IN 278

OPTIMUM SEEKING PROGRAMS FOR INCREASING THE RADAR CROSS SECTION FROM REACTIVELY LOADED SCATTERERS OVER A FREQUENCY BAND

by

Joseph R. Mautz

September 1972

ABSTRACT

This report gives two computer programs for increasing the radar cross section from reactively loaded scatterers over a frequency band. The programs are optimum seeking methods using objective functions which are a sum, weighted or unweighted, of reciprocal cross sections per wavelength squared at a number of frequencies within a frequency band. The first program is a univariate search procedure whereby one load is varied at a time to minimize the objective function. The second program uses a continued partan search procedure, which is a steepest descent method with acceleration steps. Both programs search for points of local minima, the end result depending upon the starting point. Descriptions of the theory, program instructions and listings, and sample input-output data are given for each procedure.

TABLE OF CONTENTS

PART ONE

A UNIVARIATE SEARCH PROCEDURE

1-1. Basic Theory

It is desired to obtain reactive loads to maximize the backscattering cross section per wavelength squared (σ/λ^2) over a frequency band. The scatterer is a surface S loaded at N ports, but otherwise perfectly conducting.

For unit plane wave excitation, the $\underset{\sim}{u}_r$ component of the field $\underset{\sim}{E}^s$ scattered by the loaded N port surface S is given by equations (75) and (67) of [1]

$$\underset{\sim}{E}^s \cdot \underset{\sim}{u}_r = \frac{-j\ \omega\mu\ e^{-jkr}}{4\pi r}\left[\underset{\sim}{F}_o^{sc} \cdot \underset{\sim}{u}_r - \tilde{I}^{sc}\left[Y_s + j\ B_L\right]^{-1}\vec{I}^{sc}\right] \tag{1}$$

where Y_s is the port admittance matrix, B_L is the diagonal matrix of port susceptances (B_{Li}), $\vec{I}^{sc}$ is the short circuit (all N ports short circuited) current at the ports induced by the unit plane wave, and $\underset{\sim}{F}_o^{sc} \cdot \underset{\sim}{u}_r$ is the normalized short circuit scattered field. Also, r is the distance from an origin in the vicinity of S to the point of observation of $\underset{\sim}{E}^s$. From (1), the back scattering cross section per wavelength squared can be written[2]

$$\frac{\sigma}{\lambda^2} = \frac{4\pi r^2}{\lambda^2}\left|\underset{\sim}{E}^s \cdot \underset{\sim}{u}_r\right|^2 = \frac{k^4\eta^2}{16\pi^3}\left|\underset{\sim}{F}_o^{sc} \cdot \underset{\sim}{u}_r - \tilde{I}^{sc}\left[Y_s + j\ B_L\right]^{-1}\vec{I}^{sc}\right|^2 \tag{2}$$

The port parameters Y_s, $\vec{I}^{sc}$, and $\underset{\sim}{F}_o^{sc} \cdot \underset{\sim}{u}_r$ are obtained using the programs listed in Part Two, Sections III, IV, and V of [1].

For a particular i,

$$\left[Y_s + j\ B_L\right] = \left[Y^i\right] + j\left[B_L^{\,i}\right] \tag{3}$$

where $\left[Y^i\right]$ is $\left[Y + j\ B_L\right]$ with $B_{Li} = 0$ and $\left[B_L^i\right]$ consists of just B_{Li}. The inverse of (3) is given by [3]

$$\left[Y_s + j\ B_L\right]^{-1} = \left[Y^i\right]^{-1} - \frac{j\ B_{Li}}{1 + j\ B_{Li}\left[Y^i\right]^{-1}_{ii}} \left[Y^i\right]^{-1} \left[U^i\right] \left[Y^i\right]^{-1} \quad (4)$$

where $\left[Y^i\right]^{-1}_{ii}$ is the ii^{th} element of $\left[Y^i\right]^{-1}$. All elements of $\left[U^i\right]$ are zero except the ii^{th} which is one. Substituting (4) into (2) and taking advantage of the symmetry of $\left[Y^i\right]^{-1}$,

$$\frac{\sigma}{\lambda^2} = \frac{k^4\eta^2}{16\pi^3} \left| \underset{\sim}{F}_o^{sc} \cdot \underset{\sim}{u}_r - \tilde{I}^{sc} \left[Y^i\right]^{-1} \vec{I}^{sc} + \frac{\left[\left[Y^i\right]^{-1} \vec{I}^{sc}\right]_i^2}{\left[Y^i\right]^{-1}_{ii} + j\ X_{Li}} \right|^2 \quad (5)$$

In (5), $\left[\left[Y^i\right]^{-1} \vec{I}^{sc}\right]_i^2$ is the square of the i^{th} element of the column $\left[Y^i\right] \vec{I}^{sc}$. Also,

$$X_{Li} = \frac{-1}{B_{Li}} \quad (6)$$

Since the reactance load X_{Li} is either an inductor or capacitor,

$$X_{Li} = a_i x_i \quad (7)$$

where

$$\begin{aligned} a_i &= \frac{k}{k_o} \quad , \ x_i \geq 0 \\ a_i &= \frac{k_o}{k} \quad , \ x_i < 0 \end{aligned} \quad (8)$$

The new variable x_i is the i^{th} port load reactance at propagation constant k_o.

Consider the function

$$f = \sum_{j=i}^{N_f} \frac{A_j}{\left(\frac{\sigma}{\lambda^2}\right)_j^L} \tag{9}$$

where A_j and L are positive constants and $\left(\frac{\sigma}{\lambda^2}\right)_j^L$ is $\frac{\sigma}{\lambda^2}$ evaluated at the jth frequency and raised to the Lth power. One can gradually decrease f by first minimizing f with respect to x_1, then with respect to x_2, next with respect to x_3, and so forth. After f is minimized with respect to the last x, f is minimized with respect to x_1 and the process continues. This general iteration process is called univariate search[4]. The process eventually converges because f can only decrease with each iteration and f has a lower bound. The process converges to a point at which a small change in any one of the x_i must increase f. There is no assurance that this point is the absolute minimum of f or even a relative minimum of f in the space of the variables x_i. Furthermore, convergence may be slow. Nevertheless, let us go into more detail concerning this process.

Substituting (7) into (5) and dropping the subscript i from x_i and a_i,

$$\frac{\sigma}{\lambda^2} = \frac{k^4 \eta^2}{16\pi^3} \left| E_1 + jE_2 \right|^2 \frac{(Z_1 + b_1)^2 + (Z_2 + b_2 + ax)^2}{Z_1{}^2 + (Z_2 + ax)^2} \tag{10}$$

where

$$E_1 + jE_2 = \underset{\sim}{E}{}_o^{sc} \cdot \underset{\sim}{u}_r - \tilde{I}^{sc} \left[Y^i\right]^{-1} \vec{I}^{sc} \tag{11}$$

$$b_1 + jb_2 = \frac{\left[\left[Y^i\right]^{-1} \vec{I}^{sc} \right]_i^2}{E_1 + jE_2} \tag{12}$$

$$Z_1 + jZ_2 = \left[Y^i\right]_{ii}^{-1} \tag{13}$$

Note that f of (9) with $\frac{\sigma}{\lambda^2}$ given by (10) is a continuous function of x but that $\frac{\partial f}{\partial x}$ is not continuous at x = 0.

If $x_1 < x_2 < x_3$ can be found such that

$$f(x_1) \geq f(x_2) \leq f(x_3) \tag{14}$$

is true for certain, then f(x) will have a relative minimum with respect to x in the interval (x_1, x_3). In (14), the subscripts on x denote different values of the load reactance referred to k_o at the ith port. The subscripts in (14) are not to be confused with the subscript i appearing in (8). Determine integers p and q such that

$$\begin{aligned} |f(x_o + 2^p \Delta) - f(x_o)| &\geq 10^{-4} f(x_o) \\ |f(x_o + 2^{p-1} \Delta) - f(x_o)| &< 10^{-4} f(x_o) \end{aligned} \tag{15}$$

$$\begin{aligned} |f(x_o - 2^q \Delta) - f(x_o)| &\geq 10^{-4} f(x_o) \\ |f(x_o - 2^{q-1} \Delta) - f(x_o)| &< 10^{-4} f(x_o) \end{aligned} \tag{16}$$

where x_o is the starting value of the argument x of f(x) and where Δ is a small portion ε_x of the magnitude of the input impedance (13) at the first frequency.

$$\Delta = \varepsilon_x |Z_1 + jZ_2| \tag{17}$$

It is assumed that the factor 10^{-4} in (15) and (16) is large enough so that round off error will not affect the signs of $f(x_o + 2^p \Delta) - f(x_o)$ and $f(x_o - 2^q \Delta) - f(x_o)$.

If

$$f(x_o + 2^p \Delta) > f(x_o)$$
$$f(x_o - 2^q \Delta) > f(x_o) \tag{18}$$

then (14) will be true for

$$x_1 = x_o - 2^q \Delta$$
$$x_2 = x_o \tag{19}$$
$$x_3 = x_o + 2^p \Delta$$

If (18) is not true because

$$f(x_o - 2^q \Delta) < f(x_o) \tag{20}$$

then one obtains either

$$f(x_o) > f(x_o - 2^q \Delta) < f(x_o - 2^{q+1} \Delta) \tag{21}$$

or

$$f(x_o - 2^{K-1} \Delta) \geq f(x_o - 2^K \Delta) < f(x_o - 2^{K+1} \Delta) \tag{22}$$

where K is an integer greater than q. Equation (14) may be obtained by setting x_1, x_2, and x_3 equal to the arguments appearing in whichever of (21) or (22) is true. If (18) is not true because

$$f(x_o + 2^p \Delta) < f(x_o) \tag{23}$$

then one obtains either

$$f(x_o) > f(x_o + 2^p \Delta) < f(x_o + 2^{p+1} \Delta) \tag{24}$$

or

$$f(x_o + 2^{K-1} \Delta) \geq f(x_o + 2^K \Delta) < f(x_o + 2^{K+1} \Delta) \tag{25}$$

where K is an integer greater than p. Equation (14) may be obtained by setting x_1, x_2, and x_3 equal to the arguments appearing in whichever of (24) or (25) is true. If $|x_o| > |Z_1 + jZ_2|$ at the first frequency, then the search for p and q is conducted in $\frac{-1}{x}$ with

$$\Delta = \frac{\varepsilon_x}{|Z_1 + jZ_2|} \tag{26}$$

If the search for K is being conducted in x and if $|x|$ becomes larger than $|Z_1 + j\,Z_2|$, then we switch to $\frac{-1}{x}$. Likewise if the search for K is being conducted in $\frac{-1}{x}$ and if $|\frac{-1}{x}|$ becomes larger than $\frac{1}{|Z_1 + jZ_2|}$ then we switch back to x.

Once $x_1 < x_2 < x_3$ is obtained such that (14) is true, a relative minimum of f(x) with respect to x is calculated as follows.

Step 1

If $f\left(\frac{x_2 + x_3}{2}\right) > f(x_2)$, then there is a relative minimum of f(x) in the interval $\left(x_1, \frac{x_2 + x_3}{2}\right)$ and we define

$$x_1' = x_1$$

$$x_2' = x_2 \tag{27}$$

$$x_3' = \frac{x_2 + x_3}{2}$$

But if $f\left(\frac{x_2 + x_3}{2}\right) \le f(x_2)$, then there is a relative minimum in the interval (x_2, x_3) and we define

$$
\begin{aligned}
x_1' &= x_2 \\
x_2' &= \frac{x_2 + x_3}{2} \\
x_3' &= x_3
\end{aligned}
\tag{28}
$$

Step 2

If $f\left(\dfrac{x_1' + x_2'}{2}\right) > f(x_2')$, then there is a relative minimum in the interval $\left(\dfrac{x_1' + x_2'}{2}, x_3'\right)$ and we define

$$
\begin{aligned}
x_1'' &= \frac{x_1' + x_2'}{2} \\
x_2'' &= x_2' \\
x_3'' &= x_3'
\end{aligned}
\tag{29}
$$

But if $f\left(\dfrac{x_1' + x_2'}{2}\right) \leq f(x_2')$, then there is a relative minimum in the interval (x_1', x_2') and we define

$$
\begin{aligned}
x_1'' &= x_1' \\
x_2'' &= \frac{x_1' + x_2'}{2} \\
x_3'' &= x_2'
\end{aligned}
\tag{30}
$$

Step 3

Let

$$
\begin{aligned}
x_1 &= x_1'' \\
x_2 &= x_2'' \\
x_3 &= x_3''
\end{aligned}
\tag{31}
$$

and go back to Step 1.

The procedure (Steps 1, 2, and 3) converges rapidly. For instance, consider the case in which $x_2 - x_1 = x_3 - x_2$. If $f\left(\frac{x_2 + x_3}{2}\right) > f(x_2)$, then after Steps 1 and 2,

$$\begin{aligned} x_3'' - x_1'' &= \frac{x_3 - x_1}{2} \\ x_2'' &= \frac{x_1'' + x_3''}{2} \end{aligned} \tag{32}$$

If $f\left(\frac{x_2 + x_3}{2}\right) \leq f(x_2)$, then step 1 is sufficient to halve the interval

$$\begin{aligned} x_3' - x_1' &= \frac{x_3 - x_1}{2} \\ x_2' &= \frac{x_1' + x_3'}{2} \end{aligned} \tag{33}$$

Thus n applications of the procedure (Steps 1, 2, and 3) are sufficient to reduce the interval to less than 2^{-n} of its original length.

The procedure (Steps 1, 2, and 3) is terminated when

$$|x_3 - x_1| < \varepsilon_x \, |Z_1 + jZ_2| \tag{34}$$

If the search is being conducted in $(\frac{-1}{x})$, then x_3 and x_1 are susceptances in which case (34) must be replaced by

$$|x_3 - x_1| < \frac{\varepsilon_x}{|Z_1 + jZ_2|} \tag{35}$$

The $Z_1 + jZ_2$ in (34) and (35) is evaluated at the first frequency. The procedure (Steps 1, 2, and 3) can be continued with apparent success until x_1 and x_3 are so close that they can no longer be distinguished from each other but it is doubtful whether any further accuracy is obtained beyond the point at which it becomes impossible to distinguish $f(x_1)$ from $f(x_3)$. The validity of the procedure (Steps 1, 2, and 3) does not depend upon the existence of any partial derivatives of f. The only requirement is that f be continuous.

The univariate search procedure is terminated when minimization of f with respect to the first load, the second load, and so forth on up to the last load decreases f by less than one part in 10^6.

1-2. Description of Univariate Search Program

This program computes the set of susceptive loads which renders (9) stationary. These loads are either inductors or capacitors. The susceptances printed by the program are referred to propagation constant k_o.

The activity on data sets 1 (punched card input) and 6 (direct access input-output) is as follows.

```
   READ (1, 10) N, N7, N9, NF1, L, NIV, BF, EP
10 FORMAT (6I3, 2E14.7)
   READ (1, 12) (BK (I), I = 1, NF1)
12 FORMAT (5E14.7)
   READ (1, 17) (NF (I), I = 1, NF1)
17 FORMAT (20I3)
   READ (1, 12) (A (I), I = 1, NF1)
   READ (1, 12) (BL (I), I = 1, N)
   REWIND 6
   SKIP N7 RECORDS ON DATA SET 6
   NZ = N*N
   NZ1 = 2*(NZ + N + 1)
   J1 = NZ1 * NF(NF1)
   READ(6) (ZZ (I), I = 1, J1)
```

There are N ports at which susceptive loads may be placed. The expression (9) is minimized a maximum of N9 times, each time with respect to one of the port loads. NF1 and L are respectively N_f and L appearing in (9). Although the short circuit formulation involving port parameters Y_s, $\vec{I}^{sc}$, and $\underset{\sim}{F}_o{}^{sc} \cdot \underset{\sim}{u}_r$ is emphasized, the dual open circuit formulation involving Z_s, $\vec{V}^{oc}$, and $\underset{\sim}{F}_o{}^{\tilde{oc}} \cdot \underset{\sim}{u}_r$ can also be obtained. The dual to (2) is

$$\frac{\sigma}{\lambda^2} = \frac{k^4\eta^2}{16\pi^3}\left| -\underset{\sim}{F}_o{}^{oc} \cdot \underset{\sim}{u}_r - \tilde{V}^{oc}\left[Z_s + jX_L\right]^{-1} \vec{V}^{oc}\right|^2 \tag{36}$$

Set NIV = 0 for the short circuit formulation and NIV = 1 for the open circuit formulation. BF is the reference propagation constant k_o appearing in (8) while EP is ϵ_x of (17) and (34). BK (J) is the propagation constant at which $\left(\frac{\sigma}{\lambda^2}\right)_J^L$ of (9) is evaluated. The port parameters Y_s, Z_s, $\vec{I}^{sc}$, $\underset{\sim}{F}_o{}^{sc} \cdot \underset{\sim}{u}_r$, $\vec{V}^{oc}$, and $\underset{\sim}{F}_o{}^{oc} \cdot \underset{\sim}{u}_r$ for propagation constant BK(J) are read into ZZ((NF(J) -1) * NZ1 + 1) through ZZ(NF(J) * NZ1). These port parameters have been stored on data set 6 by the program on pages 60 to 69 of [2]. The weighting factor A_J of (9) is read in via A(J). BL (I) is the starting value of the load susceptance at the Ith port referred to propagation constant k_o. The program modifies BL to improve backscattering and prints it out under the heading SUSCEPTANCE LOADS. For the open circuit formulation (NIV = 1), BL is the corresponding reactance.

The integer LA is defined early in the main program for use in a computed GO to statement later on.

LA = 1 if all $A_j = 1$ and $L = 1$

LA = 2 if at least one $A_j \neq 1$ and $L = 1$

LA = 3 if all $A_j = 1$ and $L \neq 1$

LA = 4 if at least one $A_j \neq 1$ and $L \neq 1$

DO loop 19 stores $\frac{k^2\eta}{4\pi^{3/2}}$ in C3 and $\frac{k}{k_o}$ in A1. DO loops 24 and 9 store the needed port parameters for propagation constant BK(J) in a contiguous group in ZZ((NF(J) - 1) * NZ1 + 1) through ZZ ((NF(J) - 1) * NZ1 + NZ + N + 1).

The iterative loop between statements 111 and 25 minimizes (9) with respect to the load susceptance at the J1th port. DO loop 26 stores the parameters appearing in (10) and uses then to evaluate (10) at propagation constant BK(NK). DO loop 27 stores Y_s of (2) in YP. DO loop 33 puts $\vec{I}^{sc}$ of (11) in YV. Upon exit from DO loop 28, Y^i of (11) will reside in YP. The branch to either statement 29 or 30 is necessary because the susceptive load is proportional to $\frac{k_o}{k}$ for negative susceptances and proportional to $\frac{k}{k_o}$ for positive susceptances. Statement 90 inverts Y^i of (11). DO loop 31 stores $\left[Y^i\right]^{-1} \vec{I}^{sc}$ of (11) in YI and $\tilde{I}^{sc}\left[Y^i\right]^{-1} \vec{I}^{sc}$ in U1. The parameters of (10) are stored as follows.

<u>Computer Program</u>	<u>Equation (10)</u>
E1	$\frac{k^4\eta^2}{16\pi^3}\,\lvert E_1 + jE_2\rvert^2$
E2	$\left(\frac{Z_1 + b_1}{a}\right)^2$
E3	$\frac{Z_2 + b_2}{a}$
E4	$\left(\frac{Z_1}{a}\right)^2$
E5	$\frac{Z_2}{a}$
X(1)	x
SIG	$\frac{\sigma}{\lambda^2}$

The variables E2, E3, E4, and E5 have subscript NK for $x \geq 0$ and subscript NK + NF1 for $x < 0$. The f of (9) is put in Y(1) by either DO loops 36 or 38. X(1) is actually x_o of (15) and Y(1) is $f(x_o)$.

The iterative loop between statements 84 and 108 realizes the preliminary search (16) on the negative side. The quantity $x_o - 2^{J-1} \Delta$ is put in X(J). The logic between statements 42 and 58 puts f(X(J)) in Y(J). If NIV = 0, X(J) is always a reactance throughout the program.

The iterative loop between statements 62 and 109 performs the preliminary search (15) on the positive side.

The iterative loop between statements 66 and 110 performs either the search (22) or the search (25), whichever is appropriate. If NIV = 0, JX = 1 indicates that, as in (22) or (25), the reactance is being incremented whereas JX = 0 indicates that the susceptance is being incremented.

When statement 113 is first reached, X1, X2, and X3 correspond to x_1, x_2, and x_3 of (14). The iterative loop between statements 113 and 112 carries out the procedure (Steps 1, 2, and 3) indicated by (27)-(31).

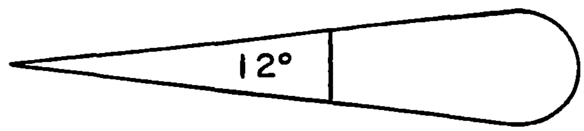
Minimum allocations are given by

```
COMPLEX C (N*N)
DIMENSION LR(N)
```

in the subroutine LINEQ and in the main program by

```
COMPLEX ZZ (2*(N*N + N + 1)*NF(NF1)), YP(N*N), YV(N), YI(N)
DIMENSION BK(NF1), NF(NF1), A(NF1), BL(N), C3(NF1), A1(NF1), E1(NF1), E2(2*NF1),
     E3(2*NF1), E4(2*NF1), E5(2*NF1), SIG(NF1)
```

The sample input data is for a wire angle-circle with a cross wire. The total angle is 12°, the



radius (measured from the center of the circle to the remote surface of the wire) of the circle is 0.055944 wavelengths (λ_o) at frequency f_o, and the radius of the wire is 0.055944 λ_o. Reactive loads are placed at

1. vertex of the angle (tip)
2. the circle end farthest from the vertex
3. the center of the cross wire

Backscattering is observed at f_o and 1.12 f_o in the direction of the vector from the center of the circle to the vertex of the angle. The output of the impedance matrix program (pages 45-52 of [1]) for this wire angle circle is as follows.

1-3. Output of Impedance Matrix Program

```
 NF N6 NP NW       RAD
  2  0 55  3 0.1985921E+00

PX
 -1.2543 -1.1498 -1.0453 -0.9408 -0.8362 -0.7317 -0.6272 -0.5226 -0.4181 -0.3136
 -0.2091 -0.1045  0.0000  0.1045  0.2091  0.3136  0.4181  0.5226  0.6272  0.7317
  0.8362  0.9408  1.0453  1.1498  1.2543  0.0000  0.6272  1.2543  1.3589  1.4634
  1.5679  1.6725  1.7770  1.6080  0.9486  0.0000 -0.9486 -1.6080 -1.7770 -1.6725
 -1.5679 -1.4634 -1.3589 -1.2543 -0.6272  0.0000 -1.0453 -1.1498 -1.2543 -0.6272
  0.0000  0.6272  1.2543  1.1498  1.0453

PY
  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
  0.0000  0.0000  0.0000  0.0000  0.0000

PZ
 11.9343 10.9397  9.9452  8.9507  7.9562  6.9617  5.9671  4.9726  3.9781  2.9836
  1.9890  0.9945  0.0000  0.9945  1.9890  2.9836  3.9781  4.9726  5.9671  6.9617
  7.9562  8.9507  9.9452 10.9397 11.9343 11.9343 11.9343 11.9343 12.9288 13.9233
 14.9178 15.9124 16.9069 17.8793 18.6137 18.8863 18.6137 17.8793 16.9069 15.9124
 14.9178 13.9233 12.9288 11.9343 11.9343 11.9343  9.9452 10.9397 11.9343 11.9343
 11.9343 11.9343 11.9343 10.9397  9.9452

LL
   1 26 47

BKK
 0.1769187E+00 0.1983635E+00

IMPEDANCE MATRIX OF ORDER 23
 0.2485E+01-0.5087E+03 0.2455E+01 0.2254E+03 0.2364E+01 0.4857E+02 0.2219E+01
 0.1074E+02 0.2026E+01 0.4544E+01

IMPEDANCE MATRIX OF ORDER 23
 0.3119E+01-0.4475E+03 0.3071E+01 0.2046E+03 0.2929E+01 0.4438E+02 0.2704E+01
 0.9971E+01 0.2409E+01 0.4119E+01
```

1-4. Listing of Univariate Search Program

```
//                 (0034,EE,20S,2),'MAUTZ,JOE',REGION=200K
// EXEC WATFIV
//GO.FT06F001 DD DSN=EE0034TM,UNIT=SYSDA,VOL=SER=SU0010,DISP=MOD,
// DCB=(RECFM=VS,LRECL=13000,BLKSIZE=13008),SPACE=(TRK,(20,10),RLSE)
//GO.SYSIN DD *
$JOB               MAUTZ,TIME=1,PAGES=30
      SUBROUTINE LINEQ(LL,C)
      COMPLEX C(100),STOR,STO,ST,S
      DIMENSION LR(40)
      DO 20 I=1,LL
      LR(I)=I
   20 CONTINUE
      M1=0
      DO 18 M=1,LL
      K=M
      DO 2 I=M,LL
      K1=M1+I
      K2=M1+K
      IF(CABS(C(K1))-CABS(C(K2))) 2,2,6
    6 K=I
    2 CONTINUE
      LS=LR(M)
      LR(M)=LR(K)
      LR(K)=LS
      K2=M1+K
      STOR=C(K2)
      J1=0
      DO 7 J=1,LL
      K1=J1+K
      K2=J1+M
      STO=C(K1)
      C(K1)=C(K2)
      C(K2)=STO/STOR
      J1=J1+LL
    7 CONTINUE
      K1=M1+M
      C(K1)=1./STOR
      DO 11 I=1,LL
      IF(I-M) 12,11,12
   12 K1=M1+I
      ST=C(K1)
      C(K1)=0.
      J1=0
      DO 10 J=1,LL
      K1=J1+I
      K2=J1+M
      C(K1)=C(K1)-C(K2)*ST
      J1=J1+LL
   10 CONTINUE
   11 CONTINUE
      M1=M1+LL
   18 CONTINUE
      J1=0
      DO 9 J=1,LL
      IF(J-LR(J)) 14,8,14
   14 LRJ=LR(J)
      J2=(LRJ-1)*LL
   21 DO 13 I=1,LL
```

```
      K2=J2+I
      K1=J1+I
      S=C(K2)
      C(K2)=C(K1)
      C(K1)=S
   13 CONTINUE
      LR(J)=LR(LRJ)
      LR(LRJ)=LRJ
      IF(J-LR(J)) 14,8,14
    8 J1=J1+LL
    9 CONTINUE
      RETURN
      END
      COMPLEX U,ZZ(3910),YP(100),YV(10),YI(10),U1
      DIMENSION BK(51),NF(51),A(51),BL(10),C3(51),A1(51),X(80),Y(80)
      DIMENSION E1(51),E2(102),E3(102),E4(102),E5(102),SIG(51)
      COMMON ZZ
      READ(1,10) N,N7,N9,NF1,L,NIV,BF,EP
   10 FORMAT(6I3,2E14.7)
      WRITE(3,11) N,N7,N9,NF1,L,NIV,BF,EP
   11 FORMAT('0  N N7 N9NF1  LNIV',7X,'BF',12X,'EP'/1X,6I3,2E14.7)
      READ(1,12)(BK(I),I=1,NF1)
   12 FORMAT(5E14.7)
      WRITE(3,13)(BK(I),I=1,NF1)
   13 FORMAT('0BK'/(1X,5E14.7))
      READ(1,17)(NF(I),I=1,NF1)
   17 FORMAT(20I3)
      WRITE(3,14)(NF(I),I=1,NF1)
   14 FORMAT('0NF'/(1X,20I3))
      READ(1,12)(A(I),I=1,NF1)
      WRITE(3,16)(A(I),I=1,NF1)
   16 FORMAT('0A'/(1X,5E14.7))
      READ(1,12)(BL(I),I=1,N)
      WRITE(3,15)(BL(I),I=1,N)
   15 FORMAT('0BL'/(1X,5E14.7))
      LA=1
      S1=0.
      DO 18 J=1,NF1
      S1=S1+ABS(1.-A(J))
   18 CONTINUE
      IF(S1.GT.1.E-6) LA=LA+1
      IF(L.NE.1) LA=LA+2
      ETA=376.730
      PI=3.141593
      U=(0.,1.)
      C1=ETA/(4.*SQRT(PI*PI*PI))
      NZ=N*N
      NZ1=2*(NZ+N+1)
      N1=N+1
      REWIND 6
      IF(N7) 20,20,21
   21 DO 22 J=1,N7
      READ(6)
   22 CONTINUE
   20 J1=NZ1*NF(NF1)
      READ(6)(ZZ(I),I=1,J1)
      WRITE(3,23)(ZZ(I),I=1,J1)
   23 FORMAT('0ZZ'/(1X,5E14.7))
      DO 19 J=1,NF1
      C3(J)=C1*BK(J)*BK(J)
```

```
      A1(J)=BK(J)/BF
      J3=(NF(J)-1)*NZ1
      J5=J3+NIV*NZ
      DO 24 I=1,NZ
      J3=J3+1
      J5=J5+1
      ZZ(J3)=ZZ(J5)
   24 CONTINUE
      J5=J5+NZ+(N1-NZ)*NIV
      DO 9 I=1,N1
      J3=J3+1
      J5=J5+1
      ZZ(J3)=ZZ(J5)
    9 CONTINUE
      IF(NIV.NE.0) ZZ(J3)=-ZZ(J3)
   19 CONTINUE
      Y4=1.E+20
      K=1
  111 J1=K-(K-1)/N*N
      X(1)=BL(J1)
      IF(X(1).EQ.0.) X(1)=.1E-5
      X(1)=-1./X(1)
      Y(1)=0.
      BL(J1)=0.
      J6=0
      IF(X(1).LT.0.) J6=NF1
      DO 26 NK=1,NF1
      J2=NZ1*(NF(NK)-1)
      DO 27 J=1,NZ
      J3=J2+J
       P(J)=ZZ(J3)
   27 CONTINUE
      DO 33 J=1,N
      J4=J3+J
      YV(J)=ZZ(J4)
   33 CONTINUE
      J5=J4+1
      J3=-N
      DO 28 J=1,N
      J3=J3+N1
      IF(BL(J)) 29,29,30
   29 YP(J3)=YP(J3)+U*BL(J)/A1(NK)
      GO TO 28
   30 YP(J3)=YP(J3)+U*BL(J)*A1(NK)
   28 CONTINUE
   90 CALL LINEQ(N,YP)
      U1=0.
      DO 31 J=1,N
      YI(J)=0.
      J3=(J-1)*N
      DO 32 I=1,N
      J4=J3+I
      YI(J)=YI(J)+YP(J4)*YV(I)
   32 CONTINUE
      U1=U1+YI(J)*YV(J)
   31 CONTINUE
      U1=ZZ(J5)-U1
      S1=CABS(U1)*C3(NK)
      E1(NK)=S1*S1
      U1=YI(J1)*YI(J1)/U1
```

```
      J4=(J1-1)*N+J1
      Z1=REAL(YP(J4))
      Z1B=Z1+REAL(U1)
      Z2=AIMAG(YP(J4))
      Z2B=Z2+AIMAG(U1)
      J7=NF1+NK
      S2=Z1B/A1(NK)
      S3=Z1B*A1(NK)
      E2(NK)=S2*S2
      E2(J7)=S3*S3
      E3(NK)=Z2B/A1(NK)
      E3(J7)=Z2B*A1(NK)
      S1=Z1/A1(NK)
      S2=Z1*A1(NK)
      E4(NK)=S1*S1
      E4(J7)=S2*S2
      E5(NK)=Z2/A1(NK)
      E5(J7)=Z2*A1(NK)
      IF(NK.NE.1) GO TO 60
      SZ=CABS(YP(J4))
      SY=1./SZ
      DSY=1.E-8*SY
      DELX=EP*SZ
      DELB=EP/SZ
      SZ2=SZ*SZ
   60 J7=NK+J6
      S3=X(1)+E3(J7)
      S4=X(1)+E5(J7)
      SIG(NK)=E1(NK)*(E2(J7)+S3*S3)/(E4(J7)+S4*S4)
      IF(K.GT.1) GO TO 26
      WRITE(3,93) C3(NK),E1(NK),E2(NK),E3(NK),E4(NK),E5(NK),SIG(NK)
   93 FORMAT('0C3,E1,E2,E3,E4,E5,SIG'/(1X,7E11.4))
   26 CONTINUE
      GO TO (34,34,35,35),LA
   34 DO 36 J=1,NF1
      Y(1)=Y(1)+A(J)/SIG(J)
   36 CONTINUE
      GO TO 37
   35 DO 38 J=1,NF1
      Y(1)=Y(1)+A(J)/SIG(J)**L
   38 CONTINUE
   37 WRITE(3,39) J1
   39 FORMAT('0MINIMIZATION OF Y=F(X) WITH RESPECT TO THE',I3,'TH LOAD')
      WRITE(3,40)(SIG(J),J=1,NF1),X(1),Y(1)
   40 FORMAT('0STARTING VALUES OF SIGMA/(LAMBDA)**2,X, AND Y'/(1X,5E14.7
     1))
      WRITE(3,41) SZ
   41 FORMAT('0SZ=',E11.4)
      EPY=Y(1)*1.E-4
      IF(ABS(X(1))-SZ)3,3,4
    3 XS=X(1)
      DX=-DELX
      JX=1
      GO TO 2
    4 XS=-1./X(1)
      DX=-DELX/SZ2
      JX=0
    2 J8=1
      DXP=-DX
      J=2
```

```
 84 J2=J
    XX=XS+DX
    X(J)=XX
    IF(JX)100,100,42
100 IF(XX.EQ.0.) XX=DSY
    X(J)=-1./XX
 42 J6=0
    IF(X(J).LT.0.) J6=NF1
    DO 44 I=1,NF1
    J7=I+J6
    S3=X(J)+E3(J7)
    S4=X(J)+E5(J7)
    SIG(I)=(E4(J7)+S4*S4)/(E1(I)*(E2(J7)+S3*S3))
 44 CONTINUE
    Y(J)=0.
    GO TO (47,48,49,50),LA
 47 DO 89 I=1,NF1
    Y(J)=Y(J)+SIG(I)
 89 CONTINUE
    GO TO (51,52,53,54,55),J8
 48 DO 56 I=1,NF1
    Y(J)=Y(J)+A(I)*SIG(I)
 56 CONTINUE
    GO TO (51,52,53,54,55),J8
 49 DO 57 I=1,NF1
    Y(J)=Y(J)+SIG(I)**L
 57 CONTINUE
    GO TO (51,52,53,54,55),J8
 50 DO 58 I=1,NF1
    Y(J)=Y(J)+A(I)*SIG(I)**L
 58 CONTINUE
    GO TO (51,52,53,54,55),J8
 51 IF(ABS(Y(J)-Y(1)).GE.EPY) GO TO 59
    DX=DX*2.
    J=J+1
    IF(J-30) 84,84,108
108 STOP
 59 WRITE(3,85)(X(I),I=1,J2)
 85 FORMAT('0PRELIMINARY SEARCH ON NEGATIVE SIDE'/(1X,5E14.7))
    WRITE(3,86)(Y(I),I=1,J2)
 86 FORMAT(1X,5E14.7)
    X(2)=X(J2)
    Y(2)=Y(J2)
    IF(Y(2).LT.Y(1)) GO TO 61
    DX=DXP
    J8=2
    J=3
 62 J2=J
    XX=XS+DX
    X(J)=XX
    IF(JX)101,101,42
101 IF(XX.EQ.0.) XX=DSY
    X(J)=-1./XX
    GO TO 42
 52 IF(ABS(Y(J)-Y(1)).GE.EPY) GO TO 63
    DX=DX*2.
    J=J+1
    IF(J-32) 62,62,109
109 STOP
 63 WRITE(3,87)(X(I),I=1,J2)
```

```
 87 FORMAT('OPRELIMINARY SEARCH ON POSITIVE SIDE'/(1X,5E14.7))
    WRITE(3,86)(Y(I),I=1,J2)
    X(3)=X(J2)
    Y(3)=Y(J2)
    IF(Y(3)-Y(1)) 64,65,65
 65 X1=X(2)
    Y1=Y(2)
    X2=X(1)
    Y2=Y(1)
    X3=X(3)
    Y3=Y(3)
    J=3
    GO TO 71
 64 X(2)=X(3)
    Y(2)=Y(3)
 61 DX=2.*DX
    J8=3
    J=3
 66 K3=J
    XX=XS+DX
    IF(JX)102,102,103
102 IF(XX.EQ.0.) XX=DSY
    X(J)=-1./XX
    IF(ABS(XX).LE.SY) GO TO 42
    JX=1
    XS=-1./XX
    DX=DX*SZ2
    GO TO 42
103 X(J)=XX
    IF(ABS(XX).LE.SZ) GO TO 42
    JX=0
    XS=-1./XX
    DX=DX/SZ2
    GO TO 42
 53 IF((Y(J)-Y(J-1)).GT.0.) GO TO 67
    DX=DX*2.
    J=J+1
    IF(J-40) 66,66,110
110 STOP
 67 K1=K3-2
    K2=K3-1
    X2=X(K2)
    Y2=Y(K2)
    J=K3
    IF(DX) 68,68,69
 68 X1=X(K3)
    X3=X(K1)
    Y1=Y(K3)
    Y3=Y(K1)
    GO TO 71
 69 X1=X(K1)
    X3=X(K3)
    Y1=Y(K1)
    Y3=Y(K3)
 71 DX=DELX
    IF(JX.NE.0) GO TO 104
    X1=-1./X1
    X2=-1./X2
    X3=-1./X3
    DX=DELB
```

```
  104 KK=1
  113 J=J+1
      J8=4
      X4=.5*(X2+X3)
      X(J)=X4
      IF(JX)105,105,42
  105 IF(X4.EQ.0.) X4=DSY
      X(J)=-1./X4
      GO TO 42
   54 IF(Y(J)-Y2) 73,73,74
   73 X1=X2
      X2=X4
      Y1=Y2
      Y2=Y(J)
      GO TO 75
   74 X3=X4
      Y3=Y(J)
   75 IF((X3-X1).LT.DX) GO TO 79
      J=J+1
      J8=5
      X4=.5*(X1+X2)
      X(J)=X4
      IF(JX)106,106,42
  106 IF(X4.EQ.0.) X4=DSY
      X(J)=-1./X4
      GO TO 42
   55 IF(Y(J)-Y2)76,76,77
   76 X3=X2
      X2=X4
      Y3=Y2
      Y2=Y(J)
      GO TO 78
   77 X1=X4
      Y1=Y(J)
   78 IF((X3-X1).LT.DX) GO TO 79
      KK=KK+1
      IF(KK-40) 113,113,112
  112 STOP
   79 WRITE(3,88)(X(I),I=1,J)
   88 FORMAT('OFINAL SEARCH'/(1X,5E14.7))
      WRITE(3,86)(Y(I),I=1,J)
      DO 80 I=1,NF1
      SIG(I)=1./SIG(I)
   80 CONTINUE
      WRITE(3,81)(SIG(I),I=1,NF1),X(J),Y(J)
   81 FORMAT(' FINAL VALUES OF SIGMA/(LAMBDA)**2,X, AND Y'/(1X,5E14.7))
      BL(J1)=-1./X(J)
      IF(J1.NE.N) GO TO 25
      WRITE(3,83)(BL(I),I=1,N)
   83 FORMAT('OSUSCEPTANCE LOADS'/(1X,5E14.7))
      IF((Y4-Y(J)).LT.(1.E-6*Y(J))) GO TO 82
      Y4=Y(J)
   25 K=K+1
      IF(K-N9) 111,111,107
  107 STOP
   82 STOP
      END
$DATA
  3  0 10  2  1  0 0.1769187E+00 0.1000000E-04
 0.1769187E+00 0.1983635E+00
```

```
  1  2
 0.1000000E+01 0.1000000E+01
-0.2349120E-01-0.7859229E-03 0.6150758E+04
$STOP
/*
//
```

1-5. Printed Output

```
  N N7 N9NF1  LNIV         BF                EP
  3  0 10  2  1  0 0.1769187E+00 0.1000000E-04

BK
 0.1769187E+00 0.1983635E+00

NF
  1  2

A
 0.1000000E+01 0.1000000E+01

BL
-0.2349120E-01-0.7859229E-03 0.6150758E+04

ZZ
 0.7885438E-04 0.1205725E-01-0.4798896E-04-0.5610914E-03 0.6312272E-04
 0.4530929E-02-0.4799328E-04-0.5611032E-03 0.8134719E-04 0.4228095E-03
-0.4650035E-04-0.3118563E-02 0.6312469E-04 0.4530933E-02-0.4650054E-04
-0.3118567E-02 0.6286116E-04 0.1999587E-04 0.8659626E+00-0.8833485E+02
 0.1339624E+01-0.1283093E+03-0.1938630E+01-0.1553283E+01 0.1339585E+01
-0.1283092E+03 0.8471759E+01-0.1841436E+03-0.6670155E+01 0.3184797E+03
-0.1938527E+01-0.1552933E+01-0.6669990E+01 0.3184807E+03 0.7124145E+01
 0.4370879E+02 0.5777996E-02-0.3496083E-02-0.8552313E-02 0.4094768E-02
 0.8741899E-03-0.3979273E-02-0.4085985E-01 0.1629616E-01-0.2022418E+00
-0.5957566E+00-0.1502222E+01-0.1168442E+01 0.1083533E+01 0.2743393E+01
-0.1461533E-01 0.1548914E-01 0.1427788E-03 0.1512150E-01-0.9078211E-04
-0.7083474E-03 0.1215035E-03 0.5202800E-02-0.9077982E-04-0.7083609E-03
 0.1250480E-03 0.1086308E-02-0.9104886E-04-0.3229622E-02 0.1214975E-03
 0.5202807E-02-0.9104720E-04-0.3229620E-02 0.1243404E-03 0.1420562E-02
 0.8072982E+00-0.6316139E+02 0.2993755E+00-0.1122266E+03-0.1971375E+01
-0.2393962E+02 0.2994356E+00-0.1122267E+03 0.7541306E+01-0.3939557E+02
-0.4937274E+01 0.3205747E+03-0.1971402E+01-0.2393942E+02-0.4937516E+01
 0.3205747E+03 0.8658371E+01 0.1134583E+03 0.5126588E-02-0.4392274E-02
-0.6329987E-02 0.7698957E-02-0.1966548E-03-0.4919115E-02-0.1820137E-01
 0.3932767E-01-0.4714749E+00-0.3997546E+00-0.1342090E+01 0.3079779E+00
 0.1995680E+01 0.2246219E+01-0.5593035E-02 0.1680823E-01

C3,E1,E2,E3,E4,E5,SIG
 0.5294E+00 0.1600E-01 0.2052E+01-0.8891E+02 0.5974E+01-0.7777E+02 0.2762E-01

C3,E1,E2,E3,E4,E5,SIG
 0.6655E+00 0.2634E-01 0.1338E+01-0.6209E+02 0.1012E+01-0.6431E+02 0.2126E-01

MINIMIZATION OF Y=F(X) WITH RESPECT TO THE  1TH LOAD

STARTING VALUES OF SIGMA/(LAMBDA)**2,X, AND Y
 0.2761532E-01 0.2126265E-01 0.4256912E+02 0.8324260E+02

SZ= 0.7781E+02
```

```
PRELIMINARY SEARCH ON NEGATIVE SIDE
 0.4256912E+02 0.4256834E+02 0.4256757E+02 0.4256601E+02 0.4256290E+02
 0.4255667E+02 0.4254422E+02 0.4251932E+02 0.4246951E+02 0.4236992E+02
 0.4217072E+02 0.4177232E+02
 0.8324260E+02 0.8324260E+02 0.8324258E+02 0.8324258E+02 0.8324258E+02
 0.8324258E+02 0.8324265E+02 0.8324269E+02 0.8324289E+02 0.8324348E+02
 0.8324570E+02 0.8325380E+02

PRELIMINARY SEARCH ON POSITIVE SIDE
 0.4256912E+02 0.4177232E+02 0.4256989E+02 0.4257066E+02 0.4257222E+02
 0.4257533E+02 0.4258156E+02 0.4259401E+02 0.4261891E+02 0.4266872E+02
 0.4276831E+02 0.4296751E+02 0.4336591E+02
 0.8324260E+02 0.8325380E+02 0.8324260E+02 0.8324255E+02 0.8324258E+02
 0.8324258E+02 0.8324257E+02 0.8324257E+02 0.8324257E+02 0.8324263E+02
 0.8324303E+02 0.8324487E+02 0.8325308E+02

FINAL SEARCH
 0.4256912E+02 0.4177232E+02 0.4336591E+02 0.4296751E+02 0.4217072E+02
 0.4276831E+02 0.4236992E+02 0.4266872E+02 0.4246951E+02 0.4261891E+02
 0.4259401E+02 0.4260646E+02 0.4260023E+02 0.4260335E+02 0.4260179E+02
 0.4260257E+02 0.4260217E+02 0.4260295E+02 0.4260275E+02
 0.8324260E+02 0.8325380E+02 0.8325308E+02 0.8324487E+02 0.8324570E+02
 0.8324303E+02 0.8324348E+02 0.8324263E+02 0.8324289E+02 0.8324257E+02
 0.8324257E+02 0.8324255E+02 0.8324255E+02 0.8324255E+02 0.8324255E+02
 0.8324255E+02 0.8324258E+02 0.8324255E+02 0.8324255E+02
FINAL VALUES OF SIGMA/(LAMBDA)**2,X, AND Y
 0.2762779E-01 0.2125528E-01 0.4260275E+02 0.8324255E+02

PLUS THE PRINTOUT FOR 8 MORE ONE VARIABLE SEARCHES.
THE LAST FOUR LINES OF PRINTOUT ARE

FINAL VALUES OF SIGMA/(LAMBDA)**2,X, AND Y
 0.2763020E-01 0.2125371E-01-0.8893920E-04 0.8324289E+02

SUSCEPTANCE LOADS
-0.2345619E-01-0.7859876E-03 0.1124363E+05
```

1-6. Discussion

Convergence of the univariate search procedure is slow for this problem. Depending upon the starting values of the susceptances, more than 100 one variable searches may be required. Because the sample starting values of the susceptances are chosen very close to a stationary point of f of (9), execution terminated after a few one variable searches and a few pages of sample printout. Before running this program with arbitrary starting values of susceptance loads, the user is advised to remove as much intermediate printout as possible. Although not included in the sample output, the dual open circuit formulation was run for equivalent starting reactances. Convergence was to values of $\frac{\sigma}{\lambda^2}$ less than 0.05% from those of the short circuit formulation.

1-7. References

[1] R. F. Harrington and J. R. Mautz, "Modal Analysis of Loaded N-Port Scatterers," Report AFCRL-72-0179, Contract No. F19628-68-C-0180 between Syracuse University and Air Force Cambridge Research Laboratories, March 1972.

[2] R. F. Harrington, "Field Computation by Moment Methods," Macmillan, 1968, page 76.

[3] D. K. Faddeev and V. N. Faddeeva, "Computational Methods of Linear Algebra," Freeman, 1963, page 174.

[4] D. A. Pierre, "Optimization Theory with Applications," Wiley, 1969, page 292.

PART TWO

METHOD OF CONTINUED PARTAN

2-1. Basic Theory

The method of continued partan is described in reference [1]. We here apply it to the problem of optimizing backscattering over a frequency band. The basic formula for radar cross section is [2].

$$\frac{\sigma}{\lambda^2} = \frac{k^4\eta^2}{16\pi^3} \left| \underset{\sim}{F}_o^{sc} \cdot \underset{\sim}{u}_r - \tilde{I}^{sc} V \right|^2 \tag{1}$$

where k is the propagation constant, η is the intrinsic impedance for empty space,

$$V = \left[Y_s + j\, B_L \right]^{-1} \vec{I}^{sc} \tag{2}$$

$\vec{I}^{sc}$ is the column of short circuit port currents, Y_s is the port admittance matrix, B_L is the diagonal matrix of susceptive loads $\{B_{Li}\}$, and $\underset{\sim}{F}_o^{sc}$ is the the normalized short circuit scattered field.

The function

$$f = \sum_{j=1}^{N_f} \frac{1}{\left(\frac{\sigma}{\lambda^2}\right)_j} \tag{3}$$

where $\left(\frac{\sigma}{\lambda^2}\right)_j$ is $\frac{\sigma}{\lambda^2}$ evaluated at the jth frequency will be successively decreased in the space u defined implicitly by

$$\left(\frac{B_{Li}}{a_i}\right) = \frac{B_o\, u_i}{1 - |u_i|} \tag{4}$$

where

$$a_i = \frac{k}{k_o}, \; B_{Li} \geq 0$$
$$a_i = \frac{k_o}{k}, \; B_{Li} < 0 \qquad (5)$$

In (4), $\frac{B_{Li}}{a_i}$ is the load susceptance at the ith port when $k = k_o$. From (4),

$$u_i = \frac{\left(\frac{B_{Li}}{a_i}\right)}{B_o + \left|\frac{B_{Li}}{a_i}\right|} \qquad (6)$$

When $\frac{B_{Li}}{a_i}$ is much larger than the judiciously chosen positive constant B_o, $|u_i|$ is nearly unity and relatively insensitive to $\frac{B_{Li}}{a_i}$. It is easier to optimize a function of u_i rather than a function of $\frac{B_{Li}}{a_i}$ because $-1 < u_i < 1$ whereas $-\infty < B_{Li} < \infty$. The transformation (4) which relates $\frac{B_{Li}}{a_i}$ to u_i is continuous and has a continuous derivative at $B_{Li} = 0$.

In u space, continued partan search proceeds in the direction $-\left(\frac{\partial f}{\partial u}\right)_{u=u^\circ}$ from the starting point u° to a relative minimum at u^1, then in the direction $-\left(\frac{\partial f}{\partial u}\right)_{u=u^1}$ from u^1 to a relative minimum at u^2. From u^2, an acceleration step proceeds in the direction $u^2 - u^0$ to a relative minimum at u^3. Thereafter, the search alternates between best step gradient searches and acceleration steps. Best step gradient searches are used to obtain u^1, u^2, u^4, u^6, u^8,· · · while acceleration steps are used to obtain u^3, u^5, u^7, u^9,· · ·. In this context, relative minimum means relative minimum of f along the line of search. The searches terminate at u^k for which

$$f(u^k) \geq f(u^{k-2}) \qquad (7)$$

Continued partan requires the minimization of $f(u^j + \alpha\, v)$ with respect to the scalar α where u^j is the initial value of u and v is the direction of travel in u space. So that α may be a positive number usually less than one, the magnitude of the element of v largest in magnitude is normalized to one and the sign of v is chosen so that

$$\left(\frac{df}{d\alpha}\right)_{\alpha=0} = \sum_i v_i \left(\frac{\partial f}{\partial u_i}\right)_{\alpha=0} < 0 \tag{8}$$

The derivative $\frac{df}{d\alpha}$ is evaluated for larger and larger $\alpha > 0$ until $\frac{df}{d\alpha} > 0$ at, let us say, α_p. Now $\frac{df}{d\alpha} < 0$ at the preceeding point α_{p-1} so that f must have a relative minimum with respect to α on the interval (α_{p-1}, α_p). If $\frac{df}{d\alpha} > 0$ at $\alpha = \frac{\alpha_{p-1} + \alpha_p}{2}$, then the relative minimum is confined to $\left(\alpha_{p-1}, \frac{\alpha_{p-1} + \alpha_p}{2}\right)$. On the other hand, if $\frac{df}{d\alpha} < 0$ at $\alpha = \frac{\alpha_{p-1} + \alpha_p}{2}$, then the relative minimum must be on $\left(\frac{\alpha_{p-1} + \alpha_p}{2}, \alpha_p\right)$. This interval halving procedure is used to obtain (or rather confine to an arbitrarily short interval) a relative minimum of f with respect to α. The interval halving process will work when $\frac{df}{d\alpha}$ is only piecewise continuous. However, if the preliminary search leading to α_{p-1} and α_p is too coarse, the process may skip over a relative minimum of f with respect to α.

If a point u^j is reached such that at least one of the components of u^j is either 0, -1, or +1 (-1 and +1 differ no more from each other than the points 0^- from 0^+), the gradient will not exist and therefore cannot be used to determine the direction of travel v. One can, however, take v in the direction of steepest descent which does exist.

$$v_i = -\frac{\partial f}{\partial u_i} \text{ for all i at which } u_i \neq 0, -1, \text{ or } 1 \tag{9}$$

For all other i, v_i is either

$$-\text{Max}\left[\left(\frac{\partial f}{\partial u_i}\right)_{u_i^-}, 0\right] \tag{10}$$

or

$$-\mathrm{Min}\left[0, \left(\frac{\partial f}{\partial u_i}\right)_{u_i^+}\right] \qquad (11)$$

whichever is larger in magnitude. In (9) - (11), the superscript j has been omitted from u_i, and v_i is not yet normalized. If $u_i = 0$, $u_i^- = 0^-$, and $u_i^+ = 0^+$. If $u_i = -1$ or $+1$, $u_i^- = 1$ and $u_i^+ = -1$.

The derivative $\frac{\partial f}{\partial u_i}$ is obtained from (1) and (3). Differentiating (1) with respect to B_{Li} and applying a formula[3] for the derivative of an inverse matrix,

$$\frac{\partial}{\partial B_{Li}}\left(\frac{\sigma}{\lambda^2}\right) = \frac{-k^4\eta^2}{8\pi^3}\,\mathrm{Imag}\left[(\underset{\sim}{F}_o{}^{sc} \cdot \underset{\sim}{u}_r - \tilde{I}^{sc}\,V)^*\,(V)_i^2\right] \qquad (12)$$

where $(V)_i^2$ is the square of the ith element of the column V. Differentiating (3) term by term and using (12) and (4),

$$\frac{\partial f}{\partial u_i} = \frac{\eta^2}{16\pi^3}\sum_{j=1}^{N_f}\frac{k_j^4}{\left(\frac{\sigma}{\lambda^2}\right)_j^2}\,\mathrm{Imag}\left[(\underset{\sim}{F}_o{}^{sc} \cdot \underset{\sim}{u}_r - \tilde{I}^{sc}\,V)^*\,(V)_i^2\right]\frac{2a_i\,B_o}{(1-|u_i|)^2} \qquad (13)$$

Equation (13) is indeterminant at $|u_i| = 1$. If $u_i = -1^+$ or 1^-, $|B_{Li}|$ is very large and the elements of the ith row and ith column of $[Y_L + j\,B_L]^{-1}$ are proportional to $\frac{1}{B_{Li}}$ while all the other elements $[Y_s + j\,B_L]$ are insensitive to B_{Li}. When $u_i = -1^+$ or 1^-, (13) becomes

$$\frac{\partial f}{\partial u_i} = \frac{\eta^2}{16\pi^3}\sum_{j=1}^{N_f}\left\{\frac{k_j^4\,B_{Li}^2}{\left(\frac{\sigma}{\lambda^2}\right)_j^2 B_o{}^2}\,\mathrm{Imag}\left[(\underset{\sim}{F}_o{}^{sc} \cdot \underset{\sim}{u}_r - \tilde{I}^{sc}\,V)^*\,(V_i)^2\right]\right\}\frac{2B_o}{a_i} \qquad (14)$$

In (14), the indeterminate form in the brackets $\{\ \}$ does not depend on B_{Li}

when $|B_{Li}|$ is large, and thus can be evaluated by setting B_{Li} equal to some number much, much larger than B_o. When $|B_{Li}|$ is very large, the ith row and ith column of $[Y_s + j\, B_L]$ have to be scaled in order to avoid an unwise choice of pivot elements in the matrix inversion routine.

2-2. Description of Continued Partan Search Program

The activity on data sets 1 (punched card input) and 6 (direct access input-output) is as follows.

```
      READ (1, 10) N, KF, N9, N7, NIV, BZ, BKK, EPS
   10 FORMAT (5I3, 3E14.7)
      READ (1, 12) (BK (I), I = 1, KF)
   12 FORMAT (5E14.7)
      READ (1, 12) (BL (I), I = 1, N)
      READ (1, 15) (MP (I), I = 1, KF)
   15 FORMAT (20I3)
      READ (1, 17) (DEL (I), I = 1, 31)
   17 FORMAT (7E11.4)
      NZ = N*N
      NZ1 = 2*(NZ + N + 1)
      J1 = NZ1*MP(KF)
      REWIND 6
      SKIP N7 RECORDS ON DATA SET 6
      READ (6) (PP (I), I = 1, J1)
```

There are N ports at which susceptive loads may be placed. N_f appearing in (3) is read in through KF. Execution will terminate either as soon as (7) is satisfied, or after a maximum of N9 one dimensional searches. NIV = 0 obtains the short circuit formulation involving port parameters Y_s, $\vec{I}^{sc}$, and $\underset{\sim}{F}_o^{sc} \cdot \underset{\sim}{u}_r$ while NIV = 1 obtains the open circuit formulation involving port parameters Z_s, $\vec{V}^{oc}$, and $\underset{\sim}{F}_o^{oc} \cdot \underset{\sim}{u}_r$. BZ is B_o appearing in (4). B_o should be positive and of the same order of magnitude as the input port susceptances when NIV = 0, but of the same order of magnitude as the input reactances when NIV = 1. BKK is the reference propagation constant k_o of (5), preferably somewhere in the band of the KF propagation constants k read in through BK. The subscript on BK corresponds to j in (3). The one dimensional search in α of (8) terminates when a point at which $\frac{df}{d\alpha}$ goes through zero is confined to an interval of length EPS in α. BL (I) is the starting value of $\frac{B_{LI}}{a_I}$ of (4). The port parameters Y_s, Z_s, $\vec{I}^{sc}$, $\underset{\sim}{F}_o^{sc} \cdot \underset{\sim}{u}_r$, $\vec{V}^{oc}$, and $\underset{\sim}{F}_o^{oc} \cdot \underset{\sim}{u}_r$ have been computed by the programs on pages 47-69 of [2]. These port parameters for propagation constant BK (I) are located in PP (MP (I) - 1)*NZ1 + 1) through PP (MP (I)*NZ1). DEL is the values of α at which f $(u^j + \alpha\ v)$ is evaluated in the preliminary search leading to α_{p-1} and α_p which appear shortly after (8).

DO loop 30 puts (6) with subscript i replaced by K in U(K). The logic between statements 52 and 21 stores $\frac{\partial f}{\partial u_I}$ of (13) in DF (I). DO loop 19 stores $\frac{B_o\ u_J}{1 - |u_J|}$ of (4) in BL (J). BLD (J) is the factor $\frac{2\ B_o}{(1 - |u_J|)^2}$ in (13). The index J of DO loop 21 corresponds to j in (13). DO loops 22 and 23 store $[Y_s + j\ B_L]$ of (2) in Y. If |BL (I)| is sufficiently large, the subroutine SCALL normalizes the Ith row and Ith column of Y by multiplying them by the appropriate BLS. Statement 9 inverts Y. DO loop 25 stores $(V)_I$ of (2) in V(I) and puts $\tilde{I}^{sc}$ V in E. $\left(\frac{\sigma}{\lambda^2}\right)_J$ is put in SIG (J). The index I of DO loop 28 corresponds to i in (13).

The logic between statements 95 and 31 is the same as a DO loop with index K running from 1 to N9, the N9 appearing in the statement following statement 31. Here, an explicit DO loop is not allowed in WATFIV because of transfers out to statement 52 and back in again to either statements 62 or 63. The index K obtains the Kth one dimensional search leading to u^K. The logic between statements 32 and 38 stores the direction v in U1 (not normalized). DO loop 203 sets U(L) = 0 if U(L) is within EPS of zero and sets U(L) = 1 if U(L) is within EPS of ±1. If none of the U's were within EPS of either 0 or ±1, the logic between statements 221 and 205 is skipped.

The logic between statements 221 and 205 overrides DF and U1. This logic is nearly the same as that between statements 52 and 21 except that for i such that $u_i = 0$ or 1, $\frac{\partial f}{\partial u_i}$ is calculated on both sides (0^- and 0^+ for $u_i = 0$ or 1^- and -1^+ for $u_i = 1$). For i such that $u_i \neq 0$ and $u_i \neq 1$, DO loop 213 puts $\frac{\partial f}{\partial u_i}$ in DF. For i such that $u_i = 0$, DO loop 213 puts $\left(\frac{\partial f}{\partial u_i}\right)_{u_i = 0^-}$ in DF and $\left(\frac{\partial f}{\partial u_i}\right)_{u_i = 0^+}$ in U1. If $u_i = 1$, DO loop 211 has multiplied $(V)_i$ by $\left(\frac{B_{Li}}{B_o}\right)$ in (14) so that DO loop 213 will automatically put $\left(\frac{\partial f}{\partial u_i}\right)_{u_i = 1^-}$ in DF and $\left(\frac{\partial f}{\partial u_i}\right)_{u_i = -1^+}$ in U1. DO loop 216 puts -v of (10) or (11) in DF. Upon exit from DO loop 222, v of (9), and (10) or (11) resides in U1.

DO loops 44 and 45 normalize the magnitude of the element of U1 largest in magnitude to one. DO loop 47 puts $\left(\frac{\partial f}{\partial \alpha}\right)_{\alpha = 0}$ of (8) in DA. The logic between statements 92 and 46 performs the preliminary search leading to α_{p-1} and α_p appearing shortly after (8). Values of α are stored in AL, values of f in F and values of $\frac{\partial f}{\partial \alpha}$ in DD. The logic between statements 94 and 72 performs the interval halving procedure. DO loop 79 stores $u°$ in U2. The statement following statement 79 stores f in F2. F2(K5 + 1) is f for the present K while F2 (K2 + 1) is f for K-2.

Minimum allocations are given by

```
COMPLEX C(N*N)
DIMENSION LR(N)
```

in the subroutine LINEQ, by

```
COMPLEX Y (N*N)
DIMENSION BLS (N), KU(N)
```

in the subroutine SCALL, and by

```
COMPLEX UB (KF*2), Y(N*N), PP ((MP (KF)*2*(N*N + N + 1)), V(N)
DIMENSION BK(KF), BL(N), MP(KF), C2(KF), BLD(N), DF(N), SIG(KF), U(N), U1(N),
     UZ(N), U2(3*N), JU(N + 1), DB2(2*KF), KU(N), BLA(N), BLS(N)
```

in the main program. There are 31 elements of DEL and a maximum of 30 interval halving iterations so that DD(61), AL(61), and F(61) are sufficient. The sample data is for the wire angle-circle used in the univariate search program description. As can be observed from the sample printout, continued partan converged to the same (within roundoff error) susceptive loads and values of $\frac{\sigma}{\lambda^2}$. Although not included in the sample printout, the open circuit formulation (NIV = 1) version of continued partan also converged to the same values of $\frac{\sigma}{\lambda^2}$, and, as expected gave equivalent reactive loads. While continued partan usually requires fewer one dimensional searches, each one dimensional search is more time consuming. In this case, it is doubtful whether continued partan is faster than univariate search.

```
      K2=J2+I
      K1=J1+I
      S=C(K2)
      C(K2)=C(K1)
      C(K1)=S
   13 CONTINUE
      LR(J)=LR(LRJ)
      LR(LRJ)=LRJ
      IF(J-LR(J)) 14,8,14
    8 J1=J1+LL
    9 CONTINUE
      RETURN
      END
      SUBROUTINE SCALL(Y,BLS,KU,K9,N)
      COMPLEX Y(100)
      DIMENSION BLS(10),KU(10)
      DO 2 J=1,K9
      J1=(KU(J)-1)*N
      J2=KU(J)
      DO 3 I=1,N
      J3=J1+I
      Y(J3)=Y(J3)*BLS(J)
      Y(J2)=Y(J2)*BLS(J)
      J2=J2+N
    3 CONTINUE
    2 CONTINUE
      RETURN
      END
      COMPLEX U3,UB(60),Y(100).PP(338),E,EC,V(10)
      COMPLEX CONJG
      DIMENSION BK(30),BL(10).MP(30),DEL(31),F2(3),C2(30),BLD(10)
      DIMENSION DF(10),SIG(30),U(10),U1(10),UZ(10),DD(100),AL(100)
      DIMENSION F(100),U2(30)
      DIMENSION U5(3),JU(11),DB2(60),KU(10),BLA(10),BLS(10)
      READ(1,10) N,KF,N9,N7,NIV,BZ,BKK,EPS
   10 FORMAT(5I3,3E14.7)
      WRITE(3,11) N,KF,N9,N7,NIV,BZ,BKK,EPS
   11 FORMAT('0   N KF N9 N7NIV',6X,'BZ',12X,'BKK',11X,'EPS'/1X,5I3,3E14.
     17)
      READ(1,12)(BK(I),I=1,KF)
   12 FORMAT(5E14.7)
      WRITE(3,13)(BK(I),I=1,KF)
   13 FORMAT('0BK'/(1X,5E14.7))
      READ(1,12)(BL(I),I=1,N)
      WRITE(3,14)(BL(I),I=1,N)
   14 FORMAT('0BL'/(1X,5E14.7))
      READ(1,15)(MP(I),I=1,KF)
   15 FORMAT(20I3)
      WRITE(3,16)(MP(I),I=1,KF)
   16 FORMAT('0MP'/(1X,20I3))
      READ(1,17)(DEL(I),I=1,31)
   17 FORMAT(7E11.4)
      WRITE(3,18)(DEL(I),I=1,31)
   18 FORMAT('0DEL'/(1X,7E11.4))
      NZ=N*N
      NZ1=2*(NZ+N+1)
      J1=NZ1*MP(KF)
      REWIND 6
      IF(N7) 81,81,82
   82 DO 322 J=1,N7
```

```
      READ(6)
 322  CONTINUE
  81  READ(6)(PP(I),I=1,J1)
      WRITE(3,325)(PP(I),I=1,J1)
 325  FORMAT('0PP'/(1X,5F14.7))
      ISTP=0
      F2(2)=1.E+25
      F2(3)=1.E+25
      ETA=376.730
      PI=3.141593
      C1=.25*ETA/SQRT(PI*PI*PI)
      U5(1)=-1.
      U5(2)=0.
      U5(3)=1.
      BZ2=BZ*2.
      BZH=BZ*100.
      NZ=N*N
      N1=N+1
      U3=(0.,1.)
      DO 24 J=1,KF
      DB2(J)=BK(J)/BKK
      UB(J)=DB2(J)*U3
      J1=J+KF
      UB(J1)=U3/DB2(J)
      DB2(J1)=1./DB2(J)
      C2(J)=C1*BK(J)*BK(J)
      J3=(MP(J)-1)*NZ1
      J5=J3+NIV*NZ
      DO 323 I=1,NZ
      J3=J3+1
      J5=J5+1
      PP(J3)=PP(J5)
 323  CONTINUE
      J5=J5+NZ+(N1-NZ)*NIV
      DO 324 I=1,N1
      J3=J3+1
      J5=J5+1
      PP(J3)=PP(J5)
 324  CONTINUE
      IF(NIV.NE.0) PP(J3)=-PP(J3)
  24  CONTINUE
      DO 30 K=1,N
      U(K)=BL(K)/(BZ+ABS(BL(K)))
  30  CONTINUE
 117  KK=1
  52  K9=0
      DO 19 J=1,N
      IF(U(J).GT.1.) U(J)=U(J)-2.
      IF(U(J).LT.-1.) U(J)=U(J)+2.
      S1=ABS(U(J))
  20  S4=1.-S1
      IF(S4.EQ.0.) S4=1.E-6
      BL(J)=BZ*U(J)/S4
      BLD(J)=BZ2/(S4*S4)
      BLA(J)=ABS(BL(J))
      IF(BLA(J).LE.BZH) GO TO 19
      K9=K9+1
      KU(K9)=J
      BLS(K9)=SQRT(BZH/BLA(J))
  19  CONTINUE
```

```
      SL=0.
      DO 27 J=1,N
      DF(J)=0.
   27 CONTINUE
      DO 21 J=1,KF
      J1=(MP(J)-1)*NZ1
      DO 22 I=1,NZ
      J2=J1+I
      Y(I)=PP(J2)
   22 CONTINUE
      J1=1
      DO 23 I=1,N
      J3=J
      IF(BL(I).LE.0.) J3=J3+KF
      Y(J1)=Y(J1)+UB(J3)*BL(I)
      J1=J1+N1
   23 CONTINUE
      IF(K9.EQ.0) GO TO 9
      CALL SCALL(Y,BLS,KU,K9,N)
    9 CALL LINEQ(N,Y)
      IF(K9.EQ.0) GO TO 8
      CALL SCALL(Y,BLS,KU,K9,N)
    8 E=0.
      DO 25 I=1,N
      V(I)=0.
      J4=(I-1)*N
      DO 26 JJ=1,N
      J5=J4+JJ
      J6=J2+JJ
      V(I)=V(I)+Y(J5)*PP(J6)
   26 CONTINUE
      J6=J2+I
      E=E+V(I)*PP(J6)
   25 CONTINUE
      E=(PP(J6+1)-E)*C2(J)
      EC=CONJG(E)
      SIG(J)=E*EC
      EC=EC*C2(J)/(SIG(J)*SIG(J))
      DO 28 I=1,N
      J2=J
      IF(BL(I).LE.0.) J2=J2+KF
      DF(I)=DF(I)+DB2(J2)*BLD(I)*AIMAG(EC*V(I)*V(I))
   28 CONTINUE
      SL=SL+1./SIG(J)
   21 CONTINUE
      GO TO (61,62,63),KK
   61 K=1
      WRITE(3,87)(SIG(J),J-1,KF)
   95 K4=K-1
      K2=K-K/3*3
      K3=K2*N
      IF(K4) 32,32,33
   32 DO 40 J=1,N
      U1(J)=-DF(J)
   40 CONTINUE
      GO TO 39
   33 IF(K-K/2*2) 41,42,41
   42 DO 43 J=1,N
      U1(J)=-DF(J)
   43 CONTINUE
```

```
      GO TO 39
   41 DO 38 J=1,N
      J1=J+K3
      U1(J)=U(J)-U2(J1)
   38 CONTINUE
   39 J9=0
      DO 203 L=1,N
      DO 204 L2=1,3
      IF(ABS(U(L)-U5(L2)).GT.EPS) GO TO 204
      U(L)=U5(L2)
      J9=J9+1
      JU(J9)=L
      IF(U(L).EQ.-1.) U(L)=1.
      GO TO 203
  204 CONTINUE
  203 CONTINUE
      IF(J9.EQ.0) GO TO 205
      WRITE(3,221)(DF(J),J=1,N)
  221 FORMAT(' DF'/(1X,7E11.4))
      K9=0
      DO 206 J=1,N
      S1=ABS(U(J))
      S4=1.-S1
      IF(S4.EQ.0.) S4=1.E-6
      BL(J)=BZ*U(J)/S4
      BLA(J)=ABS(BL(J))
      BLD(J)=BZ2/(S4*S4)
      IF(BLA(J).LE.BZH) GO TO 206
      K9=K9+1
      KU(K9)=J
      BLS(K9)=SQRT(BZH/BLA(J))
  206 CONTINUE
      DO 207 J=1,N
      DF(J)=0.
      U1(J)=0.
  207 CONTINUE
      DO 208 J=1,KF
      J1=(MP(J)-1)*NZ1
      DO 209 I=1,NZ
      J2=J1-I
      Y(I)=PP(J2)
  209 CONTINUE
      J1=1
      DO 210 I=1,N
      J3=J
      IF(BL(I).LE.0.) J3=J3+KF
      Y(J1)=Y(J1)+UB(J3)*BL(I)
      J1=J1+N1
  210 CONTINUE
      IF(K9.EQ.0) GO TO 7
      CALL SCALL(Y,BLS,KU,K9,N)
    7 CALL LINEQ(N,Y)
      IF(K9.EQ.0) GO TO 6
      CALL SCALL(Y,BLS,KU,K9,N)
    6 E=0.
      DO 211 I=1,N
      V(I)=0.
      J4=(I-1)*N
      DO 212 JJ=1,N
      J5=J4+JJ
```

```
      J6=J2+JJ
      V(I)=V(I)+Y(J5)*PP(J6)
  212 CONTINUE
      J6=J2+I
      E=E+V(I)*PP(J6)
      IF(U(I).EQ.1.) V(I)=V(I)*DB2(J)*BL(I)/BZ
  211 CONTINUE
      E=(PP(J6+1)-E)*C2(J)
      EC=CONJG(E)
      SIG(J)=E*EC
      EC=EC*C2(J)/(SIG(J)*SIG(J))
      J2=J+KF
      J1=1
      JU(J9+1)=N1
      DO 213 I=1,N
      S1=AIMAG(EC*V(I)*V(I))
      IF(JU(J1)-I) 220,219,220
  219 J1=J1+1
      S1=S1*BZ2
      DF(I)=DF(I)+DB2(J2)*S1
      U1(I)=U1(I)+DB2(J)*S1
      GO TO 213
  220 J3=J
      IF(BL(I).LE.0.) J3=J+KF
      DF(I)=DF(I)+DB2(J3)*BLD(I)*S1
  213 CONTINUE
  208 CONTINUE
      WRITE(3,87)(SIG(J),J=1,KF)
      WRITE(3,214)(DF(I),I=1,N)
  214 FORMAT(' GRADIENT ON NEGATIVE SIDE'/(1X,7E11.4))
      WRITE(3,215)(U1(I),I=1,N)
  215 FORMAT(' GRADIENT ON POSITIVE SIDE'/(1X,7E11.4))
      DO 216 J=1,J9
      J2=JU(J)
      IF(DF(J2).LT.0.) DF(J2)=0.
      IF(U1(J2).GT.0.) U1(J2)=0.
      IF(DF(J2)+U1(J2)) 218,218,216
  218 DF(J2)=U1(J2)
  216 CONTINUE
      DO 222 J=1,N
      U1(J)=-DF(J)
  222 CONTINUE
  205 S1=0.
      DO 44 J=1,N
      S2=ABS(U1(J))
      IF(S2.GT.S1) S1=S2
   44 CONTINUE
      DO 45 J=1,N
      U1(J)=U1(J)/S1
   45 CONTINUE
      DA=0.
      DO 47 J=1,N
      DA=DA+DF(J)*U1(J)
   47 CONTINUE
      IF(DA.LE.0.) GO TO 48
      DA=-DA
      DO 49 J=1,N
      U1(J)=-U1(J)
   49 CONTINUE
   48 DO 51 J=1,N
```

```
      UZ(J)=U(J)
   51 CONTINUE
      DD(1)=DA
      AL(1)=0.
      F(1)=SL
      KK=2
      L=1
   92 L1=L+1
      AL(L1)=DEL(L1)
      DO 50 J=1,N
      U(J)=UZ(J)+DEL(L1)*U1(J)
   50 CONTINUE
      GO TO 52
   62 DD(L1)=0.
      F(L1)=SL
      DO 70 J=1,N
      DD(L1)=DD(L1)+DF(J)*U1(J)
   70 CONTINUE
      IF(DD(L1).GT.0.) GO TO 71
   46 L=L+1
      IF(L-30) 92,92,93
   93 WRITE(3,83)(AL(J),J=1,L1)
   83 FORMAT(' ARGUMENT'/(1X,5E14.7))
      WRITE(3,84)(F(J),J=1,L1)
   84 FORMAT(' FUNCTION'/(1X,5E14.7))
      WRITE(3,85)(DD(J),J=1,L1)
   85 FORMAT(' DERIVATIVE'/(1X,5E14.7))
      WRITE(3,88)
   88 FORMAT('0PLACE 1')
      STOP
   71 A1=AL(L1-1)
      A2=AL(L1)
      KK=3
      L=1
   94 L1=L1+1
      AL(L1)=.5*(A1+A2)
      DO 73 J=1,N
      U(J)=UZ(J)+AL(L1)*U1(J)
   73 CONTINUE
      GO TO 52
   63 DD(L1)=0.
      DO 74 J=1,N
      DD(L1)=DD(L1)+DF(J)*U1(J)
   74 CONTINUE
      F(L1)=SL
      IF((A2-A1).LE.EPS) GO TO 78
      IF(DD(L1)) 75,75,76
   75 A1=AL(L1)
      GO TO 72
   76 A2=AL(L1)
   72 L=L+1
      IF(L-20) 94,94,78
   78 WRITE(3,202) K,(AL(J),J=1,L1)
  202 FORMAT('0',I3,' ARGUMENT'/(1X,5E14.7))
      WRITE(3,84)(F(J),J=1,L1)
      WRITE(3,85)(DD(J),J=1,L1)
      WRITE(3,200)(U(J),J=1,N)
  200 FORMAT(' U COORDINATE'/(1X,7E11.4))
      WRITE(3,201)(U1(J),J=1,N)
  201 FORMAT(' U DIRECTION'/(1X,7E11.4))
```

```
      IF(L.NE.20) GO TO 89
      WRITE(3,90)
   90 FORMAT('OPLACE 2')
      STOP
   89 WRITE(3,86)(BL(J),J=.,N)
   86 FORMAT(' BL'/(1X,5E14.7))
      WRITE(3,87)(SIG(J),J=1,KF)
   87 FORMAT(' SIG'/(1X,5E14.7))
      K5=K4-K4/3*3
      K6=K5*N
      DO 79 J=1,N
      J1=J+K6
      U2(J1)=UZ(J)
   79 CONTINUE
      F2(K5+1)=SL
      IF(SL.GE.F2(K2+1)) GO TO 80
   31 K=K+1
      IF(K-N9) 95,95,96
   96 WRITE(3,91)
   91 FORMAT('OPLACE 3')
      STOP
   80 STOP
      END
$DATA
  3  2 20  0  0 0.5000000E-02 0.1769187E+00 0.1000000E-04
 0.1769187E+00 0.1983635E+00
-0.2349120E-01-0.7859229E-03 0.6150758E+04
  1  2
 0.0000E+00 0.1000E-03 0.2000E-03 0.4000E-03 0.1000E-02 0.2000E-02 0.4000E-02
 0.1000E-01 0.2000E-01 0.4000E-01 0.1000E+00 0.2000E+00 0.3000E+00 0.4000E+00
 0.5000E+00 0.6000E+00 0.7000E+00 0.8000E+00 0.9000E+00 0.1000E+01 0.1100E+01
 (.1200E+01 0.1300E+01 0.1400E+01 0.1500E+01 0.2000E+01 0.2500E+01 0.3000E+01
 0.3500E+01 0.4000E+01 0.4500E+01
$STOP
/*
//
```

2-4. Printed Output

```
  N  KF  N9  N7NIV       BZ                BKK               EPS
  3   2  20   0   0 0.4999999E-02 0.1769187E+00 0.1000000E-04

BK
 ).1769187E+00 0.1983635E+00

BL
-0.2349120E-01-0.7859229E-03 0.6150758E+04

MP
  1  2

DEL
 0.0000E+00 0.1000E-03 0.2000E-03 0.4000E-03 0.1000E-02 0.2000E-02 0.4000E-02
 0.1000E-01 0.2000E-01 0.4000E-01 0.1000E+00 0.2000E+00 0.3000E+00 0.4000E+00
 0.5000E+00 0.6000E+00 0.7000E+00 0.8000E+00 0.9000E+00 0.1000E+01 0.1100E+01
 ).1200E+01 0.1300E+01 0.1400E+01 0.1500E+01 0.2000E+01 0.2500E+01 0.3000E+01
 0.3500E+01 0.4000E+01 0.4500E+01

PD
 '.7885438E-04 0.1205725E-01-0.4798896E-04-0.5610914E-03 0.5312272E-04
```

```
 0.4530929E-02-0.4799328E-04-0.5611032E-03 0.8134719E-04 0.4228095E-03
-0.4650035E-04-0.3118563E-02 0.6312469E-04 0.4530933E-02-0.4650054E-04
-0.3118567E-02 0.6286116E-04 0.1999587E-04 0.8659626E+00-0.8833485E+02
 0.1339624E+01-0.1283093E+03-0.1938630E+01-0.1553283E+01 0.1339585E+01
-0.1283092E+03 0.8471759E+01-0.1841436E+03-0.6670155E+01 0.3184797E+03
-0.1938527E+01-0.1552933E+01-0.6669990E+01 0.3184807E+03 0.7124145E+01
 0.4370879E+02 0.5777996E-02-0.3496083E-02-0.8552313E-02 0.4094768E-02
 0.8741899E-03-0.3979273E-02-0.4085985E-01 0.1629616E-01-0.2022418E+00
-0.5957566E+00-0.1502222E+01-0.1168442E+01 0.1083533E+01 0.2743393E+01
-0.1461533E-01 0.1548914E-01 0.1427788E-03 0.1512150E-01-0.9078211E-04
-0.7083474E-03 0.1215035E-03 0.5202800E-02-0.9077982E-04-0.7083609E-03
 0.1250480E-03 0.1086308E-02-0.9104886E-04-0.3229622E-02 0.1214975E-03
 0.5202807E-02-0.9104720E-04-0.3229620E-02 0.1243404E-03 0.1420562E-02
 0.8072982E+00-0.6316139E+02 0.2993755E+00-0.1122266E+03-0.1971375E+01
-0.2393962E+02 0.2994356E+00-0.1122267E+03 0.7541306E+01-0.3939557E+02
-0.4937274E+01 0.3205747E+03-0.1971402E+01-0.2393942E+02-0.4937516E+01
 0.3205747E+03 0.8658371E+01 0.1134583E+03 0.5126588E-02-0.4392274E-02
-0.5329987E-02 0.7698957E-02-0.1966548E-03-0.4919115E-02-0.1820137E-01
 0.3932767E-01-0.4714749E+00-0.3997546E+00-0.1342090E+01 0.3079779E+00
 0.1995680E+01 0.2246219E+01-0.5593035E-02 0.1680823E-01
SIG
 0.2761552E-01 0.2126241E-01
DF
-0.3567E+00 0.3120E+00-0.1424E+02
SIG
 0.2761540E-01 0.2126250E-01
GRADIENT ON NEGATIVE SIDE
-0.3575E+00 0.3047E+00-0.1424E+02
GRADIENT ON POSITIVE SIDE
 0.0000E+00 0.0000E+00 0.9864E+02

  L ARGUMENT
 0.0000000E+00 0.1000000E-03 0.5000000E-04 0.2500000E-04 0.1250000E-04
 0.1875000E-04 0.2187499E-04
FUNCTION
 0.8324286E+02 0.8324286E+02 0.8324284E+02 0.8324284E+02 0.8324281E+02
 0.8324281E+02 0.8324286E+02
DERIVATIVE
-0.6171889E+00 0.1871918E+01 0.6256269E+00 0.3992915E-02-0.3081486E+00
-0.1522908E+00-0.7668984E-01
U COORDINATE
-0.8245E+00-0.1359E+00 0.1000E+01
U DIRECTION
 0.1000E+01-0.8522E+00 0.0000E+00
BL
-0.2348764E-01-0.7860472E-03 0.4999996E+04
SIG
 0.2760046E-01 0.2127135E-01
DF
-0.3517E+00-0.3228E+00-0.1445E+02
SIG
 0.2760046E-01 0.2127135E-01
GRADIENT ON NEGATIVE SIDE
-0.3517E+00-0.3228E+00-0.1445E+02
GRADIENT ON POSITIVE SIDE
 0.0000E+00 0.0000E+00 0.9842E+02

PLUS THE PRINTOUT FOR FOUR MORE GRADIENT SEARCHES.
THE PRINTOUT FOR THE LAST GRADIENT SEARCH IS
```

```
 5 ARGUMENT
 0.0000000E+00 0.1000000E-03 0.5000000E-04 0.2500000E-04 0.3749999E-04
 0.3124999E-04 0.2812500E-04
FUNCTION
 0.8324278E+02 0.8324277E+02 0.8324292E+02 0.8324278E+02 0.8324275E+02
 0.8324277E+02 0.8324278E+02
DERIVATIVE
-0.2659193E+00 0.7551166E+00 0.2439805E+00-0.1099205E-01 0.1154352E+0[illegible]
 0.5368108E-01 0.2092940E-01
U COORDINATE
-0.8244E+00-0.1359E+00 0.1000E+01
U DIRECTION
 0.1000E+01-0.5307E+00 0.0000E+00
BL
-0.2347656E-01-0.7860924E-03 0.499996E+04
SIG
 0.2760166E-01 0.2127067E-01
```

2-5. References

[1] D. A. Pierre, "Optimization Theory with Applications," Wiley, 1969, page 313.

[2] R. F. Harrington and J. R. Mautz, "Modal Analysis of Loaded N-Port Scatterers," Report AFCRL-72-179 , Contract No. F19628-68-C-0180 between Syracuse University and Air Force Cambridge Research Laboratories, March 1972, page 19, equations (74) and (75).

[3] R. Bellman, "Introduction to Matrix Analysis", McGraw-Hill, 1960, page 161, 1d.