THE SHORT FAT DIPOLE
Development in APL of a MoM Solution

Douglas B. Miron
EE Dept., Box 2220
South Dakota State University
Brookings, SD 57007

Abstract

This paper describes an effort to find the best profile for a figure-of-revolution, center-fed, electrically-small dipole. It includes presentations on equation development, singularity treatment, code development, verification, and performance.

I. Prologue

The work described in this paper was undertaken to find a minimum-Q shape for an electrically-small dipole to operate over the HF band, 3-30 MHz. The starting point was the 1969 paper by Mautz and Harrington[1] on bodies of revolution. Their formulation for the integral equation and their expansion and testing functions were used without scaling. The dipole is driven only in the tangential direction, so it was assumed there are no ϕ-components to the current density. This simplified the problem considerably. There are numerous ways to handle the potential-function singularity. After trying a couple of others, subtraction of the electrostatic potential from the radiation potential function[2] was chosen. The electrostatic potential function is integratable in ϕ to a known special function, and that function is numerically integratable so that the singularity is expressed in the impedance-matrix elements in a stable way.

This paper is divided into the following sections:

One purpose of this paper is to demonstrate that programming in an interactive array-processing language (APL) makes the most efficient use of professional time. The Appendix briefly describes its basic syntax and lists the definitions of symbols and functions as used in this paper. Specific features are described as they are used in Sections II through IV . In Section V the work is tried by comparison with two classic results. In Section VI a search for a shape that gives an antenna Q close to

the theoretical minimum, $1/(\beta a)^3$ [3], is outlined. In Section VII the hardware-software performance issues are examined.

II. Geometry, the Basis Functions, and Counting

The dipole is symmetric and center-fed. It must fit in a 2 m cube and be easy to manufacture. It would be a good thing then, if its profile were made of connected straight-line segments. Figure 1 shows an example of a simple shape, along with some of the coordinates to be used in the calculations. All prospective shapes are to be figures of revolution about the z axis, with the feed gap at z=0. An x axis is necessary from which to define the rotational angle ϕ. Cylindrical radius ρ is used both for the shape description and in the calculations. The fundamental coordinate for integration is the tangential distance t, which runs from t=0 at z=-1 m to $t=t_m$ at z=+1 m, following the antenna's profile.

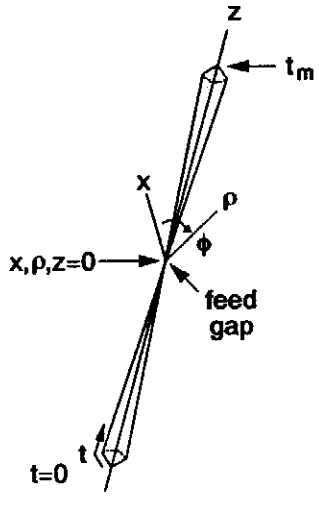

Figure 1. A representative dipole as a figure of revolution about the z axis.

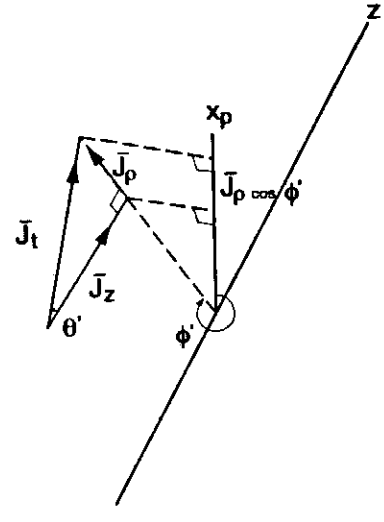

Figure 2. x and z projections of a tangential vector. x_ρ is parallel to the x axis.

One must know the projection of the current density vector at one place onto the surface at another place, to calculate the vector magnetic potential and surface Ė. Since the applied field is in the z direction, it is assumed there are no ϕ components of current or electric intensity. Figure 2 shows the components of $\bar{J}_t$ projected onto the z and x axes.

$$J_\rho = J_t \sin\theta', \quad J_z = J_t \cos\theta', \quad J_x = J_t \sin\theta' \cos\phi' \tag{1}$$

For the purpose of doing a dot product with a unit tangential vector at the observation point, the last projection should be by way of $\cos(\phi-\phi')$, but the rotational symmetry allows $\phi=0$.

The triangle function was used as the basic element of the current expansion and weighting function series. The triangle, which is shown in Figure 3, was divided by $\rho(t)$ to make the complete scalar function. Since the integrals were discretized it was easier to denote the triangle by a subscript and deal with its relation to t through their counters, which is displayed in Figure 4. However, for the sake of the functional notation used in the next section, one may say that

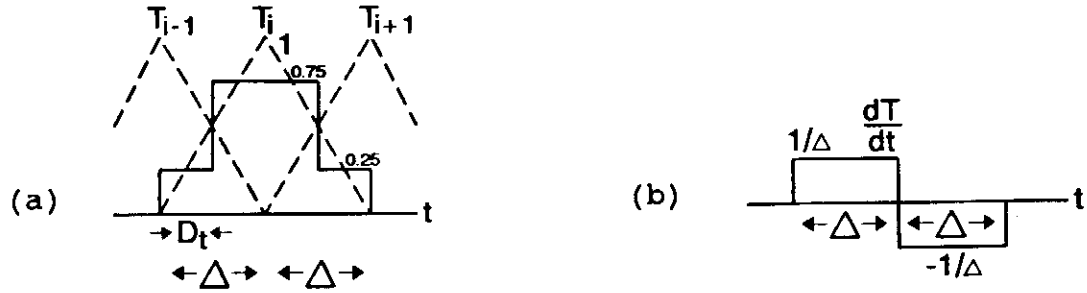

Figure 3. (a) Triangle function and a piecewise-constant approximation. (b) The derivative of a triangle function.

$$\tilde{W}_i(t) = \hat{u}_t W_i(t) = \hat{u}_t \frac{T_i(t)}{\rho(t)} \tag{2}$$

where $\hat{u}_t$ is a unit vector, $\tilde{W}_i$ are the vector basis functions, and their divergences are

$$\nabla\cdot\tilde{W}_i = \frac{1}{\rho}\frac{dT_i}{dt} \tag{3}$$

The dipole can be described by assigning vectors of ρ and z values at the corners in the profile of the half-dipole. For example

```
RHD←0.01 1 0 ◇ ZD←0 0.9 1
```

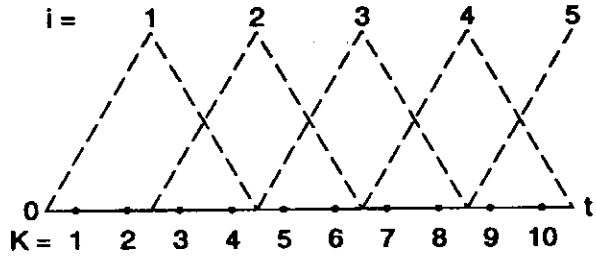

Figure 4. Relationship between the counter for t, $t_k=(k-\frac{1}{2})Dt$, and the triangle counter. T_i spans from k=2i-1 to k=2i+2.

describes a dipole which has a radius of 0.01 m at the feedpoint, cones out to a 1 m radius at z=0.9 m and cones in to a point at z=1 m. The full dipole description is formed by reversing ZD and RHD, negating the reversed ZD values, and attaching them to the originals.

```
ZV←(-⌽ZD),0,ZD                                    (4)

RHV←(⌽RHD),RHD[1],RHD                             (5)
```

The extra point in the middle is a convenience so the first z point doesn't have to be zero, and a straight pipe may be inserted before the shape takes off.

Figure 5 shows the basis for a scheme to generate the coordinate variables as functions of t. Since the profile is a straight line between corners, the values of ρ and z on a segment are displacements from their corner values which are sinθ and cosθ times the t displacement from TC[K], where K is a corner counter. The nub of the scheme is to generate the TC vector, and then write a function to find the index of the last corner for a given t. The increase in t between corners is the length of the profile segment.

```
TC[K+1]←TC[K]+(((RHV[K+1]-RHV[K])*2)+(ZV[K+1]-ZV[K])*2)*0.5
```

This expression could be put in a loop and stepped through to produce the vector TC. However, APL has a cumulative sum operator, '+\', and there are a couple of simple ways to generate the pairwise-difference of a vector. Here is one as a defined function.

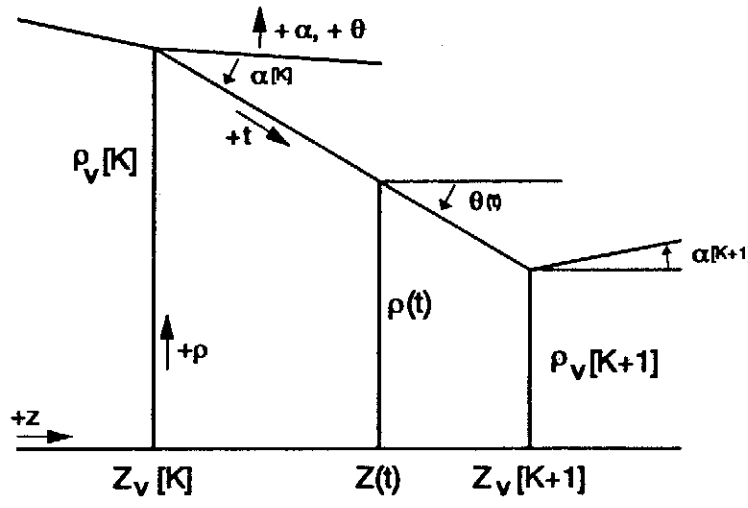

Figure 5. Illustration of a piecewise-linear antenna profile section. z and t increase to the right, ρ increases upward, α and θ are positive ccw.

```
     ∇DIFF[□]∇
[0]  Z←DIFF X
[1]  Z←(1↓X)-¯1↓X                                    (6)
```

Now the expression for TC is written

```
     TC←0,+\(((DIFF RHV)*2)+(DIFF ZV)*2)*0.5         (7)
```

For a single value of t, one way to find the index of TC just left of t is to compare the value with TC. TC<T will produce a Boolean vector with 1 for each corner up to the nearest one on the left of T, and 0 for the rest. The index value for the corner wanted is the sum of these 1s, +/TC<T. To do this for a vector of t values, the outer product operator can be used. TC∘.<,T takes each value of TC and compares it to all the values in T, producing a row in the result. The result has as many columns as there are elements in T and as many rows as there are elements in TC. The ',' next to T makes a vector of a single-valued T. To get the index values now it is necessary to sum down the columns of the Boolean matrix. These steps are put in a defined function called REGION.

```
∇REGION[□]∇
[0]  S←REGION T
[1]  S←+⌿TC°.<,T                                        (8)
```

In this function TC is used as a global variable, T and S are local variables through which data is passed into and out of the function.

For the purpose of numerical integration, the profile of the antenna, which is the range of t, is divided into increments of Δ and Dt as in Figure 3. The center of the dipole has to be a center point for an expansion function and a boundary for a Δ segment. Let NP be the number of Δ segments in the half-dipole.

```
DEL←TC[⍴TC]÷2×NP and DT←DEL÷2                           (9)
```

NP is also the number of unique current coefficients, so it is quite important later on.

Finally, $\beta=2\pi/\lambda$ (BT←○2÷LAMBDA) and β^2 (BS←BT×BT) are important values in the calculations because β measures the intervals in wavelength. LAMBDA serves as a global variable to hold λ.

III. The Integral Equation

IIIA Field Theory

The antenna is a closed, perfectly-conducting (initial assumption) surface so that the currents flowing on it and the applied voltage must produce tangential E fields that add to zero. The field due to the surface currents is

$$\dot{E}^s(\dot{r}) = -\int_0^{t_m}\int_0^{2\pi}\left[j\omega\mu\dot{J}(t')g(R)-\nabla g(R)\frac{\nabla'\cdot\dot{J}(t')}{j\omega\epsilon}\right]\rho'd\phi'dt' \qquad (10)$$

The distance function is

$$R = \sqrt{\rho^2+\rho'^2-2\rho\rho'\cos(\phi-\phi')+(z-z')^2} \qquad (11)$$

and the propagation function is

$$g(R) = \frac{e^{-j\beta R}}{4\pi R} \qquad (12)$$

The δ-gap source model was used so the incident E is

$$\bar{E}^i(\bar{r}) = \hat{u}_z V_o \delta(z) \tag{13}$$

and the boundary condition requires that (13) equal the negative of the tangential component of (10). This would be an ill-conditioned integral equation on several counts, but MoM smooths it out somewhat by multiplying both sides by a weighting, or averaging, function and then integrating the products over the surface of the antenna. Using a set of vector weighting functions and the vector dot product (inner product) also picks out the tangential component on each side. Representing a weighting function by

$$\bar{W}(t) = \hat{u}_t W(t) \tag{14}$$

the smoothed equation becomes

$$\int_0^{t_m}\int_0^{2\pi} \bar{W}\cdot\bar{E}^i \rho d\phi dt = \int_0^{t_m}\int_0^{2\pi}\int_0^{t_m}\int_0^{2\pi} j\beta\eta_o \left[\bar{W}(t)\cdot\bar{J}(t')g(R) + \bar{W}(t)\cdot\nabla g(R)\frac{\nabla'\cdot\bar{J}(t')}{\beta^2}\right] \rho' d\phi' dt' \rho d\phi dt \tag{15}$$

Again, from symmetry there are no ϕ-components to the sources or fields, so the left side integrates to

$$2\pi\rho(t_m/2)W(t_m/2)V_o$$

The first dot product on the right can be represented by the unit vectors involved,

$$\hat{u}_t\cdot\hat{u}_{t'} = \cos\theta\cos\theta' + \sin\theta\sin\theta'\cos(\phi-\phi') \tag{16}$$

from Figure 2 and the discussion around equation (1). The second dot product involves the gradient of g(R), which is already a singular function. Although a second-order singularity can be handled [4,5], Harrington and Mautz [1] used the fact that the body is a closed surface to replace the integral of $\bar{W}(t)\cdot\nabla g(R)$ in an integration by parts with $-g(R)\nabla\cdot\bar{W}(t)$.

R and the unit-vector dot product involve $\cos(\phi-\phi')$. These are the only rotational terms, so they were collected and the angle integrals done first. Because the functions really depend on the difference between the two angles, if one integrates with respect to one angle, the dependence on the other disappears. The ϕ-dependent terms can be collected in two integral functions as

$$G_0 = 2\pi\int_0^{2\pi} g(R)\,d\phi = \int_0^{\pi}\frac{e^{-j\beta R}}{R}d\phi \tag{17}$$

and likewise

$$G_1 = \int_0^{\pi}\cos\phi\frac{e^{-j\beta R}}{R}d\phi \tag{18}$$

Collecting all these substitutions brought the integral equation to

$$\frac{\rho(\frac{t_m}{2})W(\frac{t_m}{2})V_o}{j60\beta} = \int_0^{t_m}\int_0^{t_m}\{W(t)J(t')(\cos\theta(t)\cos\theta(t')G_0+$$

$$\sin\theta(t)\sin\theta(t')G_1) - G_0\nabla\cdot\mathbf{W}(t)\nabla'\cdot\mathbf{J}(t')/\beta^2\}\rho\rho'dt'dt \tag{19}$$

IIIB The Integrals in ϕ

G_0 and G_1 hold the integrations in ϕ, which must be done before the outer integrals in t and t'. They also hold the 1/R singularity. There must be some advantage to the rotational symmetry besides the simplicity of the setup, perhaps it would be in the evaluation of these integrals. At first, writing

$$\beta R=\beta(R-R_o)+\beta R_o,\quad R_o^2=\rho^2+(z-z')^2$$

was tried, R_o being the distance to the axis under the source point. For $\beta(R-R_o)<1$ a few terms in the exponential series were used, which then led to terms involving elliptic integrals of the three kinds. While this was educational, and some errors in [6] were found, the result was rather complex and slow. An interesting discovery was that, away from the singularity, it is faster to compute the integral by the piecewise-constant series approximation than to use the polynomial-and-log approximation from [7]. The speed crossover was at 20 terms in the series for six-figure accuracy.

The next approach tried was an old one used recently by Simpson, et. al. [2]. They split the exponential instead of βR.

$$\frac{e^{-j\beta R}}{R} = \frac{1}{R} + \frac{e^{-j\beta R}-1}{R} \tag{20}$$

This method removes the singularity from the propagation function to a static function which integrates to elliptic integrals. These two parts can be thought of as the circuit-element part and the radiation part, concepts which are explored in [2]. Define

$$G_{0E} = \int_0^{\pi} \frac{1}{R} d\phi \tag{21}$$

$$G_{1E} = \int_0^{\pi} \frac{\cos\phi}{R} d\phi \tag{22}$$

$$G_{0\omega} = \int_0^{\pi} \frac{e^{-j\beta R}-1}{R} d\phi \tag{23}$$

$$G_{1\omega} = \int_0^{\pi} \frac{e^{-j\beta R}-1}{R} \cos\phi d\phi \tag{24}$$

One can find the elliptic-function version of G_{0E} either by a succession of substitutions or from [6] as

$$G_{0E} = \frac{2}{\sqrt{a+b}} K(r) \tag{25}$$

with $a = \rho^2+\rho'^2+(z-z')^2$, $b = 2\rho\rho'$ and $r^2 = \frac{2b}{a+b}$

K(r) is the complete elliptic integral of the first kind whose defining equation is

$$K(r) = \int_0^{\pi/2} \frac{dx}{\sqrt{1-r^2\sin^2 x}} \tag{26}$$

Using the same substitutions and the redundancy[8]

$$\sin^2 z = \frac{1}{r^2} - \frac{1}{r^2}(1-r^2\sin^2 z) \tag{27}$$

provides

$$G_{1E} = \frac{2}{r^2\sqrt{a+b}} \left[2K(r)-2E(r)-r^2K(r) \right] \tag{28}$$

E(r) is the complete elliptic integral of the second kind, defined by

$$E(r) = \int_0^{\pi/2} \sqrt{1-r^2\sin^2 x}\, dx \tag{29}$$

No closed-form or special-function versions for $G_{0\omega}$ and $G_{1\omega}$ were found, but the integrands are smooth and well-behaved so they were simply approximated by the same piecewise-constant interval integration as used by MoM.

IIIC The Singularity

The development in IIIB moved from the static potential function, which has a 1/R singularity through the integration in ϕ to the elliptic integrals. [7] gives a quite accurate approximation which explicitly includes a logarithm which goes to infinity when $r \to 1$, which occurs when $t'=t$. For $t-t'$ small compared to ρ the argument of the approximation is proportional to $t-t'$, so the approximation can be integrated analytically. However, there are other functions which multiply the Gs, and $t-t'$ is not necessarily small compared to ρ over the interval in the discrete approximation to the integrals.

The basic approach by MoM to solving equation (20) is to break up the integrals into sums, each term of which approximates the integral over a small interval by the product of the interval length and the integrand's value at the center of the interval. This is the piecewise-constant approximation. The question then is, what value should be used to represent the G functions for intervals in which $t' \to t$? Working the charged-cylinder electrostatic problem[9, 10] showed that displacing the source and observation points by a small amount compared to the interval size gave a result independent of this displacement, once it was small enough, as far as plots of the charge density were concerned. In the present case, the results were poor for small NP, even having sign errors in the terminal impedance. Apparently the singularity was over-represented. The next best choice seemed to be to use the average value of each G function, which means numerically integrating them over the interval since no analytical integration was available. Reference [7] provided some formulas and values for functions with a logarithmic singularity. The numerical approximation has the form

$$\int_0^1 f(x)\log(x)\,dx \cong \Sigma w_i f(x_i) \tag{30}$$

where w_i and x_i are given in tables for various numbers of samples per interval. The form needed was

$$IG = \int_{-\delta}^{+\delta} G(y)\,dy = 2\int_0^{\delta} G(y)\,dy \tag{31}$$

since G is even in y. If $y=\delta x$, then the upper limit becomes 1,

$$IG = 2\int_0^1 G(\delta x)\,\delta dx \tag{32}$$

To use (30), $f(x) = G(\delta x)/\log(x)$,

$$IG \cong 2\delta\Sigma w_i \frac{G(\delta x_i)}{\log(x_i)} \tag{33}$$

The problem at hand was to integrate over a Dt interval with respect to t′ when t is at the interval center. y corresponds to t′-t, δ to Dt/2. The variables in the G functions were expressed as functions of ρ, ρ', z and z′, so they were re-expressed in terms of t′-t, ρ and $\sin\theta(t)$. This was made possible by assuming t′ and t are on the same profile segment. The geometry of this assumption gives

$$(\rho-\rho')^2+(z-z')^2 = (t'-t)^2 \tag{34}$$

$$\rho' = \rho+(t'-t)\sin\theta \tag{35}$$

so that

$$a = (t'-t)^2+2\rho(\rho+(t'-t)\sin\theta) \tag{36}$$

$$b = 2\rho(\rho+(t'-t)\sin\theta) \tag{37}$$

$$a+b = (t'-t)^2+4\rho(\rho+(t'-t)\sin\theta) \tag{38}$$

$$a-b = (t'-t)^2 \tag{39}$$

$$m = 1-r^2 = \frac{a-b}{a+b} = \frac{(t'-t)^2}{a+b} \tag{40}$$

m is the most convenient variable in which to express the elliptic integral approximations.

IV Conversion to Discrete Form

IVA Using MoM

On examining equation (20), one can see that each weighting function is multiplied by a ρ and each occurrence of the current density is multiplied by a ρ'. Furthermore, the first derivative of both functions is required. It makes good sense, in hindsight, that Mautz and Harrington[1] chose to represent both functions as a series of basis functions having a first derivative and being a simple function divided by ρ. The weighting functions are

$$\bar{W}_i(t) = \hat{u}_{ti}\frac{T_i(t)}{\rho(t)}, \quad 1\leq i\leq N \tag{41}$$

The surface current density is

$$\bar{J}(t') = \sum_{k=1}^{N}\hat{u}_{t'k}J_{tk}\frac{T_k(t')}{\rho(t')} \tag{42}$$

One can show, by taking the textbook approach of sketching a differential-sized box on the antenna surface and applying the definition of divergence as a Gauss's Law limit, that

$$\nabla\cdot\bar{W}_i = \frac{1}{\rho}\frac{\partial(\rho W_i)}{\partial t} = \frac{1}{\rho}\frac{dT_i}{dt} \tag{43}$$

and

$$\nabla'\cdot\bar{J} = \sum_{k=1}^{N} J_{tk}\frac{1}{\rho'}\frac{dT_k}{dt'} \tag{44}$$

There can easily be a notation and counting confusion because the value of t at the middle of a triangle is never actually used in the integral computations. It is simpler to express the integrals, though, using this value, so let it be τ_i for triangle T_i. Then, since the triangle functions are nonzero over a 2Δ domain the integrals are shortened up. The application of each weighting function to equation (20) generates a separate equation, the full sequence produces a set of N equations for the N J_t amplitudes.

$$\frac{T_i(\frac{t_m}{2})V_o}{j60\beta} = \sum_{k=1}^{N} J_{tk}\int_{\tau_i-\Delta}^{\tau_i+\Delta}\int_{\tau_k-\Delta}^{\tau_k+\Delta}\Big\{T_i(t)T_k(t')(\cos\theta\cos\theta' G_0+\sin\theta\sin\theta' G_1)$$

$$-\frac{G_0}{\beta^2}\frac{dT_i(t)}{dt}\frac{dT_k(t')}{dt'}\Big\}dt'dt,\quad 1\le i\le N \tag{45}$$

The left-hand sides of this equation sequence are zero except for i=NP because only that triangle function is nonzero at $t_m/2$. The results of the double integration on the right-hand sides are functions only of i and k, so they can be symbolized by the standard Z_{ik}.

$$V_i = \sum_{k=1}^{N} J_{tk}Z_{ik},\quad 1\le i\le N \tag{46}$$

IVB Preparing the Way; Geometry

The function values needed in each Z_{ik}, $\sin\theta$, T_i, etc., are used over and over again, so it doesn't pay to recompute them each time. The basic strategy was to precompute them as vectors and matrices, and call the needed data by their indices during the Z element calculation.

APL provides a function to generate a vector of index values, '⍳'. Any set of values which can be written as a function of an index can be generated as a vector, making a loop, with its repetitive interpretation, unnecessary. One can generate a function of any number of indices by using the outer product operator with the individual index vectors which will produce an array with as many axes as there are index variables. While this is quick to execute, large temporary outer products

can eat up storage to the point where there isn't enough, which will generate a WS FULL (WS for workspace) error message.

The vectors and static G functions are calculated in a function SGENERATE.

```
[0]  SGENERATE;K;T;THT;L;T1;N1;N2
```

All the variables in the header are local.

```
[1]  T1←□TS ◇ N1←2×NP ◇ N2←4×NP
```

```
[2]  ZT←ZA T←DT×¯0.5+⍳N2 ◇ RHT←RHO T ◇ THT←TH T ◇ COST←2○THT
     ◇ SINT←1○THT
```

Remember that the interpreter works from right to left in a statement. The statements in a line are executed starting with the leftmost one. The first statement in line [2] starts by generating a vector of integers from 1 to N2, then adds ¯0.5 to these integers, then multiplies them by DT and assigns the result to T, which is the vector of t values used in all subsequent functions. The T is also an input argument to a function ZA which calculates the corresponding vector of z(t) values, ZT. The next two statements in line [2] calculate ρ(t) and θ(t) vectors, RHT and THT. All three of these functions use the simple segment geometry and the function REGION (6). The next two statements in line [2] produce cosθ and sinθ using the circle function with a left argument to indicate which of the 15 transcendental functions is wanted.

```
[3]  TRT←(1 3 3 1)÷4 ◇ DTRT←(1 1 ¯1 ¯1)÷DEL
```

These are vectors to hold the sample values of the triangle function and its derivative.

```
[4]  G0←G1←(N1,N2)⍴0 ◇ K←1
```

G0 and G1 are matrices which hold the static potential integrals G_{0E} and G_{1E}. These matrices must be symmetric about their main diagonal, regardless of the geometry of a problem, and they are also symmetric about a line between rows N1 and N1+1 because of the geometric symmetry of the dipole. Use was not made of the diagonal symmetry because of the complexity of the indexing required. The row symmetry was used.

```
[5]  (⍳N1) REALGREEN ⍳N2 ◇ k←1
```

REALGREEN is a function which calculates the values for the static G functions and puts them in the arrays G0 and G1. Its input arguments are sets of row and column indices. The workings of REALGREEN and IG will be discussed below.

```
[6]  GL:L←IG RHT[K],SINT[K] ◇ G0[K;K]←L[1] ◇ G1[K;K]←L[2]
     ◇ →(N1≥K←K+1)/GL
```

This is a one-line loop to replace the diagonal elements of G0 and G1 with the average values of G_{0E} and G_{1E} on the intervals where they are singular. The calculation on these intervals in REALGREEN were prevented from overflow by replacing zeros of m with 1E⁻50. It would have been more efficient to send the results directly to G0 and G1 inside IG, but it was left in its development form which gives an explicit result.

```
[7]  G0←G0,[1]⌽G0[N1+1-⍳3;] ◇ G1←G1,[1]⌽G1[N1+1-⍳3;]
```

Not all the possible values of G0 and G1 are ever used. Again because of geometric symmetry, there are only NP unique values of J_{tk}, so only NP equations are needed. The expression N1+1-⍳3 generates a vector N1, N1-1,N1-2 which picks these three rows out of the matrix. No column index is specified so the whole row is taken. This three-row block is then reversed end-for-end, and attached to the original matrix under its columns, making the result a 2NP+3 by 4NP array.

```
[8]  (+/0 0 0 3600 60 1 1E⁻3×□TS-T1) 'SEC, STATIC G0 AND G1.'
```

This line calculates and displays the elapsed time in seconds. The mixture of numeric and character data in a vector is a feature of APL2. In standard APL the number would have to be converted to characters to be displayed in the same vector.

```
[9] □SOUND SCALE,[1.5]100
```

This line plays an ascending C major scale to get the user's attention when the function is finished executing.

IVC Preparing the Way; Potential Integrals

REALGREEN is so named because its results are Green's functions of a sort.

```
[0]  OX REALGREEN SX;A;B;SQAB;M;K;E
```

OX and SX stand for observation-point indices and source-point indices, respectively. These are the input arguments, the other variables listed in the header are local to the function and named to correspond to variables in (25) and (28). SQAB=$\sqrt{a+b}$.

```
[1]  B←2×RHT[OX]∘.×RHT[SX] ◇ AB←RHT[OX]∘.+RHT[SX]
     ◇ A←ZT[OX]∘.-ZT[SX]
```

In each of the statements in line [1] the generalized outer-product operator forms a matrix of all combinations of the left argument elements with the right argument elements, with the arithmetic function placed between them. B has as many rows as there are values in OX and as many columns as values in SX, and each element of B is $2\rho(t_i)\rho(t_k)=2\rho\rho'$. Each element of AB is $\rho+\rho'$, and each element of A is $z-z'$.

```
[2]  AB←(A×A)+AB×AB ◇ A←0ρ0 ◇ M←1E¯50+ZERO 1-2×B÷AB
```

The first statement forms a matrix of a+b values. The second statement reduces A to an empty vector to reduce storage. The third statement calculates an array of m values, defined in (40), as an argument for the elliptic integrals in the next line. The function ZERO replaces each number whose magnitude is less than 1E¯14 by zero.

```
[3]  K←(AE1 MPOLY M)-(⍟M)×BE1 MPOLY M ◇ E←(AE2 MPOLY M)
     -(⍟M)×BE2 MPOLY M
```

MPOLY is a function whose left argument is a vector of polynomial coefficients. It evaluates the polynomial for each element in M and returns the results in the same shape as M. AE1, BE1, AE2, and BE2 are the coefficients in the polynomial-and-log approximations for the elliptic integrals given in [7 p591-592, 5 terms]. ⍟M gives the natural log of the values in M.

```
[4]  G0[OX;SX]←2×K÷SQAB←AB*0.5
     ◇ G1[OX;SX]←(2÷B)×(K×(AB-B)÷SQAB)-E×SQAB
```

Line [4] expresses and assigns the blocks of element values to the indexed parts of G0 and G1. The first statement is a direct copy of (25). The second statement is a reworking of (28) to minimize the number of variables holding intermediate data.

REALGREEN called its geometry values by indexing the previously-constructed vectors for ρ and z. The average value for each G function over a singular interval requires that t′ values be calculated within the interval for the integration formula, and the G functions evaluated at these t′ values. Therefore one can't use REALGREEN and some of the same statements must be rewritten. This is done in the function IG.

```
[0]  Z←IG Y;X;K;E;A;B;AB;SQAB;XQ;M
```

Z and Y are output and input arguments for the function, each a two-element vector as can be seen from its use in line [6] of SGENERATE. The local variables listed in the header mostly have the same uses as they do in REALGREEN.

```
[1]  X←GIX×DT÷2 ◇ XQ←X×X ◇ B←2×Y[1]×Y[1]+X×Y[2]
     ◇ AB←XQ+2×B ◇ A←XQ+B
```

GIX is a vector holding the tabulated x_i values from [7 p920, n=4]. DT÷2 is the half-interval scale factor to convert the x_i to t'-t values. The statement for B is a copy of (37), letting execution order take care of the parentheses. AB is (38). A is (36). These are all four-element vectors.

```
[2]  M←XQ÷AB ◇ K←(AE1 MPOLY M)-(⍟M)×BE1 MPOLY M ◇ SQAB←AB*0.5
```

M is (40) again, and K and SQAB are as in REALGREEN.

```
[3]  E←(AE2 MPOLY M)-(⍟M)×BE2 MPOLY M ◇ Z←2×+/GIW×K÷SQAB×LGX
```

GIW is a vector holding w_i values and LGX is a vector holding $\log(x_i)$ values. '+/' does the summation so now Z holds the average of $G_{0\varepsilon}$ over a particular interval for which it is singular.

```
[4]  Z←Z,2×+/GIW×((A×K÷SQAB)-SQAB×E)÷B×LGX
```

Line [4] calculates the average for $G_{1\varepsilon}$ and catenates it with Z to form the new Z which the function passes out as the explicit result. The 2δ in (33) doesn't appear in this function because (33) is the integral, not the average. The average is the integral divided by 2δ.

The complex G functions, $G_{0\omega}$ and $G_{1\omega}$ (23,24), are initialized, collected and timed in a function WGENERATE which is nearly identical to SGENERATE. The results are held in arrays G0C and G1C. The actual integrations are done by the function CMPLXGREEN.

```
[0]  OX CMPLXGREEN SX;N;DA;ANG;R;CR;SR
```

```
[1]  N←15 ◇ DA←○÷N ◇ ANG←2○DA×¯0.5+⍳N
     ◇ R←(RHT[OX]×RHT[OX])∘.+RHT[SX]×RHT[SX]
```

The integration in ϕ is done by the piecewise-constant approximation. N is the number of intervals between 0 and π, and DA is the interval length. ANG is the vector of cosine values for the centers of these intervals. R, in this line, is a matrix of all the possible $\rho^2+\rho'^2$ values specified by the OX and SX index vectors.

```
[2]  R←R+(ZT[OX]∘.-ZT[SX])*2
     ◇ R←(((N,⍴R)⍴R)-2×ANG∘.×RHT[OX]∘.×RHT[SX])*0.5
```

The last statement in line [1] and these two statements are a three-step construction of all the possible values of the distance function required by the index vectors OX and SX and the steps in ϕ. Coming into line [2] R has all the $\rho^2+\rho'^2$ values, and the first statement adds all the $(z-z')^2$ values. Starting at the right of the second statement in line [2], the ()*0.5 will take the square root of all the elements sent to the outer parentheses. Just inside the right parenthesis $\rho\rho'$ is done as an outer product to form a matrix the same shape as the current value of R. This matrix is then outer-product multiplied by the $\cos\phi$ values in ANG to form a 3-axis array. The first axis corresponds to the steps in ϕ, the other two to the index values in OX and SX. This array is next multiplied by 2 to give the $2\rho\rho'\cos\phi$ part of the distance function. Now a copy of the present value of R needs to be generated for each step in ϕ so the subtraction part can be properly done. (N,ρR)ρR makes a 3-axis array out of the $\rho^2+\rho'^2+(z-z')^2$ values with N identical copies along the first axis.

```
[3]  G0C[;OX;SX]←DA×(+⌿CR←(¯1+2○BT×R)÷R),[0.5]+⌿SR←(1○BT×R)÷R

[4]  ANG←⍉((ΦρR)ρANG) ◇ G1C[:OX;SX]←DA×(+⌿ANG×CR),[0.5]+⌿ANG×SR
```

$e^{-j\beta R}$ is represented in rectangular form as cos(βR) (2○BT×R) and -sin(βR) (-1○BT×R). The imaginary and real parts are separately formed and summed down the first axis (the ϕ steps) and lastly multiplied by DA to complete the integration. The first statement in line [4] reshapes the $\cos\phi$ vector into a 3-axis array to match the other items in the integrand of $G_{1\omega}$.

The reader may feel that these are compact (a good thing) and formidable bits of code. They are to the developer too. They were developed by using vectors with a few elements each to test the operations and see that they worked out to the shapes needed. If an operation works for little vectors, it will work for any.

IVD Filling the Impedance Matrices

Following Simpson's path, the Z_{ik} matrix was split into circuit-element and radiation parts.

$$Z_{ik} = L_{ik}+\frac{S_{ik}}{\beta^2}+Z_{\omega ik} \tag{47}$$

The L and S matrices depend strictly on geometry with no frequency component, so that once they are computed for a given shape, they need not be recomputed. It actually costs more execution time to compute Z this way most of the time, but it was interesting to see the relative effects of the static and dynamic G functions.

$$L_{ik} = \int_{\tau_i-\Delta}^{\tau_i+\Delta}\int_{\tau_k-\Delta}^{\tau_k+\Delta} T_i T_k (G_{0E}\cos\theta\cos\theta' + G_{1E}\sin\theta\sin\theta')\,dt'dt \tag{48}$$

$$S_{ik} = -\int_{\tau_i-\Delta}^{\tau_i+\Delta}\int_{\tau_k-\Delta}^{\tau_k+\Delta} \frac{dT_i}{dt} G_{0E} \frac{dT_k}{dt'}\,dt'dt \tag{49}$$

$$Z_{\omega ik} = \int_{\tau_i-\Delta}^{\tau_i+\Delta}\int_{\tau_k-\Delta}^{\tau_k+\Delta} \left\{ T_i T_k (G_{0\omega}\cos\theta\cos\theta' + G_{1\omega}\sin\theta\sin\theta') - \frac{dT_i}{dt}\,\frac{G_{0\omega}}{\beta^2}\,\frac{dT_k}{dt'} \right\} dt'dt \tag{50}$$

Again, two kinds of functions were written for these matrices, one to initialize them, set up loops, and time the calculations, and one to do the actual double integration given the i,k values.

```
[0]  REACT;I;K;T1;NC
```

This function sets up the reactance matrices, L and S. They are called XGL and XGC respectively. The function also finds the terminal (circuit-element) reactance due to the static potentials.

```
[1]  T1←□TS ◇ I←1 ◇ XG←XGL←XGC←(NP,NC←¯1+2×NP)⍴0

[2]  IL:K←1
```

These lines initialize the starting time, loop counters, and matrices. NC is the 'N' summation limit in (46), which is the number of equations and unknowns without taking account of the geometric symmetry.

```
[3]  KL:XGL[I;K]←LIK I,K ◇ XGC[I;K]←SIK I,K
     ◇ →(NC≥K←K+1)/KL ◇ →(NP≥I←I+1)/IL
```

LIK and SIK are the functions that do the integrations.

```
[4]  XGL←((NP,NP)↑XGL)+⍤0,(NP,1-NP)↑XGL

[5]  XGC←((NP,NP)↑XGC)+⍤0,(NP,1-NP)↑XGC
     ◇ XG←XGL+XGC÷BS
```

Here is where the symmetry in t is used to reduce the matrices to NP by NP. Suppose NP=5. Then $J_6=J_4$, $J_7=J_3$, $J_8=J_2$ and $J_9=J_1$. This implies that the columns to the right of 5 can be reversed in order and added to those to the left of 5. Only NP rows are calculated, with this reduction in mind. NP,1-NP specifies NP rows and NP-1 columns from the right side of the matrix for the take (↑) function. The 0, on the left will put a column of zeros on that side of the NP by NP-1 present result. The reverse flips the matrix side-to-side so the column of zeros lines up with column NP and column NP+1 of the original matrix lines up with column NP-1. The last step in the reduction process is addition to the left NP columns of the original matrix. XG is the total reactance matrix.

```
[6]  JT←-((NP-1)ρ0),÷60×BT×DT×DT)⌹XG ◇ XTERM←÷○2×JT[NP]
```

JT is the vector of current density coefficients, J_{tk}. The V_i are NP-1 zeros, and a negative imaginary number, $-1\div(60\beta Dt^2)$. The Dt was put with V_i to reduce the multiplies. Since V_i is imaginary and XG is real, JT is imaginary, as it should be. $V_o=1$, so the terminal reactance is

$$X_{term} = \frac{1}{2\pi\rho\left(\frac{t_m}{2}\right)\frac{T_{NP}\left(\frac{t_m}{2}\right)}{\rho\left(\frac{t_m}{2}\right)}J_{tNP}}$$

$$= \frac{1}{2\pi J_{tNP}} \qquad (51)$$

The next two lines of the function print the reactance, the run time, and sound the scale.

The integration functions have the same format, so only the more interesting LIK is described.

```
[0]  Z←LIK IK;L;M

[1]  L←P+2×IK[1] ◇ M←P+IK[2]

[2]  Z←((TRT×SINT[L])+.×G1[L;M]+.×SINT[M]×TRT)+
     (TRT×COST[L])+.×G0[L;M]+.×COST[M]×TRT
```

P is a global vector whose elements are ¯1 0 1 2. This makes L and M the set of t and t′ indices needed for the four-point integration, as shown in Figure 4. The appropriate portions of the G0 and G1 matrices are called as 4x4 matrices. Each one is then the center matrix in a vector-matrix-vector multiply (inner product) to do the double sums. That's it.

Another pair of functions, IMP and ZIK, organize and compute the elements for Z_ω. The computation principle for the integration is the same as in LIK, but the complex numbers require a little more work in handling the shapes of the vectors and arrays. The reactance and impedance matrices are combined and the net terminal impedance computed in the function TERM.

IVE Current and Dissipation

Once the current density coefficients, J_{tk}, are known, the current at each sample point can be found by adding the appropriately-weighted triangle functions. The values of J_{tk} are held in the variable JTC, but only for half the dipole. To provide plots for the exercise of intuition, the current was extended for the whole dipole. All of the Δ segments have two triangles on them except the starting one at t=0 and the ending one at $t=t_m$. The total current is formed by generating a complex vector of coefficient values padded with 0 at the right end (t_m) to multiply by the first half of the triangle function, and another coefficient vector padded with 0 at the left end (t=0) to multiply by the second half of the triangle function, and then add the two.

```
ICX←(JTC,(⌽ 0 ¯1 ↓JTC),0 0)∘.×TRT[1 2]          (52)
```

Since JTC is a complex vector (2-row matrix) the outer product produces a two-plate, or 3-axis, result. The first column of the first plate is 0.25 times the real parts, the second column of the first plate is 0.75 times the real parts, and the second plate is the corresponding imaginary parts of the current coefficient times 0.25 and 0.75.

```
RC←(,ICX[1;;]),[0.5],ICX[2;;]                   (53)
```

After extending JTC, there are 2NP by 2 elements in each plate of ICX. Raveling each plate takes the plate row-by-row and makes one 4NP-element row, every other element is 0.25 times a coefficient and its neighbor to the right is 0.75 times that same coefficient. Thus, each pair in a row is the left half of a triangle function times a current coefficient. The next statement reuses ICX to multiply the current coefficients by the right half of the triangle function, with an extra zero put in at the beginning. In (55) ICX is reformatted as above and the two sets are added.

```
ICX←(0 0 ,JTC,⌽ 0 ¯1↓JTC)∘.×TRT[3 4]            (54)

RC←RC+(,ICX[1;;]),[0.5],ICX[2;;]                (55)
```

RC holds the surface-current density times the radius, because the complete basis function hasn't been used. To get the total current through a cross-section, just multiply RC by 2π.

```
IT←o2×RC                                                    (56)
```

One of the banes of electrically-small antennas is that loss resistance can be higher than radiation resistance. One of the benefits expected from a thick dipole is a lower loss resistance due to spreading the current out over a large surface. Assuming that which is desired, the dissipation should have little effect on the current distribution, so one can use the current from the lossless equation to calculate the dissipation. The total dissipated power is

$$P = 2\pi \int_0^{t_m} R_s J_t J_t^* \rho \, dt \qquad (57)$$

$$\text{with } R_s = \sqrt{\frac{\omega \mu_o}{2\sigma}} = \sqrt{\frac{\beta 60 \pi}{\sigma}} \qquad (58)$$

At this point in the programming, RC is $J_t\rho$, so magnitude squared of RC divided by RHT (ρ) is used instead of $J_t J_t^* \rho$.

```
P←o2×DT×((o60×BT÷SIGMA)*0.5)×+/(+⌿RC×RC)÷RHT               (59)
```

V Verification

One cannot build a system of APL functions, any more than one can do a long mathematical development, test it only at the end, and hope to live without ulcers. Verification begins by testing the operation of each function on simple cases that can be verified by inspection, by hand calculation, or by looking up results in tables. Also, limiting values for large or small arguments are sometimes useful checks. Both the Green's function matrices and the impedance matrices (before folding) must be symmetrical. In earlier stages of this project, more of the work was done by completely separate functions, and there were more loops in the matrix-generating lines. This allowed easier testing of things like the elliptic integrals and their argument. Once some confidence was developed in results from these simpler functions, these results were kept for comparison, in separate workspaces, against results from more sophisticated versions.

The final tests are overall performance tests. Since we are approximating a continuous current distribution by a finite set of sample values, and using approximate integrations everywhere, we cannot expect perfect accuracy. We have a right to expect that the terminal impedance values will behave smoothly as a function of the number of sample points, and converge to some result. The slow convergence of the impedance values was, at

first, daunting. Reading about the similar experience of others [14, 15] gave a little morale boost. Although quite a number of test shapes were tried, with varying success, two are offered because they are shape extremes and are represented in the literature. These are the very thin wire dipole and the sphere dipole, both operated at half wavelength.

The classical impedance for a thin halfwave dipole is 73+j42.5 Ω. To describe this shape to the functions, set RHD=1E⁻8 0, ZD=0.9999 1. There was no difference in the results for the radius two orders of magnitude larger. The triangle basis function automatically sets the current at the dipole end to zero, and the difference between the two ZD values is much less than any integration segment used, so the result is a numerical model for a very thin straight wire. The ZD values couldn't be identical because a divide-by-zero error would occur. Because of experience in control systems and broadband rf electronics, convergence was originally tested by doubling NP from one trial to the next, later switching to a 2-5-10 sequence. The largest number which could be run without overfilling the workspace was NP=150. This corresponds to 600 sample points. Finally, NP=5 10 20 50 100 was settled on as the test values. The admittance values were then fitted by polynomials in 1/NP. This course was taken because a polynomial in 1/NP goes to a constant value as NP→∞. Thus, if the solution values do converge, the polynomial-fit constants might be the right values. Table 1 shows the data and results for the thin dipole.

Table 1. Admittances for the Thin Halfwave Dipole, mS

NP	5	10	20	50	100	∞
G	11.05	10.13	9.823	9.709	9.7	9.713
B	-4.548	-5.071	-5.278	-5.42	-5.486	-5.568

$G = 9.713-2.501NP^{-1}+133NP^{-2}-889.2NP^{-3}+2270NP^{-4}$

$B = -5.568+9.077NP^{-1}-99.66NP^{-2}+772NP^{-3}-1866NP^{-4}$

$Z_{\infty} = 77.49 + j44.42\ \Omega$

The spherical dipole profile has to be approximated by a sequence of straight-line segments. A five-segment model with equal subtended angles was chosen for the half-sphere so that the longest Δ value would fit between the corners. The generating statements are ZD←0,1○0.1×○5 and RHD←1,2○0.1×○5. The results for the same trial values and curve fitting are given in Table 2.

Table 2. Admittances for the Halfwave Sphere Dipole, mS

NP	5	10	20	50	100	∞
G	17.01	16.71	16.65	16.64	16.64	16.64
B	58.68	71.41	83.78	99.96	112.2	130.1

$G = 16.64-0.1826NP^{-1}+5.024NP^{-2}+47.95NP^{-3}-112.1NP^{-4}$

$B = 130.1-2119NP^{-1}+3.592E4NP^{-2}-2.764E5NP^{-3}+7.043E5NP^{-4}$

$Z_\infty = 0.9673 - j7.563\ \Omega$

The polynomial coefficients give an indication of the rate of convergence, if it exists. The sphere represents two extremes, a case where convergence has already occurred, and a case for which it is doubtful. NP has to go over 2000 to drive the variable terms below 1 mS for B. For the thin dipole, NP=200 drives the variable part of B below 1 % of the constant, and NP=25 will do the same for G.

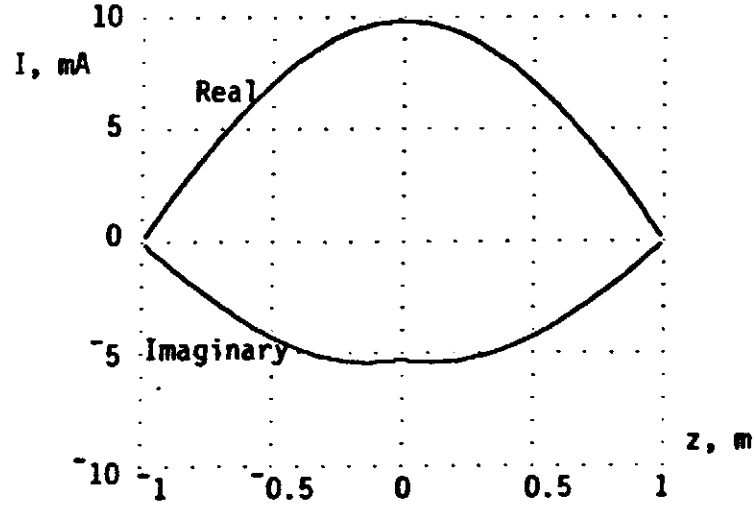

Figure 6. Cross-sectional current along the half-wave thin dipole.

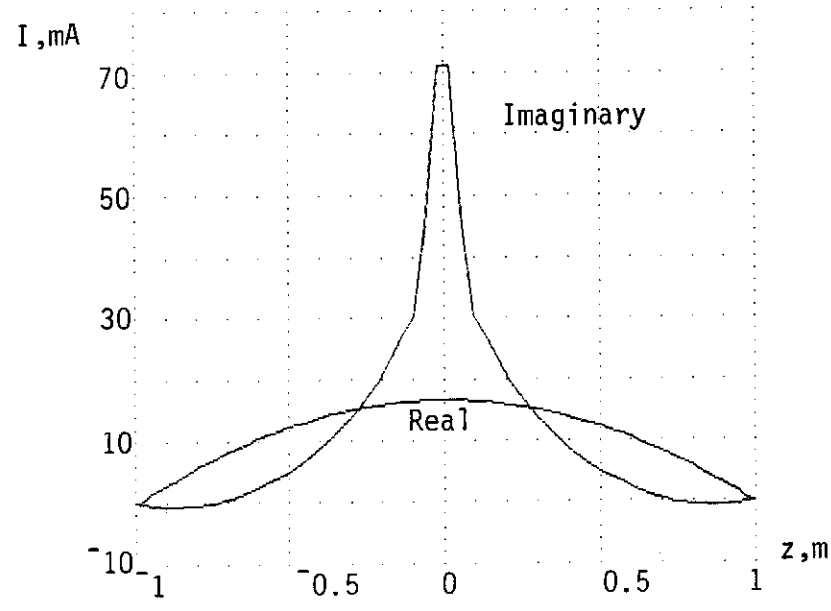

Figure 7. Cross-sectional current along the half-wave sphere.

Forging ahead, the current distributions were plotted for spheroids of various radii from 1 m on down. Figures 6 and 7 show the two extremes. The real parts show a sinusoidal shape as might be expected. The imaginary part of the thin dipole current is negative, necessary for the inductive terminal impedance, and looks sinusoidal except for a tiny upward cusp at the feedpoint. As the dipole is thickened, this cusp grows, the imaginary part of the current changes sign along the z axis, and finally becomes entirely positive. The full sphere shows a large leading current near the feed gap. Since this current is a growing function of the antenna radius, it is presumed that it is bounded as it looks, and the admittance values are reasonable, if not highly accurate.

According to Ramo and Whinnery, Stratton and Chu[11] used spherical-harmonic modes to match the boundary conditions for spheroidal antennas. Their results for the full sphere give a converging series for the conductance, but not for the susceptance. Figure 12.23d, p544, from [12] shows curves for these quantities as functions of βa, 'a' being the sphere radius. The susceptance value for $\beta a=\pi/2$ from this figure is about 27 mS, which was expected to be low, because they stopped arbitrarily at 19 terms. The conductance value is about 17 mS, which matches Table 2 above. A frequency response for βa between 0 and 3 was

VI. Impedance and Shape

For many years it seemed, from [12], that the sphere was a potential broadband antenna, but the recent reexamination of this material and the calculations described in section V show that the large gap perimeter adds too much capacitance to the antenna. At small a/λ the wire dipole has too much series body capacitance and the sphere has too much feed capacitance. A shape with more body than the wire and less gap than the sphere ought to do better than both. From Hansen's paper[3] quoting Chu[13], the minimum Q for an antenna with $\beta a<1$ is

$$Q_{min} = \frac{1+3\beta^2a^2}{\beta^3a^3[1+\beta^2a^2]} \qquad (60)$$

At 3 MHz, λ=100 m, and the antenna size limit is a=1 m. This means that Q_{min}=4063. For a simple tuned circuit this means a bandwidth of only about 750 Hz, a little tight even for SSB. This may seem a high value of Q but no reported wire antennas come anywhere near this low. Starting from the sphere, the gap was opened up by using a cone from a small gap as a transition section to a barrel and a capping cone, without good result. Then a biconical half-dipole was explored and showed that the Q improved as the cap cone was flattened. This led to a single cone flaring out from the feed to a plate cap. Is the straight-sided cone the best that can be done? Both convex and concave profiles were tried. Concave was better. Finally, the tube and cone was found to be the best of this shape type. The profile for the best tube-and-cone half-dipole is a tube of 10 cm radius from the feed gap to 0.75 m, then a cone flaring out to 1 m radius at the end, with a plate to close the figure. Table 3 gives descriptions and results for some of the trial shapes.

A basic conclusion one can draw from this work is that both gap and body capacitances need to be kept down. Even with the gap perimeter reduced going from the sphere to the barrel, there's a great deal of charging current. So why isn't a tube out to a plate the best shape? Possibly a little circuit tuning is going on. A frequency-response run for the T-C (tube and cone antenna), shown in Table 4, turned up a series resonance near 17.49 MHz.

Figures 9-12 show the current distributions along the T-C dipole. The real part is nearly flat along the tube below, at, and above resonance, while the imaginary part is closer to flat than anything else below and above the resonance. This is what one would expect from an end-loaded small dipole and supports the idea that the resonance is a circuit phenomenon. The sharp drop in the z components near the dipole ends, shown in Figure 10, are due to the abrupt change in the current's direction going from the tube to the wide-angle cone. The current shifts from axial to almost radially directed.

Table 3. Circuit Properties for Dipole Shapes

f=3 MHz, λ=100 m, dipole length=2 m, NP=20. ρ_d and z_d are the half-dipole profile description in meters. R_{loss} and Z_{in} are series equivalent circuit parameters. Q=|X|/R.

Shape Name	ρ_d/z_d				Z_{in} Ω	Q	R_{loss} mΩ Cu
Thin Spheroid 2 mm feed dia.					0.067-j11,290	168k	44.8
Full Sphere					0.0142-j267	19k	0.0128
Barrel	0.01	1	1	0	0.019-j233	12.1k	0.435
	0	0.2	0.9	1			
Double Cone	0.01	1	0		0.043-j504	11.7k	0.513
	0	0.5	1				
Cone and Cap	0.01	1	0		0.142-j673	4765	0.672
	0	0.99	1				
''	0.01	0.5	1	0	0.145-j679	4685	0.677
	0	0.5	0.999	1			
Concave Cone	0.01	0.5	1	0	0.188-j807	4293	0.895
	0	0.8	0.999	1			
Tube and Cone (T-C)	0.01	0.01	1	0	0.269-j925	3432	10.4
	0	0.75	0.999	1			

Table 4. Frequency Response of the T-C Dipole

f, MHz	3	6	10	15
Z_{in} Ω	0.27-j925	1.09-j422	3.1-j195	7.32-j52
f, MHz	17.49	20	30	
Z_{in} Ω	10.28+j0.073	14+j47	39+j215	

run, and Figure 8 shows the plot of conductance for βa between 0 and 2.5. The parameters were NP=20 and the half-sphere profile was divided into 20 equal segments. The curve has the same shape as that from [12]. That's nice, but why aren't all the conductance data plotted? Because there's a big wrinkle around βa=2.8 which would have obscured the shape of the lower-frequency curve. The sphere has a series resonance at about βa=2.752, λ=2.283, and a parallel resonance at about βa=2.882, λ=2.18 m. Tinkering with NP and segment number did not make the resonances go away so they appear to be real. Fishing for the series resonance λ was interesting because the susceptance behaved reasonably, but the conductance did strange things in magnitude and changed signs close to the resonance. Since the conductance should get large and the susceptance small at series resonance, perhaps the impedance matrix was becoming difficult to invert. The parallel resonance is smoother.

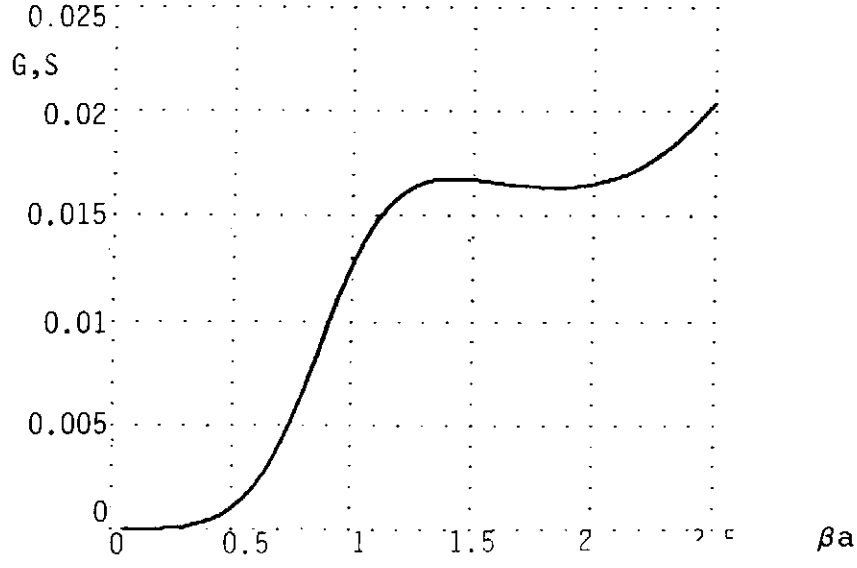

Figure 8. Conductance for a 40-segment sphere dipole.

Since agreement with two classical cases of very different shape is fair, it seems reasonable to gamble that the functions and numerical procedures are generally working and reliable. Ramo and Whinnery[12] said the series for the susceptance doesn't converge for the infinitesimal gap, so the mode analysis has the basic defect that it can't account for the circuit-element behavior of the sphere. This may be why the mode analysis doesn't show up the resonances.

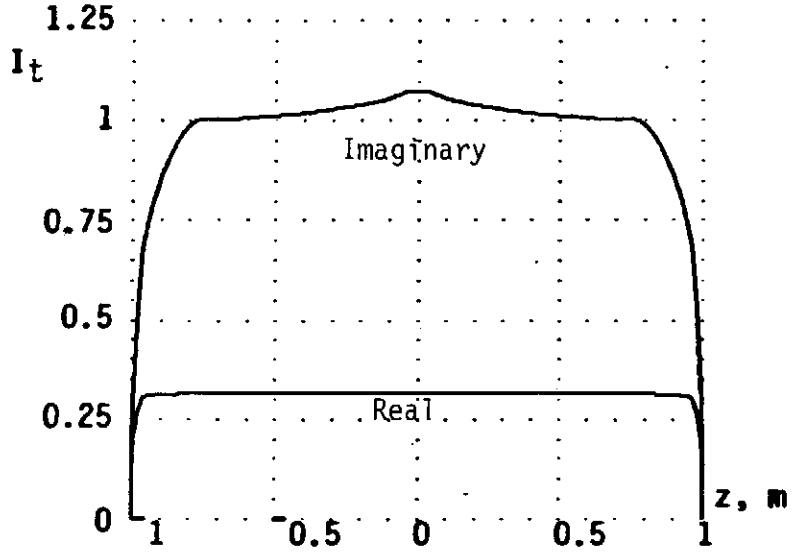

Figure 9. Current on the T-C dipole at 3 MHz. Real part in μA, imaginary part is mA.

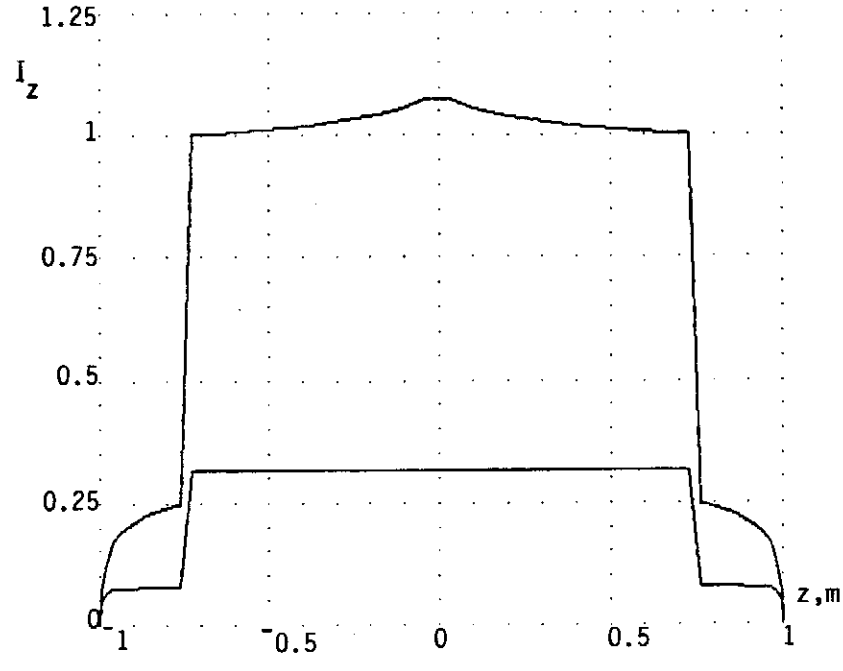

Figure 10. The z components for Figure 9.

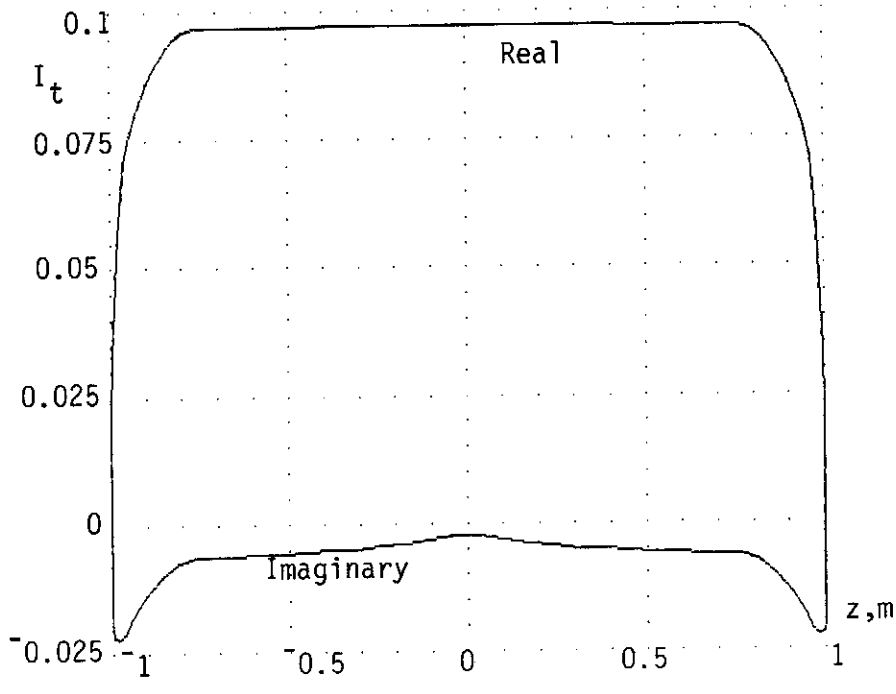

Figure 11. Current on the T-C dipole at 17.49 MHz. Real part in A, imaginary part in dA.

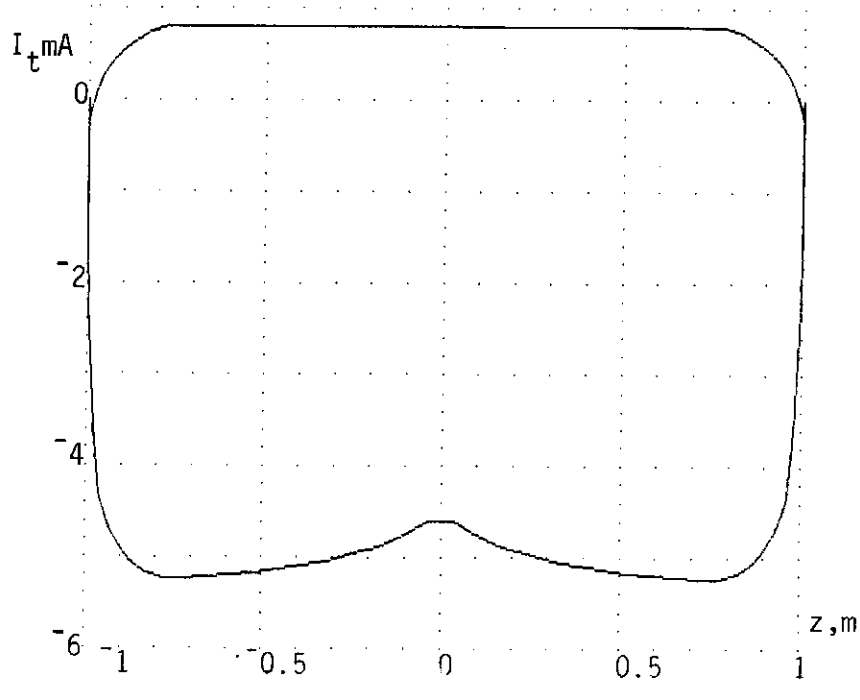

Figure 12. Current on the T-C dipole at 30 MHz.

VII. Development Time vs. Execution Time

This project was spread over a year and a half beginning in January, 1989. Too much time was spent in the first quarter of 1990 in sorting out hardware and software problems in a move to a 32-bit pc and a 32-bit APL interpreter. Writing this paper has taken three 7-day weeks. By comparison, the time spent in the code development seems only a brief pleasant memory. The reader has seen almost all the code written for this project in this paper. Think about that. Realize how closely the code mirrors the mathematics and how little overhead there is. In 1989, an 8 MHz 286/287 system with 2.7 Mb of RAM was being used for this project. A few times cases were run that took about 5 hours. With the move to the 20 MHz, 8 Mb, 32-bit system, the longest case took about an hour. The 100-point admittance calculation for Figure 8 took about 2 hours, during which markup on a student's thesis draft was started. Now this project, which is part of a larger one, is over. This code may never run again. From this perspective, even if the execution time of a compiled program was zero, it would not compensate for the excess time spent in coding and debugging the constituent functions in a scalar programming language. Human time is too precious.

Table 5 shows the execution times for various parts of a case computation. The blank entries are for cases that won't fit in 8 Mb. While these times are for a particular case, a simple shape and long wavelength, there is considerable variation from case to case. Shorter wavelengths seem to take longer, as much as 10 % on the dynamic calculations. This effect has not been investigated. The array sizes don't depend on wavelength, nor are there any loops which test on relative size for termination. The maximum problem that will fit with row looping is NP=150 (600 samples), and looping on ten-element bites out of a row allows the problem to run on a 640 kb machine with the 16-bit interpreter. The times for the G functions with ten-element-bite loops are double those for no loops on the same machine. One can see that the times generally follow an NP^2 law. The difference between total case time and time for a new frequency is a little deceiving, if one is considering whether to fill a separate static matrix or not. If the static and dynamic G functions are added and then used to fill only one impedance matrix, the time to fill that matrix would be only slightly longer than the times given for the dynamic matrix fill. In such a scheme, the time for a new frequency point in a response run would be the same, and the saving per frequency would be the time to compute the static G functions. The circuit-element reactance was calculated separately, for cases with new geometry or number of samples, for interest. The utility of this data is unclear, so it wasn't included in this paper.

Table 5. Execution Times

All times are in seconds. Cases were run on an ATronics 386B/20 (20 MHz clock) w. 64k cache card, 8 Mb RAM, Intel '387. The interpreter is STSC's APL*PLUS II, release 2.

Sample point, 4×NP.	20	40	80	200	400
Static G_0 and G_1 without loops.	1.16	3.4	11.3	63.5	
Static G_0 and G_1 with row loop.	1.49	4.12	13.5	66.8	249
Static matrix fill and invert.	1	4	16.2	103	426
Dynamic G_0 and G_1 without loops.	2.9	11.4	45		
Dynamic G_0 and G_1 with row loop.	3.1	12	47.4	284	1140
Dynamic matrix fill and invert.	1.4	5.1	22.2	157	789
Total, without G loops.	6.5	24	95		
Total, with G loops.	7	25	100	511	2704
Time for a new frequency.	4.5	17	70	441	1929

Appendix: APL Syntax, Symbols and Functions

This appendix will present only as much APL as is used in the paper. APL is fundamentally an interactive array-processing language. To be interactive, it must be an interpreter. It is economical of time in development and execution because it uses an extended symbol set from which over 80 functions and operators can be called by one or two keystrokes each. There are no data type statements, a variable is given its type by the way the user assigns data to it, and the interpreter keeps track of the type, not the user. Arithmetic is double-precision, or better. A variable may be a scalar, a vector, or an array of any number of axes (dimensions). Execution of a statement proceeds from right to left, except when interrupted by parentheses. A×B+C is not equal to C+A×B. Thus, a typical statement execution begins with the interpreter reading data (explicitly or in a variable name) at the right end and finishing either with a screen display or an assignment to a variable at the left end. Customarily, only uppercase letters are used in names for variables and user-defined functions, but the interpreter is case-sensitive so that lowercase letters can also be used. The ◇ is a statement separator, so that more than one statement can be placed on the same line. The intepreter passes from one statement to the next in the usual left-to-right manner, as if the succesive statements did have line numbers. Multiple-statement lines is a feature of STSC's APL*PLUS systems, but it is not standard. In the following listing, related symbols are grouped together to save space. S is a scalar, N is an integer, V is a vector, M is a matrix, f and g represent any built-in (primitive) function. Primitive scalar functions operate on a variable in an element-by-element manner, fM does f to each element in M and the result has the same shape as M. There are also array functions which rearrange arrays or extract blocks of data from them. Operators modify the way functions are applied to data.

APL Symbols	Meanings
+ - × ÷	Basic arithmetic functions.
*X X*Y ⍟X	e^X, X^Y, ln(X).
X←Y	X is assigned the shape and values of Y.
∇	Open or close (toggle) function-definition mode.
⍳N	A vector of integers from 1 to N.
→	Go to. →(test)/line_label. If 'test' is true control passes to the statement line beginning with 'line_label'.
LABEL:	A line label is identified by the colon.
○X	π times X.
N○X	N=1 is sinX, =2 is cosX, etc.
V←⍴X A←V⍴X	Shape is the number of elements along each axis of an array. ⍴X gives the shape of X, a vector. With a vector left argument the funtion ⍴ reshapes the data in X, taking it row by row, to the shape specified by the vector.
|X ×X	Magnitude of, sign of, each element in X.
V⌹M ⌹M	$M^{-1}V$, M^{-1}.
,X V1,V2	Make X a vector, make V1 and V2 a vector.
V1,[0.5]V2	Place V1 over V2 in a two-row matrix.
⍉ ⌽ ⊖	Matrix rearrange. Transpose, reverse left to right, reverse top to bottom.
V↑M V↓M	Take and Drop. V specifies how many rows and columns of M are affected.
f/M f⌿M	Reduction. Apply f between the elements of each row (column) of M.
M1f.gM2	Inner product. f/M1g⍉M2.
X1∘.fX2	Outer product. f is applied between each element of X1 and every element of X2. The shape equals the joining of the shapes of X1 and X2.

References

[1] Mautz, J. R. and R. F. Harrington, "Radiation and Scattering from Bodies of Revolution", Applied Scientific Research vol. 20, June 1969.

[2] Simpson, T. L, J. C. Logan, and J. W. Rockway, "Equivalent Circuits for Electrically Small Antennas Using LS-Decomposition with the Method of Moments", IEEE Trans. Ant. and Prop. vol. 37, no. 12, December 1989.

[3] Hansen, R. C., "Fundamental Limitations in Antennas", Proc. IEEE vol. 69, no. 2, February 1981.

[4] Miron, D. B. "Survey of Recent Results for the Singularity Problem in the Electromagnetic Integral Equation", Proc. North Dakota Acad. Sci., vol.38, p30, April 1984.

[5] Mahadevan, K. and H. A. Auda, "Electromagnetic Field of a Rectangular Patch of Uniform and Linear Distributions of Current", IEEE Trans. Ant. and Prop. vol.37, no. 12, December 1989

[6] Gradshteyn, I. S. and I. M. Ryzhik, Table of Integrals, Sums, and Products, Academic Press, New York, 1965, p153-154.

[7] Abramowitz, M. and I. A. Stegun, Handbook of Mathematical Functions, Dover, New York, 1972(?), p591-592.

[8] Pierce, B. O. and R. M. Foster, A Short Table of Integrals, 4th ed., p72, Ginn, New York, 1956.

[9] Neff, H. P. jr., Basic Electromagnetic Fields, 2nd ed., Harper and Row, 1987, p154-159.

[10] Miron, D. B., "The Conducting Cylinder Problem", an unpublished handout, available from the author.

[11] Stratton, J. A. and L. J. Chu, J. Appl. Physics, no. 12, pp230-248, March 1941.

[12] Ramo, S. and J. R. Whinnery, Fields and Waves in Modern Radio, 2nd ed., Wiley, New York, 1960.

[13] Chu, L. J. "Physical Limitations of Omnidirectional Antennas", J. Appl. Phys. no. 19, pp1163-1175, Dec. 1948.

[14] Sarkar, T. K., A. R. Djordjevic, and E. Arvas, "On the Choice of Expansion and Weighting Function in the Numerical Solution of Operator Equations", IEEE Trans. Ant. and Prop. vol. AP-33, no. 9, p988, September 1985.

[15] Petersen, A. F., "Difficulties Encountered When Attempting to Validate Thin-Wire Formulations for Linear Dipole Antennas", Applied Computational Electromagnetics Society Journal, Special Issue on Code Validation, p41, 1989.