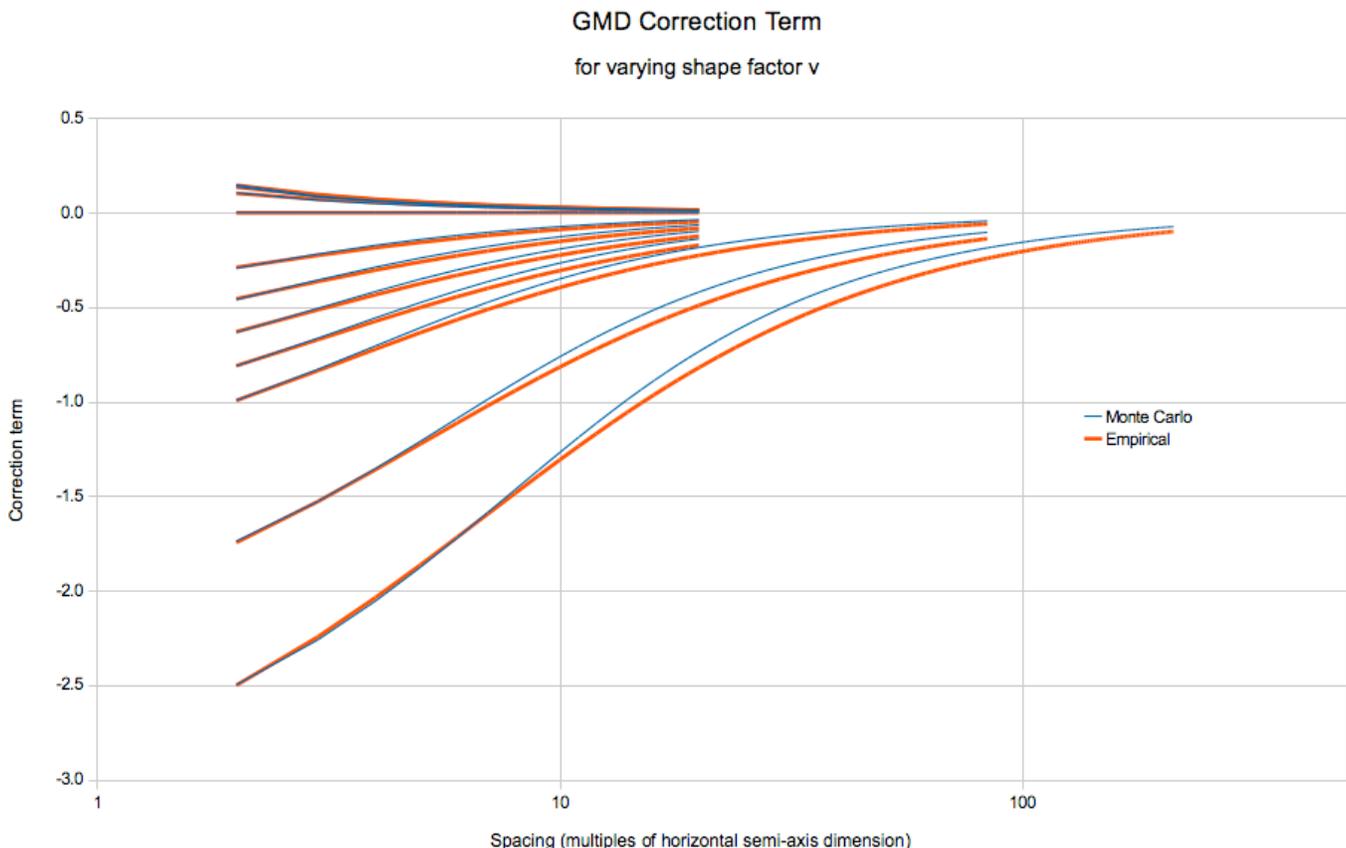
$$k_4 = 32.53253$$

$$k_5 = -0.05161$$

$$k_6 = 0.29995$$

$$k_7 = 0.86873$$

This is a large number of constants, but the formula itself is fairly simple. Overlaying the values produced by the empirical function onto the Monte Carlo calculated data, we get the following graph:



It can be seen that the approximation is very good at the left side of the graph where the offset is less than five times the ellipses' horizontal semi-axis, but then fails to decay to zero as quickly as the true data. Because this is a graph of the correction term only, the effect of the error appears to be worse than it is in practice. The overall error becomes a very small percentage of the net GMD value once the correction factor is added to the offset (0.5% peak error). Also, the way that GMD is used in multi-turn coil inductance calculations will further reduce the total error due to distant spacings. Additional work could be done to improve the decay function, but it is unlikely to provide any significant overall improvement in the final inductance calculations.

It should be remembered that this function has been normalized such that the offset value  $x$  is expressed as a multiple of the horizontal semi-axis  $b$ . Therefore,  $x$  must be calculated by dividing the actual offset distance by the horizontal semi-axis of the ellipse,  $b$ , and the result of the formula must then be multiplied by  $b$  to get the true GMD value. It should also be remembered that this function **cannot** be used to find the self GMD (by setting  $x = 0$ ), as this is well beyond the range of the data that were used for fitting the function.

In addition to ellipses, the method developed here should be readily adaptable to the creation of a GMD function for other oblong shapes. For example rectangles would be logical candidates, were it not for the fact that an analytical rectangle GMD formula already exists.

### 3.4 The Self GMD of an Elliptical Line

To distinguish between the area enclosed by an ellipse, and ellipse itself (excluding the internal area), we will refer to the hollow ellipse as an elliptical line or elliptical locus. In the previous section we discussed the GMD of the area enclosed by an ellipse. It was also briefly mentioned that a more sophisticated Monte Carlo method would be required for the case of an elliptical line. The problem arises when we try to generate uniformly distributed random points on the locus of the ellipse (or any curve for that matter). We cannot use the rejection sampling method that we used for the triangular and elliptical areas, because the locus has no area and therefore there is zero probability that a random point generated in the enclosing area will fall on the locus. We need to find another way to drop points randomly on the locus.

For the degenerate case of a circle, the solution is quite simple. We simply use the trigonometric parametric equations for a circle:

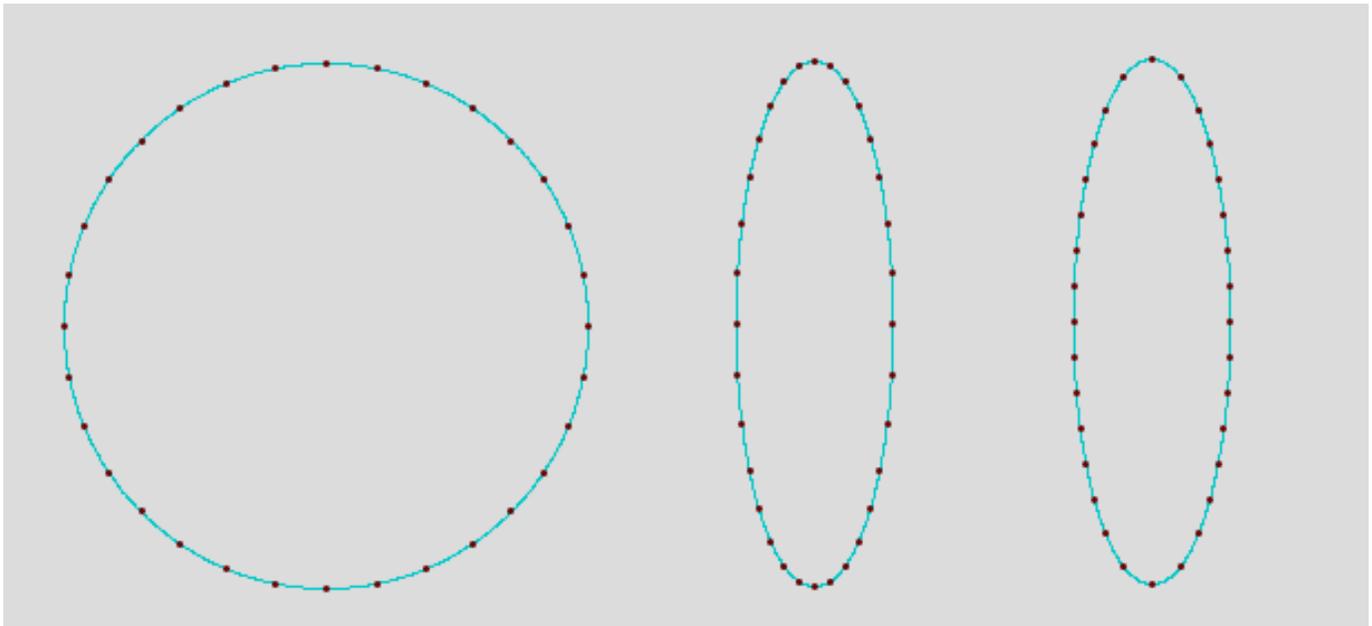
$$x = b \cos(\theta) \tag{3.4.1}$$

$$y = a \sin(\theta) \tag{3.4.2}$$

where  $a = b$ , and  $\theta$  ranges from 0 to  $2\pi$  radians.

If we assign to  $\theta$ , uniformly distributed random numbers in the range of 0 to  $2\pi$ , then we achieve the uniformly distributed points on the circular locus. However, in the more general case where  $a \neq b$ , then we have an ellipse, and this method fails. The diagram below shows what happens when we plot points at equal increments of  $\theta$  on a circle (left), and then on an ellipse (centre) with  $b = 0.3a$ . As the ellipse becomes squashed, the points become concentrated towards the ends.

Hence, if we were to assign uniformly distributed random values to  $\theta$ , the distribution of points on the ellipse would not be uniform.



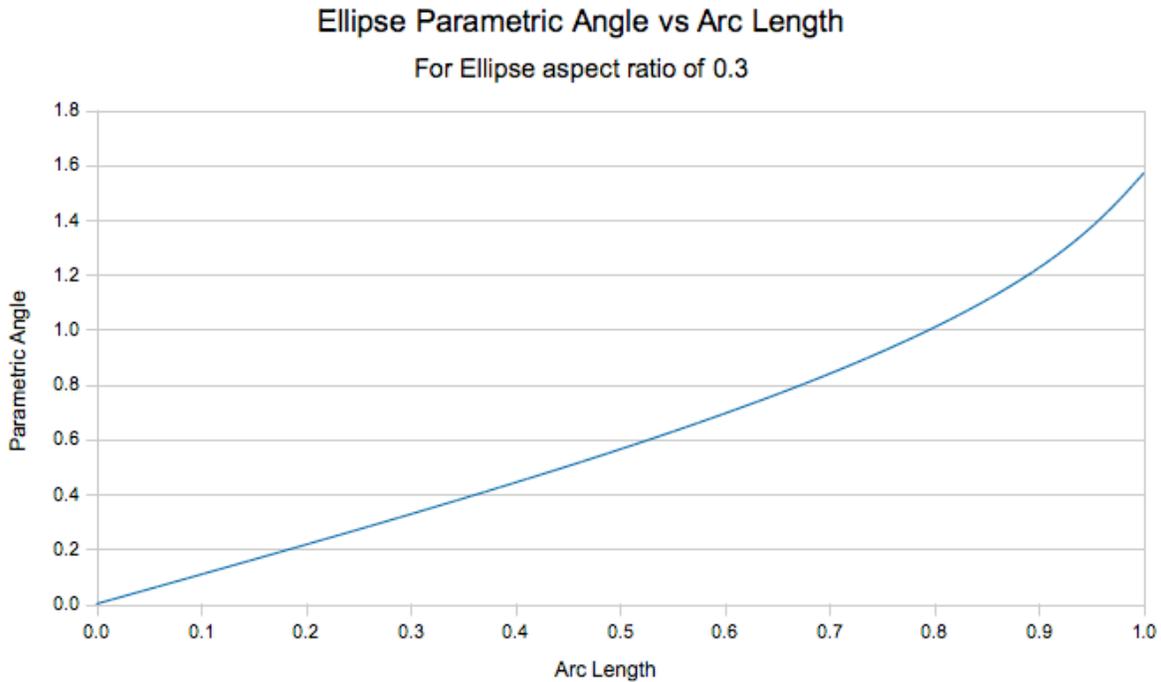
The figure on the far right, however, shows the desired uniform spacing of points. The fact that such a figure exists, with its evenly distributed points, means that there must be a way to achieve this distribution. To do this, we need to find a function that takes the distance along the locus as an argument, and returns the corresponding angle  $\theta$ . By feeding this function a set of uniformly distributed random numbers, we can then apply the resulting values of  $\theta$  as the parameters of the ellipse, to get the uniform point distribution. We will refer to this *distance to angle* function as the **mapping** function. The inverse of the formula we are looking for, would take the angle  $\theta$  as the argument and return the distance along the locus. This function exists; it is the ***Incomplete Elliptic Integral of the Second Kind***  $E(\phi, k)$ . We can use this function to generate a dataset of angle and length pairs, then create an empirical fitting function for these data that uses the arc length as argument and returns the angle, essentially the inverse of  $E(\phi, k)$ .

The first step then, is to generate the dataset. Unfortunately, an implementation of the function  $E(\phi, k)$  is not readily available—certainly not in Open Office Calc. One of the goals of this work was to avoid the use of special math software, but rather, to use simple software such as is available in a spreadsheet program, or if necessary, a simple programming language such as BASIC. Of course the function  $E(\phi, k)$  could be coded into BASIC, but this is not really necessary in order to generate the data. Instead, we simply use a spreadsheet, and the parametric equations for the ellipse to generate a list of  $x$  and  $y$  coordinates for values of  $\theta$  ranging from 0 to  $\pi/2$ , with  $\theta$  changing in very small increments. We can then calculate the length of the arc segments from one increment to the next using the formula:

$$\ell_{arc} = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad . \quad (3.4.3)$$

Then, by taking the running total of these  $\ell_{\text{arc}}$  values, we get the total length from the start of the locus to the specified point on the curve. This method has the advantage that it can also be used for shapes other than ellipses.

We need only generate the data for one quadrant of the ellipse, and then find a fitting function for that single quadrant. Because of its symmetry, the data for the other three quadrants can be easily determined from the single quadrant data set. Dealing with a single quadrant, makes it easier to find a fitting function. A graph of the data for the ellipse with an aspect ratio of  $u = b/a = 0.3$  is shown below:



Incidentally, if we had used an aspect ratio of 1 (a circle), the above graph would be a straight line. As the aspect ratio gets smaller, the line becomes more curved.

The arc length is measured from a starting point which we will define to be the point where the ellipse crosses the positive  $x$  axis. For an aspect ratio of 1 (a circle) where semi-major and semi-minor axes  $a = b = 1$ , the arc length will be equal to the angle in radians. So, for one quadrant the arc length is  $\pi/2$ . As the aspect ratio decreases, the quadrant arc length decreases until, at an aspect ratio of 0, the arc length is equal to the semi-major axis  $a$ . For this study we will normalize the quadrant arc length to a range of 0 to 1 for all aspect ratios. Hence the horizontal axis on the above graph ranges from 0 to 1.

Additionally, the arc will be measured counterclockwise from the starting point. We also define the geometry such that, as the aspect ratio becomes smaller, the ellipse is flattened in the  $x$  dimension. That is, the ellipse becomes tall and skinny. When the aspect ratio is one, the figure is a circle. When the aspect ratio is zero, the figure becomes a vertical line. This covers the entire range of shapes that the ellipse can assume.

To fit these data, a rational expression of the following form was chosen for the mapping function:

$$\theta = (k_1s + k_2s^2 + k_3s^3 + k_4s^4) / (1 + k_5s + k_6s^2 + k_7s^3) \quad (3.4.4)$$

where  $s$  is the length of the arc measured from the starting point and  $\theta$  is the parametric angle as discussed above. This function very likely has more terms than are necessary, but since it serves only a temporary purpose, maximum efficiency was not the goal. Optimizing the coefficients was done using a solver macro in an Open Office spreadsheet. For an aspect ratio of  $u = 0.3$ , the optimum coefficients are:

$$k_1 = 1.09677644$$

$$k_2 = 0.31383321$$

$$k_3 = -2.92801438$$

$$k_4 = 1.52766397$$

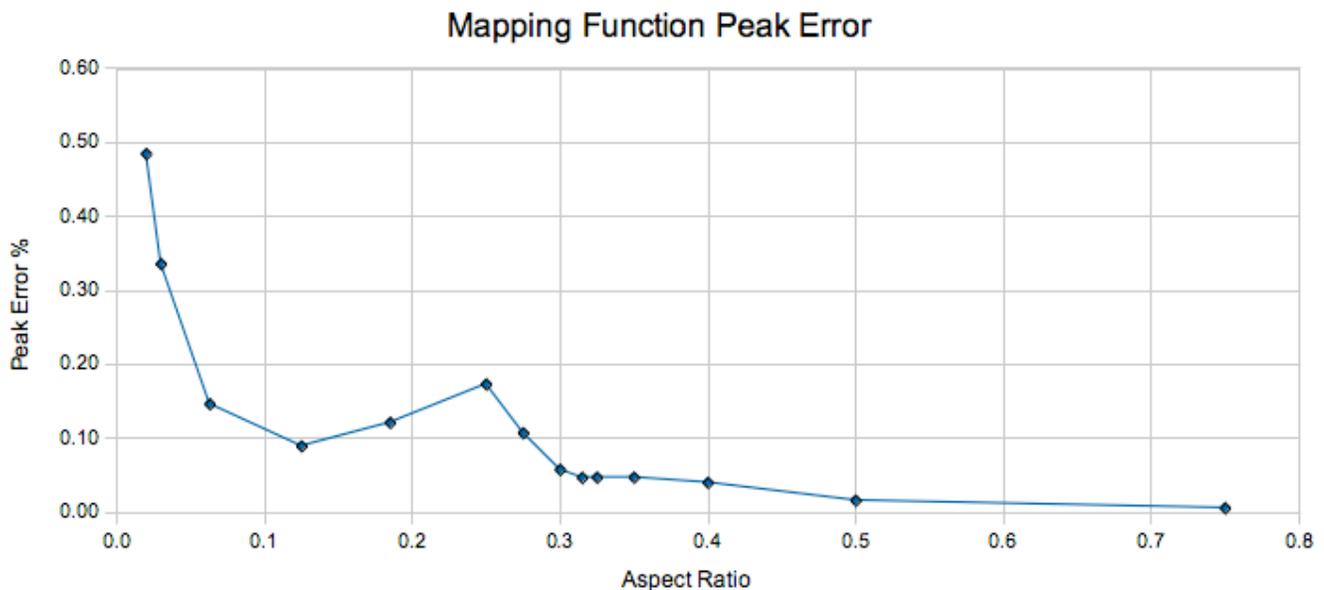
$$k_5 = 0.29919089$$

$$k_6 = -2.96132963$$

$$k_7 = 1.66867001$$

Of course this gives us a function only for the case where the aspect ratio of the ellipse is equal to 0.3. Developing a more general empirical function that takes the additional argument of the aspect ratio would be a far bigger job than is warranted. Instead, different sets of coefficients  $k_1..k_7$  were obtained for aspect ratios of 0.125, 0.25, 0.3, 0.35, 0.4, 0.5, and 0.75. After looking at the results—in particular, the fitting error—additional sets of coefficients were obtained for aspect ratios of 0.02, 0.03, 0.063, 0.315, and 0.325.

The peak errors for the various sets of coefficients of the fitting function are shown in the graph below.



It is not surprising to see the higher errors for smaller aspect ratios, because the mapping function becomes more curved for smaller ratios, and conceivably harder to fit. However, it is surprising to see the second peak at an aspect ratio of 0.25. This may simply be due the solver having difficulty finding the best fit in this case. The errors are still quite small however, and are not expected to have a significant impact on the final result.

Having created the mapping function, we can now take a set of random numbers, pass them through this function and then use the resulting values of  $\theta$  to create a set of uniformly distributed random points on an elliptical locus.

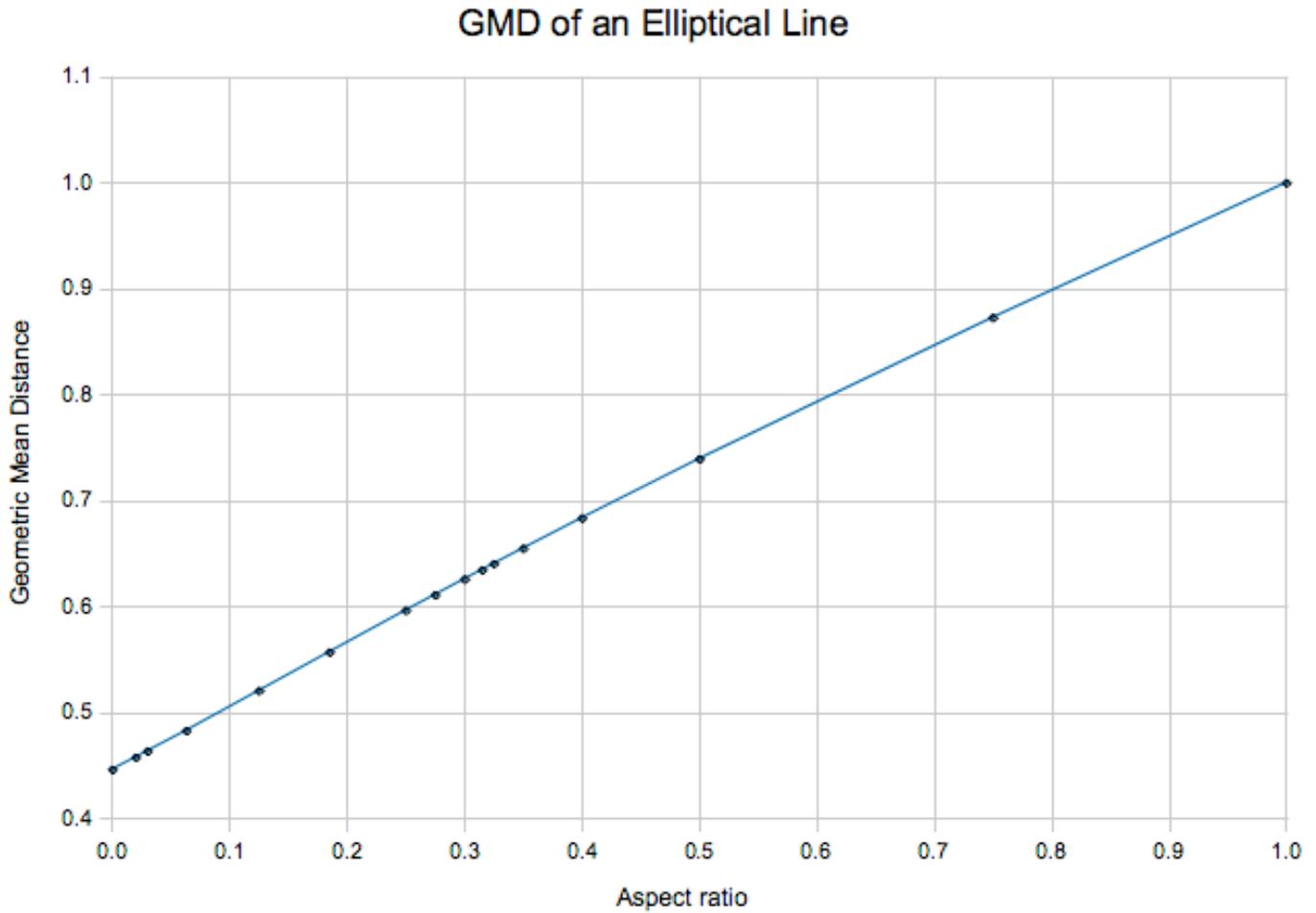
All this effort to produce a uniform distribution of random points on the locus of the ellipse has overshadowed an otherwise simple Monte Carlo algorithm. However, this mapping function is crucial, because without a means to generate uniformly distributed random points, the Monte Carlo method would be unusable.

We can now return to the primary issue. As with the previous calculations, the program code will generate random points on the elliptical locus, then measure the distance between pairs of them, take the logarithm, sum them, and hence, find the geometric mean. The program code is given in **Appendix A**. With  $10^9$  pairs of random points for each run, we get the results shown in the table to the right.

Aspect Ratio	GMD
0.020	0.4577052
0.030	0.4636561
0.063	0.4828448
0.125	0.5205759
0.185	0.5571103
0.250	0.5963800
0.275	0.6111260
0.300	0.6258739
0.315	0.6346541
0.325	0.6404787
0.350	0.6549459
0.400	0.6835489
0.500	0.7393943
0.750	0.8727630

These are normalized to a semi-major axis value of 1.0. In addition, there are two limiting values that can be included in this list. We know that for an aspect ratio of 1, the ellipse becomes a circle, and therefore, its GMD is equal to 1. Also, but not quite as obvious, when the aspect ratio goes to zero, the ellipse becomes a line. But unlike the case of an elliptical area where the point density never becomes linear, in the case of an elliptical locus, it does indeed become linear, and we can use the self GMD of a line which is  $e^{-3/2} = 0.22313$  times its length. With a semi-major axis of 1, the ellipse has a length of 2, and the GMD in this limiting case becomes  $2 \times 0.22313 = 0.44626$ .

Plotting the data from the table along with the two limiting values, we get the following graph:



Despite the ordeal of obtaining the uniformly distributed random points, we see that the final result is a nearly straight line.

It turns out that, while not a perfectly straight line, these points can be fit very well with a parabolic curve. Doing a least squares fit, the optimized formula for self GMD of an elliptical line is:

$$g_s = [2e^{-\frac{3}{2}} + 0.61578u - 0.061591u^2]a \quad . \quad (3.4.5)$$

where  $u = b/a$  (the aspect ratio).

The first term is analytical, as previously discussed, while the last two are empirical. The similarity between the two empirical coefficients (and considering that  $0.615 \approx \frac{8}{13}$ ) suggests a simpler and easier to remember formula:

$$g_s = \left[ 2e^{-\frac{3}{2}} + \frac{8}{13} \left( u - \frac{u^2}{10} \right) \right] a \quad . \quad (3.4.6)$$

This simplification gives an insignificant increase, from 0.0000448 to 0.0000453, in the total squared error, but notably, a small decrease in peak error, from 0.407% down to 0.402%. Hence, this simpler formula is recommended.

There may not be much demand for the GMD of an elliptical line<sup>4</sup>. If there was, then either Grover, or one of his contemporaries likely would have come up with an expression for it six decades ago. However, the main point is that if we do need to find the GMD of some shape that has not been previously analyzed, then we have the means to do it. An elliptical line is a good example of one of the more complicated cases. Yet, we are able to analyze it, and come up with a practical expression for it.

**Appendix A** gives the BASIC program listing for the Monte Carlo calculation of GMD of the elliptical lines discussed here. It consists of a small main program which implements the main part of the Monte Carlo simulation, and a subprogram which implements the mapping function. The sets of mapping function coefficients for different aspect ratios are readily apparent in the program code. It should be noted that this program runs much slower than the program for finding the GMD of elliptical areas. This is due to the mapping function which adds a significant number of mathematical operations inside the main calculation loop. Even though the earlier program used sample rejection to obtain the uniform distribution and therefore rejected 21% of the random points generated, the remaining operations were very few and very efficient.

As previously mentioned, the determination of the constants for the mapping function was done using a spreadsheet and a solver macro. The final fitting of the GMD data produced by the BASIC Monte Carlo program was also done using a spreadsheet and solver macro.

---

<sup>4</sup> In fact, when calculating the inductance of a coil constructed of round thin wall tubing, and having large pitch, the cross section of the conductor, taken in a plane that passes through the coil axis, will be an elliptical line. The GMD of an elliptical line then becomes useful in the analysis of the coil.

## Part 4 - Limits of Accuracy of the GMD method

### 4.1 Self Inductance

It was stated earlier that the GMD principle can be applied to conductor geometries other than straight lines as long as the conductor diameter is small relative to the overall conductor length, and the radius of curvature is large compared to the cross sectional diameter of the conductor. That is a somewhat vague statement, and deserves further discussion. Unfortunately, any research done on the limits of accuracy of the application of GMD in inductance calculations, appears to be non-existent.

In this section, inductance values calculated using the GMD method will be compared with those calculated by other methods, in order to determine the point at which the accuracy deteriorates. The simplest example for examining the effect of conductor curvature is the inductance of a circular loop of a conductor of finite diameter. We will look at what happens as the ratio of conductor diameter to loop diameter is changed. The GMD method is straightforward: we calculate the self GMD of the conductor, and then use that as the axial separation of two filamental loops of the same diameter, and plug these numbers into Maxwell's loop mutual inductance formula [1b]. This must then be compared with a method which does not make the GMD simplification. One such formula has been given by Max Wien [10], where he integrated the mutual inductance formula twice across the cross section of a round conductor to get a supposedly exact formula. However, his formula was later found to have errors in one of the terms. In addition, he discarded high order terms, which renders the formula unsuitable for this study. Other formulae followed Wien's. However, these too contained simplifications where high order terms were discarded, again making them unsuitable for use here.

The exact method is to integrate Maxwell's formula across the cross section of the conductor. Because the integrand is an elliptic integral, this must be carried out using numerical integration techniques. This poses a bit of a problem, because with numerical integration, we normally evaluate and sum a series of rectangular areas or, in the case of a circular area, circular segments. Both of these methods are rather cumbersome. We still need to calculate the mutual inductance of every possible combination of pairs of these oddly oriented and/or oddly shaped areas, and end up having to use the GMD between these areas for lack of any better method. Further complicating matters, there is no formula for the GMD of the oddly shaped pieces in all their myriad orientations, necessitating more approximations. We would expect that as the cross sectional area is divided into smaller and smaller parts, that the accuracy would improve. Yet, there is no guarantee that some systematic error will not manifest itself in the mutual inductance between very closely spaced pairs of areas, or especially the self inductance of an area. Nevertheless, this method was initially tried, and seemed to give reasonable results despite being computationally inefficient.

On the other hand, by adapting the Monte Carlo method, we replace the potential systematic error for random error. But at least the random error should have no particular bias as long as the random numbers themselves have no bias. The other advantage is that it is much simpler to code,

and is computationally much more efficient. Consequently, the Monte Carlo method is used in the following discussion.

The Monte Carlo method assumes that the conductor is composed of an infinite number of infinitesimal filaments. A representative sample of random pairs of these is taken, their mutual inductance is calculated, and the mean is taken. By taking a larger and larger sample we can reduce the random error to an acceptable value. Following is the BASIC program code which implements the Monte Carlo procedure.

```
Function McL(Rc As Double, dw As Double, yOffset As Double, N As Int64) As double
' Calculate Self/Mutual Inductance of Loop Conductor by Monte Carlo method
' Rc = Loop Radius, dw = Conductor diameter
' yOffset = axial loop offset, N = Number of random point pairs
dim i As Int64
dim tx,ty,tye,tyel,sx,sy,sye,syel,r1,r2,rw,sum,axial As Double
rw = dw/2
sum = 0
For i = 1 to N
'Generate two random test points S and T
do
tx = (rnd-0.5)*dw
ty = (rnd-0.5)*dw
tye = sqrt(rw^2-tx^2)
tyel = -tye
loop until (ty< = tye and ty> = tyel)
do
sx = (rnd-0.5)*dw
sy = (rnd-0.5)*dw
sye = sqrt(rw^2-sx^2)
syel = -sye
Loop Until (sy< = sye and sy> = syel)
'Add offsets to random points to get radius and axial displacement
axial = abs(yOffset+sx-tx)
r1 = Rc+ty
r2 = Rc+sy
sum = sum+Mut1(r1,r2,axial)
next
return sum/N
End Function
```

```
Function Mut(ByVal r1 as double,r2 as double,x as Double) As Double
' Maxwell's elliptic integral formula to calculate mutual inductance
' between two coaxial circular loops.
' Uses AGM method to calculate elliptic integrals
' r1,r2 are radii of the respective loops
' x is the axial distance separating them
muo = pi()*4e-7
a = sqr((r1+r2)^2+x^2)
b = sqr((r1-r2)^2+x^2)
m = 4*(r1*r2)/((r1+r2)^2+x^2)
c = a-b
ci = 1
cs = c*c
do
```

```

ao = (a+b)/2
b = sqr(a*b)
a = ao
co = c
c = a-b
ci = 2*ci
cs = cs+ci*c*c
loop while c<co
Mut = muo/8*pi()*cs/a
end Function

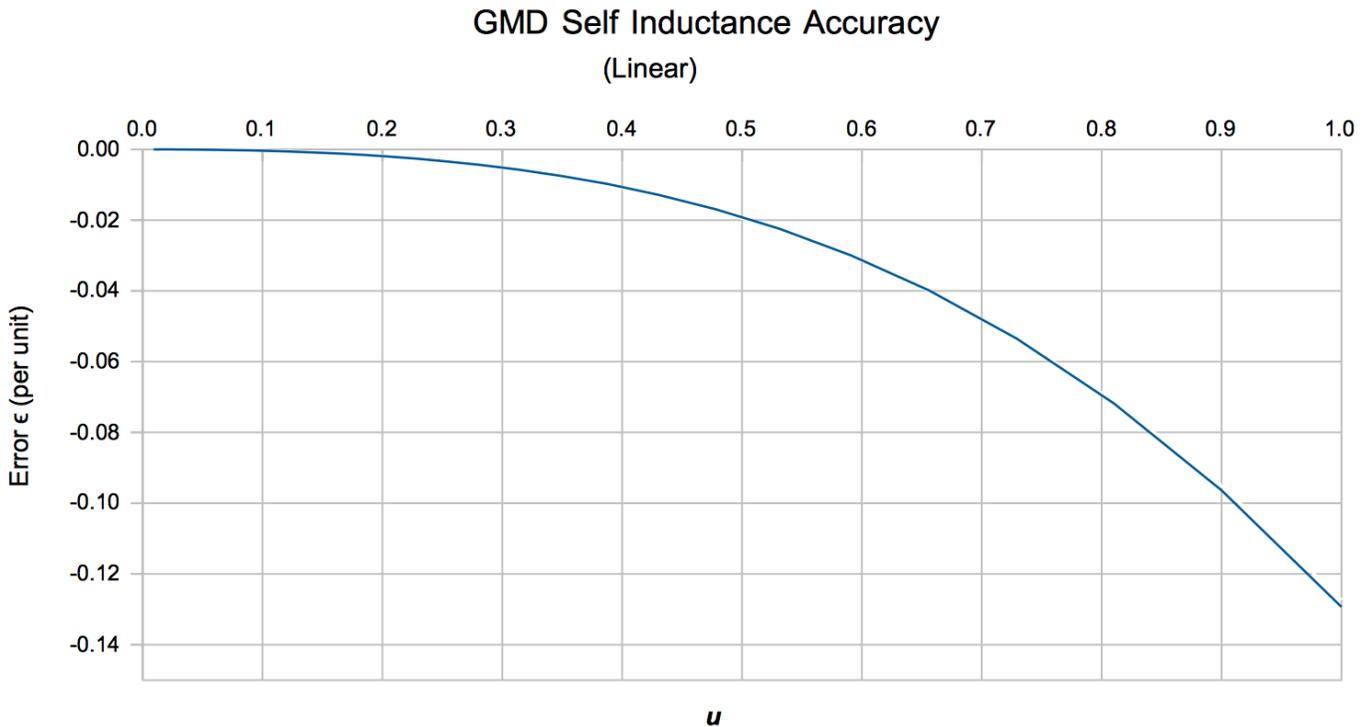
```

The results of the program are plotted in the following graph which shows the relative error of the GMD inductance value, compared to the Monte Carlo inductance value, as the ratio of conductor diameter to loop radius is varied. Note that the largest possible ratio  $d/D$  is 1. We will define the variable  $u$  as the ratio  $d/D$  in the following discussion. We will use the variable  $\epsilon$  as the relative error, and define it as:

$$\epsilon = L_{GMD}/L_{MC} - 1 \quad (4.1.1)$$

where  $L_{GMD}$  is the inductance value calculated by the GMD method, and  $L_{MC}$  is the inductance calculated by the Monte Carlo method.

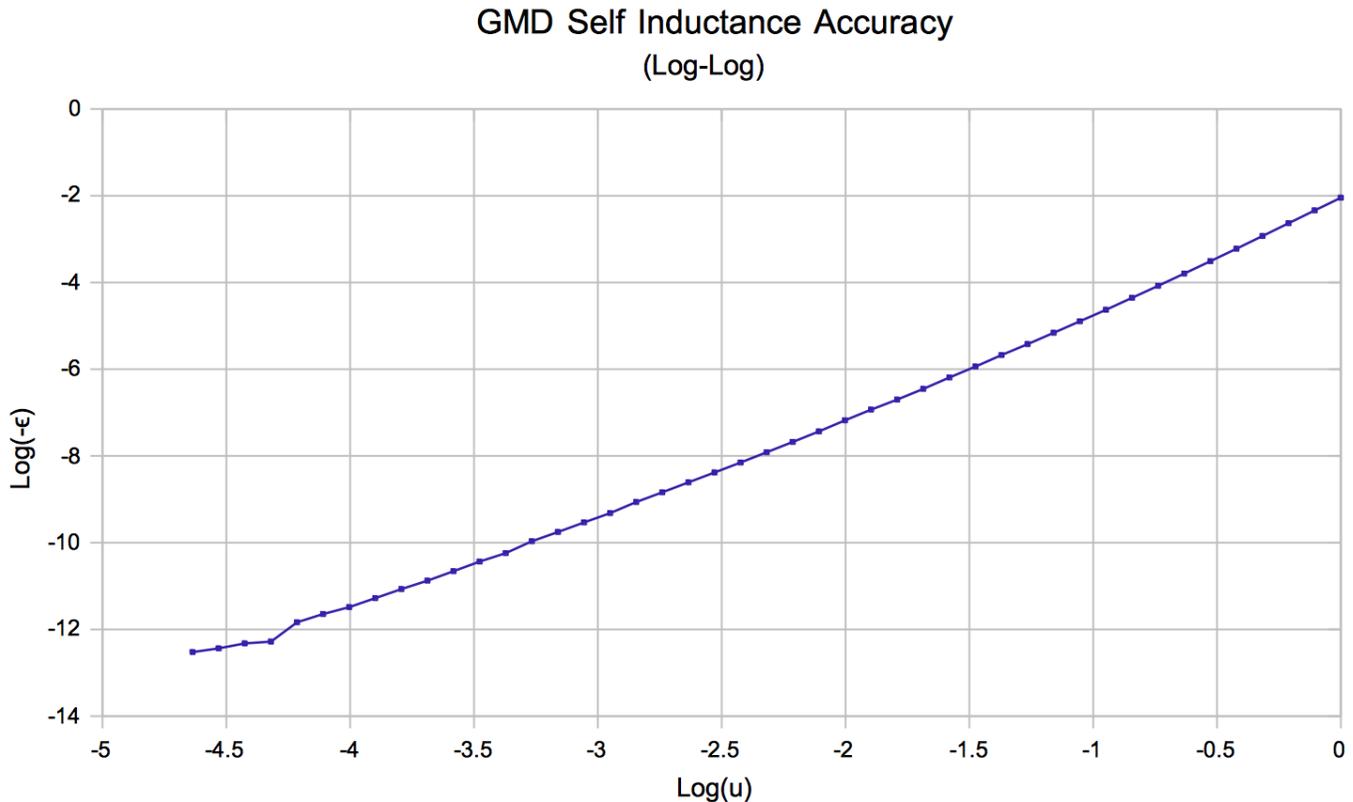
By this definition, a positive value for  $\epsilon$  means that  $L_{GMD}$  is higher than the the "true" inductance value.



Surprisingly, the error is only  $-0.1\%$  ( $-0.001$  per unit, or  $-1000$  ppm) when the wire diameter is as much as 0.15 times the loop diameter, which indicates that the GMD approach to inductance calculation is an excellent approximation for practical values of  $u$ . Even in the worst case value of

$u = 1.0$ , the error is  $-13\%$  ( $-0.13$  per unit), which is surprisingly good when one considers the puffy bagel-like torus in such a situation. Also note that the error is always negative, indicating that the GMD method will always yield a self inductance value that is slightly lower than the true value.

If we do a log-log plot<sup>5</sup> of the data we get a much straighter line, but still slightly curved. So, there is no simple power relationship:



The inherent randomness in the Monte Carlo method makes the line noisy for values of  $u$  less than  $0.02$  (i.e.,  $\log(u) < -4$ ), which is noticeable at the left side of the graph, but this is of little consequence because the error at this point is only a few parts/million, and is heading towards zero.

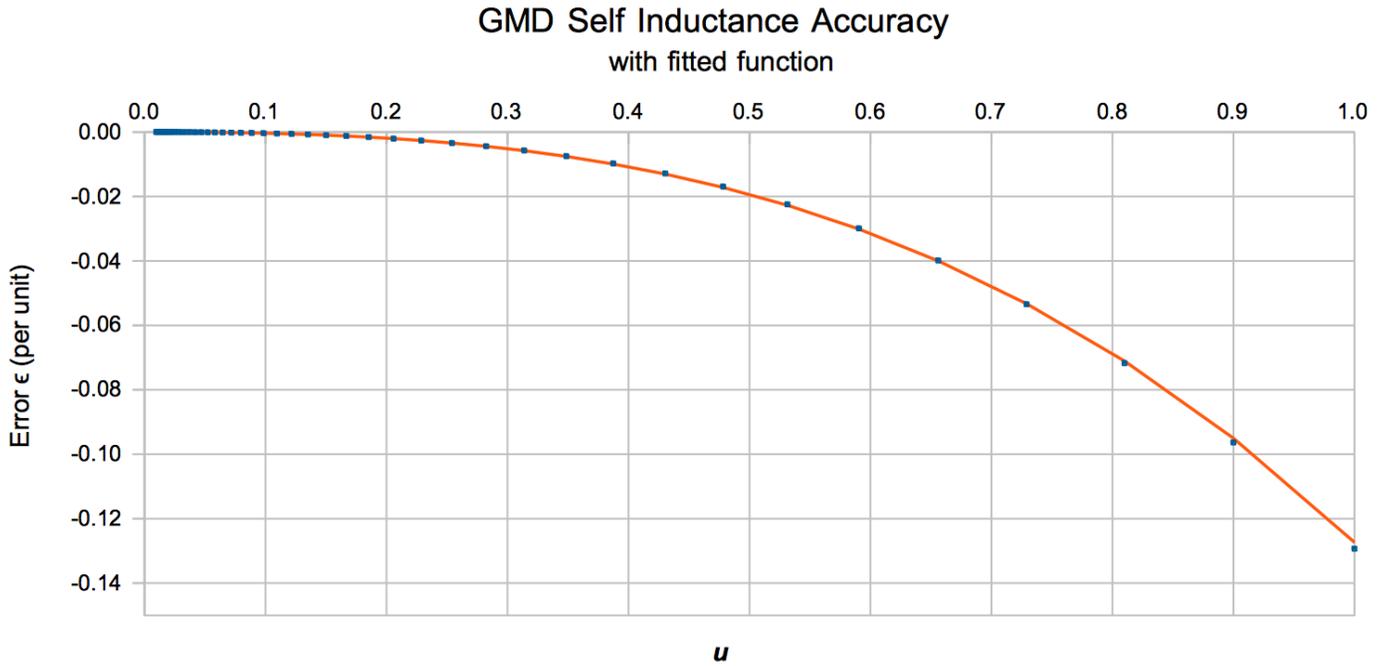
It turns out that these data can be fit very well with a third degree polynomial. Solving for the best fit coefficients for the cubic regression line, the equation for the error is:

$$\epsilon = -2.9945 \times 10^{-5} u - 0.027273 u^2 - 0.10003 u^3 \tag{4.1.2}$$

---

<sup>5</sup> Rather than plot the data on a logarithmically scaled graph as is traditionally done, we simply plot the natural logarithms of the data on a linearly scaled graph. The resulting line has the same curvature in either case.

Below are both linear and log graphs showing the fitted function (in red) along with the original data points.

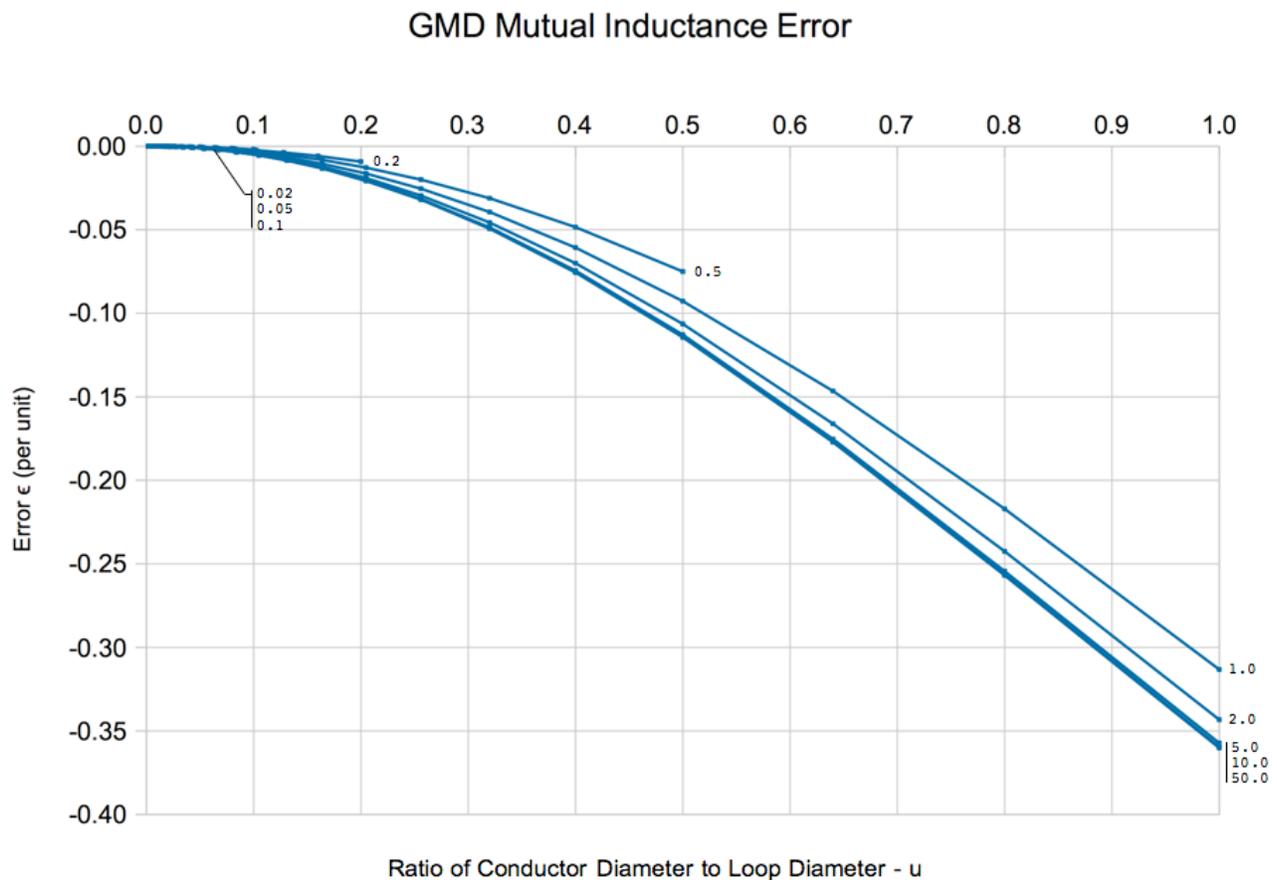


Note that there is no zero order (constant) term. It must be zero to ensure that the error value properly goes to zero as  $u$  goes to zero.

This error function could be used as a correction factor applied to GMD calculated self-inductance values, if extreme accuracy is needed in theoretical calculations. However, in real world coil production, the normal variations in coil construction would result in a much larger inductance variation than the magnitude of this error value.

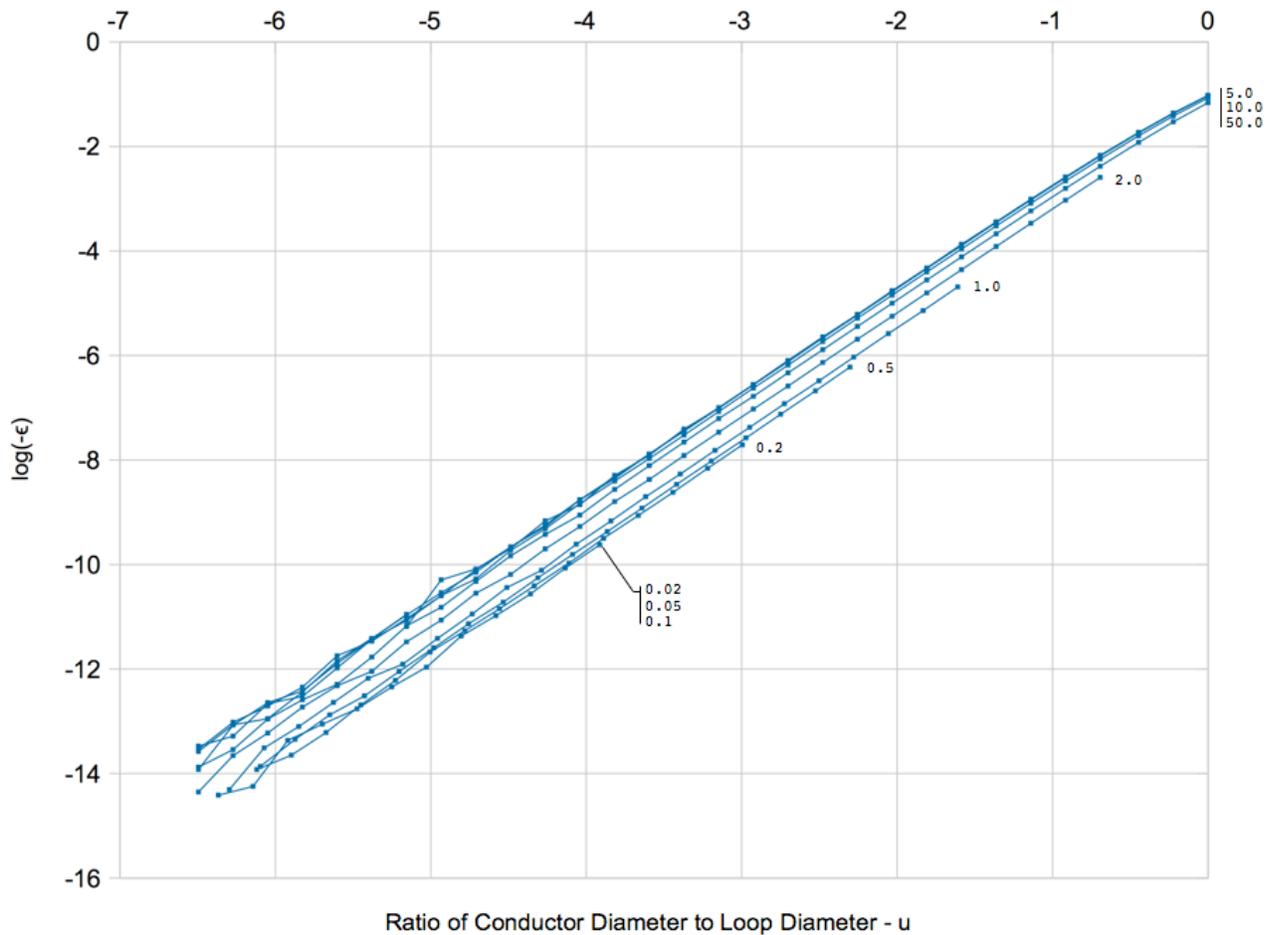
## 4.2 Mutual Inductance

Having looked at the limits of accuracy for loop self inductance, we will now look at what happens to the mutual inductance of a co-axial pair of wire loops as  $u$  is varied, and as the axial separation is varied. The same program is used, but the  $yOffset$  variable is now set to values other than zero. In this investigation, we will normalize the the axial offset as multiples of the loop diameter, so that the results may be more easily applied to other configurations. Hence, the variable  $x$  will be defined as the ratio of axial spacing to loop diameter. The following graph shows the set of curves generated by the program, each curve representing a different value of axial offset ratio  $x$ , and with  $u$  varying along the horizontal axis. Each curve is labeled with the corresponding value of  $x$ .



Note that for each curve, the maximum possible value for  $u$  is the smaller of  $x$  and 1.

Re-plotting as a log-log graph shows (below), a series of straight lines.



The log-log plot emphasizes the random scatter in the Monte Carlo calculation at very small values of  $u$ . This is a result of subtracting pairs of nearly equal values, differing only in the last couple of significant digits, causing the Monte Carlo random error to be accentuated. But again, these correspond to only a few parts/million of actual GMD error and are of minor consequence.

The curves lie very close together, indicating that the error is much more sensitive to variations of  $u$  than of  $x$ . It is also interesting to note that for very small values of  $x$ , the curves start to bunch together, and the same thing happens, for large values of  $x$ . This implies that values of  $x$  outside the range examined here, will produce error values that do not lie significantly outside the range of the curves shown.

Because the lines are straight (ignoring the noise) on the log-log plot, they can be fit by the log of the straight line formula:

$$\log(\epsilon) = a + m \log(u) \tag{4.2.1}$$

or taking the exponential function of both sides:

$$\epsilon = e^a v^m \tag{4.2.2}$$

Since  $e^a$  is a constant, we will define a new constant  $b = e^a$ , and hence:

$$\epsilon = b v^m \tag{4.2.3}$$

By performing a regression analysis on each line, we get a set of y-intercept ( $b$ ) and slope ( $m$ ) values as shown in the table at the right.

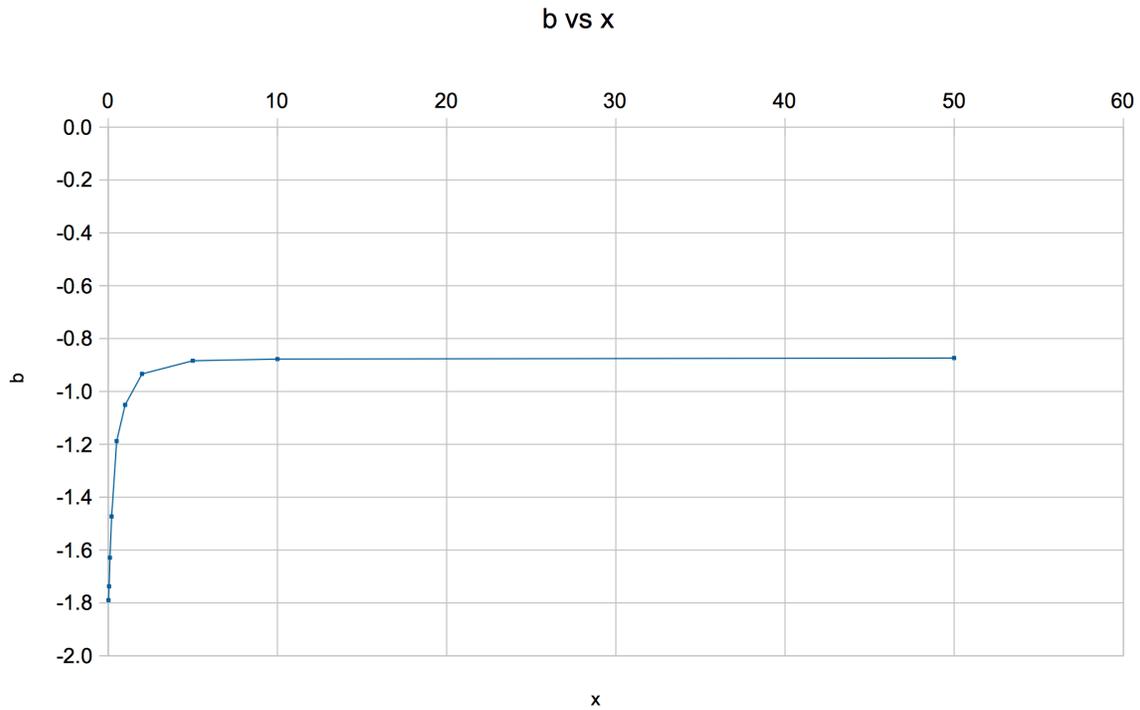
$x$	$b$	$m$
0.02	-1.65984	2.01249
0.05	-1.59745	2.03071
0.1	-1.59625	2.00430
0.2	-1.36547	2.02736
0.5	-1.17197	2.00648
1.0	-1.09616	1.94528
2.0	-0.99331	1.92341
5.0	-0.94899	1.91409
10.0	-0.93911	1.91810
50.0	-0.93638	1.91813

The slope is very close to 2 in every case. It drops very slightly where the extreme cases  $u = 1$  can be included, suggesting that the value is relatively constant at 2, but then starts to change in the region where  $u$  approaches the value 1. In fact, this can be seen at the far right side of the log-log graph. Where  $u$  approaches 1, the lines begin to curve slightly.

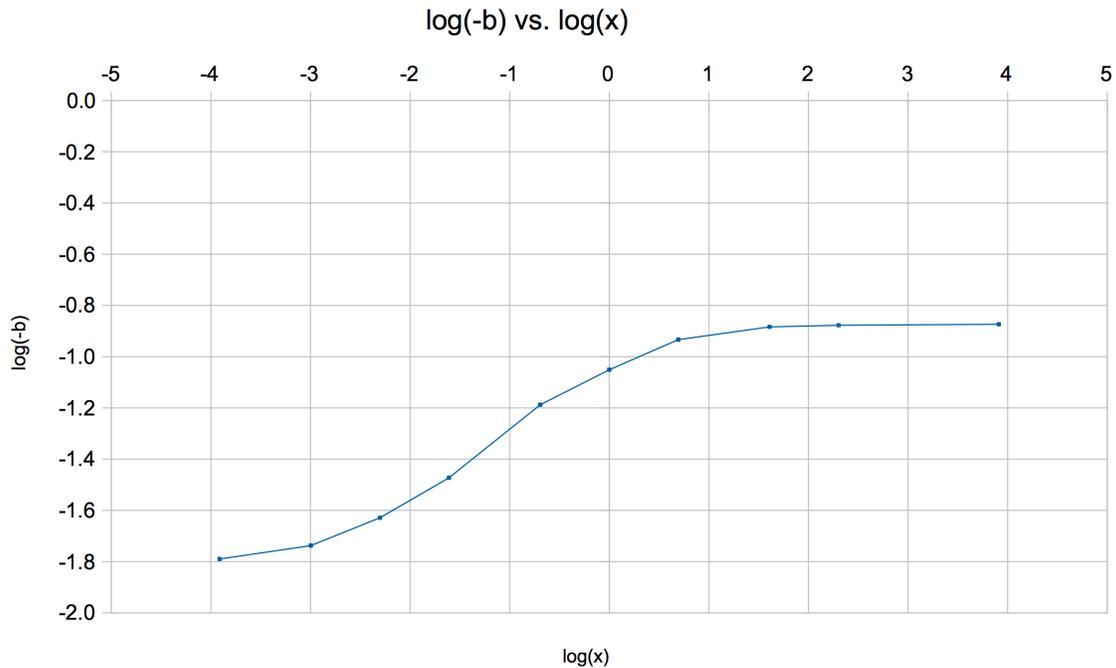
This evidence seems to be sufficiently compelling to conclude that the principal term in the error function is second order (i.e., proportional to  $u^2$ ). Hence, the GMD inductance approximation is correct to the first order, but does not fully account for second order effects, and that higher order error effects are almost negligible, except when  $u$  approaches 1. Consequently, we will presume the slope to be exactly 2, and we will then recalculate the intercept  $b$  values for best fit. This recalculation is shown in the next table to the right.

$x$	$b$	$m$
0.02	-1.79005	2.00000
0.05	-1.73758	2.00000
0.1	-1.62895	2.00000
0.2	-1.47300	2.00000
0.5	-1.18791	2.00000
1	-1.05116	2.00000
2	-0.93378	2.00000
5	-0.88396	2.00000
10	-0.87750	2.00000
50	-0.87378	2.00000

If we plot these  $b$  values against  $x$ , we see an extremely nonlinear relationship:

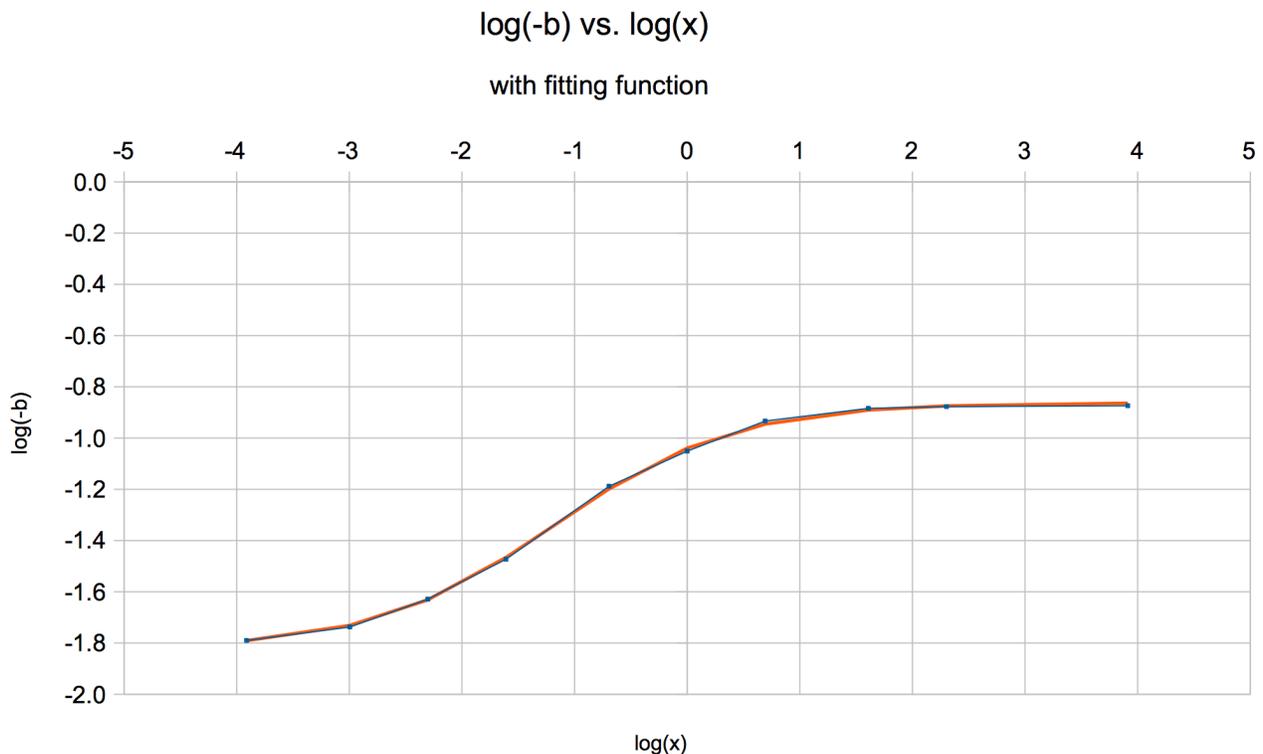


Given the hyperbola-like shape of the curve, it seems plausible that changing the ordinate of the graph from  $x$  to  $1/x$  might straighten it out. However, it turns out that it makes virtually no improvement. On the other hand, a log-log plot is more enlightening:



This is a sigmoid (S-shaped) curve with an inflection point near  $\log(x) = 0$ . The inflection point deserves some discussion. It occurs where the distance separating the loops is approximately equal to the loop diameter—a square aspect ratio. It signifies a point where one factor becomes less influential to the overall behaviour, and another factor becomes more influential. In this case, when the loops are close together, the largest influence on the mutual inductance on a point on one loop is due to the parallel current flowing in the nearest point of the other loop. This is much more significant than any other factor. As the spacing reaches the inflection point, the influence of the anti-parallel currents from diametrically opposed points on the loops become just as influential as the parallel currents, and then remain that way as the separation increases further.

Considering that the mutual inductance error curves are a function of two variables, with one of them being very nonlinear, it was questionable whether it would be feasible to try to fit an empirical error function. Nevertheless, it seemed to be worthwhile to expend some effort to find a fitting function. Given that each curve is primarily a simple square law function, the main effort then would be to find a fitting function for a sigmoid curve that predicts the  $b$  coefficients based on the value of  $x$ . There are numerous functions which have this shape, including the arctangent function and the hyperbolic tangent function, as well as some algebraic functions and some functions involving exponentials. Testing several of them, it was found that the hyperbolic tangent function fits these data very well. Since tangent functions are related to aspect ratios, perhaps this should not be too surprising. Overlaying the fitted hyperbolic tangent function (red), we get the following graph:



As can be seen, it fits the data quite well. However, some caution is required. This is a fitted function of the log of the data. We must then take the antilog of this function to fit the original data, and doing so can exacerbate any errors in the fitting function.

The function and optimized coefficients are as follows:

$$\log(-b) = k_0 + k_1 \tanh(k_2 + k_3 \log(x)) \quad (4.2.4)$$

$$k_0 = -1.34177$$

$$k_1 = 0.47936$$

$$k_2 = 0.74321$$

$$k_3 = 0.62668$$

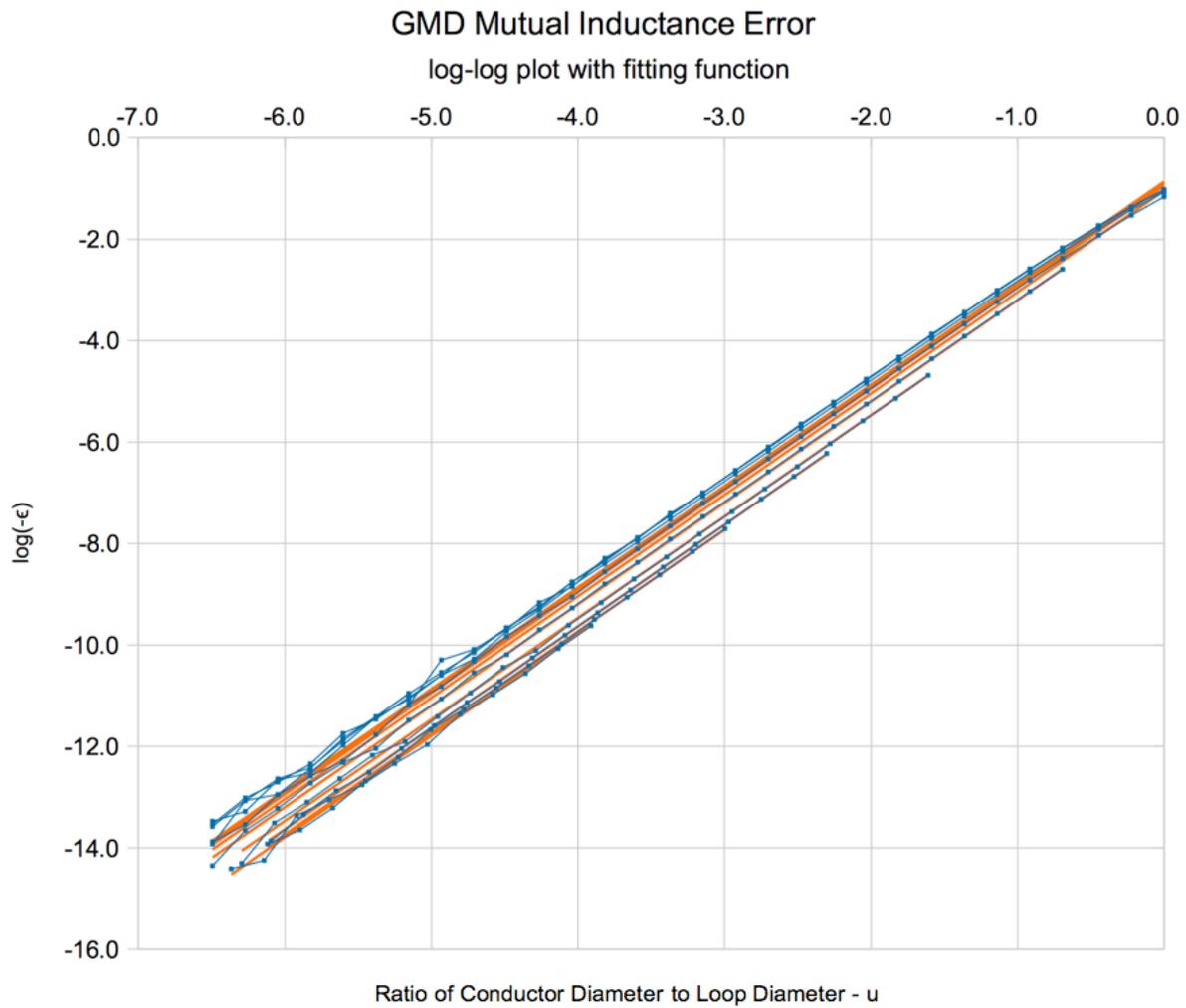
Applying the exponential function to both sides of the equation, and negating, we get:

$$b = -\exp \left[ k_0 + k_1 \tanh (k_2 + k_3 \log (x)) \right] \quad (4.2.5)$$

Having developed a relationship for the  $b$  values as a function of  $x$ , we now have the basis of a function that will predict the GMD error for co-axial loop mutual inductance. Substituting the  $b$  function into the  $\epsilon$  function we get:

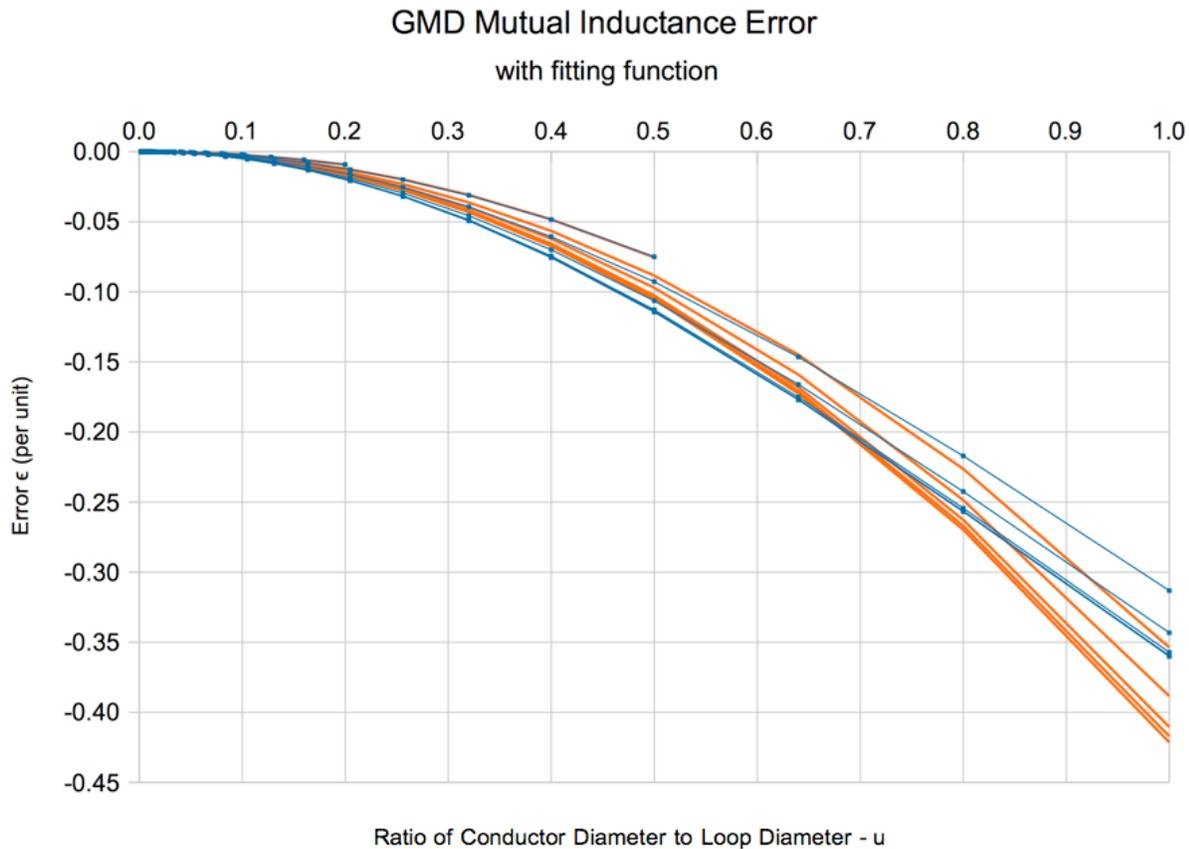
$$\epsilon = -\exp \left[ k_0 + k_1 \tanh (k_2 + k_3 \log (x)) \right] u^2 \quad (4.2.6)$$

Below is the log-log error graph with the predicted error function overlaid:



With the lines grouped so closely together it is not easy to see much beyond the fact that they do seem to line up. It can be seen that as  $\log(u)$  approaches 0 (i.e.,  $u$  approaches 1), the error curves start to bend away from a straight line, as was mentioned earlier.

Re-plotting on a linear graph, we get the following:



The correlation for  $0 < u \leq 0.5$  is reasonable. However, the region of  $0.5 < u \leq 1$  leaves much to be desired. While this range of geometry is an extreme that is unlikely to be encountered in practice, it's worth considering the inclusion of a correction term that comes into effect for  $u > 0.5$ . The correction may take either of two forms, an additive correction term, or a multiplying factor. After some experimentation, the additive term appeared to be the most promising, and that is the approach that was taken. We therefore want a term which is asymptotic to zero when  $u < 0.5$ , and then smoothly bends away from zero when  $u > 0.5$ . As was used in the development of the mutual GMD formula for elliptical areas, a rational function again appears to be suitable. A simple function that accomplishes this is:

$$k_4 \frac{u^n}{u^m + k_5}$$

The exponents  $n$  and  $m$  determine the function's behaviour when  $u$  is large. if we set  $n = m+1$ , then for large  $u$ , the function will asymptotically approach a straight line with a slope equal to  $k_4$ . The constant  $k_5$  determines the crossover point between the zero asymptote and the constant slope asymptote. Though it is proposed to set  $n$  to be one greater than  $m$ , there is nothing to suggest that

the difference cannot be greater, especially since the function is limited to a finite range of values for  $u$ , eliminating the risk of it blowing up for large values of  $u$ . It's always best to start with the simplest function, and if it's found to be unsuitable, then it can be further adjusted. With some experimentation, it was found that the optimum values for  $m$  and  $n$  are 3 and 4 respectively. Hence, the correction term is:

$$k_4 \frac{u^4}{u^3 + k_5}$$

The optimum values of  $k_4$  and  $k_5$  will be given later.

While this correction term was a good starting point and gave reasonable improvement in the fit, it was found not to be sufficient to adequately correct the empirical function. It became clear that the correction term needs to be a function of both  $u$  and  $x$ . While  $u$  is confined to the range of values from 0 to 1,  $x$  can take on any value from zero to infinity, which adds some complication to finding a function that will be well behaved for any  $x$ . As previously mentioned, for values of  $x$  greater than about 5, the error curves begin to bunch together. In fact there is no discernible difference between the curve for  $x = 10$  and for  $x = 50$ . The correction term involving  $x$  must not disturb this behaviour. Therefore, this correction factor must approach a fixed value asymptotically as  $x$  becomes large. Again a rational function is chosen. In this case:

$$\frac{x^2 + k_6}{x^2 + 1}$$

This function has a value equal to  $k_6$  when  $x = 0$ , and then asymptotically approaches 1 as  $x$  becomes large. The product of the first and second correction factors then becomes the complete correction term:

$$k_4 \left( \frac{u^4}{u^3 + k_5} \right) \left( \frac{x^2 + k_6}{x^2 + 1} \right)$$

And, the complete GMD error function is then

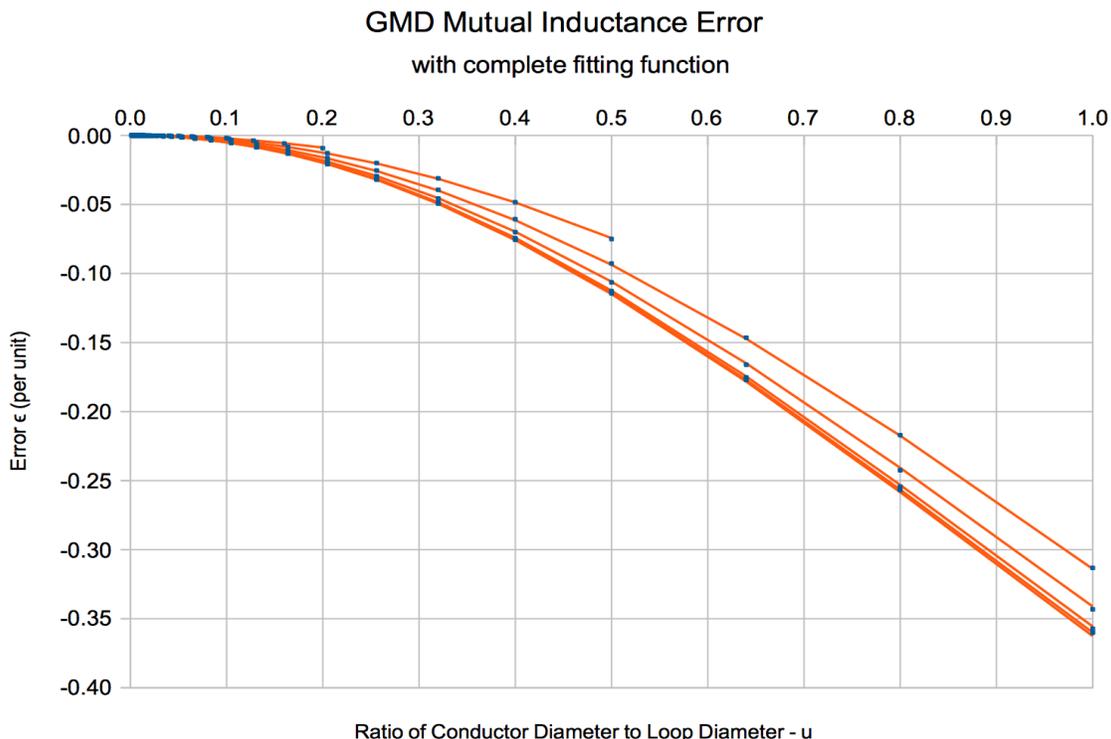
$$\epsilon = - \exp \left[ k_0 + k_1 \tanh (k_2 + k_3 \log (x)) \right] u^2 + k_4 \left( \frac{u^4}{u^3 + k_5} \right) \left( \frac{x^2 + k_6}{x^2 + 1} \right) \quad (4.2.7)$$

With the candidate function having been set, all constants (including  $k_0$ – $k_3$ ) are then re-optimized, yielding the following values:

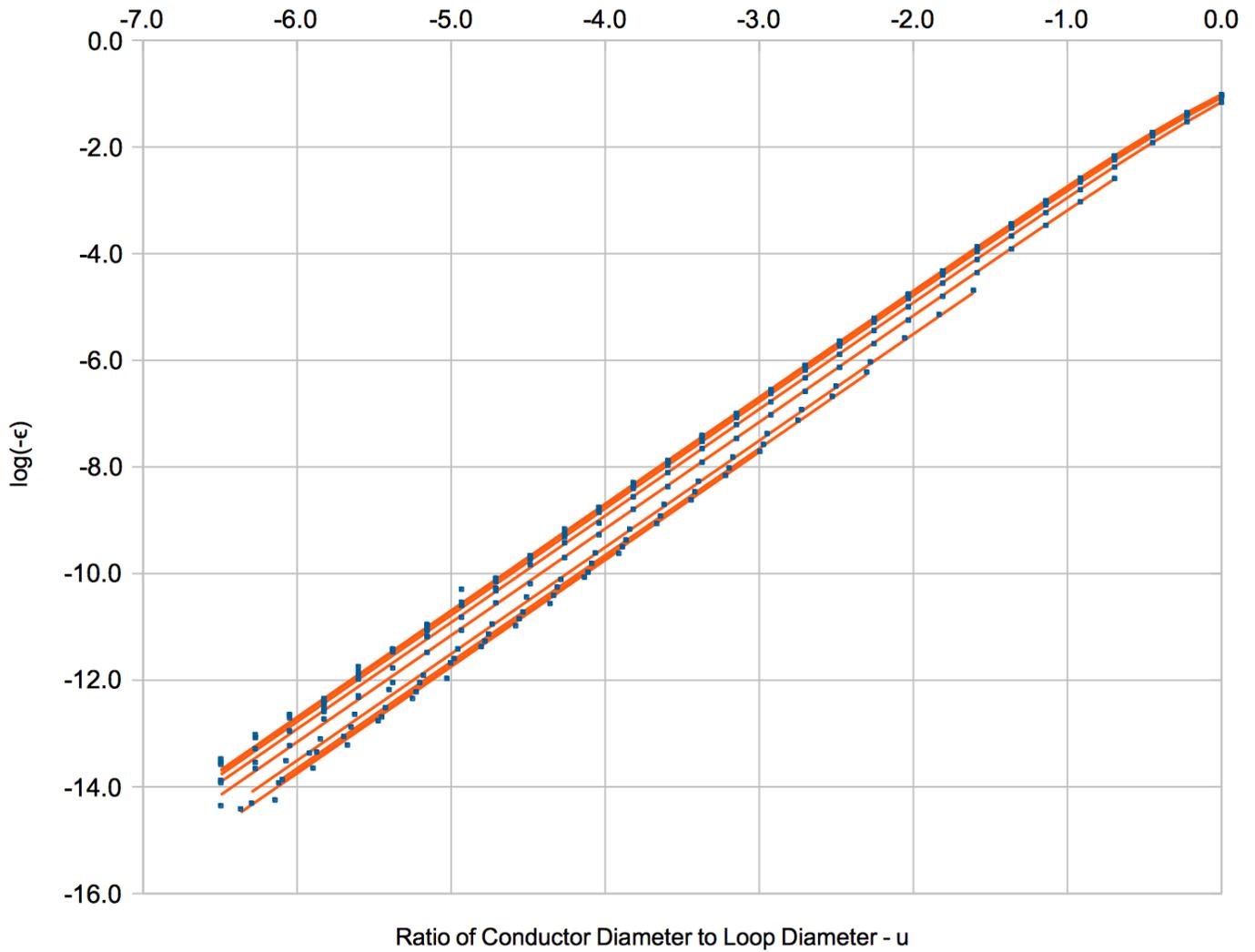
- $k_0 = -1.22339$
- $k_1 = 0.53125$
- $k_2 = 0.66240$
- $k_3 = 0.77879$
- $k_4 = 0.76927$
- $k_5 = 4.58965$
- $k_6 = 0.25572$

The optimization process required certain compromises to be made. Primarily, it was a question of whether to do a least squares fit based on absolute error or on relative error. Relative error produces a better fit for small values of  $u$ , while absolute error produces a better fit for large values of  $u$ . Normally, relative error would be the best choice, especially since the parameters are normalized so that they can be scaled for any application, and would then yield a consistent percentage error for any value. However, the large scatter of the points at small  $u$  resulted in an unsatisfactory fit. Ultimately, it was decided to use relative error as the fitting criterion, but with the data weighted by the square root of  $u$ . No statistical checks were done to verify the validity of this approach. However, this gave the most satisfactory fit based on a visible inspection of the graphs of the fitting function, when viewed at microscopic and macroscopic levels.

The following graphs show the results of the fitting process on both linear and log scales. The red curves are the fitting function, while the blue points are the data (Monte Carlo calculations) that were fitted.



GMD Mutual Inductance Error  
log-log plot with complete fitting function



While the final formula may seem slightly unwieldy, the fact that it is reasonably accurate over several decades of  $u$  and  $x$  values, the complexity is not excessive.

### 4.3 Example Calculation

At this point it is worthwhile to provide a worked example. We will show the total error in the GMD based inductance calculation of a solenoid coil of practical dimensions. For this example, the coil will have the following parameters:

$N = 30$  turns

$d = 0.5$  mm

$D = 25$  mm

Winding Pitch = 1 mm

Below is a clipping from a spreadsheet showing the inductance calculation using the summation method.

	A	B	C	D	E	F	G	H	I
1		<b>Wire Diameter</b>	0.0005 m						
2		<b>Coil Diameter</b>	0.0250 m						
3		<b>Pitch</b>	0.0010						
4		<b>Turns</b>	30						
5									
6	<b>Self Inductance Calculations</b>								
7	<b>Pair No</b>	<b>Pair Offset</b>	<b>Loop Count</b>	<b>Ls</b>	<b>Ls x Loops</b>	<b>x</b>	<b>u</b>	<b>P.U. Error</b>	<b>Abs Error</b>
8	0	0.0001947	30	0.06663	1.99887	0.000	0.020	-0.0000123	-0.000025
9	<b>Mutual Inductance Calculations</b>								
10	<b>Pair No</b>	<b>Pair Offset</b>	<b>Pair Count</b>	<b>Lm</b>	<b>Lm x Pairs</b>	<b>x</b>	<b>u</b>	<b>P.U. Error</b>	<b>Abs Error</b>
11	1	0.00100	58	0.04100	2.37814	0.040	0.020	-0.0000710	-0.000169
12	2	0.00200	56	0.03030	1.69699	0.080	0.020	-0.0000744	-0.000126
13	3	0.00300	54	0.02420	1.30688	0.120	0.020	-0.0000787	-0.000103
14	4	0.00400	52	0.02001	1.04058	0.160	0.020	-0.0000836	-0.000087
15	5	0.00500	50	0.01688	0.84421	0.200	0.020	-0.0000888	-0.000075
16	6	0.00600	48	0.01444	0.69307	0.240	0.020	-0.0000941	-0.000065
17	7	0.00700	46	0.01247	0.57356	0.280	0.020	-0.0000994	-0.000057
18	8	0.00800	44	0.01085	0.47732	0.320	0.020	-0.0001046	-0.000050
19	9	0.00900	42	0.00950	0.39883	0.360	0.020	-0.0001097	-0.000044
20	10	0.01000	40	0.00835	0.33417	0.400	0.020	-0.0001145	-0.000038
21	11	0.01100	38	0.00738	0.28052	0.440	0.020	-0.0001191	-0.000033
22	12	0.01200	36	0.00655	0.23575	0.480	0.020	-0.0001235	-0.000029
23	13	0.01300	34	0.00583	0.19820	0.520	0.020	-0.0001276	-0.000025
24	14	0.01400	32	0.00521	0.16659	0.560	0.020	-0.0001314	-0.000022
25	15	0.01500	30	0.00466	0.13989	0.600	0.020	-0.0001350	-0.000019
26	16	0.01600	28	0.00419	0.11728	0.640	0.020	-0.0001384	-0.000016
27	17	0.01700	26	0.00377	0.09807	0.680	0.020	-0.0001415	-0.000014
28	18	0.01800	24	0.00341	0.08174	0.720	0.020	-0.0001444	-0.000012
29	19	0.01900	22	0.00308	0.06781	0.760	0.020	-0.0001472	-0.000010
30	20	0.02000	20	0.00280	0.05592	0.800	0.020	-0.0001497	-0.000008
31	21	0.02100	18	0.00254	0.04576	0.840	0.020	-0.0001521	-0.000007
32	22	0.02200	16	0.00232	0.03706	0.880	0.020	-0.0001543	-0.000006
33	23	0.02300	14	0.00211	0.02961	0.920	0.020	-0.0001564	-0.000005
34	24	0.02400	12	0.00193	0.02322	0.960	0.020	-0.0001583	-0.000004
35	25	0.02500	10	0.00177	0.01773	1.000	0.020	-0.0001601	-0.000003
36	26	0.02600	8	0.00163	0.01303	1.040	0.020	-0.0001618	-0.000002
37	27	0.02700	6	0.00150	0.00899	1.080	0.020	-0.0001634	-0.000001
38	28	0.02800	4	0.00138	0.00552	1.120	0.020	-0.0001649	-0.000001
39	29	0.02900	2	0.00127	0.00255	1.160	0.020	-0.0001664	0.000000
40				<b>L sum</b>	<b>13.36784</b>	<b>μH</b>		<b>L error</b>	<b>-0.00106</b>
41									<b>μH</b>
42								<b>L adjusted</b>	<b>13.36890</b>
43									<b>μH</b>

The self inductance of each turn is calculated on row 8 by setting the pair offset value to the self GMD of the wire cross section. The mutual inductance of every combination of pairs of turns is calculated on rows 11 through 39. All self and mutual inductance calculations are done using the

**Mut()** function previously described and with the results shown in column D in units of microhenries. The mutual inductance of each pair of turns from column D is multiplied by the number of occurrences of each pair. This total value is shown in column E. The total of column E from rows 8 through 39 gives the inductance of the complete coil and is shown in cell E40 as 13.36784  $\mu\text{H}$ . Columns F and G show the  $x$  and  $u$  values which are used to calculate the predicted error  $\epsilon$  (per unit) in column H. The absolute error in microhenries shown in column I is calculated by multiplying the value of self or mutual inductance from column E by  $\epsilon/(\epsilon+1)$ .

The total of column I is the amount, in microhenries, that the inductance is predicted to be in error. The total error value shown in cell I40 is  $-0.00106 \mu\text{H}$ . That is, the calculated inductance value is predicted to be low by  $-0.00106 \mu\text{H}$ . This amounts to only about 0.008% of the calculated inductance value, and is essentially insignificant. However, if we wish to adjust the calculated value to account for the predicted error, then we can subtract this error value from the calculated value to get an “adjusted” value which is shown in cell I42. What this example tells us is that the approximation involved in calculation by the GMD method is a very good one for coils of typical geometry.

#### 4.4 Summary

Having derived the error functions for both the self and mutual loop inductance GMD based calculations, these values can be used as part of an error analysis procedure when calculating the inductance of an entire coil using the summation method.

If one were so inclined, and suitably ambitious, this study could be expanded to include loop conductors having non circular cross sections.

## Part 5 - Combined GMD of Multiple Objects

### 5.1 Basic Principles

Maxwell [1a], in addition to providing formulae for the GMD of simple geometric shapes using integration as discussed above, also showed how to combine the GMD of these simple shapes to determine the GMD of more complex shapes.

If we can calculate the GMD between objects using any of the previously discussed methods, then we can also calculate the GMD between groups of objects simply by taking the individual GMDs and finding their overall geometric mean weighted by the areas of the objects. For example, if we have three areas  $A_1$ ,  $A_2$  and  $A_3$ , and if the GMD between  $A_1$  and  $A_2$  is  $g_{12}$ , the GMD between  $A_1$  and  $A_3$  is  $g_{13}$ , then the GMD between  $A_1$  and the group  $A_2$  and  $A_3$   $g_{1-23}$  is:

$$\log(g_{1-23}) = (A_1A_2\log(g_{12})+A_1A_3\log(g_{13})) / (A_1A_2+A_1A_3) \quad (5.1.1)$$

Area  $A_1$  is included in the formula in order to show how it applies to the weighted mean, but in this case it cancels out and disappears from the formula, leaving:

$$\log(g_{1-23}) = (A_2\log(g_{12})+A_3\log(g_{13})) / (A_2+A_3) \quad (5.1.2)$$

In many other cases, the areas of all objects will remain in the formula.

In cases where all of the figures have the same area, then all of the areas disappear from the formula, and only the number of objects remains, thus becoming a simple unweighted mean.

The most general case is for the GMD between two different groups of areas. If we designate the first group  $a$  consisting of  $m$  areas  $A_{ai}$  where  $i = 1..m$ , and the second group  $b$  of  $n$  areas  $A_{bj}$  where  $j = 1..n$ , then the GMD between the two groups is:

$$\log(g) = \frac{\sum_{i=1}^m \sum_{j=1}^n A_{ai}A_{bj} \log(g_{ij})}{\sum_{i=1}^m \sum_{j=1}^n A_{ai}A_{bj}} \quad (5.1.3)$$

Now, in the special case where group  $a$  is the same group of objects as group  $b$ , then the above formula will produce the self GMD of that group of objects. Thus, for the self GMD of a group of objects, the formula becomes:

$$\log(g_S) = \frac{\sum_{i=1}^n \sum_{j=1}^n A_iA_j \log(g_{ij})}{\sum_{i=1}^n \sum_{j=1}^n A_iA_j} \quad (5.1.4)$$

These formulae are sufficiently general to cover just about any case. From here, we can move on to some special cases which result in simpler formulae. Perhaps the most common of these is where all areas are identical. This eliminates the need for weighting by area, and so the two previous formulae 5.1.3 and 5.1.4 become:

$$\log(g) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \log(g_{ij}) \quad (5.1.5)$$

and

$$\log(g_S) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \log(g_{ij}) \quad (5.1.6)$$

Since the self GMD of each component is calculated differently than the GMD between different components, it makes practical sense to separate the self GMD terms from the other GMD terms, and rewrite the expression as follows:

$$\log(g_S) = \frac{1}{n^2} \left[ \sum_{i=1}^n \log(g_{ii}) + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \log(g_{ij}) \right] \quad (5.1.7)$$

The leftmost summation is for the self GMD of each component, and the double summation is for the GMD between different components. We will adopt the term *mutual GMD* for the GMD between different objects and designate it as  $g_M$ .

The formula may also be written as:

$$\log(g_S) = \frac{1}{n} \sum_{i=1}^n \log(g_{ii}) + \frac{1}{n^2 - n} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \log(g_{ij}) \quad (5.1.8)$$

because the first summation includes  $n$  terms and the second double summation includes  $n^2 - n$  terms.

## 5.2 Formula for the GMD of a Linear Array of Circular Conductors (LACC)

A linear array of conductors frequently occurs in electrical and electronic systems. For example, high current power distribution systems often employ multiple bus bars per phase because multiple small bars will have more surface area than single large bars, allowing for better cooling, and thus higher current carrying capacity than a large bar with the same total amount of copper. As another example, a linear array of circular conductors also appears as the cross section of a single layer solenoid coil. In this section, a closed form formula will be developed for the self GMD of a linear array of circular conductors. It will be assumed that the pitch (centre-centre spacing between conductors) is constant.

Formula (5.1.7) from the previous section will be the starting point. The leftmost summation in the formula is for the self GMD component. Formula (1.4.3) for the self GMD of a circular area has been given as:

$$g_S = r e^{-1/4}$$

or

$$\log(g_S) = \log(r) - 1/4$$

where  $r$  is the radius of the circular area, (i.e., the conductor). By multiplying the self GMD by the number of occurrences  $n$ , we have the first summation of the formula:

$$n (\log(r) - 1/4)$$

For the mutual GMD terms, the GMD is simply the centre to centre distance between the various pairs of conductors. The general formula indicates that we must perform  $n^2 - n$  calculations. However, this number can be reduced significantly. Many of the calculations are identical due to the fact that there are only  $n-1$  unique spacings between conductors in the array. Consequently, it is necessary only to do  $n-1$  calculations, and then multiply these results by the number of times they occur. If we designate the pitch as  $p$ , then the distance between any two conductors will be  $i \times p$ , where  $i$  is an integer between 1 and  $n-1$ . The number of occurrences of pairs of conductors with the various spacings is equal to  $n-i$ . However, we must double this number in order to include both  $g_{ij}$  and  $g_{ji}$  cases. Taking all of this into account, the mutual GMD part of the formula becomes:

$$\sum_{i=1}^{n-1} 2(n-i) \log(p \times i)$$

And the complete formula for the self GMD of the linear array becomes:

$$\log(g_S) = \frac{1}{n^2} \left[ n(\log(r) - 0.25) + \sum_{i=1}^{n-1} 2(n-i) \log(p \times i) \right] \quad (5.2.1)$$

This summation formula can be coded as a Basic program as follows:

```
Function NetGMD(ByVal rw as double, n as double, p as double)
' Return aggregate GMD of linear array of circular areas
' rw is radius of circle; n is number of circles in the array
' p is centre to centre spacing (pitch) of circles
  s = (Log(rw)-0.25)*n
  for i = 1 to n-1
    s = s+2*(n-i)*Log(p*i)
  next
  NetGMD = exp(s/(n*n))
End Function
```

While this is useful in a hard coded computer application (and will be useful later for testing), the goal in this section is to develop the summation formula into a closed form expression with all of the summations eliminated, thus eliminating the need for *For-Next* Loops.

The first term:

$$n(\log(r) - 0.25)$$

is already a closed form expression and requires no further treatment.

For the second term, it can be seen that since  $\log(p \times i) = \log(p) + \log(i)$ , and the summation can be split into two parts:

$$\log(g_S) = \frac{1}{n^2} \left[ n(\log(r) - 0.25) + \sum_{i=1}^{n-1} 2(n-i) \log(p) + \sum_{i=1}^{n-1} 2(n-i) \log(i) \right] \quad (5.2.2)$$

It is straightforward to show that because  $p$  is a constant for a given configuration, the leftmost summation can be simplified:

$$\sum_{i=1}^{n-1} 2(n-i) \log(p) = (n^2 - n) \log(p) \quad (5.2.3)$$

This leaves a single summation which is a function of only a single variable  $n$ :

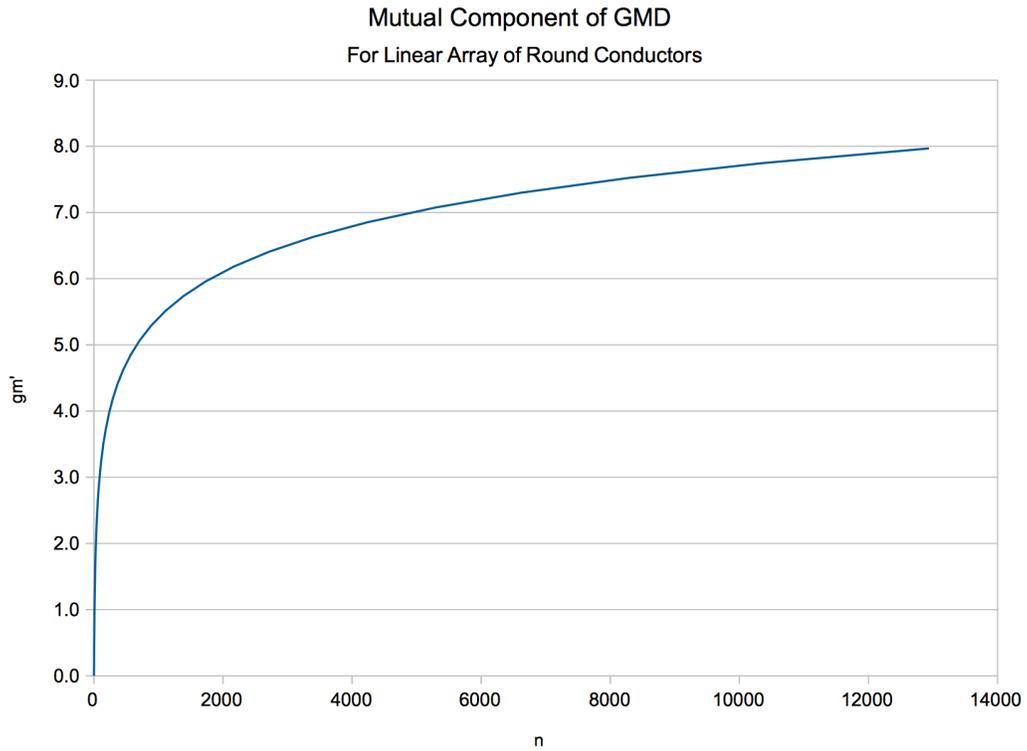
$$\log(g_S) = \frac{1}{n^2} \left[ n(\log(r) - 0.25) + (n^2 - n) \log(p) + \sum_{i=1}^{n-1} 2(n-i) \log(i) \right] \quad (5.2.4)$$

To simplify the remaining derivation, the terms inside the square brackets will be multiplied by the  $1/n^2$  factor:

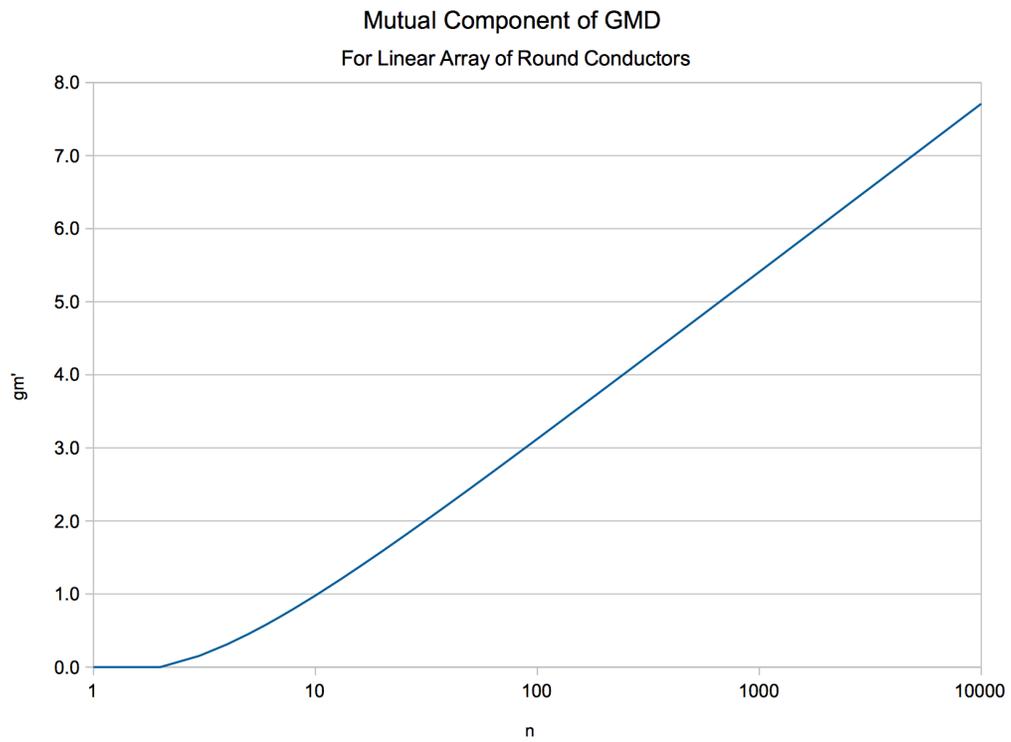
$$\log(g_S) = \frac{1}{n} (\log(r) - 0.25) + \left(1 - \frac{1}{n}\right) \log(p) + \frac{1}{n^2} \sum_{i=1}^{n-1} 2(n-i) \log(i) \quad (5.2.5)$$

We will designate the remaining summation term (including the  $1/n^2$  factor) as  $g_M'$  which refers to the mutual GMD component  $g_M$  excluding the pitch effect.

The following graph shows values of  $g_M'$  for  $n$  up to 13000.



A semi-log plot of the same function shows asymptotic behaviour:



For values of  $n > 10$  the function approximates a straight line on the semi-log graph. This indicates that for large  $n$ ,  $g_M'$  can be approximated by:

$$\exp(g_M') = k_0 n + k_1 \quad (5.2.5)$$

where  $k_0$  is the slope of the line and  $k_1$  is the intercept of the asymptote on the  $\exp(g_M')$  axis.

Analysis of the data shows the slope to be approximately 0.22313, which is readily explained. As  $n$  grows large, this term of the self GMD formula becomes dominant. At the same time, our linear array of circular areas begins to approximate a line of evenly spaced dots. Hence, the self GMD begins to approach that of a straight line which is  $e^{-3/2}$  times its length. Thus, the asymptote has a slope exactly equal to  $e^{-3/2}$  (0.223130160148...). Interestingly, the asymptote does not pass through the origin. The  $\exp(g_M')$  axis intercept can be determined from the data to be 0.410086, which gives a function that is approximately correct for large  $n$ :

$$\exp(g_M') = e^{-3/2} n + 0.410086 \quad (5.2.6)$$

The correct minimum value of the function is 0, occurring for both  $n = 1$  and  $n = 2$ . To account for the nonlinear behaviour when  $n < 10$ , it appears that a correction term of the following form can be included:

$$k_2 / (1 + k_3 n^m) \quad (5.2.7)$$

This works acceptably well, but if we consider that  $n$  is never less than one, it will turn out to be more convenient (and coincidentally, a better fit) to use this form instead:

$$k_2 / (1 + k_3 (n-1)^m) \quad (5.2.8)$$

The complete expression for  $g_M'$  is then:

$$\exp(g_M') = e^{-3/2} n + k_1 + k_2 / (1 + k_3 (n-1)^m) \quad (5.2.9)$$

In the limiting case when  $n = 1$ , the denominator of the correction term is 1 and the value of the correction term is equal to  $k_2$ . It then becomes a straightforward calculation to determine the value  $k_2$  so that the overall expression will produce the correct value of exactly 1 when  $n = 1$ . Hence:

$$k_2 = 1 - e^{-3/2} - k_1$$

Restricting the value of  $k_2$  in this way, and then optimizing only  $k_3$ , will result in no significant difference in overall error as compared to optimizing both  $k_2$  and  $k_3$ .

Choosing different values of the exponent  $m$  and adjusting the  $k_3$  value for the best fit, yields a best exponent value of  $m = 1$ , and optimum  $k_3 = 1.583635$ , resulting in this complete formula for  $g_M'$ :

$$\exp(g_M') = e^{-3/2} n + k_1 + (1 - e^{-3/2} - k_1)/(1 + k_3(n-1)) \quad (5.2.10)$$

or

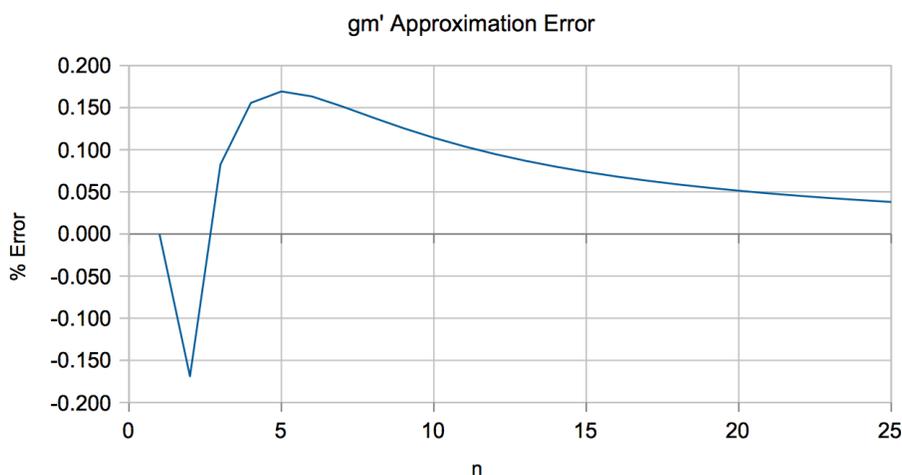
$$g_M' = \log( e^{-3/2} n + k_1 + (1 - e^{-3/2} - k_1)/(1 + k_3(n-1)) ) \quad (5.2.11)$$

where

$$k_1 = 0.410086$$

$$k_3 = 1.583635$$

This has a worst case error of 0.17% occurring at  $n = 2$  and  $n = 5$ , as shown in the following graph:



The contribution of this term to the overall GMD, at these values of  $n$ , results in about the same overall percentage error, i.e., 0.17%, but its contribution drops off rapidly as  $n$  increases, so that the overall error in GMD drops off much more rapidly than what is shown in the  $g_M'$  error graph.

We have now replaced all of the summation expressions in the GMD formula with simple closed form expressions. Combining all of the terms, we have:

$$\log(g_S) = \log \left( e^{-\frac{3}{2}} n + k_1 + \frac{1 - e^{-\frac{3}{2}} - k_1}{1 + k_3(n-1)} \right) + \left( 1 - \frac{1}{n} \right) \log(p) + \frac{1}{n} (\log(r) - 0.25) \quad (5.2.12)$$

Finally, taking the exponential function of both sides, we have the formula for the GMD of a linear array of circular conductors:

$$g_S = \exp \left[ \log \left( e^{-\frac{3}{2}} n + k_1 + \frac{1 - e^{-\frac{3}{2}} - k_1}{1 + k_3(n-1)} \right) + \left( 1 - \frac{1}{n} \right) \log(p) + \frac{1}{n} (\log(r) - 0.25) \right] \quad (5.2.13)$$

By applying various exponential and logarithmic identities, all of the log functions and the enclosing exponential function can be eliminated from the formula:

$$g_S = \left( e^{-\frac{3}{2}n} + k_1 + \frac{1 - e^{-\frac{3}{2}} - k_1}{1 + k_3(n-1)} \right) e^{\left(\frac{-1}{4n}\right)} p^{\left(1-\frac{1}{n}\right)} r^{\left(\frac{1}{n}\right)} \quad (5.2.14)$$

When  $n = 1$ , the empirical factor is also 1. Thus, in this form, it is easy to see how the formula transitions from  $r e^{-1/4}$  (the self GMD of a circular area) when  $n = 1$ , to  $n p e^{-3/2}$  (the self GMD of a line) when  $n$  is large.

The constant term,  $k_1 = 0.410086$ , inside the first factor, is significant and deserves some discussion. Because of its presence, the GMD of the linear array will always be slightly greater than the GMD of a straight line of the same overall length. The principal difference between the two geometries, is that a straight line comprises a continuous row of points of infinitesimal spacing. The linear array of circular areas has its points concentrated at intervals of  $p$ . It can be concluded that this difference accounts for the small difference in GMD between the two, which is equal to  $k_1 p$  when  $n$  is large. If the pitch is made very small, tending towards zero, corresponding to infinitesimal spacing between points, then this factor disappears, and the formula gives the same value as the formula for the GMD of a straight line. This coefficient is of sufficient importance that it should have a better name than  $k_1$ . We will refer to it as the *density uniformity* coefficient, and give it the symbol  $k_D$ . This will come up again very shortly. It should also be stated that even though we don't yet have a simple analytical expression for the value of  $k_D$ , this will be addressed later.

It is important to note that in formula (5.2.14), variables  $r$  and  $p$  must have the same units of length. The resulting value of  $g_S$  is in the same units as  $r$  and  $p$ .

Since the results are scaleable, it is possible to remove either  $p$  or  $r$  from the formula, and replace the remaining variable with the ratio of the two, which would then give a dimensionless result.

Let  $u$  be the ratio of conductor radius to pitch,  $r/p$ . Then:

$$r = u \times p$$

Looking at only the rightmost factors:

$$p^{\left(1-\frac{1}{n}\right)} r^{\left(\frac{1}{n}\right)}$$

and substituting  $u \times p$  for  $r$ :

$$p^{\left(1-\frac{1}{n}\right)} (u \times p)^{\left(\frac{1}{n}\right)}$$

Expanding the factors:

$$p^{\left(1-\frac{1}{n}\right)} p^{\left(\frac{1}{n}\right)} u^{\left(\frac{1}{n}\right)} \quad .$$

Combining the  $p$  factors:

$$p^{(1-\frac{1}{n}+\frac{1}{n})}u^{(\frac{1}{n})}$$

The exponent of  $p$  simplifies to one, and the complete GMD formula becomes:

$$g_S = \left( e^{-\frac{3}{2}}n + k_D + \frac{1 - e^{-\frac{3}{2}} - k_D}{1 + k_F(n - 1)} \right) e^{(\frac{-1}{4n})}u^{(\frac{1}{n})}p \quad (5.2.15)$$

The last remaining empirical constant  $k_3$  has been renamed  $k_F$  (for *fitting* constant). The values of these constants are as previously given:

$$k_D = 0.410086$$

$$k_F = 1.583635$$

Pitch can now be removed from the formula, if desired, leaving a dimensionless formula, where  $g_S$  is now expressed in terms (i.e., multiples) of pitch  $p$ .

We are not quite finished with this formula. It is accurate as shown, but as we continue through the next sections, it will become apparent that certain terms can be combined and simplified. Additionally, the exact value of  $k_D$  will be derived. It will be shown that  $k_D$  is exactly equal to  $\log(2\pi)e^{-3/2}$ . However, it will also be shown that  $e^{-3/2}$  is a common factor throughout the formula. It will be separated out, and with the revised formula,  $k_D$  will be redefined simply as  $\log(2\pi)$ . The formula will also be modified to allow for hollow tubular conductors as well as solid conductors. This is governed by the rightmost exponential function. By changing the constant in the exponent from  $\frac{1}{4}$  to 0, it changes from a solid conductor formula to a hollow conductor formula. This fixed constant will be replaced by the variable  $\gamma$ . Thus, the formula becomes:

$$g_S = \left( e^{-\frac{3}{2}}n + \log(2\pi)e^{-\frac{3}{2}} + \frac{1 - e^{-\frac{3}{2}} - \log(2\pi)e^{-\frac{3}{2}}}{1 + k_F(n - 1)} \right) e^{(-\frac{\gamma}{n})}u^{(\frac{1}{n})}p \quad (5.2.16)$$

As the final step here, the factor  $e^{-3/2}$  will be factored out, and taken outside of the parentheses.

$$g_S = \left( n + \log(2\pi) + \frac{e^{\frac{3}{2}} - \log(2\pi) - 1}{1 + k_F(n - 1)} \right) e^{-\frac{3}{2}}e^{(-\frac{\gamma}{n})}u^{(\frac{1}{n})}p \quad (5.2.17)$$

$$k_F = 1.583635$$

$$\gamma = \frac{1}{4} \text{ for solid conductors}$$

$$\gamma = 0 \text{ for thin tubular conductors}$$

### 5.3 Formula for the GMD of a Linear Array of Co-linear Straight Line Segments (LALS)

Having worked through the derivation of the GMD for the Linear Array of Circular Conductors (LACC), and recognizing the simple correspondence of each term in the formula to either the GMD of a circular conductor, or the GMD of a straight line, it quickly becomes apparent that it is nearly trivial to arrive at the equivalent GMD formula for a linear array of any geometric shape. A linear array of co-linear straight line segments (henceforth **LALS**) is a worthwhile example, because it can represent a current sheet which has been very well studied, and it gives us the opportunity to make comparisons between the two.

First of all, from formula (2.2.6), the self GMD factor  $\gamma$  will change from  $\frac{1}{4}$  to  $\frac{3}{2}$ .

The mutual GMD factor will change from the simple distance between circle centres to the more complicated formula (2.1.4) for the GMD for co-linear line segments. Fortunately, we already know from the previous derivation, that the mutual GMD factor must quickly tend to a straight line with a slope of  $e^{-3/2}$ , and therefore the influence of (2.1.4) can be distilled down to nothing more than the single constant,  $k_D$ , the density uniformity constant. Starting with (5.2.15) and substituting  $\gamma$  for the fixed self inductance constant, our GMD function looks like this:

$$g_S = \left( e^{-\frac{3}{2}n} + k_D + \frac{1 - e^{-\frac{3}{2}} - k_D}{1 + k_F(n-1)} \right) e^{(-\frac{\gamma}{n})u} u^{\left(\frac{1}{n}\right)} p \quad (5.3.1)$$

The variable  $u$  is now the ratio of line segment length to pitch.

$$u = s/p \quad (5.3.2)$$

The density uniformity constant  $k_D$  is not expected to be the same as for the LACC (round conductor) case, but it can be readily determined. By calculating the summations using a discrete summation formula where segment length is equal to the pitch ( $u=1$ ), the value of  $k_D$  is determined to be approximately 0.334707. However, if the ratio of segment length to pitch approaches zero, then it is apparent that the segments have an effect on the mutual component the same as infinitesimal points, and the value of  $k_D$  must become the same as for the round conductor case, 0.410086. This has been verified by further computer calculations. Therefore,  $k_D$  is not a fixed constant, but is now a function of the ratio of segment length to pitch,  $u$ . These two values for  $k_D$  are the limiting values, and  $k_D$  varies between them in a smooth curve. The function  $k_D(u)$  will be discussed shortly.

Remaining to be determined is the empirical coefficient  $k_F$  which, as before, is used to correct for the transition region between the linear large  $n$  region, and the small  $n$  region where the value must diverge from linearity to the predefined constant values. Because  $k_D$  is no longer a fixed constant, the optimum value of  $k_F$  may also vary.

For the special case where segment length/pitch ratio  $u$  is one, we can remove the  $u$  factor:

$$g_S = \left( e^{-\frac{3}{2}}n + k_D + \frac{1 - e^{-\frac{3}{2}} - k_D}{1 + k_F(n - 1)} \right) e^{\left(\frac{-3}{2n}\right)p} \quad (5.3.3)$$

Because the segment length to pitch ratio  $u$  is equal to one, this is little more than an elaborate formula for the GMD of a line of length  $n p$ . Therefore, it should give the same value as the standard formula (2.2.6), i.e.:  $n p e^{-3/2}$ . Allowing for the approximation error in (5.3.3) when  $n$  is small, these two formulae are indeed equivalent. The real value of (5.3.3) is that we can equate (5.3.3) to (2.2.6), taking the limit as  $n$  approaches infinity, to solve for  $k_D$  exactly. Doing so yields the following result:

$$k_D(1) = \frac{3}{2} e^{-3/2} = 0.33469524\dots \quad (5.3.4)$$

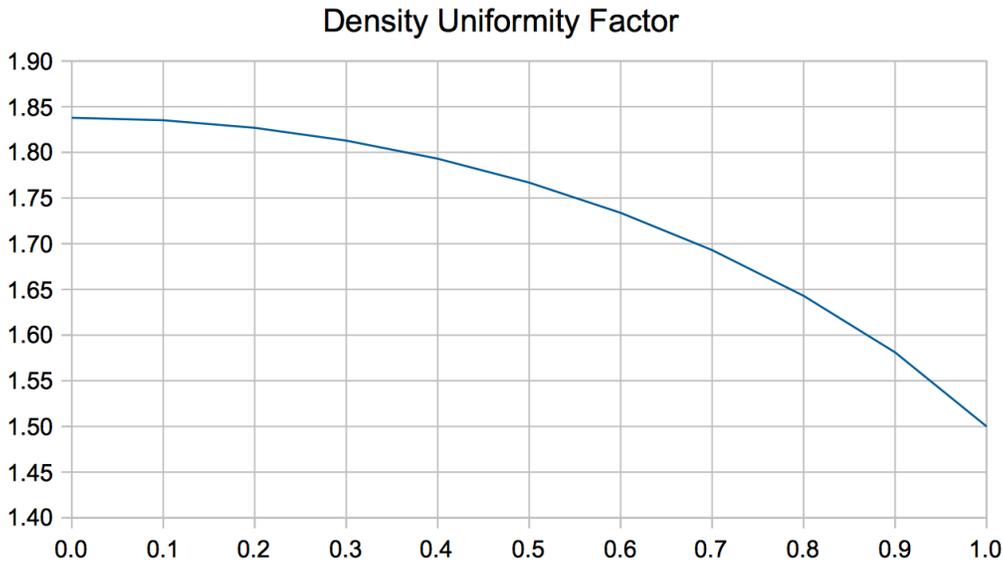
While this may seem obvious in retrospect, this value converged quite slowly when doing the numerical summations, and did not provide enough precision to use any of the available inverse symbolic calculation tools.

Having now derived the exact value of  $k_D$  for when  $u = 1$ , it is hoped that this may give some insight into the exact value for  $k_D$  when  $u = 0$ . This value does not converge very quickly, but by using a convergence acceleration software tool, the value of  $k_D(0)$  is estimated to be 0.4100858059 with an estimated error of  $\pm 1.34 \times 10^{-9}$ . A reasonable possibility is:

$$k_D(0) \stackrel{?}{=} \log(2\pi) e^{-\frac{3}{2}} = 0.4100858042\dots \quad (5.3.5)$$

which is in agreement to 8 significant figures, and the simplicity of the factors is quite compelling. The fundamental constant,  $\pi$ , has made a sudden surprise appearance. This is not entirely unexpected, because this exercise has been leading up to a comparison between round wires and current sheet segments, and the value,  $\log(2\pi) - \frac{3}{2}$ , is the limiting value, at  $n = \infty$ , of Rosa's round wire mutual inductance correction. For that reason, we should expect this term to show up at some point. Indeed, Rosa's formula for round wire corrections will be used later to demonstrate that  $k_D(0)$  is exactly equal to  $\log(2\pi) e^{-3/2}$ .

Considering that the limiting values of  $k_D$  are  $\log(2\pi) e^{-3/2}$ , and  $\frac{3}{2} e^{-3/2}$ , it appears that the common factor of  $e^{-3/2}$  can be removed, and we can focus on the unique values,  $\log(2\pi)$  and  $\frac{3}{2}$ , and then try to fathom how  $k_D$  transitions in a smooth curve between the two, as a function of  $u$ . It is not a straight line function of  $u$ . It changes rapidly in the region of  $u = 1$ , but flattens out as it approaches its lower limiting value at  $u = 0$ , as shown in the graph below.



From the graph, it can be seen that at  $u = 0$ , the slope is apparently zero, or very nearly so. There are many functions that can be adopted, having a derivative of zero at  $u = 0$ . However, given the frequency with which it has been encountered during this investigation, the exponential function seems to be a likely candidate. There is further evidence to support this hypothesis: a least squares fit using polynomials, ranging from degree 2 all the way up to degree 7, produces coefficients of roughly equal magnitude, hinting that there is an underlying power series. Unfortunately, the exponential function never has a zero derivative, no matter what value its argument takes. However, it can be very simply modified, thus:

$$u^m e^{u^m} \tag{5.3.6}$$

where  $m \geq 2$ . When  $u = 0$ , the derivative will be zero. We will start with the lowest order option,  $m = 2$ , and including a single adjustable fitting constant  $k_{F2}$ :

$$k_D = \log(2\pi) - k_{F2} u^2 e^{k_{F2} u^2} \tag{5.3.7}$$

In this form,  $k_{F_2}$  has no effect on  $k_D(0)$ . Therefore, we will calculate  $k_{F_2}$  to fit the other limiting value of  $k_D(1) = \log(2\pi) - \frac{3}{2}$

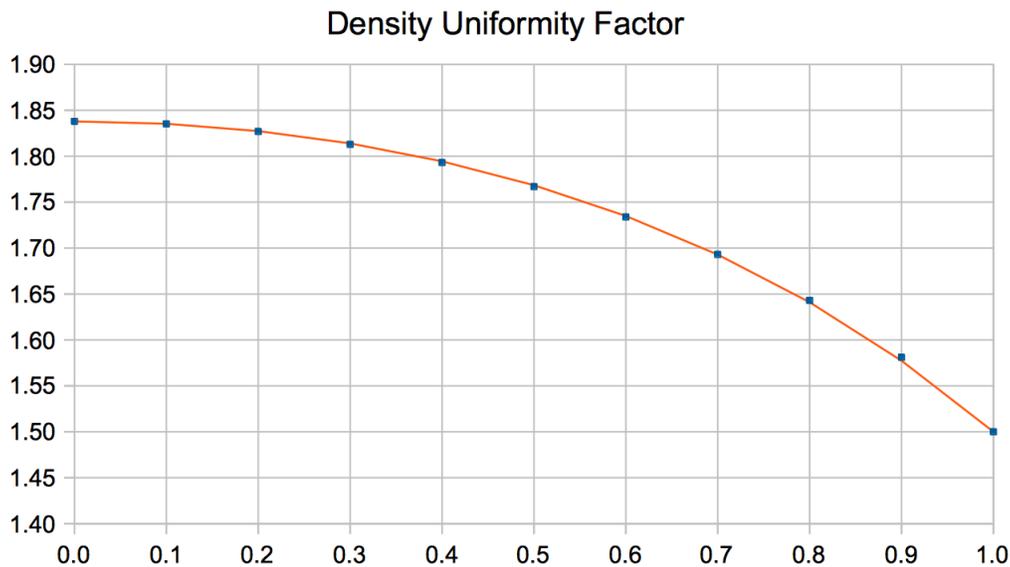
Hence:

$$k_{F_2} e^{k_{F_2}} = \log(2\pi) - \frac{3}{2} \quad (5.3.8)$$

Solving for  $k_{F_2}$  we get

$$k_{F_2} = 0.26041306 \quad (5.3.9)$$

And, plotting this prospective fitting function (red line) with superimposed data points (blue), we get:



The fit is remarkably good considering that it has been fit only to the end points. Considering that the mid region data is less accurate due to its slow convergence, it's probably best to leave things as they are, and consider our search for a fitting function to be complete.

Having found a function for  $k_D$ , we still need to adjust empirical transitional coefficient  $k_F$ . As previously mentioned, a single value for  $k_F$  will quite possibly not be optimum for all values of  $k_D$ . In testing, it was found that the optimum value of  $k_F$  varied from 1.584 for  $u=0$  to 1.709 for  $u=1$ . However the peak error was typically around 0.3%, and it was found to be a fairly broad peak. Choosing a fixed value for  $k_F = 1.6334$  gave a peak error of no more than 0.44% for any value of  $u$ , and so there appears to be no reason to complicate matters by making  $k_F$  a function of  $u$ . Therefore,  $k_F$  will be kept fixed. We now have:

$$g_S = \left( n + \log(2\pi) - k_{F_2} u^2 e^{k_{F_2} u^2} + \frac{e^{\frac{3}{2}} - \log(2\pi) - k_{F_2} u^2 e^{k_{F_2} u^2} - 1}{1 + k_F(n-1)} \right) e^{-\frac{3}{2}} e^{(-\frac{\gamma}{n})} u^{(\frac{1}{n})} p \quad (5.3.10)$$

$$k_F = 1.6334$$

$$k_{F_2} = 0.26041306$$

$$\gamma = 3/2$$

Formula (5.3.10) is a more general form than the round conductor formula (5.2.17), because it accounts for a varying density uniformity term, which is required for any geometric shape other than a circle. It is no doubt possible to adapt (5.3.10) to other oblong figures such as rectangles or ellipses, though it is not known at this point whether it is simply a matter of calculating new values for  $k_F$  and  $k_{F_2}$ . The value of  $\gamma$  will certainly become a function of the figure's shape factor.

#### 5.4 Inductance Calculation Based on Aggregate GMD

Formulae (5.2.17) for the circular array LACC GMD, and (5.3.10) for the linear array (LASL), developed previously, can be directly applied in the standard formula for the mutual inductance between two parallel straight filaments (1.4.2) to determine the self-inductance of an array of parallel round or flat conductors, such as for busbars. Below, the spacing  $x$  between filaments in formula (1.4.2) is replaced with the value  $g_S$  calculated from formula (5.2.15).

$$L = \frac{\mu_0 \ell}{2\pi} \left[ \log \left( \frac{\ell}{g_S} + \sqrt{1 + \left( \frac{\ell}{g_S} \right)^2} \right) - \sqrt{1 + \left( \frac{g_S}{\ell} \right)^2} + \frac{g_S}{\ell} \right] \quad (5.4.1)$$

What may be even more interesting, is to see how the LACC and LASL GMD formulae can be applied to calculate the self-inductance of a coil. In Part 4, it was shown that when the GMD inductance calculation method is applied to loops rather than straight conductors, the result will be accurate as long as the largest dimension of the conductor cross section is significantly smaller than the diameter of the loop. This would suggest that for a short coil, i.e., one that is significantly shorter than its diameter, the LACC and LASL GMD formulae could be used to determine the GMD of the complete cross section of the coil, and then applied just once in a parallel loop mutual inductance calculation, to determine the self-inductance of the coil. Instead of performing  $n$  mutual inductance calculations for an  $n$  turn coil, only one mutual inductance calculation would be required. The result would then be multiplied by  $n^2$  to account for effect of the multiple turns. It seems plausible that this procedure could produce useable inductance values for short coils.

A filamentary loop mutual inductance formula is given by Maxwell [1c]. Adjusted for SI units, and using this document's symbol conventions, it is:

$$M = \mu_0 r_c \left[ \left( 1 + \frac{3}{16} \frac{x^2}{r_c^2} \right) \log \frac{8r_c}{x} - \left( 2 + \frac{x^2}{16r_c^2} \right) \right] \quad (5.4.2)$$

Replacing the axial separation,  $x$ , with the self GMD of the coil,  $g_s$ , and including the adjustment for the number of turns  $n$ , we get:

$$L = \mu_0 r_c n^2 \left[ \left( 1 + \frac{3}{16} \frac{g_s^2}{r_c^2} \right) \log \frac{8r_c}{g_s} - \left( 2 + \frac{g_s^2}{16r_c^2} \right) \right] \quad (5.4.3)$$

Because  $g_s$  appears at three different places in the formula, it's not practical to expand it out into a one line formulae. Instead, we will leave this as a two part formula using (5.4.3) with either (5.2.17) or (5.3.10). Nevertheless, it is a closed form formula, requiring only direct calculation of readily available math functions, with no need for iteration. Hence, they can be placed directly into a spreadsheet for calculation.

What may be of interest is that we have only discussed the arrangement of the cross section as a linear array of circles (or line segments), without specifying its orientation with respect to the coil axis. In other words, we have not yet considered whether the coil is a helix or a spiral, and the results, so far, would indicate that in the case of short or shallow windings of large diameter, there should be no significant difference in the inductance value between the two. This is indeed what has been found in practice, and will be demonstrated here.

Having available an existing set of spreadsheet macros which use the mutual inductance summation method to calculate self inductance of a solenoid [7], it was a simple matter to compare the two calculation methods.

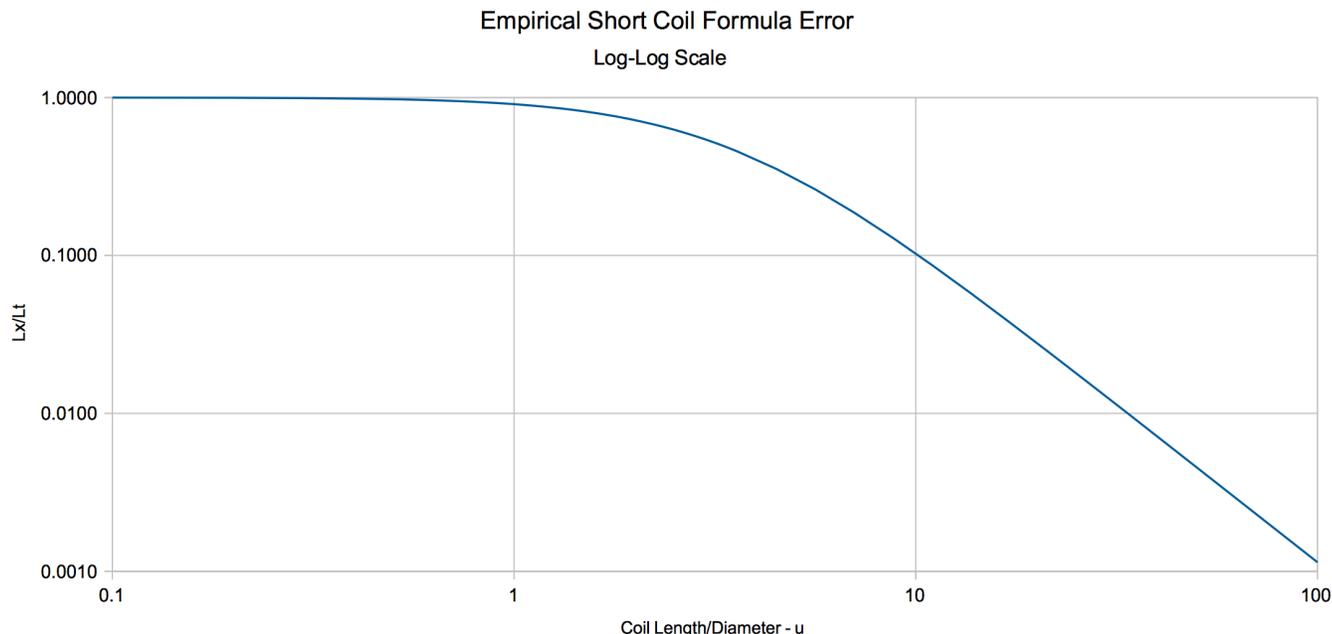
In this discussion, the inductance value as calculated by the accurate summation macros will be represented by  $L_T$  (subscript  $T$  for *true* inductance), and the value calculated by the experimental LACC method as  $L_X$ . The following graph shows the ratio of  $L_X/L_T$ , for different ratios of coil length to coil diameter. We will also define the coil shape factor (i.e., coil length to diameter ratio):

$$u_c = \ell_c / D$$

where the coil length is:

$$\ell_c = n \times p$$

Plotting  $L_X/L_T$  vs. the coil shape factor  $u_c$  gives the following graph:



The results of the comparison show that the experimental formula does produce good results (within 1%) for coils where the ratio of winding length to coil diameter is  $\frac{1}{3}$  or less.

The graph appears to be doubly asymptotic, approaching 1 for small values of  $u$ , and approaching a straight line (on the log-log graph) with a constant negative slope for large values of  $u$ . Analysis of the data shows that the slope of this asymptote is equal to  $-2$ , indicating that the  $L_X/L_T$  is a function of  $1/u_c^2$  as  $u_c$  gets larger. This would seem to indicate that the formula's accuracy could be extended for larger values of  $u_c$  by including a simple correction term. Indeed, if the pitch and conductor diameter are held constant, with the only variable being the coil length to diameter ratio  $u_c$ , then this is exactly the case. However, it must be pointed out that as  $u_c$  becomes larger, the error in the inductance value calculated by the LACC formula becomes very large, and hence, the required correction factor becomes extremely large: an electromagnetic instance of the tail wagging the dog. This, by itself, would not be so bad, except that when we now include the effect of pitch and wire diameter, even these small influences are amplified when  $u_c$  is large. As a result, a simple correction factor based only  $u_c$ , does not appear to be practical. The better option is to accept these as short coil formulae, with the benefit that they properly account for conductor shape. If calculations for longer coils are desired, then it is more practical to employ a method given in the following section.

### 5.5 Rosa's Round Wire Inductance Corrections

In the first decade of the 20th century, the science of inductance calculation was evolving rapidly. At that time, there were only two basic methods for calculating the inductance of a single layer solenoid coil: the current sheet method, and the summation method. (Now, more than a hundred years later, there are two basic methods for calculating the inductance of a single layer solenoid

coil: the current sheet method, and the summation method.) The current sheet method was best suited for long coils wound with many turns of fine wire and of small pitch. The summation method was more accurate, but involved the calculation of the self inductance of one turn of the coil, as well as the mutual inductance between a turn and every other turn in the coil. For a coil of  $n$  turns, it was necessary to do  $(n-1)$  mutual inductance calculations, a very tedious process in the days before computers. So, this method was impractical for coils of more than a few turns. Though, Strasser [11] derived such a formula and provided a table to aid in determining the mutual inductance component for coils of up to 30 turns.

The current sheet formula approximates the coil with a very thin cylindrical sheet of the same length and diameter as the coil under consideration. The inductance of the current sheet can be solved exactly, and is approximately correct for the actual coil.

In 1906, Edward Rosa, a physicist working for the U.S. National Bureau of Standards, investigated these two calculation methods [12]. He found that in certain cases, there is a small but significant difference in the results produced by these two methods. He reasoned that, by comparing the difference between the self inductance of one turn of a round wire and the self inductance of one turn of current sheet, and comparing the difference between the mutual inductance between turns of round wire and the mutual inductance between turns of current sheet that it would be possible to come up with a set of round wire correction factors that could be applied to the current sheet formula, making it as accurate as the summation method, but far easier to calculate.

Rosa derived two correction factors. The first,  $k_S$ , corrects for the difference in self inductance, and the second,  $k_M$ , corrects for the difference in mutual inductance.  $k_S$  is a function of a single argument: the ratio of pitch to wire radius  $p/r$ . Similarly,  $k_M$  is a function of a single argument: the number of turns  $n$ .

Rosa's complete correction, adjusted to SI units, is:

$$\Delta L = \mu R n (k_S + k_M) \quad (5.5.1)$$

where  $R$  is the coil radius and  $n$  is the number of turns.

This is then applied to the current sheet inductance  $L_S$  to get the corrected inductance value:

$$L = L_S - \Delta L \quad (5.5.2)$$

Note that  $\Delta L$  can be either positive or negative.

Rosa based his corrections on Maxwell's formula for the mutual inductance of two circular loops:

$$L = 4\pi R \left[ \left( 1 + \frac{3}{16} \frac{x^2}{R^2} \right) \log \left( 8 \frac{R}{x} \right) - 2 - \frac{1}{16} \frac{x^2}{R^2} \right]. \quad (5.5.3)$$

$R$  is the radius of the loops, and  $x$  is their axial separation. Rosa found that the second order terms amounted to only about 1 part/million, and would not contribute significantly to the ultimate value. He eliminated them, leaving only the log term and the numeric constant inside the brackets:

$$L = 4\pi R \left[ \log \left( 8 \frac{R}{x} \right) - 2 \right] \quad (5.5.4)$$

To calculate the self inductance of one turn of current sheet, he replaced  $x$  with the self GMD of a straight line, which is its length multiplied by  $e^{-3/2}$ . Rosa factored out the number of turns and coil length, so that the length of a current sheet segment is now simply equal to the pitch  $p$ .

To calculate the self inductance of one turn of round wire, he replaced  $x$  with the self GMD of the conductor cross section,  $r e^{-1/4}$ .

Taking the difference between these two inductance values:

$$\begin{aligned} \Delta L_S &= 4\pi R \left( \left[ \log \left( 8 \frac{R}{r e^{-\frac{1}{4}}} \right) - 2 \right] - \left[ \log \left( 8 \frac{R}{p e^{-\frac{3}{2}}} \right) - 2 \right] \right) \\ &= 4\pi R \left( \log \left( 8 \frac{R}{r e^{-\frac{1}{4}}} \right) - \log \left( 8 \frac{R}{p e^{-\frac{3}{2}}} \right) \right) \end{aligned} \quad (5.5.5)$$

Since the difference between two logarithms is the same as the logarithm of the quotient of their arguments, this then becomes:

$$\Delta L_S = 4\pi R \log \left( \frac{p e^{-\frac{3}{2}}}{r e^{-\frac{1}{4}}} \right) \quad (5.5.6)$$

The exponentials can be combined and taken outside of the log function giving a constant term of  $\frac{5}{4}$ . In addition,  $4\pi R$  appears as a common factor in all of the correction terms. And so, Rosa removed it from the individual correction factors, and applied it in the summary calculation (see formula (5.5.1)). When using SI units, the  $4\pi$  term will be replaced by the permeability constant  $\mu$ . Rosa also used the convention that the correction factor would be subtracted from the base inductance value, thus making the sign the opposite of what has been used here. Thus, Rosa's self inductance correction factor  $k_S$  is:

$$k_S = \frac{5}{4} - \log \left( \frac{p}{r} \right) \quad (5.5.7)$$

Rosa's derivation of the mutual inductance correction term  $k_M$  is more involved, requiring a summation of  $n - 1$  mutual inductances for an  $n$  turn coil. For many years the most accessible form of Rosa's  $k_M$  corrections was the tabulated data, precise to about four decimal places, in Grover's book [2](Table 39, page 150). For this correction term, Grover used the symbol  $H$  rather than  $k_M$ . In 2006, David Knight [14] created an empirical fitting function to reproduce the values in this table. In 2008, R. Weaver set about creating some computer code [15] to calculate Rosa's mutual

inductance corrections using the original summation formula. This resulted in an iterative computer function that could calculate Rosa's  $k_M$  correction to much higher precision. Because it was based on Rosa's summation method, it required one iteration for every turn of the coil, making it somewhat cumbersome, and slow for coils with many turns. However, its value has been in its ability to provide a set of accurate data which could be used as a benchmark. With a bit more research, a series formula by Grover was located [13](formula 31, page 176):

$$k_m = \log(2\pi) - \frac{3}{2} - \frac{1}{6n} \log(n) - \frac{0.330842}{n} - \frac{1}{120n^3} + \frac{1}{504n^5} \quad (5.5.8)$$

This formula is accurate for large  $n$  but is not suitable for  $n < 5$ , and notably, it gives the wrong value for  $n = 1$ . David Knight [14] adapted this formula, adding a 7th order empirical term to correct the accuracy for  $n < 5$  and a 9th order closing term to force it to the correct value of zero for  $n = 1$ . Knight's formula is:

$$k_m = \log(2\pi) - \frac{3}{2} - \frac{1}{6n} \log(n) - \frac{0.33084236}{n} - \frac{1}{120n^3} + \frac{1}{504n^5} - \frac{0.0011935}{n^7} + \frac{0.000507}{n^9} \quad (5.5.9)$$

This formula has a worst case absolute error of  $\pm 1.3 \times 10^{-8}$ , making it essentially exact for any practical purpose.

The derivation of Rosa's summation formula for  $k_M$  will not be given here, but the reader can refer to the references for details. The preceding synopsis should be sufficient background for what is to follow.

From the foregoing, it would seem apparent that by taking the difference of the logarithms of the LACC formula (5.2.15) and the co-linear line GMD formula (2.2.6) for of a line of equal length, then it should result in the same value as Rosa's  $k_s$  and  $k_M$  factors combined. Thus, we start with (5.2.15) and (2.2.6) combined:

$$k_s + k_m = n \log \left[ \left( e^{-\frac{3}{2}n} + k_D + \frac{1 - e^{-\frac{3}{2}} - k_D}{1 + k_F(n-1)} \right) e^{\left(\frac{-1}{4n}\right)u\left(\frac{1}{n}\right)p} \right] - n \log \left( npe^{-\frac{3}{2}} \right) \quad (5.5.10)$$

Note that the factor  $n$  is included outside of the log function. This is due to a primary difference between the LACC formula and Rosa's formula. The LACC formula is a GMD calculation for a complete winding, and in order to turn it into an inductance value, it must be multiplied by  $n^2$ . Rosa's formulae were developed as per-turn corrections and are only multiplied by  $n$  in his overall  $\Delta L$  correction function. Therefore, an additional factor of  $n$  needs to be included in the LACC formula to yield the same results.

Converting the difference of logarithms to the logarithm of a quotient:

$$k_s + k_m = n \log \left[ \left( e^{-\frac{3}{2}}n + k_D + \frac{1 - e^{-\frac{3}{2}} - k_D}{1 + k_F(n - 1)} \right) \frac{e^{\left(\frac{-1}{4n}\right)u\left(\frac{1}{n}\right)p}}{npe^{-\frac{3}{2}}} \right] \quad (5.5.11)$$

The pitch  $p$  appears in both numerator and denominator, and cancels out, resulting in a function of only two arguments,  $u$  and  $n$ . The value of  $k_D$ , has previously been given as  $\log(2\pi) e^{-3/2}$  (though not yet proven), and is now substituted for  $k_D$ :

$$k_s + k_m = n \log \left[ \left( e^{-\frac{3}{2}}n + \log(2\pi)e^{-\frac{3}{2}} + \frac{1 - e^{-\frac{3}{2}} - \log(2\pi)e^{-\frac{3}{2}}}{1 + k_F(n - 1)} \right) \frac{e^{\left(\frac{-1}{4n}\right)u\left(\frac{1}{n}\right)}}{ne^{-\frac{3}{2}}} \right] \quad (5.5.12)$$

Inside the first parentheses, the factor  $e^{-3/2}$  occurs in nearly every term, and will be taken outside:

$$k_s + k_m = n \log \left[ \left( n + \log(2\pi) + \frac{e^{\frac{3}{2}} - 1 - \log(2\pi)}{1 + k_F(n - 1)} \right) e^{-\frac{3}{2}} \frac{e^{\left(\frac{-1}{4n}\right)u\left(\frac{1}{n}\right)}}{ne^{-\frac{3}{2}}} \right] \quad (5.5.13)$$

The factor  $e^{-3/2}$  outside the parentheses now cancels with the factor  $e^{-3/2}$  in the denominator of the rightmost fraction. We will replace the fixed coefficient for circular areas,  $1/4$ , in the self GMD exponent, with the variable  $\gamma$ , to allow the formula to be used for both solid round wires and round tubular conductors.

$$k_s + k_m = n \log \left[ \left( n + \log(2\pi) + \frac{e^{\frac{3}{2}} - \log(2\pi) - 1}{k_F(n - 1) + 1} \right) \frac{e^{\left(\frac{-\gamma}{n}\right)u\left(\frac{1}{n}\right)}}{n} \right] \quad (5.5.15)$$

There is no reason why the constant  $k_F$  cannot be re-optimized to give the minimum error in this application. Doing so gives a marginally different value than before:

$$k_F = 1.59516$$

In testing, the formula has a worst case error of no more than about 1% in most cases. The error here is higher than the peak error of 0.17% of formula (5.2.15) by itself, because the correction formula involves the subtraction of terms of similar value, thus increasing the % error. In certain situations when Rosa's  $k_S$  and  $k_M$  corrections are roughly equal in magnitude and opposite in sign, while at the same time,  $n$  is less than 10 (the least accurate region of the empirical factor), then the % error can be considerably higher, though at the same time, the absolute error is extremely small. Therefore, it is of no great consequence, because in that case, the actual correction is negligible. However, this can be disconcerting when one looks only at relative error rather than absolute error.

To verify that this error is due only to the approximation in the empirical factor, and not a fundamental error in the derivation, the formula was also tested with the empirical factor replaced with the exact numerical summation. In this case the agreement with Rosa's original calculations was found to be exact, thus validating this formula.

—

Now that we have verified that this formula is equivalent to Rosa's formula, we are finally at a point where we can demonstrate that the value of the density uniformity constant,  $k_D$ , is in fact exactly  $\log(2\pi)$ .

Grover has demonstrated that the limiting value at  $n = \infty$  for Rosa's  $k_M$  is  $\log(2\pi) - 3/2$  [13](page 176). Also, refer to formula (5.5.8). Therefore, we must show that the limit as  $n$  goes to  $\infty$  of formula (5.5.15) also gives this same value. We will replace  $k_M$  on the left side of the formula with the limiting value. The empirical fractional term on the right side inside the parentheses will go to zero due to  $n$  being in the denominator. So, this term can immediately be eliminated. Also, as we have not yet proven that  $k_D = \log(2\pi)$ , we will put  $k_D$  back into the formula. So, we begin with:

$$\log(2\pi) - \frac{3}{2} + k_s = \lim_{n \rightarrow \infty} \left[ n \log \left( (n + k_D) \frac{e^{\left(\frac{-\gamma}{n}\right)} u^{\left(\frac{1}{n}\right)}}{n} \right) \right] \quad (5.5.16)$$

The self inductance correction  $k_s$  will be dealt with presently. Next we take the factor  $n$  inside the log term making it an exponent:

$$\log(2\pi) - \frac{3}{2} + k_s = \lim_{n \rightarrow \infty} \left[ \log \left( (n + k_D) \frac{e^{\left(\frac{-\gamma}{n}\right)} u^{\left(\frac{1}{n}\right)}}{n} \right)^n \right] \quad (5.5.17)$$

Rearranging:

$$\log(2\pi) - \frac{3}{2} + k_s = \lim_{n \rightarrow \infty} \left[ \log \left( \left( \frac{n + k_D}{n} \right)^n \left( e^{\left(\frac{-\gamma}{n}\right)} u^{\left(\frac{1}{n}\right)} \right)^n \right) \right] \quad (5.5.18)$$

Making use of this exponential relationship:

$$(a^b)^c = a^{bc}$$

the  $n$  exponent can be combined with the exponents of  $e$  and  $u$ ;  $n$  and  $1/n$  cancel each other out, leaving these terms no longer a function of  $n$ :

$$\log(2\pi) - \frac{3}{2} + k_s = \lim_{n \rightarrow \infty} \left[ \log \left( \left( \frac{n + k_D}{n} \right)^n e^{-\gamma} u^1 \right) \right] \quad (5.5.19)$$

At this point we will move the limit inside the log function. We can do this because the function is continuous.

$$\log(2\pi) - \frac{3}{2} + k_s = \log \left( \lim_{n \rightarrow \infty} \left[ \left( \frac{n + k_D}{n} \right)^n e^{-\gamma u} \right] \right) \quad (5.5.20)$$

This remaining limit is a well known exponential identity. For example, see Abramowitz and Stegun formula 4.2.21 [15] (page 70):

$$\lim_{a \rightarrow \infty} \left[ \frac{a + x}{a} \right]^a = e^x \quad (5.5.21)$$

Therefore the remaining limit can be replaced with a simple exponential  $e^{k_D}$ , giving:

$$\log(2\pi) - \frac{3}{2} + k_s = \log \left( e^{k_D} e^{-\gamma u} \right) \quad (5.5.22)$$

We must now consider that (5.5.15) includes both self and mutual inductance corrections. Therefore, we will choose a value for  $u$  that gives a self inductance correction  $k_s = 0$ . Then, the resulting value of the formula will be due to the mutual inductance correction factor alone. From Rosa:

$$k_s = \frac{5}{4} - \log \left( \frac{p}{r} \right) = \frac{5}{4} - \log \left( \frac{1}{u} \right) \quad (5.5.23)$$

Setting  $k_s$  to zero we get:

$$\log \left( \frac{1}{u} \right) = \frac{5}{4} \quad (5.5.24)$$

Hence:

$$u = e^{-\frac{5}{4}} \quad (5.5.25)$$

Substituting this into the main formula:

$$\log(2\pi) - \frac{3}{2} = \log \left( e^{k_D} e^{-\gamma} e^{-\frac{5}{4}} \right) \quad (5.5.26)$$

Combining the log and exponential terms on the righthand side of the formula:

$$\log(2\pi) - \frac{3}{2} = k_D - \gamma - \frac{5}{4} \quad (5.5.27)$$

For solid round wires,  $\gamma = 1/4$ . So:

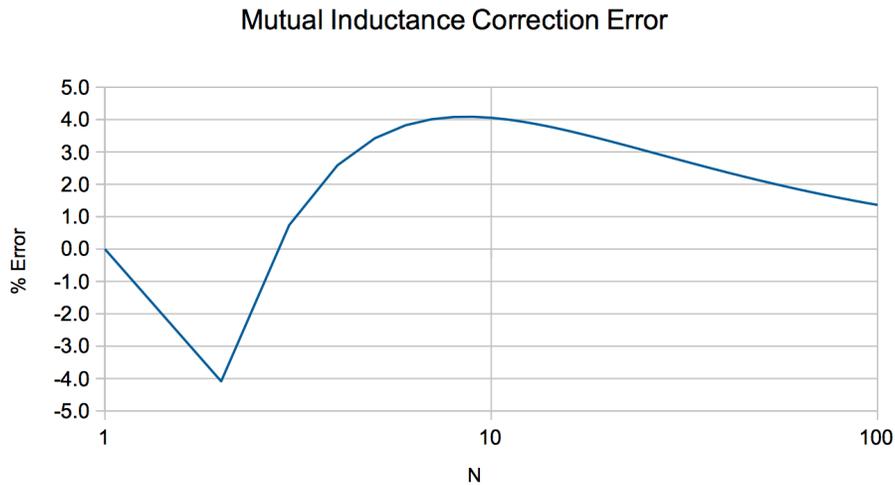
$$\log(2\pi) - \frac{3}{2} = k_D - \frac{1}{4} - \frac{5}{4} \quad (5.5.28)$$

The numeric constants cancel out, leaving:

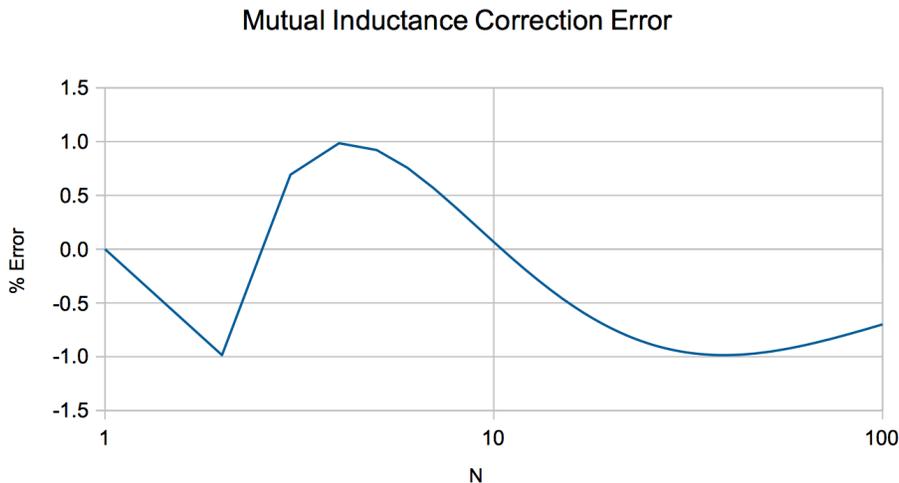
$$k_D = \log(2\pi) \quad . \quad (5.5.29)$$

Thus, we have derived the analytical value for  $k_D$ .

Now, returning to the topic of the accuracy of formula (5.5.15), it can be stated, that because the self inductance correction component is not affected by any approximations in the derivation of the formula, it is exact. So, all error in the formula is due strictly to the empirical portion of the mutual component. Therefore, to better judge accuracy, we can again set  $u$  equal to  $e^{-5/4}$ , resulting in a zero value for  $k_s$ . Then we will compare formula (5.5.15) with the formula for Rosa's  $k_M$  correction. Doing this gives the following:



The peak error is  $\pm 4\%$  occurring in two places at  $u < 10$ . Bear in mind that a mutual inductance correction with an error of 4% when combined with an exact self inductance correction and then applied to an inductance calculation will result in a much smaller overall percentage error. Nevertheless, it does seem unfortunate that after the effort of deriving this function that we can't have a smaller peak error. Therefore, a second order term was added to the empirical part of the function, and re-optimized, now giving a peak error of  $\pm 1\%$  as shown in the following graph.



The resulting formula is:

$$k_s + k_m = n \log \left[ \left( n + \log(2\pi) + \frac{e^{\frac{3}{2}} - \log(2\pi) - 1}{k_{F_0}(n-1) + k_{F_1}(n-1)^2 + 1} \right) \frac{e^{\left(\frac{-\gamma}{n}\right)} u\left(\frac{1}{n}\right)}{n} \right] \quad (5.5.30)$$

And the empirical constants are:

$$k_{F_0} = 1.530574$$

$$k_{F_1} = 0.0326627$$

In addition, the constants in the numerator of the empirical term, which have been kept in their symbolic form throughout the derivation, can be evaluated and combined, giving a value of  $e^{3/2} - \log(2\pi) - 1 = 1.643812004$ .

A peak error of  $\pm 1\%$  sounds more impressive, but makes little difference in the overall scheme of things. Of course, for utmost accuracy, Rosa's  $k_s$  formula (5.5.7) combined with Knight's  $k_M$  formula (5.5.9) will give essentially exact results.

The most useful test is in comparing actual coil inductance values calculated by:

- i) the current sheet inductance, corrected with formula (5.5.30);
- ii) the current sheet inductance, corrected with formulae (5.5.7) and (5.5.9);
- iii) the iterative summation method;

The peak error in formulae (5.5.15) and (5.5.30) always occurs at  $n = 2$ , and so we will limit our comparison to coils of two turns. We will presume method iii) to give the most accurate inductance value against which the others will be compared. This is a fair assumption when the pitch and conductor diameter are small compared to the coil diameter.

The following table compares the inductance calculated by the summation method with the inductance calculated by the current sheet method which is then corrected with formula (5.5.15), (5.5.30) and the original rosa corrections.

<b>Corrected Current Sheet Inductance compared to Summation Inductance</b>		<b>Coil Radius R</b> 100.0			<b>Conductor Radius r</b> 1.0			<b>Coil turns n</b> 2		
<b>p/d</b>	<b>u</b>	<b>Inductance</b>		<b>Corrections</b>			<b>Error (parts/million)</b>			<b>Error Uncorr.</b>
		<b>Summation</b>	<b>Cur. Sheet</b>	<b>(5.5.15)</b>	<b>(5.5.30)</b>	<b>Rosa</b>	<b>(5.5.15)</b>	<b>(5.5.30)</b>	<b>Rosa</b>	
1.00	0.500	2.591565	2.759820	0.664914	0.668439	0.669559	441.25	99.35	-9.26	64923.9
1.10	0.454	2.567640	2.711964	0.569694	0.573220	0.574340	445.50	100.42	-9.20	56208.7
1.20	0.416	2.545797	2.668272	0.482759	0.486284	0.487404	449.57	101.52	-9.03	48108.7
1.30	0.384	2.525703	2.628078	0.402780	0.406305	0.407425	453.48	102.67	-8.77	40533.3
1.40	0.357	2.507098	2.590863	0.328727	0.332252	0.333372	457.28	103.86	-8.40	33410.9
1.50	0.333	2.489778	2.556216	0.259782	0.263307	0.264427	460.98	105.10	-7.94	26684.3
1.60	0.312	2.473576	2.523805	0.195285	0.198810	0.199930	464.61	106.40	-7.38	20306.5
1.70	0.294	2.458357	2.493361	0.134697	0.138222	0.139342	468.18	107.75	-6.74	14238.8
1.80	0.278	2.444008	2.464657	0.077571	0.081097	0.082216	471.70	109.16	-6.00	8448.6
1.90	0.263	2.430436	2.437506	0.023533	0.027059	0.028178	475.19	110.62	-5.18	2908.7
2.00	0.250	2.417561	2.411748	-0.027734	-0.024208	-0.023089	478.66	112.15	-4.27	-2404.5
2.10	0.238	2.405314	2.387247	-0.076500	-0.072975	-0.071855	482.11	113.73	-3.28	-7511.3
2.20	0.227	2.393638	2.363888	-0.122999	-0.119473	-0.118353	485.55	115.37	-2.21	-12429.1
2.30	0.217	2.382482	2.341567	-0.167431	-0.163905	-0.162785	488.98	117.08	-1.06	-17173.2
2.40	0.208	2.371801	2.320198	-0.209972	-0.206447	-0.205327	492.42	118.84	0.17	-21757.2
2.50	0.200	2.361557	2.299701	-0.250778	-0.247252	-0.246132	495.86	120.66	1.48	-26193.0
2.60	0.192	2.351716	2.280009	-0.289983	-0.286457	-0.285338	499.32	122.55	2.87	-30491.1
2.70	0.185	2.342246	2.261062	-0.327709	-0.324183	-0.323064	502.78	124.49	4.33	-34661.0
2.80	0.179	2.333122	2.242804	-0.364063	-0.360538	-0.359418	506.27	126.49	5.86	-38711.2

Units are millimetres and microhenries, but the results are scaleable to any unit, and the error values will remain the same. The ratio of conductor to coil radius  $r/R = 0.01$  is very modest. However, a larger ratio would introduce factors that would overshadow the GMD error.

The first column shows the ratio of pitch to conductor diameter ( $p/d$ ). The second column is the ratio of conductor radius to pitch ( $u=r/p$ ). The next two columns show the inductance values calculated by the summation method and current sheet methods. The inductance corrections (by three methods) are shown in the next three columns, and finally the error due to the correction methods are shown in the last three columns.

Formula (5.5.15) shows an error of about 500 ppM (parts/million) for worst case 2 turn coils. Formula (5.5.30), with the additional fitting term, shows an error of about 120 ppM, while the original Rosa corrections show an error of no more than -9 ppM. Again, these numbers are for the worst case 2 turn coil. For coils of more turns, the error in formulae (5.5.15) and (5.5.30) will drop significantly, closer to the values of the original Rosa values.

Now, compare these values with the error of the current sheet, having no correction at all, as shown in the far right column. These errors are up to 65,000 ppM, or 6.5%. This gives some perspective of just how well these round wire corrections do work.

The above comparison is based on the assumption that the summation inductance calculation is completely accurate. However, the summation calculation assumes that the turns of the coil are perfect parallel circular rings spaced apart at distance equal to the pitch, ignoring the true helicity

of the coil. This is not usually a bad assumption, but if we are comparing the amount of error introduced by other approximations, then this one needs to be addressed as well.

In the following table, the summation calculation is compared to a true helical inductance calculation [17] for the same range of 2 turn coil parameters:

<b>Summation Inductance compared to Helical Inductance</b>				
p/d	u	Inductance		
		Helical	Summation	Error (ppM)
1.00	0.500	2.591303	2.591565	101.45
1.10	0.454	2.567332	2.567640	119.85
1.20	0.416	2.545447	2.545797	137.59
1.30	0.384	2.525312	2.525703	154.87
1.40	0.357	2.506667	2.507098	171.81
1.50	0.333	2.489309	2.489778	188.49
1.60	0.312	2.473069	2.473576	204.97
1.70	0.294	2.457813	2.458357	221.28
1.80	0.278	2.443428	2.444008	237.46
1.90	0.263	2.429820	2.430436	253.52
2.00	0.250	2.416910	2.417561	269.48
2.10	0.238	2.404628	2.405314	285.35
2.20	0.227	2.392918	2.393638	301.14
2.30	0.217	2.381728	2.382482	316.85
2.40	0.208	2.371013	2.371801	332.49
2.50	0.200	2.360736	2.361557	348.05
2.60	0.192	2.350861	2.351716	363.55
2.70	0.185	2.341359	2.342246	378.98
2.80	0.179	2.332202	2.333122	394.34

It can be seen that by failing to take into account the helicity of the coil, the summation method is in error by as much as 400 ppM. This is approximately the same amount of error as the simpler correction formula (5.5.15), and nearly four times the error of the more accurate correction formula (5.5.30).

## 5.6 Flat Conductor Corrections

Formulae in the general form of (5.5.15) or (5.5.30) can, of course, be simply adapted to handle corrections for other conductor shapes. This may be the most useful aspect of this form of correction expression. For example, formula (5.3.10) can be developed into a correction factor for straight co-linear conductors which have gaps between the conductors.

The derivation is exactly the same as for formula (5.5.30), and so only the final formula is given here:

$$k_S + k_M = n \log \left[ \left( n + \log(2\pi) - k_{F_2} u^2 e^{k_{F_2} u^2} + \frac{e^{\frac{3}{2}} - \log(2\pi) - k_{F_2} u^2 e^{k_{F_2} u^2} - 1}{1 + k_F(n-1)} \right) \frac{e^{(-\frac{\gamma}{n})} u^{(\frac{1}{n})}}{n} \right] \quad (5.6.1)$$

$$k_F = 1.6334$$

$$k_{F2} = 0.26041306$$

$$\gamma = 3/2$$

This correction formula would be applicable to flat spiral inductors—etched on printed circuit boards, or fabricated on silicon—when combined with a circular disk current sheet inductance formula, such as that of Rayleigh and Niven [4](formula 70, adjusted for SI units):

$$L_S = \mu\pi Rn^2 \left[ \log \frac{8R}{x} - \frac{1}{2} + \frac{x^2}{96R^2} \left( \log \frac{8R}{x} + \frac{43}{12} \right) \right] \quad (5.6.2)$$

where  $x$  is the radial depth of winding, and  $R$  is the mean radius of the winding.

At the time of writing, this correction formula has been only briefly tested, but it appears to give at least a tenfold improvement in accuracy over an uncorrected disk current sheet inductance calculation, as compared to a loop summation calculation. What is more in question at this time is the range of accuracy of available current sheet formulae. However, this will be the subject of a future article.

## Afterword

This article began as a result of my work developing a method for calculating the inductance of a helical coil that fully accounts for the effect of its helicity for any pitch: 0 to  $\infty$ . That formula made only one approximation: its use of the GMD of the cross section of the conductor, rather than integrating across the conductor cross section. That raised the question of whether the resulting formula gained accuracy due to its accounting for helicity, only to lose it again due to the use of GMD. Unfortunately, there appeared to be no information, readily available, that quantified the error caused by the use of GMD. That lack of information was the spark that launched the investigation into the accuracy of the GMD method, and which eventually became Part 4 of this document. The result of that work has shown that the error introduced by the use of GMD is almost always considerably smaller than the error introduced by the failure to account for the helicity of the coil. Neither the current sheet method nor the summation method account for helicity.

The GMD approximation error becomes significant only when the conductor diameter becomes a very significant fraction of the coil diameter. However, a consequence of this is that the coil pitch must then also become large, and therefore, the non-helicity error also becomes prominent. For coils having a pitch angle of more than about  $6^\circ$  (equivalent to  $p/D = 0.33$ ), the non-helicity error is appreciable. For  $\psi < 6^\circ$ , neither helicity nor GMD error should be significant, except for work requiring the utmost accuracy, and in that case the method of correction given in Section 4.3 may be used.

While the accuracy question has been resolved, that research spawned other lines of investigation into related GMD topics, such as numerical GMD calculation methods, which either had not been treated elsewhere, or else their sources were very obscure. As a result, it seemed worthwhile to combine these other topics to make a more general article on the subject of GMD. There continue to be related areas of interest which will be explored in due course, but if we waited for that list to be exhausted, this document would never be finished.

Finally, I would like to express my appreciation to Dr. David Knight, whose work is cited here, and who reviewed several drafts of this document, and provided valuable feedback.

## References

1. Maxwell, James Clerk; *A Treatise on Electricity and Magnetism*, Vol. 2, Third Edition, Dover 1954;  
(a) Art. 691, *On the Geometrical Mean Distance of Two Figures on a Plane*, pp. 324-326.  
(b) Art. 701, *To find M by Elliptic Integrals*, pp. 338-340.  
(c) Art. 705, *Further approximation of M*, pp. 343-345
2. Grover, Frederick W.; *Inductance Calculations—Working Formulas and Tables*, D. Van Nostrand, 1946; Reprint: Dover, 2004.
3. Dwight, Herbert B.; *Tables of Integrals and other Mathematical Data*; Fourth Edition, MacMillan, 1961.
4. Rosa, E. B. & Grover, F. W.; *Formulas and Tables for the Calculation of Mutual and Self-Inductance*. Scientific Paper 169, National Bureau of Standards, Washington DC, 1916.
5. Gray, Andrew; *Absolute Measurements in Electricity and Magnetism*; 2nd Edition; MacMillan and Co.; 1921;  
<https://archive.org/details/absolutemeasur00grayuoft>  
(Downloaded on 2014-07-25)
6. Seneff, E. Lockwood Jr.; *Study of the Method of Geometric Mean Distances Used in Inductance Calculations*; MScEE Thesis, University of Missouri School of Mines and Metallurgy; 1947;  
[https://mospace.umsystem.edu/xmlui/bitstream/handle/10355/26187/Seneff\\_1947.pdf](https://mospace.umsystem.edu/xmlui/bitstream/handle/10355/26187/Seneff_1947.pdf)  
(Downloaded on 2014-01-29)
7. Weaver, Robert S.; Webpage: Maxwell and Coaxial Circular Conductors  
<http://electronbunker.ca/eb/CalcMethods1b.html>  
(Downloaded on 2014-09-02)
8. Higgins, Thomas James; *Formulas for the Geometric Mean Distances of Rectangular Areas and of Line Segments*; *Journal of Applied Physics*, Vol. 14, April 1943, pp. 188-195.
9. Wichman, B. A. and Hill, I. D.; *Algorithm AS 183: An Efficient and Portable Pseudo-Random Number Generator*; *Applied Statistics*, 31, pp. 188-190, 1982.
10. Wien, Max; *Ueber die Berechnung und Messung kleiner Selbstpotentiale* [*On the Calculation and Measurement of Small Self-potentials*]; *Annalen der Physik*, Vol. 289, No. 13, 1894, pp. 928-947.  
[http://zs.thulb.uni-jena.de/servlets/MCRFileNodeServlet/jportal\\_derivate\\_00147928/18942891312 ftp.pdf](http://zs.thulb.uni-jena.de/servlets/MCRFileNodeServlet/jportal_derivate_00147928/18942891312 ftp.pdf)  
(Downloaded on 2015-08-13)
11. Strasser, B.; *Über die Bestimmung des Selbstinduktionskoeffizienten von Solenoiden* [*On the Determination of Self Induction coefficients of Solenoids*]; *Annalen der Physik*, 1905 pp. 763–771.  
[http://zs.thulb.uni-jena.de/servlets/MCRFileNodeServlet/jportal\\_derivate\\_00150036/19053220912 ftp.pdf](http://zs.thulb.uni-jena.de/servlets/MCRFileNodeServlet/jportal_derivate_00150036/19053220912 ftp.pdf)  
(Downloaded on 2016-03-06)

12. Grover, Frederick W.; *A Comparison of the Formulas for the Calculation of the inductance of coils and spirals wound with wire of large cross section*; National Bureau of Standards Journal of Research vol. 3, RP90.
13. Rosa, Edward, B.; *Calculation of the Self-Inductance of Single-Layer Coils*; National Bureau of Standards, Bulletin 2; 1906.
14. Knight, David; *An Introduction to the Art of Solenoid Inductance Calculation With Emphasis on Radio-Frequency Applications*; Version 0.20; 2016.  
<http://www.g3ynh.info/zdocs/magnetics/Solenoids.pdf>  
(Downloaded on 2016-02-25)
15. Weaver, Robert S.; *Investigation of E.B. Rosa's Round Wire Mutual Inductance Correction Formula*; 2008.  
[http://electronbunker.ca/DLpublic/Rosa\\_Derivation.pdf](http://electronbunker.ca/DLpublic/Rosa_Derivation.pdf)  
(Downloaded on 2016-02-25)
16. Abramowitz, Milton and Stegun, Irene A.; *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*; National Bureau of Standards, 1964.  
[http://people.math.sfu.ca/~cbm/aands/abramowitz\\_and\\_stegun.pdf](http://people.math.sfu.ca/~cbm/aands/abramowitz_and_stegun.pdf)  
(Downloaded on 2016-02-25)
17. Weaver, Robert S.; *The Inductance of a Helix of Any Pitch*; 2011.  
<http://electronbunker.ca/DLpublic/HelicalInductance.pdf>  
(Downloaded on 2016-02-28)
18. Grover, Frederick W.; *Additions to the Formulas for the Calculation of Mutual and Self Inductance*; National Bureau of Standards, Bulletin Volume 14, No. 4, July 12, 1919.

## Appendices

### A – Program Listing for calculation of the GMD of an Elliptical Locus

```
Program GMDelliptLine (n As int64)
'Calculate self gmd of elliptical lines of multiple aspect ratios.
'N is number of random pairs of points to be generated.
dim a,x1,y1,x2,y2,theta1,theta2,gmd As Double
dim i,j As Int64
dim aspect,aspectAry() As Integer
aspect = ef_pitch.Text.Val
'Load the 'aspectAry()' array with the aspect ratios to be used.
aspectAry = array(20,30,63,125,250,315,325,350,400,500,750)
Print "Aspect, gmd"
for j = 0 to Ubound(aspectAry)
    aspect = aspectAry(j)
    a = aspect/1000
    gmd = 0
    for i = 1 to n
        theta1 = ArcToAngle(rnd,aspect)
        theta2 = ArcToAngle(rnd,aspect)
        x1 = a*cos(theta1)
        y1 = sin(theta1)
        x2 = a*cos(theta2)
        y2 = sin(theta2)
        gmd = gmd+log(sqrt((x1-x2)^2+(y1-y2)^2))
    next
    gmd = exp(gmd/n)
    Print aspect, gmd
next
end

function ArcToAngle (arcLength As Double, aspect as Integer)As Double
'Given the partial arc length of an ellipse
' and aspect ratio, return the parametric angle.
'Note that the argument 'aspect' is an integer and is equal to the aspect ratio
' multiplied by 1000 to simplify its use as the parameter of the case statements.
dim k1,k2,k3,k4,k5,k6,k7,a,b,c,d,f As Double
'Determine quadrant transformation parameters according to arc length.
if arcLength<0.25 then
    'Quadrant 1
    a = arcLength
    b = 0 'Angle offset
    c = 1 'sign of function
    d = 0 'function adjustment
ElseIf arcLength<0.5 then
    'Quadrant 2
    a = 0.5-arcLength
    b = pi/2 'Angle offset
    c = -1 'sign of function
    d = pi/2 'function adjustment
ElseIf arcLength<0.75 then
    'Quadrant 3
    a = arcLength-.5
    b = pi 'Angle offset
```

```

c = 1 'sign of function
d = 0 'function adjustment
Else
  'Quadrant 4
  a = 1-arcLength
  b = 1.5*pi 'Angle offset
  c = -1 'sign of function
  d = pi/2 'function adjustment
end if
a = a*4 'scale up the argument from the range 0< = a<0.25 to 0< = a<1
'Choose the mapping function parameters according to the aspect ratio.
select case aspect
Case 20
  'Aspect = 0.0200
  k1 = 0.9954197523
  k2 = -1.5960238734
  k3 = 0.5221336302
  k4 = 0.0804990919
  k5 = -1.6485276059
  k6 = 0.5239512123
  k7 = 0.1258679037
Case 30
  'Aspect = 0.0300
  k1 = 0.9984223767
  k2 = -1.5947471825
  k3 = 0.5303252429
  k4 = 0.0687710369
  k5 = -1.6373539169
  k6 = 0.5175107258
  k7 = 0.1216076254
Case 63
  ' Aspect = 0.063
  k1 = 1.0053774155
  k2 = -1.0812115077
  k3 = -0.3381538919
  k4 = 0.4219678890
  k5 = -1.1015803243
  k6 = -0.4182729136
  k7 = 0.5249334748
Case 125
  ' Aspect = 0.125
  k1 = 1.0226488149
  k2 = -0.9357991585
  k3 = -0.3526079227
  k4 = 0.3054989960
  k5 = -0.9158611270
  k6 = -0.5115714778
  k7 = 0.4527321137
Case 185
  ' Aspect = 0.185
  k1 = 1.0440215988
  k2 = -1.7853069002
  k3 = 1.4637691010
  k4 = -0.6049651753
  k5 = -1.7200884720
  k6 = 1.3112327787

```

k7 = -0.5163293003  
Case 250  
' Aspect = 0.250  
k1 = 1.0702064611  
k2 = -1.5638577536  
k3 = 6.7291168067  
k4 = -4.6588962658  
k5 = -1.4784593304  
k6 = 6.2841564621  
k7 = -4.8020393235

Case 275  
' Aspect = 0.275  
k1 = 1.08297603  
k2 = 0.32065304  
k3 = -2.92460709  
k4 = 1.54112028  
k5 = 0.29906132  
k6 = -2.96287518  
k7 = 1.67663668

Case 300  
' Aspect = 0.300  
k1 = 1.09677644  
k2 = 0.31383321  
k3 = -2.92801438  
k4 = 1.52766397  
k5 = 0.29919089  
k6 = -2.96132963  
k7 = 1.66867001

Case 315  
' Aspect = 0.315  
k1 = 1.10471140  
k2 = 0.34291534  
k3 = -2.94324420  
k4 = 1.51517585  
k5 = 0.32691385  
k6 = -2.96787128  
k7 = 1.65340874

Case 325  
' Aspect = 0.325  
k1 = 1.11002732  
k2 = 0.34464876  
k3 = -2.92094714  
k4 = 1.49373693  
k5 = 0.32828082  
k6 = -2.93894663  
k7 = 1.62815121

Case 350  
' Aspect = 0.350  
k1 = 1.12318127  
k2 = 0.00286285  
k3 = -2.28947823  
k4 = 1.20606362  
k5 = 0.01527216  
k6 = -2.31312279  
k7 = 1.32498955

Case 400

```

' Aspect = 0.400
k1 = 1.15094948
k2 = -0.33417431
k3 = -1.63203136
k4 = 0.89775819
k5 = -0.28281778
k6 = -1.65893124
k7 = 0.99427163
Case 500
' Aspect = 0.500
k1 = 1.21132835
k2 = -0.25885181
k3 = -1.52341812
k4 = 0.81547090
k5 = -0.20722133
k6 = -1.48854536
k7 = 0.85143913
Case 750
' Aspect = 0.750
k1 = 1.3815773746
k2 = 0.26848230
k3 = -1.39594071
k4 = 0.91476754
k5 = 0.19716802
k6 = -1.17412808
k7 = 0.72109609
else
'For other cases return linear value.
'This is also the correct case for an aspect ratio of 1.
return arcLength*2*pi
end Select
'Calculate the mapping function value.
f = (k1*a+k2*a^2+k3*a^3+k4*a^4)/(1+k5*a+k6*a^2+k7*a^3)
'Apply the quadrant adjustments to the mapping function value and then return.
return d+c*f+b
end

```

## B – A Method for Finding and Eliminating Errors in Mathematical Derivations

Following is a somewhat crude but effective technique to help eliminate blunders during the derivation of formulae having many terms, using nothing more than BASIC or a similar programming language.

The idea is to take the immediate result of each step of the derivation before any simplification, and code it as a BASIC function. For example, from the section, *GMD of two co-linear lines of equal length*, the result of evaluating the  $A_1$  integral is:

$$\int_r^{r+s} A_1 dx = \left[ s(x-s) \log(x-s) - sx \right]_r^{r+s}$$

This is coded into a program function as follows:

```
function _A1(ByVal s as double, r as double)as double
  x = r+s
  b = s*(x-s)*log(x-s)-s*x
  x = r
  a = s*(x-s)*log(x-s)-s*x
  _A1 = b-a
end function
```

This is quick to do, and once this simple function template is set up, it can then be repeated for every term in the solution with just a bit of copying and pasting. Hence the complete, but messy formula for the GMD of two co-linear lines, coded into BASIC, is:

```
function _A1(ByVal s as double, r as double)as double
  x = r+s
  b = s*(x-s)*log(x-s)-s*x
  x = r
  a = s*(x-s)*log(x-s)-s*x
  _A1 = b-a
end function
function _A2(ByVal s as double, r as double)as double
  x = r+s
  b = 0.5*(x^2-s^2)*log(x-s)-0.5*s*x-0.25*x^2
  x = r
  a = 0.5*(x^2-s^2)*log(x-s)-0.5*s*x-0.25*x^2
  _A2 = b-a
end function
function _B(ByVal s as double, r as double)as double
  x = r+s
  b = s*x
  x = r
  a = s*x
  _B = b-a
end function
```

```

function _C(ByVal s as double, r as double)as double
  x = r+s
  b = 0.5*x^2*log(x)-0.25*x^2
  x = r
  a = 0.5*x^2*log(x)-0.25*x^2
  _C = b-a
end function
function _GMD(ByVal s as double, r as double)as double
  A1 = _A1(s,r)
  A2 = _A2(s,r)
  A = A1-A2
  B = _B(s,r)
  C = _C(s,r)
  D = A-B+C
  _GMD = exp(D/s^2)
end function

```

The very last function, `_GMD`, calls each individual function and sums their results to calculate the complete GMD value. Since no simplification of terms has been done at this stage, the chance of errors is greatly diminished. The functions can be verified by solving the GMD for specific values of  $s$  and  $r$ . For this example, the final formula was already known, because it has previously been published [4]. So, it is easy to verify. In the case of a new derivation, it's not difficult to come up with a ballpark estimate of the correct GMD value, because it will fall somewhere between the distance between the closest points and the distance between the furthest points on the two objects. If a better estimate of the true GMD value is needed, then the Monte Carlo method can be used to calculate a couple of examples. However, errors in the formulae tend to affect the results dramatically. So, it's not difficult to tell when an error occurs.

Once it is verified that the functions are producing correct results, each function can then be simplified, with verification that the program still produces the same correct results at each stage. If an error occurs, then it is easy to pinpoint its location before continuing.

The `_A1` function is first simplified by replacing the variable  $x$  with the limits of integration ( $r+s$ ) and ( $r$ ):

```

function _A1(ByVal s as double, r as double)as double
  b = s*((r+s)-s)*log((r+s)-s)-s*(r+s)
  a = s*((r)-s)*log((r)-s)-s*(r)
  _A1 = b-a
end function

```

Next, the  $a$  and  $b$  assignment statements are combined, paying close attention to the sign of the terms:

```

function _A1(ByVal s as double, r as double)as double
  b = s*((r+s)-s)*log((r+s)-s)-s*(r+s) - s*((r)-s)*log((r)-s)+s*(r)
  _A1 = b
end function

```

With further simplification, eventually we end up with this:

```
function _A1(ByVal s as double, r as double)as double
  b = s*r*log(r)-s^2 - s*(r-s)*log(r-s)
  _A1 = b
end function
```

This simplification is repeated with each function until the complete result is in its simplest form.

## Revision History

2016-03-05: Original Issue

2016-03-06:

- Corrected several typographical errors
- Minor revisions to the text for clarity