

TR-O-0079

40

スペクトル領域法を用いたスロット結合
円形マイクロストリップアンテナの解析

村上 康

1995. 3. 23

ATR光電波通信研究所

あらし：

本テクニカルレポートはマイクロストリップ線路、コプレーナ線路などの多層構造の伝送線路あるいはマイクロストリップアンテナなどの解析に用いられるスペクトル領域法を用いて、スロット結合型円形マイクロストリップアンテナの解析を行ったものである。

解析例としては、スロット結合型円形マイクロストリップアンテナのうち、スロットをX軸方向にオフセットした直線偏波アンテナの解析を行い、実験値との比較を行っている。また、プログラムの解説などは省略させていただくが、Y軸にオフセットしたアンテナについても解析を行っている。これら自体はすでに多くの論文が出されているが、本テクニカルレポートでは初心者にもわかるように、基礎的・理論的な側面を中心に述べたものである。より高度な解析を行おうとされる方々には多くの参考文献を挙げておくのでそちらを参照されたい。

目 次

1. スペクトル領域法	1
1. 1 スペクトル領域法の特徴	1
1. 2 多層基板内の電磁界 (一般解)	2
1. 3 グリーン関数の導出例	5
2. 解析例 I(Y 軸オフセット MSA)	8
2. 1 モデリング	8
2. 2 定式化	9
2. 3 使用されるグリーン関数と展開関数について	13
2. 4 励振方法と S パラメータの導出	17
2. 5 数値積分	18
2. 6 励振モードとスロットの関係	22
2. 7 計算結果	25
3. 解析例 II(X 軸オフセット MSA)	27
4. まとめ	28
5. 謝辞	29
参考文献	29
付録	31
解析プログラムソース	32

1 スペクトル領域法 [1][2]

1.1 スペクトル領域法の特徴

アンテナ解析で用いられるスペクトル領域法は、モーメント法的一种あるいはモーメント法そのものといっても過言ではないと思われる。通常、線状アンテナなどで用いられるモーメント法との相違点はグリーン関数の表式である。線状アンテナ解析等ではグリーン関数が空間領域において比較的容易に定式化が出来る。一方、スペクトル領域法を用いる解析の対象物はマイクロストリップアンテナ等多層構造を持つものである。この場合、空間領域においてグリーン関数を定式化することが非常に困難となる。一方層の平面内で2次元 Fourier 変換を行い、波数空間で Maxwell の方程式を考察すると、層の法線方向に対する2つの伝送方程式 (TE, TM) となる。このため多層構造のグリーン関数も容易に求めることができる。

その他に長所として以下に示す特徴をもっている。

- (1) 積分方程式ではなく代数方程式を解くことになる
 - (2) ガラーキン法を用いるため、用いられる行列を大きくすることにより解の精度をあげることができる
 - (3) 多層基板を用いても、数値処理にあまり影響を与えない (数式的には多少複雑になる)
 - (4) 解の物理的性質を解法中に用いるため、解の収束性、得られたモードの確定について不安をもつことが少ない
- 逆に以下に示すような短所もある。
- (5) 基板は横方向に無限長でなければならない (面内では一応であることが求められる)
 - (6) ストリップ導体は無限に薄くなければならない
 - (7) ストリップ導体は完全導体でなければならない (誘電体の損失についてはその限りではない)

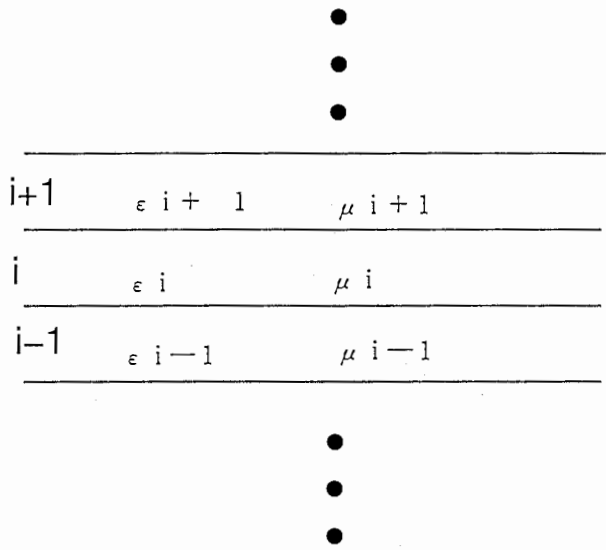


図1：多層基板

1.2 多層基板内の電磁界 (一般解)

図1に示すような多層基板内の*i*番目の層内の電磁界について考察する。ここで

$$A_z^i = \frac{\Pi_{mz}^i}{j\omega\epsilon_i} \quad (1)$$

$$F_z^i = \frac{\Pi_z^i}{j\omega\mu_i} \quad (2)$$

とする。ここで Π_{mz}^i は*i*層での Magnetic Hertz vecor の *z* 成分、 Π_z^i は Electric Hertz vecor の *z* 成分である。この2つの成分で層内の電磁界は表記され、次のように表される。

$$E_x^i = \frac{1}{j\omega\epsilon_i} \frac{\partial^2 A_z^i}{\partial x \partial z} - \frac{\partial F_z^i}{\partial y} \quad (3)$$

$$E_y^i = \frac{1}{j\omega\epsilon_i} \frac{\partial^2 A_z^i}{\partial y \partial z} + \frac{\partial F_z^i}{\partial x} \quad (4)$$

$$E_z^i = \frac{1}{j\omega\epsilon_i} \left[\frac{\partial^2 A_z^i}{\partial z^2} + k^2 A_z^i \right] \quad (5)$$

$$H_x^i = \frac{1}{j\omega\mu_i} \frac{\partial^2 F_z^i}{\partial x \partial z} + \frac{\partial A_z^i}{\partial y} \quad (6)$$

$$H_y^i = \frac{1}{j\omega\mu_i} \frac{\partial^2 F_z^i}{\partial y \partial z} - \frac{\partial A_z^i}{\partial x} \quad (7)$$

$$H_z^i = \frac{1}{j\omega\mu_i} \left[\frac{\partial^2 F_z^i}{\partial z^2} + k^2 F_z^i \right] \quad (8)$$

となる。一方、各成分を x 、 y について Fourier 変換すると、

$$\tilde{E}_x^i = \frac{k_x}{\omega\varepsilon_i} \frac{\partial \tilde{A}_z^i}{\partial z} - jk_y \tilde{F}_z^i \quad (9)$$

$$\tilde{E}_y^i = \frac{k_y}{\omega\varepsilon_i} \frac{\partial \tilde{A}_z^i}{\partial z} + jk_x \tilde{F}_z^i \quad (10)$$

$$\tilde{E}_z^i = \frac{1}{j\omega\varepsilon_i} \left[\frac{\partial^2 \tilde{A}_z^i}{\partial z^2} + k^2 \tilde{A}_z^i \right] \quad (11)$$

$$\tilde{H}_x^i = \frac{k_x}{\omega\mu_i} \frac{\partial \tilde{F}_z^i}{\partial z} + jk_y \tilde{A}_z^i \quad (12)$$

$$\tilde{H}_y^i = \frac{k_y}{\omega\mu_i} \frac{\partial \tilde{F}_z^i}{\partial z} - jk_x \tilde{A}_z^i \quad (13)$$

$$\tilde{H}_z^i = \frac{1}{j\omega\mu_i} \left[\frac{\partial^2 \tilde{F}_z^i}{\partial z^2} + k^2 \tilde{F}_z^i \right] \quad (14)$$

となる。ここで

$$\tilde{A} = \int \int_{-\infty}^{\infty} A e^{-jk_x x - jk_y y} dx dy \quad (15)$$

一方、 A_z^i 、 F_z^i とも source free の Maxwell の方程式を満足することより、

$$\nabla^2 A_z^i + k^2 A_z^i = 0 \quad (16)$$

$$\nabla^2 F_z^i + k^2 F_z^i = 0 \quad (17)$$

となる。(16)(17) 両式の両辺に Fourier 変換を行うと、

$$\frac{\partial^2}{\partial z^2} \tilde{A}_z^i + (k^2 - k_x^2 - k_y^2) \tilde{A}_z^i = 0 \quad (18)$$

$$\frac{\partial^2}{\partial z^2} \tilde{F}_z^i + (k^2 - k_x^2 - k_y^2) \tilde{F}_z^i = 0 \quad (19)$$

を得る。(18)(19) 式は z 方向への伝送方程式である。このことより、 i 層での A_z^i 、 F_z^i の一般解は次のように表される。

$$\tilde{A}_z^i = a_i (e^{-jk_{zi}z} + \Gamma_{mi} e^{jk_{zi}z}) \quad (20)$$

$$\tilde{F}_z^i = f_i (e^{-jk_{zi}z} + \Gamma_{ei} e^{jk_{zi}z}) \quad (21)$$

$$k_{zi} = \sqrt{k_i^2 - k_x^2 - k_y^2}$$

Γ_{mi} は \tilde{A}_z^i に関する反射係数、 Γ_{ei} は \tilde{F}_z^i に関する反射係数である。
ここで、

$$k_x = \beta \cos \alpha$$

$$k_y = \beta \sin \alpha$$

とし、座標変換

$$u = x \cos \alpha + y \sin \alpha$$

$$v = -x \sin \alpha + y \cos \alpha$$

を行うと、

$$\tilde{E}_u^i = \frac{\beta}{\omega \varepsilon_i} \frac{\partial \tilde{A}_z^i}{\partial z} \quad (22)$$

$$\tilde{E}_v^i = j\beta \tilde{F}_z^i \quad (23)$$

$$\tilde{E}_z^i = \frac{1}{j\omega \varepsilon_i} \left[\frac{\partial^2 \tilde{A}_z^i}{\partial z^2} + k^2 \tilde{A}_z^i \right] \quad (24)$$

$$\tilde{H}_u^i = \frac{\beta}{\omega \mu_i} \frac{\partial \tilde{F}_z^i}{\partial z} \quad (25)$$

$$\tilde{H}_v^i = -j\beta \tilde{A}_z^i \quad (26)$$

$$\tilde{H}_z^i = \frac{1}{j\omega \mu_i} \left[\frac{\partial^2 \tilde{F}_z^i}{\partial z^2} + k^2 \tilde{F}_z^i \right] \quad (27)$$

となる。(22)-(27) 式を比較すると、 $(\tilde{E}_u^i, \tilde{H}_v^i, \tilde{E}_z^i)$ は \tilde{A}_z^i のみで記述されることになり、一方、 $(\tilde{H}_u^i, \tilde{E}_v^i, \tilde{H}_z^i)$ は \tilde{F}_z^i のみで記述されることになる。それぞれは Z 方向への TM モード、TE モードに対応する。よって、電流源、磁流源のない場合の多層基板内の電磁界は TE, TM モードに分離されており、各々 1 次元伝送線路で記述されることになる。これを等価回路表示すると図 2 のようになる。但し、実際の計算では、TE、TM 両モードが励振されるため、TE モード、TM モードの重ね合わせで表記されることになる。

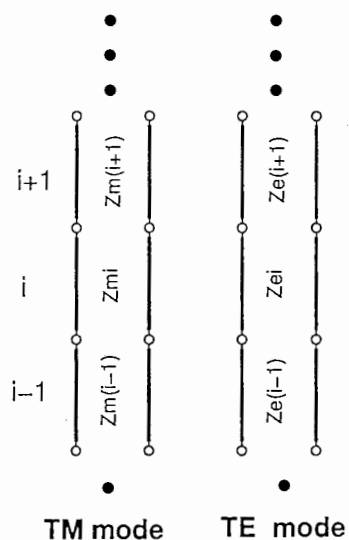


図2：多層基板の等価回路表示

1.3 グリーン関数の導出例 [3][4]

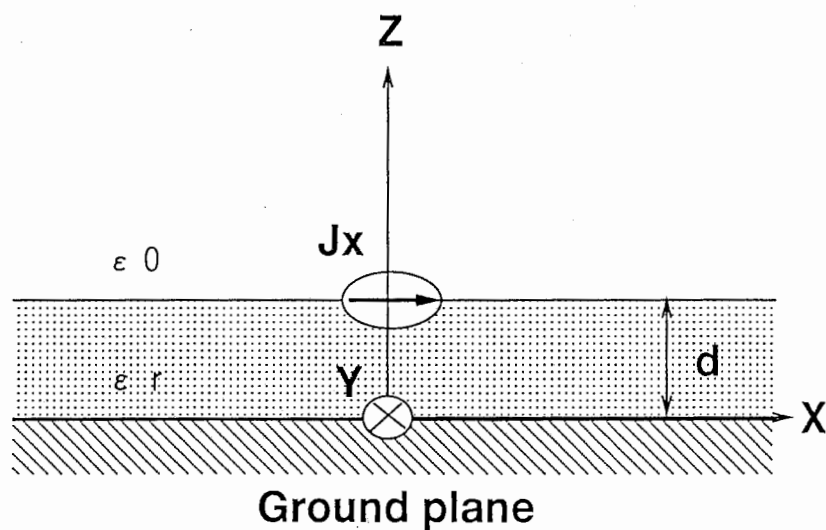


図3：マイクロストリップタイプの基板

実際のグリーン関数の導出の例として、図3に示すようなマイクロストリップタイプの基板についてグリーン関数を求める。電流源としては、誘電体と空気の境界面に x 方向を向いた無限小の電流源を考える。これを数式で表すと、

$$\vec{J}_x = \delta(x)\delta(y)\delta(z-d)\hat{x} \quad (28)$$

となる。このように多層基板構造では、境界面上に電流源あるいは磁流源があること

が多いため、デルタ関数を直接解法するよりもこれらを面電流 $\tilde{n} \times \vec{H}$ あるいは面磁流 $-\tilde{n} \times \vec{E}$ で表すことが多い。この場合には source free のグリーン関数の解と境界条件だけからグリーン関数を導出することができるので非常に有効である。

まずはグラウンド板での境界条件は

$$\begin{aligned}\frac{\partial E_z}{\partial n} &= 0 \\ H_z &= 0\end{aligned}$$

より、領域 I における \tilde{A}_z^I および \tilde{H}_z^I は次のように表される。

$$\tilde{A}_z^I = a_1 \cos k_{z1} z \quad (29)$$

$$\tilde{F}_z^I = f_1 \sin k_{z1} z \quad (30)$$

$$k_{z1} = \sqrt{\epsilon_r k_0^2 - k_x^2 - k_y^2}$$

領域 II については、空気が $z = \infty$ までであるために、無限遠からの反射を考える必要はない。よって、

$$\tilde{A}_z^{II} = a_2 e^{-jk_{z2}(z-d)} \quad (31)$$

$$\tilde{F}_z^{II} = f_2 e^{-jk_{z2}(z-d)} \quad (32)$$

$$k_{z2} = \sqrt{k_0^2 - k_x^2 - k_y^2}$$

となる。次に、 $z = d$ における境界条件

1) $\tilde{E}_x, \tilde{E}_y, \tilde{H}_x$ の連続性、

2) $\tilde{H}_y^I - \tilde{H}_y^{II} = 1$

のもとに a_1, a_2, f_1, f_2 を解くと、

$$a_1 = \frac{j\epsilon_r}{k_x^2 + k_y^2} \frac{k_{z2} k_x}{\epsilon_r k_{z2} \cos k_{z1} d + j k_{z1} \sin k_{z1} d} \quad (33)$$

$$f_1 = \frac{\omega k_y}{k_x^2 + k_y^2} \frac{1}{k_{z1} \cos k_{z1} d + j k_{z2} \sin k_{z1} d} \quad (34)$$

を得る。(33)(34) 式を (9)-(14) 式に代入することで波数空間でのグリーン関数を求めることができる。ここでは例として、 $\tilde{E}_x|_{z=d}$ を求める。このグリーン関数はマイクロストリップ線路、マイクロストリップアンテナの解析に必ずといっていいほど現れるグリーン関数である。

$$\begin{aligned}
\tilde{E}_x|_{z=d} &= \tilde{E}_x^I \\
&= \tilde{G}_{xx}^{EJ}(k_x, k_y) \\
&= -j \frac{Z_0 (\epsilon_r k_0^2 - k_x^2) k_{z1} \cos k_{z1} d + j k_{z2} (k_0^2 - k_x^2) \sin \cos k_{z1} d}{T_e T_m} \sin k_{z1} d \quad (35)
\end{aligned}$$

$$T_e = k_{z1} \cos k_{z1} d + j k_{z2} \sin k_{z1} d$$

$$T_m = \epsilon_r k_{z2} \cos k_{z1} d + j k_{z1} \sin k_{z1} d$$

となる。これを実空間上で表すと、

$$G_{xx}^{EJ} = \frac{1}{4\pi^2} \int \int_{-\infty}^{\infty} \tilde{G}_{xx}^{EJ}(k_x, k_y) e^{jk_x(x-x') + jk_y(y-y')} dk_x dk_y \quad (36)$$

となる。これより以降では G_{xx}^{EJ} は x 方向の単位電流により励振される電界の x 成分を表すものとする。

T_e 、 T_m はスラブ導波路の特性方程式である。つまりこの2つの方程式の解が表面波モードの伝搬定数である。特にマイクロストリップアンテナでは、アンテナ基板は低い誘電率、給電線路基板は薄くすることが一般的であり、2つ以上の表面波が同一基板上に励振されることは少ない。ただ、このうち TM モードの最低次モード TM_0 モードはゼロカットオフのために常に存在する。このモードはアレーアンテナでビームを走査する場合には、隣合う素子との励振差がちょうど表面波の伝搬位相と一致する時、表面波のみが励振され、スキャンブラインドネス [5]–[6] という現象を起こすことが知られている。

また、この極の扱い方については次の章で述べることにする。

2 解析例 I(Y 軸オフセット MSA)

2.1 モデリング

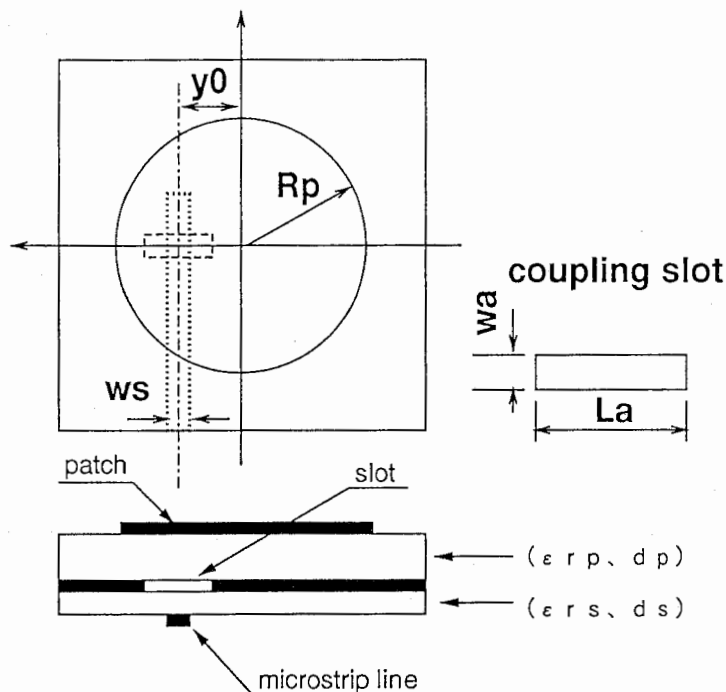


図4：スロット結合型円形マイクロストリップアンテナ
 J_x, J_y (patch)

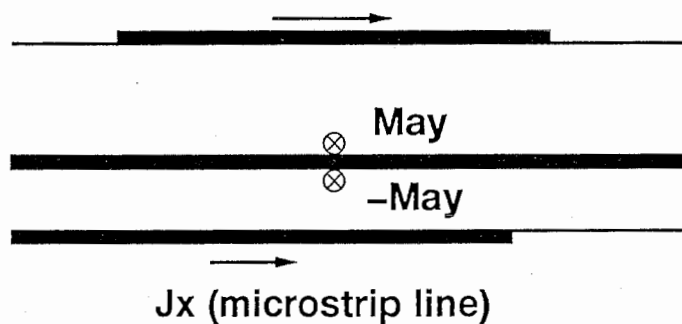


図5：スロット結合型円形マイクロストリップアンテナのモデル化

この章では、実際に図4に示すようにY方向にオフセットしたスロット結合型マイクロストリップアンテナの解析を行い、実験結果との比較を行う。まず、アンテナのモデル化であるが、これは Schaubert[7] のモデリングを用いるものとする。モデル化されたアンテナを図5に示す。マイクロストリップ線路上の電流としては、伝搬方向(x方向)のみを考え、円形アンテナ上ではx方向、y方向両方を考える。矩形アンテナの場合には、流れる電流の方向は1方向であるためx方向のみで十分であるが、円形の

場合には両方考える必要がある。またスロット上の磁流はアンテナ方向とマイクロストリップ線路方向で等振幅、逆位相で励振するものとして、電界の接線成分の連続性を満足させている。また、この磁流源の導入によりアンテナ部と線路部を別けて考えることができる。

2.2 定式化

図5に示されるモデルに必要とされる境界条件は次の3つである。

- 1) マイクロストリップ線路上で電界の接線成分が0
- 2) マイクロストリップアンテナ上で電界の接線成分が0
- 3) スロット上で磁界の接線成分の連続性

まず始めに1)について考察する。磁流源の導入によりアンテナ部と線路部を別々に考えることができるため、マイクロストリップ線路上で電界は、

$$E_x^{ml} = \int_{ml} G_{xx}^{EJ(I)} J_x^{ml} dS_{ml} - V_a \int_{slot} G_{xy}^{EM(I)} M_{ay} dS_{slot} \quad (37)$$

ここで、マイクロストリップ上の電流をある関数系で展開すると、

$$J_x^{ml} = \sum_{n=1}^M I_n J_{xn}^{ml}$$

$$E_x^{ml} = \sum_{n=1}^M I_n \int_{S_n} G_{xx}^{EJ(I)} J_{xn}^{ml} dS_{S_n} - V_a \int_{slot} G_{xy}^{EM(I)} M_{ay} dS_{slot} \quad (38)$$

S_n は J_{xn}^{ml} が定義されている領域である。ここにガラーキン法をすると、

$$\begin{aligned} & \int_{S_m} J_{xm}^{ml} E_x^{ml} dS_n \\ &= \sum_{n=1}^M I_n \int_{S_m} \int_{S_n} J_{xm}^{ml} G_{xx}^{EJ(I)} J_{xn}^{ml} dS_n dS_m - V_a \int_{S_m} \int_{slot} J_{xm}^{ml} G_{xy}^{EM(I)} M_{ay}^{ml} dS_s dS_n \end{aligned}$$

これを行列表示をすると、

$$[V_m] = [Z_{mn}^{ml}] [I_n] - [C_m^{ml}] V_a \quad (40)$$

ここで、

$$\begin{aligned}
z_{nm} &= \int_{S_m} \int_{S_n} G_{xx}^{EJ(I)}(x, y; x', y') \cdot J_{x'm}(x', y') \cdot J_{xn}(x, y) dS_m dS_n \\
&= \frac{1}{4\pi^2} \int_{S_m} \int_{S_n} \left[\int_{-\infty}^{\infty} \tilde{G}_{xx}^{EJ(I)}(k_x, k_y) e^{jk_x(x-x') + jk_y(y-y')} dk_x dk_y \right] \\
&\quad J_{x'n}^{ml}(x', y') \cdot J_{xm}^{ml}(x, y) dS_m dS_n
\end{aligned} \tag{41}$$

ここで、パーセバルの定理を用いて、積分順序を入れ換えると、

$$\begin{aligned}
z_{nm} &= \frac{1}{4\pi^2} \int \int_{-\infty}^{\infty} \left[\int_{S_n} J_{x'n}^{ml} e^{-jk_x x' - jk_y y'} dS_n \right] \left[\int_{S_m} J_{xm}^{ml} e^{jk_x x + jk_y y} dS_m \right] \\
&\quad \times \tilde{G}_{xx}^{EJ(I)}(k_x, k_y) dk_x dk_y \\
&= \frac{1}{4\pi^2} \int \int_{-\infty}^{\infty} F^*(J_{xn}^{ml}) \tilde{G}_{xx}^{EJ(I)}(k_x, k_y) F(J_{xm}^{ml}) dk_x dk_y
\end{aligned} \tag{42}$$

$F(\cdot)$: Fourier 変換
 $F^*(\cdot)$: $F(\cdot)$ の複素共役

となる。同様に、

$$C_m = \frac{1}{4\pi^2} \int \int_{-\infty}^{\infty} F^*(J_{xn}^{ml}) \tilde{G}_{xy}^{EM(I)}(k_x, k_y) F(M_{ay}) dk_x dk_y \tag{43}$$

$[V_m]$ は $1 \times M$ の電圧源ベクトル、 $[Z_{mn}^{ml}]$ は $M \times M$ のインピーダンス行列、 $[C_m^{ml}]$ は $1 \times M$ の変換ベクトル (無次元) である。

次に、境界条件 2) について考える。アンテナ上の電流を

$$\vec{J} = \sum_{n=1}^N A_n [J_{xn}^{(p)}(x, y) \hat{x} + J_{yn}^{(p)}(x, y) \hat{y}] \tag{44}$$

と展開すると、アンテナ上の電界 E_x^P 、 E_y^P は次のように表される。

$$\vec{E}_x^P = \sum_N^{n=1} A_n \int_{patch} \left(G_{xx}^{EJ(II)} J_{xn}^{(p)} + G_{xy}^{EJ(II)} J_{yn}^{(p)} \right) dS_{patch} + V_a \int_{slot} G_{xy}^{EM(II)} M_{ay} dS_{slot} \tag{45}$$

$$\vec{E}_y^P = \sum_N^{n=1} A_n \int_{patch} \left(G_{yx}^{EJ(II)} J_{xn}^{(p)} + G_{yy}^{EJ(II)} J_{yn}^{(p)} \right) dS_{patch} + V_a \int_{slot} G_{yy}^{EM(II)} M_{ay} dS_{slot} \tag{46}$$

境界条件 1) と同様にガラーキン法を適用すると、

$$\begin{aligned}
& \int_{patch} (J_{xm}^{(p)} E_x^P + J_{ym}^{(p)} E_y^P) dS_{patch} \\
&= \sum_{n=1}^N A_n \int_{patch} \int_{patch} (J_{xm}^{(p)} G_{xx}^{EJ(II)} J_{xn}^{(p)} + J_{xm}^{(p)} G_{xy}^{EJ(II)} J_{yn}^{(p)} \\
&\quad + J_{ym}^{(p)} G_{yx}^{EJ(II)} J_{xn}^{(p)} + J_{ym}^{(p)} G_{yy}^{EJ(II)} J_{yn}^{(p)}) dS_{patch} dS_{patch} \\
&\quad + V_a \int_{patch} \int_{slot} (J_{xm}^{(p)} G_{xy}^{EM(II)} + J_{ym}^{(p)} G_{yy}^{EM(II)}) M_{ay} dS_{slot} dS_{patch} \quad (47)
\end{aligned}$$

これを行列表示すると、

$$[0] = [Z_{nm}^P][A_n] + [C_m^P]V_a \quad (48)$$

ここで、

$$\begin{aligned}
z_{mn}^P &= \int_{patch} \int_{patch} (J_{xm}^{(p)} G_{xx}^{EJ(II)} J_{xn}^{(p)} + J_{xm}^{(p)} G_{xy}^{EJ(II)} J_{yn}^{(p)} \\
&\quad + J_{ym}^{(p)} G_{yx}^{EJ(II)} J_{xn}^{(p)} + J_{ym}^{(p)} G_{yy}^{EJ(II)} J_{yn}^{(p)}) dS_{patch} dS_{patch} \\
&= \frac{1}{4\pi^2} \int \int_{-\infty}^{\infty} [F^*(\vec{J}_m)]^t [\tilde{G}^{EJ(II)}] [F(\vec{J}_n)] dk_x dk_y \quad (49)
\end{aligned}$$

ここで

$$\begin{aligned}
[F(\vec{J}_n)] &= [F(J_{xn}^{(p)}), F(J_{yn}^{(p)})] \\
\tilde{G}^{EJ(II)} &= \begin{bmatrix} \tilde{G}_{xx}^{EJ(II)} & \tilde{G}_{xy}^{EJ(II)} \\ \tilde{G}_{yx}^{EJ(II)} & \tilde{G}_{yy}^{EJ(II)} \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
c_m^P &= \int_{patch} \int_{patch} (J_{xm}^{(p)} G_{xy}^{EM(II)} + J_{ym}^{(p)} G_{yy}^{EM(II)}) M_{ay} dS_{slot} dS_{patch} \\
&= \frac{1}{4\pi^2} \int \int_{-\infty}^{\infty} F^*(J_{xm}^{(p)}) \tilde{G}_{xy}^{EM(II)} + F^*(J_{ym}^{(p)}) \tilde{G}_{yy}^{EM(II)} F(M_{ay}) dk_x dk_y \quad (50)
\end{aligned}$$

である。ここで、 $[0]$ は $1 \times N$ の零ベクトルである。これはアンテナ上には電圧源、電流源、磁流源が存在しないことによる。また、 $[Z_{nm}^P]$ は $N \times N$ のインピーダンス行列、 $[C_m^P]$ は $1 \times N$ の変換ベクトルである。

最後に、3) の境界条件について考える。領域 I、II 側の磁界 H_y^I 、 H_y^{II} は次のように表される。

$$H_y^I = \sum_{n=1}^M I_n \int_{S_n} G_{yx}^{HJ(I)} J_{xn} dS_n - V_a \int_{slot} G_{yy}^{HM(I)} M_{ay} dS_{slot} \quad (51)$$

$$H_y^{II} = \sum_{n=1}^M A_n \int_{patch} (G_{yx}^{HJ(II)} J_{xn}^{(p)} + G_{yy}^{HJ(II)} J_{yn}^{(p)}) dS_n + V_a \int_{slot} G_{yy}^{HM(II)} M_{ay} dS_{slot} \quad (52)$$

ここにガラーキン法を適用すると、

$$\begin{aligned} & \int_{slot} M_{ay} (H_y^{II} - H_y^I) dS_{slot} \\ &= \sum_{n=1}^N A_n \int_{patch} \int_{slot} M_{ay} (G_{yx}^{HJ(II)} J_{xn}^{(p)} + G_{yy}^{HJ(II)} J_{yn}^{(p)}) dS_{slot} dS_{patch} \\ & \quad - \sum_{n=1}^N I_n \int_{S_n} \int_{slot} M_{ay} G_{yx}^{HJ(I)} J_{xn} dS_{slot} dS_n \\ & \quad + V_a \int_{slot} \int_{slot} M_{ay} (G_{yy}^{HM(I)} + G_{yy}^{HM(II)}) M_{ay} dS_{slot} dS_{slot} \\ &= 0 \end{aligned} \quad (53)$$

これを行列表示すると、

$$[D_m]^t [A_m] - [E_m] [I_m] + Y_a V_a = 0 \quad (54)$$

ここで、

$$\begin{aligned} D_m &= \int_{patch} \int_{slot} M_{ay} (G_{yx}^{HJ(II)} J_{xn}^{(p)} + G_{yy}^{HJ(II)} J_{yn}^{(p)}) dS_{patch} dS_{slot} \\ &= \frac{1}{4\pi^2} \int \int_{-\infty}^{\infty} F^*(M_{ay}) (\tilde{G}_{yx}^{HJ(II)} F(J_{xn}^{(p)}) + \tilde{G}_{yy}^{HJ(II)} F(J_{yn}^{(p)})) dk_x dk_y \end{aligned} \quad (55)$$

$$\begin{aligned} E_m &= \int_{patch} \int_{slot} M_{ay} G_{yx}^{HJ(I)} J_{xn} dS_{slot} dS_n \\ &= \frac{1}{4\pi^2} \int \int_{-\infty}^{\infty} F^*(M_{ay}) \tilde{G}_{yx}^{HJ(I)} F(J_{xn}) dk_x dk_y \end{aligned} \quad (56)$$

$$\begin{aligned} Y_a &= \int \int_{slot} M_{ay} (G_{yy}^{HM(I)} + G_{yy}^{HM(II)}) M_{ay} dS_{slot} dS_{slot} \\ &= \frac{1}{4\pi^2} \int \int_{-\infty}^{\infty} F^*(M_{ay}) (\tilde{G}_{yy}^{HM(I)} + \tilde{G}_{yy}^{HM(II)}) F(M_{ay}) dk_x dk_y \end{aligned} \quad (57)$$

$[D_m]$ 、 $[E_m]$ は $1 \times N$ 、 $1 \times M$ の変換行列であり、 Y_a はスロットのアドミタンスである。

以上の3つの境界条件をまとめて、1つの行列で表現すると次のようになる。

$$\left[\begin{array}{c|c|c} [Z_{nm}^P] & [C_m^P] & [0] \\ \hline [D_m]^t & Y_a & -[E_m]^t \\ \hline [0] & -[C_{nm}^{ml}] & [Z_{nm}^{ml}] \end{array} \right] \left[\begin{array}{c} [A_m] \\ \hline V_a \\ \hline [I_m] \end{array} \right] = \left[\begin{array}{c} [0] \\ \hline 0 \\ \hline [V_m] \end{array} \right] \quad (58)$$

2.3 使用されるグリーン関数と展開関数について

まず、2.2節の定式化で用いられたグリーン関数を列挙する。これらの導出は1.3節で行われた導出例と同様に行うことによって得られる。本レポートでは各グリーン関数の導出は省略し、結果のみを挙げておくことにする。

$$G_{xx}^{EJ(I)} = -j \frac{Z_0 (\epsilon_{rs} k_0^2 - k_x^2) k_{z2} \cos k_{z1} d_s + (k_0^2 - k_x^2) k_{z1} \sin k_{z1} d_s}{T_e^{(I)} T_m^{(I)}} \sin k_{z1} d_s \quad (59)$$

$$G_{yx}^{HJ(I)} = -j \frac{(\epsilon_{rs} - 1) k_x^2 \sin k_{z1} d_s}{T_e^{(I)} T_m^{(I)}} + \frac{k_{z1}}{T_e^{(I)}} \quad (60)$$

$$k_{z1} = \sqrt{\epsilon_{rs} k_0^2 - k_x^2 - k_y^2}$$

$$k_{z2} = \sqrt{k_0^2 - k_x^2 - k_y^2}$$

$$T_e^{(I)} = k_{z1} \cos k_{z1} d_s + j k_{z2} \sin k_{z1} d_s$$

$$T_m^{(I)} = \epsilon_{rs} k_{z2} \cos k_{z1} d_s + j k_{z1} \sin k_{z1} d_s$$

$$G_{xx}^{EJ(II)} = -j \frac{Z_0 (\epsilon_{rp} k_0^2 - k_x^2) k_{z3} \cos k_{z3} d_p + (k_0^2 - k_x^2) k_{z3} \sin k_{z3} d_p}{T_e^{(II)} T_m^{(II)}} \sin k_{z3} d_p \quad (61)$$

$$\begin{aligned} G_{yx}^{EJ(II)} &= j \frac{Z_0 k_x k_y (k_{z2} \cos k_{z3} d_p + j k_{z3} \sin k_{z3} d_p)}{T_e^{(II)} T_m^{(II)}} \sin k_{z3} d_p \\ &= G_{xy}^{EJ(II)} \end{aligned} \quad (62)$$

$$G_{yy}^{EJ(II)} = -j \frac{Z_0 (\epsilon_{rp} k_0^2 - k_y^2) k_{z3} \cos k_{z3} d_p + (k_0^2 - k_y^2) k_{z3} \sin k_{z3} d_p}{T_e^{(II)} T_m^{(II)}} \sin k_{z3} d_p \quad (63)$$

$$G_{yx}^{HJ(II)} = -j \frac{(\epsilon_{rp} - 1)k_x^2 \sin k_{z3}d_p}{T_e^{(II)}T_m^{(II)}} + \frac{k_{z3}}{T_e^{(II)}} \quad (64)$$

$$G_{yy}^{HJ(II)} = -j \frac{(\epsilon_{rp} - 1)k_x k_y \sin k_{z3}d_p}{T_e^{(II)}T_m^{(II)}} \quad (65)$$

$$k_{z3} = \sqrt{\epsilon_{rp}k_0^2 - k_x^2 - k_y^2}$$

$$T_e^{(II)} = k_{z3} \cos k_{z3}d_p + jk_{z2} \sin k_{z3}d_p$$

$$T_m^{(II)} = \epsilon_{rp}k_{z2} \cos k_{z3}d_p + jk_{z3} \sin k_{z3}d_p$$

$$\begin{aligned} & G_{yy}^{HM(I)} + G_{yy}^{HM(II)} \\ &= -\frac{1}{k_0 Z_0} \left[\frac{(\epsilon_{rs} - k_y^2)(k_{z1} \cos k_{z1}d_s + j\epsilon_{rs}k_{z2} \sin k_{z1}d_s)}{k_{z1}T_m^{(I)}} - \frac{(\epsilon_{rs} - 1)k_{z1}k_y^2}{T_e^{(I)}T_m^{(I)}} \right] \\ & \quad - \frac{1}{k_0 Z_0} \left[\frac{(\epsilon_{rp} - k_y^2)(k_{z3} \cos k_{z3}d_p + j\epsilon_{rp}k_{z2} \sin k_{z3}d_p)}{k_{z3}T_m^{(II)}} - \frac{(\epsilon_{rp} - 1)k_{z3}k_y^2}{T_e^{(II)}T_m^{(II)}} \right] \quad (66) \end{aligned}$$

次にマイクロストリップ線路上、スロット上、円形マイクロストリップアンテナ上の展開関数について述べる。第一にマイクロストリップ線路上の展開関数について述べる。マイクロストリップ線路は進行波型関数で展開する場合 [7]、区分正弦波関数 (PWS モード) で展開する場合 [8] の 2 つが考えられる。進行波型関数では行列が小さくて済むという長所をもつものの、マイクロストリップ線路に関するインピーダンス行列のすべての成分について計算を行わなければならないという欠点をもつ。一方、PWS モードの場合には全体の行列はおおきくなるという欠点をもつが、その性質上

$$Z_{nm} = Z_{n'm'} \quad \text{if} \quad |n - m| = |n' - m'|$$

という等式が成り立つ。そのため、 $M \times M$ の行列のうち、 Z_{1m} または Z_{n1} のみを計算すれば、行列のすべての成分が計算されることになり、時間的には短縮できる。また、この方法を採用することにより励振源が簡単になる。そのため本レポートにおいては PWS モードを用いた解析を行っている。また、電流は X 方向のみに流れるものと仮定する。この PWS モードを数式で表すと、

$$\begin{aligned} J_{xn} &= \hat{x} \frac{1}{w_s} \frac{\sin k_e(h_s - |x - x_n|)}{\sin k_e h_s} \\ & \quad |x - x_n| < h_s \\ & \quad |y - y_n| < w_s/2 \end{aligned} \quad (67)$$

となる。ここで、

k_e : マイクロストリップ線路の伝搬定数 (quasi-TEM)

h_s : セグメント長

w_s : マイクロストリップ線路幅

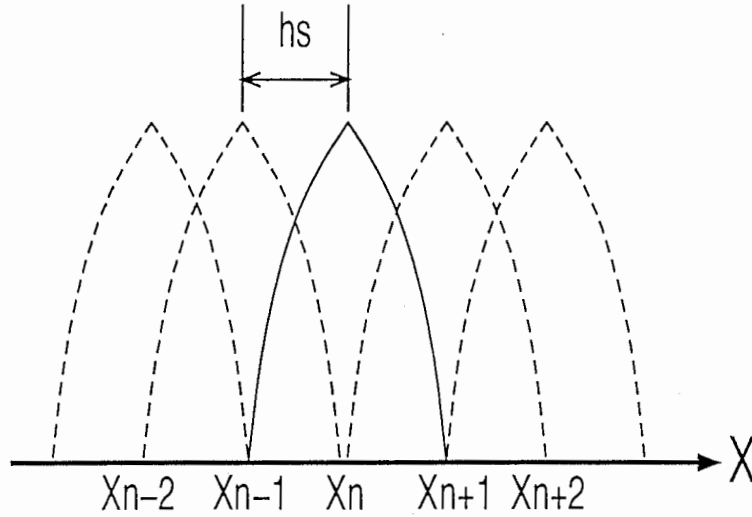


図6：PWS モード

である。また、この関数を Fourier 変換すると、

$$F(J_{xn}) = \frac{\sin \frac{k_y w_s}{2}}{\frac{k_y w_s}{2}} \frac{2k_e [\cos k_x h_s - \cos k_e h_s]}{\sin k_e h_s (k_e^2 - k_x^2)} e^{-jk_x x_n - jk_y y_n} \quad (68)$$

スロット上の磁流についても PWS モードで展開する。[9] にも述べられているように、スロットは放射素子として動作しているわけではなく結合素子として用いられているために、その長さは半波長よりも十分に小さい。そのために 1 PWS モードのみで表すと、

$$M_{ay} = \hat{y} \frac{1}{w_a} \frac{\sin k_a (h_a - |y - y_a|)}{\sin k_a h_a} \quad (69)$$

$$|y - y_a| \leq h_a$$

$$|x - x_a| \leq w_a/2$$

k_a : スロット上の波数 $\left(= \sqrt{\frac{\epsilon_{rs} + \epsilon_{rp}}{2}} \cdot k_0 \right)$

h_a : スロット長の 1 / 2

w_a : スロット幅

この関数の Fourier 変換は次のようになる。

$$F(M_{ay}) = \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} e^{-jk_x x_a - jk_y y_a} \quad (70)$$

最後に円形マイクロストリップアンテナ上を流れる電流について考察する。円形ディスク上の境界条件はエッジでの径方向成分がゼロになることである。円形マイクロストリップアンテナのキャビティモデルからの類推により以下に示す関数系での展開を行う。この関数系は参考文献 [10] から引用した。

$$\bar{J}^p(\rho, \phi) = \sum_{n,m} \left[\hat{\rho} \beta_{mn} J'_m(\beta_{mn} \rho) \cos m\phi - \hat{\phi} \frac{m}{\rho} J_m(\beta_{mn} \rho) \sin m\phi \right] \quad (71)$$

$$\bar{J}^o(\rho, \phi) = \sum_{n,m} \left[\hat{\rho} \beta_{mn} J'_m(\beta_{mn} \rho) \sin m\phi + \hat{\phi} \frac{m}{\rho} J_m(\beta_{mn} \rho) \cos m\phi \right] \quad (72)$$

$$J'_m(\beta_{mn} R_p) = \frac{\partial J_m}{\partial \rho} \Big|_{\rho=R_p} = 0$$

$J_m(\cdot)$: m 次第 1 種 Bessel 関数

R_p : 円形ディスクの半径

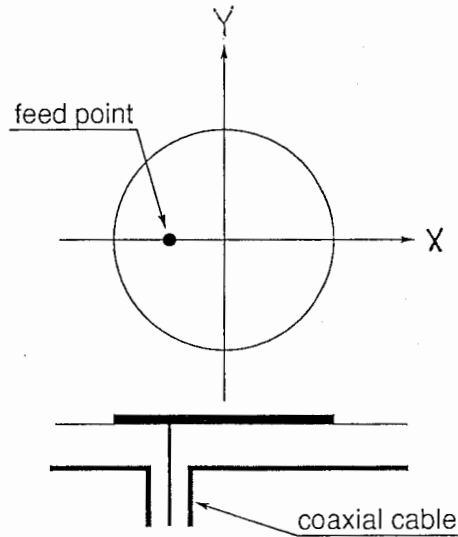


図 7 : 円形ディスク上の展開関数

ここで2つの関数系を用いているが、 \tilde{J}^p は図7に示すようにX軸上で同軸給電した際に励振されるモードに対応し、便宜上本レポートでは、 TM_{mn}^p モードと呼ぶことにする。また、 \tilde{J}^o は \tilde{J}^p に直交するモードであり、 TM_{mn}^o モードと呼ぶことにする。また、これらのモードのFourier変換は次のように表されることになる。

$$\begin{aligned}
F(\tilde{J}_{mn}^p) &= 2\pi j^{-m+1} J_m(\beta_{mn} R_p) \\
&\times \left[\hat{x} \left(\frac{\beta_{mn}^2}{\beta_{mn}^2 - \beta^2} J'_m(\beta R_p) \cos m\alpha \cos \alpha + \frac{m J_m(\beta R_p)}{\beta R_p} \sin m\alpha \sin \alpha \right) \right. \\
&\quad \left. - \hat{y} \left(\frac{\beta_{mn}^2}{\beta_{mn}^2 - \beta^2} J'_m(\beta R_p) \cos m\alpha \sin \alpha - \frac{m J_m(\beta R_p)}{\beta R_p} \sin m\alpha \cos \alpha \right) \right]
\end{aligned} \tag{73}$$

$$\begin{aligned}
F(\tilde{J}_{mn}^o) &= 2\pi j^{-m+1} J_m(\beta_{mn} R_p) \\
&\times \left[\hat{x} \left(\frac{\beta_{mn}^2}{\beta_{mn}^2 - \beta^2} J'_m(\beta R_p) \sin m\alpha \cos \alpha - \frac{m J_m(\beta R_p)}{\beta R_p} \cos m\alpha \sin \alpha \right) \right. \\
&\quad \left. + \hat{y} \left(\frac{\beta_{mn}^2}{\beta_{mn}^2 - \beta^2} J'_m(\beta R_p) \sin m\alpha \sin \alpha + \frac{m J_m(\beta R_p)}{\beta R_p} \cos m\alpha \cos \alpha \right) \right]
\end{aligned} \tag{74}$$

また、前節で得られた行列の要素のうち、グリーン関数と展開関数の性質より、

$$[C_m^p] = -[D_m^p] \tag{75}$$

$$[E_m] = -[C_m^{ml}] \tag{76}$$

が成り立つ。この証明は省略させてもらうが、この関係式が成り立つのは電磁界の双対性による。

2.4 励振方法とSパラメータの導出 [8]

マイクロストリップ線路上で電流分布をPWSモードで展開したために、電圧源としてはデルタ関数を用いる。これは一般的に用いられるモーメント法のデルタギャップ励振に対応している。この励振電圧源はアンテナあるいはスロットの影響がないように、これらから十分離れたところに置くものとする。

その結果得られる電流値より、電流の最大値 I_{max} 、最小値 I_{min} 、及びそれらの座標 $x_{max}x_{min}$ が分かる。よって、これらより、反射係数 Γ 、入力インピーダンス Z_{in} は

次のように求まる。

$$VSWR = \frac{|I_{max}|}{|I_{min}|} \quad (77)$$

$$|\Gamma| = \frac{VSWR - 1}{VSWR + 1} \quad (78)$$

$$\arg \Gamma = -2k_e x_{min} \quad (79)$$

$$Z_{in} = \frac{1 + \Gamma}{1 - \Gamma} \quad (80)$$

2.5 数値積分

(58) 式の各行列要素は (42)(43)(49)(55)(56)(57) 各式に示す通り、2重の無限積分で定義されている。そこで、以下に示すように座標変換を行うと、

$$k_x = \beta \cos \alpha$$

$$k_y = \beta \sin \alpha$$

$$\beta = \sqrt{k_x^2 + k_y^2}$$

$$\alpha = \tan^{-1} \left(\frac{k_y}{k_x} \right)$$

$$\frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\cdot) dk_x dk_y = \frac{1}{4\pi^2} \int_0^{\infty} \int_0^{2\pi} (\cdot) \beta d\beta d\alpha \quad (81)$$

となる。また α による対称性を用いることにより、解析例の場合 には、

$$\begin{aligned} [*]_{nm} &= \frac{1}{4\pi^2} \int_0^{\infty} \int_0^{2\pi} (\cdot) \beta d\beta d\alpha \\ &= \frac{1}{\pi^2} \int_0^{\infty} \int_0^{\pi/2} (\cdot) \beta d\beta d\alpha \end{aligned} \quad (82)$$

と、第一象限のみの積分で十分になる。この導出は簡単なのでここでは省略する。ここで注意して欲しいのは、すべてのアンテナに対してこの対称性により積分が第一象限に限定されるわけでない。場合によっては第一象限と第二象限の積分、あるいは四象限すべての積分が必要な場合がある。それは個々について、グリーン関数と展開関数の性質をきちんとみる必要がある。

一方、 β 方向への積分は大きく分けて4つの領域に分離することができる。

$$\begin{aligned}
[*]_{nm} &= \frac{1}{\pi^2} \int_0^\infty \int_0^{\pi/2} (\cdot) \beta d\beta d\alpha \\
&= \frac{1}{\pi^2} \int_0^{k_0} \int_0^{\pi/2} (\cdot) \beta d\beta d\alpha \\
&\quad + \frac{1}{\pi^2} \int_{k_0}^{\varepsilon_r k_0|_{\max}} \int_0^{\pi/2} (\cdot) \beta d\beta d\alpha \\
&\quad + \frac{1}{\pi^2} \int_{\varepsilon_r k_0|_{\max}}^\infty \int_0^{\pi/2} (\cdot) \beta d\beta d\alpha \\
&\quad + \sum \frac{1}{\pi^2} \int_0^{\pi/2} (\cdot) |_{\beta=\text{pole}} \beta d\alpha
\end{aligned} \tag{83}$$

ここで、1つ目は放射に関する積分、2つ目、3つ目はリアクタンスに関する積分、4つ目はグリーン関数の極、つまり表面波モードによる損失に関する積分である。また各領域において、

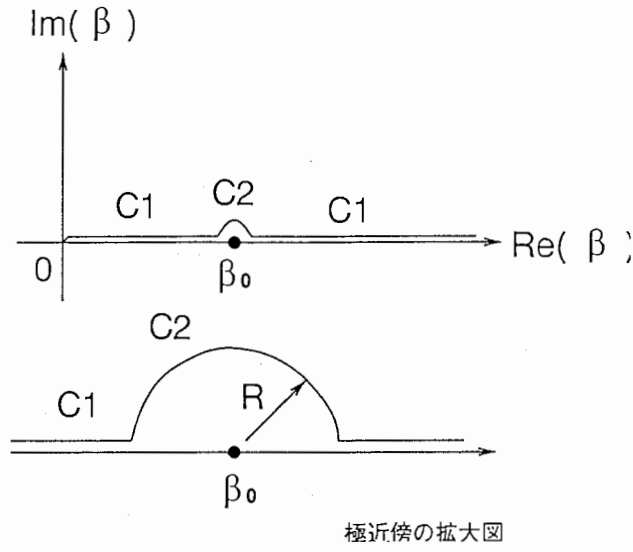
$$\begin{aligned}
k_{z1} &= \sqrt{\varepsilon_{rs} k_0^2 - \beta^2} \quad \text{Region I \& II} \\
&= -j\sqrt{\beta^2 - \varepsilon_{rs} k_0^2} \quad \text{Region III}
\end{aligned} \tag{84}$$

$$\begin{aligned}
k_{z2} &= \sqrt{k_0^2 - \beta^2} \quad \text{Region I} \\
&= -j\sqrt{\beta^2 - k_0^2} \quad \text{Region II \& III}
\end{aligned} \tag{85}$$

となる。ここで、マイナス符号は放射条件を満足するようにつけられている。

次に極の取り扱いについて述べる。(42)(43)(49)(55)(56)(57) 各式に示す通り、各行列要素には必ず TE または TM のモード特性方程式が分母に來ている。そのため、特性方程式の解が被積分関数の極になる。そのため、積分ではこの極の取り扱いに注意をしなければならない。一般にアンテナで用いる基板は使用する波長に比較して非常に薄い基板を用いるために、TE モードについては普通はカットオフ以下であり問題ない(しかしながら確かめる必要はある)。一方 TM モードについては高次モードについては普通はカットオフ以下なので問題とはならない。しかしながら、 TM_0 モードはカットオフがないために常に存在する。このモードがフェーズドアレーアンテナのスキャンブラインドネスの原因となる。

ここで、TM モード、TE モードの特性方程式であるが、それぞれ β の関数であり、 α には無関係である。よって、極の扱いは β についてのみでよいことになる。



極近傍の拡大図

図 8 : 積分路の説明

この極の計算は次のように行う。 TM_0 モードの極が $\beta = \beta_{m0}$ に存在すると仮定する。そうすると、積分路 C_1 は $\beta = \beta_{m0}$ 以外では実軸上を通り、 $\beta = \beta_{m0}$ 付近では極を避けるように極から R だけ離れた積分路 C_2 を通るものとする。この C_2 について、複素関数の積分に関する定理:

定理: 点 $\beta = \beta_0$ が一位の極である時、積分路を

$$C_r : \beta = \beta_{m0} + Re^{j\theta}$$

$$\theta_1 \leq \theta \leq \theta_2, 0 \leq \theta_1 - \theta_2 \leq 2\pi, R \geq 0$$

とすれば、

$$\lim_{R \rightarrow 0} \int_{C_r} f(\beta) d\beta = j(\theta_1 - \theta_2) \text{Res} f(\beta) \quad (86)$$

となる。

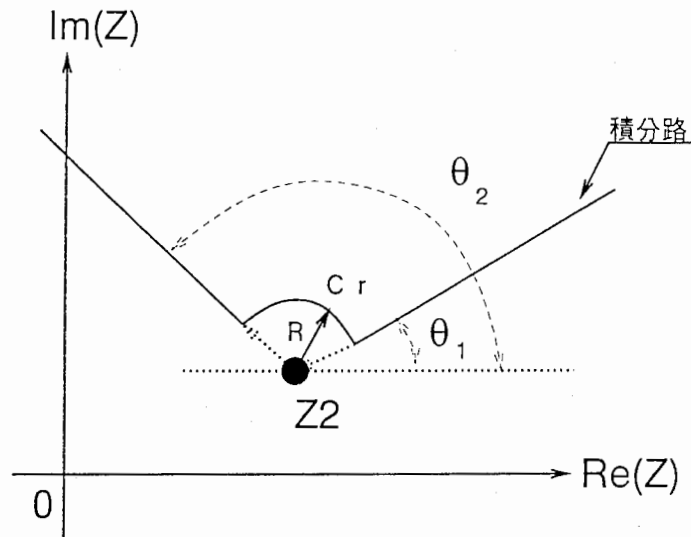


図9：複素積分の定理の説明

ここで、 $\text{Res}f(\beta)$ は $f(\beta)$ の留数であり、次のように定義される。

留数の定義： z_0 の近傍で z_0 自身を除く領域で成立するローラン展開を

$$f(z) = \sum_{-\infty}^{\infty} A_n (z - z_0)^n \quad (87)$$

とする。そのうち、 $(z - z_0)^{-1}$ の係数を留数という。

上の定義は数学的に厳密な定義であるが、実際には

$$f(z) = \frac{f_1(z)}{f_2(z)}$$

で、 $f_2(z)$ の解 z_2 のまわりでテイラー展開すると、

$$f_2(z) = \sum_n \frac{f_2^n(z_2)}{n!} (z - z_2)^n \quad (88)$$

となる。ここで、

$$f_2^n = \frac{\partial^n f_2}{\partial z^n}$$

である。よって、この場合の留数は

$$\text{Res}f(z) = \frac{f_1(z_2)}{\frac{\partial f_2}{\partial z} \Big|_{z=z_2}} \quad (89)$$

よって、(83) 式の最後の積分は積分定理 (86) 式より、

$$\frac{1}{\pi^2} \int_0^{\pi/2} (\cdot)|_{\beta=\beta_{m0}} \beta d\alpha = -j \frac{1}{\pi} \int_0^{\pi/2} \text{Res}(\cdot)|_{\beta=\beta_{m0}} \beta d\alpha \quad (90)$$

2.6 励振モードとスロットの関係

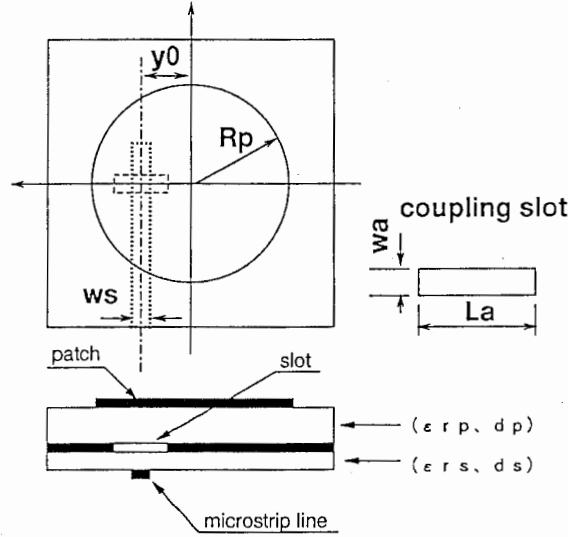


図 10 : Y オフセットした円形パッチ

図 10 に示しているようにスロットを Y 軸に沿ってオフセットした際の円形パッチ上のモードとの関係を求める。まずは Y 方向にオフセットしたスロットと TM_{1n}^p モードとの関係を求める。各展開関数の Fourier 変換は以下のように表される。

$$F(J_{1x}^p) = 2\pi R_p J_1(\beta_{1n} R_p) \left[\frac{\beta_{1n}^2}{\beta_{1n}^2 - \beta^2} J_1'(\beta R_p) \cos^2 \alpha + \frac{J_1(\beta R_p)}{\beta R_p} \sin^2 \alpha \right] \quad (91)$$

$$F(J_{1y}^p) = 2\pi R_p J_1(\beta_{1n} R_p) \left[\frac{\beta_{1n}^2}{\beta_{1n}^2 - \beta^2} J_1'(\beta R_p) - \frac{J_1(\beta R_p)}{\beta R_p} \right] \sin \alpha \cos \alpha \quad (92)$$

$$F(M_{ay}) = \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} e^{-jk_y y_0} \quad (93)$$

よって、

$$\begin{aligned}
& [F(J_{1x}^p)G_{xy}^{EM(II)}F(M_{ay})] \\
&= \frac{1}{4\pi^2} \int_0^\infty \int_0^{2\pi} 2\pi R_p J_1(\beta_{1n} R_p) \left[\frac{\beta_{1n}^2}{\beta_{1n}^2 - \beta^2} J_1'(\beta R_p) \cos^2 \alpha + \frac{J_1(\beta R_p) \sin^2 \alpha}{\beta R_p} \right] \\
&\quad \times \left[\frac{j k_x^2 (\varepsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \right] \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} e^{-j k_y y_0} \beta d \beta d \alpha \\
&= \frac{1}{\pi^2} \int_0^\infty \int_0^{\pi/2} 2\pi R_p J_1(\beta_{1n} R_p) \left[\frac{\beta_{1n}^2}{\beta_{1n}^2 - \beta^2} J_1'(\beta R_p) \cos^2 \alpha + \frac{J_1(\beta R_p) \sin^2 \alpha}{\beta R_p} \right] \\
&\quad \times \left[\frac{j k_x^2 (\varepsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \right] \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} \cos(k_y y_0) \beta d \beta d \alpha \quad (94)
\end{aligned}$$

$$\begin{aligned}
& [F(J_{1y}^p)G_{xy}^{EM(II)}F(M_{ay})] \\
&= \frac{1}{4\pi^2} \int_0^\infty \int_0^{2\pi} 2\pi R_p J_1(\beta_{1n} R_p) \left[\frac{\beta_{1n}^2}{\beta_{1n}^2 - \beta^2} J_1'(\beta R_p) + \frac{J_1(\beta R_p)}{\beta R_p} \right] \sin \alpha \cos \alpha \\
&\quad \times \frac{j k_x k_y (\varepsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} e^{-j k_y y_0} \beta d \beta d \alpha \\
&= \frac{1}{\pi^2} \int_0^\infty \int_0^{\pi/2} 2\pi R_p J_1(\beta_{1n} R_p) \left[\frac{\beta_{1n}^2}{\beta_{1n}^2 - \beta^2} J_1'(\beta R_p) + \frac{J_1(\beta R_p)}{\beta R_p} \right] \sin \alpha \cos \alpha \\
&\quad \times \frac{j k_x k_y (\varepsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} \cos(k_y y_0) \beta d \beta d \alpha \quad (95)
\end{aligned}$$

となる。よって、 $y_0 = 0$ 、つまりスロットがパッチ直下にあるときに結合が一番強くなることがわかる。

次にセカンドモードについて考える。まず、 TM_{2n}^p モードとの関係を求める。

$$F(J_{2x}^p) = -j 2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \cos 2\alpha \cos \alpha + \frac{2J_2(\beta R_p)}{\beta R_p} \sin 2\alpha \sin \alpha \right] \quad (96)$$

$$F(J_{2y}^p) = -j 2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \sin \alpha \cos 2\alpha - \frac{2J_2(\beta R_p)}{\beta R_p} \sin 2\alpha \cos \alpha \right] \quad (97)$$

$$[F(J_{2y}^p)G_{xy}^{EM(II)}F(M_{ay})]$$

$$\begin{aligned}
&= \frac{1}{4\pi^2} \int_0^\infty \int_0^{2\pi} -j2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \cos 2\alpha \cos \alpha + \frac{2J_2(\beta R_p)}{\beta R_p} \sin 2\alpha \sin \alpha \right] \\
&\quad \times \left[\frac{jk_x^2(\varepsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \right] \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} e^{-jk_y y_0} \beta d\beta d\alpha \\
&= 0
\end{aligned} \tag{98}$$

$$\begin{aligned}
&[F(J_{2y}^p) G_{xy}^{EM(II)} F(M_{ay})] \\
&= \frac{1}{4\pi^2} \int_0^\infty \int_0^{2\pi} -j2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \sin \alpha \cos 2\alpha - \frac{2J_2(\beta R_p)}{\beta R_p} \sin 2\alpha \cos \alpha \right] \\
&\quad \times \left[\frac{jk_x^2(\varepsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \right] \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} e^{-jk_y y_0} \beta d\beta d\alpha \\
&= 0
\end{aligned} \tag{99}$$

この結果より、 TM_{2n}^p モードは励振されないことになる。一方 TM_{2n}^o モードについては、

$$F(J_{2x}^o) = -j2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \sin 2\alpha \cos \alpha - \frac{2J_2(\beta R_p)}{\beta R_p} \cos 2\alpha \sin \alpha \right] \tag{100}$$

$$F(J_{2y}^o) = -j2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \sin \alpha \sin 2\alpha + \frac{2J_2(\beta R_p)}{\beta R_p} \cos 2\alpha \cos \alpha \right] \tag{101}$$

$$\begin{aligned}
&[F(J_{2x}^o) G_{xy}^{EM(II)} F(M_{ay})] \\
&= \frac{1}{4\pi^2} \int_0^\infty \int_0^{2\pi} -j2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \sin 2\alpha \cos \alpha - \frac{2J_2(\beta R_p)}{\beta R_p} \cos 2\alpha \sin \alpha \right] \\
&\quad \times \left[\frac{jk_x^2(\varepsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \right] \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} e^{-jk_y y_0} \beta d\beta d\alpha \\
&= \frac{1}{\pi^2} \int_0^\infty \int_0^{\pi/2} -j2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \sin 2\alpha \cos \alpha - \frac{2J_2(\beta R_p)}{\beta R_p} \cos 2\alpha \sin \alpha \right] \\
&\quad \times \left[\frac{jk_x^2(\varepsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \right] \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} \sin(k_y y_0) \beta d\beta d\alpha
\end{aligned} \tag{102}$$

$$\begin{aligned}
& [F(J_{2y}^o)G_{xy}^{EM(II)}F(M_{ay})] \\
&= \frac{1}{4\pi^2} \int_0^\infty \int_0^{2\pi} -j2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \sin \alpha \sin 2\alpha + \frac{2J_2(\beta R_p)}{\beta R_p} \cos 2\alpha \cos \alpha \right] \\
&\quad \times \left[\frac{jk_x^2(\epsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \right] \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} e^{-jk_y y_0} \beta d \beta d \alpha \\
&= \frac{1}{\pi^2} \int_0^\infty \int_0^{\pi/2} -j2\pi R_p J_2(\beta_{2n} R_p) \left[\frac{\beta_{2n}^2}{\beta_{2n}^2 - \beta^2} J_2'(\beta R_p) \sin \alpha \sin 2\alpha + \frac{2J_2(\beta R_p)}{\beta R_p} \cos 2\alpha \cos \alpha \right] \\
&\quad \times \left[\frac{jk_x^2(\epsilon_{rp} - 1) \sin k_{z1} d_p}{T_e^{(II)} T_m^{(II)}} \right] \frac{\sin \frac{k_x w_a}{2}}{\frac{k_x w_a}{2}} \frac{2k_a [\cos k_y h_a - \cos k_a h_a]}{\sin k_a h_a (k_a^2 - k_y^2)} \sin(k_y y_0) \beta d \beta d \alpha \quad (103)
\end{aligned}$$

よって TM_{2n}^o モードは、スロットが円形マイクロストリップアンテナの中心にあるときには励振されず、スロットがY軸に沿ってオフセットされることにより発生することがこの数式から得られる。

2.7 計算結果

次に実際の計算結果について述べる。計算に用いたY軸に沿ったオフセット結合マイクロストリップアンテナの各パラメータを表1に示す。また、円形ディスク上の展開関数としては解の収束性をみながら、 TM_{11}^p 、 TM_{12}^p 、 TM_{31}^p 、 TM_{21}^o 、 TM_{41}^o の5つのモードで展開している。スロット上は1つのPWSモード、マイクロストリップ線路上は解析領域として開放スタブを含めて1管内波長とし、その中を40個のPWSモードで展開している。よって各PWSモードの間隔 h_s は $\lambda_g/40$ としている。いずれも解の収束性をみながら決定している。

誘電体の比誘電率 (給電線路)	$\epsilon_{rs} = 2.60$
誘電体の厚さ (給電線路)	$d_s = 0.78 \text{ mm}$
誘電体の比誘電率 (アンテナ)	$\epsilon_{rs} = 2.60$
誘電体の厚さ (アンテナ)	$d_p = 3.16 \text{ mm}$
マイクロストリップ幅	$w_s = 2.24 \text{ mm} (Z_m = 50\Omega)$
スロット長	$L_a = 20.0 \text{ mm}$
スロット幅	$w_a = 1.5 \text{ mm}$
スロットオフセット	$y_0 = 20.0 \text{ mm}$
円形ディスク半径	$R_p = 32.0 \text{ mm}$
開放スタブ長	18.0 mm

表1：Yオフセット円形パッチのパラメータ

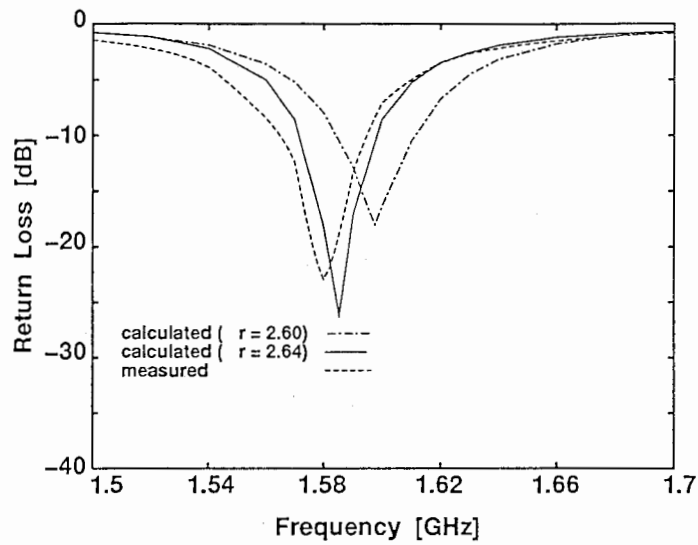


図 1.1 : 計算値と測定値の比較 (反射損失)

図 1.1 に計算した反射損失と実測値との比較を行っている。図中、計算値が 2 つ出ているが、そのうち一点鎖線が誘電体の比誘電率をカタログ値 $\epsilon_{rp} = 2.60$ にした場合の計算値、実線は $\epsilon_{rp} = 2.64$ とした場合の計算値である。 $\epsilon_{rp} = 2.60$ の計算値では実測値との間で周波数シフトがみられるが、 $\epsilon_{rp} = 2.64$ とすることによりこの周波数シフトがなくなっている。この原因のひとつとしては、実際に使用した誘電体の比誘電率がカタログ値と異なることが考えられる。

また、付録に解析に用いた Fortran プログラムを添付したので、具体的な計算のやり方はそちらを参照されたい。

3 解析例 II(X 軸オフセット MSA)

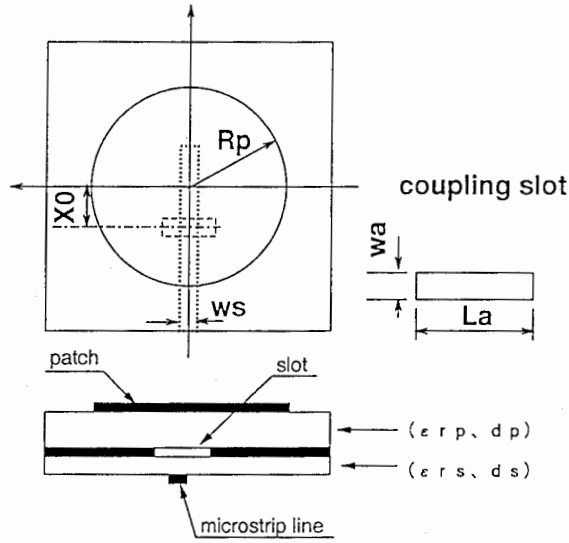


図 1 2 : X オフセットした円形パッチ

次にスロットを x 軸に沿ってオフセットした場合 (図 1 2) について、結果だけを述べる。定式化は 2 章で述べたものとはほぼ同一であり、唯一異なる点は円形ディスク上に励振されるモードが異なることである [12][13]。2 章の 2. 6 節と同様の計算を行うと励振されるモードは、

$$TM_{1n}^p, TM_{3n}^p, \dots, TM_{(2N-1)n}^p, \dots, \\ TM_{0n}^p, TM_{2n}^p, \dots, TM_{(2N)n}^p, \dots,$$

となる。奇数次モードは y 軸オフセットの場合と同様であるが、偶数次モードが y 軸オフセットの場合のモードに直交するモードが励振されることになる。

解析及び実験に用いたアンテナのパラメータを表 2 に示す。ここでも前節同様 $\epsilon_{rp} = 2.60 \rightarrow 2.62$ としている。図 1 3 にその結果を示す。両者はほぼ一致していることがわかる。前章と本章の結果より、プログラムの有効性が示された。

誘電体の比誘電率 (給電線路)	$\epsilon_{rs} = 2.60$
誘電体の厚さ (給電線路)	$d_s = 0.78 \text{ mm}$
誘電体の比誘電率 (アンテナ)	$\epsilon_{rs} = 2.60$
誘電体の厚さ (アンテナ)	$d_p = 3.16 \text{ mm}$
マイクロストリップ幅	$w_s = 2.24 \text{ mm} (Z_m = 50 \Omega)$
スロット長	$L_a = 25.0 \text{ mm}$
スロット幅	$w_a = 1.5 \text{ mm}$
スロットオフセット	$x_0 = -20.0 \text{ mm}$
円形ディスク半径	$R_p = 32.0 \text{ mm}$
開放スタブ長	24.5 mm

表 2 : X オフセット円形パッチのパラメータ

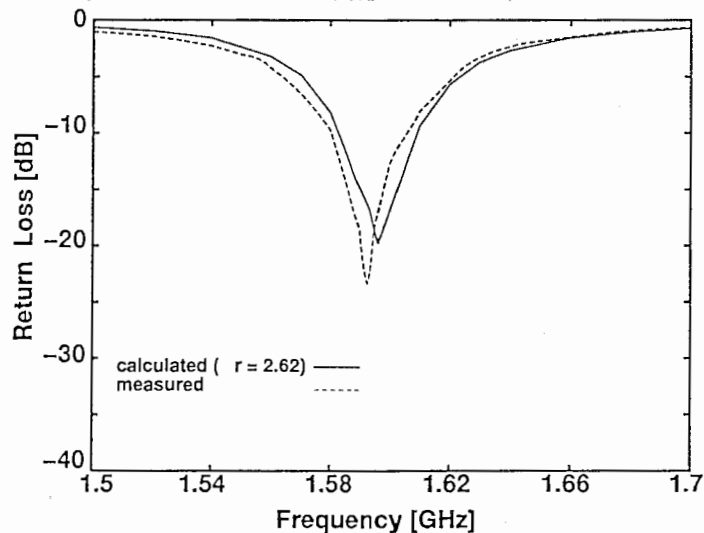


図 1 3 : 計算値と測定値の比較 (反射損失、X-offset)

4 まとめ

スロット結合マイクロストリップアンテナの解析としてスペクトル領域法について、定式化を中心として述べた。また簡単な計算例として、x 軸及び y 軸にスロットをオフセットした場合の特性について計算を行い、実測値との比較を行った。その結果、両者とも良好な結果を得た。本レポートにおいて示した定式化は一番簡単ではあるが基本となるものであり、この定式化を応用して直交 2 点給電スロット結合マイクロストリップアンテナ [12][13] などへの拡張も容易である。また今回は定式化及び計算を行わなかったが、スペクトル領域法は無限アレーアンテナの解析にも適用することができる。無限アレーへの拡張も数学的には (無限積分) \rightarrow (無限級数) となるだけであり、数値計算自体も簡単になる。無限アレーアンテナの解析に関する参考文献 [5][6][14][15] も挙げておくので、興味があった場合には参照されたい。

5 謝辞

本研究を行うにあたりご指導・御助言戴いた猪股社長、唐沢室長はじめ無線通信第一研究室の各位に深謝致します。

[参考文献]

参考文献

- [1] 伊藤龍雄, “11章 スペクトル領域法” 電磁波問題の基礎解析法, 電子情報通信学会編、1987.10.
- [2] A. K. Bhattacharyya, “Electromagnetic Fields in Multilayered Structures - Theory and Applications-” Artech House, 1994.
- [3] N. K. Das and D. M. Pozar, “A Generalized Spectral-Domain Green's Function for Multilayer Dielectric Substrates with Application to Multilayer Transmission Lines” IEEE Trans. Microwave Theory Tech., Vol. MTT-35, pp.326-335, Mar. 1987.
- [4] R. W. Jackson and D. M. Pozar, “Full-Wave Analysis of Microstrip Open-End and Gap Discontinuities” IEEE Trans. Microwave Theory Tech., Vol. MTT-33, pp.1036-1042, Oct. 1985.
- [5] D. M. Pozar and D. H. Schaubert, “Scan Blindness in Infinite Phased Arrays of Printed Dipoles” IEEE Trans. Antennas Propagat., Vol. AP-32, pp.602-610, Jun. 1984.
- [6] D. M. Pozar, “Analysis of an Infinite Phased Array of Aperture Coupled Microstrip Patches” IEEE Trans. Antennas Propagat., Vol. AP-37, pp.418-425, Apr. 1989.
- [7] D. H. Schaubert, “Analysis of an Aperture Coupled Microstrip Antenna” IEEE Trans. Antennas Propagat., Vol. AP-34, pp.977-984, Aug. 1986.

- [8] M. Davidovitz and Y. T. Lo, "Rigorous Analysis of a Circular Patch Antenna Excited by a Microstrip Transmission Line" IEEE Trans. Antennas Propagat., Vol. AP-37, pp.949-958, Aug. 1989.
- [9] D. M. Pozar, "Reciprocity Method of Analysis of Printed Slots and Slot Coupled Microstrip Antennas" IEEE Trans. Antennas Propagat., Vol. AP-34, pp.1439-1446, Apr. 1987.
- [10] J. T. Aberle and D. M. Pozar, "Analysis of Infinite Arrays of One and Two-Probe-fed circular Patches" IEEE Trans. Antennas Propagat., Vol. AP-38, pp.421-432, Apr. 1990.
- [11] 廣池, 守田, 田中, "応用解析学" 共立出版、1982.
- [12] Y. Murakami, W. Chujo, I. Chiba, and M. Fujise, "A Study on Mutual Coupling between Two Ports of Dual Slot-Coupled Microstrip Antennas" IEICE Trans. Commun. Vol. E77-B, pp.815-822, June 1994.
- [13] 村上, "スロット結合型円形マイクロストリップアンテナの端子間相互結合特性" ATR テクニカルレポート、TR-O-0067.
- [14] N. Das and D. M. Pozar, "Multiport Scattering Analysis of General Multilayered Printed Antennas Fed by Multiple Feed Ports: Part I - Theory-" IEEE Trans. Antennas Propagat. Vol. AP-40, pp.469-481, May 1992.
- [15] N. Das and D. M. Pozar, "Multiport Scattering Analysis of General Multilayered Printed Antennas Fed by Multiple Feed Ports: Part II - Applications-" IEEE Trans. Antennas Propagat. Vol. AP-40, pp.482-491, May 1992.

付録：

Y オフセットスロット結合マイクロストリップアンテナの解析プログラム

Fortran source プログラムとして、

- ・ y o f f _ l p . f
- ・ b e s j t . f
- ・ t l e d c . f

を添付する。このうち、y o f f _ l p . f は、

メインプログラム

サブルーチン swave：表面波 TM_0 モードの極を求めるサブルーチン
(Newton 法を用いている)

サブルーチン expans1：円形パッチ上の TM_{1n}^p 、 TM_1^o を求めるサブルーチン

サブルーチン expans2：円形パッチ上の TM_{2n}^p 、 TM_2^o を求めるサブルーチン

サブルーチン expans3：円形パッチ上の TM_{3n}^p 、 TM_3^o を求めるサブルーチン

サブルーチン expans4：円形パッチ上の TM_{4n}^p 、 TM_4^o を求めるサブルーチン

から構成される。一方、besjt.f、tledec.f はそれぞれ Bessel 関数を求めるサブルーチン、複素数の連立一次方程式を消去法を用いて解法するサブルーチンであり、それぞれ自作したものである。これらのサブルーチンは IMSL などの既存のサブルーチンで置き換えてもらっても構わない。

メインプログラムについて、多少述べておくと、このプログラムはきちんと階層構造になるように組まれていない。どちらかという、手計算の手法をそのまま計算機に持ち込んだものであり、そのためメインプログラムが非常に長くなっており、読みにくくなっていることは確かである。また、数値積分でも一番単純な 0 次近似つまり、

$$(\text{積分}) = \Sigma (\text{関数値}) \times (\text{面積})$$

という近似をしている。これを一次あるいは高次の近似を用いて計算の精度を上げることが可能である。しかしながら、これには計算機のパワーが必要であり、また時間が 0 次近似に比べて相当掛かるにもかかわらずあまり精度がよくならなかったのも、本レポートではやっていない。

これが、現段階でのこのプログラムが解決すべき課題である。本解析に興味を持たれ、より見やすいあるいはより精度の高いプログラムにして戴ければ幸いである。

```

1 c ##### FILE_NAME YOFF_LP.F #####
2 c =====
3 c = PROGRAM for MICROSTRIP LINE FED SLOT COUPLED =
4 c = CIRCULAR PATCH ANTENNA. =
5 c = using SPECTRUM DOMAIN MOMENT METHOD =
6 c = (epsrs > epsrp) =
7 c = yoffset case =
8 c = -----
9 c = programmed by | Yasushi Murakami | =
10 c = -----
11 c = first ed. : 1992. 4. 23. =
12 c =====
13 c ++++++
14 c + epsrs : relative permittivity of feed substrate +
15 c + epsrp : relative permittivity of patch substrate +
16 c + ds : feed side substrate thickness +
17 c + dp : patch side substrate thickness +
18 c + wa : slot width +
19 c + ala : slot length +
20 c + rad : circular patch radius +
21 c ++++++
22 c -----
23 c - ak0 : wave number in free space -
24 c - aks : wave number in the feed side substrate -
25 c - akr : wave number in the patch side substrate -
26 c - ake : effective wave number in the line -
27 c - aka : effective wave number in the aperture -
28 c - z0 : characteristic impedance in free space -
29 c - ha : segment length of aperture -
30 c -----
31 c
32 program msc
33 parameter (nn=49,n=6,na=1,ns=40)
34 implicit double complex (c)
35 implicit double precision (a,b,d-h,o-y)
36 character mode(n)*6
37 dimension ca(nn,nn),cb(nn),cc(nn),zbes(0:10)
38 c
39 data mode/'TM11p:', 'TM12p:', 'TM31p:',
40 & 'TM21o:', 'TM41o:', 'Vx' :'/
41 c
42 do i=1,nn
43 do j=1,nn
44 ca(i,j)=(0.0d0,0.0d0)
45 enddo
46 enddo
47 do i=1,nn
48 cb(i)=(0.0d0,0.0d0)
49 cc(i)=(0.0d0,0.0d0)
50 enddo
51 c
52 pi=2.0d0*asin(1.0) ! pi=3.1415926
53 ac0=0.2998d0 ! light velocity
54 eps=1.0d-15
55 cj=(0.0d0,1.0d0) ! complex unit
56 mm=ns+n
57 c
58 epsrp=2.60d0 !dielctric constant(line)
59 epsrs=2.60d0 !dielctric constant(patch)
60 c
61 ds=0.80d-3 !substrate thickness(line)
62 dp=3.20d-3 !substrate thickness(patch)
63 ws=2.25d-3 !microstrip width (unit:m)
64 wa=1.5d-3 !slot width (unit:m)
65 ala=20.0d-3 !slot length (unit:m)
66 c

```

```

67      rad=32.0d-3          !patch radius (unit:m)
68      y0=20.0d-3          !offset in y-direction(unit:m)
69      stub=18.0d-3        !open stub length (unit:m)
70 c
71      fre=1.585d0          !frequency(unit:GHz)
72 c
73      epse=0.5d0*(epsrs+1.0d0)+0.5d0*(epsrs-1.0d0)
74      & /dsqrt(1.0d0+12.0*ds/ws) !effective dielectric const.
75      ak0=2.0d0*pi*fre/ac0  ! wavenumber in free space
76      aks=dsqrt(epsrs)*ak0  ! wavenumber in line subst.
77      akr=dsqrt(epsrp)*ak0  ! wavenumber in patch subst.
78      ake=dsqrt(epse)*ak0   ! propagat. const. in MSL
79      aka=dsqrt(0.5d0*(epsrs+epsrp))*ak0 ! wavenumber in slot
80      z0=120.0d0*pi         !120*pi
81 c
82      ha=ala/(na+1)         ! segment length(slot)
83      hf=0.04d0*ac0/fre/dsqrt(epse) ! segment length(line)
84 c
85      ir1=200               !division no. in regin 1 and 2
86      ir2=1500              !division no. in regin 3-1
87      ir3=1200              !division no. in regin 3-2
88      it=90                 !division no. of alpha
89 c
90      bt1l=1.8412d0/rad
91      call besjt(1.8412e0,zbes,9,ind)
92      bja=dbl(zbes(1))      !eigenvalue of TM11
93 c
94      bt2l=3.0542d0/rad
95      call besjt(3.0542e0,zbes,9,ind)
96      bjb=dbl(zbes(2))      !eigenvalue of TM21
97 c
98      bt3l=4.2012d0/rad
99      call besjt(4.2012e0,zbes,9,ind)
100     bje=dbl(zbes(3))      !eigenvalue of TM31
101 c
102     bt12=5.3314d0/rad
103     call besjt(5.3314e0,zbes,9,ind)
104     bjc=dbl(zbes(1))      !eigenvalue of TM12
105 c
106     bt4l=5.3176d0/rad
107     call besjt(5.3176e0,zbes,9,ind)
108     bjd=dbl(zbes(4))      !eigenvalue of TM41
109 c
110     rint=ak0/ir1
111     tint=0.5d0*pi/it
112     rr=0.5d0*rint
113     ma=ns+1
114 c
115 c     ===== Integral in the first region =====
116     do 1 i=1,ir1
117         area=rr*rint*tint
118         tt=0.5d0*tint
119         do 2 j=1,it
120             akx=rr*dcos(tt)
121             aky=rr*dsin(tt)
122             akz1=dsqrt(aks*aks-rr*rr)
123             akz3=dsqrt(akr*akr-rr*rr)
124             akz2=dsqrt(ak0*ak0-rr*rr)
125             fys=2.0d0*dsin(0.5d0*aky*ws)/(aky*ws)
126             fxs=2.0d0*dsin(0.5d0*akx*ws)/(akx*ws)
127             fya=2.0d0*dsin(0.5d0*aky*wa)/(aky*wa)
128             fxa=2.0d0*dsin(0.5d0*akx*wa)/(akx*wa)
129             ctes=akz1*dcos(akz1*ds)+cj*akz2*dsin(akz1*ds)
130             ctep=akz3*dcos(akz3*dp)+cj*akz2*dsin(akz3*dp)
131             ctms=epsrs*akz2*dcos(akz1*ds)+cj*akz1*dsin(akz1*ds)
132             ctmp=epsrp*akz2*dcos(akz3*dp)+cj*akz3*dsin(akz3*dp)

```

```

133 c
134 c      ----- green's function for strip (x) to strip (x) -----
135       cfgs1=(aks*aks-akx*akx)*akz2*dcos(akz1*ds)
136       &      +cj*(ak0*ak0-akx*akx)*akz1*dsin(akz1*ds)
137       cgsxx=-cj*z0*cfgs1*dsin(akz1*ds)/(ak0*ctes*ctms)
138 c      ----- green's function for slot(y) to strip (x) -----
139       cfgs2=cj*akx*akx*(epsrs-1.0d0)*dsin(akz1*ds)/(ctes*ctms)
140       cfgs3=akz1/ctes
141       cgsxy=cfgs2-cfgs3
142 c
143 c      ----- green's function for slot(y) to slot(y) -----
144       cgsyl=akz1*dcos(akz1*ds)+cj*epsrs*akz2*dsin(akz1*ds)
145       cgsy2=(aks*aks-aky*aky)/(akz1*ctms)
146       cgsy3=aky*aky*akz1*(epsrs-1.0d0)/ctes/ctms
147       cgpy1=akz3*dcos(akz3*dp)+cj*epsrp*akz2*dsin(akz3*dp)
148       cgpy2=(akr*akr-aky*aky)/(akz3*ctmp)
149       cgpy3=aky*aky*akz3*(epsrp-1.0d0)/ctep/ctmp
150       cghmyy=-(cgsyl*cgsy2-cgsy3*cgpy1*cgpy2-cgpy3)/(ak0*z0)
151 c
152 c      ----- green's function for patch (X) to patch (X) -----
153       cfgp1=(akr*akr-akx*akx)*akz2*dcos(akz3*dp)
154       &      +cj*(ak0*ak0-akx*akx)*akz3*dsin(akz3*dp)
155       cgejxx=-cj*z0*cfgp1*dsin(akz3*dp)/(ak0*ctep*ctmp)
156 c      ----- green's function for patch (Y) to patch (Y) -----
157       cfgp2=(akr*akr-aky*aky)*akz2*dcos(akz3*dp)
158       &      +cj*(ak0*ak0-aky*aky)*akz3*dsin(akz3*dp)
159       cgejyy=-cj*z0*cfgp2*dsin(akz3*dp)/(ak0*ctep*ctmp)
160 c      ----- green's function for patch(X) to patch(Y) -----
161       cfgp3=akz2*dcos(akz3*dp)+cj*akz3*dsin(akz3*dp)
162       cgejxy=cj*z0*akx*aky*cfgp3*dsin(akz3*dp)/(ak0*ctep*ctmp)
163 c
164 c      ----- green's function for slot(y) to patch (x) -----
165       cfgp4=cj*akx*akx*(epsrp-1.0d0)*dsin(akz3*dp)/(ctep*ctmp)
166       cfgp5=akz3/ctep
167       cgemxy=cfgp4-cfgp5
168 c      ----- green's function for slot(y) to patch (y) -----
169       cgemyy=cj*akx*aky*(epsrp-1.0d0)*dsin(akz3*dp)/(ctep*ctmp)
170 c
171 c      ----- Piecewise sinusoidal mode -----
172       f0=2.0d0*ake*(dcos(ake*hf)-dcos(akx*hf))
173       &      /(akx*akx-ake*ake)/dsin(ake*hf)
174       fay=2.0d0*aka*(dcos(aka*ha)-dcos(aky*ha))
175       &      /(aky*aky-aka*aka)/dsin(aka*ha)
176       call expans1(rad,bt11,rr,tt,bja,pi,fxp11,fyp11,fxo11,fyo11)
177       call expans1(rad,bt12,rr,tt,bjc,pi,fxp12,fyp12,fxo12,fyo12)
178       call expans2(rad,bt21,rr,tt,bjb,pi,fxp21,fyp21,fxo21,fyo21)
179       call expans3(rad,bt31,rr,tt,bje,pi,fxp31,fyp31,fxo31,fyo31)
180       call expans4(rad,bt41,rr,tt,bjd,pi,fxp41,fyp41,fxo41,fyo41)
181 c
182       cga1=fxp11*fxp11*cgejxx+2.0d0*fxp11*fyp11*cgejxy
183       &      +fyp11*fyp11*cgejyy
184       cga2=fxp12*fxp12*cgejxx+2.0d0*fxp12*fyp12*cgejxy
185       &      +fyp12*fyp12*cgejyy
186       cga3=fxp11*fxp12*cgejxx+(fxp11*fyp12+fxp12*fyp11)*cgejxy
187       &      +fyp11*fyp12*cgejyy
188       cgb1=fxo21*fxo21*cgejxx+2.0d0*fxo21*fyo21*cgejxy
189       &      +fyo21*fyo21*cgejyy
190       cgb2=fxo41*fxo41*cgejxx+2.0d0*fxo41*fyo41*cgejxy
191       &      +fyo41*fyo41*cgejyy
192       cgb3=fxo21*fxo41*cgejxx+(fxo21*fyo41+fxo41*fyo21)*cgejxy
193       &      +fyo21*fyo41*cgejyy
194       cgc1=fxp31*fxp31*cgejxx*2.0d0*fxp31*fyp31*cgejxy
195       &      +fyp31*fyp31*cgejyy
196       cgc2=fxp11*fxp31*cgejxx+(fxp11*fyp31+fxp31*fyp11)*cgejxy
197       &      +fyp11*fyp31*cgejyy
198       cgc3=fxp31*fxp12*cgejxx+(fxp31*fyp12+fxp12*fyp31)*cgejxy

```

```

199      &          +fyp31*fyp12*cgejyy
200 c
201      ca( 1, 1)=ca( 1, 1)+cga1*area
202      ca( 2, 2)=ca( 2, 2)+cga2*area
203      ca( 1, 2)=ca( 1, 2)+cga3*area
204      ca( 3, 3)=ca( 3, 3)+cgc1*area
205      ca( 1, 3)=ca( 1, 3)+cgc2*area
206      ca( 2, 3)=ca( 2, 3)+cgc3*area
207      ca( 4, 4)=ca( 4, 4)+cgb1*area
208      ca( 5, 5)=ca( 5, 5)+cgb2*area
209      ca( 4, 5)=ca( 4, 5)+cgb3*area
210      ca( 6, 6)=ca( 6, 6)+fxa*fxa*fay*fay*cghmyy*area
211 c
212      ca(1,6)=ca(1,6)+fxa*fay*(cgemxy*fxp11+cgemyy*fyp11)
213      &          *dcos(aky*y0)*area
214      ca(2,6)=ca(2,6)+fxa*fay*(cgemxy*fxp12+cgemyy*fyp12)
215      &          *dcos(aky*y0)*area
216      ca(3,6)=ca(3,6)+fxa*fay*(cgemxy*fxp31+cgemyy*fyp31)
217      &          *dcos(aky*y0)*area
218      ca(4,6)=ca(4,6)-fxa*fay*(cgemxy*fxo21+cgemyy*fyo21)
219      &          *dsin(aky*y0)*area
220      ca(5,6)=ca(5,6)-fxa*fay*(cgemxy*fxo41+cgemyy*fyo41)
221      &          *dsin(aky*y0)*area
222 c
223      do 101 k=1,ns
224      xm=(k-1)*hf
225      xn=stub-(ns-k+1)*hf
226      ca(n+k,n+1)=ca(n+k,n+1)+fys*fys*f0*f0*cgsxx
227      &          *dcos(akx*xm)*area
228      ca(6,n+k)=ca(6,n+k)-fys*f0*fxa*fay*cgsxy
229      &          *dcos(akx*xn)*area
230 101 continue
231 c
232      tt=tt+tint
233 2      continue
234      rr=rr+rint
235 1      continue
236      print *, ' Integral 1 finished.'
237 c
238      call swave(epsrs,ak0,ds,poles,inds)
239      if(inds.ne.0) go to 91
240      print *, ' (poles/ak0) = ',poles/ak0
241 91      call swave(epsrp,ak0,dp,polep,indp)
242      if(indp.ne.0) go to 92
243      print *, ' (polep/ak0) = ',polep/ak0
244 c
245 92      continue
246 c
247 c      ===== Integral in the second region (2-1) =====
248      tint=0.5d0*pi/it
249      rint=(akr-ak0)/irl
250      rr=ak0+0.5d0*rint
251      do 3 i=1,irl
252      area=rr*rint*tint
253      tt=0.5d0*tint
254      akz1=dsqrt(aks*aks-rr*rr)
255      akz3=dsqrt(akr*akr-rr*rr)
256      akz2=dsqrt(rr*rr-ak0*ak0)
257      do 4 j=1,it
258      akx=rr*dcos(tt)
259      aky=rr*dsin(tt)
260 c
261      fxa=2.0d0*dsin(0.5d0*akx*wa)/(akx*wa)
262      fya=2.0d0*dsin(0.5d0*aky*wa)/(aky*wa)
263      fys=2.0d0*dsin(0.5d0*aky*ws)/(aky*ws)
264      fxs=2.0d0*dsin(0.5d0*akx*ws)/(akx*ws)

```

```

265      tes=akz1*dcos(akz1*ds)+akz2*dsin(akz1*ds)
266      tep=akz3*dcos(akz3*dp)+akz2*dsin(akz3*dp)
267      tms=epsrs*akz2*dcos(akz1*ds)-akz1*dsin(akz1*ds)
268      tmp=epsrp*akz2*dcos(akz3*dp)-akz3*dsin(akz3*dp)
269 c
270 c      ----- green's function for strip(x) to strip(x) -----
271      fgs1=(aks*aks-akx*akx)*akz2*dcos(akz1*ds)
272      &      -(ak0*ak0-akx*akx)*akz1*dsin(akz1*ds)
273      cgsxx=-cj*z0*fgs1*dsin(akz1*ds)/(ak0*tes*tms)
274 c      ----- green's function for slot(y) to strip(x) -----
275      fgs2=-akx*akx*(epsrs-1.0d0)*dsin(akz1*ds)/tes/tms
276      fgs3=-akz1/tes
277      gsxy=fgs2+fgs3
278 c
279 c      ----- green's function for slot(y) to slot(y) -----
280      gsy1=akz1*dcos(akz1*ds)+epsrs*akz2*dsin(akz1*ds)
281      gsy2=(aks*aks-aky*aky)/(akz1*tms)
282      gsy3=aky*aky*akz1*(epsrs-1.0d0)/tes/tms
283      gpy1=akz3*dcos(akz3*dp)+epsrp*akz2*dsin(akz3*dp)
284      gpy2=(akr*akr-aky*aky)/(akz3*tmp)
285      gpy3=aky*aky*akz3*(epsrp-1.0d0)/tep/tmp
286      cgmyy=-cj*(gsy1*gsy2-gsy3*gpy1*gpy2-gpy3)/(ak0*z0)
287 c
288 c      ----- green's function for patch(x) to patch(x) -----
289      fgp1=(akr*akr-akx*akx)*akz2*dcos(akz3*dp)
290      &      -(ak0*ak0-akx*akx)*akz3*dsin(akz3*dp)
291      cgejxx=-cj*z0*fgp1*dsin(akz3*dp)/(ak0*tep*tmp)
292 c      ----- green's function for patch(y) to patch(y) -----
293      fgp2=(akr*akr-aky*aky)*akz2*dcos(akz3*dp)
294      &      -(ak0*ak0-aky*aky)*akz3*dsin(akz3*dp)
295      cgejyy=-cj*z0*fgp2*dsin(akz3*dp)/(ak0*tep*tmp)
296 c      ----- green's function for patch(x) to patch(y) -----
297      fgp3=akz2*dcos(akz3*dp)-akz3*dsin(akz3*dp)
298      cgejxy=cj*z0*akx*aky*fgp3*dsin(akz3*dp)/(ak0*tep*tmp)
299 c
300 c      ----- green's function for slot(y) to patch(x) -----
301      fgp1=-akx*akx*(epsrp-1.0d0)*dsin(akz3*dp)/tep/tmp
302      fgp2=-akz3/tep
303      gemxy=fgp1+fgp2
304 c      ----- green's function for slot(y) to patch(y) -----
305      gemyy=-akx*aky*(epsrp-1.0d0)*dsin(akz3*dp)/tep/tmp
306 c
307 c      ----- Piecewise sinusoidal modes -----
308      fay=2.0d0*aka*(dcos(aka*ha)-dcos(aky*ha))
309      &      /(aky*aky-aka*aka)/dsin(aka*ha)
310      f0=2.0d0*ake*(dcos(ake*hf)-dcos(akx*hf))
311      &      /(akx*akx-ake*ake)/dsin(ake*hf)
312      call expans1(rad,bt11,rr,tt,bja,pi,fxp11,fyp11,fxo11,fyo11)
313      call expans1(rad,bt12,rr,tt,bjc,pi,fxp12,fyp12,fxo12,fyo12)
314      call expans2(rad,bt21,rr,tt,bjb,pi,fxp21,fyp21,fxo21,fyo21)
315      call expans3(rad,bt31,rr,tt,bje,pi,fxp31,fyp31,fxo31,fyo31)
316      call expans4(rad,bt41,rr,tt,bjd,pi,fxp41,fyp41,fxo41,fyo41)
317 c
318      cga1=fxp11*fxp11*cgejxx+2.0d0*fxp11*fyp11*cgejxy
319      &      +fyp11*fyp11*cgejyy
320      cga2=fxp12*fxp12*cgejxx+2.0d0*fxp12*fyp12*cgejxy
321      &      +fyp12*fyp12*cgejyy
322      cga3=fxp11*fxp12*cgejxx+(fxp11*fyp12+fxp12*fyp11)*cgejxy
323      &      +fyp11*fyp12*cgejyy
324      cgb1=fxo21*fxo21*cgejxx+2.0d0*fxo21*fyo21*cgejxy
325      &      +fyo21*fyo21*cgejyy
326      cgb2=fxo41*fxo41*cgejxx+2.0d0*fxo41*fyo41*cgejxy
327      &      +fyo41*fyo41*cgejyy
328      cgb3=fxo21*fxo41*cgejxx+(fxo21*fyo41+fxo41*fyo21)*cgejxy
329      &      +fyo21*fyo41*cgejyy
330      cgc1=fxp31*fxp31*cgejxx+2.0d0*fxp31*fyp31*cgejxy

```

```

331      &      +fyp31*fyp31*cgejyy
332      cgc2=fxp11*fxp31*cgejxx+(fxp11*fyp31+fxp31*fyp11)*cgejxy
333      &      +fyp11*fyp31*cgejyy
334      cgc3=fxp31*fxp12*cgejxx+(fxp31*fyp12+fxp12*fyp31)*cgejxy
335      &      +fyp31*fyp12*cgejyy
336 c
337      ca( 1, 1)=ca( 1, 1)+cga1*area
338      ca( 2, 2)=ca( 2, 2)+cga2*area
339      ca( 1, 2)=ca( 1, 2)+cga3*area
340      ca( 3, 3)=ca( 3, 3)+cgc1*area
341      ca( 1, 3)=ca( 1, 3)+cgc2*area
342      ca( 2, 3)=ca( 2, 3)+cgc3*area
343      ca( 4, 4)=ca( 4, 4)+cgb1*area
344      ca( 5, 5)=ca( 5, 5)+cgb2*area
345      ca( 4, 5)=ca( 5, 5)+cgb3*area
346      ca( 6, 6)=ca( 6, 6)+fxa*fxa*fay*fay*cghmyy*area
347 c
348      ca(1,6)=ca(1,6)+fxa*fay*(gemxy*fxp11+gemyy*fyp11)
349      &      *dcos(aky*y0)*area
350      ca(2,6)=ca(2,6)+fxa*fay*(gemxy*fxp12+gemyy*fyp12)
351      &      *dcos(aky*y0)*area
352      ca(3,6)=ca(3,6)+fxa*fay*(gemxy*fxp31+gemyy*fyp31)
353      &      *dcos(aky*y0)*area
354      ca(4,6)=ca(4,6)-fxa*fay*(gemxy*fxo21+gemyy*fyo21)
355      &      *dsin(aky*y0)*area
356      ca(5,6)=ca(5,6)-fxa*fay*(gemxy*fxo41+gemyy*fyo41)
357      &      *dsin(aky*y0)*area
358 c
359      do 102 k=1,ns
360      xm=(k-1)*hf
361      xn=stub-(ns-k+1)*hf
362      ca(n+k,n+1)=ca(n+k,n+1)+fys*fys*f0*f0*cgsxx
363      &      *dcos(akx*xm)*area
364      ca(6,n+k)=ca(6,n+k)-fys*f0*fxa*fay*gsxy
365      &      *dcos(akx*xn)*area
366 102 continue
367 c
368      tt=tt+tint
369 4      continue
370      rr=rr+rint
371 3      continue
372      print *, ' Integral 2-1 finished.'
373 c
374      if(epsrs.eq.epsrp) go to 96
375 c      ===== Integral in the second region (2-2) =====
376      tint=0.5d0*pi/it
377      rint=(aks-akr)/irl
378      rr=akr+0.5d0*rint
379      do 5 i=1,irl
380      area=rr*rint*tint
381      tt=0.5d0*tint
382      akz1=dsqrt(aks*aks-rr*rr)
383      akz3=dsqrt(rr*rr-akr*akr)
384      akz2=dsqrt(rr*rr-ak0*ak0)
385      do 6 j=1,it
386      akx=rr*dcos(tt)
387      aky=rr*dsin(tt)
388 c
389      fxa=2.0d0*dsin(0.5d0*akx*wa)/(akx*wa)
390      fya=2.0d0*dsin(0.5d0*aky*wa)/(aky*wa)
391      fys=2.0d0*dsin(0.5d0*aky*ws)/(aky*ws)
392      fxs=2.0d0*dsin(0.5d0*akx*ws)/(akx*ws)
393      tes=akz1*dcos(akz1*ds)+akz2*dsin(akz1*ds)
394      tep=akz3*dcosh(akz3*dp)+akz2*dsinh(akz3*dp)
395      tms=epsrs*akz2*dcos(akz1*ds)-akz1*dsin(akz1*ds)
396      tmp=epsrp*akz2*dcosh(akz3*dp)+akz3*dsinh(akz3*dp)

```



```

397 c
398 c      ----- green's function for strip(x) to strip(x) -----
399      fgs1=(aks*aks-akx*akx)*akz2*dcos(akz1*ds)
400      &      -(ak0*ak0-akx*akx)*akz1*dsin(akz1*ds)
401      cgsxx=-cj*z0*fgs1*dsin(akz1*ds)/(ak0*tes*tms)
402 c      ----- green's function for slot(y) to strip(x) -----
403      fgs2=-akx*akx*(epsrs-1.0d0)*dsin(akz1*ds)/tes/tms
404      fgs3=-akz1/tes
405      gsxy=fgs2+fgs3
406 c
407 c      ----- green's function for slot(y) to slot(y) -----
408      gsy1=akz1*dcos(akz1*ds)+epsrs*akz2*dsin(akz1*ds)
409      gsy2=(aks*aks-aky*aky)/(akz1*tms)
410      gsy3=aky*aky*akz1*(epsrs-1.0d0)/tes/tms
411      gpy1=akz3*dcosh(akz3*dp)+epsrp*akz2*dsinh(akz3*dp)
412      gpy2=(akr*akr-aky*aky)/(akz3*tmp)
413      gpy3=aky*aky*akz3*(epsrp-1.0d0)/tep/tmp
414      cghmyy=-cj*(gsy1*gsy2-gsy3*gpy1*gpy2-gpy3)/(ak0*z0)
415 c
416 c      ----- green's function for patch(x) to patch(x) -----
417      fgp1=(akr*akr-akx*akx)*akz2*dcosh(akz3*dp)
418      &      +(ak0*ak0-akx*akx)*akz3*dsinh(akz3*dp)
419      cgejxx=-cj*z0*fgp1*dsinh(akz3*dp)/(ak0*tep*tmp)
420 c      ----- green's function for patch(y) to patch(y) -----
421      fgp2=(akr*akr-aky*aky)*akz2*dcosh(akz3*dp)
422      &      +(ak0*ak0-aky*aky)*akz3*dsinh(akz3*dp)
423      cgejyy=-cj*z0*fgp2*dsinh(akz3*dp)/(ak0*tep*tmp)
424 c      ----- green's function for patch(x) to patch(x) -----
425      fgp3=akz2*dcosh(akz3*dp)+akz3*dsinh(akz3*dp)
426      cgejxy=cj*z0*akx*aky*fgp3*dsinh(akz3*dp)/(ak0*tep*tmp)
427 c
428 c      ----- green's function for slot(y) to patch(x) -----
429      fgp1=-akx*akx*(epsrp-1.0d0)*dsinh(akz3*dp)/tep/tmp
430      fgp2=-akz3/tep
431      gemxy=fgp1+fgp2
432 c      ----- green's function for slot(y) to patch(y) -----
433      gemyy=-akx*aky*(epsrp-1.0d0)*dsinh(akz3*dp)/tep/tmp
434 c
435 c      ----- Piecewise sinusoidal modes -----
436      fay=2.0d0*aka*(dcos(aka*ha)-dcos(aky*ha))
437      &      /(aky*aky-aka*aka)/dsin(aka*ha)
438      f0x=2.0d0*ake*(dcos(ake*hf)-dcos(akx*hf))
439      &      /(akx*akx-ake*ake)/dsin(ake*hf)
440      call expans1(rad,bt11,rr,tt,bja,pi,fxp11,fyp11,fxo11,fyo11)
441      call expans1(rad,bt12,rr,tt,bjc,pi,fxp12,fyp12,fxo12,fyo12)
442      call expans2(rad,bt21,rr,tt,bjb,pi,fxp21,fyp21,fxo21,fyo21)
443      call expans3(rad,bt31,rr,tt,bje,pi,fxp31,fyp31,fxo31,fyo31)
444      call expans4(rad,bt41,rr,tt,bjd,pi,fxp41,fyp41,fxo41,fyo41)
445 c
446      cga1=fxp11*fxp11*cgejxx+2.0d0*fxp11*fyp11*cgejxy
447      &      +fyp11*fyp11*cgejyy
448      cga2=fxp12*fxp12*cgejxx+2.0d0*fxp12*fyp12*cgejxy
449      &      +fyp12*fyp12*cgejyy
450      cga3=fxp11*fxp12*cgejxx+(fxp11*fyp12+fxp12*fyp11)*cgejxy
451      &      +fyp11*fyp12*cgejyy
452      cgb1=fxo21*fxo21*cgejxx+2.0d0*fxo21*fyo21*cgejxy
453      &      +fyo21*fyo21*cgejyy
454      cgb2=fxo41*fxo41*cgejxx+2.0d0*fxo41*fyo41*cgejxy
455      &      +fyo41*fyo41*cgejyy
456      cgb3=fxo21*fxo41*cgejxx+(fxo21*fyo41+fxo41*fyo21)*cgejxy
457      &      +fyo21*fyo41*cgejyy
458      cgc1=fxp31*fxp31*cgejxx+2.0d0*fxp31*fyp31*cgejxy
459      &      +fyp31*fyp31*cgejyy
460      cgc2=fxp11*fxp31*cgejxx+(fxp11*fyp31+fxp31*fyp11)*cgejxy
461      &      +fyp11*fyp31*cgejyy
462      cgc3=fxp31*fxp12*cgejxx+(fxp31*fyp12+fxp12*fyp31)*cgejxy

```

```

463      &          +fyp31*fyp12*cgejyy
464 c
465          ca( 1, 1)=ca( 1, 1)+cga1*area
466          ca( 2, 2)=ca( 2, 2)+cga2*area
467          ca( 1, 2)=ca( 1, 2)+cga3*area
468          ca( 3, 3)=ca( 3, 3)+cgc1*area
469          ca( 1, 3)=ca( 1, 3)+cgc2*area
470          ca( 2, 3)=ca( 2, 3)+cgc3*area
471          ca( 4, 4)=ca( 4, 4)+cgb1*area
472          ca( 5, 5)=ca( 5, 5)+cgb2*area
473          ca( 4, 5)=ca( 4, 5)+cgb3*area
474          ca( 6, 6)=ca( 6, 6)+fxa*fxa*fay*fay*cghmyy*area
475 c
476          ca(1,6)=ca(1,6)+fxa*fay*(gemxy*fxp11+gemyy*fyp11)
477      &          *dcos(aky*y0)*area
478          ca(2,6)=ca(2,6)+fxa*fay*(gemxy*fxp12+gemyy*fyp12)
479      &          *dcos(aky*y0)*area
480          ca(3,6)=ca(3,6)+fxa*fay*(gemxy*fxp31+gemyy*fyp31)
481      &          *dcos(aky*y0)*area
482          ca(4,6)=ca(4,6)-fxa*fay*(gemxy*fxo21+gemyy*fyo21)
483      &          *dsin(aky*y0)*area
484          ca(5,6)=ca(5,6)-fxa*fay*(gemxy*fxo41+gemyy*fyo41)
485      &          *dsin(aky*y0)*area
486 c
487          do 103 k=1,ns
488              xm=(k-1)*hf
489              xn=stub-(ns-k+1)*hf
490              ca(n+k,n+1)=ca(n+k,n+1)+fys*fys*f0x*f0x*cgsxx
491      &          *dcos(akx*xm)*area
492              ca(6,n+k)=ca(6,n+k)-fys*f0x*fxa*fay*gsxy
493      &          *dcos(akx*xn)*area
494 103      continue
495 c
496          tt=tt+tint
497 6          continue
498          rr=rr+rint
499 5          continue
500          print *, ' Integral 2-2 finished.'
501 c
502 96      continue
503 c
504 c      ===== Integral surface wave pole resisittance(strip side) =====
505 if(inds.ne.0) go to 99
506 tt=0.5d0*tint
507 rr=poles
508 akz1=dsqrt(aks*aks-rr*rr)
509 akz2=dsqrt(rr*rr-ak0*ak0)
510 area=rr*tint
511 do 11 j=1,it
512     akx=rr*dcos(tt)
513     aky=rr*dsin(tt)
514     fxa=2.0d0*dsin(0.5d0*akx*wa)/(akx*wa)
515     fya=2.0d0*dsin(0.5d0*aky*wa)/(aky*wa)
516     fys=2.0d0*dsin(0.5d0*aky*ws)/(aky*ws)
517     fxs=2.0d0*dsin(0.5d0*akx*ws)/(akx*ws)
518     tes=akz1*dcos(akz1*ds)+akz2*dsin(akz1*ds)
519     tms=epsrs*rr*dcos(akz1*ds)/akz2
520 &     +(epsrs*akz2*ds*dsin(akz1*ds)+dsin(akz1*ds)
521 &     +akz1*ds*dcos(akz1*ds))
522 &     *rr/akz1
523 c      ----- green's function for strip(x) to strip(x) -----
524     fgs1=(aks*aks-akx*akx)*akz2*dcos(akz1*ds)
525 &     +(ak0*ak0-akx*akx)*akz1*dsin(akz1*ds)
526     cgsxx=-cj*z0*fgs1*dsin(akz1*ds)/(ak0*tes*tms)
527 c      ----- green's function for slot(y) to strip(x) -----
528     fgs2=-akx*akx*(epsrs-1.0d0)*dsin(akz1*ds)/tes/tms

```

```

529      fgs3=-akz1/tes
530      gsxy=fgs2+fgs3
531 c
532 c      ----- green's function for slot(x) to slot(x) -----
533      gsy1=akz1*dcos(akz1*ds)+epsrs*akz2*dsin(akz1*ds)
534      gsy2=(aks*aks-aky*aky)/(akz1*tms)
535      gsy3=aky*aky*akz1*(epsrs-1.0d0)/tes/tms
536      cghmyy=-cj*(gsy1*gsy2-gsy3)/(ak0*z0)
537 c
538 c      ----- Piecewise sinusoidal mode -----
539      fay=2.0d0*aka*(dcos(aka*ha)-dcos(aky*ha))
540      &      /(aky*aky-aka*aka)/dsin(aka*ha)
541      f0x=2.0d0*ake*(dcos(ake*hf)-dcos(akx*hf))
542      &      /(akx*akx-ake*ake)/dsin(ake*hf)
543 c
544      ca( 6, 6)=ca( 6, 6)-cj*fxa*fxa*fay*fay*cghmyy*area
545 c
546      do 104 k=1,ns
547          xm=(k-1)*hf
548          xn=stub-(ns-k+1)*hf
549          ca(n+k,n+1)=ca(n+k,n+1)-cj*pi*fys*fys*f0x*f0x*cgsxx
550      &          *dcos(akx*xm)*area
551      &          ca(6,n+k)=ca(6,n+k)+cj*pi*fys*f0x*fxa*fay*gsxy
552      &          *dcos(akx*xn)*area
553 104      continue
554 c
555      tt=tt+tint
556 11      continue
557      print *, '    Integral surface finised(strip side)'
558 99      continue
559 c
560 c      ===== Surface wave resistance (patch side ) =====
561      if(indp.ne.0) go to 97
562      tint=0.5d0*pi/it
563      tt=0.5d0*tint
564      rr=polep
565      area=rr*tint
566      akz3=dsqrt(akr*akr-rr*rr)
567      akz2=dsqrt(rr*rr-ak0*ak0)
568      do 13 j=1,it
569          akx=rr*dcos(tt)
570          aky=rr*dsin(tt)
571          fxa=2.0d0*dsin(0.5d0*akx*wa)/(akx*wa)
572          fya=2.0d0*dsin(0.5d0*aky*wa)/(aky*wa)
573          tep=akz3*dcos(akz3*dp)+akz2*dsin(akz3*dp)
574          tmp=epsrp*rr*dcos(akz3*dp)/akz2+rr/akz3*
575      &          (epsrp*akz2*dp*dsin(akz3*dp)+dcos(akz3*dp)
576      &          +akz3*dp*dcos(akz3*dp))
577 c
578 c      ----- Green's function for slot(x) to slot(x) -----
579      gpy1=akz3*dcos(akz3*dp)+epsrp*akz2*dsin(akz3*dp)
580      gpy2=(akr*akr-aky*aky)/(akz3*tmp)
581      gpy3=aky*aky*akz3*(epsrp-1.0d0)/tep/tmp
582      cghmyy=-cj*(gpy1*gpy2-gpy3)/(ak0*z0)
583 c
584 c      ----- Green's function for patch(x) to patch(x) -----
585      fgp1=(akr*akr-akx*akx)*akz2*dcos(akz3*dp)
586      &      -(ak0*ak0-akx*akx)*akz3*dsin(akz3*dp)
587      cgejxx=-cj*z0*fgp1*dsin(akz3*dp)/(ak0*tep*tmp)
588 c      ----- Green's function for patch(y) to patch(y) -----
589      fgp2=(akr*akr-aky*aky)*akz2*dcos(akz3*dp)
590      &      -(ak0*ak0-aky*aky)*akz3*dsin(akz3*dp)
591      cgejyy=-cj*z0*fgp2*dsin(akz3*dp)/(ak0*tep*tmp)
592 c      ----- Green's function for patch(x) to patch(y) -----
593      fgp3=akz2*dcos(akz3*dp)-akz3*dsin(akz3*dp)
594      cgejxy=cj*akx*aky*z0*fgp3*dsin(akz3*dp)/(ak0*tep*tmp)

```

```

595 c
596 c ----- green's function for slot(y) to patch(x) -----
597     gemxy=-akx*akx*(epsrp-1.0d0)*dsinh(akz3*dp)/tep/tmp
598 c ----- green's function for slot(y) to patch(y) -----
599     gemyy=-akx*aky*(epsrp-1.0d0)*dsinh(akz3*dp)/tep/tmp
600 c
601 c ----- Piecewise sinusoidal modes -----
602     fay=2.0d0*aka*(dcos(aka*ha)-dcos(aky*ha))
603     & /(aky*aky-aka*aka)/dsin(aka*ha)
604     call expans1(rad,bt11,rr,tt,bja,pi,fxp11,fyp11,fxo11,fyo11)
605     call expans1(rad,bt12,rr,tt,bjc,pi,fxp12,fyp12,fxo12,fyo12)
606     call expans2(rad,bt21,rr,tt,bjb,pi,fxp21,fyp21,fxo21,fyo21)
607     call expans3(rad,bt31,rr,tt,bje,pi,fxp31,fyp31,fxo31,fyo31)
608     call expans4(rad,bt41,rr,tt,bjd,pi,fxp41,fyp41,fxo41,fyo41)
609 c
610     cga1=fxp11*fxp11*cgejxx+2.0d0*fxp11*fyp11*cgejxy
611     & +fyp11*fyp11*cgejyy
612     cga2=fxp12*fxp12*cgejxx+2.0d0*fxp12*fyp12*cgejxy
613     & +fyp12*fyp12*cgejyy
614     cga3=fxp11*fxp12*cgejxx+(fxp11*fyp12+fxp12*fyp11)*cgejxy
615     & +fyp11*fyp12*cgejyy
616     cgb1=fxo21*fxo21*cgejxx+2.0d0*fxo21*fyo21*cgejxy
617     & +fyo21*fyo21*cgejyy
618     cgb2=fxo41*fxo41*cgejxx+2.0d0*fxo41*fyo41*cgejxy
619     & +fyo41*fyo41*cgejyy
620     cgb3=fxo21*fxo41*cgejxx+(fxo21*fyo41+fxo41*fyo21)*cgejxy
621     & +fyo21*fyo41*cgejyy
622     cgc1=fxp31*fxp31*cgejxx+2.0d0*fxp31*fyp31*cgejxy
623     & +fyp31*fyp31*cgejyy
624     cgc2=fxp11*fxp31*cgejxx+(fxp11*fyp31+fxp31*fyp11)*cgejxy
625     & +fyp11*fyp31*cgejyy
626     cgc3=fxp31*fxp12*cgejxx+(fxp31*fyp12+fxp12*fyp31)*cgejxy
627     & +fyp31*fyp12*cgejyy
628 c
629     ca( 1, 1)=ca( 1, 1)+cga1*area
630     ca( 2, 2)=ca( 2, 2)+cga2*area
631     ca( 1, 2)=ca( 1, 2)+cga3*area
632     ca( 3, 3)=ca( 3, 3)+cgc1*area
633     ca( 1, 3)=ca( 1, 3)+cgc2*area
634     ca( 2, 3)=ca( 2, 3)+cgc3*area
635     ca( 4, 4)=ca( 4, 4)+cgb1*area
636     ca( 5, 5)=ca( 5, 5)+cgb2*area
637     ca( 4, 5)=ca( 4, 5)+cgb3*area
638     ca( 6, 6)=ca( 6, 6)+fxa*fxa*fay*fay*cghmyy*area
639 c
640     ca(1,6)=ca(1,6)-cj*pi*fxa*fay*(gemxy*fxp11+gemyy*fyp11)
641     & *dcos(aky*y0)*area
642     ca(2,6)=ca(2,6)-cj*pi*fxa*fay*(gemxy*fxp12+gemyy*fyp12)
643     & *dcos(aky*y0)*area
644     ca(3,6)=ca(3,6)-cj*pi*fxa*fay*(gemxy*fxp31+gemyy*fyp31)
645     & *dcos(aky*y0)*area
646     ca(4,6)=ca(4,6)+cj*pi*fxa*fay*(gemxy*fxo21+gemyy*fyo21)
647     & *dsin(aky*y0)*area
648     ca(5,6)=ca(5,6)+cj*pi*fxa*fay*(gemxy*fxo41+gemyy*fyo41)
649     & *dsin(aky*y0)*area
650 c
651     tt=tt+tint
652 13 continue
653     print *, ' Integral surface(patch) finished'
654 c
655 97 continue
656 c
657 c ===== Integral in the third region (3-1) =====
658     tint=0.5d0*pi/it
659     rint=0.025d0*ak0
660     rr=aks+0.5d0*rint

```

```

661      do 7 i=1,ir2
662          area=rr*rint*tint
663          tt=0.5d0*tint
664          akz1=dsqrt(rr*rr-aks*aks)
665          akz3=dsqrt(rr*rr-akr*akr)
666          akz2=dsqrt(rr*rr-ak0*ak0)
667      do 8 j=1,it
668          akx=rr*dcos(tt)
669          aky=rr*dsin(tt)
670          fxa=2.0d0*dsin(0.5d0*akx*wa)/(akx*wa)
671          fya=2.0d0*dsin(0.5d0*aky*wa)/(aky*wa)
672          fys=2.0d0*dsin(0.5d0*aky*ws)/(aky*ws)
673          fxs=2.0d0*dsin(0.5d0*akx*ws)/(akx*ws)
674          tes=akz1*dcosh(akz1*ds)+akz2*dsinh(akz1*ds)
675          tep=akz3*dcosh(akz3*dp)+akz2*dsinh(akz3*dp)
676          tms=epsrs*akz2*dcosh(akz1*ds)+akz1*dsinh(akz1*ds)
677          tmp=epsrp*akz2*dcosh(akz3*dp)+akz3*dsinh(akz3*dp)
678 c
679 c      ----- green's function for strip(x) to strip(x) -----
680          fgs1=(aks*aks-akx*akx)*akz2*dcosh(akz1*ds)
681      &          +(ak0*ak0-akx*akx)*akz1*dsinh(akz1*ds)
682          cgsxx=-cj*z0*fgs1*dsinh(akz1*ds)/(ak0*tms)
683 c      ----- green's function for slot(y) to patch(x) -----
684          fgs2=-akx*akx*(epsrs-1.0d0)*dsinh(akz1*ds)/tes/tms
685          fgs3=-akz1/tes
686          gsxy=fgs2+fgs3
687 c
688 c      ----- green's function for slot(y) to slot(y) -----
689          gsy1=akz1*dcosh(akz1*ds)+epsrs*akz2*dsinh(akz1*ds)
690          gsy2=(aks*aks-aky*aky)/(akz1*tms)
691          gsy3=aky*aky*akz1*(epsrs-1.0d0)/tes/tms
692          gpy1=akz3*dcosh(akz3*dp)+epsrp*akz2*dsinh(akz3*dp)
693          gpy2=(akr*akr-aky*aky)/(akz3*tmp)
694          gpy3=aky*aky*akz3*(epsrp-1.0d0)/tep/tmp
695          cghmyy=-cj*(gsy1*gsy2-gsy3*gpy1*gpy2-gpy3)/(ak0*z0)
696 c
697 c      ----- green's function for patch(x) to patch(x) -----
698          fgp1=(akr*akr-akx*akx)*akz2*dcosh(akz3*dp)
699      &          +(ak0*ak0-akx*akx)*akz3*dsinh(akz3*dp)
700          cgejxx=-cj*z0*fgp1*dsinh(akz3*dp)/(ak0*tep*tmp)
701 c      ----- green's function for patch(y) to patch(y) -----
702          fgp2=(akr*akr-aky*aky)*akz2*dcosh(akz3*dp)
703      &          +(ak0*ak0-aky*aky)*akz3*dsinh(akz3*dp)
704          cgejyy=-cj*z0*fgp2*dsinh(akz3*dp)/(ak0*tep*tmp)
705 c      ----- green's function for patch(x) to patch(x) -----
706          fgp3=akz2*dcosh(akz3*dp)+akz3*dsinh(akz3*dp)
707          cgejxy=cj*z0*akx*aky*fgp3*dsinh(akz3*dp)/(ak0*tep*tmp)
708 c
709 c      ----- green's function for slot(y) to patch(x) -----
710          fgp1=-akx*akx*(epsrp-1.0d0)*dsinh(akz3*dp)/tep/tmp
711          fgp2=-akz3/tep
712          gemxy=fgp1+fgp2
713 c      ----- green's function for slot(y) to patch(y) -----
714          gemyy=-akx*aky*(epsrp-1.0d0)*dsinh(akz3*dp)/tep/tmp
715 c
716 c      ----- Piecewise sinusoidal modes -----
717          fay=2.0d0*aka*(dcos(aka*ha)-dcos(aky*ha))
718      &          /(aky*aky-aka*aka)/dsin(aka*ha)
719          f0x=2.0d0*ake*(dcos(ake*hf)-dcos(akx*hf))
720      &          /(akx*akx-ake*ake)/dsin(ake*hf)
721          call expans1(rad,bt11,rr,tt,bja,pi,fxp11,fyp11,fxo11,fyo11)
722          call expans1(rad,bt12,rr,tt,bjc,pi,fxp12,fyp12,fxo12,fyo12)
723          call expans2(rad,bt21,rr,tt,bjb,pi,fxp21,fyp21,fxo21,fyo21)
724          call expans3(rad,bt31,rr,tt,bje,pi,fxp31,fyp31,fxo31,fyo31)
725          call expans4(rad,bt41,rr,tt,bjd,pi,fxp41,fyp41,fxo41,fyo41)
726 c

```

```

727      cga1=fxp11*fxp11*cgejxx+2.0d0*fxp11*fyp11*cgejxy
728      &      +fyp11*fyp11*cgejyy
729      cga2=fxp12*fxp12*cgejxx+2.0d0*fxp12*fyp12*cgejxy
730      &      +fyp12*fyp12*cgejyy
731      cga3=fxp11*fxp12*cgejxx+(fxp11*fyp12+fxp12*fyp11)*cgejxy
732      &      +fyp11*fyp12*cgejyy
733      cgb1=fxo21*fxo21*cgejxx+2.0d0*fxo21*fyo21*cgejxy
734      &      +fyo21*fyo21*cgejyy
735      cgb2=fxo41*fxo41*cgejxx+2.0d0*fxo41*fyo41*cgejxy
736      &      +fyo41*fyo41*cgejyy
737      cgb3=fxo21*fxo41*cgejxx+(fxo21*fyo41+fxo41*fyo21)*cgejxy
738      &      +fyo21*fyo41*cgejyy
739      cgc1=fxp31*fxp31*cgejxx+2.0d0*fxp31*fyp31*cgejxy
740      &      +fyp31*fyp31*cgejyy
741      cgc2=fxp11*fxp31*cgejxx+(fxp11*fyp31+fxp31*fyp11)*cgejxy
742      &      +fyp11*fyp31*cgejyy
743      cgc3=fxp31*fxp12*cgejxx+(fxp31*fyp12+fxp12*fyp31)*cgejxy
744      &      +fyp31*fyp12*cgejyy
745 c
746      ca( 1, 1)=ca( 1, 1)+cga1*area
747      ca( 2, 2)=ca( 2, 2)+cga2*area
748      ca( 1, 2)=ca( 1, 2)+cga3*area
749      ca( 3, 3)=ca( 3, 3)+cgc1*area
750      ca( 1, 3)=ca( 1, 3)+cgc2*area
751      ca( 2, 3)=ca( 2, 3)+cgc3*area
752      ca( 4, 4)=ca( 4, 4)+cgb1*area
753      ca( 5, 5)=ca( 5, 5)+cgb2*area
754      ca( 4, 5)=ca( 4, 5)+cgb3*area
755      ca( 6, 6)=ca( 6, 6)+fxa*fxa*fay*fay*cghmyy*area
756 c
757      ca(1,6)=ca(1,6)+fxa*fay*(gemxy*fxp11+gemyy*fyp11)
758      &      *dcos(aky*y0)*area
759      ca(2,6)=ca(2,6)+fxa*fay*(gemxy*fxp12+gemyy*fyp12)
760      &      *dcos(aky*y0)*area
761      ca(3,6)=ca(3,6)+fxa*fay*(gemxy*fxp31+gemyy*fyp31)
762      &      *dcos(aky*y0)*area
763      ca(4,6)=ca(4,6)-fxa*fay*(gemxy*fxo21+gemyy*fyo21)
764      &      *dsin(aky*y0)*area
765      ca(5,6)=ca(5,6)-fxa*fay*(gemxy*fxo41+gemyy*fyo41)
766      &      *dsin(aky*y0)*area
767 c
768      do 106 k=1,ns
769          xm=(k-1)*hf
770          xn=stub-(ns-k+1)*hf
771          ca(n+k,n+1)=ca(n+k,n+1)+fys*fys*f0x*f0x*cgsxx
772      &          *dcos(akx*xm)*area
773          ca(6,n+k)=ca(6,n+k)-fys*f0x*fxa*fay*gsxy
774      &          *dcos(akx*xn)*area
775 106      continue
776 c
777      tt=tt+tint
778 8      continue
779      rr=rr+rint
780 7      continue
781      rinto=rint
782 c
783      print *, 'Integral 3-1 finished'
784 c
785 c      ===== Integral in the third region (3-2) =====
786      tint=0.5d0*pi/it
787      rint=0.06d0*ak0
788      rr=aks+rinto*ir2+0.5d0*rint
789      do 9 i=1,ir3
790          area=rr*rint*tint
791          tt=0.5d0*tint
792          akz1=dsqrt(rr*rr-aks*aks)

```

```

793      akz3=dsqrt(rr*rr-akr*akr)
794      akz2=dsqrt(rr*rr-ak0*ak0)
795      do 10 j=1,it
796          akx=rr*dcos(tt)
797          aky=rr*dsin(tt)
798          fxa=2.0d0*dsin(0.5d0*akx*wa)/(akx*wa)
799          fya=2.0d0*dsin(0.5d0*aky*wa)/(aky*wa)
800          fys=2.0d0*dsin(0.5d0*aky*ws)/(aky*ws)
801          fxs=2.0d0*dsin(0.5d0*akx*ws)/(akx*ws)
802          tes=akz1*dcosh(akz1*ds)+akz2*dsinh(akz1*ds)
803          tep=akz3*dcosh(akz3*dp)+akz2*dsinh(akz3*dp)
804          tms=epsrs*akz2*dcosh(akz1*ds)+akz1*dsinh(akz1*ds)
805          tmp=epsrp*akz2*dcosh(akz3*dp)+akz3*dsinh(akz3*dp)
806 c
807 c      ----- green's function for strip(x) to strip(x) -----
808          fgs1=(aks*aks-akx*akx)*akz2*dcosh(akz1*ds)
809      &          +(ak0*ak0-akx*akx)*akz1*dsinh(akz1*ds)
810          cgsxx=-cj*z0*fgs1*dsinh(akz1*ds)/(ak0*tms)
811 c      ----- green's function for slot(y) to patch(x) -----
812          fgs2=-akx*akx*(epsrs-1.0d0)*dsinh(akz1*ds)/tes/tms
813          fgs3=-akz1/tes
814          gsxy=fgs2+fgs3
815 c
816 c      ----- green's function for slot(y) to slot(y) -----
817          gsy1=akz1*dcosh(akz1*ds)+epsrs*akz2*dsinh(akz1*ds)
818          gsy2=(aks*aks-aky*aky)/(akz1*tms)
819          gsy3=aky*aky*akz1*(epsrs-1.0d0)/tes/tms
820          gpy1=akz3*dcosh(akz3*dp)+epsrp*akz2*dsinh(akz3*dp)
821          gpy2=(akr*akr-aky*aky)/(akz3*tmp)
822          gpy3=aky*aky*akz3*(epsrp-1.0d0)/tep/tmp
823          cghmyy=-cj*(gsy1*gsy2-gsy3*gpy1*gpy2-gpy3)/(ak0*z0)
824 c
825 c      ----- green's function for patch(x) to patch(x) -----
826          fgp1=(akr*akr-akx*akx)*akz2*dcosh(akz3*dp)
827      &          +(ak0*ak0-akx*akx)*akz3*dsinh(akz3*dp)
828          cgejxx=-cj*z0*fgp1*dsinh(akz3*dp)/(ak0*tmp*tep)
829 c      ----- green's function for patch(y) to patch(y) -----
830          fgp2=(akr*akr-aky*aky)*akz2*dcosh(akz3*dp)
831      &          +(ak0*ak0-aky*aky)*akz3*dsinh(akz3*dp)
832          cgejyy=-cj*z0*fgp2*dsinh(akz3*dp)/(ak0*tmp*tep)
833 c      ----- green's function for patch(x) to patch(y) -----
834          fgp3=akz2*dcosh(akz3*dp)+akz3*dsinh(akz3*dp)
835          cgejxy=cj*z0*akx*aky*fgp3*dsinh(akz3*dp)/(ak0*tmp*tep)
836 c
837 c      ----- green's function for slot(y) to patch(x) -----
838          fgp1=-akx*akx*(epsrp-1.0d0)*dsinh(akz3*dp)/tep/tmp
839          fgp2=-akz3/tep
840          gemxy=fgp1+fgp2
841 c      ----- green's function for slot(y) to patch(y) -----
842          gemyy=-akx*aky*(epsrp-1.0d0)*dsinh(akz3*dp)/tep/tmp
843 c
844 c      ----- Piecewise sinusoidal modes -----
845          fay=2.0d0*aka*(dcos(aka*ha)-dcos(aky*ha))
846      &          /(aky*aky-aka*aka)/dsin(aka*ha)
847          f0=2.0d0*ake*(dcos(ake*hf)-dcos(akx*hf))
848      &          /(akx*akx-ake*ake)/dsin(ake*hf)
849          call expans1(rad,bt11,rr,tt,bja,pi,fxp11,fyp11,fxo11,fyo11)
850          call expans1(rad,bt12,rr,tt,bjc,pi,fxp12,fyp12,fxo12,fyo12)
851          call expans2(rad,bt21,rr,tt,bjb,pi,fxp21,fyp21,fxo21,fyo21)
852          call expans3(rad,bt31,rr,tt,bje,pi,fxp31,fyp31,fxo31,fyo31)
853          call expans4(rad,bt41,rr,tt,bjd,pi,fxp41,fyp41,fxo41,fyo41)
854 c
855          cga1=fxp11*fxp11*cgejxx+2.0d0*fxp11*fyp11*cgejxy
856      &          +fyp11*fyp11*cgejyy
857          cga2=fxp12*fxp12*cgejxx+2.0d0*fxp12*fyp12*cgejxy
858      &          +fyp12*fyp12*cgejyy

```

```

859      cga3=fxp11*fxp12*cgejxx+(fxp11*fyp12+fxp12*fyp11)*cgejxy
860      &      +fyp11*fyp12*cgejyy
861      cgb1=fxo21*fxo21*cgejxx+2.0d0*fxo21*fyo21*cgejxy
862      &      +fyo21*fyo21*cgejyy
863      cgb2=fxo41*fxo41*cgejxx+2.0d0*fxo41*fyo41*cgejxy
864      &      +fyo41*fyo41*cgejyy
865      cgb3=fxo21*fxo41*cgejxx+(fxo21*fyo41+fxo41*fyo21)*cgejxy
866      &      +fyo21*fyo41*cgejyy
867      cgc1=fxp31*fxp31*cgejxx+2.0d0*fxp31*fyp31*cgejxy
868      &      +fyp31*fyp31*cgejyy
869      cgc2=fxp11*fxp31*cgejxx+(fxp11*fyp31+fxp31*fyp11)*cgejxy
870      &      +fyp11*fyp31*cgejyy
871      cgc3=fxp31*fxp12*cgejxx+(fxp31*fyp12+fxp12*fyp31)*cgejxy
872      &      +fyp31*fyp12*cgejyy
873 c
874      ca( 1, 1)=ca( 1, 1)+cga1*area
875      ca( 2, 2)=ca( 2, 2)+cga2*area
876      ca( 1, 2)=ca( 1, 2)+cga3*area
877      ca( 3, 3)=ca( 3, 3)+cgc1*area
878      ca( 1, 3)=ca( 1, 3)+cgc2*area
879      ca( 2, 3)=ca( 2, 3)+cgc3*area
880      ca( 4, 4)=ca( 4, 4)+cgb1*area
881      ca( 5, 5)=ca( 5, 5)+cgb2*area
882      ca( 4, 5)=ca( 4, 5)+cgb3*area
883      ca( 6, 6)=ca( 6, 6)+fxa*fxa*fay*fay*cghmyy*area
884 c
885      ca(1,6)=ca(1,6)+fxa*fay*(gemxy*fxp11+gemyy*fyp11)
886      &      *dcos(aky*y0)*area
887      ca(2,6)=ca(2,6)+fxa*fay*(gemxy*fxp12+gemyy*fyp12)
888      &      *dcos(aky*y0)*area
889      ca(3,6)=ca(3,6)+fxa*fay*(gemxy*fxp31+gemyy*fyp31)
890      &      *dcos(aky*y0)*area
891      ca(4,6)=ca(4,6)-fxa*fay*(gemxy*fxo21+gemyy*fyo21)
892      &      *dsin(aky*y0)*area
893      ca(5,6)=ca(5,6)-fxa*fay*(gemxy*fxo41+gemyy*fyo41)
894      &      *dsin(aky*y0)*area
895 c
896      do 107 k=1,ns
897          xm=(k-1)*hf
898          xn=stub-(ns-k+1)*hf
899          ca(n+k,n+1)=ca(n+k,n+1)+fys*fys*f0*f0*cgsxx
900      &          *dcos(akx*xm)*area
901          ca(6,n+k)=ca(6,n+k)-fys*f0*fxa*fay*gsxy
902      &          *dcos(akx*xn)*area
903 107      continue
904 c
905      tt=tt+tint
906 10      continue
907      rr=rr+rint
908 9      continue
909      print *, 'All integral finished.'
910 c
911 c      ===== matrix construction =====
912      ca(2,1)=ca(1,2)
913      ca(3,1)=ca(1,3)
914      ca(3,2)=ca(2,3)
915      ca(5,4)=ca(4,5)
916 c
917      ca(6,1)=-ca(1,6)
918      ca(6,2)=-ca(2,6)
919      ca(6,3)=-ca(3,6)
920      ca(6,7)=-ca(7,6)
921      ca(6,8)=-ca(8,6)
922 c
923      do 20 i=2,ns
924 20      ca(n+i,n+i)=ca(n+1,n+1)

```



```

925      do 21 i=1,ns-1
926          do 22 j=1,ns-i
927              ca(n+i+j,n+j)=ca(n+i+1,n+1)
928              ca(n+j,n+i+j)=ca(n+i+1,n+1)
929      22      continue
930      21      continue
931      do 23 i=1,ns
932          ca(n+i,6)=-ca(6,n+i)
933      23      continue
934      c
935          cb(n+1)=(1.0d0,0.0d0)
936      c
937          call tledc(mm,ca,cb,cc,nn,eps,ind)
938      c
939          write(6,*) 'Freq.[GHz] = ',fre
940          write(6,*) ' '
941          do 31 i=1,n
942      31      write(6,*) mode(i),cc(i)
943      c
944          amax=0.0d0
945          amin=1.0d0
946          ss=-3*hf
947          do 40 i=n+3,ns+n
948              xm=stub-(ns-(i-n)+1)*hf
949              if(xm.gt.ss) go to 201
950              do 43 j=0,39
951                  s=0.025d0*j
952                  cjx=cc(i)*dsin(ake*hf*(1-s))/dsin(ake*hf)
953      &                  +cc(i+1)*dsin(ake*hf*s)/dsin(ake*hf)
954                  acc=cdabs(cjx)
955                  if(acc.gt.amax) then
956                      amax=acc
957                  else if(acc.lt.amin) then
958                      amin=acc
959                      xmin=xm+hf*s
960                  endif
961      43      continue
962      40      continue
963      201      continue
964      c
965          vswr=amax/amin
966          write(6,*) 'VSWR = ', vswr
967          agm=(vswr-1.0d0)/(vswr+1.0d0)
968          agmd=-20.0d0*dlog10(agm)
969          write(6,*) 'Return Loss(dB)=',agmd
970          dst=-xmin
971          phi=2.0d0*ake*dst
972          cgm=agm*cdexp(cj*phi)
973          cz2=(1.0d0+cgm)/(1.0d0-cgm)
974          write(6,*) ' S11 = ',cgm
975          write(6,*) ' Zin (norm. by 50 ohm) = ', cz2
976      c
977          stop
978          end
979      c
980      c
981      c
982      c      *      S u b r o u t i n e : s w a v e      *
983      c      *      Finding of the TM0 surface wave mode      *
984      c      *      pole, using Newton Lapson Method.      *
985      c      *      *      *      *      *      *      *
986      c
987      c      subroutine swave(epsr,ak0,d,pole,ind)
988      c      double precision epsr,ak0,d,pole,x0,aks,akz1,akz2,f,df,aa
989      c
990      x0=1.00001d0*ak0

```

```

991      aks=dsqrt(epsr)*ak0
992      do 100 i=1,200
993          akz1=dsqrt(aks*aks-x0*x0)
994          akz2=dsqrt(x0*x0-ak0*ak0)
995          f=epsr*akz2*dcos(akz1*d)-akz1*dsin(akz1*d)
996          df=epsr*dcos(akz1*d)*x0/akz2
997          &      +epsr*akz2*d*dsin(akz1*d)*x0/akz1
998          &      +x0*dsin(akz1*d)/akz1+x0*d*dcos(akz1*d)
999          aa=dabs(f/df)
1000         if(aa.lt.1.0d-4) go to 102
1001         x0=x0-f/df
1002         if(x0.lt.ak0.or.x0.gt.aks) go to 101
1003     100 continue
1004         ind=-1
1005         return
1006     102 pole=x0
1007         ind=0
1008         return
1009     101 ind=1
1010         return
1011         end
1012 c
1013 c *****
1014 c *  s u b r o u t i n e :  e x p a n s 1  *
1015 c *      Expansion mode on the circular patch *
1016 c *      ( dominant TM11 mode ) *
1017 c *****
1018         subroutine expans1(rad,beta1,beta,alfa,besja,pi,fxp11,fyp11,
1019         & fxo11,fyo11)
1020         implicit double precision (a,b,d-h,o-y)
1021         dimension zbes(0:5)
1022         mm=4
1023         zbeta=sngl(beta*rad)
1024 c
1025         call besjt(zbeta,zbes,mm,ind)
1026         fa1=0.5d0*db1e(zbes(0)-zbes(2))*beta1*beta1
1027         &      /(beta1*beta1-beta*beta)
1028         fa2=db1e(zbes(1))/beta/rad
1029         fx1=fa1*dcos(alfa)*dcos(alfa)+fa2*dsin(alfa)*dsin(alfa)
1030         fx2=fa1*dsin(alfa)*dcos(alfa)-fa2*dcos(alfa)*dsin(alfa)
1031         fx3=fa1*dsin(alfa)*dcos(alfa)-fa2*dcos(alfa)*dsin(alfa)
1032         fx4=fa1*dsin(alfa)*dsin(alfa)+fa2*dcos(alfa)*dcos(alfa)
1033         fxp11=2.0d0*pi*rad*besja*fx1
1034         fyp11=2.0d0*pi*rad*besja*fx2
1035         fxo11=2.0d0*pi*rad*besja*fx3
1036         fyo11=2.0d0*pi*rad*besja*fx4
1037         return
1038         end
1039 c
1040 c *****
1041 c *  s u b r o u t i n e :  e x p a n s 2  *
1042 c *      Expansion mode on the circular patch *
1043 c *      ( TM21 mode ) *
1044 c *****
1045         subroutine expans2(rad,beta21,beta,alfa,besjb,pi,
1046         & fxp21,fyp21,fxo21,fyo21)
1047         implicit double precision (a,b,d-h,o-y)
1048         dimension zbes2(0:6)
1049         mm=5
1050         zbeta=sngl(beta*rad)
1051 c
1052         call besjt(zbeta,zbes2,mm,ind)
1053         fa1=0.5d0*db1e(zbes2(1)-zbes2(3))*beta21*beta21
1054         &      /(beta21*beta21-beta*beta)
1055         fa2=2.0d0*db1e(zbes2(2))/beta/rad
1056         fx1=fa1*dcos(alfa)*dcos(2*alfa)+fa2*dsin(alfa)*dsin(2*alfa)

```

```

1057      fx2=fal*dsin(alfa)*dcos(2*alfa)-fa2*dcos(alfa)*dsin(2*alfa)
1058      fx3=fal*dcos(alfa)*dsin(2*alfa)-fa2*dsin(alfa)*dcos(2*alfa)
1059      fx4=fal*dsin(alfa)*dsin(2*alfa)+fa2*dcos(alfa)*dcos(2*alfa)
1060      fxp21=2.0d0*pi*rad*besjb*fx1
1061      fyp21=2.0d0*pi*rad*besjb*fx2
1062      fxo21=2.0d0*pi*rad*besjb*fx3
1063      fyo21=2.0d0*pi*rad*besjb*fx4
1064      return
1065      end
1066 c
1067 c *****
1068 c *  s u b r o u t i n e :  e x p a n s 3  *
1069 c *    Expansion mode on the circular patch *
1070 c *      ( TM31  mode ) *
1071 c *****
1072      subroutine expans3(rad,beta31,beta,alfa,besjb,pi,
1073      & fxp31,fyp31,fxo31,fyo31)
1074      implicit double precision (a,b,d-h,o-y)
1075      dimension zbes2(0:6)
1076      mm=5
1077      zbeta=sngl(beta*rad)
1078 c
1079      call besjt(zbeta,zbes2,mm,ind)
1080      fal=0.5d0*dbple(zbes2(2)-zbes2(4))*beta31*beta31
1081      & /(beta31*beta31-beta*beta)
1082      fa2=3.0d0*dbple(zbes2(3))/beta/rad
1083      fx1=fal*dcos(alfa)*dcos(3*alfa)+fa2*dsin(alfa)*dsin(3*alfa)
1084      fx2=fal*dsin(alfa)*dcos(3*alfa)-fa2*dcos(alfa)*dsin(3*alfa)
1085      fx3=fal*dcos(alfa)*dsin(3*alfa)-fa2*dsin(alfa)*dcos(3*alfa)
1086      fx4=fal*dsin(alfa)*dsin(3*alfa)+fa2*dcos(alfa)*dcos(3*alfa)
1087      fxp31=-2.0d0*pi*rad*besjb*fx1
1088      fyp31=-2.0d0*pi*rad*besjb*fx2
1089      fxo31=-2.0d0*pi*rad*besjb*fx3
1090      fyo31=-2.0d0*pi*rad*besjb*fx4
1091      return
1092      end
1093 c
1094 c *****
1095 c *  s u b r o u t i n e :  e x p a n s 2  *
1096 c *    Expansion mode on the circular patch *
1097 c *      ( TM41  mode ) *
1098 c *****
1099      subroutine expans4(rad,beta41,beta,alfa,besjd,pi,
1100      & fxp41,fyp41,fxo41,fyo41)
1101      implicit double precision (a,b,d-h,o-y)
1102      dimension zbes2(0:10)
1103      mm=9
1104      zbeta=sngl(beta*rad)
1105 c
1106      call besjt(zbeta,zbes2,mm,ind)
1107      fal=0.5d0*dbple(zbes2(3)-zbes2(5))*beta41*beta41
1108      & /(beta41*beta41-beta*beta)
1109      fa2=4.0d0*dbple(zbes2(4))/beta/rad
1110      fx1=fal*dcos(alfa)*dcos(4*alfa)+fa2*dsin(alfa)*dsin(4*alfa)
1111      fx2=fal*dsin(alfa)*dcos(4*alfa)-fa2*dcos(alfa)*dsin(4*alfa)
1112      fx3=fal*dcos(alfa)*dsin(4*alfa)-fa2*dsin(alfa)*dcos(4*alfa)
1113      fx4=fal*dsin(alfa)*dsin(4*alfa)+fa2*dcos(alfa)*dcos(4*alfa)
1114      fxp41=-2.0d0*pi*rad*besjd*fx1
1115      fyp41=-2.0d0*pi*rad*besjd*fx2
1116      fxo41=-2.0d0*pi*rad*besjd*fx3
1117      fyo41=-2.0d0*pi*rad*besjd*fx4
1118      return
1119      end

```

```

1 c =====
2 c =   B E S J T : subroutine header           =
3 c =   First kind Bessel function table         =
4 c =   ( single precision )                     =
5 c =   by YASUSHI MURAKAMI                       =
6 c =   DATE 1992.3.10                           =
7 c =====
8 c
9   subroutine besjt(x,besjn,nn,ind)
10  double precision ja0,ja1,ja2,ja3,ja4,ja5,ja6,ja7,
11  & c0,c1,c2,c3,c4,c5, k0,k1,k2,k3,k4,k5,
12  & aj0,aj1,aj2,aj3,aj4,aj5,aj6,aj7,ac0,ac1,ac2,ac3,ac4,
13  & ac5,ak0,ak1,ak2,ak3,ak4,ak5
14 c
15  dimension besjn(0:nn+1)
16 c
17 c ----- data for 0th order Bessel function -----
18  data aj0,aj1,aj2,aj3,aj4,aj5,aj6,aj7
19  & / 1.00000 0000d0 , -3.99999 9872d0 , 3.99999 7302d0,
20  & -1.77775 6060d0, .44435 84263d0, -.07092 53492d0,
21  & .00767 71853d0, -.00001 144115d0/
22  data ac0,ac1,ac2,ac3,ac4,ac5
23  & / .39884 22793d0 , -.00175 30620d0 , .00017 34300d0,
24  & -.00004 87613d0 , .00001 73565d0 , -.00000 37043d0/
25  data ak0,ak1,ak2,ak3,ak4,ak5
26  & / -.01246 69944d0 , .00045 64324d0 , -.00008 69791d0,
27  & .00003 42468d0 , -.00001 42078d0 , .00000 32312d0/
28 c
29 c ----- data for 1st order Bessel function -----
30  data ja0,ja1,ja2,ja3,ja4,ja5,ja6,ja7
31  & / 2.00000 00000d0 , -3.99999 99710d0 , 2.66666 6054d0,
32  & -.88888 39649d0 , .17775 82922d0 , .02366 16773d0,
33  & .00220 69155d0 , -.00012 89769d0/
34  data c0,c1,c2,c3,c4,c5
35  & / .39894 22819d0 , .00292 18256d0 , -.00022 32030d0,
36  & .00005 80759d0 , -.00002 00920d0 , .00000 42414d0/
37  data k0,k1,k2,k3,k4,k5
38  & / .03740 08364d0 , -.00063 90400d0 , .00010 64741d0,
39  & -.00003 98708d0 , .00001 62200d0 , -.00000 36594d0/
40 c
41 c
42  y=abs(x)
43  y4=y*y/16.0
44  if(x.eq.0.0) go to 10
45  if(y.ge.4.0) go to 11
46  besjn(0)=aj0+y4*(aj1+y4*(aj2+y4*(aj3+y4*(aj4+y4*(aj5+y4*
47  & (aj6+y4*aj7))))))
48  besjn(1)=(ja0+y4*(ja1+y4*(ja2+y4*(ja3+y4*(ja4+y4*(ja5+y4*
49  & (ja6+y4*ja7)))))))*0.25*x
50  go to 12
51 10 besjn(0)=1.0
52  besjn(1)=0.0
53  return
54 11 x4=1.0/y4
55  p0=ac0+x4*(ac1+x4*(ac2+x4*(ac3+x4*(ac4+x4*ac5))))
56  q0=4.0*(ak0+x4*(ak1+x4*(ak2+x4*(ak3+x4*(ak4+x4*ak5)))))/y
57  w0=cos( y - 0.78539 81634d0)
58  z0=sin( y - 0.78539 81634d0)
59  sx=2.0/sqrt(y)
60  besjn(0)=sx*(p0*w0-q0*z0)
61 c
62  p1=c0+x4*(c1+x4*(c2+x4*(c3+x4*(c4+x4*c5))))
63  q1=4.0*(k0+x4*(k1+x4*(k2+x4*(k3+x4*(k4+x4*k5)))))/y
64  w1=cos( y - 2.35619 4490d0)
65  z1=sin( y - 2.35619 4490d0)
66  sx=2.0/sqrt(y)

```

```
67      besjn(1)=sx*(p1*w1-q1*z1)
68      if(x.lt.0.0) besjn(1)=-besjn(1)
69 c
70 12    continue
71      if(nn.lt.1) return
72 c
73 c ----- ZEN_KA SHI_KI -----
74      do 20 i=2,nn
75          besjn(i)=2.0*(i-1)*besjn(i-1)/x-besjn(i-2)
76 20    continue
77 c
78      return
79      end
```

```

1 c      =====
2 c      =      LIBRARY HEADER : TLEDC.F      =
3 c      =      linear equation : elimination  =
4 c      =      ( double precision complex )   =
5 c      =      =                               =
6 c      =      programmed by Yasushi Murakami =
7 c      =      DATE : 1992. 3. 11             =
8 c      =====
9 c
10 c      subroutine tledc(n,a,b,x,nn,e,inder)
11 c
12 c      Linear equation A * X = B
13 c      n: to be considered, nn: definition matrix size
14 c
15 c      double complex a,b,x,pmax,aa,f,work
16 c      double precision piv,pivw,e,eps
17 c      dimension a(nn,n),b(n),x(n)
18 c
19 c      inder=0
20 c      nml=n-1
21 c      if(nml) 40,41,42
22 40      inder = -1000
23 c      write(6,640) n
24 640      format(1h0, ' N =',i5,' N should be positive in TLEDC')
25 c      return
26 41      if(a(1,1).eq.(0.0d0,0.0d0)) go to 21
27 c      x(1)=b(1)/a(1,1)
28 c      return
29 42      if(n.le.nn) go to 43
30 c      inder= -1001
31 c      write(6,642) n,nn
32 642      format(1h0, ' N =',i5,' NN =',i5,' NN should be less than N')
33 c      return
34 43      piv=0.0d0
35 c      do 1 i=1,n
36 c          do 1 j=1,n
37 c              if(cdabs(a(i,j)).le.piv) go to 1
38 c              piv=cdabs(a(i,j))
39 c              l=i
40 c              m=j
41 1      continue
42 c      if(piv.eq.0.0d0) go to 21
43 c      eps=e*piv
44 c      do 2 k=1,nml
45 c          kk=k+1
46 c          x(k)=dcmplx(float(m),0.0d0)
47 c          if(k.eq.1) go to 4
48 c          do 3 j=k,n
49 c              work=a(k,j)
50 c              a(k,j)=a(1,j)
51 3      a(1,j)=work
52 c          work=b(k)
53 c          b(k)=b(1)
54 c          b(1)=work
55 4      if(k.eq.m) go to 6
56 c          do 5 i=1,n
57 c              work=a(i,k)
58 c              a(i,k)=a(i,m)
59 5      a(i,m)=work
60 6      pmax=(1.0d0,0.0d0)/a(k,k)
61 c          piv=0.0d0
62 c          do 7 i=kk,n
63 c              aa=a(i,k)*pmax
64 c              do 8 j=kk,n
65 c                  a(i,j)=a(i,j)-a(k,j)*aa
66 c                  pivw=cdabs(a(i,j))

```

```

67         if(pivw.le.piv) go to 8
68         piv=pivw
69         m=j
70         l=i
71 8         continue
72 7         b(i)=b(i)-b(k)*aa
73         if(piv.le.eps) go to 22
74 2         continue
75         go to 10
76 22        do 20 i=kk,n
77             if(cdabs(b(i)).gt.eps) go to 25
78             b(i)=(0.0d0,0.0d0)
79 20        x(i)=dcmplx(float(i),0.0d0)
80         inder=n-kk+1
81         write(6,510) k
82         l=n-k+2
83         b(k)=b(k)/a(k,k)
84         if(k.eq.1) go to 15
85         go to 16
86 21        k=0
87         kk=1
88 25        inder=-(n-kk+1)
89         write(6,611) k
90         return
91 610       format(1h,'Ill conditioned.Rank=',i5,'Solution = X+A*Y')
92 611       format(1h,'Ill conditioned.Rank=',i5,'No further calculation')
93 c
94 10        b(n)=b(n)/a(n,n)
95         l=2
96 16        do 11 ii=1,n
97             i=n+1-ii
98             f=b(i)
99             iml=i+1
100          do 12 j=iml,n
101 12          f=f-a(i,j)*b(j)
102 11          b(i)=f/a(i,i)
103 15          do 13 ii=2,n
104              i=n+1-ii
105              m=real(x(ii))
106              work=b(i)
107              b(i)=b(m)
108              b(m)=work
109 13          continue
110          if(inder.ne.0) go to 30
111 17          do 14 i=1,n
112 14          x(i)=b(i)
113          return
114 30          do 50 j=kk,n
115              do 52 ii=1,k
116                  i=k-ii+1
117                  f=a(i,j)
118                  iml=i+1
119                  if(ii.eq.1) go to 52
120                  do 51 m=iml,k
121 51                  f=f+a(i,m)*a(m,j)
122 52                  a(i,j)=-f/a(i,i)
123 50          continue
124          do 31 i=1,n
125              do 32 j=1,n
126 32                  if(i.ge.kk.or.j.lt.kk) a(i,j)=(0.0d0,0.0d0)
127 31                  if(i.ge.kk) a(i,i)=(1.0d0,0.0d0)
128 33          m=real(x(k))
129          if(m.eq.k) go to 35
130          do 34 i=kk,n
131              work=a(k,i)
132              a(k,i)=a(m,i)

```

```
133 34      a(m,i)=work
134 35      k=k-1
135      if(k.ne.0) go to 33
136      go to 17
137      end
```