NAG-016

69-TR-18

DEPARTMENT OF ELECTRICAL ENGINEERING

PUBLICATIONS

RESPONSE PROGRAM FOR AN INHOMOGENEOUS CASCADE OF RECTANGULAR WAVEGUIDES

John W. Bandler
P. A. Macdonald

THE UNIVERSITY OF MANITOBA
WINNIPEG, CANADA.

June 17, 1969

Document No. NAG-016

RESPONSE PROGRAM FOR AN INHOMOGENEOUS CASCADE OF RECTANGULAR WAVEGUIDES

John W. Bandler

P. A. Macdonald

Numerical Applications Group
Department of Electrical Engineering
University of Manitoba
Winnipeg, Manitoba

Programmer P. A. Macdonald

Author John W. Bandler

Approved ______________________, ______________________

~~Dr~~.A.Wexler
Assistant Professor
Numerical Applications Group

RESPONSE PROGRAM FOR AN INHOMOGENEOUS CASCADE OF RECTANGULAR WAVEGUIDES

Description This package calculates the input admittance versus frequency to an arbitrarily terminated inhomogeneous cascade of rectangular waveguides with or without junction discontinuity effects.

Language FORTRAN IV

Authors J.W. Bandler and P.A. Macdonald, Numerical Applications Group, Electrical Engineering Department, University of Manitoba, Winnipeg, Canada.

Availability Listing presented with description.

This package of subprograms calculates the complex normalized input admittance versus frequency to an arbitrarily terminated homogeneous or inhomogeneous cascade of rectangular waveguides operating in the H_{10} mode. Discontinuity effects due to small symmetrical steps can be taken into account.

There is a LOGICAL FUNCTION subprogram (Fig. 1) which tests constraints:

CUTOFF (M, A, B, FL, FU, PRINT, UNIT, BEWARE, RANGE, SMALLA, SMALLB)

Input Variables

M an integer specifying the number of waveguide sections

A array of guide widths in cm from source to load including source and load guides

B array of guide heights in cm from source to load including source and load guides

FL the lower edge of the frequency band in GHz

FU the upper edge of the frequency band in GHz

PRINT a logical variable; when .TRUE. details of constraint violations are printed out; when .FALSE. nothing is printed out. The following can be printed out: whether the TE10 mode is cutoff or which higher-order mode (TE01, TE20 or TE30) may propagate and in which guide (counting from source to load with the source guide as 1); and whether the small step approximation is deteriorating and at which junction (counting from source to load) it occurs

UNIT an integer specifying the data set reference number of the output unit

---

This work was carried out with financial assistance from the Faculty of Graduate Studies of the University of Manitoba and the National Research Council of Canada.

Output Variables

BEWARE a logical variable; becomes .TRUE. if a higher-order mode can propagate

RANGE a logical variable; becomes .FALSE. if the frequency band is not in the range defined by the cutoff frequencies of the H_{10} and H_{30} modes

SMALLA a logical variable; becomes .FALSE. if the ratio of the widths of any adjacent guides lies outside the range 0.7 to 1/0.7

SMALLB a logical variable; becomes .FALSE. if the ratio of the heights of any adjacent guides lies outside the range 0.5 to 2

CUTOFF becomes .TRUE. if the dominant mode in any guide is cutoff in the band

The response package (Fig. 2) is called by calling COMPLEX FUNCTION subprogram

YRECT (M, A, B, L, F, YLN, EFFECT, R, FCBYF)

The variables M, A and B are the same as before. The rest are defined as follows.

Input Variables

L array of guide lengths in cm from source to load including source and load guides whose lengths are immaterial since they are not used

F operating frequency in GHz between FL and FU

YLN complex normalized load admittance at frequency F

EFFECT a logical variable; when set .TRUE. includes susceptances due to small symmetrical H- and E-plane steps; when .FALSE. ignores discontinuities

Output Variables

R the transformer impedance ratio (matched load to matched source)

FCBYF array of ratios of cutoff to operating frequency from source to load

YRECT complex normalized input admittance at the operating frequency

Variables L, PRINT, UNIT, BEWARE, RANGE, SMALLA, SMALLB, CUTOFF, YLN, EFFECT and YRECT must be defined and variables A, B, L and FCBYF must be dimensioned in the calling program. The user should first call CUTOFF. Depending on the outcome indicated by the logical variables the user can decide whether to call YRECT to evaluate the frequency response between FL and FU. If CUTOFF is .TRUE. and/or RANGE is .FALSE., then YRECT should not be called.

The theory follows Bandler[1,2] and the program has been tested on the IBM 360/65.

The cooperation of the Institute for Computer Studies of the University of Manitoba is acknowledged.

---

[1] J.W. Bandler, "Computer optimization of inhomogeneous waveguide transformers," this issue p. --

[2] However, there are some differences between this program the one used for optimizing inhomogeneous transformers. For example, most variables in the optimization program are COMMON and the magnitude of the reflection coefficient is obtained instead of the input admittance. Other calculations not required are also ommitted.

```
      LOGICAL FUNCTION CUTOFF(M, A, B, FL, FU, PRINT, UNIT, BEWARE,
     XRANGE, SMALLA, SMALLB)
      DIMENSION A(1), B(1), MODE(3)
      LOGICAL PRINT, BEWARE, RANGE, SMALLA, SMALLB
      INTEGER UNIT
      DATA CBY2/14.989625/, MODE/'TE01', 'TE20', 'TE30'/
      CUTOFF = .FALSE.
      RANGE = .TRUE.
      BEWARE = .FALSE.
      SMALLA = .TRUE.
      SMALLB = .TRUE.
      MP2 = M + 2
      DO 9 I = 1, MP2
      IM1 = I - 1
      IP1 = I + 1
      FC = CBY2 / A(I)
      FCB = CBY2 / B(I)
      IF (FU .LT. FCB) GO TO 2
      BEWARE = .TRUE.
      IF (PRINT) WRITE (UNIT, 1) MODE(1), IM1
1     FORMAT ('0'A4, ' MODE MAY PROPAGATE IN GUIDE'I3)
2     IF (FL .GT. FC) GO TO 4
      CUTOFF = .TRUE.
      RANGE = .FALSE.
      IF (PRINT) WRITE (UNIT, 3) IM1
3     FORMAT ('0TE10 MODE CUTOFF IN GUIDE'I3)
      GO TO 5
4     IF (FU .LT. FC + FC) GO TO 5
      BEWARE = .TRUE.
      IF (PRINT) WRITE (UNIT, 1) MODE(2), IM1
      IF (FU .LT. FC + FC + FC) GO TO 5
      RANGE = .FALSE.
      IF (PRINT) WRITE (UNIT, 1) MODE(3), IM1
5     IF ( .NOT. RANGE) BEWARE = .TRUE.
      IF (I .EQ. MP2) RETURN
      ALPHA = A(I) / A(IP1)
      IF (ALPHA .GT. .7 .AND. ALPHA .LT. 1. / .7) GO TO 7
      SMALLA = .FALSE.
      IF (PRINT) WRITE (UNIT, 6) I
6     FORMAT ('0A'' / A < .7 => SMALL STEP APPROXIMATION DETERIORATING A
     XT JUNCTION'I3)
7     BETA = B(I) / B(IP1)
      IF (BETA .GT. .5 .AND. BETA .LT. 2.) GO TO 9
      SMALLB = .FALSE.
      IF (PRINT) WRITE (UNIT, 8) I
8     FORMAT ('0B'' / B < .5 => SMALL STEP APPROXIMATION DETERIORATING A
     XT JUNCTION'I3)
9     CONTINUE
      END
```

Fig. 1 FORTRAN IV listing of LOGICAL FUNCTION CUTOFF

```
      COMPLEX FUNCTION YRECT(M, A, B, L, F, YLN, EFFECT, R, FCBYF)
      REAL L(1), LAMBDA
      DIMENSION A(1), B(1), FCBYF(1)
      COMPLEX YLN, YI, P
      LOGICAL EFFECT
      DATA CBY2, C, TWOBYC / 14.989625, 29.97925, .0667128 /, PI, PIBY2
     X / 3.141593, 1.570796 /
      COMMON / RECT / LAMBDA, SK, SKP1, YK, YKP1
      MP2 = M + 2
      LAMBDA = C / F
      SK = CBY2 / (F * A(MP2))
      FCBYF(MP2) = SK
      SK = SQRT(1. - SK * SK)
      YK = SK / B(MP2)
      R = 1. / YK
      YI = YK * YLN
      DO 2 I = 1, M
      K = MP2 - I
      SKP1 = SK
      YKP1 = YK
      SK = CBY2 / (F * A(K))
      FCBYF(K) = SK
      SK = SQRT(1. - SK * SK)
      YK = SK / B(K)
      IF (EFFECT) YI = YI + CMPLX(0., BT(A, B, K))
      THETA = PI * AMOD(TWOBYC * F * L(K) * SK, 1.)
      IF (ABS(THETA - PIBY2) .GT. 1.E - 5) GO TO 1
      YI = YK * YK / YI
      GO TO 2
1     P = CMPLX(0., TAN(THETA))
      YI = YK * (YI + YK * P) / (YK + YI * P)
2     CONTINUE
      YKP1 = YK
      SKP1 = SK
      SK = CBY2 / (F * A(1))
      FCBYF(1) = SK
      SK = SQRT(1. - SK * SK)
      YK = SK / B(1)
      R = R * YK
      IF (EFFECT) YI = YI + CMPLX(0., BT(A, B, 1))
      YRECT = YI / YK
      RETURN
      END
```

Fig. 2 FORTRAN IV listings of the function subprograms which calculate the response. They are brought into action by calling COMPLEX FUNCTION YRECT. The theory and most of the notation follows a paper by Bandler[1] although these programs are not identical to ones he used.

```
      FUNCTION BT(A, B, K)
      REAL LAMBDA, LAMK, LAMKP1
      DIMENSION A(1), B(1)
      COMMON / RECT / LAMBDA, SK, SKP1, YK, YKP1
      LAMK = LAMBDA / SK
      LAMKP1 = LAMBDA / SKP1
      AK = A(K)
      AKP1 = A(K + 1)
      BK = B(K)
      BKP1 = B(K + 1)
      IF (AK - AKP1) 1, 2, 3
1     BH = BNH(AKP1, AK, LAMKP1) * YKP1
      GO TO 4
2     BH = 0.
      GO TO 4
3     BH = BNH(AK, AKP1, LAMK) * YK
4     IF (BK - BKP1) 5, 6, 7
5     BE = BNE(BKP1, BK, LAMKP1) * YKP1
      GO TO 8
6     BE = 0.
      GO TO 8
7     BE = BNE(BK, BKP1, LAMK) * YK
8     BT = BE + BH
      RETURN
      END
```

```
      FUNCTION BNH(A, APRM, LAMG)
      REAL LAMBDA, LAMG, LNHB
      COMMON / RECT / LAMBDA
      BETA = 1. - APRM / A
      HBETA = .5 * BETA
      LNHB = ALOG(HBETA)
      Q = A / (1.5 * LAMBDA)
      Q = 1. - SQRT(1. - Q * Q)
      QPRM = APRM / (1.5 * LAMBDA)
      QPRM = 1. - SQRT(1. - QPRM * QPRM)
      BNH = - LAMG / (A + A) * BETA * BETA * (1. + BETA) * LNHB /
     X(HBETA - 1.) * (1. - 27. * (Q + QPRM) / (8. * (1. - 8. * LNHB)))
      RETURN
      END
```

```
      FUNCTION BNE(B, BPRM, LAMG)
      REAL LAMG
      DELTA = 1. - BPRM / B
      HDELTA = .5 * DELTA
      BBYLG = B / LAMG
      BNE = BBYLG * HDELTA * DELTA * (ALOG(HDELTA) / (HDELTA - .5) + 1.
     X + 17. / 16. * BBYLG * BBYLG)
      RETURN
      END
```