In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of Electrical Engineering
The University of British Columbia
Vancouver, Canada

Date FEB 26, 1993
Abstract

Accurate determination of frequency dependent resistance of stranded conductors in transmission lines has not been thoroughly investigated for a wide frequency range of DC to 10 MHz. The main objective of this thesis project has been to write a computer program that calculates the resistance and inductance of stranded conductors in the indicated frequency range. The fundamental idea in the implemented technique is to subdivide the cross section of each strand with circular and straight line segments forming circular and elemental shape subconductors (or elements). Then, assuming uniform current density within each area, resistance, self inductance, and mutual inductances for all subconductors are calculated. The resistances and inductances are placed into a complex impedance matrix which is then reduced through mathematical manipulations to obtain a single complex number that represents the equivalent resistance and inductance of the whole conductor. The advantage of this method is that it automatically considers the skin effect and proximity effect of all of the subconductors in the conductor.

Elemental shape subdivisions are more efficient than circular or rectangular shape subdivisions. In addition, the positioning of the elements within each strand is optimized for accuracy and CPU time.

The electromagnetic transients program EMTP requires the values of the line parameters in the frequency range of DC to 10 MHz, and it uses the "TUBE" approximation to calculate them. As shown in this thesis, the TUBE approximation does not provide accurate results for resistance of stranded conductors above 5 kHz.

The proposed program needs an IBM compatible (80386 or above) personal computer and runs with minimum user intervention.

The results obtained in this research with the Subdivision method suggest that the log(R) versus log(frequency) graph can be approximated by a straight line in its high fre-
quency range. Slopes of graphs for conductors with different number of strands have been calculated, and an exponential formula for resistance calculations above 5 kHz is developed. The resistance value at 5 kHz is found by the TUBE formula, and then using the appropriate slope, the resistance of the stranded conductor is estimated at higher frequencies.
Table of Contents

Abstract ...................................................................................................................... ii
Table of Contents ........................................................................................................ iv
List of Tables ............................................................................................................... vii
List of Figures .............................................................................................................. viii
Definitions, Abbreviations, Symbols, and Units ....................................................... x
Acknowledgment ....................................................................................................... xiii
Dedication .................................................................................................................. xiv

1. Introduction and Literature Search ....................................................................... 1
   1.1. Scope of the Thesis ....................................................................................... 5
   1.2. Historical Background ................................................................................ 9

2. Theoretical Background ....................................................................................... 11
   2.1. Resistance Calculations ............................................................................. 11
      2.1.1. Spiraling ............................................................................................. 11
      2.1.2. Temperature ...................................................................................... 12
      2.1.3. Frequency ......................................................................................... 13
      2.1.4. Current Magnitude .......................................................................... 13
   2.2. Inductance Calculations ............................................................................ 14
      2.2.1. Internal Inductance of a Solid Conductor ........................................ 15
      2.2.2. External Inductance .......................................................................... 16
      2.2.3. Total Inductance (or Self Inductance) ............................................. 17
      2.2.4. Mutual Inductance ........................................................................... 18
      2.2.5. Internal Inductance of the Entire Conductor .................................... 19

3. Program Design .................................................................................................... 20
   3.1. Finding Equal Distance and Equal Angle Strand Pairs ......................... 22
   3.2. Finding Symmetric Strand Pairs ............................................................... 23

iv
A.4.1. 37DSPR Sample Output ................................................. 61
B. Technical Details of 37HZR13 .............................................. 63
   B.1. Locate_Points Function ............................................. 63
   B.2. Find_One_GMD Function ............................................. 63
   B.3. Find_GMR Function ................................................. 64
   B.4. Find_SelfGMD_Ls_And_Resistances Function ..................... 64
   B.5. The GMD and SelfGMD Matrices ................................... 64
       B.5.1. How the GMD Matrix is Constructed ....................... 65
       B.5.2. How the SelfGMD Matrix is Constructed ................... 66
   B.6. Constructing the Huge Z Matrix ................................... 66
   B.7. Find_R_L_For_HUGEZ Function .................................... 67
   B.8. Row-Column Subtraction and Symmetry .......................... 68
   B.9. Reducing the Huge Z Matrix and Saving the Results .......... 68
       B.9.1. 37HZR13 Sample Output ..................................... 69
C. Analytic GMD Formulas .................................................. 72
List of Tables

1. Characteristics of Copper Conductors, Hard Drawn, 97.3% Conductivity .......... 3
2. Characteristics of Aluminum Cable, Steel Reinforced .................................. 4
3. Stranding Factors ......................................................................................... 12
4. Percent Conductivity, Resistivity, and Temperature Constant of Conductor Materials ........................................................................................................... 12
5. THETPoints for $M = 4$ at Different Frequencies .............................................. 34
6. RPoints for $N = 2$ and 3 at Different Frequencies .............................................. 35
7. The $N$ Variable for $M = 4$, 10, and 14 .............................................................. 36
8. Case A with 4% Tolerance ................................................................................. 37
9. Case B with 10% Tolerance ............................................................................... 37
10. Case C with 13% Tolerance ............................................................................. 38
11. Case D with 21% Tolerance ............................................................................. 38
12. High-Frequency Asymptotic Slopes for Different Numbers of Strands .............. 50
14. Contents of the GMD Matrix for $M \_ N = 4 \_ 2$ ................................................. 65
15. Lower Triangular SelfGMD Matrix Saved as a One Dimensional Matrix ............ 66
16. Sample Text Output of 37hr13.exe .................................................................. 69
List of Figures

1. Four Basic Transmission Line Parameters ................................................. 1
2. Cardinal 54/7 ACSR ........................................................................ 2
3. Concentric Stranding .............................................................................. 2
4. Circular and Elemental Shape Elements ...................................................... 5
5. Two Examples of Subdivided Strands ...................................................... 5
6. Equivalent Circuit of the Subdivided Conductor with Identical Return Path .... 6
7. Positioning R and L in the Impedance Matrix .............................................. 7
8. Calculation of the Internal Inductance of a Solid Conductor ...................... 15
9. Calculation of the External Inductance of a Solid Conductor ..................... 16
10. Calculation of Mutual Inductance ............................................................ 18
11. A 54/7 ACSR transmission line with M=1, N=1 ..................................... 21
12. A 30/7 ACSR transmission line with M=4, N=2 ..................................... 22
13. Total Current as the Volume Under the Current Density Curve .................. 25
14. M_N_R_θ = 4_2_1_5 for Each Strand ...................................................... 27
15. M_N_R_θ = 4_2_1_5 with Equal Angular Distance between THETPoints .......... 28
16. Position of RPoints within Elementals ..................................................... 28
17. Two Elements with THETPoints = 4 and RPoints = 1 for GMD Calculations .... 29
18. Calculations of GMD for Elements within One Strand for M_N = 8_4 ............... 30
19. Calculating the GMR of an Elemental .................................................... 32
20. Options A to D in the Automatic Mode of 37hzr13.exe for the Resistance ...... 39
21. Options A to D in the Automatic Mode of 37hzr13.exe for the Inductance ...... 40
22. Mtline and Subdivisions Method (Case A, 4% Tolerance) Results for the Resistance ........................................................................ 42
23. Mtline and Subdivisions Method (Case A, 4% Tolerance) Results for the Inductance ................................................................. 42
24. Inductance by Subdivision-Case-A, TUBE, and 7-TUBE for a 7-Strand Conductor

25. Inductance by Subdivision-Case-A, TUBE, and 37-TUBE for a 37-Strand Conductor

26. Percent Difference between Subdivision and Galloway for the Resistance

27. Percent Difference between Subdivision and Galloway for the Inductance

28. Percent Difference between Subdivision and TUBE for the Resistance

29. Percent Difference between Subdivision and TUBE for the Inductance

30. Percent Difference between Subdivision and 37-TUBE for the Resistance

31. Percent Difference between Subdivision and 37-TUBE for the Inductance

32. Percent Difference between Subdivision and TUBE results of 1, 7, 19, and 37-Strand Conductors for the Resistance

33. Normalized 1, 7, 19, and 37-Strand Subdivision Results for the Resistance

34. Subdivision and Exponential Formula Results of a 19-Strand Conductor for the Resistance

35. Percentage Difference between the Subdivision and Exponential Formula Results of 1, 7, 19, and 37-Strand Conductors for the Resistance

36. Lower Triangular HugeZ Matrix

37. Inside a Rectangular and a Triangular Block of HugeZ

38. Geometric Mean Distance (GMD) Analytic Formula between Two Rectangles

39. Geometric Mean Distance Analytic Formula between Two Elementals within the Same Strand
Definitions, Abbreviations, Symbols, and Units

\[ \leq \text{Symbol for "less than or equal to."} \]

\[ \alpha \text{ Attenuation Constant (Np/m).} \]

\[ A \text{ Area (m}^2\text{).} \]

AAAC All Aluminum Alloy Conductor.

AAC All Aluminum Conductor.

ACAR Aluminum Conductor Alloy Reinforced.

ACSR Aluminum Conductor Steel Reinforced (see Figure 2 on page 2).

\[ \beta \text{ Phase Constant (rad/m).} \]

\[ B \text{ Magnetic Flux Density (Wb/m}^2\text{ or T for Tesla).} \]

\[ C \text{ Capacitance (F).} \]

cmil circular mil = cross sectional area of a wire of 0.001 inch diameter ($\pi/4$ square mil).

CPU Central Processing Unit of a computer.

\[ \delta \text{ Depth of Penetration (see Skin Depth) (m).} \]

\[ ds \text{ Differential Area (m}^2\text{).} \]

Depth of Penetration (see Skin Depth).

\[ \varepsilon \text{ Permitivity} = \varepsilon_0 \varepsilon_r \text{ (F/m).} \]

\[ E \text{ Electric Field Intensity (V/m).} \]

Element A hypothetical conductor within an actual conductor (the same as subconductors, filaments).

Elemental Non-circular shapes obtained by using circles and straight lines to subdivide a circle (see Figure 4 on page 5).

emf (or electromotive force) Rate of change of flux linkages with respect to time (Wb.turn/sec).

\[ \Phi \text{ Magnetic Flux through a surface (Wb).} \]
f Frequency (Hz).

Filament A hypothetical conductor within an actual conductor (the same as subconductors, elements).

γ Propagation Constant = α + jβ (complex).

G Conductance (Mho, Simens (S), or one over ohm (1/Ω)).

GMD Geometric Mean Distance = the geometric mean of all distances between all points within two areas (m).

GMR Geometric Mean Radius or geometric mean distance of an area with itself = the geometric mean of all distances between all points within one area (m).

H Magnetic Field Intensity (A/m).

I Current (A).

J Current Density (A/m²).

λ Number of Magnetic Flux Linkages or Rings in a circuit = "summation or integral of all the elements of flux multiplied by the fraction of the total current linked by each flux element [1]" (Wb turn).

ℓ Length (m).

L Inductance (H for Henrys).

μ Permeability = μ₀μ₉ (H/m).

μ₀ Permeability of Free Space = 4π10⁻⁷ (H/m).

μ₉ Relative Permeability (unitless).

M Number of straight line divisions within each strand.

N Number of circular divisions within each strand.

P Power (W).

Percent Difference between a and b: 100(difference)/(average) = 200(a-b)/(a+b).

Proximity Effect "Tendency of the current to concentrate in the regions of the conductors where the magnetic fields that produce the currents are strongest [2]."
ρ Resistivity (Ω.m) reciprocal of conductivity σ.

R Resistance (Ω).

RPoints Points within each elemental subconductor in the radial direction.

ψ Magnetic Flux through a surface (Wb).

σ Conductivity (S/m) reciprocal of resistivity ρ.

Skin Depth (or Depth of Penetration) Distance from the surface towards the centre of a conductor where the current density decreases to 36.8% of its value at the surface of the conductor (m).

Skin Effect Tendency of the current to crowd towards the conductor surface as frequency increases.

Stranding Factor A constant that the final resistance value is multiplied by to account for increase in the length of the conductor due to stranding (see Table 3, page 12).

Subconductor A hypothetical conductor within an actual conductor (the same as elements, filaments).

Subdivision The method of dividing an area into smaller areas.

THETPoints Points within each elemental subconductor in the angular direction.

V Potential (V).

ω Radian Frequency = 2πf (rad/s).

Z Impedance = R + jωL (Ω).
God is the Light of the heavens and the earth. The likeness of His Light is a Niche wherein is a Lamp: The Lamp in a glass: the glass as it were a glittering star: kindled by a blessed Tree, an Olive that is neither of the East nor of the West, whose oil well-nigh which shine even if no fire touched it: Light upon Light! God guides to His Light whom He will (Quran).

**Acknowledgment**

This work could not be completed without the moral and financial support of my loving parents to whom I owe all my life.

My appreciation is extended to my supervisor Dr. J.R. Marti who kindly contributed his time and efforts in enhancing the scientific value of this thesis.


Thanks are due to my friend and my brother in law Kamran, my fiancé, Minoo, and my sister, Mahtab who provided kind support during my entire graduate studies.

I appreciate the efforts of my dear friend Dr. Morteza Ghomshei who provided useful comments during the drafting of the manuscript.

Finally, I would like to thank Dr. H.W. Dommel and Dr. G.E. Howard for their careful examination of this thesis.
Dedication

This work is dedicated to all the people who work towards the education and awareness of others.
1. Introduction and Literature Search

The transmission line parameters per unit length are the series resistance $R$, series inductance $L$, shunt capacitance $C$, and shunt conductance $G$, as shown in Figure 1.

The series resistance accounts for the ohmic ($Rl^2$) losses, and the series impedance, including resistance and inductive reactance, gives rise to series voltage drops along the line. Transient analysis in general and fault and surge propagation studies in particular, in addition to design studies of stranded transmission line conductors, all require accurate impedance calculations.

The goal of this thesis project is to write a computer program that calculates the series resistance and inductance of stranded conductors (with equal strand diameters), such as ACSR, AAC, AAAC, and ACAR at any given frequency.

Because Aluminum has lower cost and lighter weight than copper, it has replaced copper as the most common conductor metal for overhead transmission lines. One of the most common conductor types is ACSR which consists of layers of aluminum strands surrounding a central core of steel strands (see Figure 2). Each layer is spiraled in opposite direction to its overlaying layer to hold the strands together.
Figure 2. Cardinal 54/7 ACSR [3].

Stranded conductors are easier to manufacture since larger conductor sizes can be obtained by simply adding successive layers of strands. They are also easier to handle and more flexible than solid conductors, especially for the storage of long lengths of cable on reels. The use of steel strands gives ACSR conductors a high strength to weight ratio [3]. Different types of ACSR conductors for transmission lines are shown in Tables 1 and 2. The most common form of strand construction for high voltage cables is the concentrically stranded conductor shown in Figure 3.

Figure 3. Concentric Stranding [8].
<table>
<thead>
<tr>
<th>Size of Conductor</th>
<th>A.W.G. or BS. 5.5</th>
<th>Diameter of Individual Strands (inches)</th>
<th>Outside Diameter (inches)</th>
<th>Breaking Strength (Pounds)</th>
<th>Weight (Pounds per Mile)</th>
<th>Approximate Carrying Capacity (Amps)</th>
<th>Geometric Mean Radius (in)</th>
<th>/Ω Resistance (Ohms per Conductor at 25°C)</th>
<th>/Ω Inductive Reactance (Ohms per Conductor Per Mile at 1 Ft. Spacing)</th>
<th>/Ω Capacitive Reactance (Megohms per Conductor Per Mile at 1 Ft. Spacing)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.0184</td>
<td>0.151</td>
<td>43300</td>
<td>18300</td>
<td>1300</td>
<td>0.0368</td>
<td>0.0626</td>
<td>0.0685</td>
<td>0.0609</td>
</tr>
<tr>
<td>6/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
<tr>
<td>1/00000</td>
<td>37</td>
<td>0.1600</td>
<td>0.102</td>
<td>39310</td>
<td>16470</td>
<td>1120</td>
<td>0.0249</td>
<td>0.0365</td>
<td>0.0399</td>
<td>0.0343</td>
</tr>
</tbody>
</table>

*For conductor at 75°C air at 25°C wind 1 ft. miles per hour (37°F/sec). Frequency = 60 Hz.
<table>
<thead>
<tr>
<th>Code Word</th>
<th>Circular In.</th>
<th>Strand Diameter (Inches)</th>
<th>Outside Diameter (Inches)</th>
<th>Copper Equivalents</th>
<th>Weight (Pounds per Mile)</th>
<th>Average Current Carrying Capacity (Amps)</th>
<th>dc</th>
<th>50 Hz</th>
<th>60 Hz</th>
<th>50 Hz</th>
<th>60 Hz</th>
<th>80% Current Approx. 75% Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aluminum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 2. Characteristics of Aluminum Cable, Steel Reinforced (Aluminum Company of America).*
1.1. Scope of the Thesis

The main objective of this thesis is to write a computer program on an IBM compatible personal computer that calculates the resistance and inductance of stranded conductors in transmission lines. For the calculations, the cross section of each strand is divided into circular and elemental shape subconductors or elements (see Figure 4).

![Figure 4. Circular and Elemental Shape Elements.](image)

Figure 5 shows two examples of subdivided strands.

![Figure 5. Two Examples of Subdivided Strands.](image)

Figure 6 shows how each element is considered as an independent conductor. Each element has its own resistance \( R = 1/(\sigma A) \), where \( \sigma \) = conductivity (S/m) and \( A \) = area of the subconductor \((m^2)\), and self inductance \( L_{self} = \frac{\mu}{2\pi} \ln\left(\frac{2h}{GMR}\right) \), with \( \mu = \mu_r \mu_0 \) = per-
meability, \( h \) = average height of the transmission line, and \( GMR \) = Geometric Mean Radius of the elemental (GMR and GMD are defined later in the thesis).

Each element has also a mutual inductance to other elements in the same strand and in all other strands \( L_{\text{total}} = \frac{\mu}{2\pi} \ln\left(\frac{2h}{GMD}\right) \); where \( GMD \) = Geometric Mean Distance between two subconductors.

![Equivalent Circuit of the Subdivided Conductor with Identical Return Path](image)

Figure 6. Equivalent Circuit of the Subdivided Conductor with Identical Return Path.

Considering the ground as a perfect conductor, the return path is an identical conductor with depth equal to the height of the conductor. And assuming the height of the conductor is much larger than the conductor diameter, there will be no coupling between the going and the returning paths. The loop equations for the path of the first current and through the identical return path in Figure 6 is
\[2V_1 = 2 \left( R_1 + j \omega L_{11} \right) I_1 + j 2 \omega L_{12} I_2 + j 2 \omega L_{13} I_3 + \ldots + j 2 \omega L_{1n} I_n.\]

Dividing both sides of the above equation by 2,
\[V_1 = \left( R_1 + j \omega L_{11} \right) I_1 + j \omega L_{12} I_2 + j \omega L_{13} I_3 + \ldots + j \omega L_{1n} I_n.\]

From which,
\[Z_{11} = R_1 + j \omega L_{11}, \quad Z_{12} = j \omega L_{12}, \quad Z_{13} = j \omega L_{13}, \quad \ldots, \quad Z_{1n} = j \omega L_{1n}.\]

These values constitute the first row of the impedance matrix. By similar relations for the \(I_2\) to \(I_n\) current paths, impedance values for other rows are derived. These impedance values are placed into a huge complex impedance matrix \([Z]\) (Figure 7) to form \([V] = [Z] [I]\).

![Figure 7. Positioning R and L in the Impedance Matrix.](image)

Then the first row of \([Z]\) is subtracted from all other rows so that the equation
\[
\begin{bmatrix}
V \\
V \\
V \\
0 \\
0 \\
0
\end{bmatrix} =
\begin{bmatrix}
I_1 \\
I_2 \\
I_3 \\
I_4 \\
I_5 \\
I_n
\end{bmatrix}
\]

becomes
\[
\begin{bmatrix}
V \\
0 \\
0 \\
0
\end{bmatrix} =
\begin{bmatrix}
I_1 \\
I_2 \\
I_3 \\
I_n
\end{bmatrix}.
\]
The voltage vector has now one nonzero element only. The first column of \([Z']\) is then subtracted from all other columns. This subtraction, in effect, replaces \(I_1\) by

\[
I_{\text{total}} = I_1 + I_2 + I_3 + \ldots + I_n
\]

and the above equation becomes

\[
\begin{bmatrix}
V \\
0 \\
0 \\
\vdots \\
0 \\
I_n
\end{bmatrix}
= \begin{bmatrix}
Z'' \\
I_2 \\
I_3 \\
\vdots \\
I_n
\end{bmatrix}
\]

Current \(I_n\) is now exchanged with the \(n\)th element (\(n\)th zero) of vector \([V]\), with the result

\[
\begin{bmatrix}
V \\
0 \\
0 \\
\vdots \\
0 \\
I_n
\end{bmatrix}
= \begin{bmatrix}
Z''_n \\
I_2 \\
I_3 \\
\vdots \\
I_{n-1} \\
0
\end{bmatrix}
\]

Current \(I_{n-1}\) is now exchanged with the \((n-1)\)th element of vector \([V]\), and this process continues until the following result is obtained [4]:

\[
\begin{bmatrix}
V \\
I_2 \\
I_3 \\
\vdots \\
I_{n-1} \\
I_n
\end{bmatrix}
= \begin{bmatrix}
Z_{\text{new}} \\
0 \\
0 \\
\vdots \\
0 \\
0
\end{bmatrix}
\]

The first row of \([Z_{\text{new}}]\) times the \([I]\) vector is in effect element \((1,1)\) of \([Z_{\text{new}}]\) times \(I_{\text{tot}}\) which is equal to \(V\).

Impedance element \((1,1)\) of \([Z_{\text{new}}]\) is exactly the complex number \(Z_{11}\) that is wanted.

\(Z_{11} = R_{11} + j\omega L_{11}\) where \(R_{11}\) is the resistance per metre of the conductor, and \(L_{11}\) is the inductance per metre.
During the above procedure, the \([Z]\) matrix is changed in each step. In the computer program, those changes are applied to the stored \([Z]\) matrix step by step, and there is no need to carry along the \([V]\) and \([I]\) vectors. Detailed explanations can be found in Appendix B.9. on page 68.

1.2. Historical Background

The "Line Constants" support routine of the electromagnetic transients program EMTP [5] calculates the series resistance and internal inductance of overhead transmission lines. According to Marti [6], these line parameters have to be calculated typically from DC to 10 MHz for frequency-dependent line models.

Lewis and Tuttle [7] use the same TUBE approximation as in the Line Constants routine to simulate the skin effect in stranded conductors. In this approximation, the aluminum or copper strands are replaced by a solid conductor, and in the case of ACSR conductors, the inner steel strands are ignored. To calculate the impedance of a solid annular conductor as a function of frequency, analytical skin effect formulas are used.

According to Graneau [8], because no closed-form analytical solutions are available for impedance calculations of stranded conductors considering the skin effect, cable manufacturers have to rely on measurements of sample conductors. Some measurements are reported in Barrett, Nigol, Fehervari, and Findlay [9] and Morgan and Findlay [10] for stranded conductors at power frequencies, but no data on wide frequency range measurements is available.

Approximate formulas are presented by Malkov and Pavlov [11] for the impedance of multiwire conductors at frequencies from 10 MHz to \(10^{4}\) MHz, but this is beyond the range of interest for power system studies.
Galloway, Shorrocks, and Wedepohl [12] have presented an asymptotic formula for stranded conductors, but no verification of this formula for an extended frequency range has been reported. In this thesis, it is shown that the Galloway's formula does not give accurate results for an extended frequency range, and the TUBE formula is satisfactory only up to 5 kHz.

The concept of conductor subdivisions adopted in this work was earlier proposed by Comellini, Invernizzi, and Manzoni [13] who obtained relatively good results using circular subconductors. The technique was also used by Weeks, Wu, McAllister, and Singh [14] for rectangular conductors with good agreement between calculated and measured values over a wide frequency range from 5 kHz to 500 kHz (with a maximum error of 21% for the resistance at 500 kHz). Lucas and Talukdar [15] showed that the use of elemental-shaped subconductors (Fig. 4 on page 5) was much more efficient than using simple circular filaments (for frequencies up to 100 Hz, 36 elementals give the same accuracy as 450 circular filaments and for frequencies up to 10 kHz, 36 elementals give the same accuracy as 1007 circular filaments). Arizon and Dommel [16] used circular, rectangular, and elemental shape subconductors. They also concluded that for a given accuracy, the elemental shape subdivision method was more efficient in terms of computer CPU time and memory requirements than the other two subdivision methods. This elemental shape is the one used in the program presented in this work. Wu, Kuo, and Chang [17] used circular and rectangular subdivision for impedance computations of three dimensional multiconductor interconnection structures. Their numerical results are in good agreement with the measurement data and the available results in the literature.
2. Theoretical Background

The following assumptions are made in all calculations:

1. The current flows longitudinally in all elements.
2. The current density is constant within each element.
3. The conductivity of each element is constant and in the case of ACSR conductors, the steel strands may have a different conductivity from the aluminum strands.
4. All strands have equal diameters.
5. All strands have equal relative permeabilities.

2.1. Resistance Calculations

The formula for dc resistance of a solid conductor with circular cross section and length \( \ell \) metres is:

\[
R_{dc} = \frac{\rho \cdot \ell}{A}, \quad \text{where} \quad \rho = \text{resistivity} \ (\Omega \ m), \quad \ell = \text{length} \ (m), \quad A = \text{cross sectional area} \ (m^2),
\]

and \( \sigma = \frac{1}{\rho} = \text{conductivity} \ (S/m) \).

\[ R_{dc} / \ell = \frac{\rho}{A} \ (\Omega/m) \]  

= dc resistance per metre of the conductor.


2.1.1. Spiraling

Because of the spiraling of the strands, the current flows through a helical path in each strand. This makes the strands longer and slightly increases their dc resistance (because of tough oxide coating, there is no interwire conduction). In normal resistance calculations, the helical path is ignored, and the conductor's longitudinal length and normal cross sectional areas of the wires are used. The final result is then multiplied by a stranding factor.
that depends on the size of the conductor [8] (Table 3). In the computer program, the final resistance value is multiplied by the appropriate stranding factor. However, for comparison purposes, in none of the graphs of the subdivision method shown in the results section, stranding factors are used.

Table 3. Stranding Factors.

<table>
<thead>
<tr>
<th>No. of Wires in Conductor</th>
<th>7</th>
<th>19</th>
<th>37</th>
<th>61</th>
<th>91</th>
<th>127</th>
<th>169</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stranding Factor</td>
<td>1.145</td>
<td>1.054</td>
<td>1.028</td>
<td>1.017</td>
<td>1.011</td>
<td>1.008</td>
<td>1.006</td>
</tr>
</tbody>
</table>

2.1.2. Temperature

The resistivity varies with temperature as

$$\rho_r = \rho_r \left( \frac{T_2 + T}{T_1 + T} \right)$$

where $\rho_r$ and $\rho_r$ are resistivities at temperatures $T_2$ and $T_1$ in °C, respectively, and $T$ is a temperature constant that depends on the conductivity of the material, as listed in Table 4.

Table 4. Percent Conductivity, Resistivity, and Temperature Constant of Conductor Materials [3].

<table>
<thead>
<tr>
<th>Material</th>
<th>Percent Conductivity</th>
<th>Resistivity at 20 °C in $\Omega$ m $\times 10^8$</th>
<th>Resistivity at 20 °C in $\Omega$ cmil/ft</th>
<th>T Temperature Constant °C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annealed Copper</td>
<td>100%</td>
<td>1.72</td>
<td>10.37</td>
<td>234.5</td>
</tr>
<tr>
<td>Hard-drawn Copper</td>
<td>97.3%</td>
<td>1.77</td>
<td>10.66</td>
<td>241.5</td>
</tr>
<tr>
<td>Hard-Drawn Aluminum</td>
<td>61%</td>
<td>2.83</td>
<td>17.00</td>
<td>228.1</td>
</tr>
<tr>
<td>Material</td>
<td>Percentage</td>
<td>Conductivity</td>
<td>Tension</td>
<td>Yield Strength</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>--------------</td>
<td>---------</td>
<td>---------------</td>
</tr>
<tr>
<td>Brass</td>
<td>20–27%</td>
<td>6.4–8.4</td>
<td>38–51</td>
<td>480</td>
</tr>
<tr>
<td>Bronze</td>
<td>9–13%</td>
<td>13–18</td>
<td>78–108</td>
<td>1980</td>
</tr>
<tr>
<td>Iron</td>
<td>17.2%</td>
<td>10</td>
<td>60</td>
<td>180</td>
</tr>
<tr>
<td>Silver</td>
<td>108%</td>
<td>1.59</td>
<td>9.6</td>
<td>243</td>
</tr>
<tr>
<td>Sodium</td>
<td>40%</td>
<td>4.3</td>
<td>26</td>
<td>207</td>
</tr>
<tr>
<td>Steel</td>
<td>2–14%</td>
<td>12–88</td>
<td>72–530</td>
<td>180–980</td>
</tr>
</tbody>
</table>

In the Subdivision program, the user has to enter the correct conductivity of aluminum and steel at the desired temperature.

### 2.1.3. Frequency

The AC resistance (or "effective resistance") of a conductor can be defined as

\[ R_{ac} = \frac{P_{\text{loss}}}{|I|^2} \text{ (\Omega)} \]

where \( P_{\text{loss}} \) is the conductor's real power loss in watts and \( I \) is the rms (root mean square) of the current. As frequency increases, because of the skin effect, the conductor loss increases and by the above equation, the ac resistance increases (A mathematical proof of this phenomena is given in Graneau [8] book). So, one expects the resistance of the entire stranded conductor to increase as frequency increases, and this is what is observed from the obtained results.

### 2.1.4. Current Magnitude

For magnetic conductors such as steel conductors used for shield wires, resistance depends on the magnitude of the current. The internal flux linkages (and therefore the losses in the iron) depend on the current magnitude. For ACSR conductors, the steel core has a relatively low conductivity compared to aluminum strands (steel conductivity = 1.0e6...
(S/m), aluminum conductivity = 34.7e6 (S/m)) and therefore, the effect of the current magnitude on the resistance of the steel strands is small.

In the computer program, the steel conductivity is used to calculate the resistance of steel strands and the effect of the current magnitude is not considered. In Table 2 (on page 4) resistance values at two current levels are shown for frequencies below 60 Hz.

2.2. Inductance Calculations

Induced voltage (or electromotive force) emf in a circuit is equal to the rate of change of flux linkages with respect to time, i.e., $e = \text{emf} = \frac{d\lambda}{dt}$

where $e =$ induced voltage(volts), and $\lambda =$ flux linkages (Wb.turn).

The induced voltage $e$ is also proportional to the rate of change of current with respect to time and the constant of proportionality is the inductance, i.e.,

$\text{emf} \propto \frac{di}{dt} \Rightarrow e = L \frac{di}{dt}$;

$\text{emf} = d\lambda / dt \Rightarrow d\lambda / dt = L \frac{di}{dt} \Rightarrow L = \frac{d\lambda}{di}$.

Inductance is the rate of change of flux linkages with respect to current. For constant permeability $\mu$, $L = \frac{\lambda}{i}$ is constant.

To find the inductance of a magnetic circuit with constant permeability, one has to find:

1. Magnetic field intensity $H$ (A/m) from Ampere's law: $\oint _c \vec{H} \cdot d\vec{l} = i_{\text{enclosed}}$.

2. Magnetic flux density $B = \mu H = \mu_0 \mu_r H$.

3. Flux linkages $\lambda$ (Wb.turn) using $\int _S \vec{B} \cdot d\vec{s} = \psi_m$ (through surface S).

4. Inductance using $L = \frac{\lambda}{i}$.
To derive the formula used in the inductance calculations in the computer program, the internal, external, and the total (or self) inductance of a solid cylindrical conductor have to be calculated.

2.2.1. Internal Inductance of a Solid Conductor

Figure 8 shows a one metre section of a solid cylindrical conductor with radius $r$ carrying current $I$. For simplicity, the following assumptions are made:

1. the conductor is sufficiently long that the end effects are neglected,
2. the conductor is non magnetic ($\mu = \mu_0 = 4\pi \times 10^{-7}$ H/m), and
3. the conductor has a uniform current density (skin effect is neglected).

The magnetic field intensity at a distance $x$ from the centre of the cylindrical conductor can be calculated from Ampere's law as $\int_c \vec{H} \cdot d\vec{l} = i_{\text{enclosed}}$.

The enclosed current is equal to the total current times the ratio between the area of circle $c$ and the total area of the conductor. Therefore,

$$i_{\text{enclosed}} = \frac{2\pi x^2 I}{\pi r^2} = \int_c H_{\text{mag}}\,dl = 2\pi x \cdot H_x \Rightarrow H_x = \frac{\frac{x}{r} \cdot I}{2\pi x} = \frac{xI}{2\pi r^2};$$

$$B_x = \mu_0 \mu_r H_x; \quad \psi_m = \int_S \vec{B} \cdot d\vec{s}.$$
The differential flux through \( ds = d\psi = B_xl\,dx = \frac{\mu xl}{2\pi r^2}\,ldx \).

Since only a fraction of \( I \) is linked by the flux through \( ds \), the differential flux linkage \( d\lambda \) is ratio of areas times the differential flux through \( ds \). That is,

\[
d\lambda = \left(\frac{\pi x^2}{\pi r^2}\right)d\psi = \frac{x^2}{r^2} \frac{\mu xl}{2\pi r^2}\,ldx = \frac{\mu xl^3}{2\pi r^4}\,dx ,
\]

\[
\lambda_{\text{int}} = \int d\lambda = \frac{\mu l^3}{2\pi r^4}\int_{x=0}^{x=r} dx = \frac{\mu l^3}{8\pi} ,
\]

\[
L_{\text{int}} = \frac{\lambda_{\text{int}}}{I} = \frac{\mu l^3}{8\pi} \Rightarrow L_{\text{int}} = \frac{\mu}{8\pi} = \frac{\mu_h \mu_r(\text{int})}{8\pi} (H/m) .
\]

### 2.2.2. External Inductance

The external inductance of a solid conductor of radius \( r \) up to a point \( P \) at a distance \( D \) from the centre of the conductor can be calculated as (Figure 9):

![Diagram of external inductance](attachment:External_Inductance_Diagram.png)

**Figure 9. Calculation of the External Inductance of a Solid Conductor.**

\[
\int_c \mathbf{H} \cdot d\ell = i_{\text{enclosed}} ; \quad H_x 2\pi x = I ; \quad H_x = \frac{I}{2\pi x} ; \quad B_x = \mu H_x = \frac{\mu I}{2\pi x} ;
\]

\[
\int_s \mathbf{B} \cdot ds = \psi_{\text{thr}} ; \quad B\ell\,dx = d\psi
\]
Here, because path c encloses the entire conductor, the differential flux linkage \( (d\lambda) \) and the differential flux through \( ds \) \( (d\psi) \) are equal. Then,

\[
d\lambda = d\psi = B\ell dx = \frac{\mu \ell I}{2\pi x} \; dx ; \quad \lambda_{\text{ext}} = \int d\lambda = \int B\ell dx = \int_{x=r}^{x=D} \frac{\mu \ell I}{2\pi x} \; dx = \frac{\mu \ell I}{2\pi} \int_{x=r}^{x=D} \frac{1}{x} \; dx = \frac{\mu \ell I}{2\pi} \ln \left( \frac{D}{r} \right);
\]

and

\[
\frac{L_{\text{ext}}}{\ell} = \frac{\lambda_{\text{ext}}}{I\ell} = \frac{\mu}{2\pi} \ln \left( \frac{D}{r} \right) = \frac{\mu_0 \mu_r(\text{external})}{2\pi} \ln \left( \frac{D}{r} \right).
\]

### 2.2.3. Total Inductance (or Self Inductance)

Combining the internal and external inductances,

\[
L_{\text{int}} = \frac{L_{\text{ext}}}{8\pi}
\]

\[
L_{\text{ext}} = \frac{\mu_0 \mu_r(\text{ext})}{2\pi} \ln \left( \frac{D}{r} \right).
\]

The total inductance of a solid circular conductor is given by

\[
L_{\text{total}} = L_{\text{self}} = L_{\text{int}} + L_{\text{ext}} = \frac{\mu_0}{2\pi} \left[ \frac{\mu_r(\text{ext})}{4} + \mu_r(\text{ext}) \ln \left( \frac{D}{r} \right) \right] =
\]

\[
= \frac{\mu_0 \mu_r(\text{ext})}{2\pi} \left[ \frac{\mu_r(\text{int})}{4} + \ln \left( \frac{D}{r} \right) \right] = \frac{\mu_0 \mu_r(\text{ext})}{2\pi} \left[ \ln \left( e^{\frac{\mu_r(\text{int})}{4\mu_r(\text{ext})}} \right) + \ln \left( \frac{D}{r} \right) \right] =
\]

\[
= \frac{\mu_0 \mu_r(\text{ext})}{2\pi} \ln \left( \frac{D}{r e^{\frac{\mu_r(\text{int})}{4\mu_r(\text{ext})}}} \right) = \frac{\mu_0 \mu_r(\text{ext})}{2\pi} \ln \left( \frac{2h}{GMR} \right).
\]

Since \( \mu_r(\text{ext}) = \mu_r(\text{int}) = 1.0 \), then \( GMR = r e^{-1} \).

For an elemental shape filament, the GMR is calculated by positioning points within the elemental and finding the Geometric Mean Distance between these points. This is explained in section 3.5.4.
2.2.4. Mutual Inductance

Figure 10 shows two independent loops with their return paths in the ground at a depth equal to the height of the conductor.

Applying Ampere's law:

\[ \oint_{C} \mathbf{H} \cdot d\mathbf{l} = i_{\text{enclosed}}; \ 2\pi x H_x = I_j; \ H_x = \frac{I_j}{2\pi x}; \ B_x = \mu H_x; \]

\[ \int_{S} B \cdot ds = \psi_{\text{in}}; \ d\psi = \ell B_x dx = \ell \mu H_x dx = \frac{\ell \mu I_j}{2\pi x} dx; \ d\lambda = d\psi \Rightarrow \lambda = \int d\psi; \]

\[ \lambda_{ij} = \int_{x=d}^{x=D} \frac{\ell \mu I_j}{2\pi} \frac{1}{x} dx = \frac{\ell \mu I_j}{2\pi} \ln \left( \frac{D}{d} \right) = \text{flux linkages in loop } iq_i \text{ due to current in loop } jq_j. \]

\[ \frac{L_{ij}}{\ell} = \frac{\lambda_{ij}}{I_j \ell} = \frac{\mu}{2\pi} \ln \left( \frac{D}{d} \right) = \frac{\mu}{2\pi} \ln \left( \frac{D}{\text{GMD}} \right) \ (H/m) \]

where D= distance of conductor \( j \) to the return path of conductor \( i \), and d= GMD between conductors \( i \) and \( j \).
Similarly, \( \frac{L_{ji}}{\ell} = \frac{\mu}{2\pi} \ln \left( \frac{D}{d} \right) \) since the distance between conductor j and the return path of i is the same as the distance between conductor i and the return path of j (Figure 10, page 18).

The above formula is used in the computer program to calculate the mutual inductance between stranded conductors. Stevenson [18] considers the mutual inductance caused by both the going and the returning paths, and his formula is twice this formula.

2.2.5. Internal Inductance of the Entire Conductor

The inductance value provided by the program represents the total inductance of the stranded conductor. To obtain the internal inductance only, the external inductance of the entire conductor has to be subtracted from the final result. In section 2.2.2, the external inductance of a solid conductor to a point at distance D was derived as

\[
L_{\text{ext}} = \frac{\mu_0 \mu_r(\text{ext})}{2\pi} \ln \left( \frac{D}{r} \right).
\]

The external inductance of the entire conductor is

\[
L_{\text{ext}} = \frac{\mu_0 \mu_r(\text{ext})}{2\pi} \ln \left( \frac{2h}{r_{\text{total}}} \right),
\]

where \( h \) is the height of the conductor, and \( r_{\text{total}} \) is the external radius of the stranded conductor. In the program, the external inductance is subtracted from the final value of inductance and the result is the internal inductance of the entire conductor. However, the graphs shown in this thesis comparing the inductance obtained with the subdivision method with the results from TUBE and Galloway's formulas from the EMTP Line Constants program are in terms of the total inductance for a certain height.
3. Program Design

The computer program is written in the C++ language. The program considers the case of stranded conductors in transmission lines. Figure 11 (on page 21) shows a 54/7 Canary ACSR transmission line conductor with no subdivisions within each strand. Figure 12 (on page 22) shows a 30/7 Eagle ACSR transmission line with 4 straight line divisions (M=4) and 2 circular divisions (N=2).

The Zortech C++ compiler for personal computers [19] was used to compile the various C++ programs. All the *.CPP files were compiled with the X memory model. This model needs an 80386 or an 80486 PC to work. The advantage of the X memory model is that it can use up to 16 MBytes of extended memory if the program needs to. For some very large cases (such as the M_N = 4_9 case), the program needs up to 13.4 MBytes of memory.

The conductor strands (Figure 11) are numbered starting from zero in a counterclockwise direction, and the subconductors (elements) within each strand are numbered the same way. To refer to a specific subconductor, the notation "element number , strand number" is used. For example, element number 2 of strand 12 in Figure 12 is referred to as 2, 12 and element 4 of strand 36 as 4, 36.

Advantage is taken of the symmetry of the strands in two ways:

1. By finding strand pairs of equal distance and angle.
2. By finding strand pairs that are symmetric with respect to x and y axis, and with respect to origin.
Figure 11. A 54/7 ACSR transmission line with $M=1, N=1$.

Taking advantage of this symmetry, it results that instead of finding 666 GMD's between all strand pairs in a 30/7 ACSR, only 161 GMD's are calculated, and all the others are equal to one of these 161 values.
3.1. Finding Equal Distance and Equal Angle Strand Pairs

Looking at Figure 11 on page 21, one observes that strands 1 to 6 are at radius 2r, and that their angle increases from zero to $5\pi/3$ (radians) in $\pi/3$ increments. Strands 7 to 18 are at radius 4r, and their angle increases from zero to $11\pi/6$ in $\pi/6$ increments. Strands 19 to 36 are at radius 6r, and their angle increases from zero to $17\pi/9$ in $\pi/9$ increments. Strands 37 to 60 are at radius 8r, and their angle increases from zero to $23\pi/12$ in $\pi/12$ increments. These strand positions are passed on to the program.
Appendix A.1. on page 57 describes the 37DIST.EXE program to exploit this symmetry. Appendix A.1.1. on page 58 shows a sample screen output from the program.

Another program, 37xy.exe, stores the strand numbers for the x, y, and origin symmetric pairs in a binary file to be used for subsequent processing. Appendix A.2. on page 59 has the screen output of this file.

### 3.2. Finding Symmetric Strand Pairs

A program was written to take one pair (out of 666) at a time, and find its x, y, and origin symmetric pairs. It creates some groups and puts the symmetric strand names in those groups. When finding symmetric pairs, if a pair is already found in one of the previous groups, that pair is ignored, and the next pair is considered. If a pair has not already been found and if it is a new pair, it is placed in a new group, or in a group with x, y, or origin symmetry. The advantage of this program is that considering any of the 666 strand pairs in a 37-strand conductor, that pair itself, or its x, or y, or origin symmetric pair will be the first pair of one of the above groups. These first pairs are placed in an array and saved in a binary file on disk to be used by the 37DSPR.EXE program.

Appendix A.3. on page 59 includes a detailed explanation of this program, and Appendix A.3.1. on page 60 shows a sample screen output from the program.

### 3.3. Combining Equal-Distance-Angle and Symmetric Strand Pairs

The program 37DSPR.EXE takes the output from 37DIST.EXE program (the file 37DIST.DAT that has all the strand pairs with equal distant and angle with each other) and the output from program 37PAIR.EXE (the file 37PAIR.DAT with all of the first strand pairs of groups with x, y, or origin symmetry) and combines the two. For each group in 37DIST.DAT, the program creates a new group, if necessary, by checking if the
pair in 37DIST.DAT is in 37PAIR.DAT or not. If it is, it puts that strand pair in a new group, say group el; if it is not, it ignores that pair and goes on to the next pair in 37DIST.DAT. If none of the strand pairs in 37DIST.DAT is in 37PAIR.DAT, the program does not do anything.

The output from 37DSPR.EXE (the binary file 37DSPR.DAT) contains groups of strand pairs such that the pairs in each group have equal distance and angle with each other, and because they are derived from the 37PAIR.DAT file, all other strand pairs are x, y, or origin symmetry of these pairs.

The final program, 37HZR13.EXE needs to find the Geometric Mean Distance (GMD) between the first pairs of the 37DSPR.DAT groups. Due to the indicated symmetry pre-processing, for a 37-strand conductor, 37HZR13.EXE program has to calculate only 161 GMD's instead of 666 total values.

Appendix A.4. on page 61 includes a detailed explanation of the 37DSPR.EXE program, and Appendix A.4.1. on page 61 shows a sample screen output from this program.

Appendix A.5. on page 62 briefly explains what the 37HZR13.EXE program does before constructing the HUGEZ matrix. Appendix B. on page 63 and its subsections have detailed explanations on the different functions of the 37HZR13.EXE program.

3.4. Equal Current Criteria for Location of Circular Subdivisions

It is straightforward for the 37HZR13.EXE program to divide the cross section of each strand by even number of straight line divisions (M): it positions each straight line at 360/M degrees. But for circular divisions, there is no straightforward method.

The criteria adopted to position N circles within each strand is that of equal current. First, the total current passing through the conductor is calculated, and then, the N circles
are positioned such that the amount of current going through the central circle and through each annular section is equal to \((1/N)\) of the total current [20].

The electric field intensity in a conductor is \(E = E_0 e^{-\alpha z} e^{i(\omega t - \beta z)} a_x\) where \(\alpha\) is the attenuation constant, \(\beta\) is the phase factor, and \(z\) is the direction of propagation. In a good conductor (i.e., \(\sigma \gg \omega \varepsilon\)), \(\alpha = \beta = \sqrt{\pi \mu |\sigma|} = 1/\delta [21]\), where \(\sigma\) is the conductivity of the conductor, \(\omega = 2\pi f\), \(\varepsilon\) is the permittivity, \(\mu\) the permeability, and \(\delta\) is the skin depth (or depth of penetration) equal to \(\frac{1}{\sqrt{\pi \mu |\sigma|}}\).

In terms of the depth of penetration \(\delta\), \(E = E_0 e^{-z/\delta} e^{i(\omega t - z/\delta)} a_x\) and \(|E| = E_0 e^{-z/\delta}\). The corresponding current density is given by \(J = \sigma E\); \(|J| = \sigma E_0 e^{-z/\delta} = J_0 e^{-z/\delta}\). The total current that passes through the cross section of each strand is equal to the volume created by rotating the above current density vector \(|J|\) around the strand's radius axis \(z = r\) (see Figure 13).

From Figure 13 and using the method of cylindrical shells [22] to find the volume,

\[
\text{volume} = \int (2\pi \text{ radius})(\text{height})(\text{thickness}) = \int_{z=0}^{z=r} 2\pi (r-z)(|J|)(dz) = \int_{z=0}^{z=r} 2\pi (r-z) J_0 e^{-z/\delta} dz
\]

\[= \text{total current} = 2\pi J_0 \delta (r + \delta e^{-\delta/\delta} - \delta).\]

![Figure 13. Total Current as the Volume Under the Current Density Curve.](image-url)
The amount \((1/n)\) of the total volume is confined in the region from \(z = 0\) to \(z = z_1\) \((z_1\text{ is between zero and } r)\), and the radius of the circle through which \((\text{total current})/n\) is passing will be \(r - z_1\). Therefore,

\[
(1/n) \text{ total current} = \int_{z=0}^{z=z_1} 2\pi (r - z) J_0 e^{\frac{-z}{\delta}} \, dz = 2\pi J_0 \delta \left[ r + (z_1 + \delta - r) e^{\frac{-z_1}{\delta}} \right].
\]

Letting \((1/n)\)th of the total current be equal to the above result by combining the above two equations, one obtains:

\[
(n-1) (r - \delta) - \delta e^{\frac{-r}{\delta}} + n(z_1 + \delta - r) e^{\frac{-z_1}{\delta}} = 0.
\]

Because \(z_1\) is in polynomial form as well as exponential, iteration methods have to be used to solve for it. Using Newton's iteration formula \([22]\), \(z_1\) is found to be:

\[
z_{1\text{new}} = z_{1\text{old}} - \frac{(r - \delta)(n-1) - \delta e^{\frac{-r}{\delta}} + p_1}{n(e^\delta - p_1/\delta)},
\]

where \(e^\delta\), and \(p_1 = n(z_{1\text{old}} + \delta - r)(e^\delta)\).

With three to four iterations, identical answers for \(z_1\) up to seven decimal points are obtained.

After finding \(z_1\), the radius of the circle inside each strand is \(\text{strand_radius} - z_1\). The constant \(n\) is replaced by

\[
N \quad N/2 \quad N/3 \quad \ldots \quad N/(N-4) \quad N/(N-3) \quad N/(N-2) \quad N/(N-1)
\]

to find \(z_1\) first. The \(r - z_1\) value will then be equal to

\[
R[N-2] \quad R[N-3] \quad R[N-4] \quad \ldots \quad R[3] \quad R[2] \quad R[1] \quad R[0]
\]

respectively. For \(N=4\), there are a total of four circles including the body of the strand. The radius of the innermost circle is \(R[0]\), and the radius of the outermost circle is \(R[N-1] = \text{strand_radius}\).

This method of positioning the circles automatically makes the distance between two adjacent circles smaller as one moves towards the surface of the strand.
3.5. GMD and GMR Calculations

To calculate the GMD between different elements, points are positioned within those elements and the geometric mean distances between those points are calculated.

Figure 14 shows the cross section of one strand with $M = 4$ (number of straight line divisions within each strand), $N = 2$ (number of circular divisions within each strand), $R_{\text{Points}} = 1$ (number of points in radial direction within each elemental), and $\text{THET}_{\text{Points}} = 4$ (number of points in angular direction within each elemental).

![Figure 14. M_N_R_\theta = 4_2_1_5 for Each Strand.](image)

Total number of elements within each strand is called $\text{strnd}_\text{els}$ and is equal to $M(N-1)+1$. There are a total of $(\text{strnd}_\text{els})^2$ number of GMD's between any two strands.

3.5.1. Positioning Points within Elementals

Within elementals, $\text{THET}_{\text{Points}}$ are positioned such that the angular distances between all adjacent points are equal (Figure 15).
Figure 15. $M_{N\_R\_\theta} = 4\_2\_1\_5$ with Equal Angular Distance between THETPoints.

**Figure 16** shows that RPoints are positioned in each elemental such that the distance between any two adjacent points is the length of the elemental / RPoints. And the distance between the points adjacent to the edges of the elemental is the length of the elemental / ($2 \times$ RPoints).

Appendix B.1. on page 63 describes the function that positions points within elements.
3.5.2. GMD Calculations for Elementals Not Within the Same Strand

Figure 17 shows two elements with THETPoints = 4 and RPoints = 1 for GMD calculations.

In Appendix C on page 72, the analytic formula for GMD calculations for elements within the same strand is derived. Because of the complexity of the analytical formula, a numerical approach is used for positioning points within each elemental.

Figure 17. Two Elements with THETPoints = 4 and RPoints = 1 for GMD Calculations.

In Figure 17, 16 distances are calculated and multiplied together, and the 16th root of the product is taken. The resultant is the GMD between the two elementals. Appendix B.2. on page 63 describes the function that calculates the GMD.

3.5.3. GMD Calculations for Elementals Within the Same Strand

Figure 18 shows a strand with $M_N = 8.4$. For such cases, the function Find_SelfGMDs_Ls_And_Resistances goes through different loops, as described next with reference to Figure 18.
The function Find_SelfGMDs_Ls_And_Resistances goes through one loop to assign the GMD between 0,a−1,a to

\[
\begin{align*}
0,a-2,a &\quad 0,a-3,a &\quad \ldots &\quad 0,a-8,a \\
0,a-10,a &\quad 0,a-11,a &\quad \ldots &\quad 0,a-16,a \\
0,a-18,a &\quad 0,a-19,a &\quad \ldots &\quad 0,a-24,a
\end{align*}
\]

and to assign the GMD between 0,a−9,a to

\[
\begin{align*}
2,a-3,a &\quad 3,a-4,a &\quad \ldots &\quad 7,a-8,a &\quad 8,a-1,a \\
2,a-4,a &\quad 3,a-5,a &\quad 4,a-6,a &\quad \ldots &\quad 8,a-2,a \\
2,a-5,a &\quad 3,a-6,a &\quad 4,a-7,a &\quad \ldots &\quad 8,a-3,a \\
\end{align*}
\]

It then goes through another loop to assign the GMD between 1,a−2,a to
Within the same loop, the program repeats the above procedure for layer two and layer three elements.

If there is more than one layer (i.e. \( N > 2 \)), the program goes through another loop to determine the GMD between elements of layer \( p \) and layer \( p+1 \). For the example of Figure 18, the program assigns the GMD between 1,a–9,a to

\[
2,\text{a–10,a} \quad 3,\text{a–11,a} \quad 4,\text{a–12,a} \quad \ldots \quad 8,\text{a–16,a}
\]

and assigns the GMD between 1,a–10,a to

\[
2,\text{a–11,a} \quad 3,\text{a–12,a} \quad 4,\text{a–13,a} \quad \ldots \quad 8,\text{a–9,a}
\]

and assigns the GMD between 1,a–11,a to

\[
2,\text{a–12,a} \quad 3,\text{a–13,a} \quad 4,\text{a–14,a} \quad \ldots \quad 8,\text{a–10,a}
\]

and assigns the GMD between 1,a–12,a to

\[
2,\text{a–13,a} \quad 3,\text{a–14,a} \quad 4,\text{a–15,a} \quad \ldots \quad 8,\text{a–11,a}
\]

and assigns the GMD between 1,a–13,a to

\[
2,\text{a–14,a} \quad 3,\text{a–15,a} \quad 4,\text{a–16,a} \quad \ldots \quad 8,\text{a–12,a}
\]

At this point, the GMD values between elements of layer one and layer two are found. The same procedures is repeated to find the GMD's for elements of layer one and layer three and to find the GMD's of the elements of layer two and layer three.
3.5.4. GMR Calculations

The geometric Mean Radius (GMR) of an area is defined as the geometric mean distance of all points within that area with respect to each other [23]. Figure 19 shows eight out of 992 distances between points within one elemental.

For \( P \) total number of points within one elemental, there will be \( P(P-1) \) number of distances to be calculated. Therefore, in Figure 19, there will be \( 32 \times 31 = 992 \) distances. For \( N \) number of circles within each strand, the program calculates \( N-1 \) GMR's only. Appendix B.3. on page 64 describes the function that calculates the GMR.

3.6. Using Symmetry in Finding \( R \) and \( L \)

When the program constructs the impedance matrix (HUGEZ), it needs to calculate the mutual inductance between all of the elements and place these values in the off-diagonal locations of the matrix, and resistance and self inductance values to place in diagonal locations of HUGEZ. Appendices B.4. to B.7. on pages 64 to 67 describe functions in the program that find the GMD of the elements within a strand, find inductances and resistances, and construct the GMD, SelfGMD, and HUGEZ matrices.
The resistance and self inductance values need to be calculated only for the elements of a single strand. These values will be repeated in the other diagonal positions of HUGEZ.

Mutual inductances have to be calculated only between the elements of the strand pairs listed in the output of the 37DSPR.EXE.

In addition, there is another symmetry between the elements of the strand pairs used by the program. The rule is to find the x-symmetry of an element every time the program is finding the x-symmetry of a strand, and similarly for y-symmetry and origin symmetry. For example, looking at Figure 12 on page 22 (and remembering that the notation a,b-c,d is used to refer to the GMD between element a of strand b with element c of strand d), the GMD for 2,5-3,8 is the same as the GMD for (x-symmetry) 3,3-2,18, and as the GMD for (y-symmetry) 4,2-1,14, and as the GMD for (x-symmetry again) 1,6-4,12.

3.7. Row-Column Subtraction and Reduction Procedures

In section 1.1. on page 5, it was explained how the row-column subtraction affects the impedance matrix and how the reduction process is carried on. Appendices B.8. and B.9. on page 68 describe in more detail the functions that perform these tasks.
4. Improvements in the Basic Program Design

4.1. Automatic Choice of RPoints and THETPoints

For a 7-strand all aluminum conductor, many runs have been performed with constant M, N, and RPoints. Only the THETPoints variable (or \( \theta \) the number of points in the angular direction within each elemental) is changed. For each constant M (2, 4, 6, ..., 18, 20), a series of runs at frequencies 0.01, 0.05, 0.1, 0.5, 1.0, 5, 10, 50, 60, 100, 500, 1k, 5k, 10k, 50k, 100k, 500k, 1M, 5M, and 10MHz have been performed for different \( \theta \) values. All these runs were done with N = 2 and RPoints = 1. At each frequency, the \( \theta \) variable was increased starting from one. To find the best \( \theta \) at a certain frequency, runs with consecutive \( \theta \) values were compared with each other until the difference was less than 2%. That run is the one with the best \( \theta \). Then, all runs with all different \( \theta \) values were compared with the best run, and if the percent difference between a run and the best run was less than 2%, that \( \theta \) value is good for that frequency.

For example, M_N_R_\( \theta \) = 4_2_1_1 is compared with M_N_R_\( \theta \) = 4_2_1_2 for all of the above 20 frequencies. Then 4_2_1_2 is compared with 4_2_1_3, and 4_2_1_3 with 4_2_1_4, etc. It turns out that the difference in the resistance between 4_2_1_10 and 4_2_1_11 is less than 2% at all frequencies. Therefore, all cases are compared with 4_2_1_11 to find \( \theta \) for different frequencies. The final result for the M = 4 case is shown in Table 5.

<table>
<thead>
<tr>
<th>choose ( \theta )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>for Freq Hz</td>
<td>&lt;=500</td>
<td>&lt;=5k</td>
<td>&lt;=100k</td>
<td>&lt;=500k</td>
<td>&lt;=10M</td>
</tr>
</tbody>
</table>
Only when \( M \) changes, the criteria for choosing \( \theta \) (or THETPoints) also changes, and this is because when \( M \) increases, the number of divisions in the angular direction also increases; whereas, a change in \( N \) does not affect the criteria for choosing \( \theta \).

For all other values of \( M \), the same procedure explained above was performed.

The above rules for choosing \( \theta \) automatically were implemented into the program, and a similar method was used to make the choice of the \( R \) (RPoints) variable automatic. The criteria for choosing \( R \) changes with a change in the \( N \) variable.

For example, the case \( M_N_R = 4_2_1 \) was compared with the case \( M_N_R = 4_2_2 \), and \( 4_2_2 \) with \( 4_2_3 \). Three percent was the maximum tolerance allowed for resistance values. Because the difference between \( 4_2_2 \) and \( 4_2_3 \) was less than 3\%, all cases of \( M_N = 4_2 \) were compared with \( 4_2_3 \). After going through the same procedure for \( M_N = 4_3 \) and \( M_N = 4_4 \), the following conclusions were reached for the choice of the RPoints (R) variable:

Table 6. RPoints for \( N = 2 \) and 3 at Different Frequencies.

<table>
<thead>
<tr>
<th>For ( N = 2 ), choose R</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>for Freq (Hz) &lt;= 100k</td>
<td>&lt;=</td>
<td>&lt;=</td>
</tr>
<tr>
<td>for Freq (Hz) &lt;= 1M</td>
<td>&lt;=</td>
<td>&lt;=</td>
</tr>
</tbody>
</table>

For \( N = 3 \),

<table>
<thead>
<tr>
<th>choose R</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>for Freq (Hz) &lt;= 1M</td>
<td>&lt;=</td>
<td>&lt;=</td>
</tr>
<tr>
<td>for Freq (Hz) &lt;= 10M</td>
<td>&lt;=</td>
<td>&lt;=</td>
</tr>
</tbody>
</table>

For all larger \( N \) values, RPoints is chosen as one.

4.2. Automatic Choice of \( M \) and \( N \)

A program (for a 7-strand all aluminum conductor) was written that chooses the RPoints and THETPoints variables automatically based on the criteria explained in section 4.1. This program requires the user to input both the \( M \) and \( N \) variables. The program
keeps the M variable constant, and it changes the N variable for all of the 20 different frequencies. Allowing a maximum 4% error for the resistance, a set of appropriate N values were derived for different frequencies for each M equal to an even number between 2 and 20. Only three results are shown here, for M = 4, 10, and 14 in, Table 7.

Table 7. The N Variable for M = 4, 10, and 14.

<table>
<thead>
<tr>
<th>M = 4</th>
<th>&lt;=500</th>
<th>&lt;=10k</th>
<th>&lt;=50k</th>
<th>&lt;=10M</th>
</tr>
</thead>
<tbody>
<tr>
<td>choose N</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M = 10</th>
<th>&lt;=500</th>
<th>&lt;=10k</th>
<th>&lt;=1M</th>
<th>&lt;=10M</th>
</tr>
</thead>
<tbody>
<tr>
<td>choose N</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M = 14</th>
<th>&lt;=500</th>
<th>&lt;=10k</th>
<th>&lt;=10M</th>
</tr>
</thead>
<tbody>
<tr>
<td>choose N</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

A new program was then written for a 7-strand all aluminum conductor that uses the above rule to choose the variable N based on M. This program only requires the user to input the M variable.

The percent difference between adjacent results for resistances were obtained. The conclusion was that M = 16 has less than 4% difference with M = 18 at all frequencies. Therefore, M = 18 was taken as the best result and all other cases of M were compared with M = 18 case. It was then determined what values of M should be chosen for the various frequency ranges.

A program was then written that asks the user if he or she would like to enter M and N manually or automatically. In the manual mode, the user enters M and N. In the auto-
matic mode, the M and N variables are chosen automatically by the program, after the user enters the tolerance as shown in Tables 8 to 11.

Table 8. Case A with 4% Tolerance:

<table>
<thead>
<tr>
<th>M_N</th>
<th>Freq (Hz)</th>
<th>Elements</th>
<th>7-strand Bytes</th>
<th>7-strand hr:min:sec</th>
<th>37-strand Bytes</th>
<th>37-strand hr:min:sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>&lt;=500</td>
<td>1</td>
<td>872</td>
<td>00:00:01</td>
<td>19.3k</td>
<td>00:00:03</td>
</tr>
<tr>
<td>2 3</td>
<td>&lt;=1k</td>
<td>5</td>
<td>12.5k</td>
<td>00:00:02</td>
<td>317k</td>
<td>00:01:16</td>
</tr>
<tr>
<td>4 3</td>
<td>&lt;=10k</td>
<td>9</td>
<td>38.5k</td>
<td>00:00:04</td>
<td>1.01M</td>
<td>00:05:32</td>
</tr>
<tr>
<td>6 5</td>
<td>&lt;=100k</td>
<td>25</td>
<td>288k</td>
<td>00:00:36</td>
<td>7.69M</td>
<td>01:27:00</td>
</tr>
<tr>
<td>14 4</td>
<td>&lt;=500k</td>
<td>43</td>
<td>844k</td>
<td>00:02:46</td>
<td>22.7M</td>
<td>Impossible</td>
</tr>
<tr>
<td>16 4</td>
<td>&lt;=10M</td>
<td>49</td>
<td>1.1M</td>
<td>00:04:00</td>
<td>29.5M</td>
<td>Impossible</td>
</tr>
</tbody>
</table>

Note that the sign "\(<=\)" means less than or equal to, "Elements" means total number of elements within one strand, "Bytes" stands for minimum required extended memory, and "hr:min:sec" stands for run time in hours:minutes:seconds on a 80486–33MHz IBM compatible computer. Because up to a maximum of 16 MBytes of extended memory can be used by the program, when the required memory is more than 16 MBytes, it is "impossible" to run that case.

Table 9. Case B with 10% Tolerance:

<table>
<thead>
<tr>
<th>M_N</th>
<th>Freq (Hz)</th>
<th>Elements</th>
<th>7-strand Bytes</th>
<th>7-strand hr:min:sec</th>
<th>37-strand Bytes</th>
<th>37-strand hr:min:sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>&lt;=1k</td>
<td>1</td>
<td>872</td>
<td>00:00:01</td>
<td>19.3k</td>
<td>00:00:03</td>
</tr>
<tr>
<td>4 3</td>
<td>&lt;=10k</td>
<td>9</td>
<td>39k</td>
<td>00:00:03</td>
<td>1.01M</td>
<td>00:05:31</td>
</tr>
<tr>
<td>4 4</td>
<td>&lt;=50k</td>
<td>13</td>
<td>79k</td>
<td>00:00:08</td>
<td>2.09M</td>
<td>00:14:33</td>
</tr>
<tr>
<td>4 5</td>
<td>&lt;=100k</td>
<td>17</td>
<td>134k</td>
<td>00:00:16</td>
<td>3.56M</td>
<td>00:30:05</td>
</tr>
<tr>
<td>10 5</td>
<td>&lt;=1M</td>
<td>41</td>
<td>768k</td>
<td>00:02:34</td>
<td>20.7M</td>
<td>Impossible</td>
</tr>
<tr>
<td>14 4</td>
<td>&lt;=10M</td>
<td>43</td>
<td>844k</td>
<td>00:03:09</td>
<td>22.7M</td>
<td>Impossible</td>
</tr>
</tbody>
</table>
Table 10. Case C with 13% Tolerance.

<table>
<thead>
<tr>
<th>M_N</th>
<th>Freq (Hz)</th>
<th>Elements</th>
<th>7-strand Bytes</th>
<th>7-strand hr:min:sec</th>
<th>37-strand Bytes</th>
<th>37-strand hr:min:sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>&lt;=1k</td>
<td>1</td>
<td>872</td>
<td>00:00:01</td>
<td>19.3k</td>
<td>00:00:03</td>
</tr>
<tr>
<td>2 3</td>
<td>&lt;=5k</td>
<td>5</td>
<td>12.5k</td>
<td>00:00:02</td>
<td>317k</td>
<td>00:01:16</td>
</tr>
<tr>
<td>4 3</td>
<td>&lt;=10k</td>
<td>9</td>
<td>39k</td>
<td>00:00:04</td>
<td>1.01M</td>
<td>00:05:31</td>
</tr>
<tr>
<td>4 4</td>
<td>&lt;=50k</td>
<td>13</td>
<td>79k</td>
<td>00:00:08</td>
<td>2.09M</td>
<td>00:14:33</td>
</tr>
<tr>
<td>4 5</td>
<td>&lt;=100k</td>
<td>17</td>
<td>134k</td>
<td>00:00:16</td>
<td>3.56M</td>
<td>00:30:05</td>
</tr>
<tr>
<td>6 6</td>
<td>&lt;=500k</td>
<td>31</td>
<td>441k</td>
<td>00:01:54</td>
<td>11.9M</td>
<td>02:53:22</td>
</tr>
<tr>
<td>8 6</td>
<td>&lt;=1M</td>
<td>41</td>
<td>768k</td>
<td>00:03:03</td>
<td>20.7M</td>
<td>Impossible</td>
</tr>
<tr>
<td>14 4</td>
<td>&lt;=10M</td>
<td>43</td>
<td>844k</td>
<td>00:03:09</td>
<td>22.7M</td>
<td>Impossible</td>
</tr>
</tbody>
</table>

Table 11. Case D with 21% Tolerance:

<table>
<thead>
<tr>
<th>M_N</th>
<th>Freq (Hz)</th>
<th>Elements</th>
<th>7-strand Bytes</th>
<th>7-strand hr:min:sec</th>
<th>37-strand Bytes</th>
<th>37-strand hr:min:sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>&lt;=1k</td>
<td>1</td>
<td>872</td>
<td>00:00:01</td>
<td>19.3k</td>
<td>00:00:03</td>
</tr>
<tr>
<td>2 3</td>
<td>&lt;=10k</td>
<td>5</td>
<td>12.5k</td>
<td>00:00:02</td>
<td>317k</td>
<td>00:01:23</td>
</tr>
<tr>
<td>4 4</td>
<td>&lt;=50k</td>
<td>13</td>
<td>79k</td>
<td>00:00:08</td>
<td>2.09M</td>
<td>00:14:33</td>
</tr>
<tr>
<td>4 5</td>
<td>&lt;=100k</td>
<td>17</td>
<td>134k</td>
<td>00:00:16</td>
<td>3.56M</td>
<td>00:30:05</td>
</tr>
<tr>
<td>6 6</td>
<td>&lt;=10M</td>
<td>31</td>
<td>441k</td>
<td>00:03:06</td>
<td>11.9M</td>
<td>03:14:51</td>
</tr>
</tbody>
</table>

The programs for 1, 7, and 19-strand conductors (1hzr13.exe, 7hzr13.exe, and 19hzr13.exe) give all of the 4 options to the user in the automatic mode for all frequencies. On the other hand, the program for a 37-strand conductor (37hzr13.exe) gives 4 options to the user only for frequencies less than or equal to 100 kHz. For frequencies larger than 100 kHz the program runs with M_N = 6_6 with no options for the user.
5. Results

The results of runs with the 37hzr13.exe program in the automatic mode for all frequencies are presented. Options A to D of the automatic mode are used. Figures 20 and 21 show the results of runs with the program 37hzr13.exe for a 37 all aluminum stranded conductor for the resistance and the inductance, respectively. The strand's parameters are: strand radius = 1.72974 (mm), aluminum conductivity = 3.4662e7 (S/m), aluminum relative permeability = 1.0, and height of the transmission line = 20 (m).

37-Strand Conductor (Subdivision)
Resistance A–D_R

![Graph showing resistance vs. frequency for different options A to D.]

Figure 20. Options A to D in the Automatic Mode of 37hzr13.exe for the Resistance.

Note that the Subdivision program outputs the internal inductance of the stranded conductor. This is the final result from the Subconductors matrix reduction minus the external inductance \( I_{\text{ext}} = \frac{\mu}{2 \pi} \ln \left( \frac{2h}{r_{\text{total}}} \right) \), where \( r_{\text{total}} \) is the external radius of the conductor.

In the graphs presented; however, the inductance is the total inductance of the conductor.
with an ideal ground return. This has been done to compare directly with the output from the Line Constants routine of the EMTP (Mtline version).

Figure 21. Options A to D in the Automatic Mode of 37hrz13.exe for the Inductance.

Three different runs were done with the Mtline program which is part of the Microtran package for personal computers (UBC-EMTP version). One run was performed using Galloway's formula (e1) for the 37 all aluminum conductor described above, one using the "TUBE" formula for the entire 37-strand conductor (e6), and one run using the TUBE formula for each of the 37 strands in the conductor. Figures 22 and 23 compare the results of case A (4% tolerance) in the automatic mode of the Subdivision method with those of Mtline for the resistance and the inductance respectively.
Figure 22. Mtline and Subdivisions (Case A, 4% Tolerance) Results for the Resistance.
Figure 23 shows that Galloway's formula (e1) does not give correct answers for the inductance for frequencies below 500 Hz. Therefore, in Figures 24 and 25, Galloway's results are deleted, and only the Subdivision, TUBE and 7-TUBE (and 37-TUBE) cases for inductance of a 7-strand (and 37-strand) conductors are shown.

In Figures 21 and 25, one observes that the inductance curve does not decrease monotonically. This is due to the fact that the number of subdivisions required for frequencies above 100 kHz should be greater than $M_N = 66$ and this was not possible because the program can handle a maximum of 16 MBytes. Figure 24 shows that for the 7-strand case, where there is enough memory to run the appropriate case, there is a continuous decrease in the value of the inductance.
Figure 24. Inductance by Subdivision-Case-A, TUBE, and 7-TUBE for a 7-Strand Conductor.

Figure 25. Inductance by Subdivision-Case-A, TUBE, and 37-TUBE for a 37-Strand Conductor.
Figures 26 and 27 show the percent difference (difference divided by the average times 100) between the Subdivision results Case A, and Galloway's results from Mtline for the resistance and inductance values respectively. Figures 28 and 29 show the percent difference between the Subdivision results and the TUBE's results, and Figures 30 and 31 show the percent difference between the Subdivision and the 37-TUBE case of the Mtline.

![37-Strand % Diff. Subdiv. & Galloway](image)

Figure 26. Percent Difference between Subdivision and Galloway for the Resistance.
**37-Strand % Diff. Subdiv. & Galloway**

% Difference in Inductance PAe1_L

![Graph showing % Difference in Inductance vs Frequency](image)

Figure 27. Percent Difference between Subdivision and Galloway for the Inductance.

**37-Strand % Diff. Subdiv. & TUBE**

% Difference in Resistance PAe6_R

![Graph showing % Difference in Resistance vs Frequency](image)

Figure 28. Percent Difference between Subdivision and TUBE for the Resistance.
37–Strand % Diff. Subdiv. & TUBE

% Difference in Inductance PAe6_L

Figure 29. Percent Difference between Subdivision and TUBE for the Inductance.

37–Strand % Diff. Subdiv. & 37 TUBE

% Difference in Resistance PAe14_R

Figure 30. Percent Difference between Subdivision and 37-TUBE for the Resistance.
Figure 31. Percent Difference between Subdivision and 37-TUBE for the Inductance.
6. Exponential Formula for the Resistance Based on the Subdivision

**Results**

Figure 32 shows that the Subdivision and TUBE results for the resistance are very close to each other for frequencies up to 5 kHz.

![Graph showing percent difference between Subdivision and TUBE results for 1, 7, 19, and 37-strand conductors for resistance.](image)

Figure 32. Percent Difference between Subdivision and TUBE results of 1, 7, 19, and 37-Strand Conductors for the Resistance.

A best-fit straight-line (in a least mean squared sense [24]) can be found that passes through the 5 kHz point in the log(R) versus log(freq) plot of the Subdivision results in Figure 22 on page 41. One could then use the TUBE formula to calculate the value of the resistance at 5 kHz and use the above calculated slope to estimate the resistance at higher frequencies.

To obtain the log-log slope, the Subdivision results for 1, 7, 19, and 37-strand conductors are normalized by dividing the final resistance values by the stranding factor and
multiplying the result by the number of strands in the conductor. The stranding factor (Table 3, page 12) depends on the total number of strands in the conductor. In the plots presented in this thesis, the values of the resistance obtained by the Subdivision method are divided by the stranding factor, since this factor is not included in the TUBE and other approximations.

Figure 33 shows the normalized 1, 7, 19, and 37-strand results for the resistance by the Subdivision method.

1, 7, 19, 37 Strand Subdiv.
Normalized for Resistance nAall_R

---

Figure 33. Normalized 1, 7, 19, and 37-Strand Subdivision Results for the Resistance.

Slopes of the four straight lines of Figure 33 are shown in Table 12.
Table 12. High-Frequency Asymptotic Slopes for Different Numbers of Strands

<table>
<thead>
<tr>
<th>Number of Strands</th>
<th>Slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.418</td>
</tr>
<tr>
<td>7</td>
<td>0.748</td>
</tr>
<tr>
<td>19</td>
<td>1.09</td>
</tr>
<tr>
<td>37</td>
<td>0.922</td>
</tr>
</tbody>
</table>

The slope of the 37-strand conductor does not follow the increasing pattern in the slope as the previous cases do. This is probably due to the fact that there is a computer memory limitation of 16 MBytes for a 37-strand conductor, as explained in the previous sections.

Figure 33 shows that the 7-strand resistance result has a slope 0.33 more than the 1-strand case, and the 19-strand case has a slope 0.34 more than the 7-strand slope. However, because of the memory limitation in the 37-strand case above 100 kHz, one cannot conclude that the true slope for the 37-strand case would also be about 0.33 more than the slope of the 19-strand case.

With the resistance at 5 kHz, calculated with TUBE, the resistance at higher frequencies can then be approximated by this exponential formula:

\[ R = k f^S \]  \hspace{1cm} (1)

where \( f \) is the frequency in Hz, \( S \) is the slope in Table 12, and \( k \) is given by

\[ k = \frac{R(\text{at 5kHz by TUBE})}{5000^S} \]

Figure 34 shows the resistance obtained by the exponential formula for the 19-strand conductor and the Subdivision results. The starting point of the exponential formula is the resistance obtained by the TUBE method at 5 kHz.

Figure 35 shows the percentage difference between the Subdivision and the Exponential formula results for 1, 7, 19, and 37-strand conductors.
19-Strand Subdiv. & Exponential
For Resistance A19x_R

Figure 34. Subdivision and Exponential Formula Results of a 19-Strand Conductor for the Resistance.

%Diff. 1, 7, 19, & 37-Strand Subdiv. & Exponential for Resistance PAallx_R

Figure 35. Percentage Difference between the Subdivision and Exponential Formula Results of 1, 7, 19, and 37-Strand Conductors for the Resistance.
7. Conclusions

For the resistance, Galloway's formula has a maximum difference of 27% with respect to the Subdivision results for frequencies in the range [100Hz, 10kHz]. TUBE's formula gives satisfactory results in the range [DC, 5kHz] up to a maximum of 9% difference. The 37-TUBE formula gives good results in the frequency range of [DC, 1kHz] up to a maximum 2% difference with the Subdivision results. Outside these frequency ranges, none of the methods (except the subdivision method) is appropriate for resistance calculations, as shown in Table 13.

<table>
<thead>
<tr>
<th></th>
<th>1kHz</th>
<th>5kHz</th>
<th>10kHz</th>
<th>100kHz</th>
<th>1MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Galloway</td>
<td>-28%</td>
<td>-24%</td>
<td>8%</td>
<td>102%</td>
<td>140%</td>
</tr>
<tr>
<td>TUBE</td>
<td>-10%</td>
<td>1%</td>
<td>35%</td>
<td>122%</td>
<td>154%</td>
</tr>
<tr>
<td>37-TUBE</td>
<td>1%</td>
<td>43%</td>
<td>84%</td>
<td>150%</td>
<td>171%</td>
</tr>
</tbody>
</table>

Table 13. Galloway, TUBE, and 37-TUBE Difference with the Subdivision Results for the Resistance.

For inductance, Galloway's formula has a maximum difference of 4% with respect to the Subdivision results for frequencies in the range [100Hz, 10MHz]. TUBE's formula gives satisfactory results at all frequencies with a maximum difference of 1.3%. The 37-TUBE formula also gives good results at all frequencies up to a maximum of 0.42% difference with the Subdivision results. Therefore, using one of these methods for inductance calculations alone would be more efficient in terms of CPU time than the subdivision method.

An exponential formula is also presented that satisfactorily approximates the resistance of stranded conductors at frequencies from 5 kHz to 10 MHz. This formula is developed for 1, 7, and 19-strand conductors of equal strand diameters. The maximum error of the formula at all frequencies in the above range was 26.5%. For frequencies less than or equal to 5 kHz, the TUBE formula provides accurate answers.
To use the exponential formula, the TUBE formula is used to calculate the resistance of a 1, 7, or 19-strand conductor at 5 kHz. Formula (1) on page 50 with the slopes indicated in Table 12 on page 50 can then be used to calculate the resistance beyond this frequency point.
References


[4] Dr. H. W. Dommel, personal communications, Department of Electrical Engineering, The University of British Columbia, 2356 Main Mall, Vancouver, B.C., V6T 1Z4, Canada.


54


[20] Dr. J.R. Marti, personal communications, Department of Electrical Engineering, The University of British Columbia, 2356 Main Mall, Vancouver, B.C., V6T 1Z4, Canada.


Appendices

A. Package of Programs to Implement the Subconductors Technique for Stranded Conductors

For the case of a 37-strand conductor, five *.exe were written. The programs are run in the following order:
1. The 37DIST.EXE program finds all strand pairs that have the same distance and angle from each other and stores the final results in an array of structures on disk as the 37dist.dat binary file.
2. The 37XY.EXE program stores the symmetric strand pairs with respect to the x and y axes on disk in a binary file called 37xy.dat. The text output of this program is included in Appendix A.2., page 59.
3. The 37PAIR.EXE program uses the 37xy.dat and finds all strand pairs with x or y or x−y (origin) symmetry and stores the final results as an array of strings on disk as the 37pair.dat binary file.
4. The 37DSPR.EXE program uses the 37dist.dat and 37pair.dat and combines these two files in such a way that only 161 instead of 666 GMD's have to be calculated. It stores the final result as an array of structures on disk in the binary file 37dspr.dat.
5. The 37HZR13.EXE program uses the 37dspr.dat and 37xy.dat and creates a huge [Z] matrix. It then reduces this matrix to calculate the final R and L values of the conductor and stores the results in the text file 37hrz13.out with all the important information relating to that specific run.

A.1. Finding Strand Pairs with Equal Distances and Angles (37DIST.EXE)

The 37DIST.EXE program is compiled for a 37-strand conductor that looks the same as in Figure 11 on page 21 but without the fourth layer of strands.

In this program, an array of structures is defined such that each of the 37 elements of the array (A) contains a double precision number as the strand's magnitude and a double precision number as the strand's angle. The magnitude (in terms of the radius) and the angle (in radians) of each strand is assigned to each element of this array. For example, A[14].mag = 4, A[14].ang = 7π/6; and A[35].mag = 6, A[35].ang = 16π/9.

An array of structures B is defined with 666 elements (total number of possible pairs in a 37-strand conductor = 36 + 35 + 34 + ...+2 + 1 = 666). Each element of this array contains integer as the first strand number, an integer as the second strand number, a double (precision number) as the distance between the first and the second strands, and a double precision number as the angle between the two strands. In a loop, all 666 pairs of distances and angles are calculated and assigned to each element of array B. Angles smaller than 0.0001 radians (0.0057 degrees) are considered as zero, and all angles are in the [0, π) region. For example, the strand pair 5–31 and the pair 2–22 both have distance 4r and angle π/3 (strand pairs are referred to as "smaller strand number – larger strand number").

An array of structures C is defined with an unknown number of elements (maximum 666) such that each element has a double precision number as the distance and a double precision number as the angle between a pair of strands, an integer for the total number of pairs with equal distance and angle, and a string array that keeps the names of the pairs with equal distance and angles. The program goes through a loop starting at strand pair a-b = 0–1 then 0–2, 0–3, 0–4, ..., 0–36; 1–2, 1–3, ..., 1–36; 2–3, ....; 35–36. In each iteration, the program checks if the string a-b is already stored in the string array of C. For example, for a–b = 0–1, there is no data in the string array of C[0], so 0–1 is put as the first string of C[0]. Then the program compares the distance and angle of 0–1 with all other strand pairs, if it finds other pairs with equal distance and angle, it then puts them into C[0].equal_dist_array[i].
A new group is defined for each strand pair that has a different distance and angle from the previous groups. For instance, group 1 contains the following strand pairs: 0–1, 0–4, 1–7, 2–3, 4–13, 5–6, 7–19, 13–28 that have a distance 2r and an angle of 0.0 radians from each other (see Figure 11 on page 21). Some groups have one pair, such as group 269 with distance = 12r, angle = 80 degrees, and pair = 23–32. And some groups have eight pairs, such as group 2 with distance 2r, angle 60 degrees, and pairs: 0–2, 0–5, 1–6, 2–9, 3–4, 5–15, 9–22, 15–31 (see Appendix A.1.1. on page 58).

After running the 37DIST.EXE program, the user realizes that there are a total of 288 groups, so that the size of the array of structures C (that was unknown at the beginning) should be chosen as 288.

The contents of the structure array C are stored on the hard disk as the binary file "37dist.dat." This file will be used later by the program 37DSPR.EXE.

### A.1.1. 37DIST Sample Output

The contents of the structure array C are:

- **Group: 1 dist:** 2.00r  angle: 0.00 deg,  pairs: 0–1  0–4  1–7  2–3  4–13
  - 5–6  7–19  13–28 total pairs = 8
- **Group: 2 dist:** 2.00r  angle: 60.00 deg,  pairs: 0–2  0–5  1–6  2–9
  - 3–4
- **Group: 3 dist:** 2.00r  angle: 120.00 deg,  pairs: 0–3  0–6  1–2  3–11  4–5
  - 6–17 11–25 17–34 total pairs = 24
- **Group: 4 dist:** 4.00r  angle: 0.00 deg,  pairs: 0–7  0–13  1–4  1–19  4–28  9–11  15–17 total pairs = 31
- **Group: 5 dist:** 4.00r  angle: 30.00 deg,  pairs: 0–8  0–14  10–12  16–18 total pairs = 35
- **Group: 6 dist:** 4.00r  angle: 60.00 deg,  pairs: 0–9  0–15  2–5  2–22  5–31
  - 7–17 11–13 total pairs = 42
- **Group: 7 dist:** 4.00r  angle: 90.00 deg,  pairs: 0–10  0–16  8–18  12–14 total pairs = 46
- **Group: 8 dist:** 4.00r  angle: 120.00 deg,  pairs: 0–11  0–17  3–6  3–25  6–34
  - 7–9 13–15 total pairs = 53

...  
- **Group: 46 dist:** 4.18r  angle: 150.57 deg,  pairs: 1–36  4–27 total pairs = 184
- **Group: 47 dist:** 2.48r  angle: 6.21 deg,  pairs: 2–8  5–14 total pairs = 186
- **Group: 48 dist:** 2.48r  angle: 113.79 deg,  pairs: 2–10  5–16 total pairs = 188
- **Group: 49 dist:** 4.47r  angle: 176.57 deg,  pairs: 2–12  5–18 total pairs = 190
- **Group: 50 dist:** 5.29r  angle: 19.11 deg,  pairs: 2–13  3–28  5–7  6–19 total pairs = 196

...  
- **Group: 267 dist:** 4.10r  angle: 10.00 deg,  pairs: 23–25  32–34 total pairs = 629
- **Group: 268 dist:** 7.71r  angle: 30.00 deg,  pairs: 23–27  32–36 total pairs = 631
- **Group: 269 dist:** 12.00r angle: 80.00 deg,  pairs: 23–32 total pairs = 632
- **Group: 270 dist:** 11.82r angle: 90.00 deg,  pairs: 23–33  24–32 total pairs = 634
- **Group: 271 dist:** 11.28r angle: 100.00 deg,  pairs: 23–34  25–32 total pairs = 636

...  
- **Group: 286 dist:** 12.00r angle: 140.00 deg,  pairs: 26–35 total pairs = 663
- **Group: 287 dist:** 11.82r angle: 150.00 deg,  pairs: 26–36  27–35 total pairs = 665
- **Group: 288 dist:** 12.00r angle: 160.00 deg,  pairs: 27–36 total pairs = 666

Good Bye 37dist!
A.2. 37XY Output

Here is part of the screen output of 37XY.EXE (see Figure 11 on page 21).

This is 37XY program that stores the strand numbers symmetric with respect to x and y axes on hard disk and shows them on the screen.

<table>
<thead>
<tr>
<th>Strand</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-symmetric</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>18</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>Y-symmetric</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strand</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-symmetric</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>19</td>
<td>36</td>
</tr>
<tr>
<td>Y-symmetric</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>18</td>
<td>17</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>28</td>
<td>27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strand</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-symmetric</td>
<td>35</td>
<td>34</td>
<td>33</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td>26</td>
</tr>
<tr>
<td>Y-symmetric</td>
<td>26</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td>36</td>
<td>35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strand</th>
<th>31</th>
<th>32</th>
<th>33</th>
<th>34</th>
<th>35</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-symmetric</td>
<td>34</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>21</td>
</tr>
<tr>
<td>Y-symmetric</td>
<td>33</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td></td>
</tr>
</tbody>
</table>

A.3. Finding Symmetric Strand Pairs (37PAIR.EXE)

This program uses the 37xy.dat binary file created by 37xy.exe. It then outputs all x, y, and origin symmetric strand pairs on the screen. A structure array D is defined to store symmetric pairs of strands with an unknown dimension at the beginning. After running 37PAIR.EXE, the dimension of D is found to be 185. The first pair of each of the 185 symmetric groups of D is now stored in the structure array D1, and D1 is saved as the binary file 37pair.dat on disk.

Program 37pair.exe finds the pairs of strands in a 37-strand conductor that are symmetric with respect to the x, y, and xy axes. First, the pair i—j, where 0 <= i <= 35 and i+1 <= j <= 36 is found. If this pair is not in any of the previously found pairs of D, a new group (dgroup) in D is created and this pair is put as the first element of that group. Then, the program finds the x-symmetric pair of i—j, say a—b, and searches to see if this pair already exists. If it is already found, the program does not do anything; and if it is not already found, the program puts this a—b pair in the same group as i—j. It then finds the y-symmetric pair of a—b, say c—d, and it checks if c—d is already found, if not, it puts c—d in the same group as i—j. Finally, the program gets the x-symmetric pair of c—d, say e—f, and if it is not already found, it puts e—f in the i—j group. The program now goes to the next i-j and repeats the above procedure for all i and all j. At the end, it outputs all distinct groups and their elements. Obviously, in each group there can be a maximum of 4 pairs (like 2—11, 6—15, 5—17, 3—9) and a minimum of 1 pair like 1—4 (see Figure 11 on page 21). Finally, the first strand pairs from each group of D are put into the array D1.

The importance of this program is that considering any pair of strands (among 666 possibility in a 37-strand conductor), either that pair itself, or its x-symmetric pair, or its y-symmetric pair, or its x—y-symmetric pair will be exactly one of the pairs in the structure array D1.
A.3.1. 37PAIR Sample Output

Here is a sample screen output from the program 37PAIR.EXE.

This is the 37PAIR program that finds the pairs of strands symmetric with respect to x and y axes and puts them into groups that have no common pairs. At the end all groups with their members will be shown. The miracle of this program is that any pair of strands you consider (among 666 possibility in a 30/7 ACSR), either that pair itself or its x-symmetric pair of its y-symmetric pair or its x–y-symmetric pair will be exactly the FIRST pair of one of the groups in structure array D. Then we put all of the first pairs of D into another structure array D1 and save D1 on disk to be used by 37DSPR program.

Dgroup # 1 has symmetric pairs: 0–1 0–4 total pairs = 2
Dgroup # 2 has symmetric pairs: 0–2 0–6 0–5 0–3 total pairs = 6
Dgroup # 3 has symmetric pairs: 0–7 0–13 total pairs = 8
Dgroup # 4 has symmetric pairs: 0–8 0–18 0–14 0–12 total pairs = 12
...
Dgroup # 12 has symmetric pairs: 1–2 1–6 4–5 3–4 total pairs = 40
Dgroup # 13 has symmetric pairs: 1–3 1–5 4–6 2–4 total pairs = 44
Dgroup # 14 has symmetric pairs: 1–4 total pairs = 45
Dgroup # 15 has symmetric pairs: 1–7 4–13 total pairs = 47
...
Dgroup # 36 has symmetric pairs: 2–8 6–18 5–14 3–12 total pairs = 119
Dgroup # 37 has symmetric pairs: 2–9 6–17 5–15 3–11 total pairs = 123
Dgroup # 38 has symmetric pairs: 2–10 6–16 5–16 3–10 total pairs = 127
Dgroup # 39 has symmetric pairs: 2–11 6–15 5–17 3–9 total pairs = 131
Dgroup # 40 has symmetric pairs: 2–12 6–14 5–18 3–8 total pairs = 135
Dgroup # 41 has symmetric pairs: 2–13 6–13 5–7 3–7 total pairs = 139
...
Dgroup # 49 has symmetric pairs: 2–21 6–35 5–30 3–26 total pairs = 171
Dgroup # 50 has symmetric pairs: 2–22 6–34 5–31 3–25 total pairs = 175
Dgroup # 51 has symmetric pairs: 2–23 6–33 5–32 3–24 total pairs = 179
Dgroup # 52 has symmetric pairs: 2–24 6–32 5–33 3–23 total pairs = 183
...
Dgroup # 105 has symmetric pairs: 8–34 18–22 14–25 12–31 total pairs = 382
Dgroup # 106 has symmetric pairs: 8–35 18–21 14–26 12–30 total pairs = 386
Dgroup # 107 has symmetric pairs: 8–36 18–20 14–27 12–29 total pairs = 390
Dgroup # 108 has symmetric pairs: 9–10 16–17 15–16 10–11 total pairs = 394
...
Dgroup # 182 has symmetric pairs: 22–34 25–31 total pairs = 660
Dgroup # 183 has symmetric pairs: 23–24 32–33 total pairs = 662
Dgroup # 184 has symmetric pairs: 23–32 24–33 total pairs = 664
Dgroup # 185 has symmetric pairs: 23–33 24–32 total pairs = 666

These pairs are put into D1 array: 0–1 0–2 0–7 0–8 0–9 0–10 0–19 0–20 0–21 0–22 0–23 1–2 1–3 1–4 1–7 1–8

60
A.4. Combining Equal-Distance-Angle Pairs with Symmetric Pairs

(37DSPR.EXE)

The program 37DSPR.EXE reads the structure array C (C contains all strand pairs with equal distance and angle from each other) from the output of 37dist.exe (file 37dist.dat), and it reads the structure array D1 (D1 contains the first pair of each group in structure array D) from the output of 37pair.exe (file 37pair.dat) from the hard disk. Then for each pair of each group in C, say a—b in group g, it looks through all pairs of D1. If it finds a—b in D1, it puts a—b in a new e—group, say e1, in structure array E, and it records the distance between a and b in group e1.

Then the program looks at the second pair in group g of C, say c—d, and it searches to find if c—d is in D1 or not. If yes, it puts c—d beside a—b in group e1 in E, if not, it ignores c—d and goes to the next pair of strands in g. When all pairs of group g are done, it goes to group g+1 in C and for any pair in group g+1 that is in D1, it creates a new e—group in E that is group e1+1.

If any pair of a group in C is not found in D1, the program does not do anything and goes to the next pair in C like group 15 in C (see the output of 37DSPR program in Appendix A.4.1. on page 61). The pairs in each group of E are not symmetric with each other, but they have the same distance and angle from each other.

For example, if 6 out of 8 pairs of group g1 in C are in D1, then this program puts all of these 6 pairs into group say e3 of E. Next, if none of pairs of g2 in C is in D1, it does not do anything. Next, if 3 out of 4 pairs of g3 are in D1, it puts these 3 pairs in group e4 of E.

The important result from this program is that any pair of strands such as a—b will be either exactly the same as one of the pairs in E (say c—d) or a—b will be the same as an x-symmetric pair of c—d, or the y-symmetric pair of c—d, or xy-symmetric pair of c—d. Therefore, not all GMD’s have to be calculated. Only the GMD values for the first pair of each group in E have to be calculated. The other pairs in the same group will have the same GMD values because these pairs have equal distance and angle from each other (they are derived from the array C that contains the strand pairs with equal distance and angle).

A.4.1. 37DSPR Sample Output

Here is the screen output of 37DSPR.EXE.

<table>
<thead>
<tr>
<th>Group #</th>
<th>pair distance</th>
<th>pairs</th>
<th>total pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2r</td>
<td>0-1</td>
<td>2-3 7-19</td>
</tr>
<tr>
<td>2</td>
<td>2r</td>
<td>0-2</td>
<td>2-9 9-22</td>
</tr>
<tr>
<td>3</td>
<td>2r</td>
<td>1-2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4r</td>
<td>0-7</td>
<td>1-4 1-19 9-11</td>
</tr>
<tr>
<td>5</td>
<td>4r</td>
<td>0-8</td>
<td></td>
</tr>
</tbody>
</table>

...
A.5. Before Constructing the Huge Z Matrix (37HZR13.EXE)

All arrays and structures in the 37HZR13.EXE program are defined dynamically, so that the memory taken by some arrays can be freed when appropriate to be used by other arrays.

The program 37HZR13.EXE reads the binary files 37xy.dat and 37dspr.dat created by previous programs into structure arrays A and E respectively (the same structure arrays as in 37xy.exe and 37dspr.exe). It then asks the user to input the frequency of operation, and if he/she would like the computer to work in the automatic mode or in the manual mode.

In the automatic mode, for frequencies less than or equal to 100 kHz, the user will have four options for percent tolerances, namely 4%, 10%, 13%, and 21%. For frequencies above 100 kHz, the user will have no option, and the computer will run with 21% tolerance. In the automatic mode, all variables including M the number of straight line divisions and N the number of circular divisions within each strand are chosen by the computer. In the manual mode, the user has to enter M and N.

In both the automatic, and the manual modes, the computer chooses the number of points in the angular direction within each elemental (THETPoints), and the number of points in radial direction within each elemental (RPoints) to calculate the GMD between the two elements.

To find the GMD between any two elements that do not belong to the same strand, (THETPoints * RPoints)² number of distances are found and multiplied together, and the (THETPoints * RPoints)² th root of the product is taken (in function Find_One_GMD). The program asks the user to enter the radius of one strand (all strands have equal radii) and the average height of the transmission line from ground. The program gives the option to the user to change the values for Aluminum conductivity σₐl, Steel conductivity σₙst (for ACSR conductors), and Aluminum relative permeability μᵣₜₐₙₜ.

The program 37HZR13.EXE allocates memory for Aluminum and Steel Resistance matrices. Each of these matrices contains N double precision numbers (for any M). For example, if M_N = 8_3, as in Figure 5 on page 5, the program stores the resistance of the central circle, with radius R[0], the resistance of one of the elements between the central circle and circle R[1], and the resistance of one of the elements between circle R[1] and the strand body (circle R[2]).
B. Technical Details of 37HZR13

B.1. Locate_Points Function

The Locate_Points function takes the element number, "el", the strand number, "st," RPoints, THETPoints, and the address of points1 or points2 complex arrays. If the element number "el" turns out to be zero, that means the element is a circle, and this function puts all of the RPoints and THETPoints at the centre of the circle. For elemental shape elements, it positions RPoints in the radial direction and THETPoints in the angular direction within the specific element "el, st", as shown in Figure 15 on page 28 and Figure 16 on page 28.

B.2. Find_One_GMD Function

This function receives two complex arrays and RP an integer for the number of points in the radial direction (equivalent to RPoints) and TP an integer for the number of points in the angular direction (equivalent to THETPoints). It checks the locations of the points in the two complex arrays to identify whether the element is a circle or an elemental.

If none of the two regions is a circle, it calculates n distances, where n = RP * TP, and returns the \((n^2)^{\frac{1}{n}}\) th root of the product of \(n^2\) distances.

If only one of the regions is a circle, it puts one point at the centre of that region and uses \(n = RP * TP\) points in the elemental shape region and returns the nth root of the product of the n distances as the GMD. By writing a separate computer program to find the GMD between a circle and an elemental in many different ways, it was discovered that one point at the centre of the circle gives exactly the same result as all the other methods.

If both regions are circles, this function returns the distance between the centres of the two circles.

To find the GMD within one strand, the program positions SelfR_Times times RPoints and SelfT_Times times THETPoints number of points in the radial and angular directions within each elemental, and then it finds the GMD. To find a correct value for the SelfR_Times = SelfT_Times constants, the final resistance and inductance values of SelfR_Times = SelfT_Times = 2 were compared with SelfR_Times = SelfT_Times = 3 (where M_N_R_0 = 4_2_1_4, and GMRR_Times = GMRT_Times = 4 while a 7 strand conductor was used). The difference was found to be less than 0.12% for resistance and less than 0.11% for inductance at all frequencies. Therefore, SelfR_Times = SelfT_Times = 2 is chosen for all runs.

The exact locations of the points within each element are stored inside each of the complex arrays: points1 and points2. These two complex arrays, as well as the values of RPoints and THETPoints, are given to a function called Find_One_GMD. Inside this function, TOTPoints = RPoints * THETPoints number of distances are calculated, multiplied together, and the TOTPoints root of the product is taken. This function returns the resultant as a double precision number. This double precision number is stored in the GMD matrix as the GMD between two elements, but it is immediately replaced by the mutual inductance between the two elements.

The mutual inductance is \(L_{\text{mutual}} = \frac{\mu}{2\pi} \ln \left( \frac{2h}{GMD} \right)\). The above procedure is repeated for all of the strand elements of each of the first strand pairs in the structure array E.
B.3. Find_GMR Function

To find the GMR of one elemental, the program positions GMRR_Times times RPoints and GMRT_Times times THETPoints number of points in the radial and angular directions, respectively, within one elemental, and then it finds the GMR. To find a correct value for the GMRR_Times = GMRT_Times constants (where M_N_R_0 = 4 2 1 4, and SelfR_Times = SelfT_Times = 2, while a 7 strand conductor was used), the final resistance and inductance values of GMRR_Times = GMRT_Times = 2 were compared with 3, and 3 with 4 and 4 with 5. For the case of comparing 4 and 5, the difference was found to be less than 1.2% for the resistance and less than 0.1% for the inductance at all frequencies. Therefore, GMRR_Times = GMRT_Times = 4 was chosen for all runs. In Figure 19 on page 32, RPoints = 1, and THETPoints = 2 are multiplied by 4 for GMR calculations.

B.4. Find_SelfGMD_Ls_And_Resistances Function

This function finds the self and mutual GMD's for elements within the same strand and stores them in the SelfGMD matrix. The self GMD of a circle = GMR = radius * exp(-0.25) = radius * exp(-Circular_GMRconst). Self GMD of an elemental shape is calculated numerically by positioning points within one sample elemental and finding their GMD's.

The data stored in the SelfGMD matrix will be used over and over when constructing the huge Z matrix because for each strand, the SelfGMD values are the same. Because the GMD's calculated all belong to the same strand, it is a good idea to use more number of points within each elemental for GMD calculations. That is why for GMD calculations within a single strand, twice the THETPoints and RPoints are sent to the Locate_Points function.

In this function, the resistance of each of the elements within one strand is defined using the conductivity and the element area.

B.5. The GMD and SelfGMD Matrices

A dynamic array called GMD is introduced that stores E_Dimension * (strnd_els)^2 double precision numbers. For example, for Figure 12 on page 22 where M=4, and N=2, strnd_els = M(N-1)+1 = 5, the GMD array stores 161 * 25 = 4025 double precision numbers.

There is another dynamic array called SelfGMD that stores the GMD between elements within the same strand. Each element has a GMD by itself (called GMR) and a GMD with each of the other elements within the same strand. Therefore, the SelfGMD matrix has to store strnd_el * (strnd_els + 1)/2 double precision numbers. For example, if strnd_els = 5, then SelfGMD has to have space for 5 * 6 / 2 = 15 double precision numbers.

Two dynamic complex arrays are defined, points1 and points2, such that each one stores TOTPoints = THETPoints * RPoints complex numbers. The program goes through the first strand pair of the structure array E for which it has to find all of the GMD’s between different elements of the two strands.
B.5.1. How the GMD Matrix is Constructed

How the GMD matrix is filled in is the key for accessing the right value of mutual inductance at the time the huge Z matrix is being constructed. The first pair of the first group in E is referred to as E[0] (from Appendix A.4.1. on page 61, this is the strand pair 0–1), and the first pair of the second group in E is referred to as E[1] (which is 0–2), etc. In Figure 12 on page 22, where M=4, and N=2, the GMD matrix will have elements from GMD[0] to GMD[4024] with mutual inductances between elements according to Table 14.

Table 14. Contents of the GMD Matrix for M_N = 4_2.

E[0], corresponding to strand pair 0–1, contributes to:

<table>
<thead>
<tr>
<th>GMD</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_{mutual}</td>
<td>0,0–0,1</td>
<td>1,0–0,1</td>
<td>2,0–0,1</td>
<td>3,0–0,1</td>
<td>4,0–0,1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GMD</th>
<th>5</th>
<th>6</th>
<th>...</th>
<th>10</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_{mutual}</td>
<td>0,0–1,1</td>
<td>1,0–1,1</td>
<td>...</td>
<td>0,0–2,1</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GMD</th>
<th>15</th>
<th>20</th>
<th>...</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_{mutual}</td>
<td>0,0–3,1</td>
<td>...</td>
<td>0,0–4,1</td>
<td>...</td>
</tr>
</tbody>
</table>

E[1], corresponding to strand pair 0–2, contributes to:

<table>
<thead>
<tr>
<th>GMD</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_{mutual}</td>
<td>0,0–0,2</td>
<td>1,0–0,2</td>
<td>2,0–0,2</td>
<td>3,0–0,2</td>
<td>4,0–0,2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GMD</th>
<th>30</th>
<th>35</th>
<th>...</th>
<th>49</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_{mutual}</td>
<td>0,0–1,2</td>
<td>...</td>
<td>0,0–2,2</td>
<td>...</td>
</tr>
</tbody>
</table>

E[160], corresponding to strand pair 23–33, contributes to:

<table>
<thead>
<tr>
<th>GMD</th>
<th>4000</th>
<th>4001</th>
<th>4002</th>
<th>4003</th>
<th>4004</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_{mutual}</td>
<td>0,23–0,33</td>
<td>1,23–0,33</td>
<td>2,23–0,33</td>
<td>3,23–0,33</td>
<td>4,23–0,33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GMD</th>
<th>4005</th>
<th>4006</th>
<th>...</th>
<th>4010</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_{mutual}</td>
<td>0,23–1,33</td>
<td>1,23–1,33</td>
<td>...</td>
<td>0,23–2,33</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GMD</th>
<th>4015</th>
<th>...</th>
<th>4020</th>
<th>...</th>
<th>4024</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_{mutual}</td>
<td>0,23–3,33</td>
<td>...</td>
<td>0,23–4,33</td>
<td>...</td>
<td>4,23–4,33</td>
</tr>
</tbody>
</table>

In general, each E[i] contributes to (strnd_els)^2 number of mutual inductances starting at GMD[i * (strnd_els)^2] up to and including GMD[(i+1) * (strnd_els)^2 – 1]. In the above example, each E[i] contributes to (strnd_els)^2 = 25 values of mutual L in GMD matrix. These 25 values are divided into (strnd_els) = 5 sections each having (strnd_els) = 5 elements. Within each section, the element number of the second strand stays unchanged (such as 0,33 for GMD[4000] to GMD[4004]) and the element number of the first strand increases from 0 to strnd_els – 1 = 4.
B.5.2. How the SelfGMD Matrix is Constructed

The SelfGMD matrix has \((\text{strnd}_\text{els})*(\text{strnd}_\text{els} + 1) / 2 = 5 \times 6 / 2 = 15\) (for \(M_N = 4_{-2}\)) double precision numbers. GMD between \(e1_{st} - e2_{st}\) is stored in \(\text{SelfGMD}[e1 + e2(e2+1)/2]\). This matrix is stored as a lower triangular matrix. For example for \(M_N = 4_{-2}\) in Table 15, \(\text{SelfGMD}[4] = 1,a-2,a\) is shown as \([4]_{1,a-2,a}\) (this means GMD between element 1 strand a and element 2 strand a, taking "a" as a general strand number).

Table 15. Lower Triangular SelfGMD Matrix Saved as a One Dimensional Matrix.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0,a-0,a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1,a-1,a</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0,a-3,a</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0,a-4,a</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B.6. Constructing the Huge Z Matrix

The huge Z matrix is a symmetric complex matrix because mutual inductances \(L_{ij}\) and \(L_{ji}\) are equal. Only the diagonal elements of huge Z have real parts (resistances) and all non diagonal elements are purely imaginary (inductive reactances).

Here in Figure 36, a lower triangular matrix is shown that represents HugeZ where diagonal elements are triangular blocks and non diagonal elements are rectangular blocks.

If \(M_N = 4_{-2}\), the total number of elements within each strand will be \(M(N-1)+1 = 5\), and therefore, each rectangular block of huge Z will be a 5 x 5 square matrix, and each triangular block will be a 5 x 5 lower triangular matrix as shown in Figure 37.
Notice that in the computer program, the HugeZ matrix is stored as a one dimensional matrix the same way that GNLD and SelfGMD matrices (see Table 15 on page 66) are stored.

8.7. Find_R_L_For_HUGEZ Function

Four integers are transferred to this function, and a complex number representing \( R + jwL \) is returned. The four integers are \( e_1 \) (element number of first strand), \( s_1 \) (first strand number), \( e_2 \) (element number of second strand), and \( s_2 \) (second strand number).

If \( s_1 \) is equal to \( s_2 \), a triangular block of Huge \( Z \) (Fig. 36) is being processed. The function uses the SelfGMD array that contains the self and mutual inductances between the elements of a single strand and \( e_1 \) and \( e_2 \) to obtain the appropriate inductance values.

If \( s_1 \) is different from \( s_2 \), a rectangular block of Huge \( Z \) (Fig. 36) is being processed. The function uses the GMD matrix that contains the mutual inductances between elements of different strands and the symmetry property to find the correct inductance. The E_Array that was created by the 37DSPR.EXE program contains the strand pairs with \( x \), \( y \), and origin symmetry. For example, the strand pairs 2-14 is in E_Array, and all mutual inductances between the elements of these two strands are stored in the GMD matrix. In Figure 12 on page 22, GMD(1,2-4,14) is the GMD between element 1 of strand 2 and element 4 of strand 14. This GMD is equal to (x symmetry) GMD(4,6-1,12) and equal to (y symmetry) GMD(3,5-2,8) and equal to (x symmetry again) GMD(2,3-3,18).

The program goes through the above steps by first checking if the given strand numbers \( s_1 \) and \( s_2 \) are in E_Array. If not, the program checks for \( x \)-symmetric pairs of \( s_1 \) and \( s_2 \), if not, it checks for origin symmetric pairs of \( s_1 \) and \( s_2 \), and finally, it may check for \( y \)-symmetric pairs of \( s_1 \) and \( s_2 \). In any step, if
the pair is found, the program refers to the GMD matrix and extracts the appropriate mutual inductance and returns it to the calling function.

**B.8. Row–Column Subtraction and Symmetry**

As it is explained in section 1.1. on page 5., after the Huge Z matrix is constructed, the first row is subtracted from all other rows. This in effect, replaces the [V] vector with a non-zero element only. Then the first column of the resulting Z matrix is subtracted from all other columns, which replaces the $I_1$ of the [I] vector by $I_{total}$. After the row-column subtraction, the final matrix will be symmetric; there is, therefore, no need to use a square matrix, and a lower triangular one-dimensional matrix is sufficient.

The row–column subtraction is performed at the same time that the Huge Z matrix is constructed, one element at a time from top to bottom and left to right.

**B.9. Reducing the Huge Z Matrix and Saving the Results**

It is explained in section 1.1. on page 5 how, in general, the elements of the voltage and current vectors are exchanged [4]. Here, additional details of the procedure are explained. For simplicity, taking a 3 x3 symmetric impedance matrix, the equation $[V] = [Z] [I]$ is written as:

\[
\begin{bmatrix}
  v1 \\
v2 \\
v3
\end{bmatrix} =
\begin{bmatrix}
z11 & z12 & z13 \\
z12 & z22 & z23 \\
z13 & z23 & z33
\end{bmatrix}
\begin{bmatrix}
i1 \\
i2 \\
i3
\end{bmatrix}
\]

In the reduct procedure, if $v2$ to $vn$ are substituted by their negative values, the impedance matrix remains symmetric at each step of the reduction. On the other hand, after row column subtraction, voltage $v2$ to $vn$ will become zero. Replacing $v2$ to $vn$ by their negative values:

\[
\begin{bmatrix}
v1 \\
-v2 \\
-v3
\end{bmatrix} = \begin{bmatrix}
z11 & z12 & z13 \\
z12 & z22 & z23 \\
z13 & z23 & z33
\end{bmatrix}
\begin{bmatrix}
i1 \\
i2 \\
i3
\end{bmatrix}
\]

Expanding,

\[
v1 = z11 i1 + z12 i2 + z13 i3; \\
v2 = -z12 i1 + z22 i2 - z23 i3; \\
v3 = -z13 i1 + z23 i2 - z33 i3.
\]

Using this last equation to obtain $i3$, \[i3 = -v3/z33 - z13 i1/z33 - z23 i2/z33.\]

Substituting $i3$ into the first two equations:

\[
v1 = z11 i1 + z12 i2 + z13 (-v3/z33 - z13 i1/z33 - z23 i2/z33) = (z11 - z13 z13/z33) i1 + (z12 - z13 z23/z33) i2 - (z13/z33) v3
\]

and

\[
-v2 = z12 i1 + z22 i2 + z23 (-v3/z33 - z13 i1/z33 - z23 i2/z33) = (z12 - z23 z13/z33) i1 + (z22 + z23 z23/z33) i2 - (z23/z33) v3.
\]

Looking at the last three equations, one can write:

\[
\begin{bmatrix}
v1 \\
-v2 \\
i3
\end{bmatrix} =
\begin{bmatrix}
z11 - (z13)^2 / z33 & z12 - z13 z23 / z33 & -z13 / z33 \\
z12 - z23 z13 / z33 & z22 + (z23)^2 / z33 & -z23 / z33 \\
-z13 / z33 & -z23 / z33 & -1 / z33
\end{bmatrix}
\begin{bmatrix}
i1 \\
i2 \\
v3
\end{bmatrix}
\]

68
To simplify the explanation, the contents of the new impedance matrix is replaced by A, B, C, etc. Following the same procedure for finding i2 in terms of v2 and substituting into the vi and i3 equations, i2 will be exchanged with v2 and the above equation will become:

\[
\begin{bmatrix}
  v1 \\
  i2 \\
  i3
\end{bmatrix}
= \begin{bmatrix}
  A & B & C \\
  B & D & E \\
  C & E & F
\end{bmatrix}
\begin{bmatrix}
  i1 \\
  i2 \\
  v3
\end{bmatrix}.
\]

Since the final goal is to obtain element (1,1) of the reduced impedance matrix, there is no need to calculate the values of the last rows of the impedance matrix. That is, while exchanging v3 by i3, there is no need to find the values of the third row of the impedance matrix, only the first two rows are important. Similarly, while exchanging v2 by i2, there is no need to find the values of the second row of the impedance matrix; only the first row values are important. Indeed only element (1,1) is important and not the other elements of the first row. However, the impedance matrix is stored as a one dimensional lower triangular matrix, and the only element to be calculated when reaching row one is element (1,1).

The program saves the final result on disk in a text file. A sample of the output from 37HZR13.EXE is shown in Table 16.

### B.9.1. 37HZR13 Sample Output

Here is a sample of the output produced by the program 37HZR13.EXE with the A option of the automatic mode. Note that all inductance values shown here correspond to the internal inductance of the conductor, but the inductance curves in the Results section, for comparison with Mline results, are the total inductance of the conductor with ground return. The run times are measured on a 80486-33 MHz personal computer.

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>0.000e+00</th>
<th>1.000e-02</th>
<th>5.000e-02</th>
<th>1.000e-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>M N R 0</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>Resistance (ohms/km)</td>
<td>8.527588650e-02</td>
<td>8.527588654e-02</td>
<td>8.527588744e-02</td>
<td>8.527589024e-02</td>
</tr>
<tr>
<td>Inductance (H/km)</td>
<td>5.285104995e-05</td>
<td>5.285104994e-05</td>
<td>5.285104967e-05</td>
<td>5.285104882e-05</td>
</tr>
<tr>
<td>Run Time (hr:min:sec)</td>
<td>0: 0:3</td>
<td>0: 0:3</td>
<td>0: 0:3</td>
<td>0: 0:3</td>
</tr>
<tr>
<td>Radii (mm)</td>
<td>1.729740</td>
<td>1.729740</td>
<td>1.729740</td>
<td>1.729740</td>
</tr>
<tr>
<td>Line Height (m)</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
</tr>
<tr>
<td>Aluminum Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Steel Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Aluminum Relative Permeability</td>
<td>1.000000e+00</td>
<td>1.000000e+00</td>
<td>1.000000e+00</td>
<td>1.000000e+00</td>
</tr>
<tr>
<td>Minimum RAM (bytes)</td>
<td>19252</td>
<td>19252</td>
<td>19252</td>
<td>19252</td>
</tr>
<tr>
<td>Inductance (H/km)</td>
<td>5.285102157e-05</td>
<td>5.285093643e-05</td>
<td>5.284821217e-05</td>
<td>5.283970237e-05</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Run Time (hr:min:sec)</td>
<td>0: 0:3</td>
<td>0: 0:3</td>
<td>0: 0:3</td>
<td>0: 0:3</td>
</tr>
<tr>
<td>Radii (mm)</td>
<td>1.729740</td>
<td>1.729740</td>
<td>1.729740</td>
<td>1.729740</td>
</tr>
<tr>
<td>Line Height (m)</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
</tr>
<tr>
<td>Steel Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Aluminum Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Aluminum Relative Permeability</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
</tr>
<tr>
<td>Minimum RAM (bytes)</td>
<td>19252</td>
<td>19252</td>
<td>19252</td>
<td>19252</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>5.000e+01</th>
<th>6.000e+01</th>
<th>1.000e+02</th>
<th>5.000e+02</th>
</tr>
</thead>
<tbody>
<tr>
<td>M N R 0</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>Resistance (ohms/km)</td>
<td>8.620112384e-02</td>
<td>8.660248624e-02</td>
<td>8.887089029e-02</td>
<td>1.332864441e-01</td>
</tr>
<tr>
<td>Inductance (H/km)</td>
<td>5.257015801e-05</td>
<td>5.244838516e-05</td>
<td>5.176105277e-05</td>
<td>3.882569495e-05</td>
</tr>
<tr>
<td>Run Time (hr:min:sec)</td>
<td>0: 0:3</td>
<td>0: 0:3</td>
<td>0: 0:3</td>
<td>0: 0:3</td>
</tr>
<tr>
<td>Radii (mm)</td>
<td>1.729740</td>
<td>1.729740</td>
<td>1.729740</td>
<td>1.729740</td>
</tr>
<tr>
<td>Line Height (m)</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
</tr>
<tr>
<td>Steel Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Aluminum Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Aluminum Relative Permeability</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
</tr>
<tr>
<td>Minimum RAM (bytes)</td>
<td>19252</td>
<td>19252</td>
<td>19252</td>
<td>19252</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>1.000e+03</th>
<th>5.000e+03</th>
<th>1.000e+04</th>
<th>5.000e+04</th>
</tr>
</thead>
<tbody>
<tr>
<td>M N R 0</td>
<td>2 3 1 3</td>
<td>2 3 1 5</td>
<td>4 4 1 3</td>
<td></td>
</tr>
<tr>
<td>Resistance (ohms/km)</td>
<td>1.701295549e-01</td>
<td>3.248302953e-01</td>
<td>5.498413733e-01</td>
<td>4.37691981e+00</td>
</tr>
<tr>
<td>Inductance (H/km)</td>
<td>2.999090483e-05</td>
<td>2.132589891e-05</td>
<td>1.989521306e-05</td>
<td>2.29607304c-05</td>
</tr>
<tr>
<td>Run Time (hr:min:sec)</td>
<td>0: 0:3</td>
<td>0: 1:16</td>
<td>0: 1:23</td>
<td>0: 14:33</td>
</tr>
<tr>
<td>Radii (mm)</td>
<td>1.729740</td>
<td>1.729740</td>
<td>1.729740</td>
<td>1.729740</td>
</tr>
<tr>
<td>Line Height (m)</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
</tr>
<tr>
<td>Steel Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Aluminum Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Aluminum Relative Permeability</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
</tr>
<tr>
<td>Minimum RAM (bytes)</td>
<td>19252</td>
<td>2087020</td>
<td>2087020</td>
<td>2087020</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>1.000e+05</th>
<th>5.000e+05</th>
<th>1.000e+06</th>
<th>5.000e+06</th>
</tr>
</thead>
<tbody>
<tr>
<td>M N R 0</td>
<td>4 5 1 3</td>
<td>6 6 1 7</td>
<td>6 6 1 9</td>
<td>6 6 1 1</td>
</tr>
<tr>
<td>Resistance (ohms/km)</td>
<td>7.845298575e+00</td>
<td>2.272808254e+01</td>
<td>4.103950308e+01</td>
<td>2.383459102e+02</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Inductance (H/km)</td>
<td>2.425358324e-05</td>
<td>2.196185705e-05</td>
<td>2.05036831e-05</td>
<td>1.821123543e-05</td>
</tr>
<tr>
<td>Run Time (hr:min:sec)</td>
<td>0:30:5</td>
<td>2:53:22</td>
<td>3:2</td>
<td>3:14:50</td>
</tr>
<tr>
<td>Radii (mm)</td>
<td>1.70672, 1.523172, 1.613942, 1.678969, 1.729740</td>
<td>1.529218, 1.606554, 1.651935, 1.684194, 1.709248, 1.729740</td>
<td>1.584477, 1.640590, 1.673463, 1.696808, 1.714928, 1.729740</td>
<td>1.662784, 1.68679, 1.703831, 1.714583, 1.729740</td>
</tr>
<tr>
<td>Line Height (m)</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
<td>20.000000</td>
</tr>
<tr>
<td>Aluminum Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Steel Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
<td>3.46620e+07</td>
</tr>
<tr>
<td>Aluminum Relative Permeability</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
<td>1.00000e+00</td>
</tr>
<tr>
<td>Minimum RAM (bytes)</td>
<td>3559892</td>
<td>11800372</td>
<td>11800372</td>
<td>11800372</td>
</tr>
<tr>
<td>Frequency (Hz)</td>
<td>1.000e+07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M N R 0</td>
<td>6.61.11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resistance (ohms/km)</td>
<td>5.215313610e+02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inductance (H/km)</td>
<td>1.896013710e-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run Time (hr:min:sec)</td>
<td>3:14:51</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radii (mm)</td>
<td>1.682071, 1.700510, 1.711297, 1.718951, 1.724888, 1.729740</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line Height (m)</td>
<td>20.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aluminum Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steel Conductivity (S/m)</td>
<td>3.46620e+07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aluminum Relative Permeability</td>
<td>1.00000e+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum RAM (bytes)</td>
<td>11800372</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Date and Time</td>
<td>Tue Jan 19 09:31:37 1993</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
C. Analytic GMD Formulas

Figure 38 shows two rectangles in Cartesian coordinates.

The natural logarithm of the GMD between the two rectangles is derived as:

\[
\ln (GMD_{\text{rectangle}}) = \frac{1}{2(b-a)(g-f)(e-c)(k-h)} \int_{h}^{k} \int_{c}^{e} \int_{a}^{b} \int_{d}^{g} \ln \left[ \left( x_2 - x_1 \right)^2 + \left( y_2 - y_1 \right)^2 \right] dx_1 dy_1 dx_2 dy_2
\]

It is possible to evaluate this integral analytically. According to [14], if

\[
F(x, y) = \left( \frac{x^4 - 6x^2 y^2 + y^4}{24} \right) \ln \left( x^2 + y^2 \right) - \frac{xy}{3} \left( x^2 \arctan \left( \frac{y}{x} \right) + y^2 \arctan \left( \frac{x}{y} \right) \right); \text{ then,}
\]

\[
\frac{\partial^4 F(x_1 - x_2, y_1 - y_2)}{\partial x_1 \partial y_1 \partial x_2 \partial y_2} = -\ln \left[ \left( x_2 - x_1 \right)^2 + \left( y_2 - y_1 \right)^2 \right] - \frac{25}{6}.
\]

To find the Geometric Mean Radius (GMR) of a rectangle, the two rectangles have to coincide, that is the limits of integration have to be changed such that a = c, b = e, f = h, and g = k.

Figure 39 shows two elementals that belong to the same strand.

The natural logarithm of the GMD between the two elementals of Figure 39 is derived as:

\[
\ln (GMD_{\text{elemental}}) = \frac{1}{(\theta_4 - \theta_3)(\theta_4 - \theta_2)(\theta_2 - \theta_1)(\theta_3 - \theta_1)} \int_{r_2}^{r_1} \int_{\theta_2}^{\theta_4} \int_{r_3}^{r_4} \int_{\theta_3}^{\theta_4} \ln \left| l_1 \alpha - l_2 \beta \right| dl_1 \ d\alpha \ d\beta \ dl_2
\]

where \( \ln \left| l_1 \alpha - l_2 \beta \right| = \ln \left| l_1 \cos \alpha + l_2 \sin \alpha \right| = \frac{1}{2} \ln \left[ (l_1 \cos \alpha - l_2 \cos \beta) \right] = \frac{1}{2} \ln \left[ l_1^2 + l_2^2 - 2 l_1 l_2 \cos (\alpha - \beta) \right] \).
To solve this integral, the author has used a number of possible methods, but unfortunately has been unsuccessful. One method that comes to mind is to convert the integral from Polar to Cartesian coordinates by the following substitutions:

\[ l_1 = \sqrt{x_2^2 + y_2^2}, \quad \alpha = \arctan\left(\frac{y_2}{x_2}\right), \quad l_2 = \sqrt{x_1^2 + y_1^2}, \quad \beta = \arctan\left(\frac{y_1}{x_1}\right) \]

therefore,

\[
\frac{1}{2} \iiint \ln \left[ l_1^2 + l_2^2 - 2 l_1 l_2 \cos(\alpha - \beta) \right] dl_1 \, dl_2 \, d\alpha \, d\beta =
\]

\[
\frac{1}{2} \iiint \ln \left[ (x_2 - x_1)^2 + (y_2 - y_1)^2 \right] \frac{1}{\sqrt{x_1^2 + y_1^2} \cdot \sqrt{x_2^2 + y_2^2}} \, dx_1 \, dy_1 \, dx_2 \, dy_2
\]

It was not possible to solve this integral in Cartesian coordinates either. That is why the computer program uses points within each elemental to find the GMD between two elementals. Because of complexity, the analytic formula for the GMD between elementals that do not have the same centre has not been derived.

To obtain the GMR of an elemental, the following parameters have to be set equal: \( r_1 = r_3, \quad r_2 = r_4, \quad \theta_1 = \theta_3, \quad \text{and} \quad \theta_2 = \theta_4 \).