UNIVERSITY OF SOUTHAMPTON

Faculty of Mathematical Studies

Numerical synthesis of a single offset reflector with dielectric cone feed

by

Parviz Sargolzaei

June, 1996

Thesis submitted for the degree of Doctor of Philosophy

To my wife Parvaneh

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my supervisor Professor B. S. Westcott, whose intelligence, commitment and intuition made this work possible.

I am grateful to Dr. B. Craine, Dr. D. Adams, Dr. G. Danesh and R. Brown for their guidance and assistance in computing. Many thanks also go to my fellow students who have made life in the office pleasurable: Masoud, Denise, Rory and Chris.

My sincere appreciation to my wife Parvaneh, my daughters Samira, Sahar and my son Mohammad for their support and patience throughout my study.

Finally, I acknowledge the financial support of the Islamic Republic of Iran .

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF MATHEMATICAL STUDIES

Doctor of Philosophy

Numerical synthesis of a single offset reflector with dielectric cone feed

by Parviz Sargolzaei.

This thesis concentrates on the numerical solution of the boundary-value problem arising in the synthesis of a single offset reflector with dielectric cone feed. By employing a complex notation for vectors, under the assumptions of Geometrical Optics, it has been shown that the design problem may be expressed as boundary-value problem in which a second order non-linear partial differential equation of Monge-Ampère form must be solved.

For the circular aperture with Gaussian amplitude distribution, the Monge-Ampère equation is solved by a process of linearisation and finite discretisation. This gives a system of linear equations, which may solved iteratively to give the solution on a grid in the feed domain. From this solution, the reflector and dielectric surfaces may be calculated. Reflection losses together with a GO cross-polarisation calculation, are considered. A number of computed examples are presented.

Contents

1 **Introduction** **1**

1.1 Complex coordinates and notations 3

1.2 Monge-Ampère equation 9

1.3 Single reflector synthesis for direct far-field 14

1.4 Numerical Solution 15

2 **Synthesis of a single offset reflector with dielectric cone feed** **17**

2.1 Introduction 17

2.2 Transformation of the axes 18

2.3 Ray geometry 19

2.4 The mapping $\eta \longrightarrow \xi$ 20

2.5 The mapping $\eta \longrightarrow \omega$ 22

2.6 The reflector surface 24

2.7 Transformation of energy 25

2.8 Reflection losses 26

2.9 Polarisation of incident ray 29

2.10 Polarisation of transmitted ray 31

2.11 Polarisation of finally reflected ray 31

2.12 Lens design 33

2.13 Boundary condition 35

2.14 Symmetry 36

3 Analytical models **37**

3.1 Introduction 37

3.2 Conic model 38

3.3 Aperture power distribution 40

3.3.1 Spherical dielectric case 43

3.3.2 Gaussian power distribution 49

4 Linearisation and discretisation **57**

4.1 Introduction 57

4.2 Linearisation of Monge Ampère equation with dielectric feed horn 57

4.3 Linearisation of boundary condition 65

4.4 Numerical Solution 67

4.4.1 Finite Difference Scheme 68

4.5 Power density normalisation factor 70

5 Implementation of the synthesis procedure and computed results **76**

5.1 Introduction 76

5.2 Synthesis procedure 77

5.3 Compensation of the power normalisation factor 77

5.4 Computer results 79

5.5 Application of aperture power taper 79

5.6 Cross-polarisation 86

5.7 Power density transfer ratio t_{12} 90

5.8 Effect of varying input parameters 107

5.9 Conclusion 111

A Polarisation of refracted ray **112**

B The Cartesian grid **114**

B.1 Grid numbering function 115

C Cartesian finite difference scheme **119**

C.1 Derivatives of L at the internal points 119

C.2 Derivatives of L 'on the boundary points 124

C.2.1 j boundary points 124

C.2.2 i boundary points 125

C.3 Three point Lagrange interpolation for i boundary points . . . 128

D Matrix representation of the boundary-value problem **132**

E Computer programme **134**

References **203**

Chapter 1

Introduction

In the last 5-10 years satellite communications has placed stringent demands upon the electrical performance of ground station antennas. To improve upon the performance of these antennas dual offset reflector systems have been used to yield better antenna efficiency and cross-polar performance than existing axi-symmetric systems. This is because the dual offset system eliminates blockage and by through shaping the reflector surfaces will yield either high aperture efficiency or a shaped beam. For most synthesis techniques involving dual reflector antennas, the following field requirements are imposed:

1- specified amplitude distribution across the aperture,

2- uniform phase distribution across the aperture,

3- minimum cross-polarisation.

If we assume that the rim of the aperture is fixed, these three requirements can only be met in an approximate sense since there are only two degrees of freedom available in the synthesis, corresponding to the shapes of the two reflectors.

It has recently been suggested by Lier (1991) that a dielectric lens/reflector system illuminated by a feed horn is an alternative way of achieving the two degrees of freedom in the synthesis procedure, where the lens might be considered as a transparent subreflector. Such lens feeds can be made compact and mechanically robust, but possibly heavy. A broad-band phase-corrected lens feed with good cross-polar performance can be realised by a corrugated horn with a dielectric lens in the aperture (see Kildal (1984)). Alternatively, it can be implemented as a dielectric horn which has a natural built-in lens (see Lier and Aas (1985). In this feed the conical core surface may be provided with circumferential slots to provide the necessary low-dielectric layer between the core and the horn wall. This horn may also be useful for elliptical structures (see Lier (1990), Lier, Rahmat-Samii and Regaarajan (1991)) or in millimeter wave applications. Dielectric horns have been used in feeds to unshaped reflector systems by Olver and Voglis (1984), Voglis and Olver (1985).

Various shaping or synthesis techniques are available in the literature, see for example Hoerner (1978), Kildal (1990), Galindo,Imbriale and Mittra (1987), Marder (1981), Westcott (1983), Schruben(1972), Brickell, Marder and Westcott (1977), Westcott, Stevens and Brickell (1981), Westcott,Graham and Wolton (1986), Galindo and Mittra(1984), Lee, Parad and Chu, (1979), Westcott, Brickell and Wolton (1987).

Geometrical-optics (GO) is a good approximation for predicting the performance of systems with large electrical dimensions (see for example Silver(1949) chapter 4, Rusch and Potter (1970)).

In this thesis we will study the synthesis of a dielectric cone fed single reflector antenna system. Both surfaces of the dielectric cone and the reflector will be shaped under GO assumptions to yield prescribed aperture field distributions.

Unlike the work of Lier and Skyttemyr (1991) we shall take into account reflection losses at the dielectric/free space interface.

The treatment is based upon the work of Westcott (1993) who gave a theoretical method for the design synthesis under GO assumptions given aperture field specification. The relevant equations are given in Chapter 2.

In fact Westcott (1993) formulated a GO synthesis for such a single reflector system by using complex coordinates to parametrise ray and polarisation direction.

In order to introduce the method of complex coordinates we reproduce the work of Westcott (1983) in sections 1.1-1.3.

1.1 Complex coordinates and notations

Complex-coordinates are employed in the synthesis method to parametrise both ray directions and the position of points on the planar aperture. The complex notation has two advantages; firstly the resulting mathematical formulae are relatively simple, and secondly interesting implications are obtained by invoking theorems of complex variable theory. Westcott and Norris (1975), Brickell, Marder and Westcott (1977) and Westcott (1983) showed that considerable simplification in the mathematics could be achieved by parametrising ray directions, using a single complex-coordinate.

Suppose $f = u + iv$ is a complex valued function, where u and v are real functions of the real variables x and y. If we define $\eta = x + iy$, it follows that f may also be regarded as a function of the complex coordinate η. Partial derivatives of f with respect to the complex coordinate are defined by

$$\begin{aligned} f_\eta &= \frac{1}{2}(f_x - if_y) = \frac{1}{2}(u_x + v_y + i(v_x - u_y)) \\ f_{\bar\eta} &= \frac{1}{2}(f_x + if_y) = \frac{1}{2}(u_x - v_y + i(v_x + u_y) \\ f_{\eta\bar\eta} &= \frac{1}{4}(f_{xx} + f_{yy}) = \frac{1}{4}(u_{xx} + u_{yy} + i(v_{xx} + v_{yy})) = f_{\bar\eta\eta} \end{aligned} \tag{1.1}$$

where $\bar\eta$ denotes the complex conjugate of η.

When f is a real valued function of η it follows that

$$\bar{f}_\eta = f_{\bar\eta}, \tag{1.2}$$

a result which is used in connection with the mapping function L appearing in this chapter and Chapter 2.

Let the unit vector $\mathbf{p}$ be the direction of interest, s be the unit sphere centred at the origin O of $\mathbf{p}$ and introduce the Cartesian coordinate axes x, y, z as shown in Fig.1.1. Denote by P and N the points at which $\mathbf{p}$ and the z axis intersect the sphere, respectively. Under stereographic projection from N, the point P corresponds to a unique point P' in the plane z=0. Given the coordinates $(x, y, 0)$ of P', the complex coordinate η of $\mathbf{p}$ is now defined as $\eta = x + iy$. Thus the direction of $\mathbf{p}$ is parametrised by η.

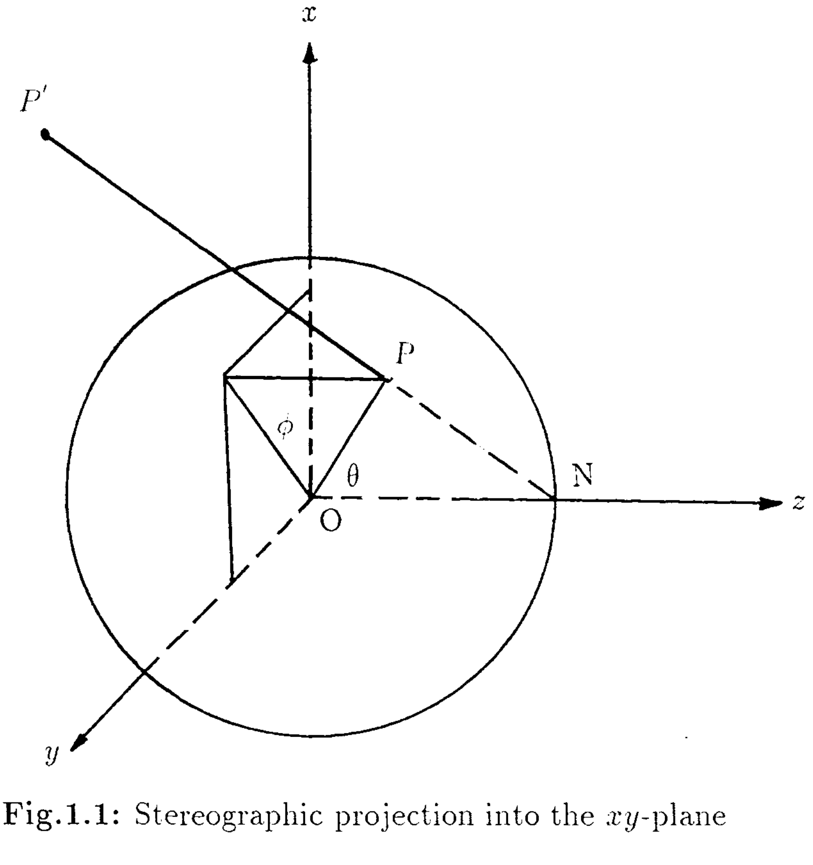


Fig.1.1: Stereographic projection into the xy-plane

Geometrical considerations may be used to show that the vector $\mathbf{p}$ is given in terms of η, by

$$\mathbf{p} = \frac{1}{|\eta|^2 + 1}(\eta + \bar{\eta}, i(\bar{\eta} - \eta), |\eta|^2 - 1). \tag{1.3}$$

The components of the unit vector $\mathbf{p}$ can also be expressed in terms of its complex coordinate η as

$$\left(\frac{2\eta}{|\eta|^2+1}, \frac{|\eta|^2-1}{|\eta|^2+1}\right). \tag{1.4}$$

The derivative of equation (1.3), in terms of the coordinate η, is given by

$$\left(|\eta|^2+1\right)^2 \mathbf{p}_\eta = \left(1-\bar{\eta}^2, -i(1+\bar{\eta}^2), 2\bar{\eta}\right), \tag{1.5}$$

where $|\eta|$ is the modulus of η. It follows that the spherical coordinates of $\mathbf{p}$ are related to η by

$$\begin{aligned} \sin\theta e^{i\phi} &= \frac{2\eta}{|\eta|^2+1}, \\ \cos\theta &= \frac{|\eta|^2-1}{|\eta|^2+1}, \end{aligned} \tag{1.6}$$

so that

$$\eta = \cot(\frac{1}{2}\theta)e^{i\phi}. \tag{1.7}$$

If a general vector $\mathbf{a}$ has components (a_1, a_2, a_3), then the vector $\mathbf{a}$ can be written as $\mathbf{a} = (a_1 + ia_2, a_3)$, and its complex coordinate η may be expressed in terms of α and a_3 as follows

$$\alpha = a_1 + ia_2 = \frac{2a\eta}{1+|\eta|^2}, \tag{1.8}$$

$$a_3 = a\frac{|\eta|^2-1}{|\eta|^2+1}, \tag{1.9}$$

where $a = |\mathbf{a}|$. From equation (1.9) we have

$$|\eta|^2 = \frac{a+a_3}{a-a_3}, \tag{1.10}$$

substituting this in equation (1.8) we obtain

$$\eta = \frac{a_1+ia_2}{a-a_3} = \frac{\alpha}{a-a_3}, \tag{1.11}$$

where a is the length of the vector $\mathbf{a}$.

With this notation, for any $\mathbf{b} = (b_1 + ib_2, b_3)$, the scalar product of $\mathbf{a}$ and $\mathbf{b}$ is

$$\mathbf{a}.\mathbf{b} = \frac{1}{2}(\alpha\bar{\beta} + \bar{\alpha}\beta) + a_3b_3, \tag{1.12}$$

where $\beta = b_1 + ib_2$ and $\bar{\alpha}, \bar{\beta}$ denote the complex conjugates of α and β.

Scalar and vector products of two unit vectors $\mathbf{p}$ and $\mathbf{t}$, with complex coordinates η, ξ respectively, can be obtained as follows

$$\mathbf{p}.\mathbf{t} = \frac{2(\eta\bar{\xi} + \bar{\eta}\xi) + (|\eta|^2 - 1)(|\xi|^2 - 1)}{(|\eta|^2 + 1)(|\xi|^2 + 1)}, \tag{1.13}$$

$$\mathbf{p} \times \mathbf{t} = \frac{2i\left[\eta\xi(\bar{\eta} - \bar{\xi}) + \eta - \xi, \eta\bar{\xi} - \bar{\eta}\xi\right]}{(|\eta|^2 + 1)(|\xi|^2 + 1)}. \tag{1.14}$$

The following lemmas will be used in later work.

Lemma 1.1 *Let f be a function with continuous first order derivatives. There exist a real value function g such that $g_\eta = f$ if, and only if, $f_{\bar{\eta}}$ is real valued.*

Proof: If g exists then $f_{\bar{\eta}} = g_{\eta\bar{\eta}} = \frac{1}{4}\Delta g$, where Δg is the Laplacian of g and is therefore real valued. Conversely, if $f_{\bar{\eta}}$ is real-valued then, from equation (1.1), $u_y = -v_x$ and consequently there exists a real-valued function ϕ such that $u = \phi_x, v = -\phi_y$. The function $g = 2\phi$ satisfies $g_\eta = f$.

Lemma 1.2 *Let $\mathbf{b} = (\beta, g)$ denote a vector with magnitude b, $\mathbf{k}$ a unit vector of complex coordinate μ and m be a scalar. Define real scalars $H = m^2 - b^2$, $K = m - \mathbf{k}.\mathbf{b}$. If $K > 0$, then the equation*

$$\mathbf{a} - a\mathbf{k} = \mathbf{b} - m\mathbf{k}, \tag{1.15}$$

has the unique solution

$$\mathbf{a} = \mathbf{b} - (\frac{H}{2K})\mathbf{k}. \tag{1.16}$$

The length a of $\mathbf{a}$ is given by

$$a = m - (\frac{H}{2K}), \tag{1.17}$$

and its direction is the unit vector of complex coordinate

$$\nu = \frac{g + m - \beta\bar{\mu}}{\bar{\beta} + (g - m)\bar{\mu}}. \tag{1.18}$$

Proof : Any solution to equation (1.15) is necessarily of the form

$$\mathbf{a} = \mathbf{b} + \lambda\mathbf{k}, \tag{1.19}$$

where λ is a scalar, and the expression is a solution if, and only if, there exists $a > 0$ such that

$$\begin{aligned} a &= \lambda + m, & (1.20)\\ a^2 &= b^2 + \lambda^2 + 2\lambda\mathbf{b}.\mathbf{k}. & (1.21) \end{aligned}$$

Substituting $\lambda + m$ for a in the second equation, we obtain

$$\lambda^2 + m^2 + 2m\lambda = b^2 + \lambda^2 + 2\lambda\mathbf{b}.\mathbf{k}.$$

Hence,

$$\lambda = \frac{-(m^2 - b^2)}{2(m - \mathbf{b}.\mathbf{k})} = \frac{-H}{2K}. \tag{1.22}$$

which is a unique solution (see Brickell and Westcott (1978)). Therefore, from (1.19) we have

$$\mathbf{a} = \mathbf{b} + (\frac{-H}{2K})\mathbf{k}, \tag{1.23}$$

and from (1.20), we have

$$\begin{aligned} a &= m + \lambda = m - \frac{H}{2K} = \frac{2Km - H}{2K} \\ &= \frac{2(m - \mathbf{k}.\mathbf{b})m - (m^2 - b^2)}{2K} = \frac{m^2 + b^2 - 2m\mathbf{k}.\mathbf{b}}{2K} \\ &= \frac{|\mathbf{b} - m\mathbf{k}|^2}{2K} > 0, & (1.24) \end{aligned}$$

where $|\mathbf{b} - m\mathbf{k}|$ is the length of the vector $\mathbf{b} - m\mathbf{k}$. It follows from equation (1.23), that

$$\begin{aligned} \mathbf{a} &= \mathbf{b} - (\frac{H}{2K})\mathbf{k} \\ &= \left(\beta - \frac{2\mu H}{2K(|\mu|^2 + 1)}, g - \frac{H(|\mu|^2 - 1)}{2K(|\mu|^2 + 1)}\right). & (1.25) \end{aligned}$$

Using equation (1.11), the complex coordinate of **a** is given by the expression

$$\nu = \frac{\alpha}{a - a_3} = \frac{K(|\mu|^2+1)\beta - \mu H}{(|\mu|^2+1)K(m-g) - H}. \tag{1.26}$$

By the definitions of H and K, we have

$$\begin{aligned} H &= m^2 - b^2 = m^2 - \beta\bar{\beta} - g^2, \\ K &= m - \mathbf{k}.\mathbf{b} = \frac{(m-g)|\mu|^2 + m + g - \mu\bar{\beta} - \beta\bar{\mu}}{|\mu|^2+1}. \end{aligned} \tag{1.27}$$

Hence

$$(|\mu|^2+1)K = (m-g)|\mu|^2 + m + g - \mu\bar{\beta} - \beta\bar{\mu}. \tag{1.28}$$

Substituting these in (1.26), we obtain

$$\nu = \frac{g + m - \beta\bar{\mu}}{\bar{\beta} + (g-m)\bar{\mu}}. \tag{1.29}$$

1.2 Monge-Ampère equation

Let O be the source point of incident rays and the center of a unit radius sphere. Suppose $\mathbf{p}$ and $\mathbf{q}$ are given unit vectors, with end points P and Q respectively, and $\mathbf{r} = r\mathbf{p}$ is the position vector of the point of reflection R (see Fig.1.2).

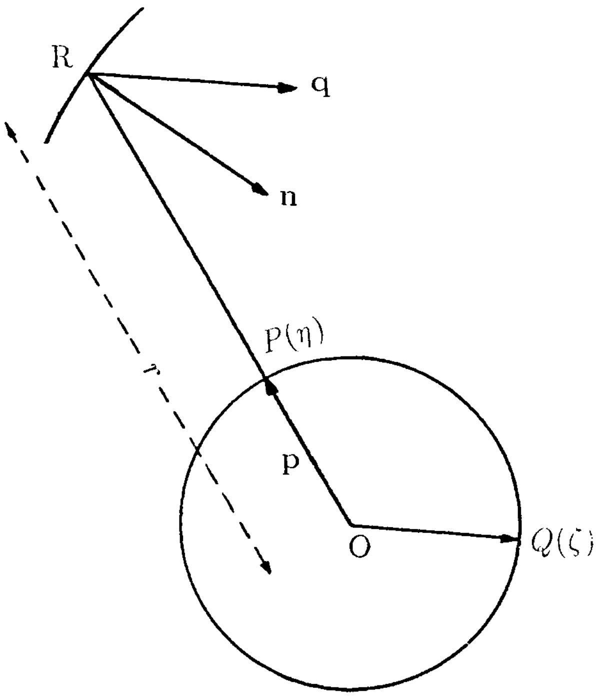


Fig 1.2: Incident and reflector ray directions

Now consider a curve $\eta = \eta(t)$ on the sphere where t is a real parameter. This curve has a tangent vector given by

$$\frac{\partial}{\partial t}(\mathbf{p}(t)) = \mathbf{p}_\eta \frac{\partial \eta}{\partial t} + \mathbf{p}_{\bar{\eta}} \frac{\partial \bar{\eta}}{\partial t}. \tag{1.30}$$

Hence

$$\begin{aligned} \left|\frac{\partial \mathbf{p}}{\partial t}\right|^2 &= \left(\mathbf{p}_\eta \frac{\partial \eta}{\partial t} + \mathbf{p}_{\bar{\eta}} \frac{\partial \bar{\eta}}{\partial t}\right) \cdot \left(\mathbf{p}_{\bar{\eta}} \frac{\partial \bar{\eta}}{\partial t} + \mathbf{p}_\eta \frac{\partial \eta}{\partial t}\right) \\ &= \frac{4}{(1+|\eta|^2)^2} \left|\frac{\partial \eta}{\partial t}\right|^2, \end{aligned} \tag{1.31}$$

and the length of $\frac{\partial \mathbf{p}}{\partial t}$ is

$$\left|\frac{\partial \mathbf{p}}{\partial t}\right| = 2\left|\frac{\partial \eta}{\partial t}\right|\left(1+|\eta|^2\right)^{-1}. \tag{1.32}$$

Suppose that $\tau : P(\eta) \longrightarrow Q(\zeta)$, is a local mapping of the unit sphere into itself, given by $\zeta = \zeta(\eta)$ in terms of the complex coordinates η of P and ζ of Q. The curve $\eta = \eta(t)$ transforms into the curve $\zeta = \zeta(\eta(t))$ and by (1.32) the length of the tangent vector to this curve is

$$2\left|\zeta_\eta \frac{\partial \eta}{\partial t} + \zeta_{\bar\eta}\frac{\partial \bar\eta}{\partial t}\right|\left(1+|\zeta|^2\right)^{-1}. \tag{1.33}$$

Using this formula the circle of unit vectors tangent to the sphere at P transforms into an ellipse of tangent vectors at Q with major and minor axes of length

$$\frac{1+|\eta|^2}{1+|\zeta|^2}\left(||\zeta_\eta| \pm |\zeta_{\bar\eta}||\right). \tag{1.34}$$

The consequence of the formula (1.34) is that τ alters elementary areas by the factor

$$\left(\frac{1+|\eta|^2}{1+|\zeta|^2}\right)^2 |J(\tau)|, \tag{1.35}$$

where $J(\tau)$, the Jacobian of τ is given by

$$J(\tau) = |\zeta_\eta|^2 - |\zeta_{\bar\eta}|^2. \tag{1.36}$$

We relate the far-field power density pattern G($\mathbf{q}$) and the source power density pattern I($\mathbf{p}$) to the geometry of the reflector under the law of geometrical optics. We know that in this case energy is conserved under reflection. Therefore it follows that the τ has to alter elementary areas on the unit sphere by the factor $F = I/G$. Hence equation (1.35) leads to one of the differential equations

$$|\zeta_\eta|^2 - |\zeta_{\bar\eta}|^2 = \pm\left(\frac{1+|\zeta|^2}{1+|\eta|^2}\right)^2 F, \tag{1.37}$$

which is not the only restriction on the mapping τ. We shall see that $\rho = ln(r)$ also satisfies a differential equation whose integrability condition gives a second condition.

Since $\mathbf{r}_\eta$ is tangential vector and $\mathbf{r} - r\mathbf{q}$ is normal to the reflector surface it follows that

$$\mathbf{r}_\eta.(\mathbf{r} - r\mathbf{q}) = 0. \tag{1.38}$$

We have $\mathbf{r} = r\mathbf{p}$, hence

$$\mathbf{r}_\eta = r_\eta\mathbf{p} + r\mathbf{p}_\eta. \tag{1.39}$$

If we substitute (1.39) in (1.38) we get

$$(r_\eta\mathbf{p} + r\mathbf{p}_\eta).(\mathbf{r} - r\mathbf{q}) = 0. \tag{1.40}$$

Thus

$$r_\eta(1 - \mathbf{p}.\mathbf{q}) - r\mathbf{p}_\eta.\mathbf{q} = 0. \tag{1.41}$$

Hence

$$\rho_\eta = \frac{r_\eta}{r} = \frac{\mathbf{p}_\eta.\mathbf{q}}{1 - \mathbf{p}.\mathbf{q}} = \frac{\mathbf{p}_\eta.\mathbf{q}}{\Lambda}, \tag{1.42}$$

where

$$\begin{aligned} \Lambda &= 1 - \mathbf{p}.\mathbf{q} \qquad (1.43) \\ &= 1 - \frac{(\eta + \bar{\eta}, i(\bar{\eta} - \eta), |\eta|^2 - 1)\big(\zeta + \bar{\zeta}, i(\bar{\zeta} - \zeta), |\zeta|^2 - 1\big)}{(1 + |\eta|^2)(1 + |\zeta|^2)} \\ &= \frac{2|\zeta - \eta|^2}{(1 + |\eta|^2)(1 + |\zeta|^2)}. \end{aligned}$$

Suppose

$$\phi(\eta, \zeta) = -ln\Lambda,$$

then

$$\begin{aligned} \rho_\eta &= \frac{r_\eta}{r} = \phi_\eta(\eta, \zeta(\eta)) = -\left(\frac{\partial\Lambda}{\partial\eta}\right)/\Lambda \\ &= \frac{1}{\zeta - \eta} + \frac{\bar{\eta}}{1 + |\eta|^2}. \qquad (1.44) \end{aligned}$$

If we define

$$L(\eta) = ln(\frac{r}{1+|\eta|^2}), \tag{1.45}$$

then

$$L_\eta = \frac{r_\eta}{r} - \frac{\bar{\eta}}{1+|\eta|^2}. \tag{1.46}$$

Substituting equation (1.44) in equation (1.46) we obtain

$$L_\eta = \frac{1}{\zeta - \eta}, \tag{1.47}$$

where L is a real-valued function and lemma 1.1 implies that $L_{\eta\bar{\eta}}$ must be real. The differentiation of equation (1.47) gives

$$L_{\eta\bar{\eta}} = -\frac{1}{(\zeta-\eta)^2}\zeta_{\bar{\eta}}. \tag{1.48}$$

For example, when the reflector is a sphere with the source at center O, then the ray reflects back along itself. Hence $\mathbf{q} = -\mathbf{p}$. This implies that

$$\left(\frac{2\zeta}{|\zeta|^2+1}, \frac{|\zeta|^2-1}{|\zeta|^2+1}\right) = \left(\frac{-2\eta}{|\eta|^2+1}, \frac{1-|\eta|^2}{|\eta|^2+1}\right). \tag{1.49}$$

Equating coefficients gives

$$\frac{2\zeta}{|\zeta|^2+1} = \frac{-2\eta}{|\eta|^2+1}, \tag{1.50}$$

$$\frac{|\zeta|^2-1}{|\zeta|^2+1} = \frac{1-|\eta|^2}{|\eta|^2+1}. \tag{1.51}$$

Equation (1.51) gives

$$|\zeta|^2 = \frac{1}{|\eta|^2}. \tag{1.52}$$

From equation (1.52) and equation (1.50) we have

$$\zeta = \frac{-\eta}{|\eta|^2} = \frac{-1}{\bar{\eta}}, \tag{1.53}$$

and hence

$$\zeta_{\bar{\eta}} = \frac{1}{\bar{\eta}^2}. \tag{1.54}$$

Substituting this in (1.48) we obtain

$$\begin{aligned} L_{\eta\bar{\eta}} &= \left(\frac{1}{\bar{\eta}^2}\right) / \left(\frac{-1}{((-1/\bar{\eta}) - \eta)^2}\right) \\ &= \frac{-1}{(1+\eta\bar{\eta})^2}, \end{aligned} \qquad (1.55)$$

which is real. This is the second condition on the mapping τ. Conversely, if τ is a local mapping of the sphere into itself satisfying the conditions (1.37) and (1.55), corresponding reflectors can be constructed. According to lemma (1.1), there exists a real-valued function $L(\eta)$ satisfying equation (1.47) and the reflector is given (to within a multiplicative constant) by

$$r = (1 + |\eta|^2)e^{L(\eta)} . \qquad (1.56)$$

For a given non-zero F two type of reflectors can be obtained, depending on the choice of sign in equation (1.37). We shall refer to the choice of the +(-) sign as the hyperbolic (elliptic) case respectively.

We can deduce from equation (1.47) that

$$\begin{aligned} L_{\eta\eta} - L_\eta^2 &= -L_\eta^2\zeta_\eta & (1.57) \\ L_{\eta\bar{\eta}} &= -L_\eta^2\zeta_{\bar{\eta}}. & (1.58) \end{aligned}$$

Consequently from equation (1.37) we obtain

$$\left|L_{\eta\eta} - L_\eta^2\right|^2 - |L_{\eta\bar{\eta}}|^2 = \pm |L\eta|^4 \left(\frac{1+|\zeta|^2}{1+|\eta|^2}\right) F. \qquad (1.59)$$

When the variable ζ is replaced by

$$\zeta = \eta + \frac{1}{L_\eta}, \qquad (1.60)$$

the equation (1.59) becomes a partial differential equation of second order for L in term of η. It is a Monge-Ampère equation of hyperbolic or elliptic type depending on the choice of the sign.

The boundary condition is formulated from the requirement that rays from the boundary of the source region must map, under a single reflection, to the boundary of the aperture. This is presented for a circular aperture in Chapter 2.

Under the law of geometrical optics total energy is conserved between the source and the aperture. This implies that the total energy radiated by the source must be equal to the total energy intercepting the aperture. If the source radiates into a region of solid angle s, and the area of the aperture is u then we have

$$\int_s I ds = \int_u G du. \tag{1.61}$$

We shall refer to equation (1.61) as the energy compatibility equation.

1.3 Single reflector synthesis for direct far-field

Westcott (1983) has shown that the formula (1.59) can be used for the synthesis of a reflector to produce a far-field radiation pattern, where $F = I(\eta)/G(\zeta)$. Here $I(\eta)$ and $G(\zeta)$ denote the source power density and far-field power density respectively and η, ζ are complex coordinates which parametrise the incident and reflected directions respectively.

The solution for $L(\eta)$ is obtained from the equation(1.59). It is evident that all the equations to be solved involve the derivatives of L rather than L. It follows that a solution for L can be established only to within an additive arbitrary constant, unless a further condition is imposed. This implies that the reflector surface is derived from equation (1.56), to within an arbitrary multiplication scale factor, by the equation:

$$r = (1 + |\zeta|^2)|L_\eta|^2 e^{-L}, \tag{1.62}$$

A development of single reflector synthesis for direct far-field specification is given by Mehler and Adatia (1986) for practical design.

1.4 Numerical Solution

A number of authors have considered the numerical synthesis of a reflector system with a plane of symmetry using the Monge-Ampère approach (see (Westcott, Stevens and Brickell (1981)) and (Mehler and Adatia (1986))). Since the Monge-Ampère equation and the boundary condition are nonlinear, linearised forms of these equations may be obtained by a generalised Newton technique. Approximate solutions to the linearised forms are then obtained iteratively. To start the iterative sequence an initial solution is required.

As in Westcott (1983) finite differences are used to calculate the required derivatives at each point of a grid. Then the synthesis equations are expressed in terms of Cartesian coordinates using the finite difference forms. The corresponding differential operators result in a matrix representation of the boundary-value problem, which can then be solved to find L at the grid points.

Starting with an initial approximation for L, the matrix equation may be written as

$$O^k L^{k+1} = V^k, \qquad k = 1, 2, 3, ..., n$$

where k is the number of the iteration, O is a sparse matrix and V is a column matrix.

This will be solved repeatedly until the successive iterates of L at each grid point converge. The arbitrary constant, implied by the initial approximation to L, is kept fixed during the iterative sequence at a grid point. This point is called an *'anchor point'*.

The work on geometrical optics for the synthesis of a single reflector with dielectric cone feed which appeared in Westcott (1993) is discussed in Chapter 2. The exact formulation, which monitors the cross-polar levels and takes into account reflection losses, leads to a nonlinear second order partial differential equation of the Monge-Ampère equation type.

The basic equations governing the design of a dielectric lens shaped to produce an arbitrary power distribution over a uniformly phased aperture are presented. We formulate analytical models which we shall use in initial solutions for the Monge-Ampère equation.

To produce far-field patterns appropriate to focussed beams with low sidelobe levels we introduce tapered power distribution over the aperture (governed by Gaussian models) in Chapter 3.

Linearisation of the Monge-Ampère equation with a dielectric feed horn is discussed in Chapter 4. For the circular aperture the power density function and corresponding normalisation factor are studied in this chapter. To maintain numerical stability, it is shown that a small change in normalisation factor may be required to reduce the power deviation at the anchor point.

The input geometrical parameter values required by the system are defined in Chapter 5 and chosen so that there is no blockage to the aperture. The computer programme which is described in Appendix E has been used successfully in the synthesis of a number of reflector systems. Computer results and some examples related to the mapping L, the location and magnitude of the maximum cross-polar level and the minimum power density transfer coefficient for a dielectric surface are presented.

It has been found that the cross-polar level depends on the shape of the dielectric surface, and the input parameters: a list of which appears in Appendix E. A range of parameter values relating to the system geometry has been considered for a single reflector with dielectric cone feed in which both the source cone and aperture are circular.

Chapter 2

Synthesis of a single offset reflector with dielectric cone feed

2.1 Introduction

The objective of this chapter is to reproduce the geometric optics (GO) synthesis of a single offset reflector with dielectric cone feed as given in Westcott(1993). For convenience the relevant equations are reproduced here. The exact formulation, which monitors the cross-polar levels and takes into account reflection losses, leads to a nonlinear second-order partial differential equation of the Monge-Ampère equation type.

The basic equations governing the design of a dielectric lens shaped to produce an arbitrary power distribution over a uniformly phased reflector aperture are presented here.

The method of complex coordinates Westcott(1983) to track ray directions and polarisation has been used and applied here to study refraction at a lens surface.

2.2 Transformation of the axes

Let x', y', z' be a new coordinate system such that $y' = -y, -z'$ is in η_0 direction, then the relation between η' and an arbitrary direction η is

$$\eta' = \frac{\eta_0 - \eta}{1 + \eta\eta_0} = x' + iy' = tan(\frac{1}{2}\theta)e^{i\phi}. \tag{2.1}$$

Hence

$$\eta = \frac{\eta_0 - \eta'}{1 + \eta_0\eta'}$$

and

$$\begin{aligned}\frac{\partial\eta'}{\partial\eta} &= \frac{-(1+\eta\eta_0) - \eta_0(\eta_0 - \eta)}{(1+\eta\eta_0)^2} = -\frac{1+\eta_0^2}{(1+\eta\eta_0)^2} \\ &= -\frac{(1+\eta_0^2)}{\left(1+\eta_0(\frac{\eta_0-\eta'}{1+\eta_0\eta'})\right)^2} = -\frac{(1+\eta_0\eta')^2}{1+\eta_0^2}.\end{aligned} \tag{2.2}$$

Since η_0 corresponds to a direction in the $y = 0$ plane, it is real and related to ψ by the $\eta_0 = \cot(\frac{1}{2}\psi)$ (see Fig.2.1).

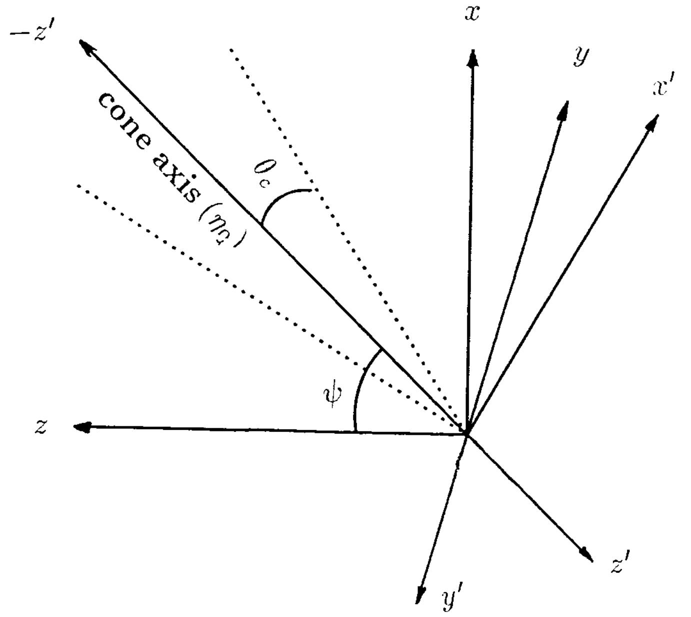


Fig.2.1: Source axes (x', y', z') in relation to system axes (x, y, z).

2.3 Ray geometry

We now apply the complex notation to the ray geometry, in the system of Figure(2.2), by assuming the dielectric is homogeneous with permeability μ_0 and a real refractive index N.

A ray emanating from the source at O is refracted at a point R and then reflected at S, in a direction parallel to the -z axis, and intercepts the aperture plane $z = d$ at a point $Q(\omega, d)$, where ω is a complex number parametrising the position of Q. We assume that all reflected rays, from the reflector, are parallel and the output aperture is a circle. Let $\mathbf{r}(\eta) = r\mathbf{p}(\eta)$ and $\mathbf{s}(\xi) = s\mathbf{t}(\xi)$, where $\mathbf{p}(\eta)$ and $\mathbf{t}(\xi)$ are unit vector of $\mathbf{r}$ and $\mathbf{s}$ parametrised by complex coordinates η and ξ (see Fig.2.2). The position vector of $\mathbf{Q}$ with respect to O is denoted by $\mathbf{v}$ and the direction of ray through Q is denoted by unit vector $\mathbf{q}(\zeta)$.

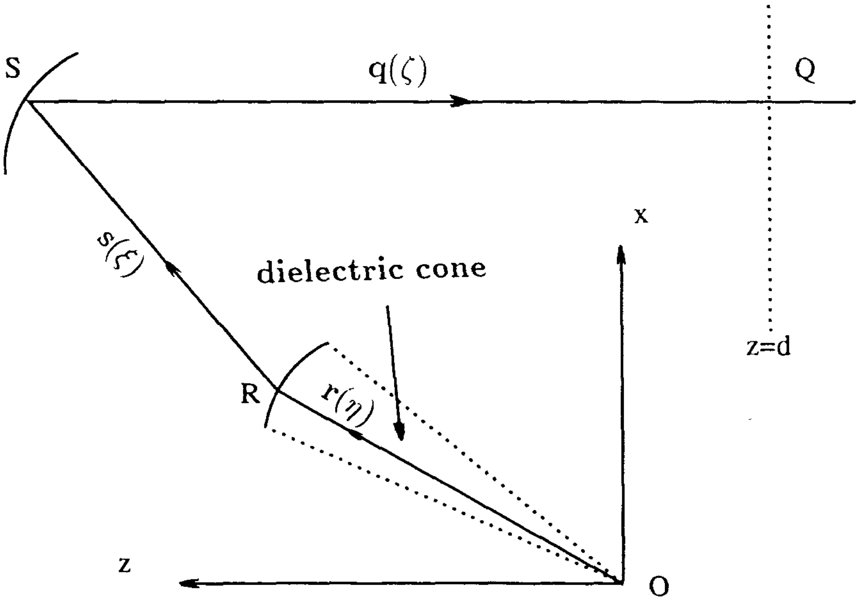


Fig 2.2: Ray geometry through the system.

2.4 The mapping $\eta \longrightarrow \xi$

Since $\mathbf{r}_\eta (= \partial r/\partial\eta)$ is a tangential at R and by the geometrical optic laws, $N\mathbf{p}-\mathbf{t}$ is a normal vector at R, we have

$$\mathbf{r}_\eta.(N\mathbf{p}-\mathbf{t}) = 0. \tag{2.3}$$

By putting $\mathbf{r} = r\mathbf{p}$ in equation (2.3) we have

$$(r\mathbf{p}_\eta + r_\eta\mathbf{p}).(N\mathbf{p}-\mathbf{t}) = 0. \tag{2.4}$$

Thus

$$-r\mathbf{p}_\eta.\mathbf{t} - r_\eta\mathbf{p}.\mathbf{t} + Nr_\eta = 0 \tag{2.5}$$

or

$$\frac{\mathbf{p}_\eta.\mathbf{t}}{N-\mathbf{p}.\mathbf{t}} = \frac{r_\eta}{r} = \rho_\eta, \tag{2.6}$$

where $\rho = ln(r)$.

Now $\Lambda = -\mathbf{p}.\mathbf{t} + N > 0$, and

$$\begin{aligned}\Lambda &= \frac{-1}{1+|\eta|^2}(2\eta, |\eta|^2-1).\frac{1}{|\xi|^2+1}(2\xi, |\xi|^2-1) + N \\ &= \frac{2|\xi-\eta|^2 - (|\eta|^2+1)(|\xi|^2+1)}{(1+|\eta|^2)(1+|\xi|^2)} + N \\ &= \frac{2|\xi-\eta|^2}{(1+|\eta|^2)(1+|\xi|^2)} - 1 + N. \end{aligned} \tag{2.7}$$

Furthermore

$$\rho_\eta = \frac{-\Lambda_\eta}{\Lambda}. \tag{2.8}$$

The refracting surface $r(\eta)$ is used to define the real valued function L by

$$L(\eta) = \ln\left(\frac{r}{1+|\eta|^2}\right). \tag{2.9}$$

Thus the equations (2.8) and (2.9)) give

$$\begin{aligned} L_\eta &= \rho_\eta - \frac{\bar\eta}{1+|\eta|^2} \\ &= \frac{2(\bar\xi-\bar\eta) + \bar\eta(1-N)(1+|\xi|^2)}{2|\xi-\eta|^2 + (N-1)(1+|\eta|^2)(1+|\xi|^2)} \\ &= \frac{2(\bar\xi-\bar\eta) + \bar\eta(1-N)(1+|\xi|^2)}{N(1+|\eta|^2)(1+|\xi|^2) - |1+\eta\bar\xi|^2 + |\xi-\eta|^2}. \end{aligned} \tag{2.10}$$

We write equation (2.10) in the form

$$\xi = \frac{A + B\bar{\xi}}{C + D\bar{\xi}}, \tag{2.11}$$

where

$$\begin{aligned}
A &= (1-N)L_\eta - (1+N)\bar{\eta}(1+\eta L_\eta), \\
B &= 2(1+\eta L_\eta), \\
C &= -2\bar{\eta} L_\eta, \\
D &= (1+N)L_\eta - (1-N)\bar{\eta}(1+\eta L_\eta).
\end{aligned}$$

If we take the complex conjugate of equation (2.11) and substitute for $\bar{\xi}$ in equation (2.11) then we have

$$a\xi^2 + 2b\xi + c = 0, \tag{2.12}$$

where

$$\begin{aligned}
a &= (1+N)L_\eta - (1-N)\bar{\eta}(1+\bar{\eta}L_{\bar{\eta}}), \\
b &= -(1+N)\eta L_\eta - (1-N)\bar{\eta}L_{\bar{\eta}} - 1, \\
c &= (1+N)\eta(1+\eta L_\eta) - (1-N)L_{\bar{\eta}}.
\end{aligned} \tag{2.13}$$

Hence

$$\xi = \frac{(1+N)\eta L_\eta + (1-N)\bar{\eta}L_{\bar{\eta}} + 1 - \sqrt{1+(1-N^2)|\bar{\eta}+(1+|\eta|^2)L_\eta|^2}}{(1+N)L_\eta - (1-N)\bar{\eta}(1+\bar{\eta}L_{\bar{\eta}})}. \tag{2.14}$$

Let

$$E = N(1+|\eta|^2)(1+|\xi|^2) - |1+\eta\bar{\xi}|^2 + |\xi-\eta|^2. \tag{2.15}$$

The fact that $E > 0$ follows from equation (2.7), because $\Lambda > 0$.

We define

$$L_\eta = S_\eta(\xi, \eta), \tag{2.16}$$

where

$$S = -\ln E(\xi,\eta), \tag{2.17}$$

is a real function, in which η, ξ can be regarded as independent variables. Further differentiation gives

$$\begin{aligned}
S_{\eta\xi} &= \frac{-2(N+1)(\bar{\xi}-\bar{\eta})^2}{E^2}, \\
S_{\eta\bar{\xi}} &= \frac{2(N-1)(1+\xi\bar{\eta})^2}{E^2}, \\
S_{\eta\bar{\eta}} &= \frac{(1-N^2)(1+|\xi|^2)^2}{E^2}, \\
S_{\eta\eta} &= S_{\eta}^2 = L_{\eta}^2, \\
S_{\bar{\eta}\bar{\eta}} &= S_{\bar{\eta}}^2 = L_{\bar{\eta}}^2.
\end{aligned} \tag{2.18}$$

2.5 The mapping $\eta \longrightarrow \omega$

The ray which firstly refracts at R and then reflects at S intercepts the planar aperture at $Q(\omega, d)$ on the plane $z = d$ (see Fig.2.2). Let l denote the optical path length from O to Q, then

$$rN + s + SQ = l. \tag{2.19}$$

But

$$\mathbf{r} + \mathbf{s} + (SQ)\mathbf{q} = \mathbf{v} \tag{2.20}$$

and therefore, from (2.19) and (2.20), we obtain

$$\mathbf{s} - s\mathbf{q} = \mathbf{v} - \mathbf{r} - (l - rN)\mathbf{q}. \tag{2.21}$$

By applying lemma (1.2), providing $K > 0$ the equation (2.21) gives

$$\mathbf{s} = \mathbf{v} - \mathbf{r} - (\frac{H}{2K})\mathbf{q}, \tag{2.22}$$

with

$$s = l - rN - \frac{H}{2K},$$

where

$$H = (l - rN)^2 - |\mathbf{v} - \mathbf{r}|^2 \tag{2.23}$$

and

$$K = l - rN - \mathbf{q}.(\mathbf{v} - \mathbf{r}). \tag{2.24}$$

If we substitute for $l - rN$ and $\mathbf{v} - \mathbf{r}$, from (2.19) and (2.20), we find that

$$K = s + SQ - \mathbf{q}.(\mathbf{s} + (SQ)\mathbf{q}) = s - \mathbf{q}.\mathbf{s} > 0. \tag{2.25}$$

To obtain the direction of $\mathbf{s}(\xi)$ we apply lemma (1.2), by choosing

$$\begin{aligned} \mathbf{k}(\mu) &= \mathbf{q}(\zeta), \mathbf{b} = (\mathbf{v} - \mathbf{r}), && (2.26) \\ (\beta, g) &= (\omega, d) - \frac{r}{1 + |\eta|^2}(2\eta, |\eta|^2 - 1) \\ &= \left(\omega - 2\eta e^L, d - e^L(|\eta|^2 - 1)\right), && (2.27) \\ m &= l - rN, \end{aligned}$$

where $e^L = r/(1 + |\eta|^2)$. Hence

$$\begin{aligned} \xi &= \frac{g + m - \beta\bar{\mu}}{\bar{\beta} + (g - m)\bar{\mu}} \\ &= \frac{l + d - rN - e^L(|\eta|^2 - 1) - (\omega - 2\eta e^L)\bar{\zeta}}{\bar{\omega} - 2\bar{\eta}e^L + (d - l + rN - e^L(|\eta|^2 - 1))\bar{\zeta}}, && (2.28) \end{aligned}$$

where l and ζ are functions of ω.

For the uniform phased aperture, in which the final reflected rays are in the negative z direction, we need to put l =constant and $\zeta = 0$.

Thus equation (2.28) gives

$$\xi = \frac{l + d - rN - e^L(|\eta|^2 - 1)}{\bar{\omega} - 2\bar{\eta}e^L}, \tag{2.29}$$

or

$$\omega - 2\eta e^L = \frac{A - e^L[(N + 1)|\eta|^2 + N - 1]}{\bar{\xi}}, \tag{2.30}$$

where $A = l + d$.

Equation (2.30), together with equation (2.14), may be used as defining the mapping η to ω. In case of no dielectric, we put $N = 1$, hence equation (2.14) gives $\xi = \eta$ and equation (2.30) gives

$$\omega = \frac{l+d}{\bar{\eta}}, \tag{2.31}$$

corresponding to a paraboloidal reflector.

2.6 The reflector surface

Equation(2.14) and (2.30) are used to define the mapping η to ω in the future work.

Once $L(\eta)$ is known then, from the equation (2.9), the dielectric surface is known and the reflector surface may be derived from formula given by

$$\mathbf{r} + \mathbf{s} = \mathbf{v} - (\frac{H}{2K})\mathbf{q}, \tag{2.32}$$

where $\mathbf{q} = (0, -1)$.

Therefore the coordinates of the point S are obtained as follows

$$\begin{aligned} \mathbf{v} - (\frac{H}{2K})\mathbf{q} &= (\omega, d) - \frac{H}{2K}(0, -1) = (w, d) + (0, \frac{H}{2K}) \\ &= (\omega, d + \frac{H}{2K}), \end{aligned} \tag{2.33}$$

where

$$\begin{aligned} H &= (l - rN)^2 - |\mathbf{v} - \mathbf{r}|^2 \\ &= [l - Ne^L(|\eta|^2 + 1)]^2 - |(\omega, d) - \frac{r}{1 + |\eta|^2}(2\eta, |\eta|^2 - 1)|^2 \\ &= [l - Ne^L(|\eta|^2 + 1)]^2 - |\omega - 2\eta e^L|^2 - [d - e^L(|\eta|^2 - 1)]^2, \end{aligned} \tag{2.34}$$

$$\begin{aligned} K &= l - rN - \mathbf{q}.(\mathbf{v} - \mathbf{r}) \\ &= l - (0, -1)(\omega - 2\eta e^L, d - e^L(|\eta|^2 - 1)) - rN \\ &= l + d - e^L[(N + 1)|\eta|^2 + N - 1] \\ &= A - e^L[(N + 1)|\eta|^2 + N - 1] \end{aligned} \tag{2.35}$$

and

$$\begin{aligned} d+\frac{H}{2K} &= \frac{2Kd+H}{2K} \\ &= \frac{2Kd+[l-Ne^L|\eta|^2-Ne^L]^2-[d-e^L|\eta|^2+e^L]^2-|\omega-2\eta e^L|^2}{2K} \\ &= \frac{2dK+K[l-d-e^L[(N-1)|\eta|^2+N+1]]-|\omega-2\eta e^L|^2}{2K} \\ &= \frac{K[A-e^L[(N-1)|\eta|^2+N+1]]-|\omega-2\eta e^L|^2}{2K}. \end{aligned} \tag{2.36}$$

2.7 Transformation of energy

Let $I(\eta)$ denote the power density of source and $G(\omega)$ the power density flow normal to the aperture. If there were no losses then, by comparing elementary areas subtended by ray tubes on a unit sphere centre O and on the aperture, it can be shown (see Westcott (1983), p.85) that

$$\frac{I(\eta)}{G(\omega)} = \frac{(1+|\eta|^2)^2}{4}\left||\omega_\eta|^2-|\omega_{\bar{\eta}}|^2\right|, \tag{2.37}$$

where the quantity between the moduli signs on the right hand side of equation (2.37) is the Jacobian of the mapping from η to ω.

To study the energy transfer from source to aperture, the reflection losses at the dielectric surface must be taken into account.

2.8 Reflection losses

A ray emanating from a point O gives rise to reflected and transmitted rays at point R (see Fig.2.3).

Under geometrical optics assumptions the vectors **p**, **t** and the normal vector

$$\mathbf{n} = \frac{N\mathbf{p} - \mathbf{t}}{|N\mathbf{p} - \mathbf{t}|} \tag{2.38}$$

are in the plane of incidence.

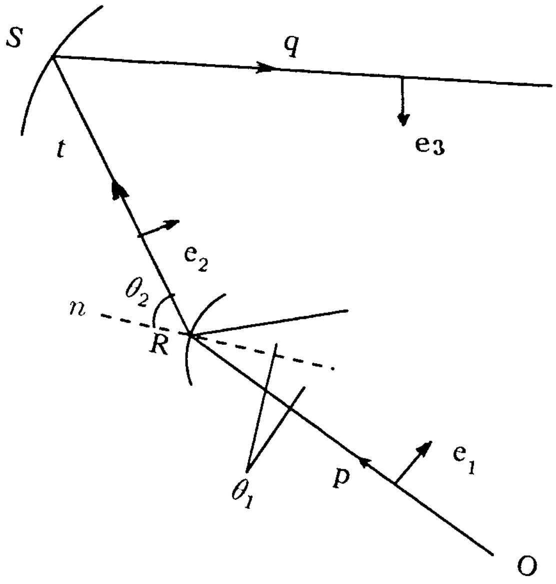


Fig.2.3: Passage of polarised rays.

Since the dielectric cone is homogeneous with refractive indices N, and unit vectors in the incident and refracted ray directions are $\mathbf{p}(\eta)$ and $\mathbf{t}(\xi)$, under the assumptions of geometrical optics, Snell's law of refraction gives

$$\mathbf{r}_\eta.(N\mathbf{p} - \mathbf{t}) = 0. \tag{2.39}$$

Using equation (2.39) we obtain,

$$N \sin \theta_1 = \sin \theta_2, \tag{2.40}$$

where θ_1 is the incidence angle (equal to the angle of reflection) and θ_2 is the angle of refraction at R.

If $\phi = \theta_2 - \theta_1$ then, $\mathbf{p}.\mathbf{t} = \cos \phi$ and $|\mathbf{p} \times \mathbf{t}| = \sin \phi$. Therefore $\mathbf{p} \times \mathbf{t}/\sin \phi$ is a unit vector perpendicular to the plane of incidence and $(\mathbf{t} - \mathbf{p}\cos \phi)/\sin \phi$ is a unit vector parallel to the plane of incidence and orthogonal to the direction of $\mathbf{p}$. So we can generally define the polarisation of an incident ray by the unit vector

$$e_1 = \frac{A(\mathbf{p} \times \mathbf{t}) + B(\mathbf{t} - \mathbf{p}\cos \phi)}{\sin \phi}, \tag{2.41}$$

where A and B are real numbers such that $A^2 + B^2 = 1$ (see Appendix A).

In order to study polarisation of the system, we examine the way the polarisation of a ray is transformed as it passes through the system in Appendix A.

If $T_\perp, T_{||}$ are transmission coefficients for the perpendicularly and parallel polarised **E** fields at the point of refraction R, then by continuity of tangential **E**, **H** field components we obtain (see Westcott (1993))

$$\begin{aligned} T_\perp &= \tfrac{2N(N - \cos \phi)}{N^2 - 1}, \\ T_{||} &= T_\perp \sec \phi, \end{aligned} \tag{2.42}$$

which we can express in terms of η and ξ.

Hence the polarisation of the transmitted ray can be written in the form

$$\mathbf{e_2} = \frac{T_\perp}{(A^2 T_\perp^2 + B^2 T_{||}^2)^{\frac{1}{2}}} [A(\mathbf{p} \times \mathbf{t}) + B(\mathbf{t} - \mathbf{p}\sec(\phi))]/\sin \phi. \tag{2.43}$$

If the area of interception of an elementary tube of rays incident on the surface $\mathbf{r}(\eta)$ at R is dA, then the fraction of power transmitted across the surface at R is the ratio

$$t_{12} = \frac{I_2 dA \cos \theta_2}{I_1 dA \cos \theta_1} = \frac{I_2 \cos \theta_2}{I_1 \cos \theta_1}, \tag{2.44}$$

where dA $\cos\theta_1, dA\cos\theta_2$ are the areas of cross-section of incident and transmitted ray tube, and I_1, I_2 are respectively the magnitudes of the Poynting vector of the incident and transmitted fields at R (see Deschamps (1972)).

It can be shown that

$$\frac{I_2}{I_1} = \frac{1}{N}(A^2T_\perp^2 + B^2T_\parallel^2). \tag{2.45}$$

We denoted the incident power density by $I(\eta)$. After crossing the interface, the new power density, ie., the transmitted power density, will be

$$t_{12}I(\eta) = \frac{(A^2T_\perp^2 + B^2T_\parallel^2)}{N}\frac{\cos\theta_2}{\cos\theta_1}I(\eta), \tag{2.46}$$

where the ratio

$$\frac{\cos\theta_1}{\cos\theta_2} = \frac{N - \cos\phi}{N\cos\phi - 1}. \tag{2.47}$$

Replacing $I(\eta)$ in (2.37) by $t_{12}I(\eta)$ as given by equation (2.46) we obtain

$$\frac{I(\eta)}{G(\omega)} = \frac{N(N-\cos\phi)(1+|\eta|^2)^2}{4(N\cos\phi - 1)(A^2T_\perp^2 + B^2T_\parallel^2)}||\omega_\eta|^2 - |\omega_{\bar\eta}|^2|, \tag{2.48}$$

where

$$\begin{aligned} ||\omega_\eta|^2 - |\omega_{\bar\eta}|^2| &= \begin{pmatrix} \omega_\eta & \omega_{\bar\eta} \\ \omega_{\bar\eta} & \bar\omega_{\bar\eta} \end{pmatrix} \\ &= ||\omega_{\eta'}|^2 - |\omega_{\bar\eta'}|^2||\eta'_\eta|^2 \\ &= ||\omega_{\eta'}|^2 - |\omega_{\bar\eta'}|^2|\frac{|1+\eta_0\eta'|^4}{(1+\eta_0^2)^2} \end{aligned} \tag{2.49}$$

and

$$1 + |\eta|^2 = \frac{(1+\eta_0^2)(1+|\eta'|^2)}{|1+\eta_0\eta'|^2}. \tag{2.50}$$

Substituting (2.49) and (2.50) in (2.48) we obtain

$$\frac{I(\eta)}{G(\omega)} = \frac{N(N-\cos\phi)(1+|\eta'|^2)^2}{4(N\cos\phi - 1)(A^2T_\perp^2 + B^2T_\parallel^2)}||\omega_{\eta'}|^2 - |\omega_{\bar\eta'}|^2|. \tag{2.51}$$

Finally we must ensure that the total power transmitted through the dielectric falls upon the aperture. Energy compatibility leads to the relation

$$\int\int_{aperture} GdXdY = \int\int_{cone} \frac{4FI}{(1+|\eta'|^2)^2}dx'dy', \tag{2.52}$$

where

$$F = \frac{(N\cos\phi - 1)(A^2T_\perp^2 + B^2T_\parallel^2)}{N(N - \cos\phi)}, \tag{2.53}$$

and we have used $\omega = X + iY, \eta' = x' + iy'$. It will convenient in the remainder of the thesis to delete the dashes from x', y', dx' and dy' so that now $\eta' = x + iy$.

If we choose $I = I_0 = 1$ then compatibility of energy gives

$$\int\int_{aperture} G dX dY = \int\int_{cone} \frac{4F}{(1 + |\eta'|^2)^2} dx dy, \tag{2.54}$$

2.9 Polarisation of incident ray

In order to study polarisation of the system, we examine the way the polarisation of a ray is transformed as it passes through the system.

Let the unit vector $\mathbf{e_1}$ be parametrised by a complex coordinate λ. Then by equation (1.4)

$$\mathbf{e_1} = \frac{(2\lambda, |\lambda|^2 - 1)}{|\lambda|^2 + 1}. \tag{2.55}$$

Since $\mathbf{e_1}$ is orthogonal to the given direction $\mathbf{p}$ then $\mathbf{e_1}.\mathbf{p} = 0$. Hence

$$2(\lambda\bar{\eta} + \bar{\lambda}\eta) + (|\lambda|^2 - 1)(|\eta|^2 - 1) = 0, \tag{2.56}$$

or

$$\frac{\lambda\bar{\eta} + 1}{\lambda - \eta} = \frac{\bar{\lambda}\eta + 1}{\bar{\lambda}\eta - \bar{\eta}}.$$

Taking

$$\epsilon = \frac{\lambda\bar{\eta} + 1}{\lambda - \eta}, \tag{2.57}$$

then we get

$$\lambda = \frac{1 + \epsilon\eta}{\epsilon - \bar{\eta}}, \tag{2.58}$$

where $|\epsilon| = 1$, but otherwise ϵ is arbitrary.

Substituting (2.58) in (2.55) we obtain

$$\mathbf{e_1} = \frac{(\bar{\epsilon} - \epsilon\eta^2, \epsilon\eta + \bar{\epsilon}\bar{\eta})}{1 + |\eta|^2}. \tag{2.59}$$

To find the value of A and B in (2.41) we multiply both side of this equation by $\mathbf{t}$, hence we have

$$\begin{aligned}\mathbf{e}_1.\mathbf{t} &= [A(\mathbf{p}\times\mathbf{t}).\mathbf{t} + B(\mathbf{t}^2 - \mathbf{p}.\mathbf{t}\cos\phi)]/\sin\phi \\ &= B(1-\cos^2\phi)/\sin\phi \\ &= B\sin\phi,\end{aligned}$$

where

$$\begin{aligned}\sin\phi &= \sqrt{1-\cos^2\phi}, \\ &= \frac{2|\xi-\eta||1+\bar{\xi}\eta|}{(1+|\eta|^2)(1+|\xi|^2)}.\end{aligned} \tag{2.60}$$

But

$$\mathbf{e}_1.\mathbf{t} = \frac{\epsilon(\xi-\eta)(1+\bar{\xi}\eta)+\bar{\epsilon}(\bar{\xi}-\bar{\eta})(1+\xi\bar{\eta})}{(1+|\eta|^2)(1+|\xi|^2)}. \tag{2.61}$$

It follows that

$$B = \frac{\epsilon(\xi-\eta)(1+\bar{\xi}\eta)+\bar{\epsilon}(\bar{\xi}-\bar{\eta})(1+\xi\bar{\eta})}{2|\xi-\eta||1+\bar{\xi}\eta|}. \tag{2.62}$$

and since $A^2+B^2=1$,

$$A = \frac{i\bar{\epsilon}(\bar{\xi}-\bar{\eta})(1+\xi\bar{\eta})-i\epsilon(\xi-\eta)(1+\bar{\xi}\eta)}{2|\xi-\eta||1+\bar{\xi}\eta|}. \tag{2.63}$$

Therefore the combination of equations (2.62) and (2.63) gives

$$A+iB = \frac{i\bar{\epsilon}(\bar{\xi}-\bar{\eta})(1+\xi\bar{\eta})}{|\xi-\eta||1+\bar{\xi}\eta|}. \tag{2.64}$$

The form of ϵ for a Huygen's source is now derived.

For the Huygen's source we have

$$\mathbf{e}_1 = \sin\phi_0\hat{\theta}_0 + \cos\phi_0\hat{\phi}_0, \tag{2.65}$$

where θ_0, ϕ_0 are the usual spherical polar coordinates referred to source axes x', y', z' taken at O with $-z'$ axis coincident with the cone axis (see Fig. 2.1) and y' axis coincident with the system $-y$ axis.

The complex parameter ϵ of equation (2.58), can be obtained as

$$\epsilon = -i\frac{(1+\eta_0\eta')}{(1+\eta_0\bar{\eta}')}, \tag{2.66}$$

where η_0 is the direction of the cone axis referred to system axes (see Westcott (1983), p.120).

When $y = 0$ (η is real) $\epsilon = -i$ and hence $\mathbf{e_1} = (i, 0)$ which is indicating polarisation in the positive y direction (negative y' direction).

2.10 Polarisation of transmitted ray

Let $\mathbf{e_2}$ be the polarisation direction of the refracted ray at R parametrised by complex coordinate μ.

Since $\mathbf{e_2}.\mathbf{t} = 0$ the complex coordinate of $\mathbf{e_2}$ is of form

$$\mu = \frac{1+\gamma\xi}{\gamma-\bar{\xi}}, \tag{2.67}$$

where $|\gamma| = 1$ and $\mathbf{e_2}$ can be expressed in the form

$$\mathbf{e_2} = \frac{(\bar{\gamma}-\gamma\xi^2, \gamma\xi+\bar{\gamma}\bar{\xi})}{1+|\xi|^2}. \tag{2.68}$$

Equation (A.13) in the Appendix A shows that

$$\gamma = \frac{\nu}{|\nu|}, \tag{2.69}$$

where

$$\nu = \epsilon(1+\bar{\xi}\eta)^2 + \bar{\epsilon}(\bar{\xi}-\bar{\eta})^2. \tag{2.70}$$

2.11 Polarisation of finally reflected ray

Since the final reflected rays are assumed to be a parallel beam in the negative z direction, the final ray direction has zero complex coordinate. The polarisation direction $\mathbf{e_3}$ of the finally reflected ray has a complex coordinate

$$\chi = \frac{\gamma\xi}{\bar{\xi}}, \tag{2.71}$$

(see Westcott (1993)). Therefore, by equation (1.4)

$$\mathbf{e_3} = (\chi, 0). \tag{2.72}$$

In the $y = 0$ plane $\gamma = -i$, hence $\chi = -i$ so that $\mathbf{e_3} = (-i, 0)$ indicating the co-polar direction for the reflected rays.

2.12 Lens design

The synthesis of a dielectric lens to produce a given power distribution over a uniformly phased aperture is now formulated, and leads to a nonlinear partial differential equation of the Monge-Ampère type.

In equations (2.18) if we put $p = S_{\eta\xi}$ and $q = S_{\eta\bar{\xi}}$ then

$$\begin{aligned} L_{\eta\eta} - S_{\eta\eta} &= p\xi_\eta + q\bar{\xi}_\eta, && (2.73)\\ L_{\bar\eta\bar\eta} - S_{\bar\eta\bar\eta} &= \bar{p}\bar{\xi}_{\bar\eta} + \bar{q}\xi_{\bar\eta}, && (2.74)\\ L_{\eta\bar\eta} - S_{\eta\bar\eta} &= p\xi_{\bar\eta} + q\bar{\xi}_{\bar\eta} = \bar{p}\bar{\xi}_\eta + \bar{q}\xi_\eta, && (2.75)\end{aligned}$$

where $L_{\eta\eta}, L_{\bar\eta\bar\eta}$ and $L_{\eta\bar\eta}$ are calculated from equation (2.10). Therefore

$$\begin{bmatrix} L_{\eta\eta} - S_{\eta\eta} & L_{\eta\bar\eta} - S_{\eta\bar\eta} \\ L_{\eta\bar\eta} - S_{\eta\bar\eta} & L_{\bar\eta\bar\eta} - S_{\bar\eta\bar\eta} \end{bmatrix} = \begin{bmatrix} p & q \\ \bar{q} & \bar{p} \end{bmatrix} \begin{bmatrix} \xi_\eta & \xi_{\bar\eta} \\ \bar{\xi}_\eta & \bar{\xi}_{\bar\eta} \end{bmatrix}. \tag{2.76}$$

Equation (2.30) gives

$$\omega - 2\eta e^L = \frac{K}{\bar{\xi}}, \tag{2.77}$$

where the real number K is defined by equation (2.35).

Now express the derivatives ξ_η and $\xi_{\bar\eta}$ in term of ω_η and $\omega_{\bar\eta}$ by using equation (2.77).

$$\begin{pmatrix} \xi_\eta & \xi_{\bar\eta} \\ \bar{\xi}_\eta & \bar{\xi}_{\bar\eta} \end{pmatrix} = \begin{pmatrix} 0 & \frac{\xi^2}{K} \\ \frac{\bar{\xi}^2}{K} & 0 \end{pmatrix} \left[\begin{pmatrix} P & Q \\ \bar{Q} & \bar{P} \end{pmatrix} - \begin{pmatrix} \omega_\eta & \omega_{\bar\eta} \\ \bar{\omega}_\eta & \bar{\omega}_{\bar\eta} \end{pmatrix} \right]. \tag{2.78}$$

where

$$\begin{aligned} P &= 2e^L(1 + \eta L_\eta) + K_\eta/\bar{\xi}, && (2.79)\\ Q &= 2\eta L_{\bar\eta} e^L + K_{\bar\eta}/\bar{\xi}. && (2.80)\end{aligned}$$

Hence equation (2.76) gives

$$\begin{pmatrix} L_{\eta\eta} - S_{\eta\eta} & L_{\eta\bar{\eta}} - S_{\eta\bar{\eta}} \\ L_{\eta\bar{\eta}} - S_{\eta\bar{\eta}} & L_{\bar{\eta}\bar{\eta}} - S_{\bar{\eta}\bar{\eta}} \end{pmatrix} =$$

$$\begin{pmatrix} p & q \\ \bar{q} & \bar{p} \end{pmatrix} \begin{pmatrix} 0 & \frac{\xi^2}{K} \\ \frac{\bar{\xi}^2}{K} & 0 \end{pmatrix} \left[\begin{pmatrix} P & Q \\ \bar{Q} & \bar{P} \end{pmatrix} - \begin{pmatrix} \omega_\eta & \omega_{\bar{\eta}} \\ \bar{\omega}_\eta & \bar{\omega}_{\bar{\eta}} \end{pmatrix} \right], \tag{2.81}$$

or

$$\begin{pmatrix} L_{\eta\eta} - S_{\eta\eta} - X & L_{\eta\bar{\eta}} - S_{\eta\bar{\eta}} - Y \\ L_{\eta\bar{\eta}} - S_{\eta\bar{\eta}} - \bar{Y} & L_{\bar{\eta}\bar{\eta}} - S_{\bar{\eta}\bar{\eta}} - \bar{X} \end{pmatrix} =$$

$$- \begin{pmatrix} p & q \\ \bar{q} & \bar{p} \end{pmatrix} \begin{pmatrix} 0 & \frac{\xi^2}{K} \\ \frac{\bar{\xi}^2}{K} & 0 \end{pmatrix} \begin{pmatrix} \omega_\eta & \omega_{\bar{\eta}} \\ \bar{\omega}_\eta & \bar{\omega}_{\bar{\eta}} \end{pmatrix}, \tag{2.82}$$

where

$$X = (p\bar{Q}\xi^2 + qP\bar{\xi}^2)/K, \tag{2.83}$$

$$Y = (p\bar{P}\xi^2 + qQ\bar{\xi}^2)/K. \tag{2.84}$$

Taking determinants of both sides of equation (2.82) we obtain

$$|L_{\eta\eta} - S_{\eta\eta} - X|^2 - (L_{\eta\bar{\eta}} - S_{\eta\bar{\eta}} - Y)^2 = \frac{|\xi|^4}{K^2} \left(|q|^2 - |p|^2\right) \left(|\omega_\eta|^2 - |\omega_{\bar{\eta}}|^2\right). \tag{2.85}$$

If we combine equation (2.48) and (2.85) we obtain a second-order nonlinear partial differential equation of the Monge-Ampère type for $L(\eta)$.

$$|L_{\eta\eta} - S_{\eta\eta} - X|^2 - (L_{\eta\bar{\eta}} - S_{\eta\bar{\eta}} - Y)^2 = \pm V \frac{I(\eta)}{G(\omega)}, \tag{2.86}$$

$$V = \frac{4\cos\theta_2 \left(A^2 T_\perp^2 + B^2 T_\parallel^2\right) |\xi|^4}{N \cos\theta_1 \left(1 + |\eta|^2\right)^2 K^2} \left(|q|^2 - |p|^2\right), \tag{2.87}$$

$S_{\eta\eta}$, $S_{\eta\bar{\eta}}$, p, q are given by equations (2.18), $T_\perp$, $T_\parallel$ by equation (2.42) and A, B by choice of source polarisation (see Appendix A). It can be shown that $V > 0$.

The solution for L yields the surface of the dielectric cone which is given by

$$r = \frac{e^L}{1 + |\eta|^2},$$

while the reflector surface is derived from equation (2.33).

The choice of the negative sign on the right hand side of equation (2.86) yields the elliptic form of the equation, which has been solved as a boundary-value problem.

2.13 Boundary condition

To set up the boundary-value problem, it will be assumed that the source cone has a semivertex θ_c and that the cone axis lies in the $y = 0$ plane inclined at an angle ψ to the positive z axis (see Fig.2.4). The circular aperture has radius R_a and lies in the plane $z = d$ with centre at $x = \omega_0$ and $y = 0$. Points on the boundary of the aperture are characterised by

$$|\omega - \omega_0|^2 = R_a^2, \tag{2.88}$$

where ω is defined by equation (2.30), and the boundary condition requires that any point on the source domain boundary should map to a corresponding point on the aperture domain boundary (see Fig. 2.4). The boundary condition is therefore imposed by combining equations (2.30) and (2.88).

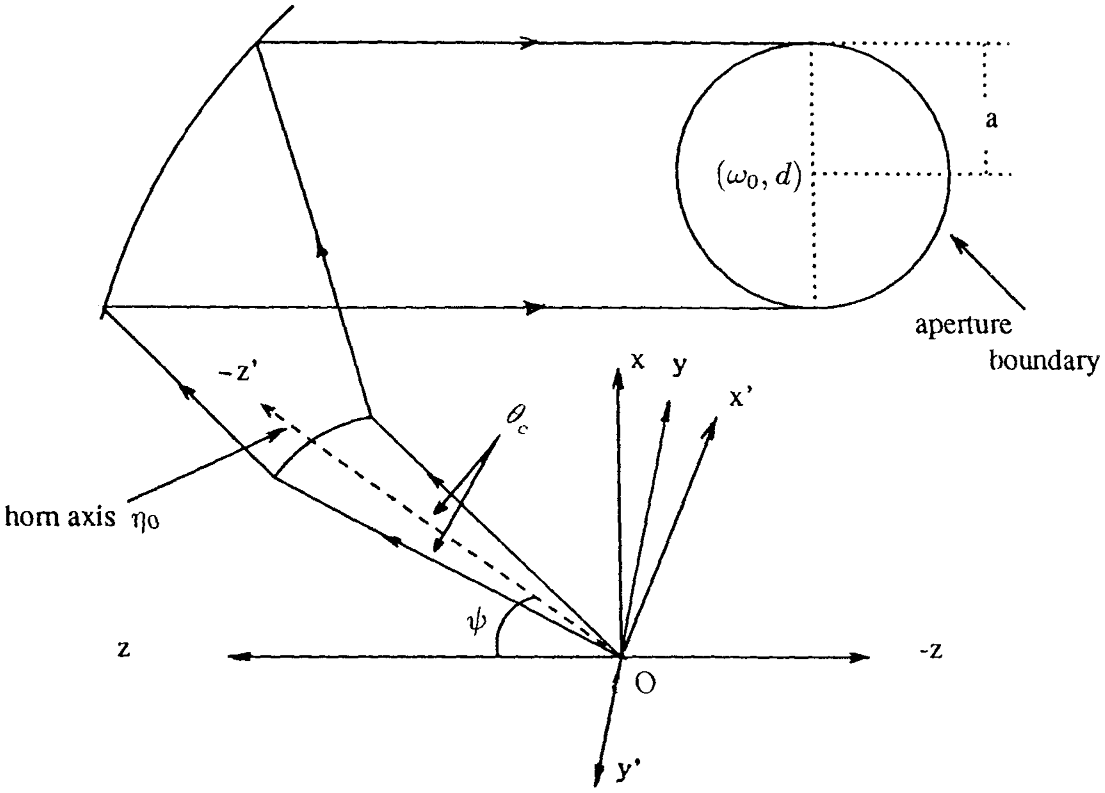


Fig.2.4: Geometry of boundary-value problem

2.14 Symmetry

If the required reflector system is symmetrical about the plane $y = 0$, the solution to the boundary-value problem need only be sought on one half of the aperture domain. This reduces the amount of computational effort required when implementing the numerical method of solution. In this case the additional boundary condition

$$L_y = 0$$

is imposed along the diameter $y = 0$ of the aperture domain.

Chapter 3

Analytical models

3.1 Introduction

In chapter 2 it was shown how the problem of synthesising a single reflector system under the assumptions of GO could be formulated in terms of a non-linear boundary-value problem defined on a region subtended by the dielectric cone feed on the unit sphere.

In this chapter we formulate analytical models which we shall use in initial solutions for the Monge-Ampère equation. Formulae for the corresponding power densities are presented.

For the special case of a spherical dielectric surface the Monge-Ampère formula becomes particularly simple and the corresponding power density ratio is presented.

To be able to shape a system to produce a tapered distribution on the aperture, we study Gaussian power distributions over the aperture.

3.2 Conic model

By geometrical considerations, the incident ray direction $\mathbf{p}$ in terms of its complex coordinate η is given by equation (1.3).

Suppose the dielectric ellipsoid axis direction is (referred to x, y, z axes)

$$(\sin\gamma, 0, -\cos\gamma),$$

where γ is the angle of rotation axis, and the incident ray direction in terms of η is

$$\frac{1}{1+|\eta|^2}\left(\eta+\bar{\eta}, i(\bar{\eta}-\eta), |\eta|^2-1\right). \tag{3.1}$$

Then

$$\cos\theta = \left[\frac{1}{1+|\eta|^2}((\eta+\bar{\eta})\sin\gamma - (|\eta|^2-1)\cos\gamma)\right], \tag{3.2}$$

where θ is the angle between the incident ray and the ellipsoid axis (see Fig.3.1).

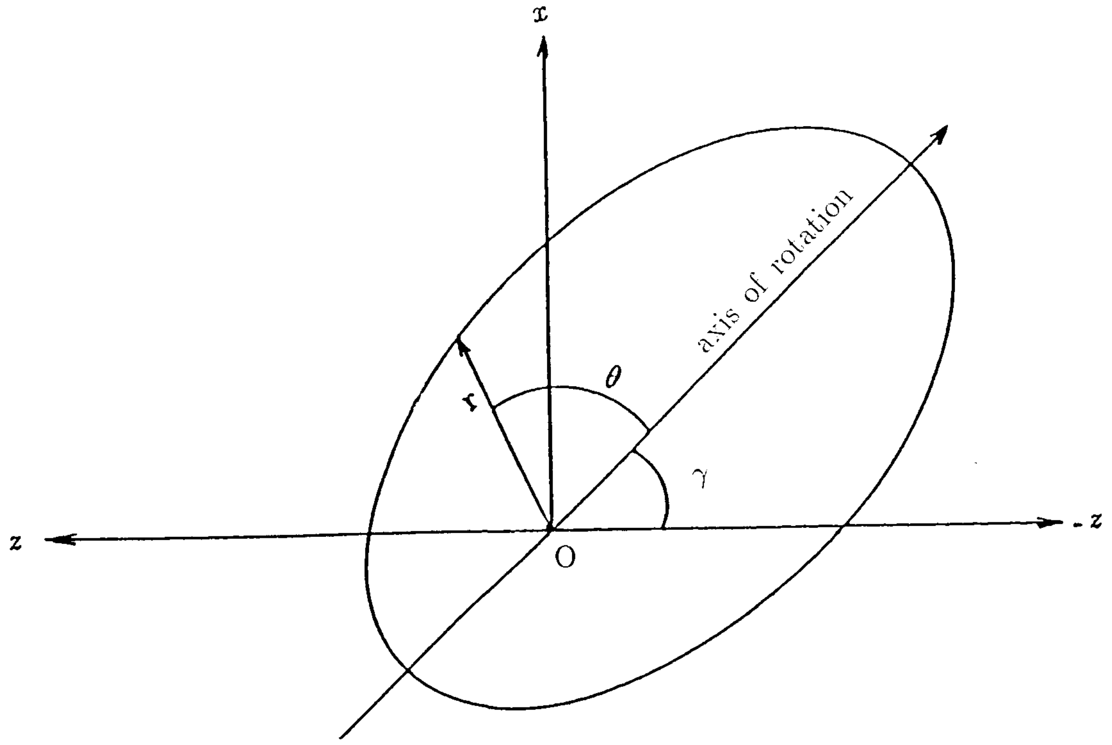


Fig. 3.1: Geometry of ellipsoid

Thus the equation of the ellipsoid is

$$\begin{aligned}\frac{k}{r} &= 1 - e\cos\theta \\ &= \frac{1 - e\cos\gamma - e\sin\gamma(\eta+\bar{\eta}) + |\eta|^2(1+e\cos\gamma)}{1+|\eta|^2}, \end{aligned} \tag{3.3}$$

where k is the semi-latus rectum, e is the eccentricity of ellipsoid and r is the length of the incident ray. Hence

$$r = \frac{k(1+|\eta|^2)}{1 - e\cos\gamma - e\sin\gamma(\eta+\bar{\eta}) + |\eta|^2(1+e\cos\gamma)}. \tag{3.4}$$

Parameter values are chosen to yield reasonably compact systems for which there is no aperture blockage.

Substituting equation (3.4) in (2.9) gives

$$\begin{aligned} L(\eta) &= ln\left(\frac{k}{1 - e\cos\gamma - e\sin\gamma(\eta+\bar{\eta}) + |\eta|^2(1+e\cos\gamma)}\right) \\ &= ln(k) - ln(1 - e\cos\gamma - e(\eta+\bar{\eta})\sin\gamma + |\eta|^2(1+e\cos\gamma)). \end{aligned} \tag{3.5}$$

Derivatives of L with respect to $\eta, \bar{\eta}, \eta\eta, \bar{\eta}\bar{\eta}$ and $\eta\bar{\eta}$ are

$$L_\eta = \frac{e\sin\gamma - \bar{\eta}(1+e\cos\gamma)}{1 - e\cos\gamma - e(\eta+\bar{\eta})\sin\gamma + |\eta|^2(1+e\cos\gamma)}, \tag{3.6}$$

$$\begin{aligned} L_{\eta\eta} &= \left(\frac{e\sin\gamma - \bar{\eta}(1+e\cos\gamma)}{1 - e\cos\gamma - e(\eta+\bar{\eta})\sin\gamma + |\eta|^2(1+e\cos\gamma)}\right)^2 \\ &= L_\eta^2, \end{aligned} \tag{3.7}$$

$$\begin{aligned} L_{\eta\bar{\eta}} &= \frac{-(1+e\cos\gamma)}{1 - e\cos\gamma - e(\eta+\bar{\eta})\sin\gamma + |\eta|^2(1+e\cos\gamma)} \\ &+ \frac{(e\sin\gamma - \bar{\eta}(1+e\cos\gamma))(e\sin\gamma - \eta(1+e\cos\gamma))}{(1 - e\cos\gamma - e(\eta+\bar{\eta})\sin\gamma + |\eta|^2(1+e\cos\gamma))^2} \\ &= \frac{-(1+e\cos\gamma)}{1 - e\cos\gamma - e(\eta+\bar{\eta})\sin\gamma + |\eta|^2(1+e\cos\gamma)} + L_\eta . L_{\bar{\eta}}, \end{aligned} \tag{3.8}$$

$$L_{\bar{\eta}} = \overline{L_\eta}, \tag{3.9}$$

$$L_{\bar{\eta}\bar{\eta}} = L_{\bar{\eta}}^2. \tag{3.10}$$

We now examine the corresponding power density ratio for aperture /feed.

3.3 Aperture power distribution

Let $I(\eta)$ denote the source power density and $G(\omega)$ the power density flow normal to the aperture. Then the power ratio G/I can be obtained from equation (2.86). Hence

$$\frac{G(\eta)}{I(\omega)}=\frac{-V}{|L_{\eta\eta}-S_{\eta\eta}-X|^2-(L_{\eta\bar{\eta}}-S_{\eta\bar{\eta}}-Y)^2}, \tag{3.11}$$

where V has given by equation (2.87).

Initial power density over aperture can be obtained from the above equation. We choose $I(\omega)=1$, hence equation (3.11) gives

$$G(\eta)=\frac{-V}{|L_{\eta\eta}-S_{\eta\eta}-X|^2-(L_{\eta\bar{\eta}}-S_{\eta\bar{\eta}}-Y)^2}. \tag{3.12}$$

Computer results for G on the plane of symmetry when (for the conic model)

γ=50	θ_c=28	k=0.14
ψ=50	N=2.6	e=0.1

are shown in table (3.1).

The graphs of aperture boundary and normalised aperture power distribution (G) in the plane of symmetry are shown in Fig.(3.2).

Table 3.1: Computer results for G on the plane of symmetry

i	ω	G	i	ω	G
1	-0.4393	1.3217	24	0.0220	0.9822
2	-0.4210	1.3114	25	0.0442	0.9643
3	-0.4026	1.3006	26	0.0665	0.9462
4	-0.3840	1.2892	27	0.0891	0.9281
5	-0.3653	1.2774	28	0.1118	0.9100
6	-0.3464	1.2651	29	0.1348	0.8918
7	-0.3274	1.2524	30	0.1580	0.8736
8	-0.3082	1.2392	31	0.1813	0.8553
9	-0.2888	1.2255	32	0.2049	0.8371
10	-0.2693	1.2115	33	0.2287	0.8190
11	-0.2496	1.1971	34	0.2526	0.8008
12	-0.2298	1.1823	35	0.2768	0.7828
13	-0.2098	1.1671	36	0.3012	0.7648
14	-0.1896	1.1516	37	0.3259	0.7469
15	-0.1692	1.1358	38	0.3507	0.7291
16	-0.1487	1.1196	39	0.3758	0.7114
17	-0.1280	1.1032	40	0.4011	0.6939
18	-0.1071	1.0866	41	0.4266	0.6766
19	-0.0861	1.0696	42	0.4523	0.6593
20	-0.0648	1.0525	43	0.4783	0.6423
21	-0.0434	1.0352	44	0.5045	0.6254
22	-0.0218	1.0177	45	0.5309	0.6088
23	0.0000	1.0000	46	0.5487	0.5978

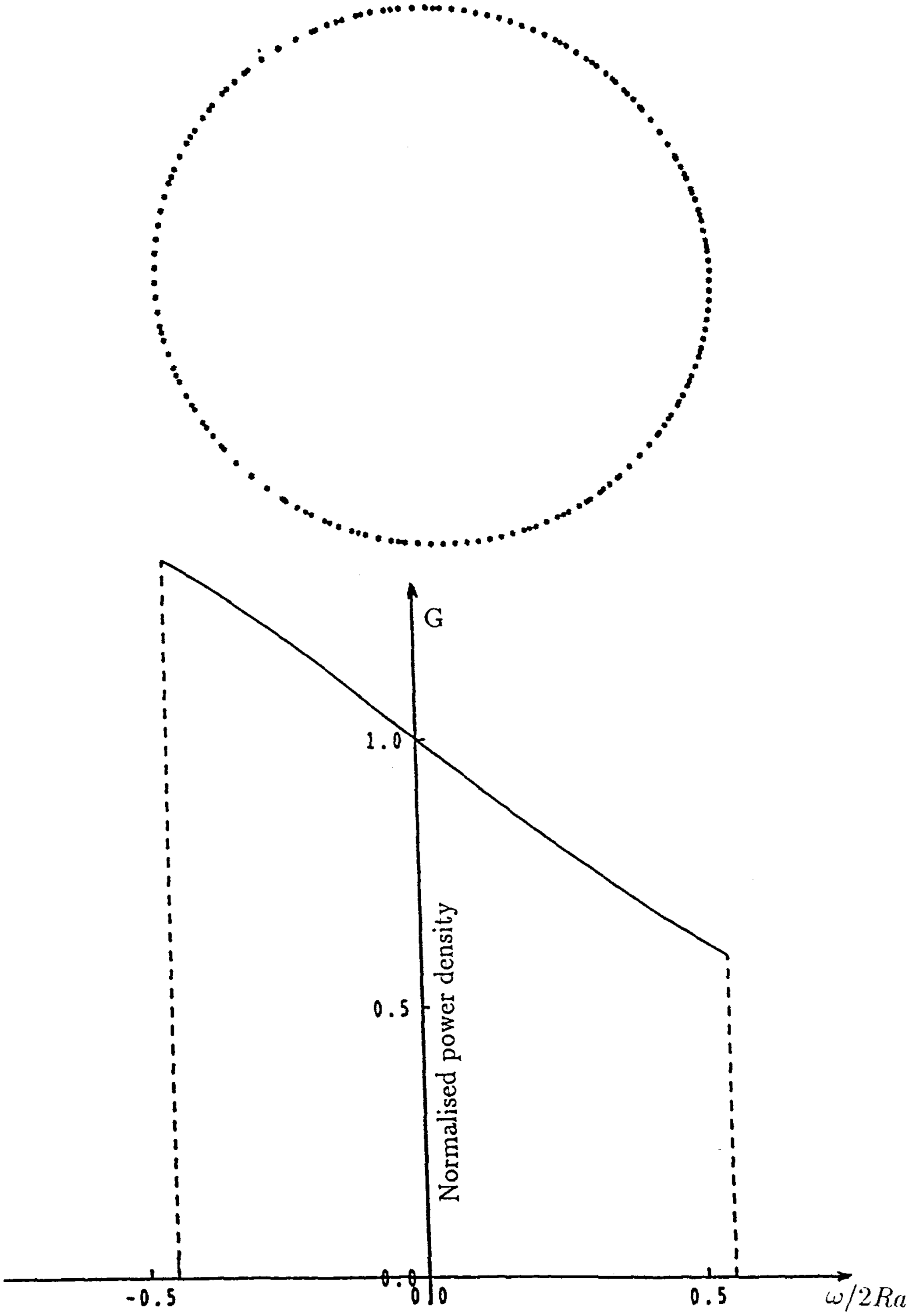


Figure 3.2: diagram showing aperture boundary and aperture power distribution in the plane of symmetry

3.3.1 Spherical dielectric case

For the spherical dielectric case, we have $e = 0$. Hence from equation (3.3) we obtain $r = k$, so equation (2.9) becomes

$$L = ln(r) - ln(1 + |\eta|^2) = ln(\frac{k}{1 + |\eta|^2}), \tag{3.13}$$

or

$$e^L = \frac{k}{1 + |\eta|^2}. \tag{3.14}$$

Derivatives of the above equation with respect to $\eta, \bar{\eta}, \eta\eta, \bar{\eta}\bar{\eta}$ and $\eta\bar{\eta}$ are

$$L_\eta = \frac{-\bar{\eta}}{1 + |\eta|^2}, \tag{3.15}$$

$$L_{\bar{\eta}} = \frac{-\eta}{1 + |\eta|^2}, \tag{3.16}$$

$$L_{\eta\eta} = \frac{\bar{\eta}^2}{(1 + |\eta|^2)^2} = L_\eta^2, \tag{3.17}$$

$$L_{\bar{\eta}\bar{\eta}} = \frac{\eta^2}{(1 + |\eta|^2)^2} = L^2\bar{\eta}, \tag{3.18}$$

$$L_{\eta\bar{\eta}} = \frac{-1}{(1 + |\eta|^2)^2}. \tag{3.19}$$

In equation (2.14) we suppose that

$$\begin{aligned} \xi_1 &= (N+1)\eta L_\eta - (N-1)\bar{\eta} L_{\bar{\eta}}, \\ \xi_2 &= 1 - \left[1 - (N^2 - 1)|\bar{\eta} + (1 + |\eta|^2)L_\eta|^2\right]^{\frac{1}{2}}, \\ \xi_3 &= (N+1)L_\eta + (N-1)\bar{\eta}(1 + \bar{\eta}L_{\bar{\eta}}). \end{aligned}$$

By substituting equations (3.15) and (3.16) in the above equations, we obtain

$$\begin{aligned} \xi_1 &= (N+1)\frac{\eta(-\bar{\eta})}{1 + |\eta|^2} - (N-1)\bar{\eta}\frac{(-\eta)}{1 + |\eta|^2} \\ &= \frac{-2|\eta|^2}{1 + |\eta|^2}, \end{aligned}$$

$$\begin{aligned}
\xi_2 &= 1-\left[1-(N^2-1)|\bar{\eta}+(1+|\eta|^2)\frac{-\bar{\eta}}{1+|\eta|^2}|^2\right]^{\frac{1}{2}} \\
&= 1-(1-(N^2-1)(\bar{\eta}-\bar{\eta})) \\
&= 0, \\
\xi_3 &= (N+1)\frac{-\bar{\eta}}{1+|\eta|^2}+(N-1)\bar{\eta}(1+\bar{\eta}\frac{-\eta}{1+|\eta|^2}) \\
&= (-(N+1)\bar{\eta}+(N-1)\bar{\eta}(1+|\eta|^2-|\eta|^2))\frac{1}{1+|\eta|^2} \\
&= \frac{-2\bar{\eta}}{1+|\eta|^2}.
\end{aligned}$$

The use of the above equations in equation (2.14) yields

$$\xi=\frac{\xi_1+\xi_2}{\xi_3}=\frac{-2|\eta|^2}{-2\bar{\eta}}=\eta. \tag{3.20}$$

It shows that η and ξ are in the same direction as expected. Therefore equation (2.30) becomes

$$\begin{aligned}
\omega &= 2\eta e^L+\frac{A-e^L[(N+1)|\eta|^2+N-1]}{\bar{\eta}} \\
&= \frac{A-e^L(|\eta|^2+1)(N-1)}{\bar{\eta}} \\
&= \frac{A-k(N-1)}{\bar{\eta}}.
\end{aligned} \tag{3.21}$$

Also it follows that

$$\begin{aligned}
&\mathbf{p}.\mathbf{t}=\cos\phi=\frac{2(|\eta|^2+|\eta|^2)+(|\eta|^2-1)(|\eta|^2-1)}{(|\eta|^2+1)(|\eta|^2+1)}=1, \\
&\phi=0, \\
&E=(N-1)(1+|\eta|^2)^2, \\
&p=S_{\eta\xi}/E^2=-2(N-1)(\eta-\eta)/E^2=0, \\
&q=S_{\eta\bar{\xi}}/E^2=2(N-1)(1+|\eta|^2)^2/E^2=\frac{2}{(N-1)(1+|\eta|^2)^2},
\end{aligned} \tag{3.22}$$

$$
\begin{aligned}
&K_\eta = -e^L(N+1)\bar{\eta} + ((N+1)|\eta|^2 + N - 1)\frac{k\bar{\eta}}{(1+|\eta|^2)^2} = \frac{-2\bar{\eta}e^L}{1+|\eta|^2},\\
&K_{\bar{\eta}} = \frac{-2\eta e^L}{1+|\eta|^2},\\
&P = 2e^L(1+\eta\frac{-\bar{\eta}}{1+|\eta|^2}) + \frac{K_\eta}{\bar{\eta}} = 0,\\
&Q = 2\eta e^L(\frac{-\eta}{1+|\eta|^2}) + \frac{K_{\bar{\eta}}}{\bar{\eta}} = \frac{-2\eta e^L}{\bar{\eta}},\\
&L_{\eta\eta} - S\eta\eta = 0,\\
&X = p\overline{Q}\eta^2 + qP\bar{\eta} = 0,\\
&T_\perp = \frac{2N(N-1)}{N^2-1} = \frac{2N}{N+1},\\
&T_{||} = \frac{T_\perp}{\cos\phi} = T_\perp,\\
&KY = qQ\bar{\eta}^2 = \frac{-4\eta\bar{\eta}e^L}{(N-1)(1+|\eta|^2)^2} = \frac{-4k|\eta|^2}{(N-1)(1+|\eta|^2)^3},\\
&S_{\eta\bar{\eta}} = \frac{(1-N^2)(1+|\eta|^2)^2}{(1-N)^2(1+|\eta|^2)^4} = \frac{N+1}{(1-N)(1+|\eta|^2)^2},\\
&L_{\eta\bar{\eta}} - S_{\eta\bar{\eta}} - Y = \frac{2}{(N-1)}\left(\frac{1}{(1+|\eta|^2)^2} + \frac{2k|\eta|^2}{(1+|\eta|^2)^3K}\right).
\end{aligned}
\tag{3.23}
$$

Substituting these equations in equation (2.87) we obtain

$$
\begin{aligned}
V &= \frac{4T_\perp^2(A^2+B^2)|\eta|^4}{N((1+|\eta|^2)^2K^2}(\frac{2}{(N-1)(1+|\eta|^2)^2})^2\\
&= \frac{16}{N}(\frac{2N}{N+1})^2\frac{|\eta|^4}{K^2(1+|\eta|^2)^6(N-1)^2}\\
&= \frac{64N|\eta|^4}{(N^2-1)^2K^2(1+|\eta|^2)^6},
\end{aligned}
\tag{3.24}
$$

where $A^2 + B^2$=1.

Hence the power density ratio equation (3.11) for the spherical case becomes

$$\begin{aligned}\frac{G(\omega)}{I(\eta)} &= \frac{V}{(L_{\eta\bar{\eta}} - S_{\eta\bar{\eta}} - Y)^2} \\ &= \frac{16N|\eta|^4}{(N+1)^2(A-k(N-1))^2(1+|\eta|^2)^2}. \end{aligned} \tag{3.25}$$

For a uniform feed power distribution we set I=1, hence equation (3.25) gives

$$\begin{aligned} G &= \frac{16N|\eta|^4}{(N+1)^2(A-k(N-1))^2(1+|\eta|^2)^2} \\ &= \frac{16N}{(N+1)^2(A-k(N-1))^2(\frac{1}{|\eta|^2}+1)^2}. \end{aligned} \tag{3.26}$$

Equation(3.21) gives

$$\frac{1}{|\eta|^2} = \frac{|\omega|^2}{(A-k(N-1))^2}. \tag{3.27}$$

Hence

$$1 + \frac{1}{|\eta|^2} = \frac{|\omega|^2 + (A-k(N-1))^2}{(A-k(N-1))^2}. \tag{3.28}$$

Substituting (3.28) in (3.26) gives

$$G = \frac{16N(A-k(N-1))^2}{(N+1)^2((A-k(N-1))^2+|\omega|^2)^2}, \tag{3.29}$$

which is a simple result for G. Computer results for G on the plane of symmetry are shown in table (3.2). Fig.3.3 shows the aperture power density graph corresponding to the spherical dielectric surface when N=2.6 and k=0.14.

Table 3.2: Table below shows results for aperture power density (G) on the plane of symmetry corresponding to spherical dielectric surface when N=2.6 and k=0.14

i	ω	G	i	ω	G
1	-0.5000	1.5464	27	0.0280	0.9698
2	-0.4797	1.5274	28	0.0483	0.9481
3	-0.4594	1.5078	29	0.0686	0.9268
4	-0.4391	1.4877	30	0.0889	0.9057
5	-0.4188	1.4671	31	0.1092	0.8849
6	-0.3985	1.4461	32	0.1295	0.8644
7	-0.3782	1.4246	33	0.1498	0.8442
8	-0.3579	1.4028	34	0.1701	0.8243
9	-0.3375	1.3806	35	0.1904	0.8048
10	-0.3172	1.3582	36	0.2107	0.7856
11	-0.2969	1.3356	37	0.2310	0.7668
12	-0.2766	1.3127	38	0.2513	0.7483
13	-0.2563	1.2897	39	0.2717	0.7301
14	-0.2360	1.2666	40	0.2920	0.7123
15	-0.2157	1.2434	41	0.3123	0.6948
16	-0.1954	1.2201	42	0.3326	0.6777
17	-0.1751	1.1969	43	0.3529	0.6610
18	-0.1548	1.1736	44	0.3732	0.6446
19	-0.1345	1.1504	45	0.3935	0.6285
20	-0.1142	1.1273	46	0.4138	0.6128
21	-0.0939	1.1043	47	0.4341	0.5975
22	-0.0736	1.0814	48	0.4544	0.5825
23	-0.0533	1.0587	49	0.4747	0.5678
24	-0.0329	1.0361	50	0.4950	0.5535
25	-0.0126	1.0138	51	0.5000	0.5501
26	0.0077	0.9917			

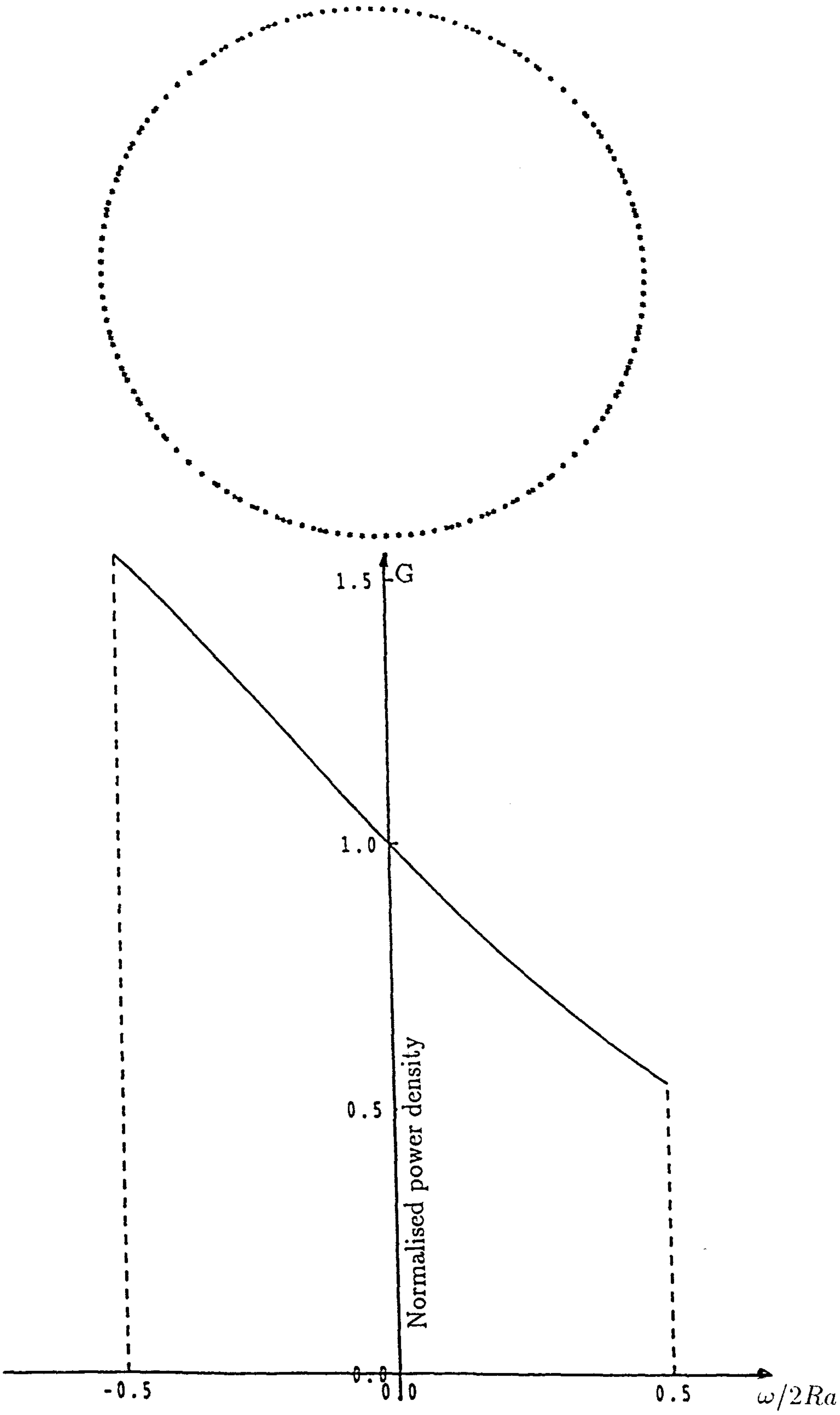


Figure 3.3:aperture power density corresponding to the spherical dielectric surface when N=2.6 and k=0.14.

3.3.2 Gaussian power distribution

In the next chapter we will shape a system to produce a tapered distribution on the aperture. We will assume this to be Gaussian power distribution defined as follows

$$G_2 = be^{-a\rho^2}, \qquad 0 \le \rho \le R_a, \tag{3.30}$$

where R_a is the aperture radius and a, b are constants.

To find a let

$$10log_{10}e^{-aR_a^2} = -D \tag{3.31}$$

thus

$$e^{-aR_a^2} = 10^{\frac{-D}{10}}. \tag{3.32}$$

where D is the power density taper in dBs at the edge of aperture. Hence

$$a = \frac{Dln10}{10R_a^2}. \tag{3.33}$$

Suppose P is the total power density over the source cone which is calculated from equation (2.54) using the initial solution, hence

$$P = \int\int_{cone} \frac{4F}{(1+|\eta'|^2)^2} dxdy, \tag{3.34}$$

where F is defined by equation (2.53) and calculated from the initial design.

To find b we replace G by G_2 in equation (2.54) and equating the equation (3.34)to the total Gaussian power distribution over aperture. Therefore

$$\begin{aligned} P &= \int\int_{aperture} G_2 dX dY \\ &= 2\pi b \int_0^{R_a} \rho e^{-a\rho^2} d\rho \\ &= \frac{\pi b}{a}\left(1 - e^{-aR_a^2}\right) \\ &= \frac{\pi b}{a}\left(1 - 10^{-D/10}\right). \end{aligned} \tag{3.35}$$

Hence

$$b = \frac{P.a}{\pi\left(1 - 10^{-D/10}\right)}. \tag{3.36}$$

Normalised Gaussian power distributions over the aperture when D=0,2,5 and 10 dBs are shown in Fig.(3.4).

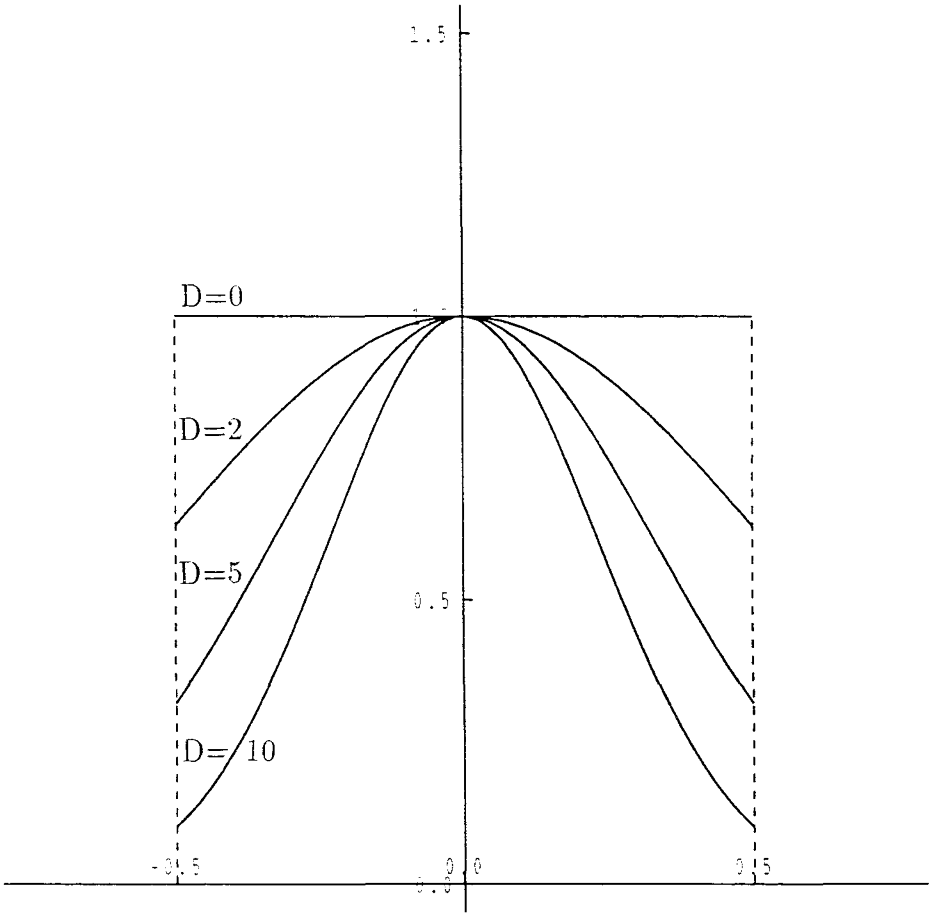


Figure 3.4: Gaussian power distributions over aperture when D=0,2,5 and 10 dBs

The following examples refer to an analytical solution in which the lens surface is designed to be part of a conic surface. The boundary condition is not imposed in these examples evidenced by the non-circular aperture boundary for large e. **Example 3.1**

An illustration (on the plane of symmetry) of the single reflector offset fed by a lens when

γ=79	k=0.14	N=1.7
ψ=45	e=0.3	A=1
θ_c=30		

is shown in Fig.(3.5). The mapping between η and ω is illustrated in Fig.(3.6).

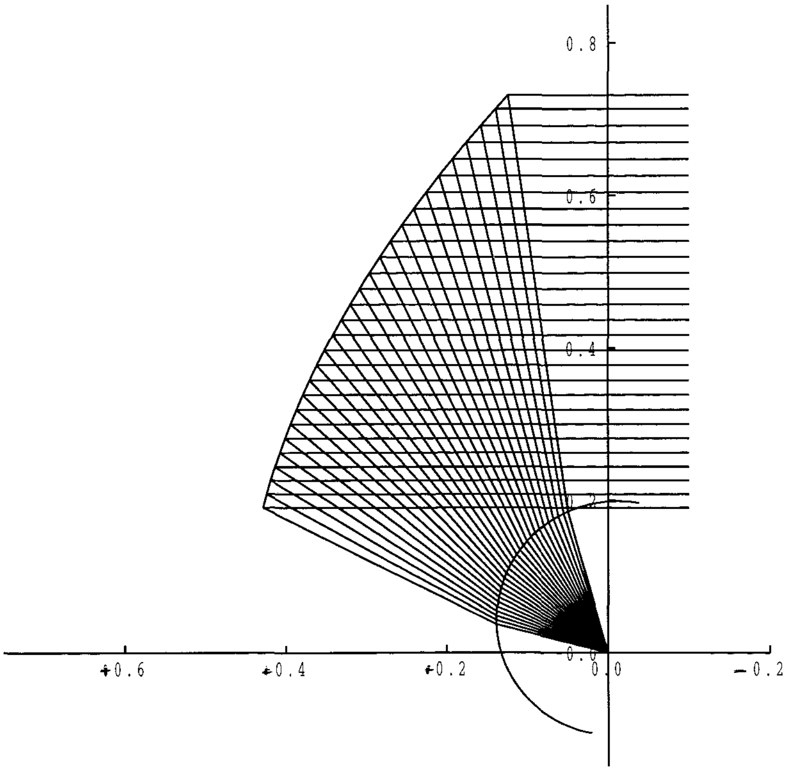


Figure 3.5: Ray diagram in plane of symmetry of the single reflector.

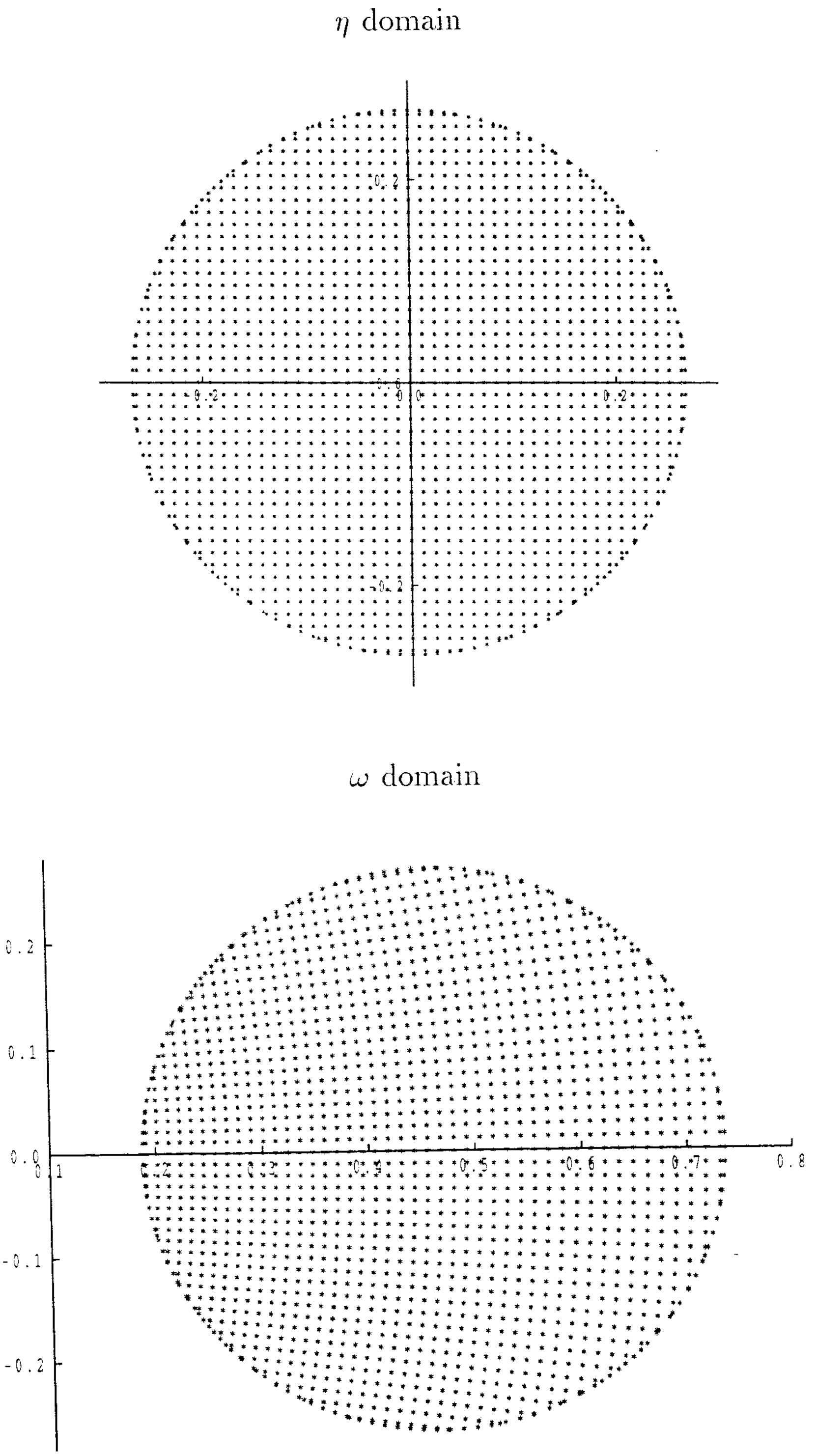


Figure 3.6: Mapping η → ω for example 3.1

Example 3.2 Illustration (on the plane of symmetry) of the single reflector offset fed by a lens when

γ=50	θ_c=28	k=0.14
ψ=50	N=2.6	e=0.1

is shown in Fig.(3.7). The mapping between η and ω is illustrated in Fig.(3.8).

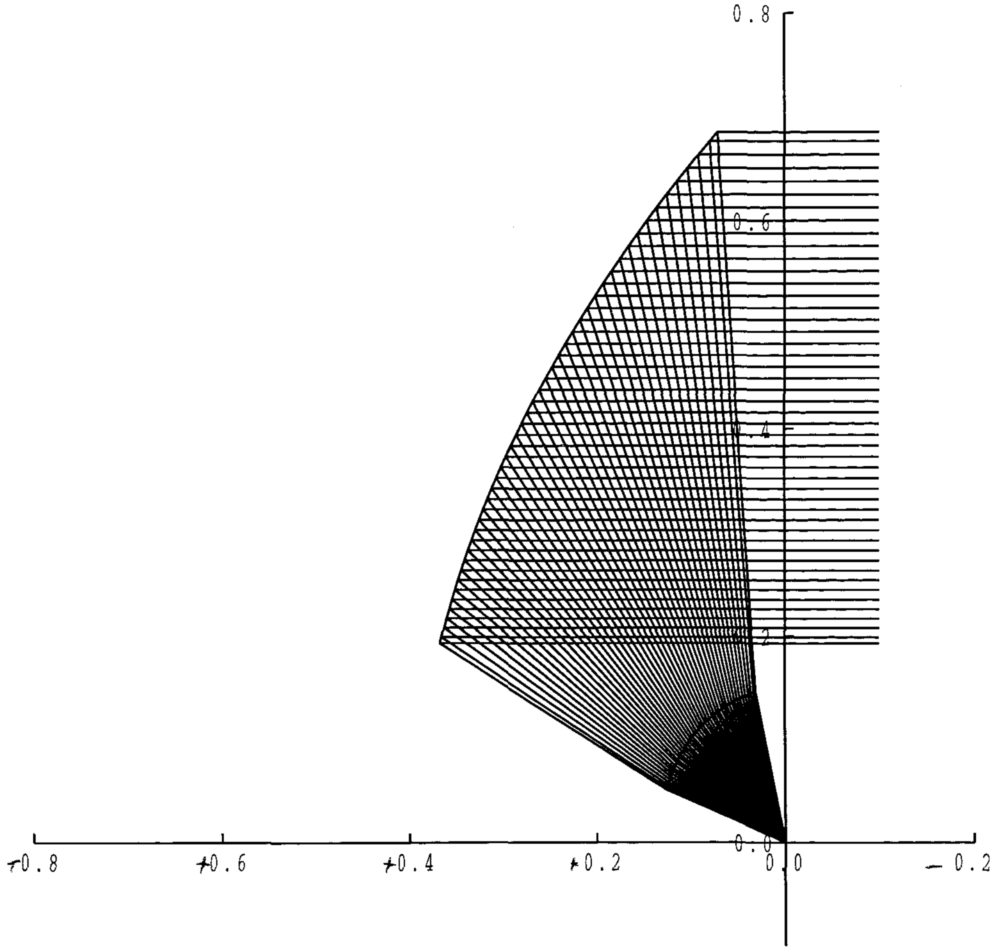
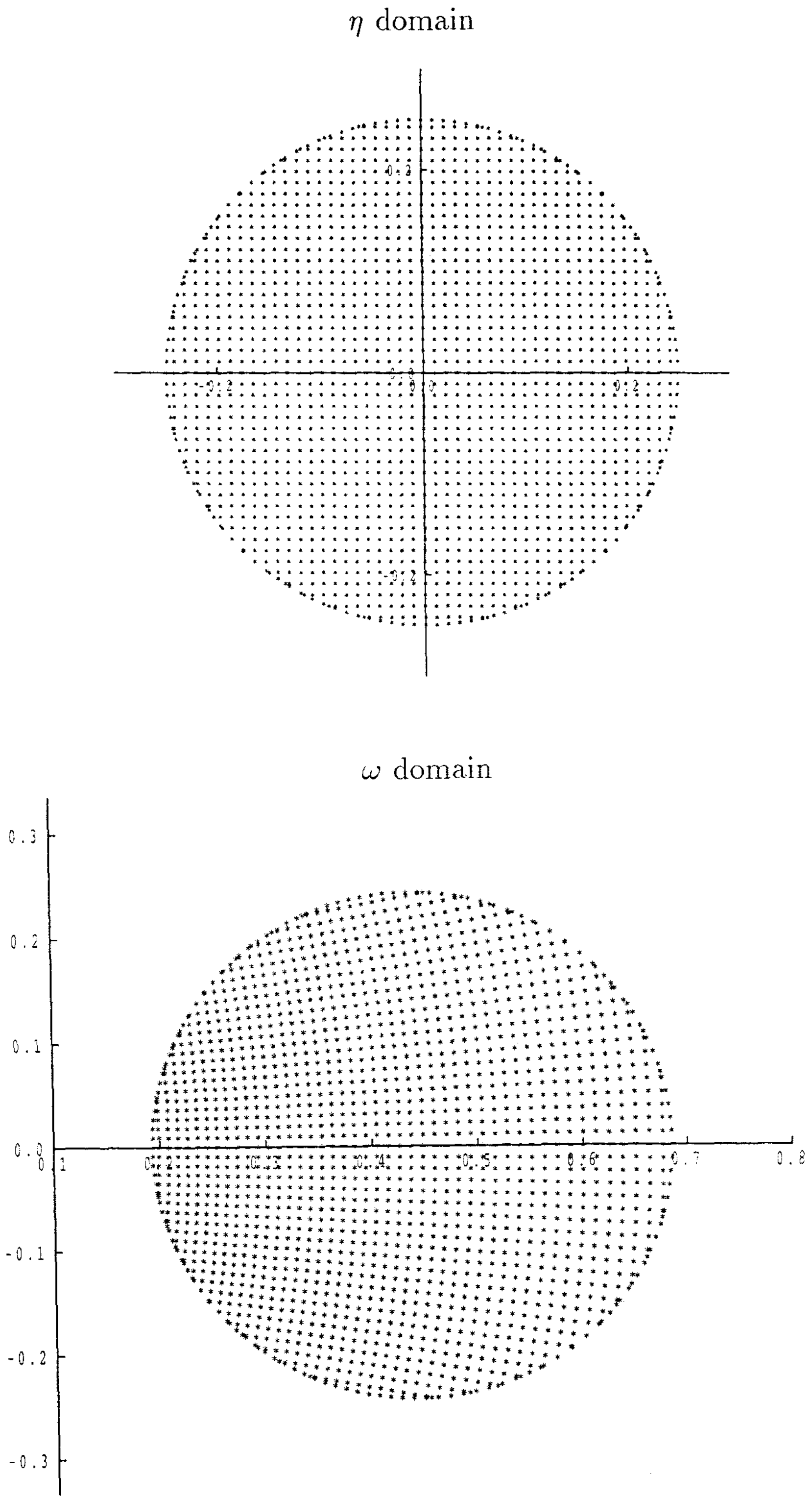


Figure 3.7: Ray diagram in plane of symmetry of the single reflector for example (3.2).


η domain

ω domain

Figure 3.8: Mapping $\eta \rightarrow \omega$ for

example 3.2

Example 3.3 Illustration of the single reflector offset fed by spherical lens ($e = 0$, hence $r = k$ and $\eta = \xi$) is shown in Fig.(3.9). The mapping between η and ω is illustrated in Fig.(3.10).

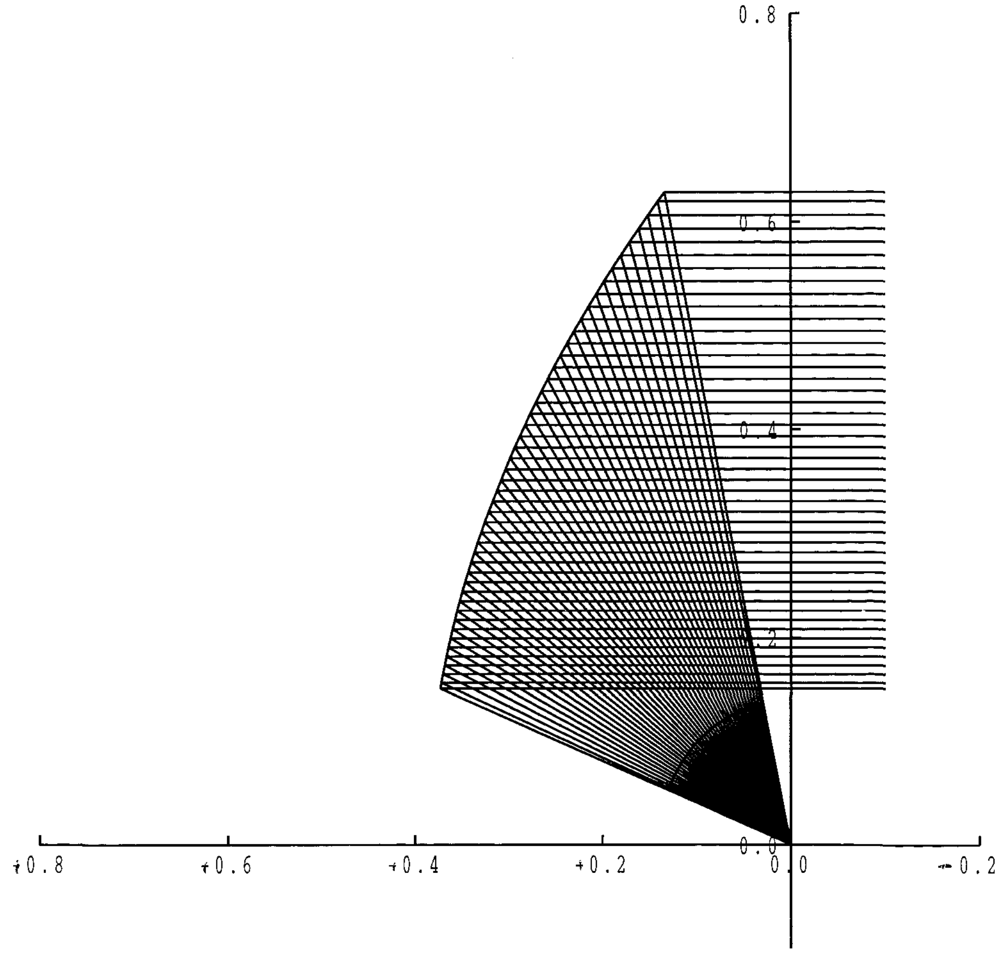
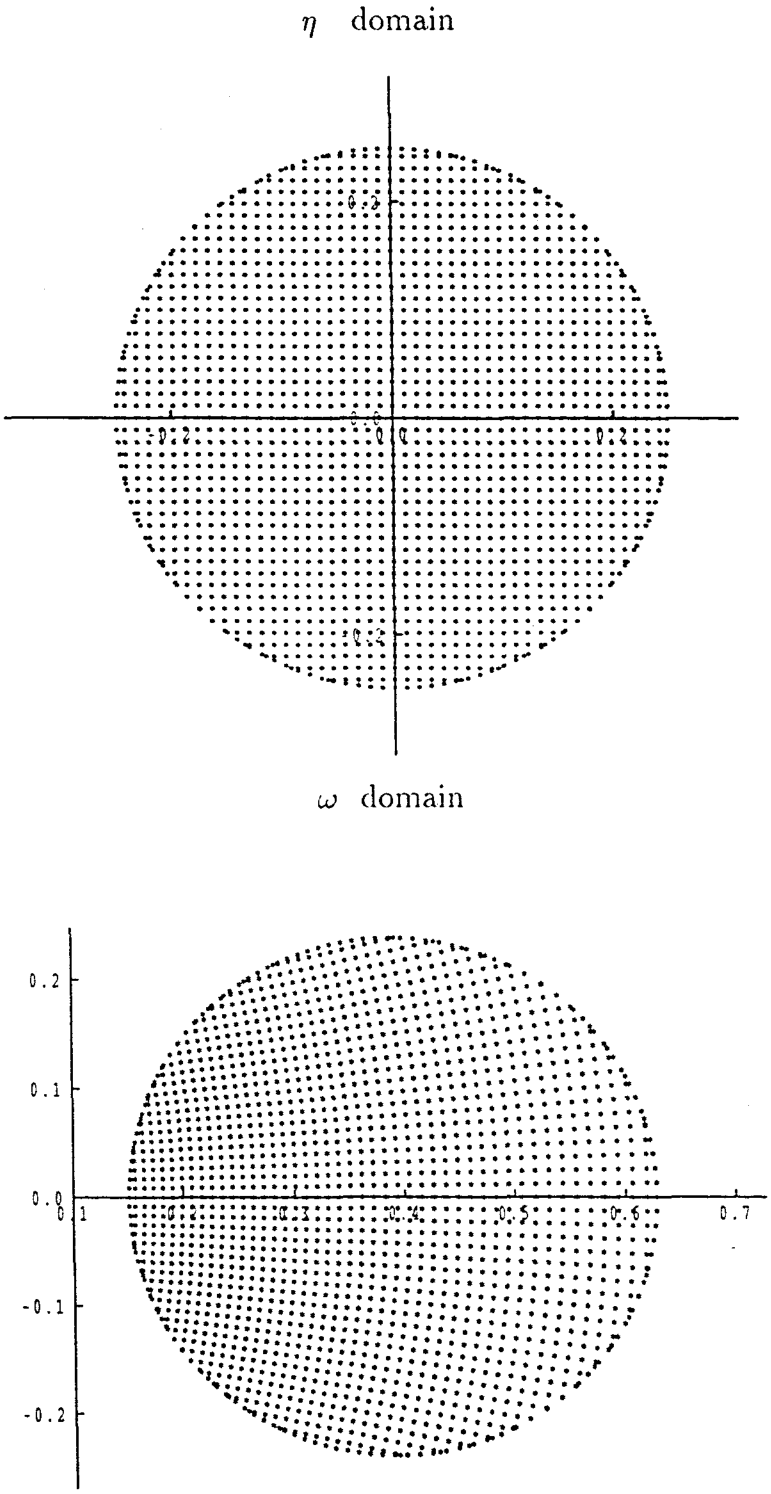


Figure 3.9: Illustration of the single reflector offset fed by spherical lens for example (3.2)



η domain

ω domain

Figure 3.10: Mapping $\eta \rightarrow \omega$ for example 3.3

Chapter 4

Linearisation and discretisation

4.1 Introduction

For the numerical solution of nonlinear Monge-Ampère equation and its boundary condition linearised forms of these equations are obtained by a generalised Newton technique.

Finite differences are used to calculate the required derivatives at each point of a grid. Then the synthesis equations are expressed in terms of Cartesian coordinates using the finite difference forms. The corresponding differential operators which result in a matrix representation of the boundary-value problem, to be solved to find L at the grid points, is formulated in this chapter.

The development of a specified power density function over a circular aperture iterated from an initial distribution is discussed.

4.2 Linearisation of Monge Ampère equation with dielectric feed horn

The Monge-Ampère equation (2.86) and the boundary condition (2.88) are nonlinear, but linearised forms of these equations may be obtained by using a generalised Newton technique.

We follow Rall (1969) and represent a nonlinear equation in operator form

$$F(L) = 0 \tag{4.1}$$

If L^0 is a close approximation to L then

$$L \approx L^0 - \frac{F(L^0)}{F'(L^0)} \tag{4.2}$$

gives a new approximation to L.

If we associate L^0 and L in the above expression with the k and $k+1$ th iterates L^k and L^{k+1} respectively, then from (4.2) we have

$$L^{k+1} = L^k - \frac{F(L^k)}{F'(L^k)} \tag{4.3}$$

or

$$F'(L^k)L^{k+1} = F'(L^k)L^k - F(L^k). \tag{4.4}$$

To apply this technique to equation (2.86) we set

$$\begin{aligned} \varphi &= |L_{\eta\eta} - S_{\eta\eta} - X|^2 - (L_{\eta\bar\eta} - S_{\eta\bar\eta} - Y)^2 + V\frac{I(\eta)}{G(\omega)} \qquad (4.5) \\ &= (L_{\eta\eta} - S_{\eta\eta} - X)(L_{\bar\eta\bar\eta} - S_{\bar\eta\bar\eta} - \bar X) - (L_{\eta\bar\eta} - S_{\eta\bar\eta} - Y)^2 + V\frac{I(\eta)}{G(\omega)}. \end{aligned}$$

The Monge Ampère equation is of course $\varphi = 0$ (see equation (2.86)). Thus the linearised form of equation (4.5) becomes

$$\begin{aligned} &\left[\frac{\partial\varphi}{\partial L_{\eta\eta}}\right] L_{\eta\eta}^{k+1} + \left[\frac{\partial\varphi}{\partial L_{\bar\eta\bar\eta}}\right] L_{\bar\eta\bar\eta}^{k+1} + \left[\frac{\partial\varphi}{\partial L_{\eta\bar\eta}}\right] L_{\eta\bar\eta}^{k+1} \\ &+ \left[\frac{\partial\varphi}{\partial L_{\eta}}\right] L_{\eta}^{k+1} + \left[\frac{\partial\varphi}{\partial L_{\bar\eta}}\right] L_{\bar\eta}^{k+1} + \left[\frac{\partial\varphi}{\partial L}\right] L^{k+1} = DEL, \end{aligned} \tag{4.6}$$

where

$$\begin{aligned} DEL &= \left[\frac{\partial\varphi}{\partial L_{\eta\eta}}\right] L_{\eta\eta}^{k} + \left[\frac{\partial\varphi}{\partial L_{\bar\eta\bar\eta}}\right] L_{\bar\eta\bar\eta}^{k} + \left[\frac{\partial\varphi}{\partial L_{\eta\bar\eta}}\right] L_{\eta\bar\eta}^{k} + \left[\frac{\partial\varphi}{\partial L_{\eta}}\right] L_{\eta}^{k} \\ &\quad + \left[\frac{\partial\varphi}{\partial L_{\bar\eta}}\right] L_{\bar\eta}^{k} + \left[\frac{\partial\varphi}{\partial L}\right] L^{k} - \varphi^k. \end{aligned} \tag{4.7}$$

Let

$$
\begin{aligned}
A &= \left[\frac{\partial\varphi}{\partial L_{\eta\eta}}\right] = \left[L_{\bar\eta\bar\eta} - S_{\bar\eta\bar\eta} - \overline{X}\right], \\
B &= \left[\frac{\partial\varphi}{\partial L_{\bar\eta\bar\eta}}\right] = \left[L_{\eta\eta} - S_{\eta\eta} - X\right], \\
C &= \left[\frac{\partial\varphi}{\partial L_{\eta\bar\eta}}\right] = -2\left[L_{\eta\bar\eta} - S_{\eta\bar\eta} - Y\right], \\
D &= \left[\frac{\partial\varphi}{\partial L_{\eta}}\right] = -\left[L_{\bar\eta\bar\eta} - S_{\bar\eta\bar\eta} - \overline{X}\right]\left[\frac{\partial S_{\eta\eta}}{\partial L_\eta} + \frac{\partial X}{\partial L_\eta}\right] \\
&\quad -\left[L_{\eta\eta} - S_{\eta\eta} - X\right]\left[\frac{\partial \overline{X}}{\partial L_\eta}\right] + 2\left[L_{\eta\bar\eta} - S_{\eta\bar\eta} - Y\right]\left[\frac{\partial S_{\eta\bar\eta}}{\partial L_\eta} + \frac{\partial Y}{\partial L_\eta}\right] \\
&\quad +\left[\frac{\partial V}{\partial L_\eta}.\frac{I(\eta)}{G(\omega)}\right] \\
&= -A\left[\frac{\partial S_{\eta\eta}}{\partial L_\eta} + \frac{\partial X}{\partial L_\eta}\right] - B\left[\frac{\partial \overline{X}}{\partial L_\eta}\right] - C\left[\frac{\partial S_{\eta\bar\eta}}{\partial L_\eta} + \frac{\partial Y}{\partial L_\eta}\right] + \left[\frac{\partial V}{\partial L_\eta}.\frac{I(\eta)}{G(\omega)}\right], \\
E &= \left[\frac{\partial\varphi}{\partial L_{\bar\eta}}\right] = -\left[L_{\bar\eta\bar\eta} - S_{\bar\eta\bar\eta} - \overline{X}\right]\left[\frac{\partial X}{\partial L_{\bar\eta}}\right] \\
&\quad -\left[L_{\eta\eta} - S_{\eta\eta} - X\right]\left[\frac{\partial S_{\bar\eta\bar\eta}}{\partial L_{\bar\eta}} + \frac{\partial \overline{X}}{\partial L_{\bar\eta}}\right] \\
&\quad +2\left[L_{\eta\bar\eta} - S_{\eta\bar\eta} - Y\right]\left[\frac{\partial S_{\eta\bar\eta}}{\partial L_{\bar\eta}} + \frac{\partial Y}{\partial L_{\bar\eta}}\right] + \left[\frac{\partial V}{\partial L_{\bar\eta}}.\frac{I(\eta)}{G(\omega)}\right] \\
&= -A\left[\frac{\partial X}{\partial L_{\bar\eta}}\right] - B\left[\frac{\partial S_{\bar\eta\bar\eta}}{\partial L_{\bar\eta}} + \frac{\partial \overline{X}}{\partial L_{\bar\eta}}\right] - C\left[\frac{\partial S_{\eta\bar\eta}}{\partial L_{\bar\eta}} + \frac{\partial Y}{\partial L_{\bar\eta}}\right] + \left[\frac{\partial V}{\partial L_{\bar\eta}}.\frac{I(\eta)}{G(\omega)}\right]
\end{aligned}
$$

and

$$
\begin{aligned}
F &= \left[\frac{\partial\varphi}{\partial L}\right] = -\left[L_{\bar\eta\bar\eta} - S_{\bar\eta\bar\eta} - \overline{X}\right]\left[\frac{\partial X}{\partial L}\right] - \left[L_{\eta\eta} - S_{\eta\eta} - X\right]\left[\frac{\partial \overline{X}}{\partial L}\right] \\
&\quad +2\left[L_{\eta\bar\eta} - S_{\eta\bar\eta} - Y\right]\left[\frac{\partial Y}{\partial L}\right] + \left[\frac{\partial V}{\partial L}.\frac{I(\eta)}{G(\omega)}\right] \\
&= -A\left[\frac{\partial X}{\partial L}\right] - B\left[\frac{\partial \overline{X}}{\partial L}\right] - C\left[\frac{\partial Y}{\partial L}\right] + \left[\frac{\partial V}{\partial L}.\frac{I(\eta)}{G(\omega)}\right].
\end{aligned}
$$

The derivatives

$$
\begin{gathered}
\frac{\partial X}{\partial L_\eta}, \frac{\partial \overline{X}}{\partial L_\eta}, \frac{\partial X}{\partial L_{\bar\eta}}, \frac{\partial \overline{X}}{\partial L_{\bar\eta}}, \frac{\partial X}{\partial L}, \frac{\partial \overline{X}}{\partial L}, \\
\frac{\partial S_{\eta\eta}}{\partial L_\eta}, \frac{\partial S_{\eta\bar\eta}}{\partial L_\eta}, \frac{\partial S_{\bar\eta\bar\eta}}{\partial L_{\bar\eta}}, \frac{\partial S_{\eta\bar\eta}}{\partial L_{\bar\eta}},
\end{gathered}
$$

$$\frac{\partial Y}{\partial L_\eta}, \frac{\partial Y}{\partial L_{\bar\eta}}, \frac{\partial Y}{\partial L},$$
$$\frac{\partial V}{\partial L_\eta}, \frac{\partial V}{\partial L_{\bar\eta}} \frac{\partial V}{\partial L}$$

have been obtained algebraically using Maple.

Substituting from above equations in (4.6) and (4.7) gives

$$A\ L^{k+1}_{\eta\eta} + B\ L^{k+1}_{\bar\eta\bar\eta} + C\ L^{k+1}_{\eta\bar\eta} + D\ L^{k+1}_{\eta} + E\ L^{k+1}_{\bar\eta} + F\ L^{k+1} = DEL, \qquad (4.8)$$

where

$$DEL = A\ L^{k}_{\eta\eta} + B\ L^{k}_{\bar\eta\bar\eta} + C\ L^{k}_{\eta\bar\eta} + D\ L^{k}_{\eta} + E\ L^{k}_{\bar\eta} + F\ L^{k} - \varphi^k. \qquad (4.9)$$

By transforming the system $(x\ y\ z)$ to $(x'\ y'\ z')$ then $\eta \rightarrow \eta'$ and we obtain

$$L_\eta = \left(\frac{\partial L}{\partial \eta}\right) = \left(\frac{\partial L}{\partial \eta'}\right)\left(\frac{\partial \eta'}{\partial \eta}\right) = \left(\frac{\partial \eta'}{\partial \eta}\right) L_{\eta'}, \qquad (4.10)$$

$$\begin{aligned}
L_{\eta\eta} &= \frac{\partial}{\partial \eta}(L_\eta) = \frac{\partial}{\partial \eta'}(L_\eta)\left(\frac{\partial \eta'}{\partial \eta}\right) = \left(\frac{\partial \eta'}{\partial \eta}\right)\left[\frac{\partial}{\partial \eta'}\left((\frac{\partial \eta'}{\partial \eta})L_{\eta'}\right)\right] \\
&= \left(\frac{\partial \eta'}{\partial \eta}\right)\left[\frac{\partial}{\partial \eta'}\left(\frac{\partial \eta'}{\partial \eta}\right)L_{\eta'} + \left(\frac{\partial \eta'}{\partial \eta}\right)\left(\frac{\partial}{\partial \eta'}L_{\eta'}\right)\right] \\
&= \left(\frac{\partial \eta'}{\partial \eta}\right)\left[\frac{\partial}{\partial \eta'}\left(\frac{-(1+\eta'\eta_0)^2}{(1+\eta_0^2)}\right)L_{\eta'}\right] + \left(\frac{\partial \eta'}{\partial \eta}\right)^2 L_{\eta'\eta'} \\
&= \left(\frac{\partial \eta'}{\partial \eta}\right)\left[\frac{-2\eta_0(1+\eta'\eta_0)}{1+\eta_0^2}\right]L_{\eta'} + \left(\frac{\partial \eta'}{\partial \eta}\right)^2 L_{\eta'\eta'} \\
&= \left(\frac{\partial \eta'}{\partial \eta}\right)\left[-2\eta_0\left(\frac{(1+\eta'\eta_0)(1+\eta'\eta_0)}{(1+\eta'\eta_0)(1+\eta_0^2)}\right)\right]L_{\eta'} + \left(\frac{\partial \eta'}{\partial \eta}\right)^2 L_{\eta'\eta'} \\
&= \left(\frac{\partial \eta'}{\partial \eta}\right)\left[\frac{2\eta_0}{1+\eta'\eta_0}.\frac{-(1+\eta'\eta_0)^2}{1+\eta_0^2}\right]L_{\eta'} + \left(\frac{\partial \eta'}{\partial \eta}\right)^2 L_{\eta'\eta'} \\
&= \left(\frac{\partial \eta'}{\partial \eta}\right).\left[\frac{2\eta_0}{1+\eta'\eta_0}.\left(\frac{\partial \eta'}{\partial \eta}\right)L_{\eta'}\right] + \left(\frac{\partial \eta'}{\partial \eta}\right)^2 L_{\eta'\eta'} \\
&= \left(\frac{\partial \eta'}{\partial \eta}\right)^2 L_{\eta'\eta'} + \frac{2\eta_0}{1+\eta'\eta_0}\left(\frac{\partial \eta'}{\partial \eta}\right)^2 L_{\eta'}. \qquad (4.11)
\end{aligned}$$

Similarly we can show that

$$L_{\bar\eta\bar\eta} = \left(\frac{\partial \bar\eta'}{\partial \bar\eta}\right)^2 L_{\bar\eta'\bar\eta'} + \frac{2\eta_0}{1+\bar\eta'\eta_0}\left(\frac{\partial \bar\eta'}{\partial \bar\eta}\right)^2 L_{\bar\eta'}, \qquad (4.12)$$

$$
\begin{aligned}
L_{\eta\bar{\eta}} &= \left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right) L_{\eta'\bar{\eta}'}\,, && (4.13)\\
L_{\bar{\eta}} &= \left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right) L_{\bar{\eta}'}. && (4.14)
\end{aligned}
$$

Substituting the above equations into (4.8) gives

$$
\begin{aligned}
& A\Big[\left(\frac{\partial\eta'}{\partial\eta}\right)^{2} L^{k+1}_{\eta'\eta'} + \frac{2\eta_0}{1+\eta'\eta_0}\left(\frac{\partial\eta'}{\partial\eta}\right)^{2} L^{k+1}_{\eta'}\Big]\\
+ \;& B\Big[\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^{2} L^{k+1}_{\bar{\eta}'\bar{\eta}'} + \frac{2\eta_0}{1+\bar{\eta}'\eta_0}\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^{2} L^{k+1}_{\bar{\eta}'}\Big]\\
+ \;& C\Big[\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right) L^{k+1}_{\eta'\bar{\eta}'}\Big]\\
+ \;& D\Big[\left(\frac{\partial L}{\partial\eta}\right) = \left(\frac{\partial L}{\partial\eta'}\right)\left(\frac{\partial\eta'}{\partial\eta}\right) = \left(\frac{\partial\eta'}{\partial\eta}\right) L^{k+1}_{\eta'}\Big]\\
+ \;& E\Big[\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right) L^{k+1}_{\bar{\eta}'}\Big]\\
+ \;& F.L^{k+1} = DEL
\end{aligned}
$$

or

$$
\begin{aligned}
& A\;\left(\frac{\partial\eta'}{\partial\eta}\right)^{2} L^{k+1}_{\eta'\eta'} + B\;\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^{2} L^{k+1}_{\bar{\eta}'\bar{\eta}'} + C\;\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right) L^{k+1}_{\eta'\bar{\eta}'} \qquad (4.15)\\
+ \;& \left(\frac{2\eta_0 A}{1+\eta'\eta_0}(\frac{\partial\eta'}{\partial\eta}) + D\right)\left(\frac{\partial\eta'}{\partial\eta}\right) L^{k+1}_{\eta'}\\
+ \;& \left(\frac{2\eta_0 B}{1+\bar{\eta}'\eta_0}(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}) + E\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right) L^{k+1}_{\bar{\eta}'} + F\;L^{k+1} = DEL,
\end{aligned}
$$

where by a similar procedure equation (4.9) gives

$$
\begin{aligned}
DEL = \;& A\;\left(\frac{\partial\eta'}{\partial\eta}\right)^{2} L^{k}_{\eta'\eta'} + B\;\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^{2} L^{k}_{\bar{\eta}'\bar{\eta}'} + C\;\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right) L^{k}_{\eta'\bar{\eta}'}\\
+ \;& \left(\frac{2\eta_0 A}{1+\eta'\eta_0}(\frac{\partial\eta'}{\partial\eta}) + D\right)\left(\frac{\partial\eta'}{\partial\eta}\right) L^{k}_{\eta'} \qquad (4.16)\\
+ \;& \left(\frac{2\eta_0 B}{1+\bar{\eta}'\eta_0}(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}) + E\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right) L^{k}_{\bar{\eta}'} + F\;L^{k} - \varphi^{k}.
\end{aligned}
$$

Note that we assumed that

$$
\eta' = x + iy, \qquad (4.17)
$$

hence with refer to Section 1.1 it follows that

$$\begin{aligned}
L_{\eta'} &= \frac{1}{2}(L_x - iL_y), && (4.18)\\
L_{\bar{\eta}'} &= \frac{1}{2}(L_x + iL_y), && (4.19)\\
L_{\eta'\eta'} &= \frac{1}{4}(L_{xx} - L_{yy} - 2iL_{xy}), && (4.20)\\
L_{\bar{\eta}'\bar{\eta}'} &= \frac{1}{4}(L_{xx} - L_{yy} + 2iL_{xy}) && (4.21)
\end{aligned}$$

and

$$L_{\eta'\bar{\eta}'} = \frac{1}{4}(L_{xx} + L_{yy}). \tag{4.22}$$

Substituting the above equations into (4.15) and (4.16) gives

$$\begin{aligned}
&A\left(\frac{\partial\eta'}{\partial\eta}\right)^2\left[\frac{1}{4}(L_{xx} - L_{yy} - 2iL_{xy})\right]^{k+1}\\
&+B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2\left[\frac{1}{4}(L_{xx} - L_{yy} + 2iL_{xy})\right]^{k+1}\\
&+C\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)\left[\frac{1}{4}(L_{xx} + L_{yy})\right]^{k+1}\\
&+\left[D + \frac{2\eta_0 A}{1+\eta'\eta_0}(\frac{\partial\eta'}{\partial\eta})\right](\frac{\partial\eta'}{\partial\eta})\left[\frac{1}{2}(L_x - L_y)\right]^{k+1}\\
&+\left[E + \frac{2\eta_0 B}{1+\bar{\eta}'\eta_0}(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\right](\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\left[\frac{1}{2}(L_x + L_y)\right]^{k+1} + F.L^{k+1}\\
&= \frac{1}{4}\left[A\left(\frac{\partial\eta'}{\partial\eta}\right)^2 + B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2 + C\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)\right]L_{xx}^{k+1}\\
&+\frac{1}{4}\left[-A\left(\frac{\partial\eta'}{\partial\eta}\right)^2 - B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2 + C\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)\right]L_{yy}^{k+1}\\
&+\frac{i}{2}\left[-A\left(\frac{\partial\eta'}{\partial\eta}\right)^2 + B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2\right]L_{xy}^{k+1}\\
&+\frac{1}{2}\left[\left(D + \frac{2\eta_0 A}{1+\eta'\eta_0}(\frac{\partial\eta'}{\partial\eta})\right)\left(\frac{\partial\eta'}{\partial\eta}\right) + \left(E + \frac{2\eta_0 B}{1+\bar{\eta}'\eta_0}(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\right)(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\right]L_x^{k+1}\\
&+\frac{i}{2}\left[-\left(D + \frac{2\eta_0 A}{1+\eta'\eta_0}(\frac{\partial\eta'}{\partial\eta})\right)\left(\frac{\partial\eta'}{\partial\eta}\right)\right.\\
&\left.+\left(E + \frac{2\eta_0 B}{1+\bar{\eta}'\eta_0}(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\right)(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\right]L_y^{k+1} + FL^{k+1} = DEL,
\end{aligned}\tag{4.23}$$

where

$$
\begin{aligned}
DEL = &\frac{1}{4}\Big[A\left(\frac{\partial\eta'}{\partial\eta}\right)^2 + B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2 + C\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)\Big]L^k_{xx}\\
&+\frac{1}{4}\Big[-A\left(\frac{\partial\eta'}{\partial\eta}\right)^2 - B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2 + C\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)\Big]L^k_{yy}\\
&+\frac{i}{2}\Big[-A\left(\frac{\partial\eta'}{\partial\eta}\right)^2 + B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2\Big]L^k_{xy}\\
&+\frac{1}{2}\Big[\left(D + \frac{2\eta_0 A}{1+\eta'\eta_0}(\frac{\partial\eta'}{\partial\eta})\right)\left(\frac{\partial\eta'}{\partial\eta}\right) + \left(E + \frac{2\eta_0 B}{1+\bar{\eta}'\eta_0}(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\right)(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\Big]L^k_x\\
&+\frac{i}{2}\Big[-\left(D + \frac{2\eta_0 A}{1+\eta'\eta_0}(\frac{\partial\eta'}{\partial\eta})\right)\left(\frac{\partial\eta'}{\partial\eta}\right)\\
&+\left(E + \frac{2\eta_0 B}{1+\bar{\eta}'\eta_0}(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\right)(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\Big]L^k_y + FL^k - \varphi^k.
\end{aligned}
\tag{4.24}
$$

Equations (4.23) and (4.24) can be written as

$$
\alpha_1 L^{k+1}_x + \alpha_2 L^{k+1}_{xx} + \beta_1 L^{k+1}_y + \beta_2 L^{k+1}_{yy} + \gamma_1 L^{k+1}_{xy} + \omega_1 L^{k+1} = DEL, \tag{4.25}
$$

where

$$
\begin{aligned}
DEL &= \alpha_1 L^k_x + \alpha_2 L^k_{xx} + \beta_1 L^k_y + \beta_2 L^k_{yy} + \gamma_1 L^k_{xy} + \omega_1 L^k - \varphi^k, \qquad (4.26)\\
\alpha_1 &= \frac{1}{2}\Big[\left(D + \frac{2\eta_0 A}{1-\eta'\eta_0}(\frac{\partial\eta'}{\partial\eta})\right)\left(\frac{\partial\eta'}{\partial\eta}\right)\\
&+ \left(E + \frac{2\eta_0 B}{1-\bar{\eta}'\eta_0}(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)\Big]^k,\\
\beta_1 &= \frac{i}{2}\Big[-\left(D - \frac{2\eta_0 A}{1-\eta'\eta_0}(\frac{\partial\eta'}{\partial\eta})\right)\left(\frac{\partial\eta'}{\partial\eta}\right)\\
&+ \left(E - \frac{2\eta_0 B}{1-\bar{\eta}'\eta_0}(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}})\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)\Big]^k,\\
\alpha_2 &= \frac{1}{4}\left[A\left(\frac{\partial\eta'}{\partial\eta}\right)^2 + B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2 + C\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)\right]^k,\\
\beta_2 &= \frac{1}{4}\left[-A\left(\frac{\partial\eta'}{\partial\eta}\right)^2 - B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2 + C\left(\frac{\partial\eta'}{\partial\eta}\right)\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)\right]^k,\\
\gamma_1 &= \frac{i}{2}\left[-A\left(\frac{\partial\eta'}{\partial\eta}\right)^2 + B\left(\frac{\partial\bar{\eta}'}{\partial\bar{\eta}}\right)^2\right]^k
\end{aligned}
$$

and

$$\omega_1 = [F]^k. \tag{4.27}$$

The equation (4.25) and (4.26) are the final linearised form of the Monge-Ampère equation.

4.3 Linearisation of boundary condition

The boundary condition is $\phi = 0$, where

$$\phi = |\omega - \omega_0|^2 - R_a^2, \tag{4.28}$$

ω is defined by equation (2.30), ω_0 and R_a are the centre and radius of the aperture circle respectively.

Thus, the linearised boundary form is

$$A_1 L_{\eta'}^{k+1} + B_1 L_{\bar{\eta}'}^{k+1} + C_1 L^{k+1} = G_1, \tag{4.29}$$

where

$$\begin{aligned}
G_1 &= A_1 L_{\eta'}^{k} + B_1 L_{\bar{\eta}'}^{k} + C_1 L^{k} - \phi^k, \qquad (4.30)\\
A_1 &= \left[\frac{\partial \phi}{\partial L_\eta}.\frac{\partial \eta'}{\partial \eta}\right]^k,\\
B_1 &= \left[\frac{\partial \phi}{\partial L_{\bar{\eta}}}.\frac{\partial \bar{\eta}'}{\partial \bar{\eta}}\right]^k,\\
C_1 &= \left[\frac{\partial \phi}{\partial L}\right]^k,\\
\frac{\partial \phi}{\partial L_\eta} &= \frac{\partial \phi}{\partial \omega}\frac{\partial \omega}{\partial L_\eta} + \frac{\partial \phi}{\partial \bar{\omega}}\frac{\partial \bar{\omega}}{\partial L_\eta}\\
&= \frac{\partial \phi}{\partial \omega}\left(\frac{\partial \omega}{\partial \bar{\xi}}\frac{\partial \bar{\xi}}{\partial L_\eta}\right) + \frac{\partial \phi}{\partial \bar{\omega}}\left(\frac{\partial \bar{\omega}}{\partial \xi}\frac{\partial \xi}{\partial L_\eta}\right),\\
\frac{\partial \phi}{\partial L} &= \frac{\partial \phi}{\partial \omega}\frac{\partial \omega}{\partial L} + \frac{\partial \phi}{\partial \bar{\omega}}\frac{\partial \bar{\omega}}{\partial L},
\end{aligned}$$

and similarly

$$\frac{\partial \phi}{\partial L_{\bar{\eta}}} = \frac{\partial \phi}{\partial \omega}\left(\frac{\partial \omega}{\partial \bar{\xi}}\frac{\partial \bar{\xi}}{\partial L_{\bar{\eta}}}\right) + \frac{\partial \phi}{\partial \bar{\omega}}\left(\frac{\partial \bar{\omega}}{\partial \xi}\frac{\partial \xi}{\partial L_{\bar{\eta}}}\right). \tag{4.31}$$

Substituting (4.18), (4.19) into (4.29) and (4.30) gives the final linearised boundary form (see Section 4.2)

$$\delta_1 L_x^{k+1} + \delta_2 L_y^{k+1} + \delta_3 L^{k+1} = G_1, \tag{4.32}$$

where

$$\begin{aligned} G_1 &= \delta_1 L_x^k + \delta_2 L_y^k + \delta_3 L^k - \phi^k, \\ \delta_1 &= \frac{1}{2}(A_1 + B_1), \\ \delta_2 &= \frac{i}{2}(-A_1 + B_1), \\ \delta_3 &= C. \end{aligned} \tag{4.33}$$

4.4 Numerical Solution

If the boundary-value problem has y=0 as a plane of symmetry we need to consider only one half of the aperture domain. This reduces the amount of computational effort when numerical methods of solution implemented.

To synthesise a system with a circular aperture under the symmetry condition, the Monge-Ampère equation (2.86) for the boundary condition (2.88) must be solved, (see Westcott (1981)).

Since the Monge-Ampère equation and the boundary problem are nonlinear, they are linearised as in Section 4.3 assuming a small perturbation from an initial solution. A Cartesian grid is used to subdivide the region as illustrated in Appendix B. Finite differences are used to calculate the required derivatives at each grid point (see Appendix C). Therefore the synthesis equations when expressed in terms of coordinates using the finite difference forms and corresponding differential operators result in a matrix representation of the boundary-value problem, which must be solved to evaluate L at the grid points.

Starting with an initial approximations for L (equation (3.5)) and the power density G (equation (3.12)), the matrix equation

$$O^k L^{k+1} = V^k$$

where k is the iteration number, O is a sparse matrix and V is a column matrix, is solved repeatedly until the successive iterates of L at each grid point converge (see Appendix D).

In the iterative sequence the value of L is kept fixed at its initial value at one particular grid point. This is a requirement of the iteration sequence to converge. This grid point is called the "*anchor point*".

Using NAG library routines the method has been programmed (Appendix E) and run on the Sun SPARCcentre 1000 computer.

4.4.1 Finite Difference Scheme

The finite difference scheme for the Monge-Ampère equation involves the first and second derivatives of L with respect to x and y at the internal grid points and first derivatives of L with respect to x and y at its boundary points (see for example Westcott (1986)). We consider these in the next two subsections.

a: Points on the Boundary

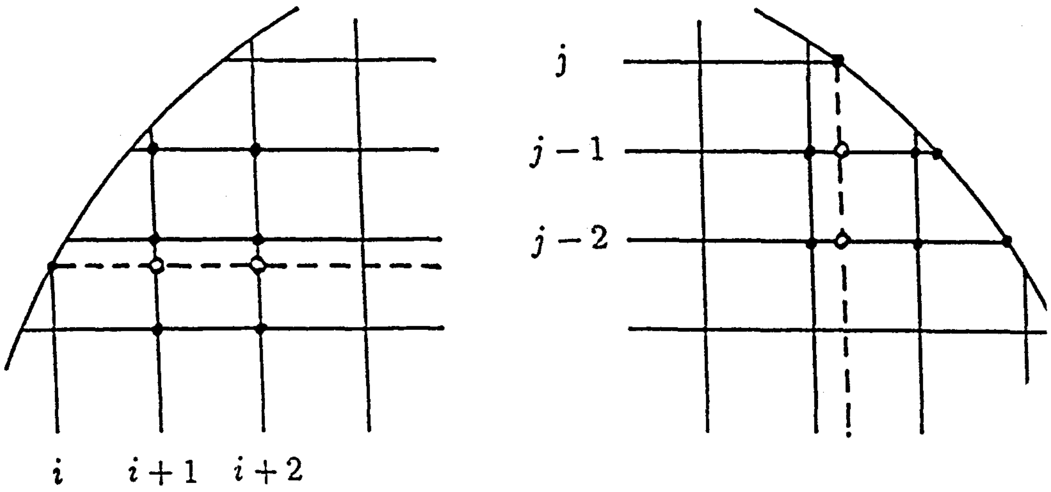
The boundary condition (2.88) involves only the first order derivatives L_x and L_y. At j boundary points, three points forward or backward difference forms and at i boundary points three points backward difference forms are used to approximate L_x and L_y respectively. The Lagrangian interpolation or extrapolation scheme is applied to evaluate L_y at the j boundary points, and L_x at the i boundary points (Fig.4.1). The solid black dots are grid points and the white circle dots are those obtained from the interpolation.

b: At the Internal Points

On the internal grid, three point central difference forms may be used to approximate the quantities L_x, L_y, L_{xx} and L_{yy}. To approximate the mixed derivative L_{xy} at (i, j) position, away from to the boundary we may use a nine point molecule (see Appendix C), central difference at $(i, j-1), (i, j)$ and $(i, j+1)$ to evaluate L_x and then central difference for L_y at each of these three points. In the case when point (i, j) is close to the boundary the backward difference form is used to evaluate L_y and then a central difference form is applied to calculate L_x (Fig.4.2). On the symmetry line $j = 0$, by the symmetry conditions, we have

$$L_y = L_{xy} = 0.$$

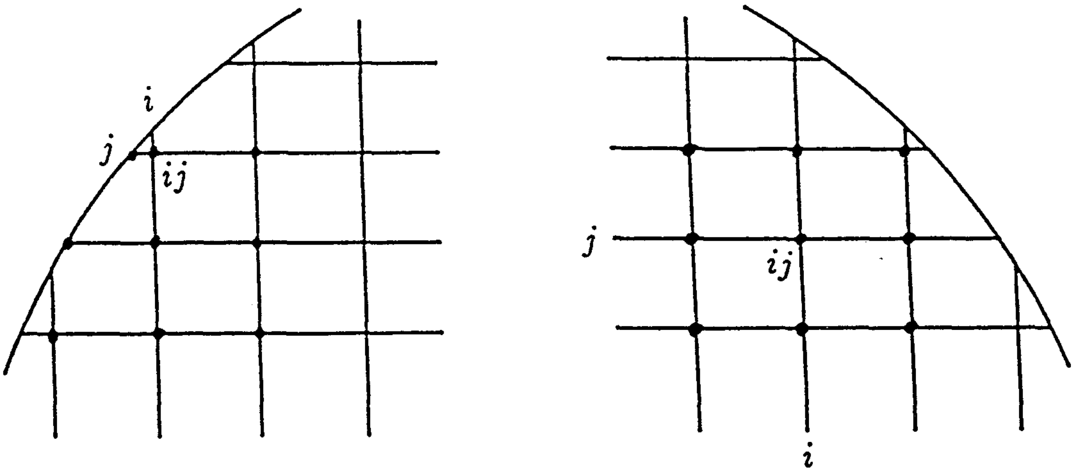
see Appendix B for more details.



a: i boundary point

b: j boundary point

Figure. 4.1: points required for evaluation of L_x and L_Y by interpolation.



c: point close to boundary

d: internal point

Figure. 4.2: points used for evaluation of L_{xy}.

4.5 Power density normalisation factor

For the circular aperture of radius R_a the power density function is chosen such that

$$G = f[(1-\mu)G_0 + \mu G_2] \tag{4.34}$$

where G_0 is initial aperture power density defined by equation (3.12), G_2 is Gaussian power density defined in Section 3.3.2, f is a normalisation factor which is approximately equal to 1 and μ is a parameter taking increasing values between 0 and 1 so that G_2 is the final power density.

The normalisation factor f is chosen to satisfy the energy conservation requirement. The total energy incident in the reflector must equal the energy passing through the output aperture. To calculate the total energy intercepted by the reflector we must allow for the fact that energy is lost by refraction at the dielectric interface.

To determine the normalisation factor f we integrated equation (4.34) over the aperture and equated to the total power density over aperture. Hence

$$\begin{aligned}
\int\int_{aperture} GdXdY &= f\int\int_{aperture} [(1-\mu)G_0 + \mu G_2]\, dXdY \\
&= f\Big[(1-\mu)\int\int_{aperture} G_0 dXdY \\
&\quad +\mu\int\int_{aperture} G_2 dXdY\Big] \\
&= f\left[(1-\mu)P + \mu\frac{\pi b}{a}(1 - e^{-aRa^{2}})\right] \\
&= f\left[(1-\mu)P + \mu\frac{\pi b}{a}(1 - 10^{\frac{-D}{10}})\right] \qquad (4.35)
\end{aligned}$$

where a, b, D are defined in Section (3.3.2) and P is calculated from equation (3.34) using the initial design. Equation (2.54) implies that

$$f\left[(1-\mu)P + \mu\frac{\pi b}{a}(1 - 10^{\frac{-D}{10}})\right] = \int\int_{cone} \frac{4F}{(1+|\eta'|^2)^2}dxdy, \tag{4.36}$$

where F is defined by equation (2.53). Thus

$$f = \frac{\int\int_{cone} \frac{4F}{(1+|\eta'|^2)^2} dx dy}{(1-\mu)P + \mu \frac{\pi b}{a}(1 - 10^{\frac{-D}{10}})}. \tag{4.37}$$

With a suitable adjusted grid, convergence must normally occur within a few iterations, otherwise the numerical procedure may become unstable. The integration of power to produce energy compatibility is given by equation (4.37). The integration on the right hand side of the equation is done numerically and may result in a small error. A larger error occurs at the anchor point where the Monge-Ampère equation constraint is not imposed. This error manifests itself as a distortion of the mapping around the anchor point. The distortion may be considerably reduced by either increasing the number of grid points or by using a nonuniform distribution of grid points. In the present series of calculations using about 1200 points, it was found advantageous to use a compensation factor δf to the normalisation factor f to reduce the distortion to a desired result.

Example 4.1: Let

γ=80	θ_c=20	k=0.14	D=10
ψ=33	N=2.6	e=0.22	

Then the ray diagram on the plane of symmetry and graph when $\mu = 0$ and $f = 1$ are shown in Fig.(4.3) and Fig.(4.4).

As μ is changed iteratively from 0 to 1 and f is recalculated at the end of each iteration. When $\mu = 1$ the value of $f = 0.9904$. The ray diagram on the plane of symmetry and aperture graph using this value of f for $\mu = 1$ are shown in Fig.(4.5) and Fig.(4.6) respectively. Clearly in these figures, the mapping is distorted around the anchor point. This is corrected in next example.

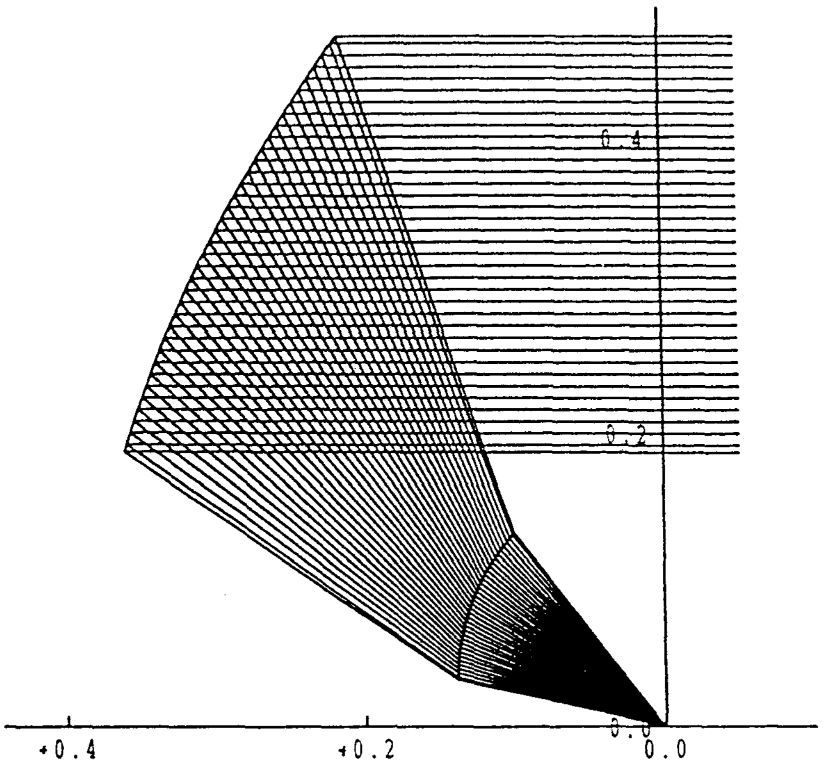


Figure 4.3: Ray diagram on the plane of the symmetry for example 4.1 when $\mu = 0$ and f=1.

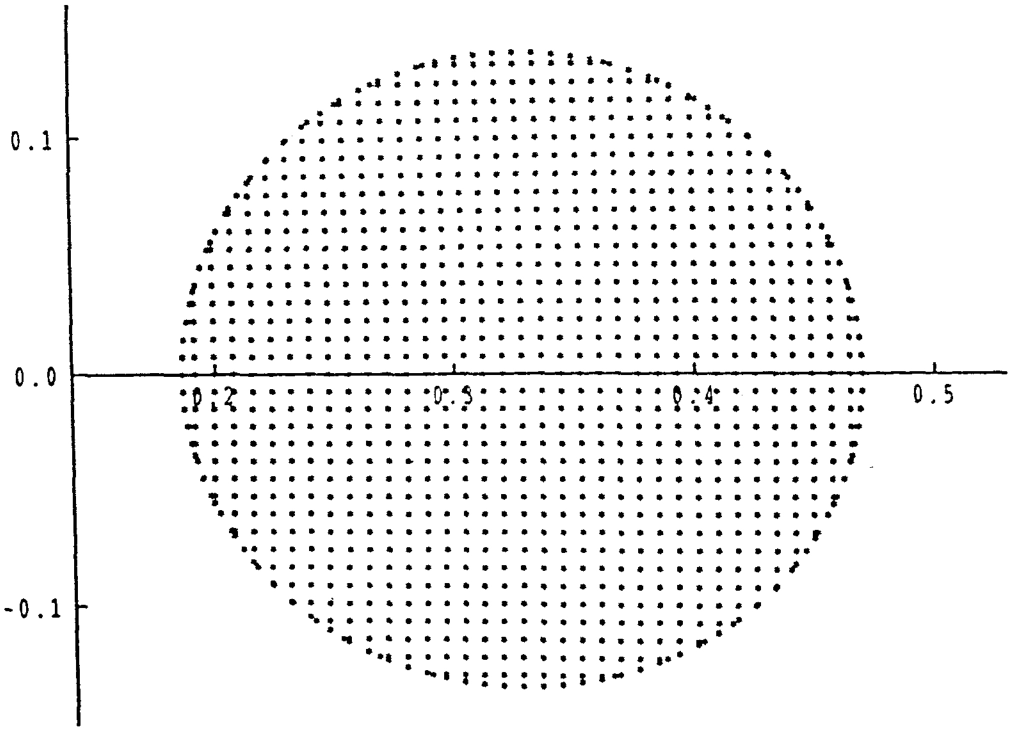


Figure 4.4: Mapping of uniform Cartesian grid onto circular region for example 4.1 when $\mu = 0$ and f=1.

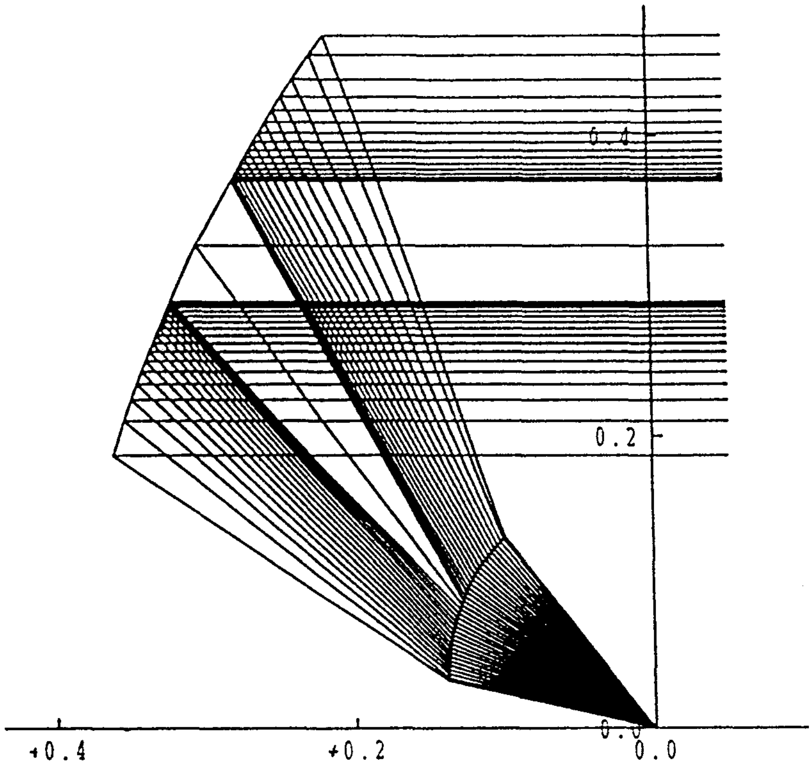


Figure 4.5: Ray diagram on the plane of the symmetry for example 4.1 when $\mu = 1$ and f=0.9904.

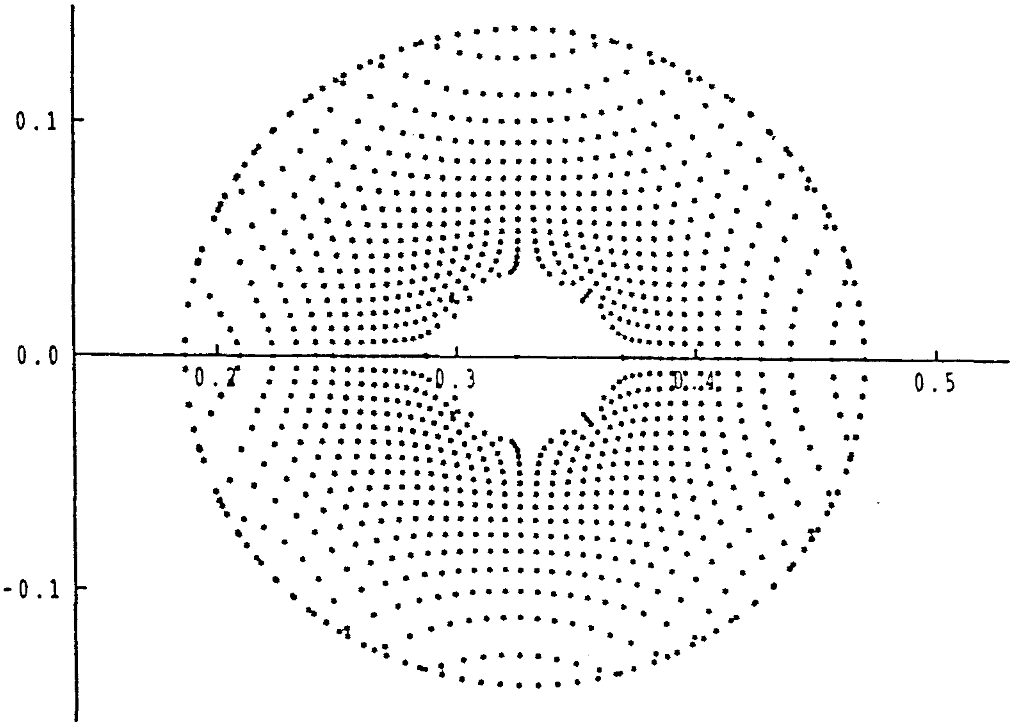


Figure 4.6: Mapping of uniform cartesian grid onto circular region for example 4.1 when $\mu = 1$ and f=0.9904.

Example 4.2: In example (4.1) if we replaced the value f by

$$f = f + \delta f \tag{4.38}$$

where $\delta f = -0.1$ and so $f = 0.8904$, then the ray diagram on the plane of symmetry and aperture graph when $\mu = 1$ is shown in Fig.(4.7) and Fig.(4.8) respectively. Clearly a major improvement in the mapping is obtained.

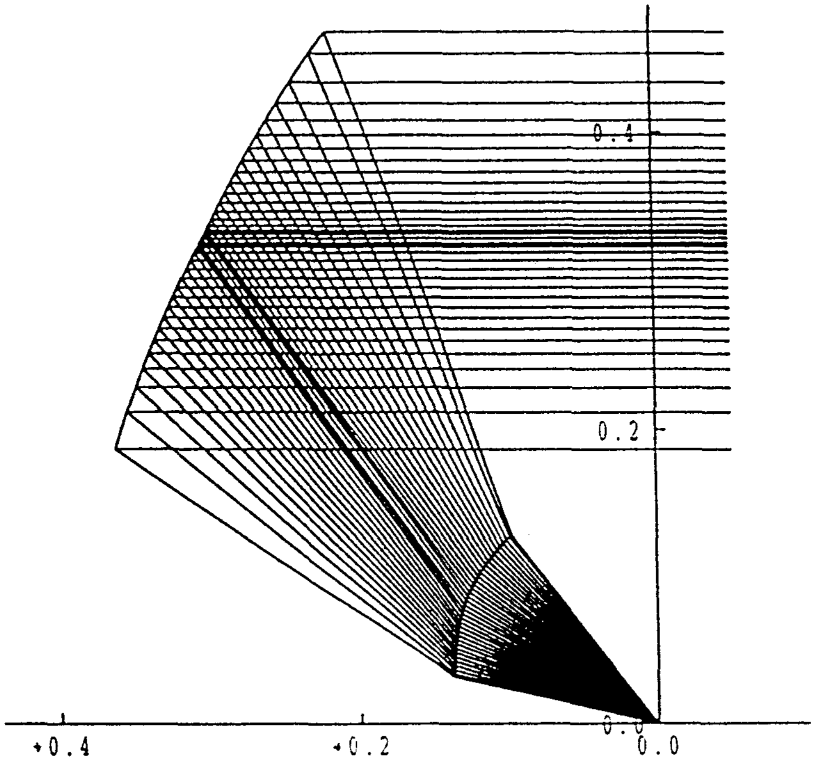


Figure 4.7: Ray diagram on the plane of the symmetry for example 4.2 when $\mu = 1$ and f=0.8904.

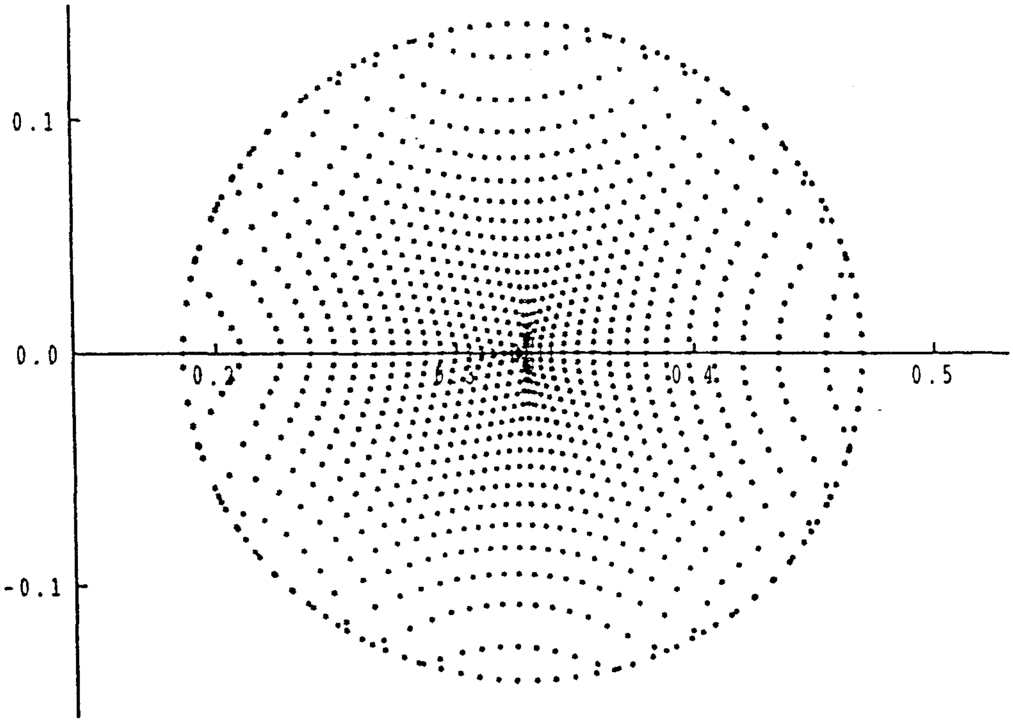


Figure 4.8: Mapping of uniform cartesian grid onto circular region for example 4.2 when $\mu = 1$ and f=0.8904.

Chapter 5

Implementation of the synthesis procedure and computed results

5.1 Introduction

In previous chapters it was shown that the problem of a synthesising a dielectric single reflector system under the assumptions of GO may be formulated in terms of a non-linear boundary-value problem. The technique of solution involves linearising the synthesis equations and approximating the derivatives by the application of a finite difference model. A uniform Cartesian grid over the circular source region was used.

Since the power normalisation factor may be incompatible with the requirement to maintain numerical stability some compensation of the power normalisation factor is necessary to reduce the power deviation at the anchor point.

The input geometrical parameter values required by the system are defined in Appendix E. These parameters are chosen so that there is no blockage to the aperture. Many of the steps involved in the procedure are dealt with automatically in the computer programme which is described in Appendix E and has been used successfully in the synthesis of a number of reflector systems. Computer results are presented relating to the mapping L, the location and magnitude of the maximum cross-polar component, and the power density transfer coefficient for the dielectric surface.

5.2 Synthesis procedure

A suitable procedure for the synthesis of a system with circular aperture is described in this section. It should noted at the outset that the iterative process involved is very sensitive to the accuracy of any initial approximations used.

A procedure which has been applied for the synthesis of a number of single reflector systems is outlined as below

a) choose Cartesian grid ($\delta x = \delta y$)

b) generate initial solution

c) synthesise circular aperture power distribution from uniform source distribution

d) apply power density normalisation factor f

e) generate dielectric and main reflector surface from the converged equations (2.9) and (2.33).

5.3 Compensation of the power normalisation factor

With the grid suitably adjusted and energy conserved between source and aperture during the synthesis, convergence normally occurs within a few iterations, otherwise if energy is not accurately conserved between the source and the aperture during the synthesis, the iterative process may become unstable and fail to converge. In fact, many of the distortions which occur in the computer examples can indeed be explained in terms of a failure to conserve energy.

From computed results, it becomes evident that a pronounced distortion of the source to aperture ray mapping can occur in the region of the anchor point, where L is held constant. The level of distortion is seen to be strongly dependent on the amount of edge-taper applied to aperture power distributions. Such an effect is undesirable since it will lead to large deviations from the aperture distribution, G, that the reflector is required to produce from a given source pattern, I. A method of reducing the distortion is therefore sought.

A possible explanation for the effect arises from the observation that it occurs in the neighbourhood of a point where the Monge-Ampère equation is not applied. The power normalisation is calculated under the assumption that the differential power equalisation as embodied in the Monge Ampère equation is imposed over the whole of the aperture domain. If energy is not conserved between source and aperture during the synthesis, the iterative procces may become unstable and fail to converge.

Since the power normalisation factor f may be incompatible with the requirement to maintain numerical stability, it is necessary to adjust the value of f until the power deviation at the anchor point is reduced to an acceptable level and produces satisfactory solutions to the boundary-value problem (see Section 4.5). The value of the compensation factor δf required to minimise distortion of the ray mapping is determined experimentally for each reflector system. For all cases it is found that $f \approx 1$, indicating that δf is small but a sensitive parameter.

5.4 Computer results

A number of cases of a single reflector with dielectric cone feed have been synthesised by using a computer programme to obtain numerical solutions to the boundary-value problem of chapter 2. This involves the use of a Cartesian grid on the source domain. At each grid point a converged value of the mapping function L is sought and when this is achieved the required dielectric and reflector surfaces may be obtained at those points corresponding to the grid points.

A parameter study of two basic cases is given in examples 5.1 and 5.4. Different values of output power taper are applied via the distributions described in section 3.3.2 and the effect on the $\xi \rightarrow \omega$ ray mapping is displayed for each example. Distortion of the mapping occurs predominantly in the region of the anchor point, which is chosen to be at the centre of the source domain in all cases.

5.5 Application of aperture power taper

We would like to be able to shape a system to produce a tapered distribution on the aperture. As we discussed in Section 3.3.2 we suppose this is a Gaussian power distribution over the aperture which is defined by equation (3.30). This equation depends on the power density tapered D in dBs at edge of aperture which is defined by equation (3.31). The power distribution G_2 is uniform when D=0 (see Fig.3.4).

In the computations presented in examples 5.1, 5.2 and 5.3, the edge tapers of 2, 5 and 10 dBs are specified. Some computed results of these examples is followed by the graphs of ray tracing on the plane of symmetry and mapping of the uniform Cartesian grid onto the circular aperture region.

Example 5.1: Let

$\gamma = 50$ $\theta_c = 28$ ψ=50 $\delta x = \delta y = 0.011$

k=0.15 N=2.6 e=0.1.

Some computed results of this example are shown in Table 5.1. The ray diagram on the plane of symmetry and aperture graph when $D = 2$ dB is shown in Fig.5.1 and Fig.5.2.

μ	number of convergence iteration	μ	number of convergence iteration	other output
0.00	4	0.55	2	$n_x = 46$
0.05	2	0.60	2	$n_y = 23$
0.10	2	0.65	2	number of elements
0.15	2	0.70	2	are 1780
0.20	2	0.75	2	user cpu time is
0.25	2	0.80	2	146.197 second
0.30	2	0.85	2	
0.35	2	0.90	2	
0.40	2	0.95	2	
0.45	2	1.00	2	
0.50	2			

Table 5.1: Convergence detail example 5.1 when D=2 dB.

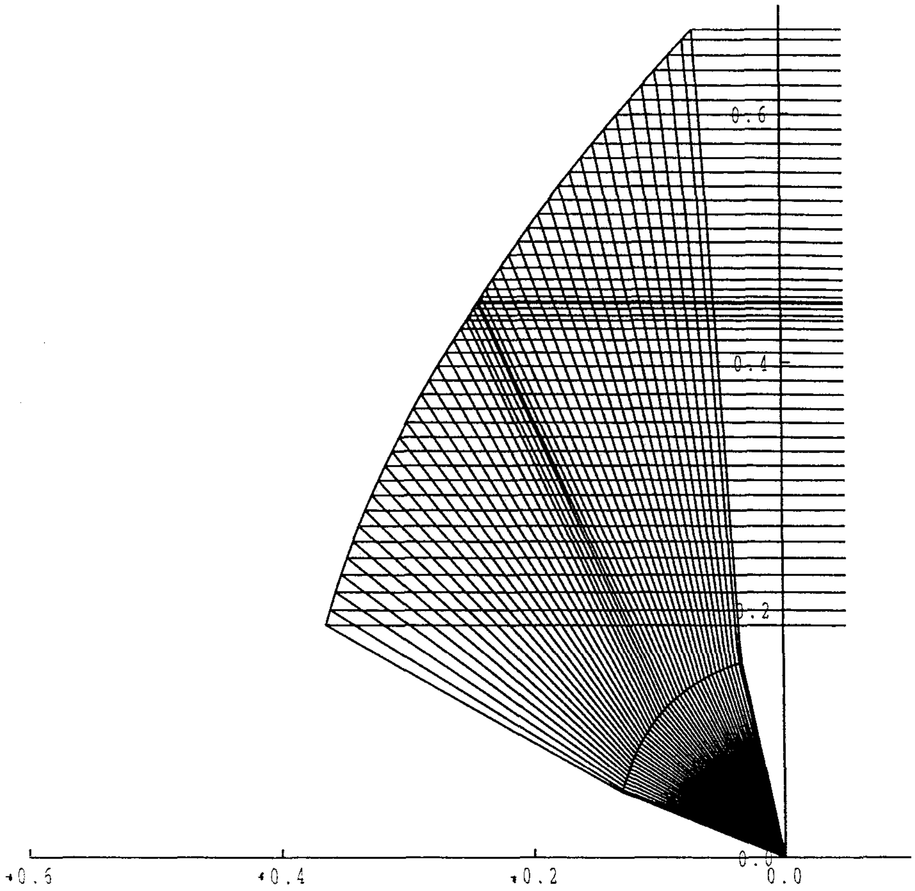


Figure 5.1: Ray diagram on the plane of the symmetry for example 5.1.

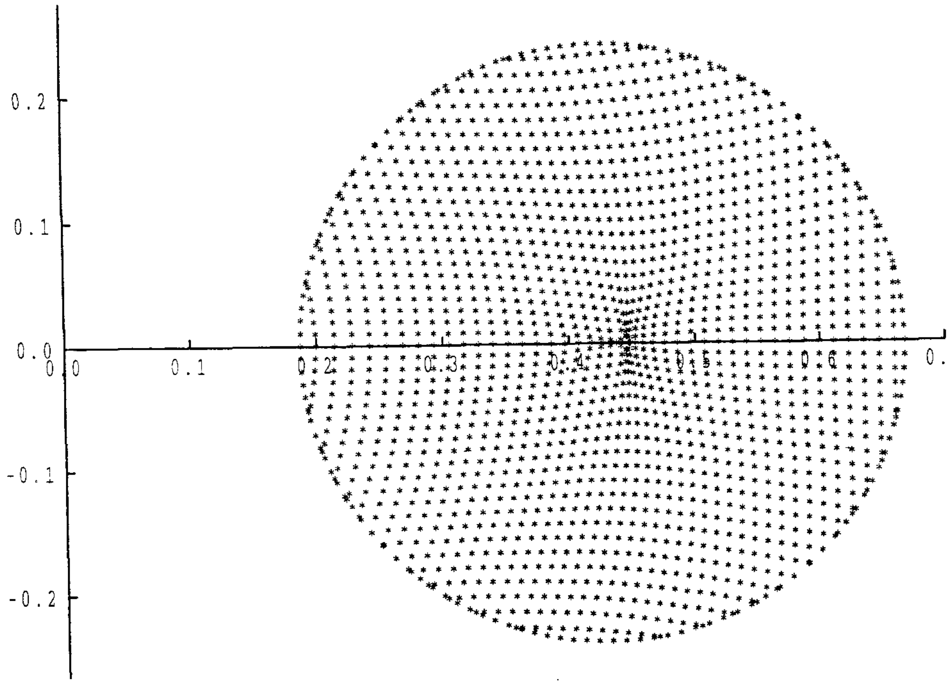


Figure 5.2: Mapping of uniform Cartesian grid onto circular region for example 5.1.

Example 5.2:

In example 5.1 if we replaced the value of D by $D = 5$, then some computed results of this example are shown in Table 5.2. The ray diagram on the plane of symmetry and aperture graph is shown in Fig.5.3 and Fig.5.4 respectively.

μ	number of convergence iteration	μ	number of convergence iteration	other output
0.00	4	0.55	3	$n_x = 46$ $n_y = 23$ number of elements are 1780 user cpu time is 235.700 second
0.05	2	0.60	4	
0.10	2	0.65	4	
0.15	3	0.70	4	
0.20	3	0.75	4	
0.25	3	0.80	4	
0.30	3	0.85	4	
0.35	3	0.90	4	
0.40	3	0.95	5	
0.45	3	1.00	5	
0.50	3			

Table 5.2: Convergence detail for example 5.2 when D=5 dB.

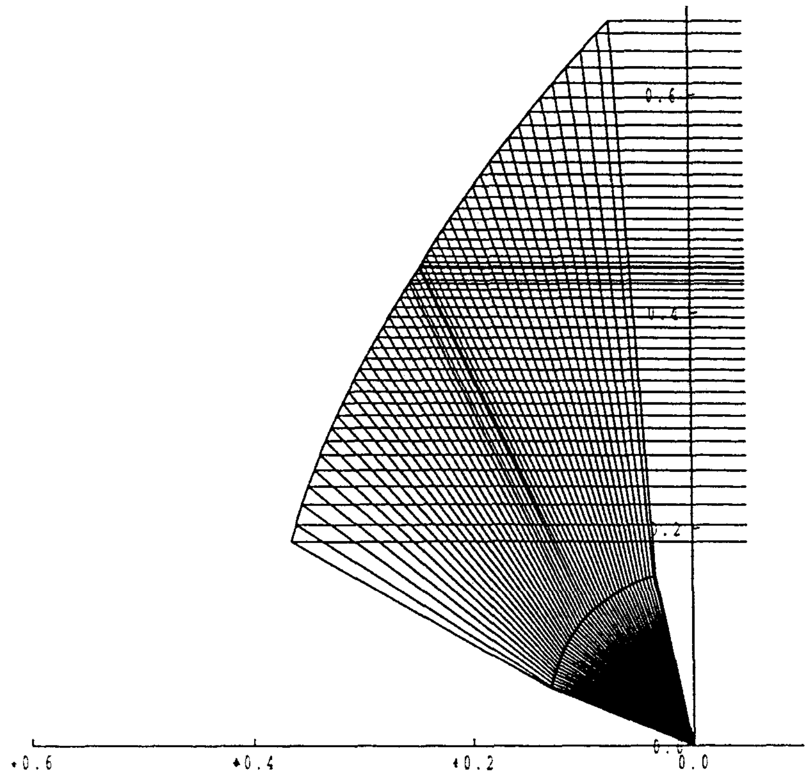


Figure 5.3: Ray diagram on the plane of the symmetry for example 5.2.

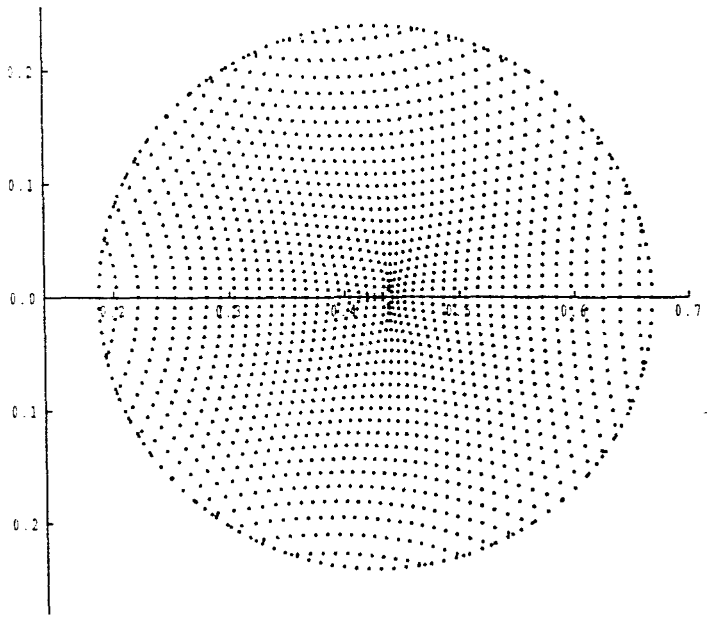


Figure 5.4: Mapping of uniform Cartesian grid onto circular region for example 5.2.

Example 5.3:

In example 5.1 if we replaced the value of D by $D = 10$ then some computed results of this example are shown in Table 5.3. The ray diagram on the plane of symmetry and aperture graph is shown in Fig.5.5 and Fig.5.6 respectively.

μ	number of convergence iteration	μ	number of convergence iteration	other output
0.00	4	0.55	6	$n_x = 46$ $n_y = 23$ number of elements are 1780 user cpu time is 410.170 second
0.05	3	0.60	6	
0.10	3	0.65	6	
0.15	3	0.70	7	
0.20	4	0.75	7	
0.25	4	0.80	8	
0.30	4	0.85	8	
0.35	4	0.90	9	
0.40	5	0.95	10	
0.45	5	1.00	10	
0.50	5			

Table 5.3: Convergence detail for example 5.3 when D=10 dB.

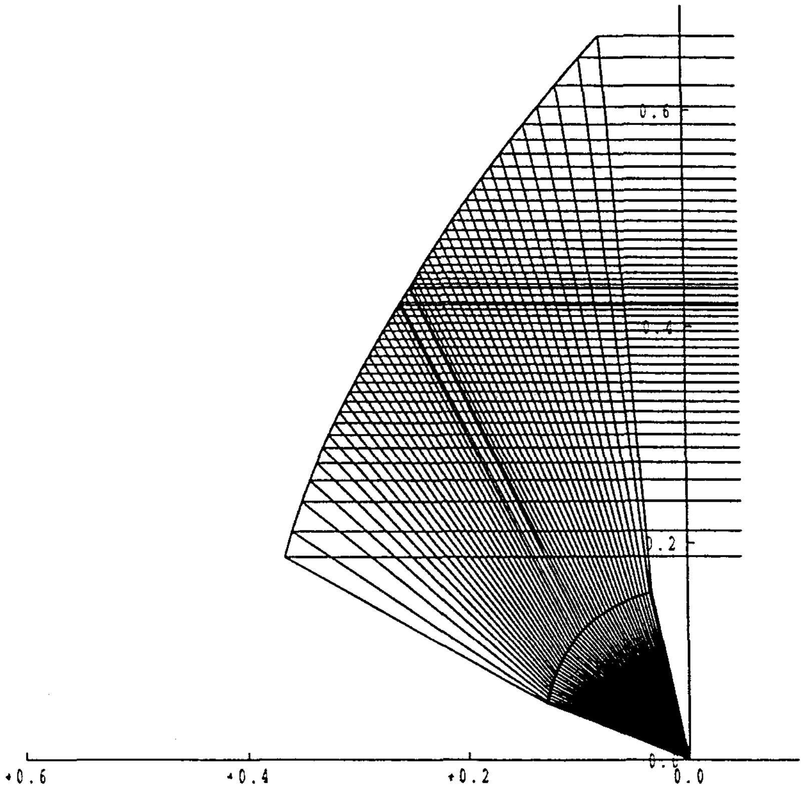


Figure 5.5: Ray diagram on the plane of the symmetry for example 5.3.

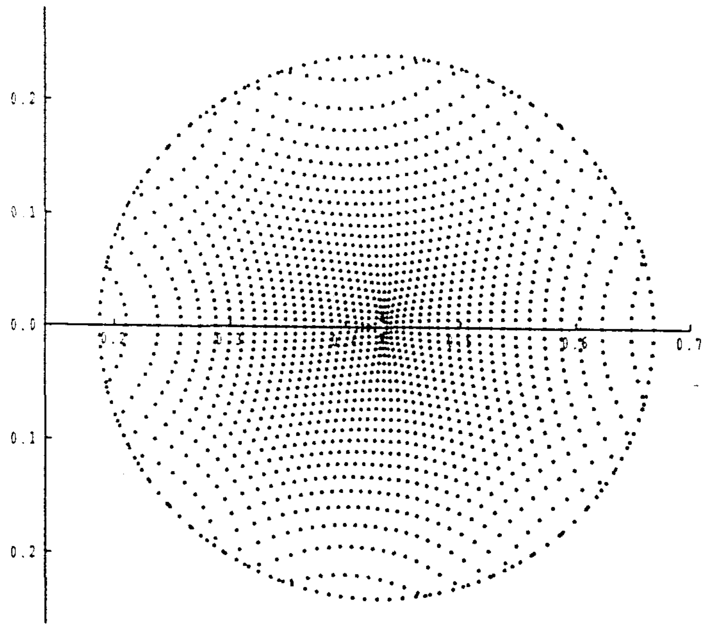


Figure 5.6: Mapping of uniform Cartesian grid onto circular region for example 5.3.

5.6 Cross-polarisation

The level of cross-polarisation power ratio on the aperture is defined on GO assumptions by the equation

$$10log_{10}\left(\frac{Re(\chi)}{Im(\chi)}\right)^2$$

, where χ is defined by equation (2.71) (see Westcott 1993). In general the cross-polar level will of course depends on the shape of the dielectric surface through L(η). The calculation of maximum cross-polar is complicated due to equation (2.14).

Table 5.4 shows the power density compensation δf and maximum cross-polar in dB for examples 5.1, 5.2 and 5.3 when D=2, 5 and 10 dBs.

In the Figures 5.7, 5.8 and 5.9 the position of the maximum cross-polar has been found and shown by solid dots.

D	δf	*max. cross-polar*
2	-0.114	-15.059241
5	-0.083	-15.008158
10	-0.060	-14.95059

Table 5.4: Table shows power density compensation δf and maximum cross-polar for examples 5.1, 5.2 and 5.3.

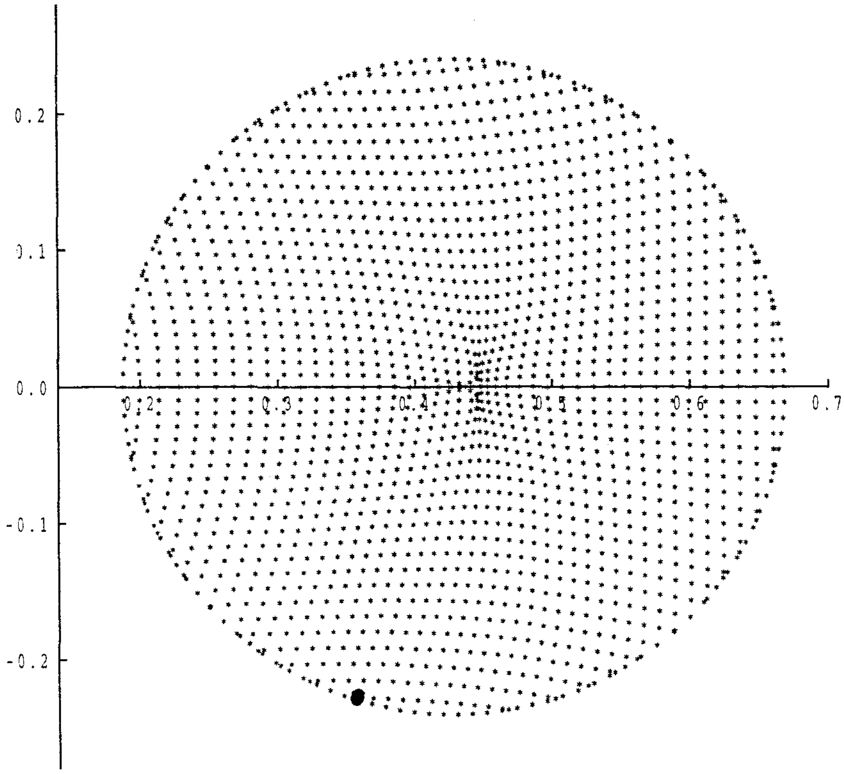


Figure 5.7: Solid dot shows the position (ir(i,j)=861) of maximum cross-polar for examples 5.1.

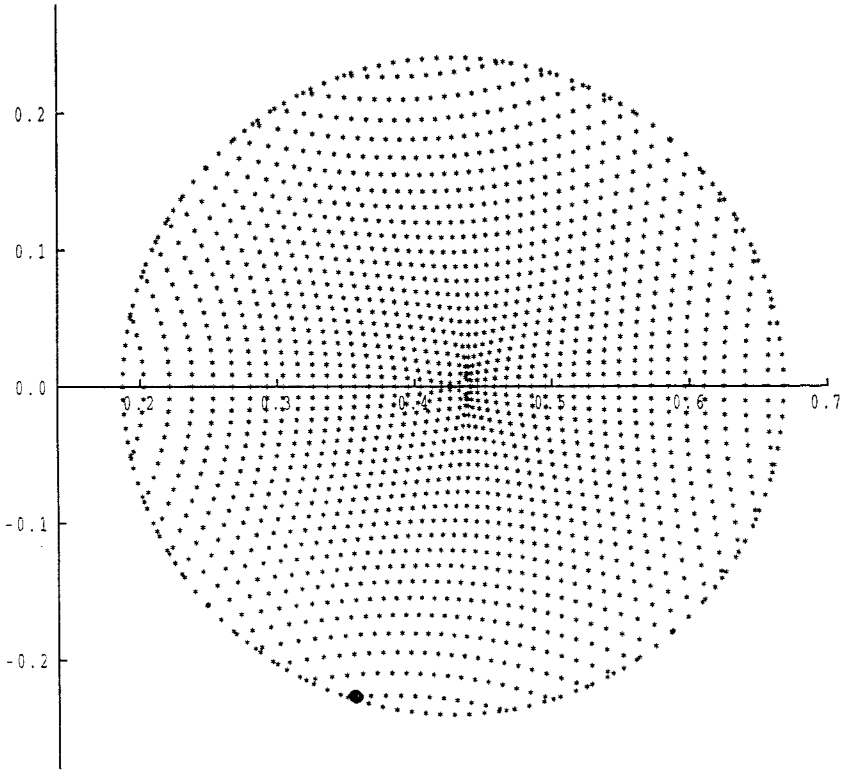


Figure 5.8: Solid dot shows the position (ir(i,j)=861) of maximum cross-polar for examples 5.2.

i+2

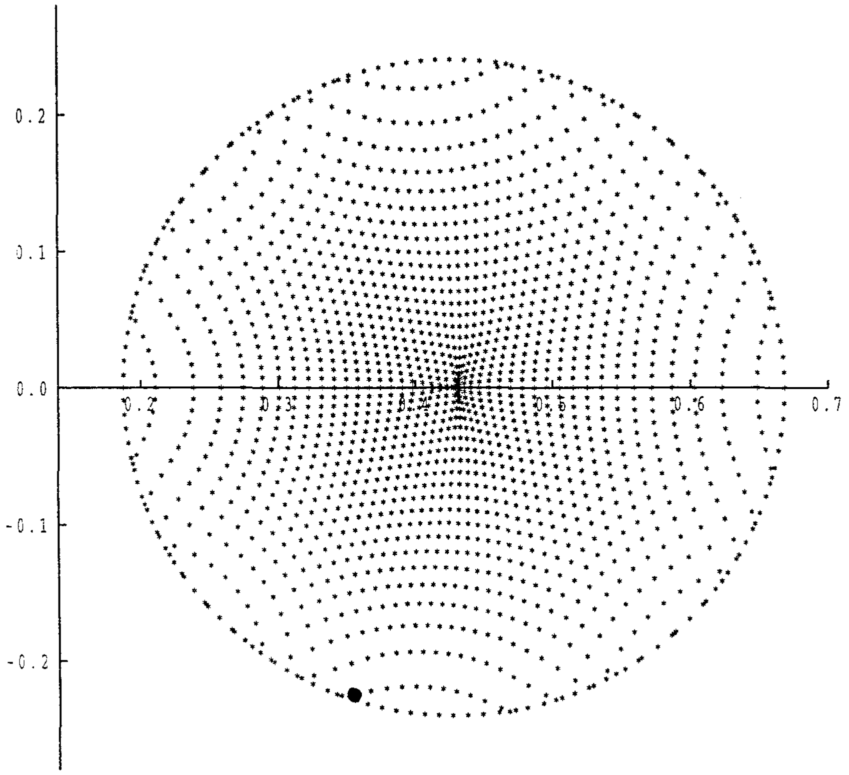


Figure 5.9: Solid dot shows the position (ir(i,j)=861) of maximum cross-polar for examples 5.3.

Computational results show that the level of maximum cross-polarisation increases when the edge taper increases (see Table 5.4). The position of the peak cross-polarisation remains unchanged when the edge taper changes (see Figures 5.7, 5.8 and 5.9).

5.7 Power density transfer ratio t_{12}

If the incident power density is I(η), then after crossing the interface the transmitted power density is $t_{12}I(\eta)$, where $t_{12}(\eta)$, the fraction of power density transmitted across the dielectric surface, is defined by equation (2.44) (see also Fig.2.3).

For the spherical dielectric when N=1, $\theta_1 = \theta_2$, and hence $cos(\theta_1 - \theta_2) = cos(0) = 1$, equation (2.42) gives $T_{\perp} = T_{||}$. Since $A^2 + B^2 = 1$, it follows that $I_2 = I_1$, therefore equation (2.44) gives $t_{12} = 1$. For other cases (eg. N$\neq$1) $t_{12} < 1$. In the following examples the minimum power density transfer ratio t_{12} has been found and its position shown by solid dot.

Table 5.5 shows power density compensation δf and minimum power density transfer ratio t_{12} for examples 5.1, 5.2 and 5.3 when D=2, 5 and 10.

Figures 5.10, 5.11 and 5.12 give the position of the minimum power density transfer ratio t_{12} for examples 5.1, 5.2 and 5.3.

D	δf	min. t_{12}
2	-0.114	0.758412
5	-0.083	0.757951
10	-0.060	0.745191

Table 5.5: Table shows power density compensation δf and minimum power density transfer ratio t_{12} for examples 5.1, 5.2 and 5.3.

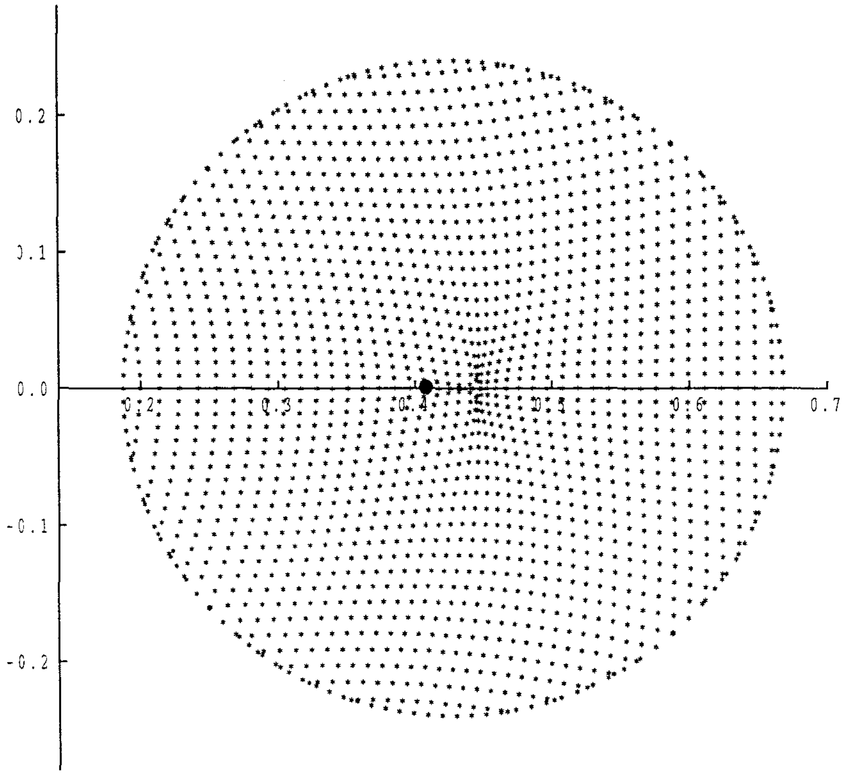


Figure 5.10: Solid dot shows the position (ir(i,j)=18) of minimum power density transfer ratio t_{12} across the dielectric surface for example 5.1 when D=2 dB.

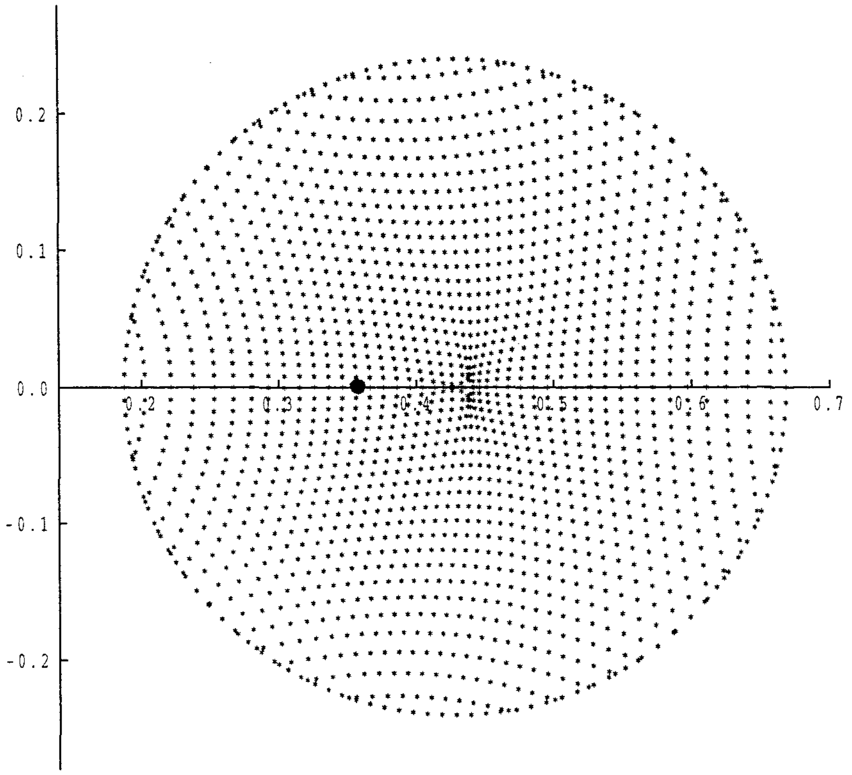


Figure 5.11: Solid dot shows the position (ir(i,j)=13) of minimum power density transfer ratio t_{12} across the dielectric surface for example 5.2 when D=5 dB.

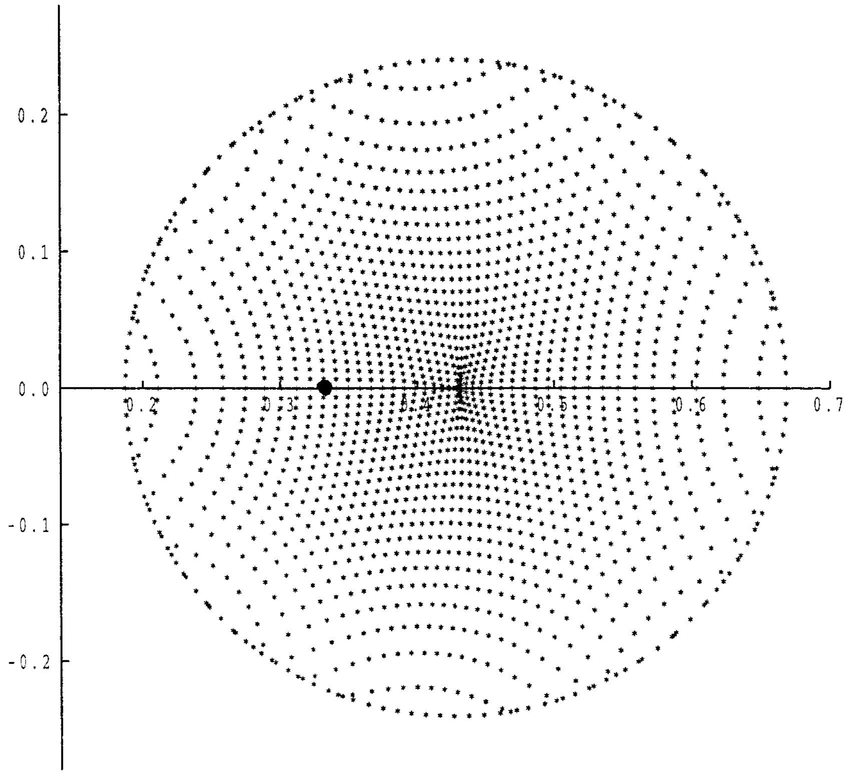


Figure 5.12: Solid dot shows the position (ir(i,j)=9) of minimum power density transfer ratio t_{12} across the dielectric surface for example 5.3 when D=10 dB.

Table 5.5 shows that when edge taper D increases from 2 to 10 dBs, then the value of the minimum power density transfer ratio t_{12} decreases. The position of the t_{12} migrates in the negative x direction when D increases (see Figures 5.10, 5.11 and 5.12).

Example 5.4: Let

$\gamma = 80$	$\theta_c = 20$	ψ=33	$\delta x = \delta y = 0.01$
k=0.14	N=2.6	e=0.22.	

We summarise the values of maximum cross-polar, minimum power density transfer ratio and δf for this example in Table 5.6 when D=2, D=5, D=10 and D=12 dBs. The table shows that when D increases from 2 to 12 dBs, then the level of maximum cross-polar increases and the minimum power density ratio t_{12} will decrease.

Some computed results for this example when D increases from 2 to 12 dBs are shown in Table 5.7, 5.8, 5.9 and 5.10 respectively. The ray diagram on the plane of symmetry, aperture graph, position of the maximum cross-polar and minimum power density transfer ratio t_{12} across the dielectric surface when $D = 2, D = 5, D = 10$ and $D = 12$ dBs are shown in Figures 5.13-5.28.

D	δf	*max. cross-polar*	$min.t_{12}$
2	-0.160	-29.550845	0.712060
5	-0.122	-29.245812	0.707213
10	-0.100	-28.853015	0.769634
12	-0.090	-28.729871	0.684210

Table 5.6: Table shows power density compensation δf, maximum cross-polar and minimum density transfer ratio t_{12} for example 5.4

μ	number of convergence iteration	μ	number of convergence iteration	other output
0.00	7	0.55	2	$n_x = 36$ $n_y = 18$ number of elements are 1112 user cpu time is 109.580 second
0.05	2	0.60	2	
0.10	2	0.65	2	
0.15	2	0.70	2	
0.20	2	0.75	2	
0.25	2	0.80	2	
0.30	2	0.85	2	
0.35	2	0.90	1	
0.40	2	0.95	1	
0.45	2	1.00	1	
0.50	2			

Table 5.7: Convergence detail for example 5.4 when D=2 dB.

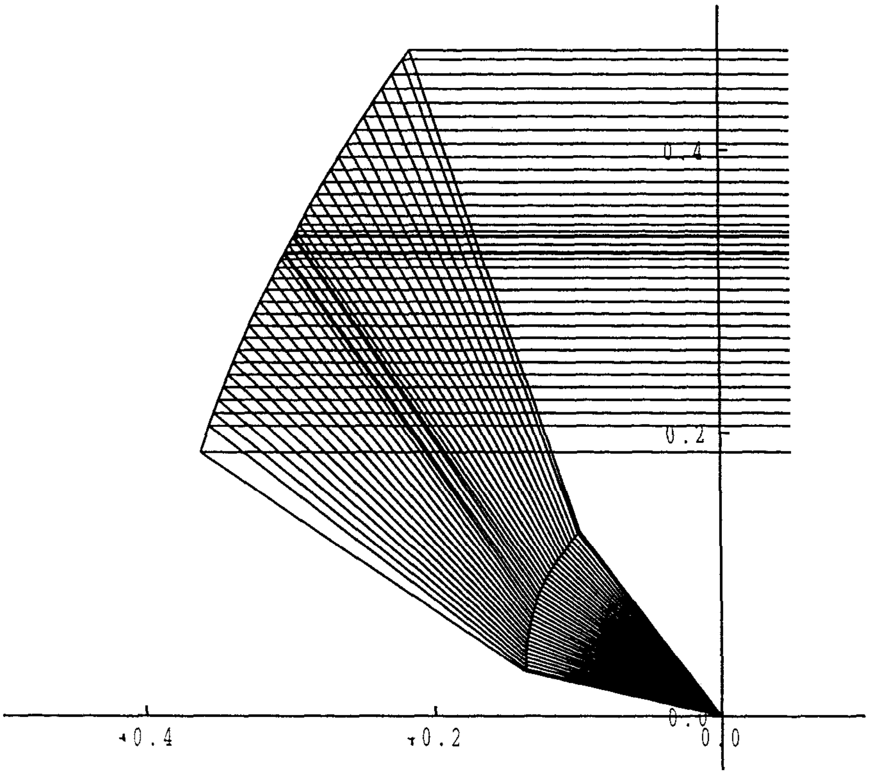


Figure 5.13: Ray diagram on the plane of the symmetry for example 5.4. when D=2.

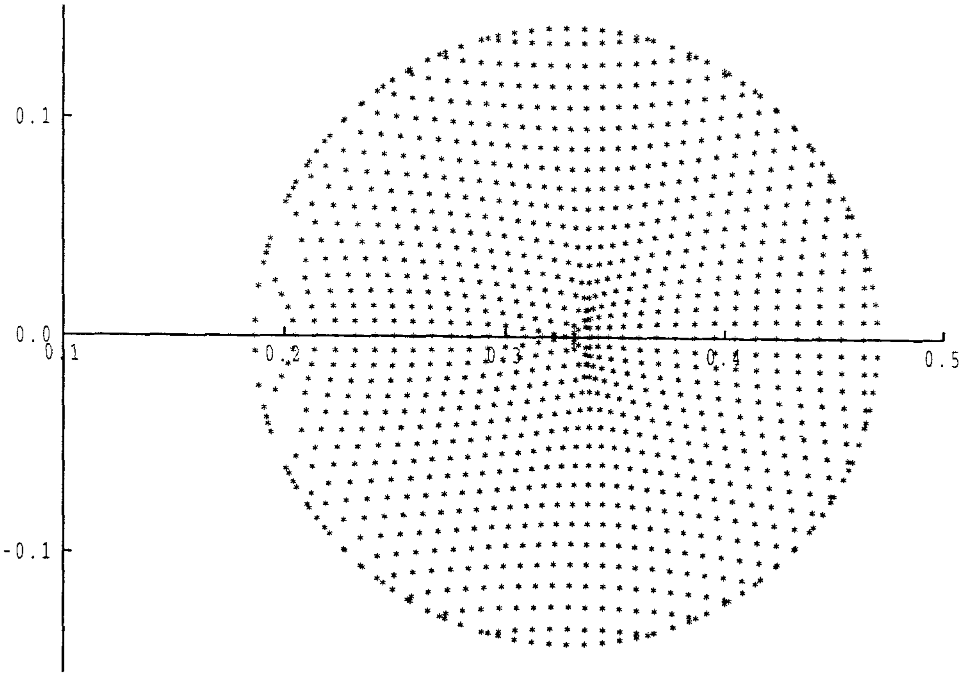


Figure 5.14: Mapping of uniform Cartesian grid onto circular region for example 5.4 when D=2.

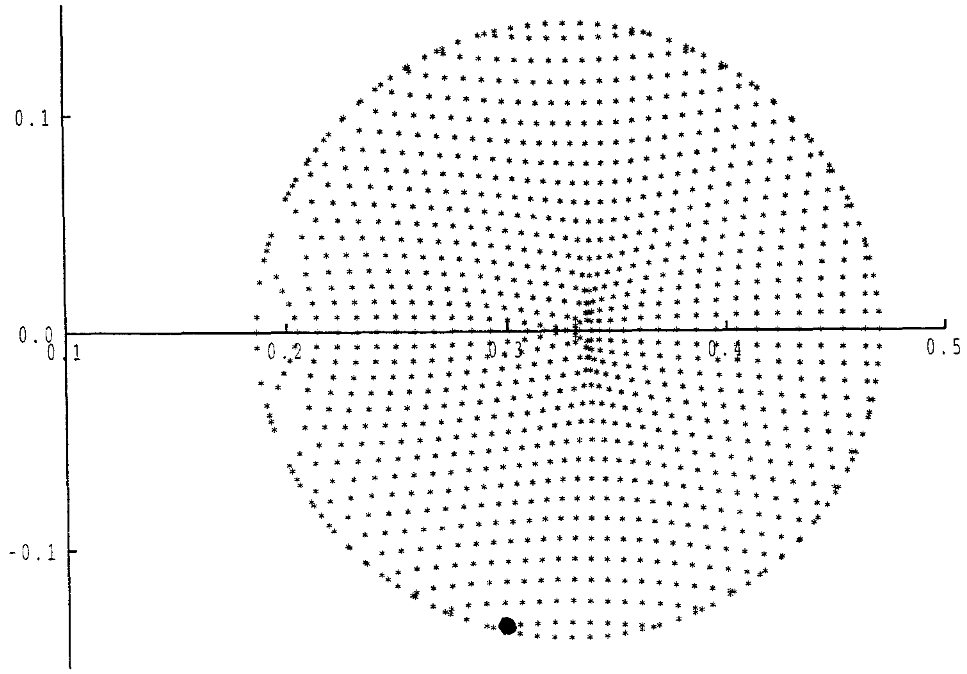


Figure 5.15: Solid dot shows the position (ir(i,j)=530) of maximum cross-polar for examples 5.4 when D=2 dB.

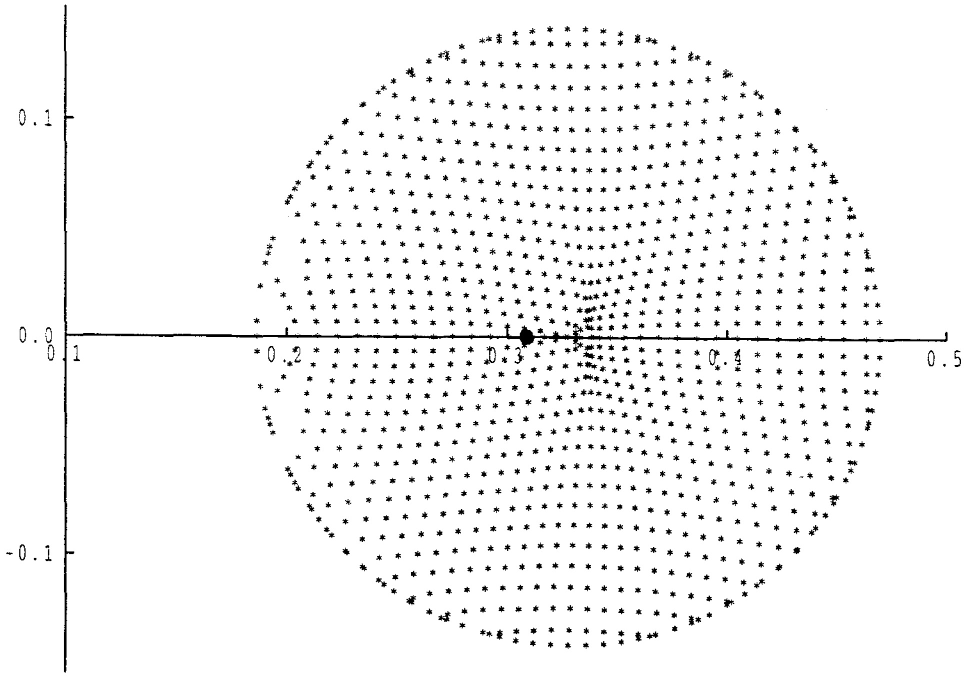


Figure 5.16: Solid dot shows the position (ir(i,j)=13) of minimum power density transfer ratio t_{12} across the dielectric surface for example 5.4 when D=2 dB.

μ	number of convergence iteration	μ	number of convergence iteration	other output
0.00	7	0.55	1	$n_x = 36$
0.05	2	0.60	1	$n_y = 18$
0.10	2	0.65	1	number of elements
0.15	1	0.70	1	are 1112
0.20	1	0.75	1	user cpu time is
0.25	1	0.80	1	75.660 second
0.30	1	0.85	1	
0.35	1	0.90	1	
0.40	1	0.95	1	
0.45	1	1.00	1	
0.50	1			

Table 5.8: Convergence detail for example 5.4 when D=5 dB.

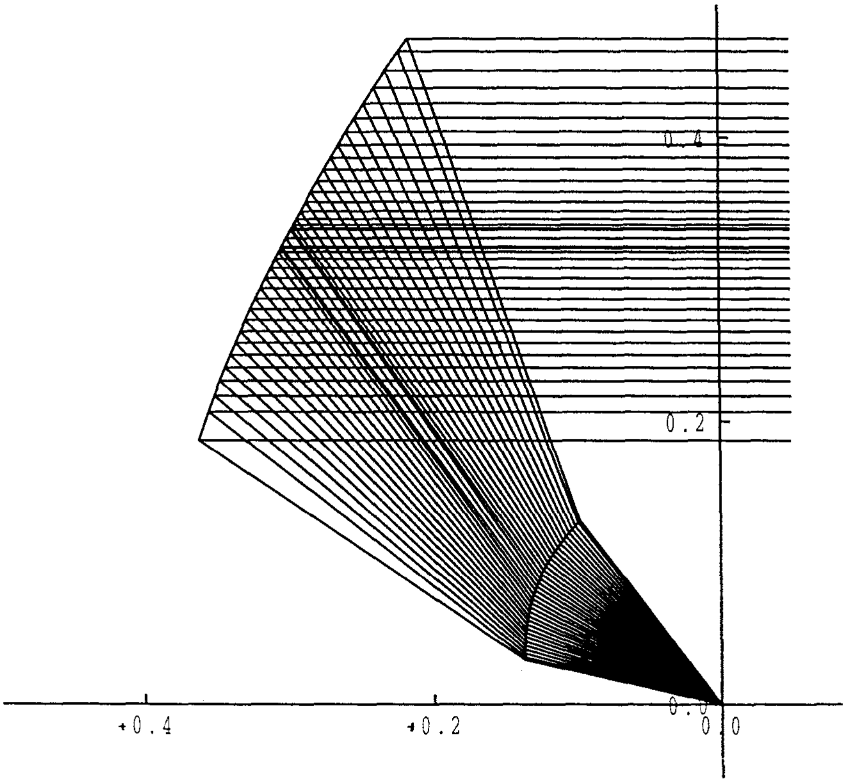


Figure 5.17: Ray diagram on the plane of the symmetry for example 5.4. when D=5 dB.

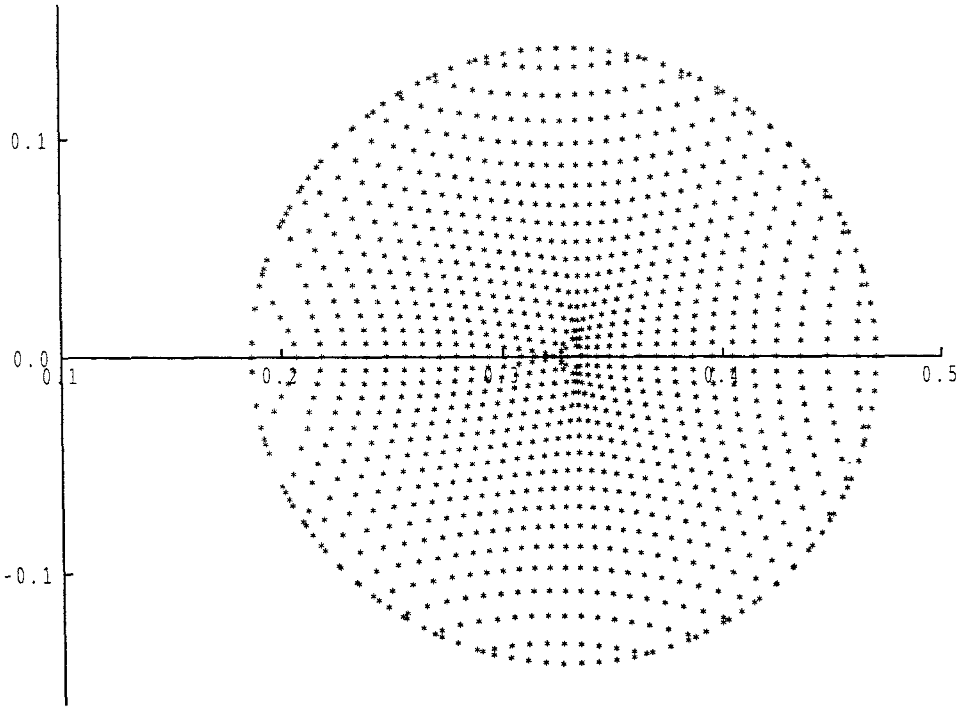


Figure 5.18: Mapping of uniform Cartesian grid onto circular region for example 5.4 when D=5 dB.

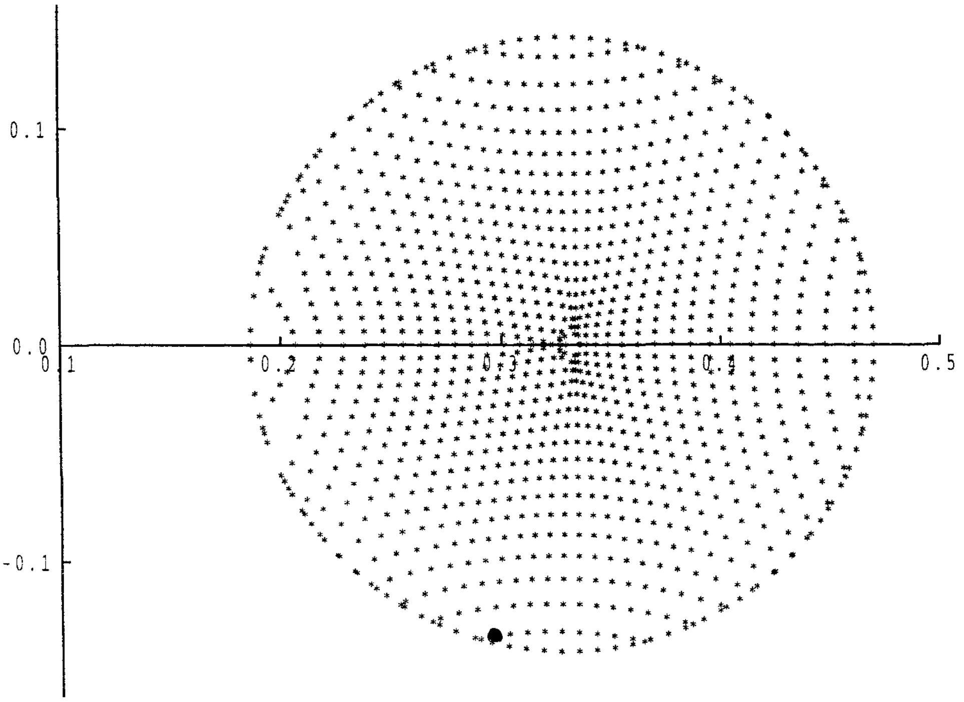


Figure 5.19: Solid dot shows the position (ir(i,j)=530) of maximum cross-polar for examples 5.4 when D=5 dB.

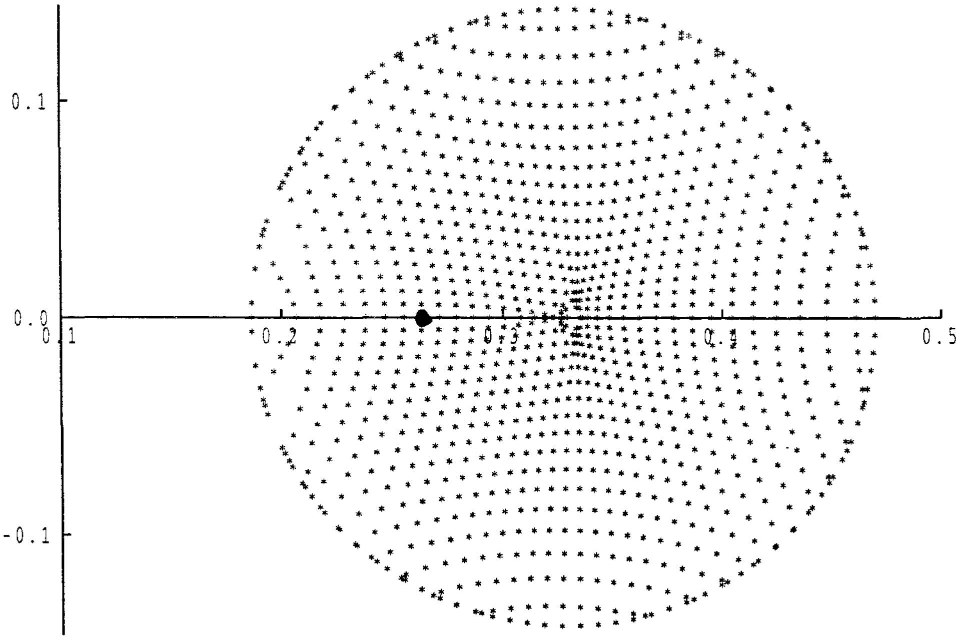


Figure 5.20: Solid dot shows the position (ir(i,j)=7) of minimum power density transfer ratio t_{12} across the dielectric surface for example 5.4 when D=5 dB.

μ	number of convergence iteration	μ	number of convergence iteration	other output
0.00	7	0.55	5	$n_x = 36$ $n_y = 18$ number of elements are 1112 user cpu time is 274.980 second
0.05	2	0.60	6	
0.10	3	0.65	6	
0.15	3	0.70	6	
0.20	3	0.75	7	
0.25	3	0.80	7	
0.30	4	0.85	8	
0.35	4	0.90	8	
0.40	4	0.95	9	
0.45	5	1.00	10	
0.50	5			

Table 5.9: Convergence detail for example 5.4 when D=10 dB.

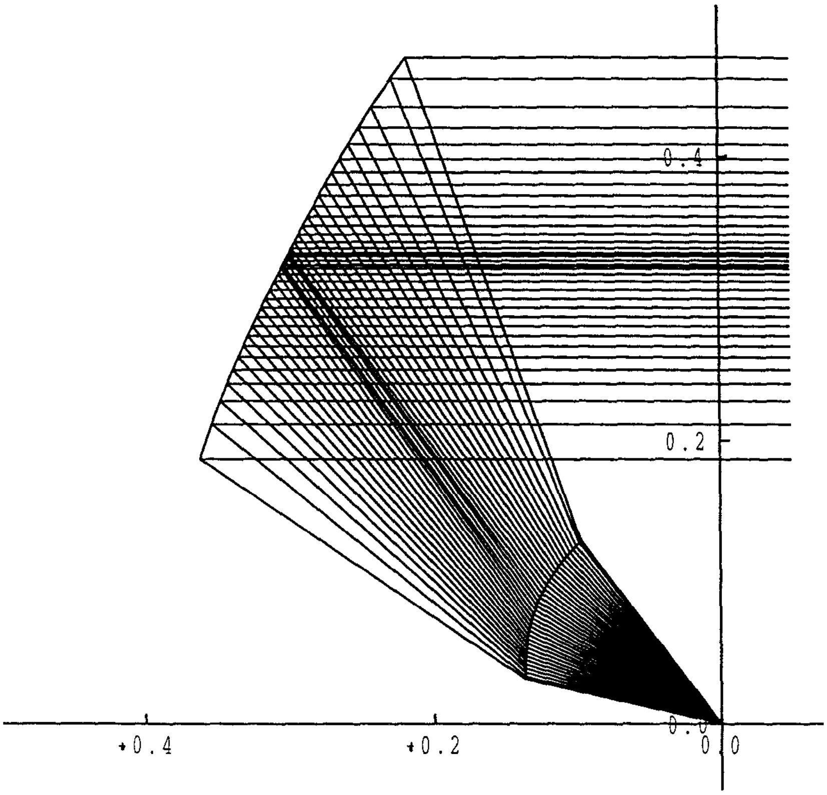


Figure 5.21: Ray diagram on the plane of the symmetry for example 5.4. when D=10 dB.

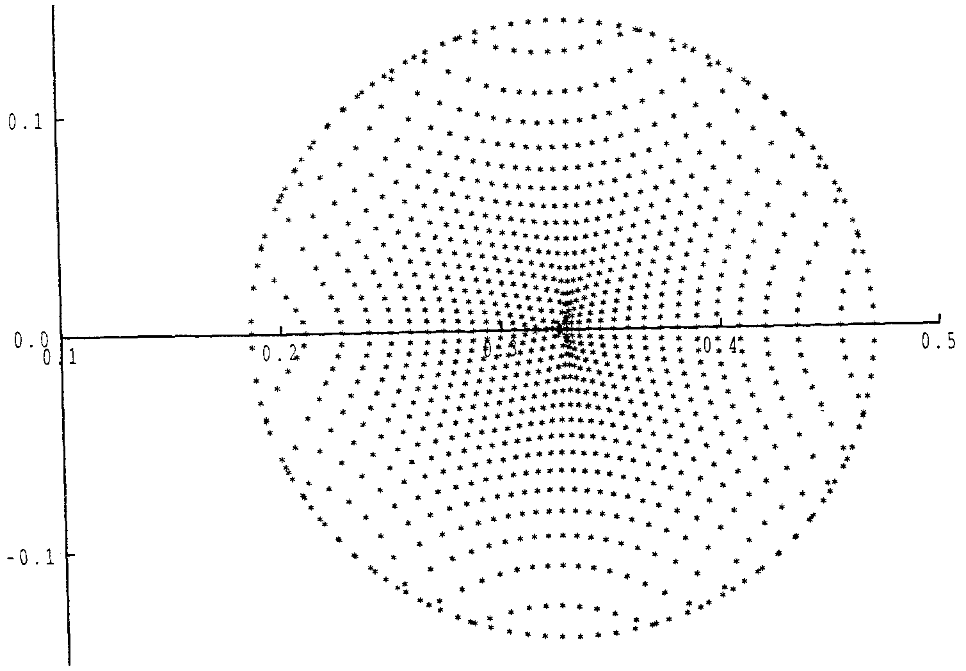


Figure 5.22: Mapping of uniform Cartesian grid onto circular region for example 5.4 when D=10 dB.

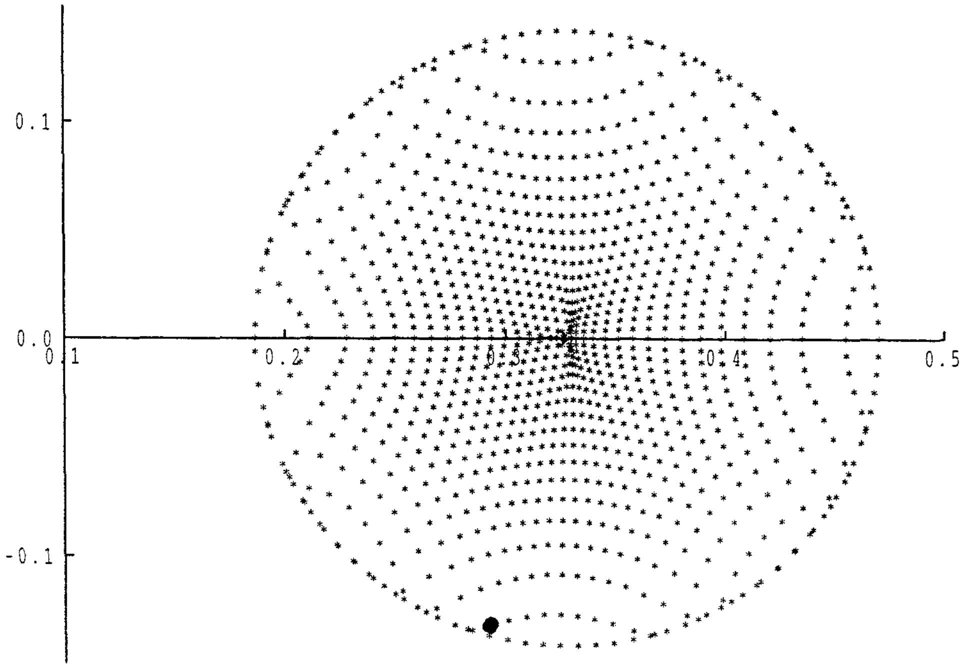


Figure 5.23: Solid dot shows the position (ir(i,j)=530) of maximum cross-polar for examples 5.4 when D=10 dB.

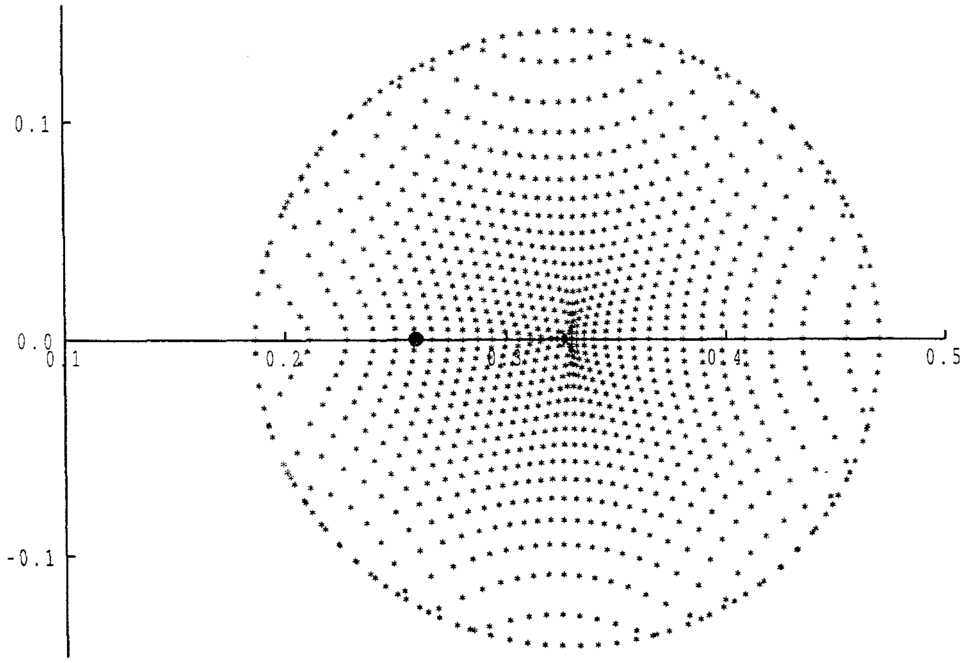


Figure 5.24: Solid dot shows the position (ir(i,j)=5) of minimum power density transfer ratio t_{12} across the dielectric surface for example 5.4 when D=10 dB.

μ	number of convergence iteration	μ	number of convergence iteration	other output
0.00	7	0.55	6	$n_x = 36$
0.05	2	0.60	7	$n_y = 18$
0.10	3	0.65	7	number of elements
0.15	3	0.70	8	are 1112
0.20	4	0.75	8	user cpu time is
0.25	4	0.80	9	334.750 second
0.30	4	0.85	10	
0.35	4	0.90	12	
0.40	5	0.95	14	
0.45	5	1.00	15	
0.50	6			

Table 5.10: Convergence detail for example 5.4 when D=12 dB.

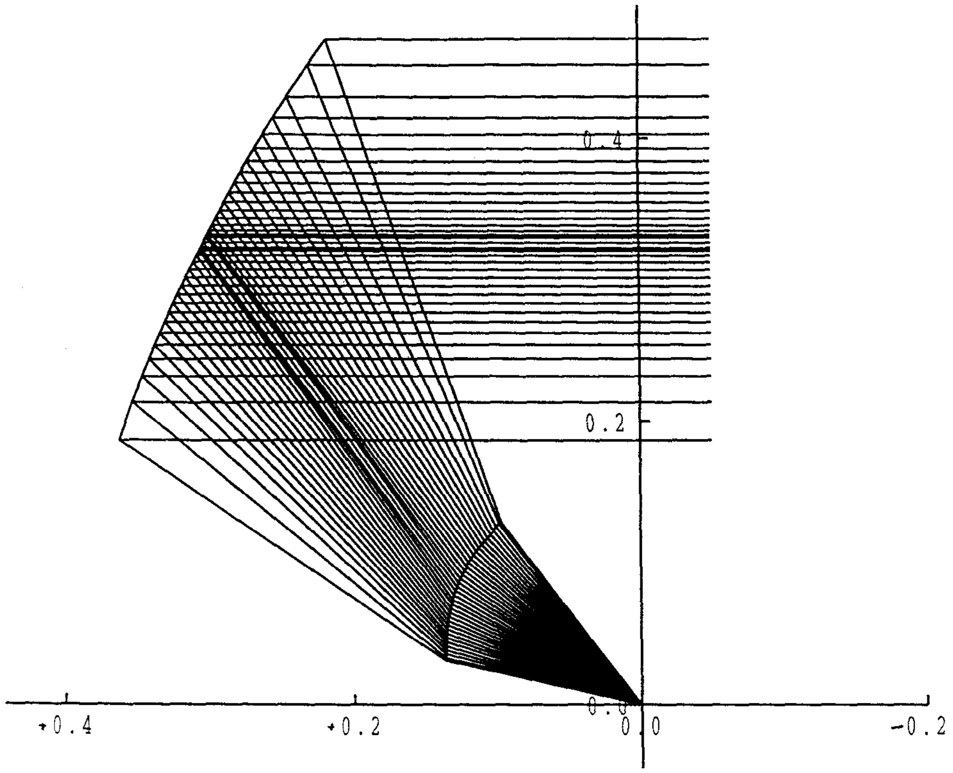


Figure 5.25: Ray diagram on the plane of the symmetry for example 5.4. when D=12 dB.

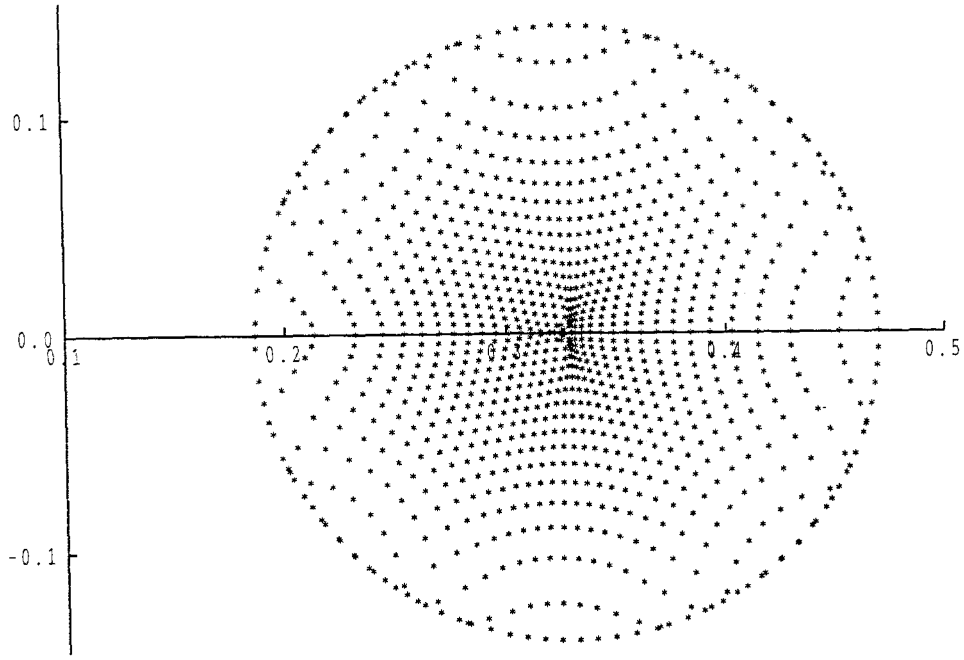


Figure 5.26: Mapping of uniform Cartesian grid onto circular region for example 5.4 when D=12 dB.

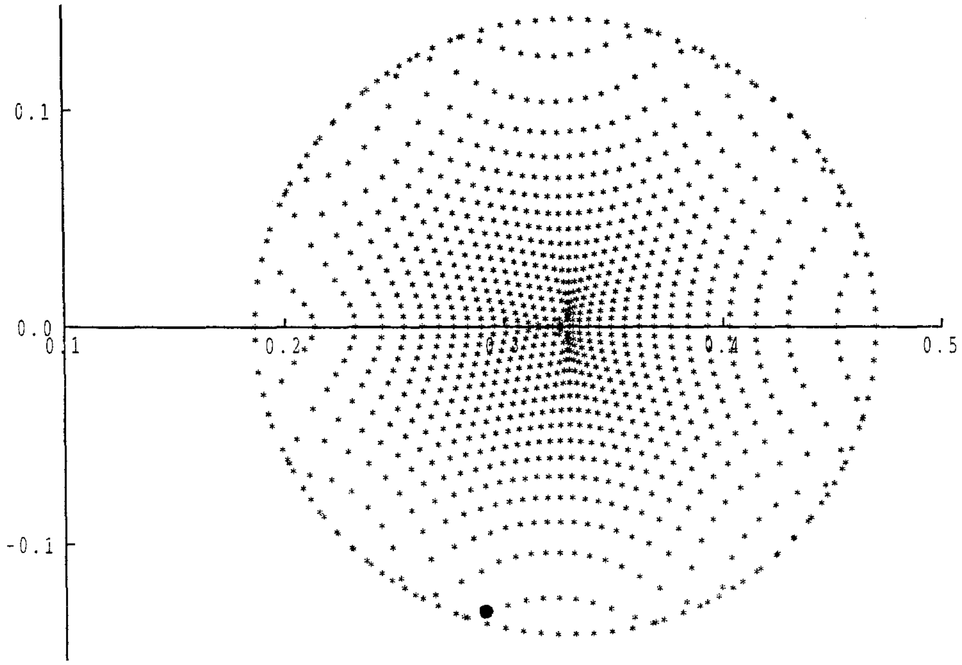


Figure 5.27: Solid dot shows the position (ir(i,j)=530) of maximum cross-polar for examples 5.4 when D=12 dB.

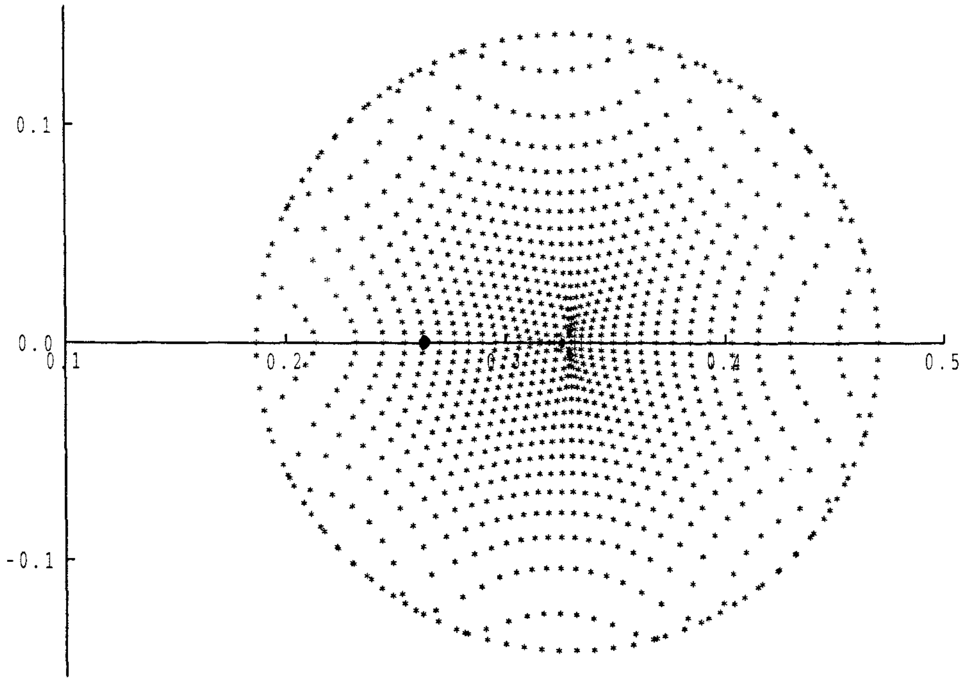


Figure 5.28: Solid dot shows the position (ir(i,j)=5) of minimum power density transfer ratio t_{12} across the dielectric surface for example 5.4 when D=12 dB.

5.8 Effect of varying input parameters

Example 5.5: Let nominal values of the input parameters be

$\gamma = 80$	$\theta_c = 20$	ψ=33	$\delta x = \delta y = 0.01$
N=2.6	e=0.22	k=0.14	D=5.

Table 5.11 shows that when the value of the semi-latus rectum k increases then the level of maximum cross-polar decreases.

k	δf	*max. cross-polar*
0.09	0.08	-28.90
0.12	0.10	-29.17
0.14	0.12	-29.24

Table 5.11: Table shows power density compensation and maximum cross-polar for example 5.5 for D=5 when the value of k is varied.

Table 5.12 shows that when the value of ψ is increased the level of maximum cross-polar is increased.

ψ	δf	*max. cross-polar*
31.5	0.120	-30.445
33.0	0.012	-29.241
35.0	0.010	-27.900

Table 5.12: Table shows power density compensation and maximum cross-polar level as ψ is varied.

Table 5.13 and 5.14 show that when the value of ellipsoid eccentricity e and refractive index N increase, the level of maximum cross-polar decreases.

e	δf	*max. cross-polar*
0.16	-0.08	-25.620
0.20	-0.12	-27.779
0.22	-0.17	-29.241

Table 5.13: Table shows power density compensation and maximum cross-polar as e is varied for example 5.5.

N	δf	*max. cross-polar*
2.0	-0.09	-24.699
2.2	-0.10	-25.951
2.8	-0.14	-31.565

Table 5.14: Table shows power density compensation and maximum cross-polar as N is varied for example 5.5.

Table 5.15 shows that when the value of the cone semi angle θ_c decreases then the level of maximum cross-polar also decreases.

θ_c	δf	*max. cross-polar*
20	-0.122	-29.241
18	-0.138	-30.660
17	-0.120	-30.988

Table 5.15: Table shows power density compensation and maximum cross-polar as θ_c is varied for example 5.5.

Table 5.16 shows that when the value of axis rotation angle γ increases then the level of maximum cross-polar increases.

γ	δf	*max. cross-polar*
80	-0.122	-29.241
90	-0.123	-29.050
92	-0.130	-28.960

Table 5.16: Table shows power density compensation and maximum cross-polar as γ is varied for example 5.5.

Additional results

In example 5.1 when the value of the semi-latus rectum k increases Table 5.17 shows that the level of maximum cross-polar decreases.

k	δf	*max. cross-polar*
0.09	-0.038	-14.7219
0.11	-0.054	-14.9082
0.13	-0.070	-14.9522
0.14	-0.083	-15.0081

Table 5.17: Table shows power density compensation and maximum cross-polar for examples 5.1 for D=5 when the value of k is varied.

Also Table 5.18 shows that when the value of ellipsoid eccentricity e increases, then the level of maximum cross-polar decreases.

e	δf	*max. cross-polar*
0.09	-0.065	-14.798
0.1	-0.070	-14.952
0.11	-0.085	-15.110
0.14	-0.100	-15.598

Table 5.18: Table shows power density compensation and maximum cross-polar for examples 5.1.

Tables 5.11, 5.13, 5.14, 5.17, and 5.18 shows that the peak cross-polarisation improves for two different configurations when the semi-latus rectum, eccentricity and permittivity of lens are increased. In all examples the value of ψ has been chosen so that there is no blockage.

5.9 Conclusion

A dielectric horn offset fed reflector antenna, in which the dielectric lens is incorporated into the feed horn is analysed on the basis of geometrical optics. It has been shown that the numerical technique introduced in chapter 4, may be adopted for the synthesis of the system. Finite difference forms of the elliptic differential equation and boundary condition are solved by a generalised Newton method. Anchoring of the mapping function, L, is necessary, but the stability of the mapping is very sensitive to the parameter δf and may manifest itself, by a large deviation in the power level calculated for points in the neighbourhood of the anchor point where the Monge-Ampère constraint is not imposed.

In all of the examples considered here the anchor point effect has been greatly reduced by a careful choice of reflector geometry and the level of compensation of the power normalisation factor necessary to produce satisfactory solution to the boundary-value problem. These results, together with cross-polarisation calculation, are presented. In general the cross-polar level depends on the shape of the dielectric surface, and for the examples chosen, improves when ψ is decreased or the permittivity is increased. Also it has been found that the peak cross-polarisation increases when the size of the lens cone angle is increased, or when the rotation axis angle γ is increased.

The power density transfer coefficient t_{12} was monitored for many examples. This gave a quantitative measure of the loss of energy transmitted through the dielectric surface caused by reflections. The maximum value of t_{12} is unity for perfect transmission. The minimum value was shown to be as low as 0.684210 in the examples investigated.

A range of parameters values relating to the system geometry has been considered for a single reflector with dielectric cone feed in which both the source cone and aperture are circular. Further research is needed to optimise the system geometry in any application.

Appendix A

Polarisation of refracted ray

To obtain the complex coordinate of $\mathbf{e_2}$, we know that $\mathbf{b} \times \mathbf{t} / \sin\phi$ is a unit vector orthogonal to $\mathbf{t}$, hence its complex coordinate can be written in the form of equation (2.58) as follows

$$\frac{1+\delta_1\xi}{\delta_1-\bar{\xi}}, \qquad |\delta_1| = 1. \tag{A.1}$$

To find δ_1, from equation (1.14) we have

$$\frac{\mathbf{p}\times\mathbf{t}}{\sin\phi} = \frac{i\left[\eta\xi(\bar{\eta}-\bar{\xi})+\eta-\xi\ ,\ \eta\bar{\xi}-\bar{\eta}\xi\right]}{|\xi-\eta||1+\eta\bar{\xi}|}, \tag{A.2}$$

with complex coordinate

$$\frac{i\left[\eta\xi(\eta-\bar{\xi})+\eta-\xi\right]}{|\xi-\eta||1+\eta\bar{\xi}|-i(\eta\bar{\xi}-\bar{\eta}\xi)}. \tag{A.3}$$

From equations (A.1) and (A.3) we obtain that

$$\delta_1 = ie^{(-i\beta)}\ , \tag{A.4}$$

where

$$\beta = \arg\left(\frac{\xi-\eta}{1+\eta\bar{\xi}}\right). \tag{A.5}$$

The unit vector

$$(\mathbf{t}\cos\phi-\mathbf{p})/\sin\phi, \tag{A.6}$$

is also orthogonal to $\mathbf{t}$ and its complex coordinate can be written in the form

$$\frac{1+\delta_2\xi}{\delta_2-\bar{\xi}}, \qquad |\delta_2| = 1. \tag{A.7}$$

By using equation (1.11) to express $\mathbf{t}(\eta)$ and $\mathbf{p}(\xi)$ in terms of their complex coordinate, we can show that the complex coordinate of (A.6) is

$$\frac{2\xi(|\eta|^2+1)\cos\phi - 2\eta(|\xi|^2+1)}{(|\eta|^2+1)(|\xi|^2+1)\sin\phi - (|\eta|^2+1)(|\xi|^2-1)\cos\phi + (|\eta|^2-1)(|\xi|^2+1)} \tag{A.8}$$

By substituting from equation (1.13) for $\cos\phi$ and from equation (2.60) for $\sin\phi$ in equation (A.8) and equating the expressions (A.7), (A.8) gives

$$\delta_2 = ie^{(-i\beta)}\ , \tag{A.9}$$

where β is defined by equation (A.5).

If we express the vectors (A.2) and (A.6) in terms of their complex coordinates δ_1 , δ_2 and substitute in (2.43) we obtain

$$\begin{aligned}
\mathbf{e_2} &= \frac{AT_\perp(\bar{\delta}_1-\delta_1\xi^2\ ,\ \delta_1\xi+\bar{\delta}_1\bar{\xi}) + BT_{||}(\bar{\delta}_2-\delta_2\xi^2\ ,\ \delta_2\xi+\bar{\delta}_2\bar{\xi})}{(|\xi|^2+1)\left(A^2T_\perp^2+B^2T_{||}^2\right)^{\frac{1}{2}}}, && \text{(A.10)}\\
&= \frac{(\bar{\gamma}-\gamma\xi^2\ ,\ \gamma\xi+\bar{\gamma}\bar{\xi})}{1+|\xi|^2}, && \text{(A.11)}
\end{aligned}$$

where

$$\gamma = \frac{AT_\perp\delta_1 + BT_{||}\delta_2}{\left(A^2T_\perp^2+B^2T_{||}^2\right)^{\frac{1}{2}}}, \qquad |\gamma| = 1. \tag{A.12}$$

By combining equations (2.42), (2.62), (2.63), (A.4), (A.5), (A.9) and (A.12) we obtain that

$$\gamma = \frac{\epsilon(1+\bar{\xi}\eta)^2 + \bar{\epsilon}(\bar{\xi}-\bar{\eta})^2}{|\epsilon(1+\bar{\xi}\eta)^2 + \bar{\epsilon}(\bar{\xi}-\bar{\eta})^2|} \tag{A.13}$$

(see Westcott (1993)).

Appendix B

The Cartesian grid

The η-domain is governed by the cross-section of the source cone which we take to be circular. The half domain is therefore, a semi circle and a Cartesian grid is used to subdivide the region as illustrated in Fig.B.1.

Grid lines for which x is a constant are numbered by the integer i and those for which y is a constant are numbered by the integer j. The maximum value of i and j are given by n_x and n_y respectively. Both i and j grid lines will usually contribute separate boundary points although it is possible for two grid lines to intercept the boundary at the same point. Any point at which an i grid line intercepts the curved boundary will be referred to as an "i boundary point" and, similarly, the interception of a j grid line with the boundary will be referred to as a "j boundary point". The point of intersection of an i and j grid line which is also coincident with curved boundary will be referred to as an "ij boundary point" (see Fig.B.2). To avoid possible inaccuracies arising from small separation of grid points, any boundary points falling within a distance ϵ of an internal point is not counted separately when forming the grid. For the radius of R, ϵ is commonly set to be order of (R/1000).

The number of grid points lying along i and j grid line depends upon the values dx and dy. Here to provide a uniform grid we choose dx=dy for two successive internal points on each grid line. Each internal point is identified by the j value of the intercepting grid lines. We denote the j value of the final point on the line by jmax(i) so that, for a given i grid line, $0 \leq j \leq jmax(i)$. If this final point does not lie on a j grid line, jmax(i) is assigned the value of the j grid line immediately external to the boundary.

Similarly, the i value of grid points along a given j grid line take minimum and maximum values which will be denoted by imin(j) and imax(j). Hence, for any j grid line $imin(j) \leq i \leq imax(j)$ where imin(j) and imax(j) are assigned the value of the i grid lines on, or immediately external to the boundary. A further quantity which is found to be useful when applying finite difference formula is the i value, denoted $iapex$, for which the y coordinate of the corresponding boundary point is the maximum attained for points on the boundary.

The grid increments may be chosen as noted above to provide a uniform grid. However, in general the increment values depend on both i and j (see for example, Fig.B.3). Hence x and y increments are identified by two subscripts and are denoted by $\delta x_{ij} = \delta x_{(i,j)}$ and δy_{ij} respectively.

B.1 Grid numbering function

A systematic method characterising each grid point by a unique integer value is required for the matrix representation of the boundary-value problem. However, a difficulty arises for points on the boundary since a given i,j combination may not be unique to one grid point. An example of this illustrated in Fig.B.4 in which both the i and j boundary points shown are characterised by the same i, j value.

The following numbering technique treats internal points, j boundary points and ij boundary points. Starting at grid (0,0) and counting the number of points along each successive j grid line, a grid point (i,j) is assigned the the integer value ir(i,j) where, for $1 \leq j \leq n_y - 1$

$$ir(i,j) = j + (i - imin(j)) + \sum_{n=0}^{j-1}(imax(n) - imin(n))$$

where j=0 ,ir(i,j)=i-imin(j). Therefore the numbering function starts at the value ir(0,0)=0.

The remaining grid points are boundary points and numbered by irb(i) which continues counting from the maximum value of ir(i,j). Hence

$$irb(i) = \sum_{n=0}^{i} ibind(n) + max(ir),$$

where max(ir) is the maximum value of ir(i,j); ibind(i) takes the value 0 if row i has an ij boundary point and takes the value 1 if row i has a separate i boundary point.

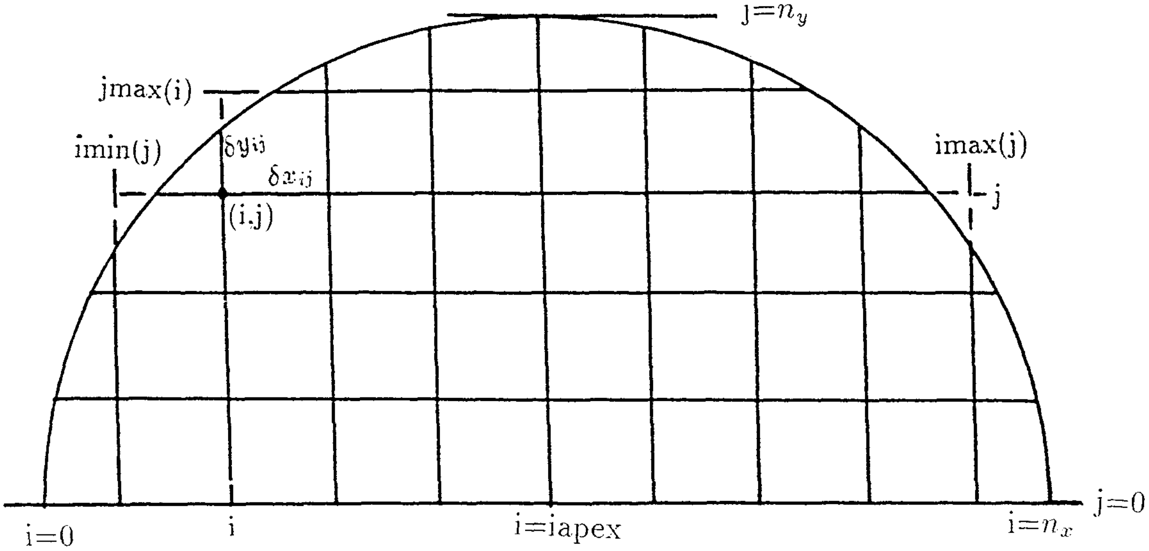


Fig.B.1: Cartesian grid on the half source

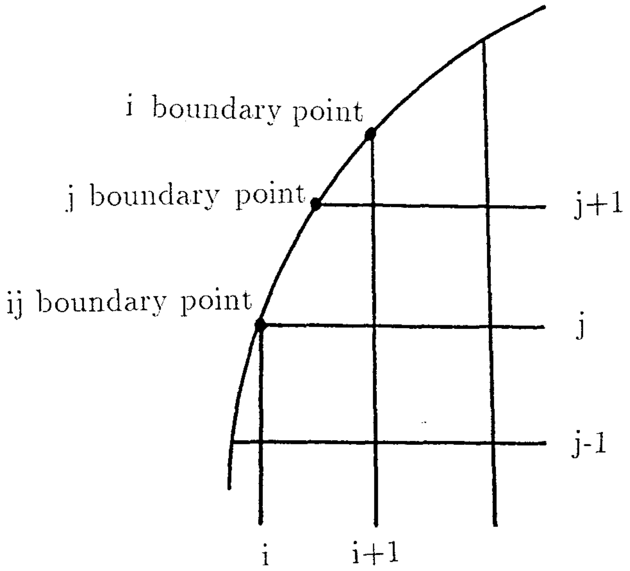


Fig.B.2: Classification of boundary points

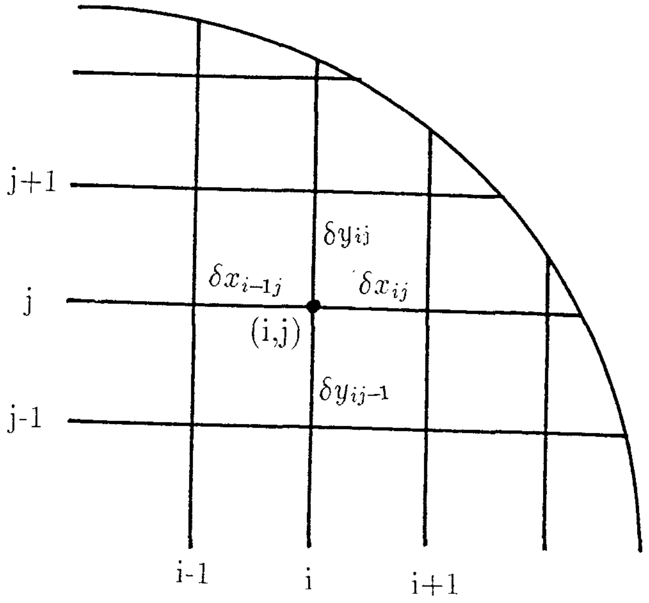


Fig.B.3: Grid increments of point (i,j)

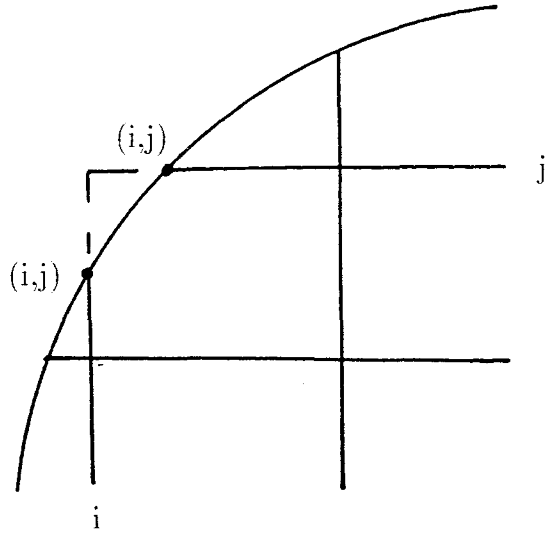


Fig.B.4: Ambiguity in identification of i and j boundary

Appendix C

Cartesian finite difference scheme

In order to be able to calculate the derivative of L on the aperture, under the linearised boundary condition, a finite difference scheme is used in conjunction with the Cartesian grid. Derivatives of L with respect to x and y are thus approximated at any grid point in terms of the values of the function at this, and neighbouring, grid points.

It is noted by Ames(1965) that the simplest finite difference structure that may be used in the solution of partial differential equations containing the mixed derivative, is the nine point molecule. This is illustrated in Fig.C.1.

Application of Taylor's theorem enables us to approximate the finite difference formulae for various partial derivatives using this molecule.

C.1 Derivatives of L at the internal points

At internal grid points, when $1 \leq j \leq n_y - 1$ and $imin(j)+1 \leq i \leq imax(j)-1$, finite difference formulae are required for L_x, L_y, L_{xx}, L_{yy} and L_{xy}. Each (i,j) internal point will be enclosed by at least one other point in either direction i or j grid lines, as shown in Fig.C.1. Thus, three point central difference forms may be used to approximate the derivatives L_x, L_{xx}, L_y, L_{yy} as follows

$$\begin{aligned} L_x &= \frac{-\delta x_{ij}}{\delta x_{i-1j}\,(\delta x_{ij}+\delta x_{i-1j}\,)} L_{i-1j} \\ &+ \frac{\delta x_{ij}-\delta x_{i-1j}}{\delta x_{ij}\delta x_{i-1j}} L_{ij} + \frac{\delta x_{i-1j}}{\delta x_{ij}(\delta x_{ij}+\delta x_{i-1j}\,)} L_{i+1j} \end{aligned} \tag{C.1}$$

$$\begin{aligned} L_{xx} &= \frac{2}{\delta x_{i-1j}\,(\delta x_{ij}+\delta x_{i-1j}\,)} L_{i-1j} - \frac{2}{\delta x_{ij}\delta x_{i-1j}} L_{ij} \\ &+ \frac{2}{\delta x_{ij}(\delta x_{ij}+\delta x_{i-1j}\,)} L_{i+1j} \end{aligned} \tag{C.2}$$

$$\begin{aligned} L_y &= \frac{-\delta y_{ij}}{\delta y_{ij-1}\,(\delta y_{ij}+\delta y_{ij-1}\,)} L_{ij-1} \\ &+ \frac{\delta y_{ij}-\delta y_{ij-1}}{\delta y_{ij}\delta y_{ij-1}} L_{ij} + \frac{\delta y_{ij-1}}{\delta y_{ij}(\delta y_{ij}+\delta y_{ij-1}\,)} L_{ij+1} \end{aligned} \tag{C.3}$$

$$\begin{aligned} L_{yy} &= \frac{2}{\delta y_{ij-1}\,(\delta y_{ij}+\delta y_{ij-1}\,)} L_{ij-1} - \frac{2}{\delta y_{ij}\delta y_{ij-1}} L_{ij} \\ &+ \frac{2}{\delta y_{ij}(\delta y_{ij}+\delta y_{ij-1}\,)} L_{ij+1} \end{aligned} \tag{C.4}$$

where $\delta x_{i+1j} = \delta x_{(i+1,j)}$, δx_{ij} and δy_{ij} are the incremental lengths in the i and j directions respectively. To approximate the mix derivative L_{xy}, depending upon the position of point i,j in relation to boundary. The nine grid points (i-1,j-1), (i,j-1), (i+1,j-1), (i-1,j), (i,j), (i+1,j), (i-1,j+1), (i,j+1) and (i+1,j+1) surrounded the point (i,j), defining the corners of a rectangle, as shown in Fig.C.2, for example, first central difference with respect to y and then central difference with respect to x may be used to evaluate L_{xy}. Hence

$$\begin{aligned} L_{xy} &= \frac{-\delta y_{ij}}{\delta y_{ij-1}\,(\delta y_{ij}+\delta y_{ij-1}\,)} \Bigg(\frac{-\delta x_{ij-1}}{\delta x_{i-1j-1}\,(\delta x_{ij-1}\,+\delta x_{i-1j-1}\,)} L_{i-1j-1} \\ &+ \frac{\delta x_{ij-1}\,-\delta x_{i-1j-1}}{\delta x_{ij-1}\,\delta x_{i-1j-1}} L_{ij-1} \\ &+ \frac{\delta x_{i-1j-1}}{\delta x_{ij-1}\,(\delta x_{ij-1}\,+\delta x_{i-1j-1}\,)} L_{i+1j-1}\,\Bigg) \\ &+ \frac{\delta x_{ij}-\delta y_{ij-1}}{\delta y_{ij}\delta y_{ij-1}} \Bigg(\frac{-\delta x_{ij}}{\delta x_{i-1j}\,(\delta x_{ij}+\delta x_{i-1j}\,)} L_{i-1j} \\ &+ \frac{\delta x_{ij}-\delta x_{i-1j}}{\delta x_{ij}x_{i-1j}} L_{ij} \end{aligned} \tag{C.5}$$

$$
\begin{aligned}
&+ \quad \frac{\delta x_{i-1j}}{\delta x_{ij}(\delta x_{ij} + \delta x_{i-1j})} L_{i+1j} \Bigg) \\
&+ \quad \frac{\delta y_{ij-1}}{\delta y_{ij}(\delta y_{ij} + \delta y_{ij-1})} \Bigg(\frac{-\delta x_{ij+1}}{\delta x_{i-1j+1}\,(\delta x_{ij+1} + \delta x_{i-1j+1})} L_{i-1j+1} \\
&+ \quad \frac{\delta x_{ij+1} - \delta x_{i-1j+1}}{\delta x_{ij+1}\, x_{i-1j+1}} L_{ij+1} \\
&+ \quad \frac{\delta x_{i-1j+1}}{\delta x_{ij+1}(\delta x_{ij+1} + \delta x_{i-1j+1})} L_{i+1j+1} \Bigg)
\end{aligned}
$$

In the vicinity of the curved boundary, points are on the boundary, a forward or backward difference formulae with respect to x are used according to whether $i < iapex$ or $i \geq iapex$ (see Appendix B), then the central difference with respect to y may be used to evaluate L_{xy}. In the first case for example, the points indicated in Fig.C.3 are used to evaluate L_{xy}. Hence

$$
\begin{aligned}
L_{xy} \quad &= \quad \frac{-(\delta x_{i+1j} + 2\delta x_{ij})}{\delta x_{ij}(\delta x_{ij} + \delta x_{i+1j})} \Bigg(\frac{-\delta y_{ij}}{\delta y_{ij-1}\,(\delta y_{ij} + \delta y_{ij-1})} L_{ij-1} \\
&+ \quad \frac{\delta y_{ij} - \delta y_{ij-1}}{\delta y_{ij} y_{ij-1}} L_{ij} \\
&+ \quad \frac{\delta y_{ij-1}}{\delta y_{ij}(\delta y_{ij} + \delta y_{ij-1})} L_{ij+1} \Bigg) \\
&+ \quad \frac{\delta x_{ij} + \delta x_{i+1j}}{\delta x_{ij} \delta x_{i+1j}} \Bigg(\frac{-\delta y_{i+1j}}{\delta y_{i+1j-1}\,(\delta y_{i+1j} + \delta y_{i+1j-1})} L_{i+1j-1} \\
&+ \quad \frac{\delta y_{i+1j} - \delta y_{i+1j-1}}{\delta y_{i+1j} y_{i+1j-1}} L_{i+1j} \\
&+ \quad \frac{\delta y_{i+1j-1}}{\delta y_{i+1j}(\delta y_{i+1j} + \delta y_{i+1j-1})} L_{i+1j+1} \Bigg) \\
&+ \quad \frac{-\delta x_{ij}}{\delta x_{i+1j}(\delta x_{ij} + \delta x_{i+1j})} \Bigg(\frac{-\delta y_{i+2j}}{\delta y_{i+2j-1}\,(\delta y_{i+2j} + \delta y_{i+2j-1})} L_{i+2j-1} \\
&+ \quad \frac{\delta y_{i+2j} - \delta y_{i+2j-1}}{\delta y_{i+2j} y_{i+2j-1}} L_{i+2j} \\
&+ \quad \frac{\delta y_{i+2j-1}}{\delta y_{i+2j}(\delta y_{i+2j} + \delta y_{i+2j-1})} L_{i+2j+1} \Bigg) \qquad \text{(C.6)}
\end{aligned}
$$

If $i \geq iapex$, the points indicated in Fig.C.4 are used in the backward difference form.

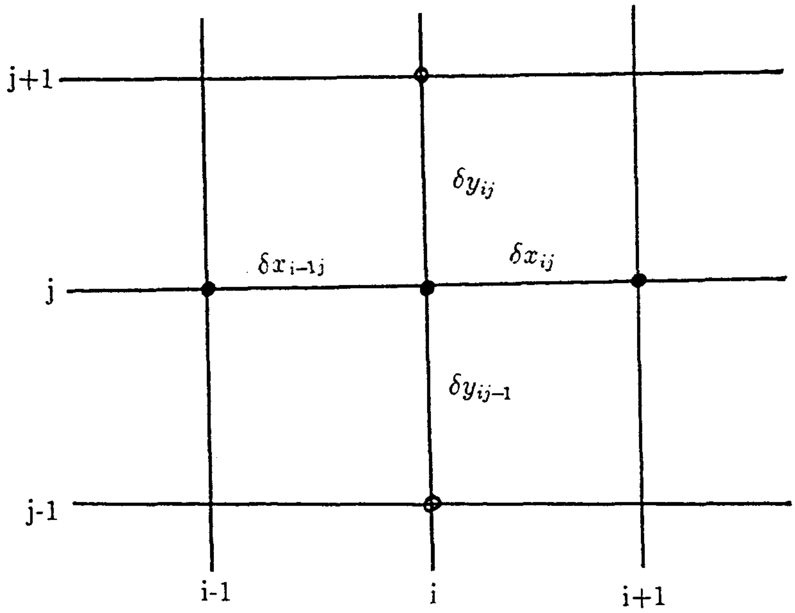


Fig.C.1: Point required for evaluation of L_x, L_{xx}, L_y *and* L_{yy}

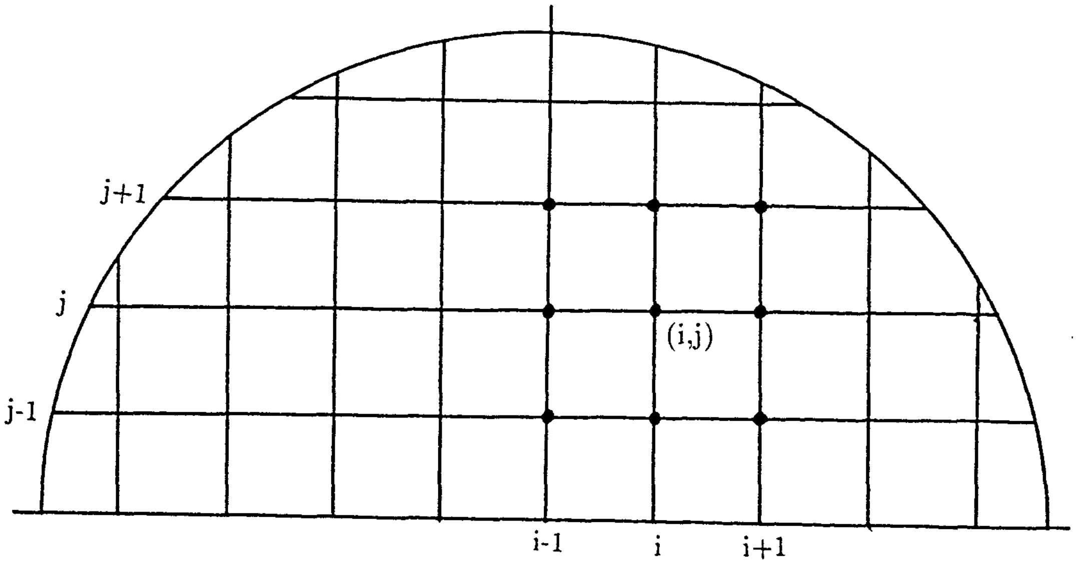


Fig.C.2: Illustaration of the nine point molecule for the mixed derivative for internal points

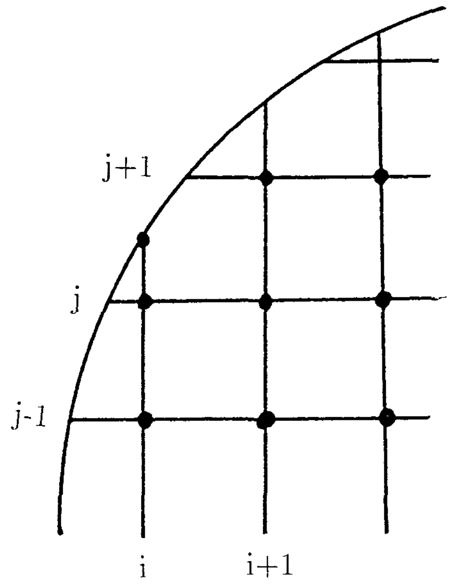


Fig.C.3: Points used for forward evaluation of L_{xy} when $i < iapex$

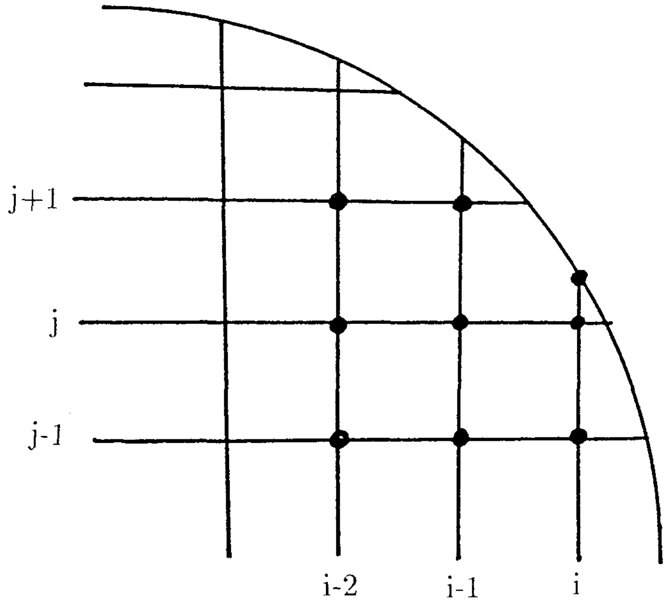


Fig.C.4: Points used for backward evaluation of L_{xy} when $i \geq iapex$

C.2 Derivatives of L on the boundary points

The boundary condition (2.88) involves only the first order derivatives L_x, and L_y. At i and j boundary points, the distribution of the neighbouring grid points is unsuitable for the direct application of a conventional difference form for L_x and L_y. Thus, a Lagrange interpolation technique is used to provide values of L at the points required by the chosen finite difference scheme.

We discussed in the following section the treatment of j boundary points, and the methods employed similarly for i boundary points.

C.2.1 j boundary points

Finite difference scheme for L_x

Since j boundary points are situated on j grid lines, points are available for 3-point forward or backward difference formulae for L_x depending on whether $i < iapex$ or $i \geq iapex$, respectively (see Fig.C.5). Note that the grid must be chosen so that all j grid lines contain at least three grid points. Hence, for example, the forward difference formulae for L_x on the j boundary can be written as follows

$$\begin{aligned} L_x &= \frac{-\delta x_{i+1j} + 2\delta x_{ij}}{\delta x_{ij}(\delta x_{ij} + \delta x_{i+1j})} L_{ij} + \frac{\delta x_{ij} + \delta x_{i+1j}}{\delta x_{ij}\delta x_{i+1j}} L_{i+1j} \\ &+ \frac{-\delta x_{ij}}{\delta x_{i+1j}(\delta x_{ij} + \delta x_{i+1j})} L_{i+2j} \end{aligned} \tag{C.7}$$

where grid point (i, j) is an j boundary point.

Finite difference scheme for L_y

A similar 3-point approximation is not possible for L_y since the j boundary point does not lie on a i grid line. Thus an interpolation (when $j \geq 1$) or extrapolation (when $j = 0$ ($i = 0$ or $i = n_x$)) technique involving neighbouring grid points may be used to predict values of L at the positions required by a conventional 3-point formula (see for example, Fig.C.6). In Fig.C.6 when $j \geq 1$ interpolated values of L at the positions A and B would allow a finite difference form based on a 3-point backward difference scheme to be used in approximating L_y at the boundary point.

A simple interpolation scheme to predict a value at position A for example, is a three point linear Lagrange interpolation, which interpolates the value of A between the points i and i+1 on the same j grid line as A.

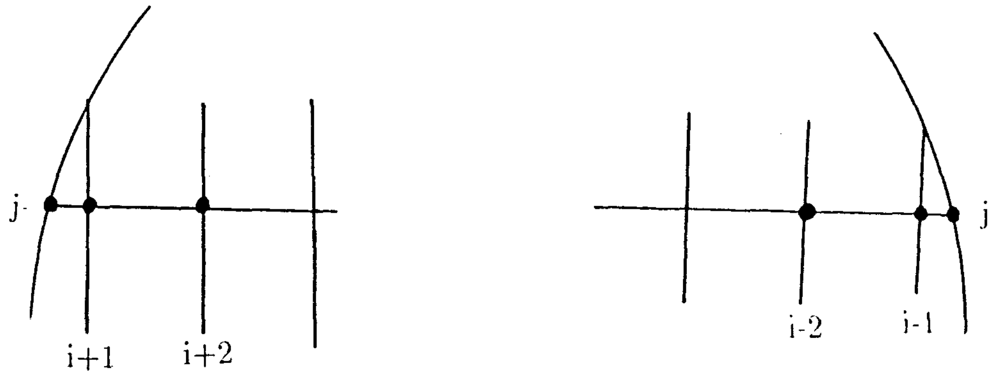
C.2.2 i boundary points

Finite difference scheme for L_y

A backward difference approximation is used to evaluate L_y on the boundary points when $i \geq 1$.

Finite difference scheme for L_x

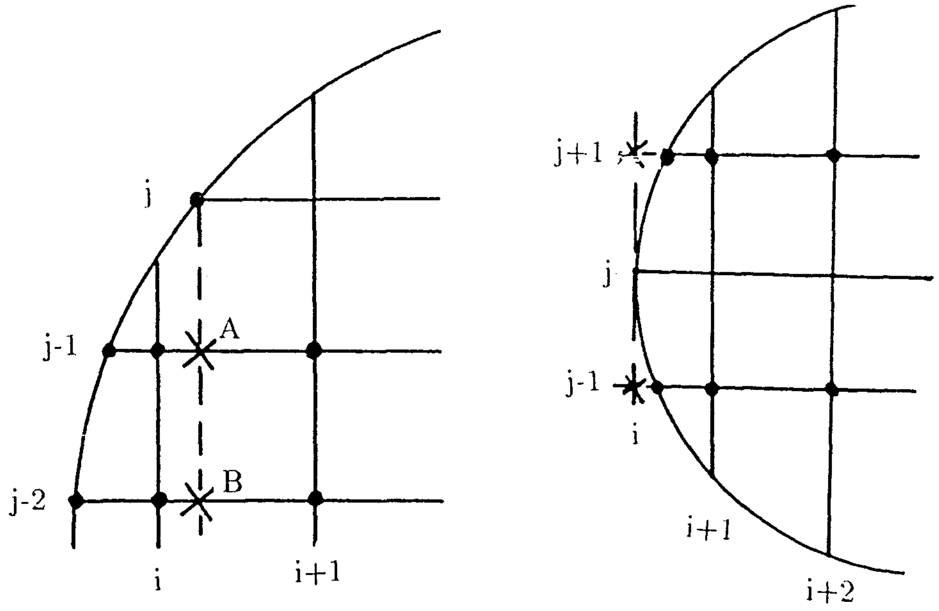
Forward or backward Lagrange 3-point interpolation along i grid lines is used in conjunction with suitable finite difference forms (see Fig.C.7). Lagrange 3-point extrapolation when i=iapex is used to evaluate L_x (see Fig.C.8).



forward difference

backward difference

Fig.C.5: Points used to evaluate L_x



backward difference

central difference

Fig.C.6: Points used for backward and central difference approximation of L_y

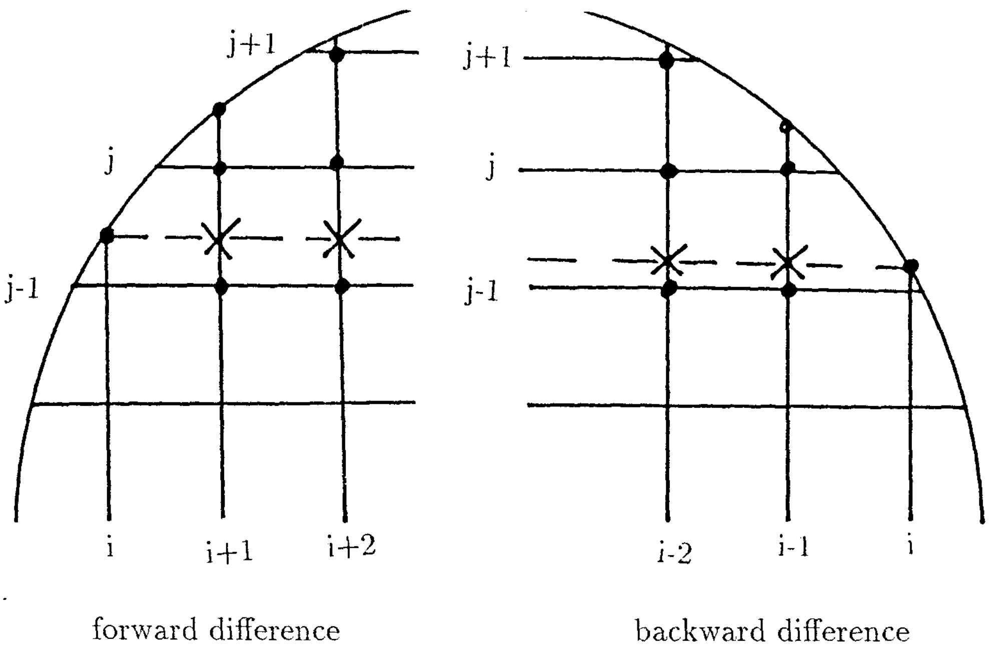


Fig.C.7: Points used for forward and backward difference approximation of L_x

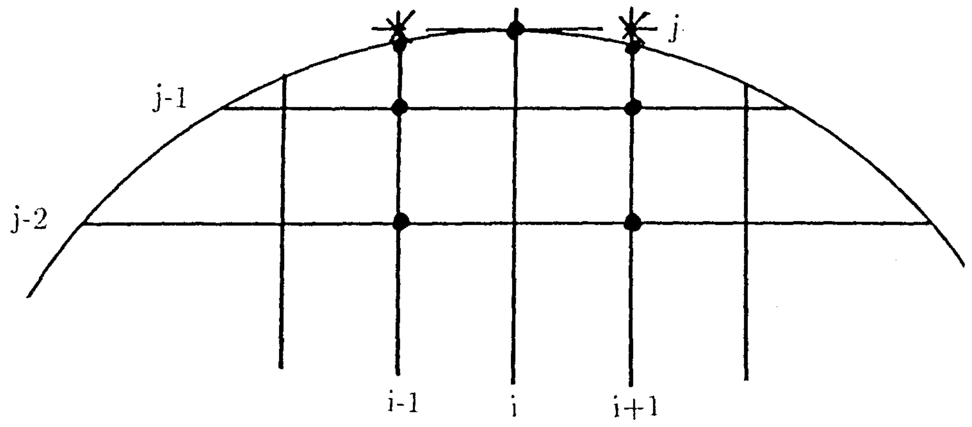


Fig.C.8: Points required to evaluate L_x when $i = iapex$ (central difference approximation)

C.3 Three point Lagrange interpolation for i boundary points

Suppose an interpolated value of L is required at the point C on the line i. We choose three close points to this boundary point as shown in Fig.C.9. We label these points 1, 2 and 3. Let y_{i1}, y_{i2}, and y_{i3}, be the corresponding y-coordinate and L_{i1}, L_{i2} and l_{i3} the mapping function values at these points. The interpolated value at point C(i, j) is then given by

$$L^{(i)} = l_{i1}L_{i1} + l_{i2}L_{i2} + l_{i3}L_{i3} = \sum_{k=1}^{3} l_{ik}L_{ik} \tag{C.8}$$

where the Lagrangian coefficients are

$$\begin{aligned} l_{i1} &= (y_{ic} - y_{i2})(y_{ic} - y_{i3})/(y_{i1} - y_{i2})(y_{i1} - y_{i3}) && \text{(C.9)} \\ l_{i2} &= (y_{ic} - y_{i1})(y_{ic} - y_{i3})/(y_{i2} - y_{i1})(y_{i2} - y_{i3}) && \text{(C.10)} \\ l_{i3} &= (y_{ic} - y_{i1})(y_{ic} - y_{i2})/(y_{i3} - y_{i1})(y_{i3} - y_{i2}) && \text{(C.11)} \end{aligned}$$

where y_{ic} is the y-coordinate of C and superscript(i) denotes interpolation on grid line i.

To evaluate L_x at an i boundary point, interpolated values will be required on two neighbouring i grid lines (see Fig.7). The finite difference form for L_x may now be derived using the following (a), (b) and (c) scheme.

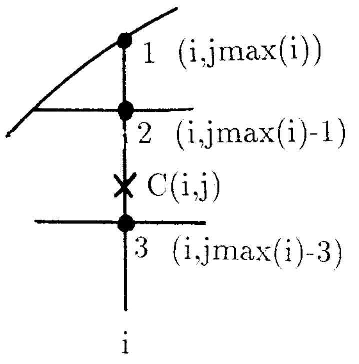


Fig.C.9: Points used for evaluating Lagrange interpolation on i grid line

a $1 \leq j \leq n_y - 1$ **and** $i < iapex$

In this case where n_y is the maximum value of j a forward difference scheme for L_x is appropriate (see for example, Fig.C.7). Interpolated values $L^{(i+1)}$ and $L^{(i+2)}$ are therefore required on grid lines i+1 and i+2, at the positions indicated by crosses. The 3-point forward difference formulae may be written as follow

$$L_x = d_i L_{i+1} + d_{i+1} L^{(i+1)} + d_{i+2} L^{(i+2)} \tag{C.12}$$

where the coefficients d_i, d_{i+1} and d_{i+2} depend on the grid increments δx_{ij} and δx_{i+1j}. Hence

$$d_i = -(\delta x_{i+1j} + 2\delta x_{ij})/\delta x_{ij}(\delta x_{ij} + \delta x_{i+1j}) \tag{C.13}$$

$$d_{i+1} = (\delta x_{ij} + \delta x_{i+1j})/\delta x_{ij}\delta x_{i+1j} \tag{C.14}$$

$$d_{i+2} = -\delta x_{ij}/\delta x_{i+1j}(\delta x_{ij} + \delta x_{i+1j}) \tag{C.15}$$

For this case equation (C.8) gives

$$L^{(i+1)} = l_{i+1j-1}\, L_{i+1j-1} + l_{i+1j} L_{i+1j} + l_{i+1j+1}\, L_{i+1j+1} \tag{C.16}$$

and

$$L^{(i+2)} = l_{i+2j-1}\, L_{i+2j-1} + l_{i+2j} L_{i+2j} + l_{i+2j+1}\, L_{i+2j+1} \tag{C.17}$$

where the Lagrange coefficients are determined from equation C.9, C.10 and C.11.

Substitution of C.16 and C.17 into C.12 gives the results in the 7-point difference formulae

$$\begin{aligned} L_x &= d_i L_{ij} + d_{i+1} l_{i+1j-1}\, L_{i+1j-1} + d_{i+1} l_{i+1j} L_{i+1j} \\ &+ d_{i+1} l_{i+1j+1}\, L_{i+1j+1} + d_{i+2} l_{i+2j-1}\, L_{i+2j-1} + d_{i+2} l_{i+2j} L_{i+2j} \\ &+ d_{i+2} l_{i+2j+1}\, L_{i+2j+1}\,. \end{aligned} \tag{C.18}$$

Therefore the grid points which are required to evaluate L_x at this i boundary point are shown by black solid dots in Fig.C.7.

b $\quad j = n_y \quad (i = iapex)$

In this case the distribution of neighbouring points is more suited to the case of the central difference form for L_x (see Fig.C.8). Extrapolated values of L are therefore required and the extended grid line i-1 and i+1 at the positions indicated by crosses. Using a 3-point central difference scheme

$$L_x = d_{i-1}\, L^{(i-1)} + d_i L_{ij} + d_{i+1} L^{(i+1)} \tag{C.19}$$

where

$$\begin{aligned} d_{i-1} &= -\delta x_{ij} / \delta x_{i-1j} (\delta x_{ij} + \delta x_{i-1j}) && \text{(C.20)} \\ d_i &= (\delta x_{ij} - \delta x_{i-1j}) / \delta x_{ij} \delta x_{i-1j} && \text{(C.21)} \\ d_{i+1} &= \delta x_{i-1j} / \delta x_{ij} (\delta x_{ij} + \delta x_{i-1j}) && \text{(C.22)} \\ L^{(i-1)} &= l_{i-1j-2}\, L_{i-1j-2} + l_{i-1j-1}\, L_{i-1j-1} + l_{i-1j}\, L_{i-1j} && \text{(C.23)} \\ L^{(i+1)} &= l_{i+1j-2}\, L_{i+1j-2} + l_{i+1j-1}\, L_{i+1j-1} + l_{i+1j} L_{i+1j}\,. && \text{(C.24)} \end{aligned}$$

Substitution for $L^{(i-1)}$ and $L^{(i+1)}$ in C.19 gives

$$\begin{aligned} L_x &= d_{i-1}\, l_{i-1j-2}\; L_{i-1j-2} \;+ d_{i-1}\, l_{i-1j-1}\; L_{i-1j-1} \\ &+ d_{i-1}\, l_{i-1j}\, L_{i-1j} \;+ d_i L_{ij} + d_{i+1} l_{i+1j-2}\, L_{i+1j-2} \\ &+ d_{i+1} l_{i+1j-1}\; L_{i+1j-1} \;+ d_{i+1} l_{i+1j} L_{i+1j}\,. \end{aligned} \tag{C.25}$$

which is a 7-point formulae involving the grid points indicated in Fig.C.8.

c $\qquad 1 \le j \le n_y - 1 \qquad i \ge iapex$

A backward difference scheme is appropriate for i boundary points situated on this part of the boundary. Using interpolated value $L^{(i-2)}$ and $L^{(i-1)}$, the backward difference formulae can be written as

$$L_x = d_{i-2}\, L^{(i-2)} \;+ d_{i-1}\, L^{(i-1)} \;+ d_i L_{ij} \tag{C.26}$$

where

$$\begin{aligned} d_{i-2j} &= \delta x_{i-1j} \,/\delta x_{i-2j}\, (\delta x_{i-1j} \;+ \delta x_{i-2j}\,) && \text{(C.27)} \\ d_{i-1} &= -(\delta x_{i-1j} \;+ \delta x_{i-2j}\,)/\delta x_{i-1j}\, \delta x_{i-2j} && \text{(C.28)} \\ d_i &= 2(\delta x_{i-1j} \;+ \delta x_{i-2j}\,)/\delta x_{i-1j}\, (\delta x_{i-1j} \;+ \delta x_{i-2j}\,) && \text{(C.29)} \\ L^{(i-2)} &= l_{i-2j-1}\; L_{i-2j-1} \;+ l_{i-2j}\, L_{i-2j} \;+ l_{i-2j+1}\, L_{i-2j+1} && \text{(C.30)} \\ L^{(i-1)} &= l_{i-1j-1}\; L_{i-1j-1} \;+ l_{i-1j}\, L_{i-1j} \;+ l_{i-1j+1}\, L_{i-1j+1}\;. && \text{(C.31)} \end{aligned}$$

Hence for the seven grid points indicated in Fig.C.7, the finite difference formulae for L_x is

$$\begin{aligned} L_x &= d_{i-2}\, l_{i-2j-1}\; L_{i-2j-1} \;+ d_{i-2}\, l_{i-2j}\, L_{i-2j} \\ &+ d_{i-2}\, l_{i-2j+1}\; L_{i-2j+1} \;+ d_{i-1}\, l_{i-1j-1}\; L_{i-1j-1} \;+ d_{i-1}\, l_{i-1j}\, L_{i-1j} \\ &+ d_{i-1}\, l_{i-1j+1}\; L_{i-1j+1} \;+ d_i L_{ij} \end{aligned} \tag{C.32}$$

Appendix D

Matrix representation of the boundary-value problem

We refer to the linearised Monge-Ampère equation (4.26) and examine its finite difference structure at an internal point i,j for which $1 \leq i \leq n_x - 1$ and $0 \leq j \leq n_y - 1$. The coefficients $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_1$ and DEL may be calculated using the finite differences formulae and relevant equations from Section 4.2. If L^{k+1} is the column vector formed from the value of iterate L^{k+1} at the various grid points, then equation (4.26) may be written as follows

$$(\alpha_1 D_x + \alpha_2 D_{xx} + \beta_1 D_y + \beta_2 D_{yy} + \gamma_1 D_{xy} + \omega_1) L^{k+1} = DEL^k \tag{D.1}$$

where the row vectors D_x, D_{xx}, D_y, D_{yy} and D_{xy} are finite difference representation of the operators $\frac{\partial}{\partial x}, \frac{\partial^2}{\partial x \partial x}, \frac{\partial}{\partial y}, \frac{\partial^2}{\partial y \partial y}$ and $\frac{\partial^2}{\partial x \partial y}$. For example, equation (C.1) may be written in the form

$$\begin{aligned} D_x L^{k+1} = & \left[\frac{-\delta x_{ij}}{\delta x_{i-1j} (\delta x_{ij} + \delta x_{i-1j})}, \right. \\ & \left. \frac{\delta x_{ij} - \delta x_{i-1j}}{\delta x_{ij} \delta x_{i-1j}}, \frac{\delta x_{i-1j}}{\delta x_{ij} (\delta x_{ij} + \delta x_{i-1j})} \right] \begin{bmatrix} L_{i-1j} \\ L_{ij} \\ L_{i+1j} \end{bmatrix}^{k+1} . \end{aligned} \tag{D.2}$$

Thus, the mapping function value corresponding to each grid point i, j, forms an element of the column vector L. Those elements corresponding to internal points are arranged in ascending order of the grid numbering function ir(i,j) and are followed by the elements corresponding to i boundary points, arranged in ascending order of the separate numbering function $ir_b(i)$.

Using the finite difference model described above, differential operators may now be represented as row vectors. Operators corresponding to each of the derivatives required at any grid point may be found and combined to form a row vector representation of the synthesis equation applying at the point. Following this procedure for all grid points results in set of linear simultaneous equations which may be represented in the matrix form

$$O^k L^{k+1} = V^k \tag{D.3}$$

where k is the number of the iteration, O is a sparse matrix and V, is a column matrix. Solution of this equation, which is the matrix representation of the boundary-value problem, supplies the next iterate L^{k+1} of the mapping function at each grid point. The process is repeated until the iterative sequence $L^0, L^1, L^2, ..., L^k, ...$ converges to within a specified tolerance (for example $\epsilon = 5.10^{-5}$).

Appendix E

Computer programme

The solution of the Monge-Ampère equation under the boundary condition may be obtained by linearising the equations under the assumption of an approximate solution, applying a finite difference model to the linearised form and then using an iterative procedure to converge to the actual solution.The method has been programmed in Fortran and has been implemented on a " Sun SPARCcentre 1000 " computer. The programme requires the use of the NAG library for the solution of equation (D.3) and graphical software. The programme also has provision for generation of plotting data for subsequent processing and data files. When checking for convergence, the current iterate, L^k, and derived iterate, L^{k+1}, are compared at all grid points. The convergence criterion generally used requires that these iterates shall differ by less than 10^{-5} at each of these points. At the end of last iteration a small change in f (f=f+δf) may be needed to reduce the power deviation at the anchor point to an acceptable level. The computed mapping function is then used to generate dielectric and reflector surfaces. Finally the GO cross-polarisation in the aperture may also be calculated. The number of iterations required obviously depends on the boundary condition, and for a given geometry more specifically on the feed and aperture taper.

The following parameters are required as input to the programme.

γ	the angle of rotation axis of initial ellipsoid
ψ	elevation of the source cone axis
θ_c	semi-vertex angle of source cone
k	semi-latus rectum of ellipsoid
e	eccentricity of ellipsoid
N	dielectric refractive index
D	power density taper in dBs at edge of aperture
du	increment of power density parameter μ
df	power deviation adjustment at the anchor point
dx	grid increment along the x axis
dx	grid increment along the y axis
itmax	maximum number of iterations to be allowed for mapping function to converge at any stage

From these parameters the other relevant parameters are calculated by the programme. The computer programme listing which is more than 3.5K lines and 40 subroutines will follow.

```
C   ********************************************************
c            The synthesis of a single offset reflector
c                    with  dielectric cone feed
c
c     Parviz Sargolzaei,   supervisor Professor B. S. Westcott
c
c     Faculty of Mathematical Studies
c     University of Southampton
c     May 1996
c   *********************************************************
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      complex*16 z(0:4000)
      LOGICAL FIRST
      real time(2)
      COMMON/NGC/NG,NG1,NG2
      COMMON/PIVOT/FIRST
      COMMON/SIGC/SIG(0:4000)
      COMMON/init/sinit(0:4000)
      COMMON/UCOM/U
      COMMON/DUCOM/DU
      COMMON/IUCOM/IU
      COMMON/ILCOM/IL
      COMMON/CMZ/Z
      COMMON/PCOM/P
c     ---------------------------------
c       et=etime(time)
       Open(Unit=9,File='fincm.d3')
c      OPEN(UNIT=113,FILE='intial.d')
c      OPEN(UNIT=188,FILE='z.in')
c      OPEN(UNIT=189,FILE='fl.d')
C      ---------------------------------
c      t = dtime(time)
c      print *, 'CPU time =', time(1)
C     ----------------------------------
      FIRST=.TRUE.
                 CALL GRID
                 CALL INDIFF
                 CALL  BDIFF
                 CALL INTSOL
                 CALL RAWO
                 CALL CONEG
                 CALL TOTE
                 CALL AGBG
                 CALL TAPER
C                CALL GPD
C                CALL ANAPERG
C     ----------------------------------
       U=0.0D0
       DO 5 IL=0,IU
c     initial graph
       IF(IL.EQ.0)THEN
           CALL GRA
           CALL APERG
```

```
        ENDIF
           CALL ITERAT
        IF(IL.EQ.IU)THEN
C        final graph  g1
           CALL GRA
           CALL APERG
        ENDIF
        U=U+du
5       CONTINUE
C     ----------------------------------
        CALL TRANSMIT
        CALL CROSSP
        CALL CPAPERG
        CALL TRANSG
c     CLOSE(UNIT=15)
       et=etime(time)
       WRITE(9,*)'  user cpu time =',time(1),'  second'
       print *,'    user  cpu time is =',time(1),'  second'
c      DO 40 LI=0,NG
c      WRITE(188,*)Z(LI)
c     WRITE(189,*)'finial l=',sig(LI)
c40    continue
      STOP
      END
c     ************************************
      SUBROUTINE READG
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER  ITMAX
      LOGICAL FULL
      COMMON/INPT/GAMA,TETAC,PH,E,E0
      COMMON/FULLC/FULL
      COMMON/RKCOM/RK
      COMMON/RACOM/R,A,EN
      COMMON/CITMAX/ITMAX
      COMMON/CIANC/INANC
      COMMON/CMDEV/DEV
      COMMON/DLXY/DLX,DLY
      COMMON/IUCOM/IU
      COMMON/DUCOM/DU
      COMMON/DCOM/D
      COMMON/delc/delta
      COMMON/ITERATC/NIT
      COMMON/IMARKC/IMARK
C     --------------------------------
       OPEN(UNIT=5,FILE='input.dat')
       READ(5,*)delta
       READ(5,*)GAM
       READ(5,*)TETA
       READ(5,*)FI
       READ(5,*)RK
       READ(5,*)E
       READ(5,*)EN
       READ(5,*)DLX
```

```
        READ(5,*)DLY
        READ(5,*)A
        READ(5,*)ITMAX
        READ(5,*)INANC
        READ(5,*)DEV
        READ(5,*)IU
        READ(5,*)DU
        READ(5,*)D
        READ(5,*)NIT
        READ(5,*)IMARK
        CLOSE(UNIT=5)
C       ------------------
        FULL=.TRUE.
        WRITE(9,*)'    d in log(d)=',D
        WRITE(9,5)GAM,TETA,FI
        WRITE(9,6)RK,E,EN
        WRITE(9,7)A,DLX,DLY
        WRITE(9,8)IU,DU,NIT
        WRITE(9,9)ITMAX, DEV,D
        WRITE(9,*)'    ITMAX',ITMAX
        WRITE(9,*)'    DEV=',DEV
        WRITE(9,*)'    D IN LOG(D)=',D
5       FORMAT(5X,'GAMA=',F8.3,9X,'TETAC=',F8.3,5X,'PHI=',F8.3)
6       FORMAT(5X,'Rk  =',F12.4,7X,'  E=',F8.4,7X,'  N =',F8.4)
7       FORMAT(5X,'a=1+d=',f6.2,9x,' DX = ',F6.3,8X,'DY =',F6.3)
8       FORMAT(5X,'IU=',I2,17X,'du=',F6.3,12X,'no of iterations=',I3)
9       FORMAT(5X,'ITMAX=',I2,5X,'dev=',E14.7,5X,'d in log(d)=',f5.1)
C       ------------------
        PI=2*ASIN(1.0D0)
        GAMA=PI*GAM/180.
        TETAC=TETA*PI/180.
        PH=FI*PI/180.
        R=TAN(TETAC/2.)
        WRITE(9,10)R
10      FORMAT(5X,'RC   =',F12.8)
        RETURN
        END
C      *************************************
       SUBROUTINE DLXG(DLX,NX,XG,DXG)
       DOUBLE PRECISION DXG(0:70),XG(0:4000),DLX
       INTEGER NX
       DO 10 I=1,NX-1
       XG(I)=-FLOAT((NX/2-I))*DLX
10     CONTINUE
       DO 20 I=1,NX-2
       DXG(I)=DLX
20     CONTINUE
       RETURN
       END
c      **************************************
       SUBROUTINE GRID
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DOUBLE PRECISION XG(0:4000),YG(0:4000)
```

```
      DOUBLE PRECISION YB(70)
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000)
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      INTEGER INT(0:70,70)
      INTEGER NX,NY,IAPEX,IBMAX
      INTEGER NG,NG1,NG2
      LOGICAL FULL
      COMPLEX*16 Z(0:4000)
      COMPLEX*16 IM
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/DXYG/DXG,DYG,DX,DY
      COMMON/NGC/NG,NG1,NG2
      COMMON/FULLC/FULL
      COMMON/CMZ/Z
      COMMON/DLXY/DLX,DLY
      COMMON/RACOM/R,A,EN
      COMMON/NXCOM/NNX
      PARAMETER(ZERO=0.0D0,ONE=1.0D0,IM=(0.0D0,1.0D0))
C     --------------------------------------
      EPS=1.D-2
                CALL READG
      XG(0)=-R
      NDX=(R/DLX)
      DX2=R-FLOAT(NDX)*DLX
      IF(DX2.EQ.ZERO)THEN
       NX=2*NDX
       CALL DLXG(DLX,NX,XG,DXG)
       DXG(0)=DLX
       DXG(NX-1)=DLX
      ELSE
       NX=2*(NDX+1)
       DLX1=DX2
          CALL DLXG(DLX,NX,XG,DXG)
        DXG(0)=DLX1
        DXG(NX-1)=DLX1
      ENDIF
      DXG(NX)=ZERO
      XG(NX)=R
      YG(0)=ZERO
      NDY=(R/DLY)
      DY2=R-FLOAT(NDY)*DLY
      IF(DY2.EQ.ZERO)THEN
       NY=NDY
       DLY1=DLY
      ELSE
       NY=NDY+1
       DLY1=DY2
      ENDIF
      YG(NY)=R
      DYG(NY-1)=DLY1
      DO 20 J=0,NY-2
```

```
       YG(J+1)=YG(J)+DLY
       DYG(J)=DLY
20    CONTINUE
      DYG(NY)=ZERO
c          OPEN(UNIT=50,FILE='outg.d3')
c      WRITE(50,310)NX,NY,R,DLX ,DLY
c      WRITE(50,25)
25    FORMAT(//7X,'I',5X,'XG(I)',3X,'DXG(I)'/)
c      DO 30 I=0,NX
c      WRITE(50,50)I,XG(I),DXG(I)
c30    CONTINUE
c      WRITE(50,60)
c      DO 40 J=0,NY
c      WRITE(50,50)J,YG(J),DYG(J)
c40    CONTINUE
c50    FORMAT(5X,I3,4X,F6.2,2X,F6.2)
60    FORMAT(//7X,'J',5X,'YG(J)',3X,'DYG(J)'/)
       NNX=NX
       RSQ=R*R
       DO 70 J=0,NY
        DO 70 I=0,NX
         ASQ=XG(I)*XG(I)+YG(J)*YG(J)
         INT(I,J)=0
         IF((ONE+1.D-3)*RSQ-ASQ.GE.ZERO)INT(I,J)=1
         IF(J.EQ.NY)INT(I,J)=0
70     CONTINUE
       YMAX=ZERO
       DO 80 I=1,NX-1
        YB(I)=SQRT(RSQ-XG(I)*XG(I))
        IF(YB(I).GT.YMAX)THEN
        YMAX=YB(I)
        IAPEX=I
        ENDIF
        IBIND(I)=1
80     CONTINUE
       IMIN(0)=0
       IBIND(0)=0
       DO 90 I=0,NX-1
       IR(I,0)=I
       DX(I)=DXG(I)
90     CONTINUE
       IR(NX,0)=NX
       IMAX(0)=NX
       IBIND(NX)=0
       K=NX
       DO 120 J=1,NY-1
        XB=DSQRT(RSQ-YG(J)*YG(J))
        XBM=-XB
        XBP=XB
        DO 110 I=1,NX
         IF(INT(I,J).EQ.1)THEN
          K=K+1
      IF(XG(I)-XBM.LT.EPS*DXG(I-1).OR.YB(I)-YG(J).LT.
     + EPS*DYG(J-1))THEN
           IR(I,J)=K
           IMIN(J)=I
           DX(K)=DXG(I)
           IBIND(I)=0
          ELSE
           IMIN(J)=I-1
           IR(I-1,J)=K
           DX(K)=XG(I)-XBM
         END IF
         DO 100 I1=IMIN(J)+1,NX
          K=K+1
          IR(I1,J)=K
           IF(INT(I1,J).EQ.1)THEN
            IF(I1-IMIN(J).GT.1)DX(K-1)=DXG(I1-1)
           ELSE
      IF(XBP-XG(I1-1).LT.EPS*DXG(I1-1).OR.YB(I1-1)-YG(I1-1).LT.
     + EPS*DYG(J-1))THEN
            IMAX(J)=I1-1
            IBIND(I1-1)=0
            K=K-1
           ELSE
            IMAX(J)=I1
            DX(K-1)=XBP-XG(I1-1)
           END IF
            GO TO 120
           END IF
100      CONTINUE
        END IF
110   CONTINUE
120   CONTINUE
C   --------------------------------------------
      IBMAX=0
      DO 140 I=1,NX-1
       DO 130 J=1,NY
        IF(INT(I,J).EQ.0)THEN
         IF(IBIND(I).EQ.0)THEN
          JMAX(I)=J-1
         ELSE
          JMAX(I)=J
          IBIND(I)=1
          IBMAX=IBMAX+1
          IB(IBMAX)=I
          IRB(I)=K+IBMAX
          DY(IR(I,J-1))=YB(I)-YG(J-1)
         END IF
         GOTO 140
        ELSE
        DY(IR(I,J-1))=DYG(J-1)
        END IF
130    CONTINUE
       WRITE(6,*)'NO END POINT ON I LINE',I
       STOP
```

```
140   CONTINUE
      NG1=K+IBMAX
C  -------------------------
      DO 150 J=0,NY-1
      DO 150 I=IMIN(J),IMAX(J)
       IF(I.EQ.IMIN(J))THEN
        Z(IR(I,J))=XG(I+1)+IM*YG(J)-DX(IR(I,J))
       ELSE
        IF(I.EQ.IMAX(J))THEN
         Z(IR(I,J))=XG(I-1)+IM*YG(J)+DX(IR(I-1,J))
        ELSE
         Z(IR(I,J))=XG(I)+IM*YG(J)
        ENDIF
       ENDIF
150      CONTINUE
      DO 160 ID=1,IBMAX
        I=IB(ID)
        J=JMAX(I)
        Z(IRB(I))=XG(I)+IM*YG(J-1)+IM*DY(IR(I,J-1))
160   CONTINUE
c      WRITE(50,330)
      DO 170 I=1,NX-1
       IF(IBIND(I).EQ.0)INUM=IR(I,JMAX(I))
        IF(IBIND(I).EQ.1)INUM=IRB(I)
c          WRITE(50,320)I,JMAX(I),IBIND(I),INUM
IF(JMAX(I).LT.2)THEN
WRITE(6,99)
WRITE(6,*)'There are less than 3 grids points on I line',I
STOP
ENDIF
 170   CONTINUE
c       WRITE(50,340)
       DO 180 J=0,NY-1
c       WRITE(50,350)J,IMIN(J),IMAX(J),IR(IMIN(J),J),IR(IMAX(J),J)
       IF(IMAX(J)-IMIN(J).LT.2)THEN
WRITE(6,99)
WRITE(6,*)'Tere are less than 3 grid points on J line',J
STOP
       ENDIF
 180    CONTINUE
c       WRITE(50,360)
       DO 190 J=0,NY-1
         DO 190 I=IMIN(J),IMAX(J)
c           WRITE(50,370)I,J,IR(I,J)
 190   CONTINUE
 99    FORMAT(/'warning from subroutine GRID')
C      I   BOUNDARY POINTS
c       WRITE(50,380)
       DO 200 ID=1,IBMAX
       I=IB(ID)
       J=JMAX(I)
c       WRITE(50,370)I,J,IRB(I)
 200   CONTINUE
```

```
      K=NG1
C                        ****  NEGATIVE HALF OF GRID  ****
      DO 210 J=1,NY-1
       DO 210 I=IMIN(J),IMAX(J)
        K=K+1
        IR(I,-J)=K
        Z(K)=CONJG(Z(IR(I,J)))
 210  CONTINUE
      DO 220 ID=1,IBMAX
       I=IB(ID)
       K=K+1
       IRBM(I)=K
       Z(K)=CONJG(Z(IRB(I)))
 220  CONTINUE
      NG2=K
WRITE(9,*)'NX=',nx,'    NY=',ny
WRITE(9,*)'number of elements are ',ng2
c      CLOSE(UNIT=15)
       NG=NG1
       IF(FULL)NG=NG2
IF(NG2.GT.4000)THEN
WRITE(6,99)
WRITE(6,*)'Too many grid points ,increase array size'
STOP
ENDIF
c      WRITE(50,390)
      DO 250 J=1,NY-1
        DO 250 I=IMIN(J),IMAX(J)
c         WRITE(50,370)I,-J,IR(I,-J)
250   CONTINUE
      DO 260 ID=1,IBMAX
       I=IB(ID)
       J=-JMAX(I)
c       WRITE(50,370)I,J,IRBM(I)
260   CONTINUE
c      WRITE(50,400)
      DO 270 J=0,NY-1
       DO 270 I=IMIN(J),IMAX(J)
        II=IR(I,J)
c        WRITE(50,420)II,J,DX(II),DY(II),Z(II)
 270   CONTINUE
      DO 280 ID=1,IBMAX
       I=IB(ID)
       J=JMAX(I)
       K=IRB(I)
c       WRITE(50,420)K,J,DX(K),DY(K),Z(K)
 280   CONTINUE
      DO 290 J=1,NY-1
       DO 290 I=IMIN(J),IMAX(J)
        IN=IR(I,-J)
        DX(IN)=DX(IR(I,J))
        DY(IN)=DY(IR(I,J))
c       WRITE(50,420)IN,-J,DX(IN),DY(IN),Z(IN)
```

```
290   CONTINUE
      DO 300 ID=1,IBMAX
       I=IB(ID)
       J=-JMAX(I)
       K=IRBM(I)
c      WRITE(50,420)K,J,DX(K),DY(K),Z(K)
300   CONTINUE
c       CLOSE(UNIT=50)
C_---------------------------------------------------------------------
310    FORMAT(//5X,'NX=',I2,5x,'NY=',I2,5x,'R=',F6.2,5X,'DX=',F6.2,5X,'
     +DY=',F6.2)
320   FORMAT(2X,I3,I8,10X,I4,10X,I4)
330   FORMAT(///5X,'BONDARY VALUES FOR I BONDARY POINTS'/
     +5X,'IF IBIND(I)=0 THE POINT I ALSO A J BONDARY POINT'//
     +1X,'   I    JMAX(I)      IBIND(I)      IR(I,JMAX(I))'/)
340    FORMAT(/5X,'BOUNDARY VALUE FOR J BOUNDRAY POINTS '//1X,
     +'   J    IMIN(J) IMAX(J)  IR(IMIN(J),J)   IR(IMAX(J),J)'/)
350    FORMAT(1X,I4,4X,I4,5X,I4,5X,I6,10X,I6)
360   FORMAT(/5X,'GRID NUMBERING FUNCTION',/5X,'POSITIVE HALF OF GRID'/
     +5X,'INTERNAL AND J BOUNDARY POINTS'//1X,'   I     J     IR(I,J)'/)
370   FORMAT(1X,I4,2X,I4,I9)
380   FORMAT(/5X,'I  BONDARY POINTS'//1X,'   I     J     IRB(I)'/)
390   FORMAT(/5X,'GRID NUMBERING FUNCTION',/5X,'NEGATIVE HALF OF GRID'/
     +5X,'INTERNAL AND J BOUNDARY POINTS'//1X,'   I     J     IR(I,J)'/)
400   FORMAT(//8X,'K',5X,'J',8X,'DX(K)',8X,'DY(K)',6X,'X(K)',7X,'Y(K)'/)
420     FORMAT(5X,I4,2X,I4,6X,F8.5,5X,F8.5,2F10.4)
      RETURN
      END
C
C  ********************************
      SUBROUTINE INDIFF
C  Finds grid point numbering and operator coefficients for finite
C  difference approximation to first & second derivatives at INTERNAL
C  grid points.
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION OX(3,0:4000),OY(3,0:4000),OXX(3,0:4000)
      DOUBLE PRECISION OYY(3,0:4000),OXY(9,0:4000)
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000)
      INTEGER IMIN(0:70),IMAX(0:70),JMAX(70),IR(0:70,-40:40)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      INTEGER NX,NY,IAPEX,IBMAX
      INTEGER NG,NG1,NG2
      INTEGER IRX(3,0:4000),IRY(3,0:4000),IRXY(9,0:4000)
      LOGICAL FULL
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/DXYG/DXG,DYG,DX,DY
      COMMON/NGC/NG,NG1,NG2
      COMMON/FULLC/FULL
      COMMON/INDOP/OX,OXX,OY,OYY,OXY,IRX,IRY,IRXY
      PARAMETER(ZERO=0.0D0,ONE=1.0D0,TWO=2.0D0)
C      --------------------------------
      DO 10 J=0,NY-1
      DO 10 I=IMIN(J)+1,IMAX(J)-1
       IF(JMAX(I).EQ.J) GOTO 10
       ID=IR(I,J)
C  Determine the 3 points IRX(1-3) to be used in the central
C  difference approximations to SIGx and SIGxx at point I,J. Set
C  the corresponding finite difference formula coefficients OX(1-3)
C  and OXX(1-3)
       IRX(1,ID)=IR(I-1,J)
       IRX(2,ID)=IR(I,J)
       IRX(3,ID)=IR(I+1,J)
       DX1=DX(IRX(1,ID))
       DX2=DX(IRX(2,ID))
       OX(1,ID)=-DX2/(DX1*(DX2+DX1))
       OX(2,ID)=(DX2-DX1)/(DX2*DX1)
       OX(3,ID)=DX1/(DX2*(DX2+DX1))
       OXX(1,ID)=TWO/(DX1*(DX2+DX1))
       OXX(2,ID)=-TWO/(DX2*DX1)
       OXX(3,ID)=TWO/(DX2*(DX2+DX1))
C  Special case of points on the line J=0.  Determine the 3 points
C  IRY(1-3) to be used in the central difference approximations to
C  SIGy and SIGyy at point (I,0).  Set the corresponding finite
C  difference formula coefficients OY(1-3) and OYY(1-3)
       IF(J.EQ.0) THEN
        IRY(1,ID)=IR(I,-1)
        IRY(2,ID)=IR(I,0)
        IRY(3,ID)=IR(I,1)
        DY1=DY(IR(I,0))
       IF(.NOT.FULL)THEN
         OY(1,ID)=ZERO
         OY(2,ID)=ZERO
         OY(3,ID)=ZERO
         OYY(1,ID)=ZERO
         OYY(3,ID)=TWO/(DY1*DY1)
         OYY(2,ID)=-OYY(3,ID)
        ELSE
c       if full
         OY(1,ID)=-ONE/(TWO*DY1)
         OY(2,ID)=ZERO
         OY(3,ID)=-OY(1,ID)
         OYY(1,ID)=ONE/(DY1*DY1)
         OYY(2,ID)=-TWO*OYY(1,ID)
         OYY(3,ID)=OYY(1,ID)
        ENDIF
       ELSE
C  If J is not zero find SIGy and SIGyy by a similar procedure to that
C  for SIGx and SIGxx but note that point IRY(3) may be an I boundary
C  point and hence numbered by IRB(I) rather than IR(I,J)
        IRY(1,ID)=IR(I,J-1)
        IRY(2,ID)=IR(I,J)
        IF(JMAX(I).EQ.J+1.AND.IBIND(I).EQ.1) THEN
         IRY(3,ID)=IRB(I)
        ELSE
         IRY(3,ID)=IR(I,J+1)
        END IF
        DY1=DY(IRY(1,ID))
        DY2=DY(IRY(2,ID))
        OY(1,ID)=-DY2/(DY1*(DY2+DY1))
        OY(2,ID)=(DY2-DY1)/(DY2*DY1)
        OY(3,ID)=DY1/(DY2*(DY2+DY1))
        OYY(1,ID)=TWO/(DY1*(DY2+DY1))
        OYY(2,ID)=-TWO/(DY2*DY1)
        OYY(3,ID)=TWO/(DY2*(DY2+DY1))
       END IF
 10    CONTINUE
C   Use symmetry for points on the negative half of the grid
C    -----------------------------------------------------
       IF(FULL)THEN
       NEG=NG2-NG1
       DO 11 JJ=1,NY-1
       J=-JJ
       DO 11 I=IMIN(JJ)+1,IMAX(JJ)-1
       IF(JMAX(I).EQ.JJ) GO TO 11
       NM=IR(I,J)
       NP=IR(I,JJ)
        DO 12 L=1,3
         IRX(L,NM)=IRX(L,NP)+NEG
         OX(L,NM)=OX(L,NP)
         OXX(L,NM)=OXX(L,NP)
           IF(IRY(L,NP).LE.NX)THEN
            IRY(L,NM)=IRY(L,NP)
           ELSE
            IRY(L,NM)=IRY(L,NP)+NEG
           ENDIF
         OY(L,NM)=-OY(L,NP)
         OYY(L,NM)=OYY(L,NP)
 12      CONTINUE
 11     CONTINUE
        ENDIF
                     CALL CRSDIF
C  Determine expressions for the cross-derivative at each point
C  on positive half of aperture
60     FORMAT(6X,' K    ID      IRXY(K,ID)         OXY(K,ID)')
      DO 101 J=0,NY-1
      DO 101 I=IMIN(J)+1,IMAX(J)-1
       IF(JMAX(I).EQ.J)GO TO 101
       ID=IR(I,J)
       DO 90 K=1,9
90     CONTINUE
101    CONTINUE
       IF(FULL)THEN
C  Use symmetry for points on the negative half of the grid
       DO 14 JJ=1,NY-1
       J=-JJ
       DO 14 I=IMIN(JJ)+1,IMAX(JJ)-1
       IF(JMAX(I).EQ.JJ) GO TO 14
       NM=IR(I,J)
       NP=IR(I,JJ)
       DO 13 L=1,9
       IF(IRXY(L,NP).LE.NX)THEN
        IRXY(L,NM)=IRXY(L,NP)
       ELSE
        IRXY(L,NM)=IRXY(L,NP)+NEG
       ENDIF
       OXY(L,NM)=-OXY(L,NP)
 13    CONTINUE
 14    CONTINUE
       ENDIF
      DO 37 J=0,NY-1
      DO 37 I=IMIN(J)+1,IMAX(J)-1
      IF(JMAX(I).EQ.J)GO TO 37
       ID=IR(I,J)
 37    CONTINUE
      IF(FULL)THEN
      DO 39 J=1,NY-1
      DO 39 I=IMIN(J)+1,IMAX(J)-1
      IF(JMAX(I).EQ.J)GO TO 39
       JJ=-J
       ID=IR(I,JJ)
 39    CONTINUE
       ENDIF
      DO 83 J=0,NY-1
      DO 83 I=IMIN(J)+1,IMAX(J)-1
      IF(JMAX(I).EQ.J)GO TO 83
      ID=IR(I,J)
83    CONTINUE
      DO 73 J=0,NY-1
      DO 73 I=IMIN(J)+1,IMAX(J)-1
      IF(JMAX(I).EQ.J)GO TO 73
      ID=IR(I,J)
73    CONTINUE
      DO 63 J=0,NY-1
      DO 63 I=IMIN(J)+1,IMAX(J)-1
      IF(JMAX(I).EQ.J)GO TO 63
      ID=IR(I,J)
63    CONTINUE
      DO 43 J=0,NY-1
      DO 43 I= IMIN(J)+1,IMAX(J)-1
      IF(JMAX(I).EQ.J)GO TO 43
      ID=IR(I,J)
43    CONTINUE
       DO 35 JJ=1,NY
        J=-JJ
        DO 35 I=IMIN(JJ)+1,IMAX(JJ)-1
        IF(JMAX(I).EQ.JJ)GO TO 35
         NM=IR(I,J)
35     CONTINUE
       DO 135 JJ=1,NY-1
        J=-JJ
        DO 135 I=IMIN(JJ)+1,IMAX(JJ)-1
```

```
         IF(JMAX(I).EQ.JJ)GO TO 135
          NM=IR(I,J)
135     CONTINUE
       DO 235 JJ=1,NY-1
        J=-JJ
        DO 235 I=IMIN(JJ)+1,IMAX(JJ)-1
         IF(JMAX(I).EQ.JJ)GO TO 235
          NM=IR(I,J)
235     CONTINUE
       DO 335 JJ=1,NY-1
        J=-JJ
        DO 335 I=IMIN(JJ)+1,IMAX(JJ)-1
         IF(JMAX(I).EQ.JJ)GO TO 335
          NM=IR(I,J)
335   CONTINUE
      RETURN
      END
C  *****************************************
      SUBROUTINE CRSDIF
C  Finds  grid point numbering and operator coefficients for finite
C  difference approximation to cross derivative wrt. x & y
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION OX(3,0:4000),OY(3,0:4000),OXX(3,0:4000)
      DOUBLE PRECISION OYY(3,0:4000),OXY(9,0:4000)
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000)
      DOUBLE PRECISION BOX(3),BOY(3)
      INTEGER NX,NY,IAPEX,IBMAX
      INTEGER IRX(3,0:4000),IRY(3,0:4000),IRXY(9,0:4000)
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      LOGICAL FULL
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/FULLC/FULL
      COMMON/DXYG/DXG,DYG,DX,DY
      COMMON/INDOP/OX,OXX,OY,OYY,OXY,IRX,IRY,IRXY
      PARAMETER(TWO=2.0D0)
C       --------------------------------
      DO 100 J=0,NY-1
      DO 100 I=IMIN(J)+1,IMAX(J)-1
      ID=IR(I,J)
      K=0
      IF(JMAX(I).GT.J+1) THEN
C  Use centred differences wrt. Y and centred differences wrt. X
C       --------------------------------
      DO 200 L=1,3
      IJD=IR(I,J-2+L)
      DO 200 M=1,3
      K=K+1
      IRXY(K,ID)=IRX(M,IJD)
      OXY(K,ID)=OX(M,IJD)*OY(L,ID)
200   CONTINUE
      GOTO 100
      ELSE IF(JMAX(I).EQ.J+1 .AND. J.GT.1) THEN
C  Use backward differences wrt. Y and centred differences wrt. X
      DY1=DY(IR(I,J-2))
      DY2=DY(IR(I,J-1))
      BOY(1)=DY2/(DY1*(DY2+DY1))
      BOY(2)=-(DY2+DY1)/(DY2*DY1)
      BOY(3)=(TWO*DY2+DY1)/(DY2*(DY2+DY1))
      DO 300 L=1,3
      IJD=IR(I,J-3+L)
      DO 300 M=1,3
      K=K+1
      IRXY(K,ID)=IRX(M,IJD)
      OXY(K,ID)=OX(M,IJD)*BOY(L)
300   CONTINUE
GOTO 100
      ELSE IF(JMAX(I).EQ.J+1 .AND. J.EQ.1) THEN
      IF(I.LT.IAPEX) THEN
C  Use forward differences wrt. X and centred differences wrt. Y
      DX1=DX(IR(I,J))
      DX2=DX(IR(I+1,J))
      BOX(1)=-(DX2+TWO*DX1)/(DX1*(DX1+DX2))
      BOX(2)=(DX1+DX2)/(DX1*DX2)
      BOX(3)=-DX1/(DX2*(DX1+DX2))
      DO 400 M=1,3
      IJD=IR(I-1+M,1)
      DO 400 L=1,3
      K=K+1
      IRXY(K,ID)=IRY(L,IJD)
      OXY(K,ID)=OY(L,IJD)*BOX(M)
400   CONTINUE
      GOTO 100
      ELSE IF(I.GE.IAPEX) THEN
C  Use backward differences wrt. X and centred differences wrt. Y
      DX1=DX(IR(I-2,J))
      DX2=DX(IR(I-1,J))
      BOX(1)=DX2/(DX1*(DX2+DX1))
      BOX(2)=-(DX2+DX1)/(DX2*DX1)
      BOX(3)=(DX1+TWO*DX2)/(DX2*(DX2+DX1))
      DO 500 M=1,3
      IJD=IR(I-3+M,1)
      DO 500 L=1,3
      K=K+1
      IRXY(K,ID)=IRY(L,IJD)
      OXY(K,ID)=OY(L,IJD)*BOX(M)
500   CONTINUE
      END IF
      END IF
100   CONTINUE
      RETURN
      END
C    ******************************************
      SUBROUTINE BDIFF
C  Finds index numbers of grid points, and the corresponding operator
```

```
C  coefficients, used to evaluate finite difference approximations to
C  first derivatives wrt. X & Y at boundary points
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION OX(7,300),OY(7,300),OOY(7),OOX(7)
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000)
      INTEGER IRX(7,300),IRY(7,300)
      INTEGER ISUMX(300),ISUMY(300),IBOUND(300)
      INTEGER IMIN(0:70),IMAX(0:70),IRBM(0:70),IIRX(7),IIRY(7)
      INTEGER JMAX(70),IB(0:70),IRB(0:70),IBIND(0:70),IR(0:70,-40:40)
      INTEGER NX,NY,IAPEX,IBMAX
      INTEGER NG,NG1,NG2,NB1,NB2
      LOGICAL FULL
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/DXYG/DXG,DYG,DX,DY
      COMMON /NGC/NG,NG1,NG2
      COMMON /BDOP/ OX,OY,IRX,IRY,ISUMX,ISUMY,IBOUND,NB1,NB2
      COMMON /FULLC/FULL
      PARAMETER(TWO=2.0D0)
C     --------------------------------
      NB1=IBMAX+2*NY
      NB2=2*NB1-2
C  J boundary points
      DO 200 J=0,NY-1
C  J boundary point at I=IMIN(J)
      I=IMIN(J)
      IRIJ=IR(I,J)
      ID=J*2+1
      IBOUND(ID)=IRIJ
      ISUMX(ID)=3
      IF(J.EQ.0 .AND. .NOT.FULL) THEN
      ISUMY(ID)=0
      ELSE
      ISUMY(ID)=7
C  Seven point finite differences are used for derivative wrt. Y
      CALL FDYMIN(I,J,IIRY,OOY)
      DO 210 L=1,7
      IRY(L,ID)=IIRY(L)
      OY(L,ID)=OOY(L)
210   CONTINUE
      END IF
C  A 3 point forward difference formula may be used for X-derivative
      IRX(1,ID)=IRIJ
      IRX(2,ID)=IR(I+1,J)
      IRX(3,ID)=IR(I+2,J)
      DX1=DX(IRX(1,ID))
      DX2=DX(IRX(2,ID))
      OX(1,ID)=-(DX2+TWO*DX1)/(DX1*(DX1+DX2))
      OX(2,ID)=(DX1+DX2)/(DX1*DX2)
      OX(3,ID)=-DX1/(DX2*(DX1+DX2))
C  J boundary point at I=IMAX(J)
      I=IMAX(J)
      IRIJ=IR(I,J)
      ID=J*2+2
      IBOUND(ID)=IRIJ
      ISUMX(ID)=3
      IF(J.EQ.0 .AND. .NOT.FULL) THEN
      ISUMY(ID)=0
      ELSE
      ISUMY(ID)=7
C  Seven point finite differences are used for derivative wrt. Y
      CALL FDYMAX(I,J,IIRY,OOY)
      DO 220 L=1,7
      IRY(L,ID)=IIRY(L)
      OY(L,ID)=OOY(L)
220   CONTINUE
      END IF
C  A 3 point backward difference formula may be used for X-derivative
      IRX(1,ID)=IR(I-2,J)
      IRX(2,ID)=IR(I-1,J)
      IRX(3,ID)=IRIJ
      DX1=DX(IRX(1,ID))
      DX2=DX(IRX(2,ID))
      OX(1,ID)=DX2/(DX1*(DX2+DX1))
      OX(2,ID)=-(DX2+DX1)/(DX2*DX1)
      OX(3,ID)=(DX1+TWO*DX2)/(DX2*(DX2+DX1))
200   CONTINUE
C  I boundary points
      DO 100 IBD=1,IBMAX
      ID=2*NY+IBD
      I=IB(IBD)
      J=JMAX(I)
      IRBI=IRB(I)
      IBOUND(ID)=IRBI
C  Seven point finite differences are used for derivatives wrt X
      CALL FDXMAX(I,J,IIRX,OOX)
      ISUMX(ID)=7
      DO 110 L=1,7
      IRX(L,ID)=IIRX(L)
      OX(L,ID)=OOX(L)
110   CONTINUE
C  A 3 point backward difference formula may be used for Y-derivative
      IRY(1,ID)=IR(I,J-2)
      IRY(2,ID)=IR(I,J-1)
      IRY(3,ID)=IRBI
      DY1=DY(IRY(1,ID))
      DY2=DY(IRY(2,ID))
      OY(1,ID)=DY2/(DY1*(DY2+DY1))
      OY(2,ID)=-(DY2+DY1)/(DY2*DY1)
      OY(3,ID)=(TWO*DY2+DY1)/(DY2*(DY2+DY1))
      ISUMY(ID)=3
100   CONTINUE
C  Use symmetry for derivatives at points on negative half of grid
         IF(FULL)THEN
      NEG=NG2-NG1
      DO 300 N=1,NB1-2
      NM=NB1+N
      NP=2+N
      ISUMX(NM)=ISUMX(NP)
      IBOUND(NM)=IBOUND(NP)+NEG
      DO 400 L=1,ISUMX(NP)
      IRXLNP=IRX(L,NP)
      IF(IRXLNP.LE.NX) THEN
       IRX(L,NM)=IRXLNP
      ELSE
       IRX(L,NM)=IRXLNP+NEG
      END IF
      OX(L,NM)=OX(L,NP)
400   CONTINUE
      ISUMY(NM)=ISUMY(NP)
      DO 500 L=1,ISUMY(NP)
      IRYLNP=IRY(L,NP)
      IF(IRYLNP.LE.NX) THEN
       IRY(L,NM)=IRYLNP
       ELSE IF(IRYLNP.LE.NG1) THEN
        IRY(L,NM)=IRYLNP+NEG
       ELSE
        IRY(L,NM)=IRYLNP-NEG
       END IF
      OY(L,NM)=-OY(L,NP)
500   CONTINUE
300   CONTINUE
      ENDIF
      RETURN
      END
C   ********************************
      SUBROUTINE FDXMAX(I,J,IRX,OX)
C   Finds grid points and corresponding operator coefficients for
C   7 point finite difference formula for derivatives wrt X
C   at I boundary point I,J
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION L1,L2,L3
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000),OX(7)
      INTEGER IMIN(0:70),IMAX(0:70),JMAX(70),IR(0:70,-40:40),IRX(7)
      INTEGER IB(0:70),IRB(0:70),IRBM(0:70),IBIND(0:70)
      INTEGER NX,NY,IAPEX,IBMAX
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/DXYG/DXG,DYG,DX,DY
      PARAMETER(ZERO=0.0D0,ONE=1.0D0,TWO=2.0D0,THREE=3.0D0)
C      -----------------------------
      IM1=I-1
      IP1=I+1
      JM1=J-1
C   For I < IAPEX and J < NY-1, a 7 point interpolation forward
C   difference is used. This involves point I,J and 3 points each
C   from rows I+1 and I+2.
      IF(I.LT.IAPEX) THEN
C   Determine coefficients D1,D2,D3 of the forward difference w.r.t x
```

```
       D1=-THREE/(TWO*DXG(I))
       D2=TWO/DXG(I)
       D3=-ONE/(TWO*DXG(I))
       OX(1)=D1
       IRX(1)=IRB(I)
C  Determine the 3 interpolation points to be used on rows I+1 and I+2.
C  The grid point numbers are stored in IRX(1) to IRX(7).
      INC=0
      D=D2
      DO 100 IL=IP1,I+2
      IF(JMAX(IL).GE.J+1) THEN
       IRX(2+INC)=IR(IL,JM1)
       IRX(3+INC)=IR(IL,J)
       IF(JMAX(IL).EQ.J+1.AND.IBIND(IL).EQ.1) THEN
        IRX(4+INC)=IRB(IL)
       ELSE
        IRX(4+INC)=IR(IL,J+1)
       END IF
      ELSE IF(JMAX(IL).EQ.J) THEN
       IRX(2+INC)=IR(IL,J-2)
       IRX(3+INC)=IR(IL,JM1)
       IF(IBIND(IL).EQ.1) THEN
       IRX(4+INC)=IRB(IL)
      ELSE
       IRX(4+INC)=IR(IL,J)
      END IF
      END IF
C  Find the Lagrange interpolation coefficients L1,L2,L3 and hence the
C  operator coefficients for the interpolation difference.
      CALL LAG(IRX(1),IRX(2+INC),IRX(3+INC),IRX(4+INC),L1,L2,L3,1)
      OX(2+INC)=D*L1
      OX(3+INC)=D*L2
      OX(4+INC)=D*L3
      INC=3
      D=D3
100   CONTINUE
C  For J=NY a 7 point extrapolation central difference is
C  used. This involves point I,J and 3 points each from rows I-1 & I+1
      ELSE IF(I.EQ.IAPEX) THEN
C  Determine coefficients D1,D2,D3 for central differences wrt. X
       D1=-ONE/(TWO*DXG(I))
       D2=ZERO
       D3=-D1
       OX(4)=D2
       IRX(4)=IRB(I)
C  Determine interpolation points to be used on rows I-1 and I-2
      INC=0
      D=D1
      DO 200 IL=IM1,IP1,2
      IRX(1+INC)=IR(IL,JMAX(IL)-2)
      IRX(2+INC)=IR(IL,JMAX(IL)-1)
      IF(IBIND(IL).EQ.1) THEN
      IRX(3+INC)=IRB(IL)
      ELSE
      IRX(3+INC)=IR(IL,JMAX(IL))
      END IF
C  Find the Lagrange interpolation coefficients L1,L2,L3
      CALL LAG(IRX(4),IRX(1+INC),IRX(2+INC),IRX(3+INC),L1,L2,L3,1)
      OX(1+INC)=D*L1
      OX(2+INC)=D*L2
      OX(3+INC)=D*L3
      INC=4
      D=D3
200   CONTINUE
C  For I > IAPEX and J < NY-1, a 7 point interpolation backward
C  difference is used. This involves point I,J and 3 points each
C  from rows I-1 and I-2.
      ELSE IF(I.GT.IAPEX) THEN
C  Determine coefficients D1,D2,D3 for backward differences wrt. X
       D1=ONE/(TWO*DXG(IM1))
       D2=-TWO/DXG(IM1)
       D3=THREE/(TWO*DXG(IM1))
      OX(7)=D3
      IRX(7)=IRB(I)
C Determine interpolation points to be used on rows I-1 and I-2
      INC=0
      D=D1
      DO 300 IL=I-2,IM1
      IF(JMAX(IL).GE.J+1) THEN
      IRX(1+INC)=IR(IL,JM1)
      IRX(2+INC)=IR(IL,J)
      IF(JMAX(IL).EQ.J+1.AND.IBIND(IL).EQ.1) THEN
      IRX(3+INC)=IRB(IL)
      ELSE
      IRX(3+INC)=IR(IL,J+1)
      END IF
      ELSE IF(JMAX(IL).EQ.J) THEN
      IRX(1+INC)=IR(IL,J-2)
      IRX(2+INC)=IR(IL,JM1)
      IF(IBIND(IL).EQ.1) THEN
      IRX(3+INC)=IRB(IL)
      ELSE
      IRX(3+INC)=IR(IL,J)
      END IF
      END IF
C  Find the Lagrange interpolation coefficients L1,L2,L3
      CALL LAG(IRX(7),IRX(1+INC),IRX(2+INC),IRX(3+INC),L1,L2,L3,1)
      OX(1+INC)=D*L1
      OX(2+INC)=D*L2
      OX(3+INC)=D*L3
      INC=3
      D=D2
300   CONTINUE
      END IF
      END
C  ********************************
```

```
      SUBROUTINE FDYMAX(I,J,IRY,OY)
C  Finds grid points and corresponding operator coefficients for
C  7 point finite difference formula for derivatives wrt Y
C  at the J boundary point I,J   (where I = IMAX(J))
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION L1,L2,L3
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000),OY(7)
      INTEGER IMIN(0:70),IMAX(0:70),JMAX(70),IR(0:70,-40:40),IRY(7)
      INTEGER  IB(0:70),IRB(0:70),IRBM(0:70),IBIND(0:70)
      INTEGER NX,NY,IAPEX,IBMAX
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/DXYG/DXG,DYG,DX,DY
      PARAMETER(ONE=1.0D0,TWO=2.0D0,THREE=3.0D0)
C     -------------------------------
      JM1=J-1
C  For IMAX(J)=NX a 7 point interpolation/extrapolation central
C  difference is used . This involves point I,J and 3 points each
C  from rows J-1 and J+1
      IF( J.LE.1) THEN
      IF(J.EQ.0) THEN
      D=1.0D0/(TWO*DYG(0))
      IRY(4)=IR(NX,0)
      OY(4)=0.0D0
      IRY(1)=IR(NX-2,-1)
      IRY(2)=IR(NX-1,-1)
      IRY(3)=IR(NX,-1)
      IRY(5)=IR(NX-2,1)
      IRY(6)=IR(NX-1,1)
      IRY(7)=IR(NX,1)
C  Find the Lagrange interpolation coefficients L1,L2,L3
      CALL LAG(IRY(4),IRY(5),IRY(6),IRY(7),L1,L2,L3,0)
      OY(5)=D*L1
      OY(6)=D*L2
      OY(7)=D*L3
      OY(1)=-OY(5)
      OY(2)=-OY(6)
      OY(3)=-OY(7)
      ELSE
C  Determine coefficients D1,D2,D3 for central differences wrt Y
      D1=-DYG(J)/(DYG(JM1)*(DYG(J)+DYG(JM1)))
      D2=(DYG(J)-DYG(JM1))/(DYG(J)*DYG(JM1))
      D3=DYG(JM1)/(DYG(J)*(DYG(J)+DYG(JM1)))
      OY(4)=D2
      IRY(4)=IR(I,J)
C  Determine interpolation points to be used on rows J+1 and J-1
      INC=0
      D=D1
      DO 100 JL=JM1,J+1,2
      IRY(1+INC)=IR(IMAX(JL)-2,JL)
      IRY(2+INC)=IR(IMAX(JL)-1,JL)
      IRY(3+INC)=IR(IMAX(JL),JL)
C  Find the Lagrange interpolation coefficients L1,L2,L3
```

```
      CALL LAG(IRY(4),IRY(1+INC),IRY(2+INC),IRY(3+INC),L1,L2,L3,0)
      OY(1+INC)=D*L1
      OY(2+INC)=D*L2
      OY(3+INC)=D*L3
      D=D3
      INC=4
100   CONTINUE
      END IF
C  For IMAX(J) < NX a 7 point interpolation backward difference is used
C  This involves point I,J and three points each from rows J-2 and J-1
      ELSE IF(J.GT.1) THEN
C  Determine coefficients D1,D2,D3 for backward differences wrt Y
       D1=ONE/(TWO*DYG(JM1))
       D2=-TWO/DYG(JM1)
       D3=THREE/(TWO*DYG(JM1))
      OY(7)=D3
      IRY(7)=IR(I,J)
C  Determine interpolation points to be used on rows J-2 and J-1
      INC=0
      D=D1
      DO 200 JL=J-2,JM1
      IF(IMAX(ABS(JL)).GE.I+1) THEN
         IRY(1+INC)=IR(I-1,JL)
         IRY(2+INC)=IR(I,JL)
         IRY(3+INC)=IR(I+1,JL)
        ELSE IF(IMAX(ABS(JL)).EQ.I) THEN
         IRY(1+INC)=IR(I-2,JL)
         IRY(2+INC)=IR(I-1,JL)
         IRY(3+INC)=IR(I,JL)
        END IF
C  Find the Lagrange interpolation coefficients L1,L2,L3
       CALL LAG(IRY(7),IRY(1+INC),IRY(2+INC),IRY(3+INC),L1,L2,L3,0)
       OY(1+INC)=D*L1
       OY(2+INC)=D*L2
       OY(3+INC)=D*L3
       INC=3
       D=D2
200    CONTINUE
      END IF
      END
C  ****************************************
      SUBROUTINE FDYMIN(I,J,IRY,OY)
C  Finds grid points and corresponding operator coefficients for
C  7 point finite difference formula for derivatives wrt Y
C  at J boundary point I,J   (I = IMIN(J))
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION L1,L2,L3
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000),OY(7)
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70),IRY(7)
      INTEGER IB(0:70),IRB(0:70),IRBM(0:70),IBIND(0:70)
      INTEGER NX,NY,IAPEX,IBMAX
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
```

```
      COMMON/DXYG/DXG,DYG,DX,DY
      PARAMETER(ONE=1.0D0,TWO=2.0D0,THREE=3.0D0)
C     --------------------------------
      JM1=J-1
C  For IMIN(J)=0 a 7 point interpolation/extrapolation central
C  difference is used. This involves point I,J and 3 points each
C  each from rows J-1 and J+1
      IF(J.LE.1) THEN
      IF(J.EQ.0) THEN
       D=1.0D0/(TWO*DYG(0))
       IRY(4)=IR(0,0)
       OY(4)=0.0D0
       IRY(1)=IR(0,-1)
       IRY(2)=IR(1,-1)
       IRY(3)=IR(2,-1)
       IRY(5)=IR(0,1)
       IRY(6)=IR(1,1)
       IRY(7)=IR(2,1)
C  Find the Lagrange interpolation coefficients L1,L2,L3
       CALL LAG(IRY(4),IRY(5),IRY(6),IRY(7),L1,L2,L3,0)
       OY(5)=D*L1
       OY(6)=D*L2
       OY(7)=D*L3
       OY(1)=-OY(5)
       OY(2)=-OY(6)
       OY(3)=-OY(7)
      ELSE
C  Determine coefficients D1,D2,D3 for central differences wrt Y
        D1=-ONE/(TWO*DYG(J))
        D2=0.0d0
        D3=-D1
       OY(4)=0.0d0
       IRY(4)=IR(I,J)
C Determine interpolation points to be used on rows j-1 and j+1.
       INC=0
       D=D1
       DO 100 JL=JM1,J+1,2
       IRY(1+INC)=IR(IMIN(JL),JL)
       IRY(2+INC)=IR(IMIN(JL)+1,JL)
       IRY(3+INC)=IR(IMIN(JL)+2,JL)
C  Find the Lagrange interpolation coefficients L1,L2,L3
       CALL LAG(IRY(4),IRY(1+INC),IRY(2+INC),IRY(3+INC),L1,L2,L3,0)
       OY(1+INC)=D*L1
       OY(2+INC)=D*L2
       OY(3+INC)=D*L3
       INC=4
       D=D3
100    CONTINUE
      END IF
C  For IMIN(J) > 0 a 7 point interpolation backward difference is used.
C  This involves point I,J and 3 points each from rows J-2 & J-1
      ELSE IF(J.GT.1) THEN
C  Determine coefficients D1,D2,D3 for backward differences wrt Y
```

```
        D1=ONE/(TWO*DYG(JM1))
        D2=-TWO/DYG(JM1)
        D3=THREE/(TWO*DYG(JM1))
       OY(7)=D3
       IRY(7)=IR(I,J)
C Determine interpolation points to be used on rows J-2 and J-1
      INC=0
      D=D1
      DO 200 JL=J-2,JM1
      IF(IMIN(ABS(JL)).LE.I-1) THEN
       IRY(1+INC)=IR(I-1,JL)
       IRY(2+INC)=IR(I,JL)
       IRY(3+INC)=IR(I+1,JL)
      ELSE IF(IMIN(ABS(JL)).EQ.I) THEN
       IRY(1+INC)=IR(I,JL)
       IRY(2+INC)=IR(I+1,JL)
       IRY(3+INC)=IR(I+2,JL)
      END IF
C  Find the Lagrange interpolation coefficients L1,L2,L3
      CALL LAG(IRY(7),IRY(1+INC),IRY(2+INC),IRY(3+INC),L1,L2,L3,0)
      OY(1+INC)=D*L1
      OY(2+INC)=D*L2
      OY(3+INC)=D*L3
      INC=3
      D=D2
200   CONTINUE
      END IF
      END
C  ****************************************
      SUBROUTINE LAG(IR,IR1,IR2,IR3,L1,L2,L3,IJ)
C 3-point Lagrange interpolation. If ij=0, interpolate w.r.t x as
C required for j boundary points. Otherwise w.r.t y as required
C for i boundary points
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION L1,L2,L3
      COMPLEX*16 Z(0:4000),IM
      COMMON/CMZ/Z
      PARAMETER(IM=(0.0D0,1.0D0))
C     --------------------------------
      IF(IJ.EQ.0)THEN
      U=REAL(Z(IR))
      U1=REAL(Z(IR1))
      U2=REAL(Z(IR2))
      U3=REAL(Z(IR3))
      ELSE
      U=REAL(-IM*Z(IR))
      U1=REAL(-IM*Z(IR1))
      U2=REAL(-IM*Z(IR2))
      U3=REAL(-IM*Z(IR3))
      END IF
      L1=(U-U2)*(U-U3)/((U1-U2)*(U1-U3))
      L2=(U-U1)*(U-U3)/((U2-U1)*(U2-U3))
      L3=(U-U1)*(U-U2)/((U3-U1)*(U3-U2))
      RETURN
      END
C    ****************************************************
      SUBROUTINE INTSOL
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMPLEX*16 Z,ED,ETA,ETAB
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/RKCOM/RK
      COMMON/NGC/NG,NG1,NG2
      COMMON/SIGC/SIG(0:4000)
      COMMON/init/sinit(0:4000)
      COMMON/CMMD/SIGD(0:4000)
      COMMON/CMZ/Z(0:4000)
      PARAMETER(TWO=2.0D0,ONE=1.0D0)
C-------------------------------------
c      OPEN(UNIT=10,FILE='iout.d')
      E0=ONE/TAN(PH/TWO)
      DO 40 M=0,NG
      ED=Z(M)
       ETA=(E0-ED)/(ONE+E0*ED)
       ETAB=CONJG(ETA)
       ETAP=ETA+ETAB
       ETA2=ETA*ETAB
      SIG(M)=DLOG(RK)-DLOG(ONE-E*DCOS(G)-E*(ETAP)*DSIN(G)+
     +ETA2*(ONE+E*DCOS(G)))
40     CONTINUE
       DO 50 I=0,NG
       SIGD(I)=SIG(I)
       sinit(i)=sig(i)
c               write(113,60)I,SIG(I)
c60             FORMAT(3X,'SIG(',I3,')=',E15.8)
50     CONTINUE
      RETURN
      END
C     ********************************
      SUBROUTINE RAW0
C  evalute radious(Ra) and center of the apertur circle(W0)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION CO(2)
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/RACOM/R,A,EN
      COMMON/WOR/RA,W0
      PARAMETER (ONE=1.0D0,TWO=2.0D0)
C     ------------------------------
      XR=-R
      DO 10 I=1,2
      ED=XR
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=ETA
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      CUS=(ETAP*SIN(G)-(ETA2-ONE)*COS(G))/(ONE+ETA2)
          CALL WR(ETA,ETAB,CUS,W)
      CO(I)=W
      XR=R
 10   CONTINUE
      RA=ABS(CO(1)-CO(2))/TWO
      write(6,*)'      ra=',ra
      write(9,*)'      ra=',ra
      W0=(CO(1)+CO(2))/TWO
      RETURN
      END
C     ***********************************
      SUBROUTINE WR(ETA,ETAB,CUS,W)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/RKCOM/RK
      COMMON/RACOM/R,A,EN
      PARAMETER (ONE=1.0D0)
C     --------------------------------
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      ELE=(E*SIN(G)-ETAB*(ONE+E*COS(G)))/(
     + ONE-E*COS(G)-E*ETAP*SIN(G)+ETA2*(ONE+E*COS(G)))
      SQ=ETAB+(ONE+ETA2)*ELE
      SQ1=SQ*SQ
      XI1=(EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELE+ONE
      SQT=ONE-(EN*EN-ONE)*SQ1
      XI2=DSQRT(SQT)
       if(sqt.lt.0.0d0)then
        write(6,*)'in wr, dsqrt(xi)=negative.'
       stop
        endif
      XI3=(EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELE)
      XI=(XI1-XI2)/XI3
      XIB=XI
      REP=RK/(ONE-E*CUS)
      EXPL=REP/(ONE+ETA2)
      W=2.0D0*ETA*EXPL+(A-EXPL*((EN+ONE)*ETA2+EN-ONE))/XIB
      RETURN
      END
C     *************************************
      SUBROUTINE ITERAT
C Iterates the solution procedure for the mapping functions OMEGA &SIGMA
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      INTEGER ICON,ITMAX,it
      COMMON/CITMAX/ITMAX
      COMMON/ITCOM/IT
      COMMON/ITERATC/NIT
      COMMON/UCOM/U
      COMMON/IUCOM/IU
      COMMON/ILCOM/IL
      COMMON/DKCOM/DK
      COMMON/delc/delta
C     --------------------------------
      WRITE(6,*) '**************************** U=',U
      DO 100 IT=1,ITMAX
C     --------------------------------
               WRITE(6,*)' IT=',IT
C  Calculate new values of SIGMA
      CALL BOUND
      If(il.eq.0.and.it.eq.1)then
                 CALL INANC
c       CALL GAUSS(ICON)
      else
                 CALL GI
       If(il.eq.iu.and.it.eq.nit)then
       WRITE(9,998)U,IT
       WRITE(9,*)'k=',dk,'   deltak=',delta
       dk=dk+delta
                 CALL INANC
       CALL GAUSS(ICON)
       RETURN
       endif
                 CALL INANC
      endif
       CALL GAUSS(ICON)
C If convergence achieved (ICON = 1) no further iterations necessary.
      IF(ICON.EQ.1)THEN
              WRITE(9,998)U,IT
998   FORMAT('u= ',f6.3,'** Convergence achieved after'
     +,I3,' iterations')
       RETURN
       ENDIF
100    CONTINUE
       WRITE(9,999)U,ITMAX
999   FORMAT('u=',f6.3,'  *** Convergence not achieved after'
     +,I3,'  iterations')
       STOP
       END
C**************************
      SUBROUTINE BOUND
C  Finds operator coefficients and matrix element numbering for the
C  linearised boundary condition equation
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION OX(7,300),OY(7,300)
      DOUBLE PRECISION OPS(300000),VEC(0:4000)
      INTEGER IRN(150000),ICN(300000)
      INTEGER IRX(7,300),IRY(7,300)
      INTEGER ISUMX(300),ISUMY(300),IBOUND(300)
      INTEGER NB1,NB2,KA
      LOGICAL FULL
      COMMON/BDOP/OX,OY,IRX,IRY,ISUMX,ISUMY,IBOUND,NB1,NB2
      COMMON/FULLC/FULL
      COMMON/OPVEC/OPS,VEC,KA,IRN,ICN
      COMMON/SIGC/SIG(0:4000)
C     --------------------------------
      KA=0
      NB=NB1
      IF(FULL) NB=NB2
      DO 300 IB=1,NB
          CALL BDER(IB,SIGX,SIGY)
C      Find the coefficients of the linearised boundary condition
       IJ=IBOUND(IB)
          CALL LINBC (IJ,SIGX,SIGY,ALPHA,BETA,GAMA,DEL)
C  Assign the vector element and operator matrix elements corresponding
C  to this boundary point
      VEC(IJ)=DEL
C alpha'(d/dx)
      DO 110 L=1,ISUMX(IB)
      KA=KA+1
      OPS(KA)=ALPHA*OX(L,IB)
      IRN(KA)=IJ+1
      ICN(KA)=IRX(L,IB)+1
110   CONTINUE
C Beta'(d/dy)
      DO 120 L=1,ISUMY(IB)
      KA=KA+1
      OPS(KA)=BETA*OY(L,IB)
      IRN(KA)=IJ+1
      ICN(KA)=IRY(L,IB)+1
120   CONTINUE
C     Gama
      KA=KA+1
      OPS(KA)=GAMA
      IRN(KA)=IJ+1
      ICN(KA)=IJ+1
300   CONTINUE
      RETURN
      END
C     **************************************************
      SUBROUTINE LINBC (IJ,SIGX,SIGY,ALPHA,BETA,GAMA,DEL)
C     Finds coefficients of the differential operators
C of the linearised boundary condition equation.
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMPLEX*16 XI,XIB,XIE,XIEB,XIBEB,XIBE
      COMPLEX*16 WB,WXIB,WBXI,WL,WBL
      COMPLEX*16 TW,TWB,TE
      COMPLEX*16 ED,EDB,EDE,EDBEB
      COMPLEX*16 IM,A,B
      COMMON/SIGC/SIG(0:4000)
      COMMON/KWW/WXIB,WBXI,WL,WBL
      COMMON/XICM/XI,XIB
      COMMON/XIECM/XIE,XIBEB,XIEB,XIBE
      COMMON/WOR/RA,WO
      COMMON/EDMAPC/ED,EDB,EDE,EDBEB
      COMMON/WS/WB
      PARAMETER (TWO=2.0D0,IM=(0.0D0,1.0D0))
C     --------------------------------
                CALL MAPCON(IJ,SIG(IJ),SIGX,SIGY)
        TW=WB-WO
        TWB=CONJG(TW)
        TE=TW*(WXIB*XIBE)+TWB*(WBXI*XIE)
        TL=TW*WL+TWB*WBL
      A=EDE*TE
      B=CONJG(A)
      C=TL
      PHI=TWB*TW-RA*RA
      ALPHA=(A+B)/TWO
      BETA=IM*(B-A)/TWO
      GAMA=C
      DEL=ALPHA*SIGX+BETA*SIGY+GAMA*SIG(IJ)-PHI
      RETURN
      END
C ******************************
      SUBROUTINE MAPCON (IR,L,XL,YL)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION L,XL,YL,PP,SQ
      INTEGER IR
      COMPLEX*16 XI1,XI2,XI1E,XI1EB,XIB1,XIB1E,XIB1EB
      COMPLEX*16 XIB2,XI2EB,XIB2E
      COMPLEX*16 ETA,ETAB,XI,XIB,XIE,XIBE,XIEB,XIBEB
      COMPLEX*16 ED,EDB,EDE,EDBEB
      COMPLEX*16 ELE,ELEB,ELD
      COMPLEX*16 IM,Z
      COMPLEX*16 SQE,SQEB
      COMPLEX*16 AS
      COMPLEX*16 EDG
      COMMON/ITCOM/IT
      COMMON/RACOM/R,A,EN
      COMMON/CMZ/Z(0:4000)
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/EDMAPC/ED,EDB,EDE,EDBEB
      COMMON/ETACM/ETA,ETAB
      COMMON/XICM/XI,XIB
      COMMON/XIECM/XIE,XIBEB,XIEB,XIBE
      COMMON/EDGCOM/EDG
      PARAMETER(ONE=1.0D0,TWO=2.0D0,IM=(0.0D0,1.0D0))
C     --------------------------------
      ED=Z(IR)
      EDB=CONJG(ED)
      EDG=ED
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)*(ONE+ED*E0)/(ONE+E0*E0)
      EDBEB=CONJG(EDE)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      PP=ONE-(EN*EN-ONE)*AS*CONJG(AS)
       if(pp.lt.0.0d0)then
        write(6,*)'****in mapcon, dsqrt(pp),pp is negative.***',ir,pp
        stop
        endif
       SQ=DSQRT(PP)
       XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-DSQRT(PP))/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
       SQE=(-(EN**2-ONE)*(ONE+ETA2)*(ETA+(ONE+ETA2)*ELEB))/(TWO*SQ)
       SQEB=(-(EN**2-ONE)*(ONE+ETA2)*(ETAB+(ONE+ETA2)*ELE))/(TWO*SQ)
       XI1=(EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ
       XI2=(EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB)
       XIB=CONJG(XI)
       XI1E=(EN+ONE)*ETA-SQE
       XI2E=EN+ONE
       XI1EB=-(EN-ONE)*ETAB-SQEB
       XI2EB=(EN-ONE)*ETAB**2
       XIE=(XI1E*XI2-XI2E*XI1)/(XI2**2)
       XIEB=(XI1EB*XI2-XI2EB*XI1)/(XI2**2)
       XIB1=(EN+ONE)*ETAB*ELEB-(EN-ONE)*ETA*ELE+ONE-SQ
       XIB1E=-(EN-ONE)*ETA-SQE
       XIB1EB=(EN+ONE)*ETAB-SQEB
       XIB2=(EN+ONE)*ELEB+(EN-ONE)*ETA*(ONE+ETA*ELE)
       XIB2E=(EN-ONE)*(ETA**2)
       XIB2EB=(EN+ONE)
       XIBE=(XIB1E*XIB2-XIB2E*XIB1)/(XIB2**2)
       XIBEB=(XIB1EB*XIB2-XIB2EB*XIB1)/(XIB2**2)
       CALL KW(L,EN,A,XIB)
       RETURN
       END
C  ****************************
      SUBROUTINE KW(L,EN,A,XIB)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION L,K,KL
      COMPLEX*16 ETA,ETAB,XIB
      COMPLEX*16 W,WB,WL,WBL,WXIB,WBXI
      COMPLEX*16 KE,KEB,KEL,KEBL
      COMMON/ETACM/ETA,ETAB
      COMMON/WS/WB
      COMMON/WCOM/W
      COMMON/KWCK/K,KL,KE,KEL,KEB,KEBL
      COMMON/KWW/WXIB,WBXI,WL,WBL
      PARAMETER(ONE=1.0D0,TWO=2.0D0)
c     ---------------------------------
      ETA2=ETA*ETAB
      KL=-EXP(L)*((EN+ONE)*ETA2+EN-ONE)
      K=A+KL
c   exp(l) is function of the eta  exp(l)=r/(1+eta2)
      KE=-TWO*ETAB*EXP(L)/(ONE+ETA2)
      KEB=CONJG(KE)
      KEL=KE
      KEBL=KEB
      W=TWO*ETA*EXP(L)+K/XIB
      WB=CONJG(W)
      WL=TWO*ETA*EXP(L)+KL/XIB
      WBL=CONJG(WL)
      WXIB=-K/(XIB*XIB)
      WBXI=CONJG(WXIB)
      RETURN
      END
C    ****************
      SUBROUTINE INANC
C  Sets up scan of internal grid points in order to evaluate coefficient
C  of the linearised Monge-Ampere equation
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION OPS(300000),VEC(0:4000)
      INTEGER IRN(150000),ICN(300000),IMIN(0:70),IMAX(0:70),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IRBM(0:70),IBIND(0:70),IR(0:70,-40:40)
      INTEGER NX,NY,IAPEX,IBMAX,IT,KA
      LOGICAL LSYM,FULL
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/SIGC/SIG(0:4000)
      COMMON/FULLC/FULL
      COMMON/OPVEC/OPS,VEC,KA,IRN,ICN
      COMMON/ITCOM/IT
C     --------------------------------
c  Scan points on the line J=0, which is possibly a line of symmetry
      DO 100 I=1,NX-1
      IJ=IR(I,0)
      IF(I.EQ.IAPEX)THEN
C  Keep value of SIGMA fixed at the point I = IAPEX , J = 0
C      WRITE(6,'('' SOLUTION ANCHORED'')')
       VEC(IJ)=SIG(IJ)
       KA=KA+1
       OPS(KA)=1.0D0
       IRN(KA)=1+IJ
       ICN(KA)=1+IJ
       ELSE
C  Determine the coefficients of the linearised Monge-Ampere equation
      LSYM=.TRUE.
      IF(FULL)LSYM=.FALSE.
      CALL IELEM(IJ,LSYM)
      END IF
100   CONTINUE
C                    Scan points on the remaining J lines
       DO 200 J=1,NY-1
      DO 200 I=IMIN(J)+1,IMAX(J)-1
      CALL IELEM(IR(I,J),.FALSE.)
      IF(FULL) CALL IELEM(IR(I,-J),.FALSE.)
200   CONTINUE
      RETURN
      END
C  ****************************
      SUBROUTINE IELEM(IJ,LSYM)
C  Assigns vector and operator matrix elements for the linearised
C  Monge-Ampere equation at the internal point given by IJ
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION OPS(300000),VEC(0:4000)
      DOUBLE PRECISION OX(3,0:4000),OY(3,0:4000),OXX(3,0:4000)
      DOUBLE PRECISION OYY(3,0:4000),OXY(9,0:4000)
      INTEGER IRX(3,0:4000),IRY(3,0:4000),IRXY(9,0:4000)
      INTEGER IRN(150000),ICN(300000),KA
      LOGICAL LSYM
      COMMON/INDOP/OX,OXX,OY,OYY,OXY,IRX,IRY,IRXY
      COMMON/OPVEC/OPS,VEC,KA,IRN,ICN
C  Evaluate the coefficients of the linearised Monge-Ampere equation
      CALL LINMA(IJ,LSYM,ALPHA1,ALPHA2,GAMA,BETA1,BETA2,OMEGA,DELTA)
C Assign the vector element and operator matrix elements corresponding
C to point IJ
      VEC(IJ)=DELTA
c         WRITE(6,*)'IJ',IJ,'  VEC=DEL',DEL
C alpha(d2/dx2)
      DO 100 L=1,3
       KA=KA+1
       OPS(KA)=ALPHA1*OX(L,IJ)+ALPHA2*OXX(L,IJ)
c      WRITE(14,*)' KA',KA,'  OPS(KA)=',OPS(KA)
       IRN(KA)=1+IJ
       ICN(KA)=1+IRX(L,IJ)
100   CONTINUE
      IF(LSYM)THEN
       DO 200 L=2,3
       KA=KA+1
       OPS(KA)=BETA2*OYY(L,IJ)
       IRN(KA)=1+IJ
       ICN(KA)=1+IRY(L,IJ)
200   CONTINUE
      ELSE
      DO 300 L=1,3
       KA=KA+1
       OPS(KA)=BETA1*OY(L,IJ)+BETA2*OYY(L,IJ)
       IRN(KA)=1+IJ
       ICN(KA)=1+IRY(L,IJ)
300   CONTINUE
C gamma(d2/dxdy)
      DO 400 L=1,9
       KA=KA+1
       OPS(KA)=GAMA*OXY(L,IJ)
       IRN(KA)=1+IJ
       ICN(KA)=1+IRXY(L,IJ)
400   CONTINUE
      ENDIF
C omega(*SIG)
      KA=KA+1
      OPS(KA)=OMEGA
      IRN(KA)=1+IJ
      ICN(KA)=1+IJ
      RETURN
      END
C ************************
      SUBROUTINE LINMA(IJ,LSYM,ALPH1,ALPH2,GAM,BET1,BET2,OMEG,DEL)
C Finds the coefficients of the differential operators of the linearised
C Monge-Ampere equation and the quantity DEL which forms the r.h.s of
```

```
C the equation
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
c
      COMPLEX*16 A,B,D,EP
      COMPLEX*16 IM
      COMPLEX*16 ED,EDB,EDE,EDBEB
      LOGICAL LSYM
      COMMON/SIGC/SIG(0:4000)
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/EDMAPC/ED,EDB,EDE,EDBEB
      PARAMETER(ONE=1.0D0,TWO=2.0D0,FOUR=4.0D0,IM=(0.0D0,1.0D0))
C     -------------------------------
C  Evaluate derivatives of SIGMA
      CALL INDER(IJ,LSYM,SIGX,SIGY,SIGXX,SIGYY,SIGXY)
      CALL MAPCON (IJ,SIG(IJ),SIGX,SIGY)
      CALL ABCDEF(IJ,SIG(IJ),SIGX,SIGY,SIGXX,SIGYY,SIGXY,A,B,C,D,EP,
     + F,PSI)
C     -------------------------------
      ALPH2=(A*EDE**2+B*EDBEB**2+C*EDE*EDBEB)/FOUR
      BET2=(-A*EDE**2-B*EDBEB**2+C*EDE*EDBEB)/FOUR
      GAM=IM*(-A*EDE**2+B*EDBEB**2)/TWO
      ALPH1=((D+EDE*(TWO*E0*A)/(ONE+ED*E0))*EDE+(EP+EDBEB*(TWO*E0*B)/
     +(ONE+EDB*E0))*EDBEB)/TWO
      BET1=IM*((-D+EDE*(TWO*E0*A)/(ONE+ED*E0))*EDE+(EP+EDBEB*(TWO*E0*B)/
     +(ONE+EDB*E0))*EDBEB)/TWO
      OMEG=F
      DEL=ALPH1*SIGX+ALPH2*SIGXX+GAM*SIGXY+BET1*SIGY+BET2*SIGYY+
     +OMEG*SIG(IJ)-PSI
      RETURN
      END
C  ****************************************
      SUBROUTINE INDER(IJ,LSYM,SIGX,SIGY,SIGXX,SIGYY,SIGXY)
C  Evaluates derivatives at the internal point given by IJ
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION OX(3,0:4000),OY(3,0:4000)
      DOUBLE PRECISION OXX(3,0:4000),OYY(3,0:4000),OXY(9,0:4000)
      INTEGER IRX(3,0:4000),IRY(3,0:4000),IRXY(9,0:4000)
      LOGICAL LSYM
      COMMON/SIGC/SIG(0:4000)
      COMMON/INDOP/OX,OXX,OY,OYY,OXY,IRX,IRY,IRXY
      PARAMETER (ZERO=0.0D0)
C     -------------------------------
      SIGX=ZERO
      SIGXX=ZERO
      DO 80 L=1,3
      SIGX=SIGX+OX(L,IJ)*SIG(IRX(L,IJ))
      SIGXX=SIGXX+OXX(L,IJ)*SIG(IRX(L,IJ))
80    CONTINUE
      SIGY=ZERO
      SIGYY=ZERO
      SIGXY=ZERO
      IF(LSYM) THEN
C  On line of symmetry SIGY & SIGXY are zero
```

```
      SIGYY=OYY(2,IJ)*SIG(IRY(2,IJ))+OYY(3,IJ)*SIG(IRY(3,IJ))
      ELSE
      DO 100 L=1,3
      SIGY=SIGY+OY(L,IJ)*SIG(IRY(L,IJ))
      SIGYY=SIGYY+OYY(L,IJ)*SIG(IRY(L,IJ))
100   CONTINUE
      DO 200 L=1,9
      SIGXY=SIGXY+OXY(L,IJ)*SIG(IRXY(L,IJ))
200   CONTINUE
      END IF
      RETURN
      END
C ***************************
      SUBROUTINE ABCDEF (IJ,L,XL,YL,XXL,YYL,XYL,A,B,C,D,EP,F,PSI)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000)
      DOUBLE PRECISION L,E,SEEB,C,F,PSI,ELDDB,vl,v,YC,YCL
      DOUBLE PRECISION GI(3000),G2(3000),G0(3000)
      COMPLEX*16 XI,XIB,XIE,XIEB,XIBEB,XIBE
      COMPLEX*16 IM,ELE,ELEB,ELEE,ELEBEB
      COMPLEX*16 ELD,ELDD
      COMPLEX*16 SEE
      COMPLEX*16 SEEBE
      COMPLEX*16 XC,XCB,XCE,XCBE,XCEB,XCBEB,XCL,XCBL
      COMPLEX*16 YCE,YCEB
      COMPLEX*16 ED,EDB,EDE,EDBEB
      COMPLEX*16 VE,VEB
      COMPLEX*16 A,B,D,EP
      COMPLEX*16 EE,EEB
      COMPLEX*16 W
      COMMON/DXYG/DXG,DYG,DX,DY
      COMMON/XICM/XI,XIB
      COMMON/XIECM/XIE,XIBEB,XIEB,XIBE
      COMMON/INPT/G,TETAC,PH,EF,E0
      COMMON/RACOM/R,AP,EN
      COMMON/ELECM/ELE,ELEB,ELEE,ELEBEB,ELEEB
      COMMON/EDMAPC/ED,EDB,EDE,EDBEB
      COMMON/EMAPC/E,EE,EEB
      COMMON/ITCOM/IT
      COMMON/ILCOM/IL
      COMMON/UCOM/U
      COMMON/IUCOM/IU
      COMMON/ITERATC/NIT
      COMMON/DKCOM/DK
      COMMON/GOCOM/GO
      COMMON/NXCOM/NNX
      COMMON/WCOM/W
      PARAMETER (ONE=1.0D0,TWO=2.0D0,FOUR=4.0D0,IM=(0.0D0,1.0D0))
C     --------------------------------
      ELD=(XL-IM*YL)/TWO
      ELDD=(XXL-YYL-TWO*IM*XYL)/FOUR
      ELDDB=(XXL+YYL)/FOUR
      ELE=EDE*ELD
      ELEB=CONJG(ELE)
      ELEE=EDE*EDE*ELDD+TWO*E0/(ONE+ED*E0)*EDE*EDE*ELD
      ELEBEB=CONJG(ELEE)
      ELEEB=EDE*EDBEB*ELDDB
                  CALL EMAP
      XI2=XI*XIB
      SEE=ELE*ELE
c      SEBEB=CONJG(SEE)
      SEEB=(ONE-EN*EN)*(ONE+XI2)**2/E**2
      SEEBE=(TWO*(ONE-EN*EN)*(ONE+XI2)*(XIE*XIB+XI*XIBE))/E**2
     +-TWO*(ONE-EN**2)*(ONE+XI2)**2*EE/E**3
        CALL CPQXY(L,XC,XCB,XCE,XCBE,XCEB,XCBEB,XCL,XCBL,YC,YCE,YCL,
     +  YCEB)
        CALL VMAP(IJ,V,VE,VL,VEB)
      B=ELEE-SEE-XC
      A=CONJG(B)
      C=-TWO*(ELEEB-SEEB-YC)
       if(il.eq.0)then
        if(it.eq.1)then
          G0(IJ)=-V/(A*B-C*C/FOUR)
c         u=0   , dk=1
          GI(IJ)=G0(IJ)
        else
c         u=0   ,it # 1
          GI(IJ)=dk*G0(IJ)
        endif
       else
         call gs2(gss)
         g2(ij)=gss
         GI(IJ)=dk*((ONE-U)*G0(IJ)+U*G2(ij))
        IF(IL.EQ.IU.AND.IT.EQ.NIT)THEN
         IF(IJ.LE.NNX)THEN
          RW=REAL(W)
c10     FORMAT(I3,3x,F6.3,3x,F6.3)
         ENDIF
        ENDIF
       ENDIF
       D=(-TWO*ELE-XCE)*A-XCBE*B-(SEEBE+YCE)*C+VE/GI(IJ)
       EP=CONJG(D)
       F=-XCL*A-XCBL*B-YCL*C+VL/GI(IJ)
       PSI=A*B-C*C/FOUR+V/GI(IJ)
       RETURN
       END
C     ***************************************
      SUBROUTINE CPQXY(L,XC,XCB,XCE,XCBE,XCEB,XCBEB,XCL,XCBL,
     +YC,YCE,YCL,YCEB)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION L,K,KL,YC,YCL
      COMPLEX*16 P,PB,PE,PEB,PBE,PBEB
      COMPLEX*16 Q,QB,QE,QEB,QBE,QBEB
      COMPLEX*16 XC,XCB,XCE,XCBE,XCEB,XCBEB,XCL,XCBL
      COMPLEX*16 YCE,YCEB
      COMPLEX*16 ETA,ETAB,XI,XIB,XIE,XIEB,XIBEB,XIBE
      COMPLEX*16 CP,CPB,CPE,CPEB,CPBE,CPBEB,CPL,CPBL
      COMPLEX*16 CQ,CQB,CQE,CQEB,CQBE,CQBEB,CQL,CQBL
      COMPLEX*16 KE,KEB,KEL,KEBL
      COMPLEX*16 ELE,ELEB,ELEE,ELEBEB
      COMMON/RACOM/R,AP,EN
      COMMON/ETACM/ETA,ETAB
      COMMON/XICM/XI,XIB
      COMMON/XIECM/XIE,XIBEB,XIEB,XIBE
      COMMON/KWCK/K,KL,KE,KEL,KEB,KEBL
      COMMON/ELECM/ELE,ELEB,ELEE,ELEBEB,ELEEB
      COMMON/PQMAPC/P,PB,PE,PEB,PBE,PBEB,Q,QB,QE,QEB,QBE,QBEB
      PARAMETER (ONE=1.0D0,TWO=2.0D0)
C     -------------------------------
      CP=TWO*EXP(L)*(ONE+ETA*ELE)+KE/XIB
      CPB=TWO*EXP(L)*(ONE+ETAB*ELEB)+KEB/XI
      CPE=TWO*EXP(L)*ETA-KE*XIBE/XIB**2
      CPEB=-KE*XIBEB/XIB**2
      CPBE=-KEB*XIE/XI**2
      CPBEB=TWO*EXP(L)*ETAB-KEB*XIEB/XI**2
      CQ=TWO*ETA*ELEB*EXP(L)+KEB/XIB
      CQB=TWO*ETAB*ELE*EXP(L)+KE/XI
      CQE=-(KEB*XIBE)/(XIB*XIB)
      CQEB=TWO*EXP(L)*ETA-(KEB*XIBEB)/(XIB*XIB)
      CQBE=TWO*EXP(L)*ETAB-(KE*XIE)/(XI*XI)
      CQBEB=-(KE*XIEB)/(XI*XI)
      CQL=TWO*ETA*ELEB*EXP(L)+KEBL/XIB
      CQBL=TWO*ETAB*ELE*EXP(L)+KEL/XI
      CPL=TWO*EXP(L)*(ONE+ETA*ELE)+KEL/XIB
      CPBL=TWO*EXP(L)*(ONE+ETAB*ELEB)+KEBL/XI
      CALL PQMAP(P,PB,PE,PEB,PBE,PBEB,Q,QB,QE,QEB,QBE,QBEB)
      XC=(P*CQB*XI**2+Q*CP*XIB**2)/K
      XCB=(PB*CQ*XIB**2+QB*CPB*XI**2)/K
      XCL=(P*CQBL*XI**2+Q*CPL*XIB**2)/K
     +-(P*CQB*XI**2+Q*CP*XIB**2)*KL/K**2
      XCE=(PE*CQB*XI**2+P*CQBE*XI**2+TWO*P*CQB*XI*XIE
     ++QE*CP*XIB**2+Q*CPE*XIB**2+TWO*Q*CP*XIB*XIBE)/K
      XCEB=(PEB*CQB*XI**2+P*CQBEB*XI**2+TWO*P*CQB*XI*XIEB
     ++QEB*CP*XIB**2+Q*CPEB*XIB**2+TWO*Q*CP*XIB*XIBEB)/K
      XCBL=(PB*CQL*XIB**2+QB*CPBL*XI**2)/K-(PB*CQ*XIB**2
     ++QB*CPB*XI**2)*KL/K**2
      XCBE=(PBE*CQ*XIB**2+PB*CQE*XIB**2+TWO*PB*CQ*XIB*XIBE+QBE*CPB*
     +XI**2+QB*CPBE*XI**2+TWO*QB*CPB*XI*XIE)/K
      XCBEB=(PBEB*CQ*XIB**2+PB*CQEB*XIB**2+TWO*PB*CQ*XIB*XIBEB+QBEB*CPB*
     +XI**2+QB*CPBEB*XI**2+TWO*QB*CPB*XI*XIEB)/K
      YC=(P*CPB*XI**2+Q*CQ*XIB**2)/K
      YCL=(P*CPBL*XI**2+Q*CQL*XIB**2)/K
     +-(P*CPB*XI**2+Q*CQ*XIB**2)*KL/K**2
      YCE=(PE*CPB*XI**2+P*CPBE*XI**2+TWO*P*CPB*XI*XIE+QE*CQ*XIB**2
     ++Q*CQE*XIB**2+TWO*Q*CQ*XIB*XIBE)/K
      YCEB=(PEB*CPB*XI**2+P*CPBEB*XI**2+TWO*P*CPB*XI*XIEB+QEB*CQ*XIB**2
     ++Q*CQEB*XIB**2+TWO*Q*CQ*XIB*XIBEB)/K
       RETURN
       END
```

```
C**************************************
      SUBROUTINE VMAP(IJ,V,VE,VL,VEB)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION K,KL,V1,V2,V3,PT,T1,T11,V2L,V,VL
      COMPLEX*16 AE,AEB,BE,BEB
      COMPLEX*16 VE,VEB,V1E,V1EB,V2E,V2EB,V3E,V3EB
      COMPLEX*16 P,PB,PE,PEB,PBE,PBEB
      COMPLEX*16 Q,QB,QE,QEB,QBE,QBEB
      COMPLEX*16 QP2E,QP2EB
      COMPLEX*16 ETA,ETAB,XI,XIB,XIE,XIEB,XIBEB,XIBE
      COMPLEX*16 PTE,PTEB,T1E,T1EB,T11E,T11EB
      COMPLEX*16 KE,KEB,KEL,KEBL
      COMPLEX*16 XI4E,XI4EB
      COMPLEX*16 UT,UP,PXT
      COMMON/PXTCM/UP,UT,PXT
      COMMON/RACOM/R,AP,EN
      COMMON/ETACM/ETA,ETAB
      COMMON/XICM/XI,XIB
      COMMON/XIECM/XIE,XIBEB,XIEB,XIBE
      COMMON/KWCK/K,KL,KE,KEL,KEB,KEBL
      COMMON/PQMAPC/P,PB,PE,PEB,PBE,PBEB,Q,QB,QE,QEB,QBE,QBEB
      COMMON/RKCOM/RK
      PARAMETER (ONE=1.0D0,TWO=2.0D0,FOUR=4.0D0)
C     -------------------------------
      ETA2=ETA*ETAB
      XI2=XI*XIB
      XI4=XI2*XI2
      XI4K=XI4/K**2
      XI4E=TWO*XI*XIB**2*XIE+TWO*XI**2*XIB*XIBE
      XI4EB=TWO*XI*XIB**2*XIEB+TWO*XI**2*XIB*XIBEB
      P2=P*PB
      Q2=Q*QB
      QP2=Q2-P2
      QP2E=QE*QB+Q*QBE-PE*PB-P*PBE
      QP2EB=QEB*QB+Q*QBEB-PEB*PB-P*PBEB
      CALL PTMAP(PT,PTE,PTEB,T1,T1E,T1EB,T11,T11E,T11EB)
      CALL ABMAP(A,AE,AEB,B,BE,BEB)
      V1=FOUR*(EN*PT-ONE)*(A*A*T1*T1+B*B*T11*T11)
      V2=QP2*XI4K
      V3=EN*(EN-PT)*(ONE+ETA2)**2
      V=V1*V2/V3
      V1E=FOUR*EN*PTE*(A*A*T1*T1+B*B*T11*T11)
     ++FOUR*(EN*PT-ONE)*(TWO*A*T1*T1*AE+TWO*A*A*T1*T1E+TWO*B*T11*T11*BE+
     +TWO*B*B*T11*T11E)
      V1EB=FOUR*EN*PTEB*(A*A*T1*T1+B*B*T11*T11)
     ++FOUR*(EN*PT-ONE)*(TWO*A*T1*T1*AEB+TWO*A*A*T1*T1EB+TWO*B*T11*T11*
     +BEB+TWO*B*B*T11*T11EB)
      V2L=-TWO*QP2*XI4*KL/K**3
      V2E=QP2E*XI4/K**2+QP2*XI4E/K**2
      V2EB=QP2EB*XI4/K**2+QP2*XI4EB/K**2
      V3E=-EN*PTE*(ONE+ETA2)**2
      V3EB=-EN*PTEB*(ONE+ETA2)**2
      VL=V1*V2L/V3
      VE=V1E*V2/V3+V1*V2E/V3-V1*V2*V3E/V3**2
      VEB=V1EB*V2/V3+V1*V2EB/V3-V1*V2*V3EB/V3**2
      RETURN
      END
C*****************************
      SUBROUTINE PQMAP(P,PB,PE,PEB,PBE,PBEB,Q,QB,QE,QEB,QBE,QBEB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX*16 ETA,ETAB,XI,XIB,XIE,XIEB,XIBEB,XIBE
      COMPLEX*16 P,PB,PE,PEB,PBE,PBEB
      COMPLEX*16 Q,QB,QE,QEB,QBE,QBEB
      COMPLEX*16 EE,EEB
      COMMON/ETACM/ETA,ETAB
      COMMON/XICM/XI,XIB
      COMMON/XIECM/XIE,XIBEB,XIEB,XIBE
      COMMON/RACOM/R,AP,EN
      COMMON/EMAPC/E,EE,EEB
      PARAMETER(ONE=1.0D0,TWO=2.0D0,FOUR=4.0D0)
C     -------------------------------
      P=-TWO*(EN+ONE)*(XIB-ETAB)**2/(E*E)
      PB=-TWO*(EN+ONE)*(XI-ETA)**2/(E*E)
      Q=TWO*(EN-ONE)*(ONE+XI*ETAB)**2/(E*E)
      QB=TWO*(EN-ONE)*(ONE+XIB*ETA)**2/(E*E)
      PE=-FOUR*((EN+ONE)*(XIB-ETAB)*XIBE)/(E*E)
     ++FOUR*((EN+ONE)*(XIB-ETAB)**2*EE)/(E**3)
      PEB=-FOUR*((EN+ONE)*(XIB-ETAB)*XIEB)/(E*E)
     ++FOUR*((EN+ONE)*(XIB-ETAB)**2*EEB)/(E**3)
      PBE=-FOUR*((EN+ONE)*(XI-ETA)*XIE)/(E*E)
     ++FOUR*((EN+ONE)*(XI-ETA)**2*EE)/E**3
      PBEB=-FOUR*((EN+ONE)*(XI-ETA)*XIEB)/(E*E)
     ++FOUR*((EN+ONE)*(XI-ETA)**2*EEB)/E**3
      QE=FOUR*((EN-ONE)*(ONE+XI*ETAB)*XIE*ETAB)/(E*E)
     +-FOUR*((EN-ONE)*(ONE+XI*ETAB)**2*EE)/(E**3)
      QEB=FOUR*((EN-ONE)*(ONE+XI*ETAB)*XIEB*ETAB)/(E*E)
     +-FOUR*((EN-ONE)*(ONE+XI*ETAB)**2*EEB)/(E**3)
      QBE=FOUR*((EN-ONE)*(ONE+XIB*ETA)*XIBE*ETA)/(E*E)
     +-FOUR*((EN-ONE)*(ONE+XIB*ETA)**2*EE)/(E**3)
      QBEB=FOUR*((EN-ONE)*(ONE+XIB*ETA)*XIBEB*ETA)/(E*E)
     +-FOUR*((EN-ONE)*(ONE+XIB*ETA)**2*EEB)/(E**3)
      RETURN
      END
C **********************
      SUBROUTINE EMAP
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION E,E1,E2,E3
      COMPLEX*16 ETA,ETAB,XI,XIB,XIE,XIBEB,XIBE,XIEB
      COMPLEX*16 EE,EEB
      COMMON/ETACM/ETA,ETAB
      COMMON/RACOM/R,AP,EN
      COMMON/XICM/XI,XIB
      COMMON/XIECM/XIE,XIBEB,XIEB,XIBE
      COMMON/EMAPC/E,EE,EEB
      PARAMETER (ONE=1.0D0)
C     -------------------------------
```

```
      ETA2=ETA*ETAB
      XI2=XI*XIB
      E1=EN*(ONE+ETA2)*(ONE+XI2)
      E2=(ONE+ETA*XIB)*(ONE+ETAB*XI)
      E3=(XI-ETA)*(XIB-ETAB)
      E=E1-E2+E3
      EE=EN*(ONE+ETA2)*(XIE*XIB+XI*XIBE)-ETA*XIBE*(ONE+ETAB*XI)-
     +(ONE+ETA*XIB)*ETAB*XIE+XIE*(XIB-ETAB)+(XI-ETA)*XIBE
      EEB=EN*(ONE+ETA2)*(XIEB*XIB+XI*XIBEB)-ETA*XIBEB*(ONE+ETAB*XI)
     +-(ONE+ETA*XIB)*ETAB*XIEB+XIEB*(XIB-ETAB)+(XI-ETA)*XIBEB
      RETURN
      END
C  *******************
      SUBROUTINE ABMAP(A,AE,AEB,B,BE,BEB)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION A3,B3
      COMPLEX*16 ETA,ETAB,XI,XIB,XIE,XIBEB,XIEB,XIBE
      COMPLEX*16 EPS,EPSB,A1,A2,A1E,A1EB,A2E,A2EB,A3E,A3EB,AE,AEB
      COMPLEX*16 B1,B2,B1E,B1EB,B2E,B2EB,B3E,B3EB,BE,BEB
      COMPLEX*16 IM
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/ETACM/ETA,ETAB
      COMMON/XICM/XI,XIB
      COMMON/XIECM/XIE,XIBEB,XIEB,XIBE
      COMMON/EPSCOM/EPS
      PARAMETER(ONE=1.0D0,TWO=2.0D0,IM=(0.0D0,1.0D0))
C     --------------------------------
      EPS=-IM*(ONE+E0*ETAB)/(ONE+E0*ETA)
      EPSB=CONJG(EPS)
      A1=(XIB-ETAB)*(ONE+XI*ETAB)
      A2=(XI-ETA)*(ONE+XIB*ETA)
      A31=(XI-ETA)*(XIB-ETAB)*(ONE+XIB*ETA)*(ONE+XI*ETAB)
      A3=TWO*DSQRT(A31)
      A=IM*(A1*EPSB-A2*EPS)/A3
      B1=(XI-ETA)*(ONE+XIB*ETA)
      B2=(XIB-ETAB)*(ONE+XI*ETAB)
      B3=A3
      B=(EPS*B1+EPSB*B2)/B3
      A1E=XIBE*(ONE+XI*ETAB)+(XIB-ETAB)*XIE*ETAB
      A1EB=XIBEB*(ONE+XI*ETAB)+(XIB-ETAB)*XIEB*ETAB
      A2E=XIE*(ONE+XIB*ETA)+(XI-ETA)*XIBE*ETA
      A2EB=XIEB*(ONE+XIB*ETA)+(XI-ETA)*XIBEB*ETA
      A3E=(XIE*(XIB-ETAB)*(ONE+XIB*ETA)*(ONE+XI*ETAB)
     ++(XI-ETA)*XIBE*(ONE+XIB*ETA)*(ONE+XI*ETAB)
     ++(XI-ETA)*(XIB-ETAB)*XIBE*ETA*(ONE+XI*ETAB)
     ++(XI-ETA)*(XIB-ETAB)*(ONE+XIB*ETA)*XIE*ETAB)/
     +DSQRT(A31)
      A3EB=(XIEB*(XIB-ETAB)*(ONE+XIB*ETA)*(ONE+XI*ETAB)
     ++(XI-ETA)*XIBEB*(ONE+XIB*ETA)*(ONE+XI*ETAB)
     ++(XI-ETA)*(XIB-ETAB)*XIBEB*ETA*(ONE+XI*ETAB)
     ++(XI-ETA)*(XIB-ETAB)*(ONE+XIB*ETA)*XIEB*ETAB)/
     +DSQRT(A31)
      B1E=XIE*(ONE+XIB*ETA)+(XI-ETA)*XIBE*ETA
      B1EB=XIEB*(ONE+XIB*ETA)+(XI-ETA)*XIBEB*ETA
      B2E=XIBE*(ONE+XI*ETAB)+(XIB-ETAB)*XIE*ETAB
      B2EB=XIBEB*(ONE+XI*ETAB)+(XIB-ETAB)*XIEB*ETAB
      B3E=A3E
      B3EB=A3EB
      AE=IM*(A1E*EPSB-A2E*EPS)/A3-IM*(A1*EPSB-A2*EPS)*A3E/(A3*A3)
      AEB=IM*(A1EB*EPSB-A2EB*EPS)/A3-IM*(A1*EPSB-A2*EPS)*A3EB/(A3*A3)
      BE=(B1E*EPS+B2E*EPSB)/B3-(B1*EPS+B2*EPSB)*B3E/(B3*B3)
      BEB=(B1EB*EPS+B2EB*EPSB)/B3-(B1*EPS+B2*EPSB)*B3EB/(B3*B3)
      RETURN
      END
C***********************
      SUBROUTINE PTMAP (PT,PTE,PTEB,T1,T1E,T1EB,T11,T11E,T11EB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION PT1,PT2,PT,T1,T11
      COMPLEX*16 ETA,ETAB,XI,XIB,XIE,XIBEB,XIEB,XIBE
      COMPLEX*16 PTE,PTEB,PT1E,PT1EB,PT2E,PT2EB,T1E,T1EB,T11E,T11EB
      COMPLEX*16 UP,UT,PXT,IM
      COMMON/PXTCM/UP,UT,PXT
      COMMON/ETACM/ETA,ETAB
      COMMON/XICM/XI,XIB
      COMMON/XIECM/XIE,XIBEB,XIEB,XIBE
      COMMON/RACOM/R,AP,EN
      PARAMETER(zero=0.0d0,ONE=1.0D0,TWO=2.0D0)
      PARAMETER(IM=(0.0D0,1.0D0))
C     --------------------------------
      ETA2=ETA*ETAB
      XI2=XI*XIB
      PT1=TWO*(ETA*XIB+ETAB*XI)+(ETA2-ONE)*(XI2-ONE)
      PT2=(ETA2+ONE)*(XI2+ONE)
      PT=PT1/PT2
      UP=(TWO*ETA+IM*(ETA2-ONE))/(ETA2+ONE)
      UT=(TWO*XI+IM*(XI2-ONE))/(XI2+ONE)
      PXT=(TWO*IM*(ETA*XI*(ETAB-XIB)+ETA-XI+IM*(ETA*XIB-ETAB*XI)))/PT2
      ptn=en*pt-one
      if(ptn.le.zero)then
       write(6,*)'N.COS(PHI)< OR = 0',en
       STOP
      endif
      PT1E=TWO*XIBE*ETA+TWO*XIE*ETAB+(ETA2-ONE)*(XIE*XIB+XI*XIBE)
      PT1EB=TWO*XIBEB*ETA+TWO*XIEB*ETAB+(ETA2-ONE)*(XIEB*XIB+XI*XIBEB)
      PT2E=(ETA2+ONE)*(XIE*XIB+XI*XIBE)
      PT2EB=(ETA2+ONE)*(XIEB*XIB+XI*XIBEB)
      PTE=(PT1E*PT2-PT1*PT2E)/PT2**2
      PTEB=(PT1EB*PT2-PT1*PT2EB)/PT2**2
      T1=TWO*EN*(EN-PT)/(EN*EN-ONE)
      T1E=-TWO*EN*PTE/(EN*EN-ONE)
      T1EB=-TWO*EN*PTEB/(EN*EN-ONE)
      T11=T1/PT
      T11E=(T1E*PT-PTE*T1)/(PT*PT)
      T11EB=(T1EB*PT-PTEB*T1)/(PT*PT)
      RETURN
      END
```

```
C*********************************************
      SUBROUTINE GAUSS(ICON)
C  Solution of sparse system of N equations using NAG routines
C  The operator matrix is first decomposed using F01BRF and the
C  solution obtained using F04AXF. After the first decomposition
C  pivotal values are stored and F01BSF is used in subsequent
C  decompositions
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION OPS(300000),VEC(4001),W(4001),IW(30000)
      INTEGER IRN(150000),ICN(300000),ICNL(300000),IDISP(20)
      INTEGER IKEEP(20000)
      INTEGER NG,NG1,NG2,KA
      LOGICAL ABORT(4),LBLOCK,GROW,FIRST,FULL
C  common blocks
      COMMON/CMDEV/DEV
      COMMON/CMMD/SIGD(0:4000)
      COMMON/NGC/NG,NG1,NG2
      COMMON/SIGC/SIG(0:4000)
      COMMON/OPVEC/OPS,VEC,KA,IRN,ICN
      COMMON/PIVOT/FIRST
      COMMON/FULLC/FULL
      COMMON /ITCOM/IT
C     --------------------------------
      N=NG+1
      LICN=300000
      LIRN=150000
      LBLOCK=.TRUE.
      GROW=.TRUE.
      ABORT(1)=.TRUE.
      ABORT(2)=.TRUE.
      ABORT(3)=.FALSE.
      ABORT(4)=.FALSE.
      IFAIL=-10
      OPEN(UNIT=8,FILE='ogas.d3')
      IF(FIRST) THEN
       WRITE(8,'(/4X,''INFORMATION OUTPUT FROM NAG ROUTINE F01BRF'')')
       WRITE(8,*) 'NUMBER OF GRID POINTS',N,'  DEV=',DEV
       WRITE(8,*) 'NUMBER OF NON ZERO ELEMENTS',KA
       WRITE(9,*) '---------------------------------------------------'
       WRITE(9,'(/4X,''INFORMATION OUTPUT FROM NAG ROUTINE F01BRF'')')
       WRITE(9,*) 'NUMBER OF GRID POINTS',N,'  DEV=',DEV
       WRITE(9,*) 'NUMBER OF NON ZERO ELEMENTS',KA
         DO 100 J=1,KA
         ICNL(J)=ICN(J)
100      CONTINUE
       CALL F01BRF(N,KA,OPS,LICN,IRN,LIRN,ICNL,1.0D0,IKEEP,IW,W,LBLOCK
     1 ,GROW,ABORT,IDISP,IFAIL)
       WRITE(8,*) 'IDISP 3,4,6,7',IDISP(3),IDISP(4),IDISP(6),IDISP(7)
       WRITE(9,*) 'IDISP 3,4,6,7',IDISP(3),IDISP(4),IDISP(6),IDISP(7)
       WRITE(9,*) '---------------------------------------------------'
c        IF(GROW) WRITE(9,'(/'' ON EXIT FROM F01BRF W(I) ='',g12.3)')W(1)
       FIRST=.FALSE.
       ELSE
```

```
        CALL F01BSF(N,KA,OPS,LICN,IRN,ICN,ICNL,IKEEP,IW,W,GROW,1.D-4,
     1  RPMIN,ABORT(4),IDISP,IFAIL)
c       IF(GROW) WRITE(9,'('' ON EXIT FROM F01BSF W(I) ='',G12.3)')W(1)
       END IF
C  Calculate solution
      CALL F04AXF(N,OPS,LICN,ICNL,IKEEP,VEC,W,1,IDISP,RESID)
C  Set new values of sigma and test for convergence
      ICON=1
      DO 200 J=0,NG
      SIG(J)=VEC(J+1)
       IF(ABS((SIGD(J)-SIG(J))/SIG(J)).GT.DEV) ICON=0
      SIGD(J)=SIG(J)
200   CONTINUE
c*      WRITE(9,*)'ICON=',ICON
      RETURN
      END
C     *********************************
      SUBROUTINE GI
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000)
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      COMPLEX*16 ETA,ETAB,XI,XIB,AS,ASB
      COMPLEX*16 ED,EDB,EDE
      COMPLEX*16 EPS,EPSB,A1,A2,B1,B2
      COMPLEX*16 Z,IM
      COMPLEX*16 ELE,ELEB,ELD
      LOGICAL LSYM
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/DXYG/DXG,DYG,DX,DY
      COMMON/WOR/RA,WO
      COMMON/INPT/G,TETAC,PH,E,EO
      COMMON/RACOM/RC,AA,EN
      COMMON/CMZ/Z(0:4000)
      COMMON/UCOM/U
      COMMON/PCOM/P
      COMMON/DCOM/D
      COMMON/AGBGC/AG,BG
      COMMON/DKCOM/DK
      PARAMETER(ZERO=0.0D0,ONE=1.0D0,TWO=2.0D0,FOUR=4.0D0)
      PARAMETER(IM=(0.0D0,1.0D0))
C     ----------------------------
      LSYM=.FALSE.
      PI=TWO*ASIN(1.0D0)
      RA2=RA*RA
      F=ZERO
      DO 100  J=0,NY-1
      DO 100  I=IMIN(J)+1,IMAX(J)-1
      IJ=IR(I,J)
      ED=Z(IJ)
      EDB=CONJG(ED)
      ED2=ED*EDB
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
c      EDBEB=CONJG(EDE)
C     ----------------------------------
      CALL INDER(IJ,LSYM,XL,YL,XXL,YYL,XYL)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.0.0d0)then
       write(6,*)'****in gi, dsqrt(pp),pp is negative.***',ij,pp
c       stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      XI2=XI*XIB
      EPS=-IM*(ONE+E0*ETAB)/(ONE+E0*ETA)
      EPSB=CONJG(EPS)
      A1=(XIB-ETAB)*(ONE+XI*ETAB)
      A2=(XI-ETA)*(ONE+XIB*ETA)
      A31=(XI-ETA)*(XIB-ETAB)*(ONE+XIB*ETA)*(ONE+XI*ETAB)
      A3=TWO*SQRT(A31)
      A=IM*(A1*EPSB-A2*EPS)/A3
      B1=CONJG(A1)
      B2=CONJG(A2)
      B3=A3
      B=(EPS*B1+EPSB*B2)/B3
      PT=(TWO*(ETA*XIB+ETAB*XI)+(ETA2-ONE)*(XI2-ONE))/
     +((ETA2+ONE)*(XI2+ONE))
      T1=TWO*EN*(EN-PT)/(EN*EN-ONE)
      T11=T1/PT
      FIJ=FOUR*((A*A*T1*T1+B*B*T11*T11)*(EN*PT-ONE))*DX(IJ)*DY(IJ)/
     +(EN*(ONE+ED2)**2*(EN-PT))
      F=F+FIJ
100   CONTINUE
      RHS=TWO*F
      DK=RHS/((ONE-U)*P+U*BG*PI*(ONE-EXP(-AG*RA2))/AG)
      RETURN
      END
C     **************************************************
      SUBROUTINE GRA
      IMPLICIT DOUBLE  PRECISION (A-H,O-Z)
      DOUBLE PRECISION K,KL,EX,SIG,DX,DY,XL,YL,RW
      DOUBLE PRECISION UMIN,UMAX,VMIN,VMAX,XMIN,XMAX,YMIN,YMAX
      DOUBLE PRECISION G,PP,E0,AA,RK,E,EN,CUS,SQ,ETAP,ETA2
      DOUBLE PRECISION DHK,A1,A2,B1,B2,A,B,C,D,C1,C2,D1,D2,ASS
      DOUBLE PRECISION COS,SIN,DSQRT,FUU,FVV,WEL,PPP,TETA
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      INTEGER MARGIN,NOUT,IJ,I,J,il
      COMPLEX*16 ETA,ETAB,XI,XIB,AS,ASB,W,WE
      COMPLEX*16 ED,EDE
      COMPLEX*16 Z
      COMPLEX*16 ELE,ELEB,ELD,IM
      LOGICAL LSYM
      INTRINSIC COS,SIN,DSQRT,ACOS,ABS
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/RKCOM/RK
      COMMON/RACOM/RC,AA,EN
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/CMZ/Z(0:4000)
      COMMON/SIGC/SIG(0:4000)
      COMMON/ILCOM/IL
      COMMON/IUCOM/IU
      EXTERNAL X01AAF,J06VAF,J06VCF
      EXTERNAL J06WAF,J06WBF,J06WCF,J06XFF,J06WZF
      EXTERNAL J06ABF,J06YAF,J06YCF,XXXXXX
      EXTERNAL FUU,FVV,INDER
      PARAMETER(MARGIN=1,NOUT=6,ZERO=0.0D0,
     + ONE=1.0D0,TWO=2.0D0,IM=(0.0D0,1.0D0))
C     ---------------------
        if(il.eq.0)then
         OPEN(UNIT=7,FILE='gar0.ps3')
       else if(il.eq.iu)then
         OPEN(UNIT=7,FILE='garf.ps3')
      endif
      CALL XXXXXX
      CALL J06VAF(1,NOUT)
      CALL J06VCF(1,7)
C     ---------------------
      DX=0.2
      DY=0.2
        DATA XMIN,XMAX,YMIN,YMAX/-0.5,0.2,-0.1,0.6/
        DATA UMIN,UMAX,VMIN,VMAX/0.0,0.8,0.0,0.8/
       CALL J06WAF
      CALL J06WBF(XMIN,XMAX,YMIN,YMAX,MARGIN)
      CALL J06WCF(UMIN,UMAX,VMIN,VMAX)
      CALL J06ABF(DX,DY)
      CALL J06XFF(2)
C     ----------------------
      LSYM=.FALSE.
      PPP=0.05d0
      J=0
      DO 100  I=IMIN(J),IMAX(J)
      IJ=IR(I,J)
      IF(I.EQ.IMIN(J))THEN
      ID=2*J+1
          CALL BDER(ID,XL,YL)
      ELSE IF(I.EQ.IMAX(J))THEN
      ID=2*J+2
          CALL BDER(ID,XL,YL)
      ELSE
      CALL INDER(IJ,LSYM,XL,YL,XXL,YYL,XYL)
      ENDIF
      ED=Z(IJ)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
      if(il.eq.0)then
       ELE=(E*SIN(G)-ETAB*(ONE+E*COS(G)))/
     + (ONE-E*COS(G)-E*ETAP*SIN(G)+ETA2*(ONE+E*COS(G)))
      endif
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.ZERO)then
       write(6,*)'****in gra, dsqrt(pp),pp is negative.***',ij,pp
c       stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
        EX=EXP(SIG(IJ))
      KL=-EX*((EN+ONE)*ETA2+EN-ONE)
      K=AA+KL
      W=TWO*ETA*EX+K/XIB
         if(il.eq.iu.and.i.eq.iapex)then
         w0=w
         write(9,*)'  w0=',w0
         endif
      rw=real(w)
      WE=W-TWO*ETA*EX
      WEL=WE*CONJG(WE)
      DHK=(AA-EX*((EN-ONE)*ETA2+EN+ONE)-WEL/K)/TWO
      CUS=(ETAP*SIN(G)-(ETA2-ONE)*COS(G))/(ONE+ETA2)
      TETA=ACOS(CUS)
      if(il.eq.0)then
         RT=RK/(ONE-E*CUS)
         if(i.eq.iapex)then
         write(9,*)'  initial r0=',rt
         endif
      else if(il.eq.iu)then
         RT=EX*(ONE+ETA2)
         if(i.eq.iapex)then
         write(9,*)'  final r0=',rt
         endif
      ENDIF
  A=ZERO
  B=ZERO
  C=FUU(TETA,RT,G)
  D=FVV(TETA,RT,G)
  CALL J06YAF(A,B)
  CALL J06YCF(C,D)
  CALL J06YAF(C,D)
  CALL J06YCF(-DHK,RW)
  CALL J06YAF(-DHK,RW)
  CALL J06YCF(PPP,RW)
          IF(IJ.EQ.0)THEN
           c1=c
           d1=d
           A1=-DHK
           B1=RW
          dbk=rw
          ELSE
           c2=c
           d2=d
           A2=-DHK
           B2=RW
        IF(IJ.EQ.imax(j))df=d
        if(df.gt.dbk)then
        write(6,*)'blockage'
        endif
        if(b2.lt.b1)then
        write(6,*)'the rays crossing eachother'
        endif
           CALL J06YAF(c1,d1)
           CALL J06YCF(c2,d2)
           CALL J06YAF(A1,B1)
           CALL J06YCF(A2,B2)
           c1=c2
           d1=d2
           A1=A2
           B1=B2
          ENDIF
100   CONTINUE
        CALL J06WZF
c        CLOSE(UNIT=7)
      return
      END
C     ************
DOUBLE PRECISION FUNCTION FUU(T,RT,G)
DOUBLE PRECISION T,RT,G,COS
INTRINSIC COS
FUU=RT*COS(T+G)
RETURN
END
C     ***********
```

```
      DOUBLE PRECISION FUNCTION FVV(T,RT,G)
      DOUBLE PRECISION T,RT,G,SIN
      INTRINSIC SIN
      FVV=RT*SIN(T+G)
      RETURN
      END
C     ***************************************************************
      SUBROUTINE BDER(IB,SIGX,SIGY)
      IMPLICIT DOUBLE  PRECISION (A-H,O-Z)
      DOUBLE PRECISION OX(7,300),OY(7,300)
      INTEGER IRX(7,300),IRY(7,300)
      INTEGER ISUMX(300),ISUMY(300),IBOUND(300)
      INTEGER NB1,NB2
      COMMON/BDOP/OX,OY,IRX,IRY,ISUMX,ISUMY,IBOUND,NB1,NB2
      COMMON/SIGC/SIG(0:4000)
      PARAMETER(ZERO=0.0D0)
C     --------------------------------
      SIGX=ZERO
      DO 100 L=1,ISUMX(IB)
100   SIGX=SIGX+OX(L,IB)*SIG(IRX(L,IB))
      SIGY=ZERO
      DO 200 L=1,ISUMY(IB)
200   SIGY=SIGY+OY(L,IB)*SIG(IRY(L,IB))
      RETURN
      END
C     ************************************
      SUBROUTINE GS2(G2)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMPLEX*16 W,WG
      COMMON/WCOM/W
      COMMON/WOR/RA,WO
      COMMON/AGBGC/AG,BG
C     -------------------------------------
      WG=W-WO
      RO2=WG*CONJG(WG)
      G2=BG*EXP(-AG*RO2)
      RETURN
      END
C     ***************************************
      SUBROUTINE TOTE
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION  ABSACC,ANS,FA,RC,P1,P2,P
      INTEGER IFAIL,NPTS
      COMMON/RACOM/RC,AA,EN
      COMMON/PCOM/P
      EXTERNAL FA,P1,P2
C     --------------------
      IFAIL=1
      ABSACC=1.0D-6
      CALL D01DAF(0.0D0,RC,P1,P2,FA,ABSACC,ANS,NPTS,IFAIL)
      P=2.0D0*ANS
      WRITE(6,*)'INTEGRAL=',P,' npts',NPTS,' ifail',IFAIL
      RETURN
      END
C     *************************************************************
      DOUBLE PRECISION FUNCTION FA(X,Y)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMPLEX*16 ED,ETA,ETAB,XI,XIB
      COMPLEX*16 ELE,ELEB
      COMPLEX*16 IM
      COMPLEX*16 AS
      COMPLEX*16 EPS,EPSB,A1,A2,B1,B2
      COMMON/RACOM/RC,AA,EN
      COMMON/INPT/G,TETAC,PH,E,E0
      PARAMETER(ONE=1.0D0,TWO=2.0D0,FOUR=4.0D0,IM=(0.0D0,1.0D0))
C     -------------------------------
      ED=X+IM*Y
      ED2=ED*CONJG(ED)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETA2=ETA*ETAB
      ETAP=ETA+ETAB
      ELE=(E*DSIN(G)-ETAB*(ONE+E*DCOS(G)))/(
     + ONE-E*DCOS(G)-E*ETAP*DSIN(G)+ETA2*(ONE+E*DCOS(G)))
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      PP=ONE-(EN*EN-ONE)*AS*CONJG(AS)
       if(pp.lt.0.0d0)then
       write(6,*)'****in TOTE, dsqrt(pp),pp is negative.***’,pp
       stop
       endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      XI2=XI*XIB
      EPS=-IM*(ONE+E0*ETAB)/(ONE+E0*ETA)
      EPSB=CONJG(EPS)
      A1=(XIB-ETAB)*(ONE+XI*ETAB)
      A2=CONJG(A1)
      A31=(XI-ETA)*(XIB-ETAB)*(ONE+XIB*ETA)*(ONE+XI*ETAB)
      A3=TWO*DSQRT(A31)
      A=IM*(A1*EPSB-A2*EPS)/A3
      B1=A2
      B2=A1
      B3=A3
      B=(EPS*B1+EPSB*B2)/B3
      PT1=TWO*(ETA*XIB+ETAB*XI)+(ETA2-ONE)*(XI2-ONE)
      PT2=(ETA2+ONE)*(XI2+ONE)
      PT=PT1/PT2
      T1=TWO*EN*(EN-PT)/(EN*EN-ONE)
      T11=T1/PT
      ABT=A*A*T1*T1+B*B*T11*T11
      FA=(FOUR*ABT*(EN*PT-ONE))/(EN*(ONE+ED2)**2*(EN-PT))
      RETURN
      END
```

```
C     ********************************
  DOUBLE PRECISION FUNCTION P1(Y)
DOUBLE PRECISION Y,RC,AA,EN
        COMMON/RACOM/RC,AA,EN
        P1=-DSQRT(RC*RC-Y*Y)
      RETURN
      END
C     ********************************
  DOUBLE PRECISION FUNCTION P2(Y)
DOUBLE PRECISION Y,RC,AA,EN
        COMMON/RACOM/RC,AA,EN
        P2=DSQRT(RC*RC-Y*Y)
      RETURN
      END
C     ********************************
      SUBROUTINE AGBG
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/WOR/RA,WO
      COMMON/PCOM/P
      COMMON/DCOM/D
      COMMON/AGBGC/AG,BG
      PARAMETER(ONE=1.0D0,TWO=2.0D0,TEN=10.0D0)
C     --------------------------------
      PI=TWO*ASIN(ONE)
      RA2=RA*RA
      AG=D*LOG(TEN)/(TEN*RA2)
      BG=P*AG/(PI*(ONE-EXP(-AG*RA2)))
      RETURN
      END
C     *****************************************
      SUBROUTINE CROSSP
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K,KL
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      COMPLEX*16 ETA,ETAB,XI,XIB,AS,ASB
      COMPLEX*16 ED,EDB,EDE
      COMPLEX*16 EPS,EPSB,W,wmax
      COMPLEX*16 Z,IM
      COMPLEX*16 ANU,XK,GAMA
      COMPLEX*16 ELE,ELEB,ELD
      LOGICAL LSYM
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
       COMMON/INPT/G,TETAC,PH,E,E0
       COMMON/RACOM/RC,AA,EN
       COMMON/CMZ/Z(0:4000)
       COMMON/SIGC/SIG(0:4000)
       COMMON/cpijm/ijmax
      PARAMETER(ONE=1.0D0,TWO=2.0d0,IM=(0.0D0,1.0D0))
C     --------------------------------
c      OPEN(UNIT=26,FILE='dsib.d3')
      LSYM=.FALSE.
      PI=TWO*ASIN(ONE)
      do 110 l=1,2
      DO 100  J=1,NY-1
      DO 100  I=IMIN(J)+1,IMAX(J)-1
      IJ=IR(I,J)
         if(l.eq.2)then
         IJ=IR(I,J)
         endif
      ED=Z(IJ)
      EDB=CONJG(ED)
      ED2=ED*EDB
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
C     ----------------------------------
      CALL INDER(IJ,LSYM,XL,YL,XXL,YYL,XYL)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.0.0d0)then
       write(6,*)'****in crossp, dsqrt(pp),pp is negative.***',ij,pp
c       stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      KL=-EXP(SIG(IJ))*((EN+ONE)*ETA2+EN-ONE)
      K=AA+KL
      W=TWO*ETA*EXP(SIG(IJ))+K/XIB
      EPS=-IM*(ONE+E0*ETAB)/(ONE+E0*ETA)
      EPSB=CONJG(EPS)
      ANU=EPS*(ONE+XIB*ETA)**2+EPSB*(XIB-ETAB)**2
      ANU2=ANU*CONJG(ANU)
      GAMA=ANU/DSQRT(ANU2)
      XK=GAMA*XI/XIB
      REX=REAL(XK)
      AIMX=REAL(-IM*XK)
      DSIB=10.0D0*LOG((REX/AIMX)**2)/lOG(10.0D0)
c       WRITE(26,*)IJ,' Dsib=',DSIB
c       WRITE(26,*)IJ,' W',W
c       WRITE(26,*)
c      WRITE(26,10)IJ,DSIB,W
c10    format(2x,i3,3x,f16.11,4x,'(',x,f16.14,3x,f16.14,')')
      if(j.eq.1.and.i.eq.imin(1)+1)dsimax=dsib
      if(dsib.gt.dsimax)then
      dsimax=dsib
      ijmax=ij
      wmax=w
      endif
100   continue
110   continue
      WRITE(9,*)'  **************************'
      WRITE(9,*)'  max c.p ij',IJMAX
      WRITE(9,*)'  max c.p  =',DSIMAX
      WRITE(9,*)'  max c.p  w',WMAX
      WRITE(9,*)'  **************************'
c      CLOSE(UNIT=26)
      RETURN
      END
C     ****************************************
      SUBROUTINE TRANSMIT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION DXG(0:70),DYG(0:40),DX(0:4000),DY(0:4000)
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      COMPLEX*16 ETA,ETAB,XI,XIB,AS,ASB
      COMPLEX*16 ED,EDB,EDE
      COMPLEX*16 EPS,EPSB,A1,A2,B1,B2
      COMPLEX*16 Z,IM
      COMPLEX*16 ELE,ELEB,ELD
      LOGICAL LSYM
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/DXYG/DXG,DYG,DX,DY
      COMMON/WOR/RA,WO
      COMMON/INPT/G,TETAC,PH,E,EO
      COMMON/RACOM/RC,AA,EN
      COMMON/CMZ/Z(0:4000)
      COMMON/UCOM/U
      COMMON/PCOM/P
      COMMON/DCOM/D
      COMMON/AGBGC/AG,BG
      COMMON/DKCOM/DK
      COMMON/t12min/ijmin
      PARAMETER(ONE=1.0D0,TWO=2.0D0)
      PARAMETER(IM=(0.0D0,1.0D0))
C     --------------------------------
      LSYM=.FALSE.
      DO 110  l=0,2
      DO 100  J=0,NY-1
      DO 100  I=IMIN(J)+1,IMAX(J)-1
      IJ=IR(I,J)
         if(l.eq.2)then
         IJ=IR(I,J)
         endif
      ED=Z(IJ)
      EDB=CONJG(ED)
      ED2=ED*EDB
      ETA=(EO-ED)/(ONE+EO*ED)
      ETAB=CONJG(ETA)
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      CALL INDER(IJ,LSYM,XL,YL,XXL,YYL,XYL)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.0.0d0)then
       write(6,*)'****in TRANSMIT, dsqrt(pp),pp is negative.***',ij,pp
c       stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      XI2=XI*XIB
      EPS=-IM*(ONE+E0*ETAB)/(ONE+E0*ETA)
      EPSB=CONJG(EPS)
      A1=(XIB-ETAB)*(ONE+XI*ETAB)
      A2=(XI-ETA)*(ONE+XIB*ETA)
      A31=(XI-ETA)*(XIB-ETAB)*(ONE+XIB*ETA)*(ONE+XI*ETAB)
      A3=TWO*SQRT(A31)
      A=IM*(A1*EPSB-A2*EPS)/A3
      B1=CONJG(A1)
      B2=CONJG(A2)
      B3=A3
      B=(EPS*B1+EPSB*B2)/B3
      PT=(TWO*(ETA*XIB+ETAB*XI)+(ETA2-ONE)*(XI2-ONE))/
     +((ETA2+ONE)*(XI2+ONE))
      T1=TWO*EN*(EN-PT)/(EN*EN-ONE)
      T11=T1/PT
      T12=(EN*PT-ONE)*(A*A*T1*T1+B*B*T11*T11)/(EN*(EN-PT))
      if(j.eq.0.and.i.eq.1)t12max=t12
      if(t12.gt.t12max)then
      t12max=t12
      ijmax=ij
      endif
      if(j.eq.0.and.i.eq.1)t12min=t12
      if(t12.lt.t12min)then
      t12min=t12
      ijmin=ij
      endif
100   CONTINUE
110   CONTINUE
      WRITE(9,*)'  **************************'
      WRITE(9,*)'  ijmaxt12',IJMAX,'    t12max=',T12MAX
      WRITE(9,*)'  ijmin',IJMIN
      WRITE(9,*)'    t12min=',T12MIN
      RETURN
      END
```

```
C     **********************************************
      SUBROUTINE APERG
c  aperture garph
      IMPLICIT DOUBLE  PRECISION (A-H,O-Z)
      DOUBLE PRECISION OX(7,300),OY(7,300)
      DOUBLE PRECISION K,KL,EX,SIG,XL,YL,RW,RWB,AA
      DOUBLE PRECISION UMIN,UMAX,VMIN,VMAX,XMIN,XMAX,YMIN,YMAX
      DOUBLE PRECISION SQ,ETAP,ETA2
      DOUBLE PRECISION size,x1,y1
      DOUBLE PRECISION A1,B1,ASS
      INTEGER IRX(7,300),IRY(7,300)
      INTEGER ISUMX(300),ISUMY(300),IBOUND(300),NB1
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      INTEGER MARGIN,NOUT,IJ,I,J,imark
      COMPLEX*16 ETA,ETAB,XI,XIB,AS,ASB,W
      COMPLEX*16 ED,EDB,EDE,EDBEB
      COMPLEX*16 Z
      COMPLEX*16 ELE,ELEB,ELD,IM
      INTRINSIC COS,SIN,DSQRT,ACOS
      LOGICAL LSYM
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/BDOP/OX,OY,IRX,IRY,ISUMX,ISUMY,IBOUND,NB1,NB2
      COMMON/RKCOM/RK
      COMMON/RACOM/RC,AA,EN
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/CMZ/Z(0:4000)
      COMMON/SIGC/SIG(0:4000)
      COMMON/ILCOM/IL
      COMMON/IUCOM/IU
      COMMON/IMARKC/IMARK
      EXTERNAL J06YAF,J06VJF,J06YGF
      EXTERNAL X01AAF,J06VAF,J06VCF
      EXTERNAL J06WAF,J06WBF,J06WCF,J06XFF,J06WZF
      EXTERNAL J06ABF,J06YAF,J06YCF,XXXXXX
      PARAMETER(MARGIN=1,NOUT=6,ZERO=0.0D0,
     + ONE=1.0D0,TWO=2.0D0,IM=(0.0D0,1.0D0))
C    ----------------------
       LSYM=.FALSE.
        if(il.eq.0)then
          OPEN(UNIT=17,FILE='aper0.ps3')
       else if(il.eq.iu)then
          OPEN(UNIT=17,FILE='aperf.ps3')
      endif
      CALL J06VAF(1,NOUT)
      CALL XXXXXX
      CALL J06VCF(1,17)
      DX=0.1
      DY=0.1
        DATA XMIN,XMAX,YMIN,YMAX/0.1,0.5,-0.2,0.2/
         DATA UMIN,UMAX,VMIN,VMAX/0.0,0.7,0.0,0.7/
      CALL J06WAF
      CALL J06WBF(XMIN,XMAX,YMIN,YMAX,MARGIN)
      CALL J06WCF(UMIN,UMAX,VMIN,VMAX)
      CALL J06ABF(DX,DY)
      CALL J06XFF(2)
        SIZE=ABS(XMAX-XMIN)/100.0e0
        call J06YJF(size)
      PPP=0.0d0
c    for the symmetry
      do 220 ll=1,2
      NB=NB1
      DO 150 ID=1,NB
       CALL BDER(ID,XL,YL)
       IJ=IBOUND(ID)
C     --------------
      ED=Z(IJ)
      EDB=CONJG(ED)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      EDBEB=CONJG(EDE)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
       if(il.eq.0)then
       ELE=(E*DSIN(G)-ETAB*(ONE+E*DCOS(G)))/
     + (ONE-E*DCOS(G)-E*ETAP*DSIN(G)+ETA2*(ONE+E*DCOS(G)))
       endif
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.ZERO)then
      write(6,*)'****in gra, dsqrt(pp),pp is negative.***',ij,pp
C     stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      EX=EXP(SIG(IJ))
      KL=-EX*((EN+ONE)*ETA2+EN-ONE)
      K=AA+KL
      W=TWO*ETA*EX+K/XIB
      rw=dreal(w)
      if(ll.eq.2)w=-w
      rwb=dreal(-im*w)
         A1=RW
 B1=RWB
  CALL J06YAF(A1,B1)
          CALL J06YGF(imark)
150   CONTINUE
```

```
220   CONTINUE
      do 320 ii=1,2
      DO 300 J=0,NY-1
      do 300 I=IMIN(J)+1,IMAX(J)-1
      IJ=IR(I,J)
      CALL INDER(IJ,LSYM,XL,YL,XXL,YYL,XYL)
      ED=Z(IJ)
      EDB=CONJG(ED)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      EDBEB=CONJG(EDE)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
       if(il.eq.0)then
       ELE=(E*DSIN(G)-ETAB*(ONE+E*DCOS(G)))/
     + (ONE-E*DCOS(G)-E*ETAP*DSIN(G)+ETA2*(ONE+E*DCOS(G)))
       endif
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.ZERO)then
      write(6,*)'****in gra, dsqrt(pp),pp is negative.***',ij,pp
C     stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      EX=EXP(SIG(IJ))
      KL=-EX*((EN+ONE)*ETA2+EN-ONE)
      K=AA+KL
      W=TWO*ETA*EX+K/XIB
      rw=real(w)
      if(ii.eq.2)w=-w
      rwb=real(-im*w)
        X1=RW
        Y1=RWB
        CALL J06YAF(X1,Y1)
        CALL J06YGF(imark)
300   continue
320   continue
       CALL J06WZF
       CLOSE(UNIT=17)
      return
      END
c    ************************************
      SUBROUTINE coneg
      IMPLICIT DOUBLE  PRECISION (A-H,O-Z)
      DOUBLE PRECISION UMIN,UMAX,VMIN,VMAX,XMIN,XMAX,YMIN,YMAX
      DOUBLE PRECISION X1,Y1,SIZE
      INTEGER MARGIN,NOUT,IMARK,NG
      COMPLEX*16 Z(0:4000),IM
      COMMON/NGC/NG,NG1,NG2
      COMMON/IMARKC/IMARK
      COMMON/CMZ/Z
      EXTERNAL J06YAF,J06VJF,J06YGF
      EXTERNAL X01AAF,J06VAF,J06VCF
      EXTERNAL J06WAF,J06WBF,J06WCF,J06XFF,J06WZF
      EXTERNAL J06ABF,J06YAF,J06YCF,XXXXXX
      PARAMETER(MARGIN=1,NOUT=6,ZERO=0.0D0,
     + ONE=1.0D0,TWO=2.0D0,IM=(0.0D0,1.0D0))
C    ---------------------
          OPEN(UNIT=27,FILE='cone.ps3')
      CALL XXXXXX
      CALL J06VAF(1,NOUT)
      CALL J06VCF(1,27)
      DX=0.2
      DY=0.2
        DATA XMIN,XMAX,YMIN,YMAX/0.1,0.5,-0.2,0.2/
        DATA UMIN,UMAX,VMIN,VMAX/0.0,0.7,0.0,0.7/
      CALL J06WAF
      CALL J06WBF(XMIN,XMAX,YMIN,YMAX,MARGIN)
      CALL J06WCF(UMIN,UMAX,VMIN,VMAX)
      CALL J06ABF(DX,DY)
      CALL J06XFF(2)
       DO 40 LI=0,NG2
        X1=REAL(Z(LI))
        Y1=REAL(-IM*Z(LI))
        SIZE=ABS(XMAX-XMIN)/100.0e0
        call J06YJF(size)
        CALL J06YAF(X1,Y1)
        CALL J06YGF(imark)
40     continue
       CALL J06WZF
       CLOSE(UNIT=27)
      return
      END
c    ************************************
      SUBROUTINE TAPER
      IMPLICIT DOUBLE  PRECISION (A-H,O-Z)
      DOUBLE PRECISION UMIN,UMAX,VMIN,VMAX,XMIN,XMAX,YMIN,YMAX
      DOUBLE PRECISION X1,X2,Y1,Y2,AG,X,DELX,RA,D,RA2
      INTEGER MARGIN,NOUT,I,J,L
      COMMON/WOR/RA,WO
      EXTERNAL X01AAF,J06VAF,J06VCF
      EXTERNAL J06WAF,J06WBF,J06WCF,J06XFF,J06WZF
      EXTERNAL J06ABF,J06YAF,J06YCF,XXXXXX
      EXTERNAL FUU,FVV,INDER
      PARAMETER(ZERO=0.0D0,ONE=1.0D0,TWO=2.0D0,FIVE=5.0D0,TEN=10.0D0)
C    ---------------------
      RA2=RA*RA
```

```
          OPEN(UNIT=47,FILE='taperg.ps3')
      CALL XXXXXX
      CALL J06VAF(1,NOUT)
      CALL J06VCF(1,47)
      DX=0.5
      DY=0.5
        DATA XMIN,XMAX,YMIN,YMAX/0.1,0.5,-0.2,0.2/
         DATA UMIN,UMAX,VMIN,VMAX/0.0,0.7,0.0,0.7/
      CALL J06WAF
      CALL J06WBF(XMIN,XMAX,YMIN,YMAX,MARGIN)
      CALL J06WCF(UMIN,UMAX,VMIN,VMAX)
      CALL J06ABF(DX,DY)
      CALL J06XFF(2)
      DO 10 L=1,4
       IF(1.EQ.1)D=0.0d0
       IF(1.EQ.2)D=TWO
       IF(1.EQ.3)D=FIVE
       IF(1.EQ.4)D=TEN
       AG=D*LOG(TEN)/(TEN*RA2)
       DELX=0.01
      X=ZERO
      J=RA*100
      DO 50 I=1,J+3
      IF(I.EQ.1)THEN
        X1=ZERO
        Y1=EXP(-AG*X1*X1)
      ELSE
       X=X+DELX
       IF(X.GT.RA)X=RA
       X2=X
       Y2=EXP(-AG*X2*X2)
           CALL J06YAF(X1/RA,Y1)
           CALL J06YCF(X2/RA,Y2)
       X1=X2
       Y1=Y2
      ENDIF
50    CONTINUE
10    CONTINUE
           CALL J06YAF(one,0.0d0)
           CALL J06YCF(one,one)
        CALL J06WZF
        CLOSE(UNIT=47)
      return
      END
C     ************
      SUBROUTINE CPAPERG
c  cross-polar position on the aperture
      IMPLICIT DOUBLE  PRECISION (A-H,O-Z)
      DOUBLE PRECISION OX(7,300),OY(7,300)
      DOUBLE PRECISION K,KL,EX,SIG,XL,YL,RW,RWB,AA
      DOUBLE PRECISION UMIN,UMAX,VMIN,VMAX,XMIN,XMAX,YMIN,YMAX
      DOUBLE PRECISION SQ,ETAP,ETA2
      DOUBLE PRECISION size,x1,y1
      DOUBLE PRECISION A1,B1,ASS
      INTEGER IRX(7,300),IRY(7,300)
      INTEGER ISUMX(300),ISUMY(300),IBOUND(300),NB1
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      INTEGER MARGIN,NOUT,IJ,I,J,imark,jmark,ijmin
      COMPLEX*16 ETA,ETAB,XI,XIB,AS,ASB,W
      COMPLEX*16 ED,EDB,EDE,EDBEB
      COMPLEX*16 Z
      COMPLEX*16 ELE,ELEB,ELD,IM
      INTRINSIC COS,SIN,DSQRT,ACOS
      LOGICAL LSYM
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/BDOP/OX,OY,IRX,IRY,ISUMX,ISUMY,IBOUND,NB1,NB2
      COMMON/RKCOM/RK
      COMMON/RACOM/RC,AA,EN
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/CMZ/Z(0:4000)
      COMMON/SIGC/SIG(0:4000)
      COMMON/IMARKC/IMARK
      COMMON/cpijm/ijmax
      COMMON/t12min/ijmin
      EXTERNAL J06YAF,J06VJF,J06YGF
      EXTERNAL X01AAF,J06VAF,J06VCF
      EXTERNAL J06WAF,J06WBF,J06WCF,J06XFF,J06WZF
      EXTERNAL J06ABF,J06YAF,J06YCF,XXXXXX
      PARAMETER(MARGIN=1,NOUT=6,ZERO=0.0D0,
     + ONE=1.0D0,TWO=2.0D0,IM=(0.0D0,1.0D0))
C     ---------------------
      jmark=2
       LSYM=.FALSE.
          OPEN(UNIT=15,FILE='cpaper.ps')
      CALL J06VAF(1,NOUT)
      CALL XXXXXX
      CALL J06VCF(1,15)
      DX=0.1
      DY=0.1
        DATA XMIN,XMAX,YMIN,YMAX/0.1,0.5,-0.2,0.2/
         DATA UMIN,UMAX,VMIN,VMAX/0.0,0.7,0.0,0.7/
      CALL J06WAF
      CALL J06WBF(XMIN,XMAX,YMIN,YMAX,MARGIN)
      CALL J06WCF(UMIN,UMAX,VMIN,VMAX)
      CALL J06ABF(DX,DY)
      CALL J06XFF(2)
        SIZE=ABS(XMAX-XMIN)/100.0e0
        call J06YJF(size)
      PPP=0.0d0
c    for the symmetry
      do 220 ll=1,2
      NB=NB1
      DO 150 ID=1,NB
       CALL BDER(ID,XL,YL)
       IJ=IBOUND(ID)
      ED=Z(IJ)
      EDB=CONJG(ED)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      EDBEB=CONJG(EDE)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.ZERO)then
      write(6,*)'****in gra, dsqrt(pp),pp is negative.***',ij,pp
C     stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      EX=EXP(SIG(IJ))
      KL=-EX*((EN+ONE)*ETA2+EN-ONE)
      K=AA+KL
      W=TWO*ETA*EX+K/XIB
      rw=dreal(w)
      if(ll.eq.2)w=-w
      rwb=dreal(-im*w)
         A1=RW
 B1=RWB
  CALL J06YAF(A1,B1)
          CALL J06YGF(imark)
150   CONTINUE
c      write(6,*)'ijwmax=', ijwmax, '    wbmax', wbmax
220   CONTINUE
      do 320 ii=1,2
      DO 300 J=0,NY-1
      do 300 I=IMIN(J)+1,IMAX(J)-1
           if(ii.eq.2)then
            IJ=IR(I,-J)
           else
            IJ=IR(I,J)
           endif
      CALL INDER(IJ,LSYM,XL,YL,XXL,YYL,XYL)
      ED=Z(IJ)
      EDB=CONJG(ED)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      EDBEB=CONJG(EDE)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.ZERO)then
      write(6,*)'****in gra, dsqrt(pp),pp is negative.***',ij,pp
C     stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      EX=EXP(SIG(IJ))
      KL=-EX*((EN+ONE)*ETA2+EN-ONE)
      K=AA+KL
      W=TWO*ETA*EX+K/XIB
      rw=real(w)
      rwb=real(-im*w)
        X1=RW
        Y1=RWB
        CALL J06YAF(X1,Y1)
       if(ij.eq.ijmax)then
        CALL J06YGF(jmark)
       else
           CALL J06YGF(imark)
       endif
300   continue
320   continue
       CALL J06WZF
       CLOSE(UNIT=15)
      return
      END
c    *********************************
      SUBROUTINE transg
c  t12 position on the aperture
      IMPLICIT DOUBLE  PRECISION (A-H,O-Z)
      DOUBLE PRECISION OX(7,300),OY(7,300)
      DOUBLE PRECISION K,KL,EX,SIG,XL,YL,RW,RWB,AA
      DOUBLE PRECISION UMIN,UMAX,VMIN,VMAX,XMIN,XMAX,YMIN,YMAX
      DOUBLE PRECISION SQ,ETAP,ETA2
      DOUBLE PRECISION size,x1,y1
      DOUBLE PRECISION A1,B1,ASS
      INTEGER IRX(7,300),IRY(7,300)
      INTEGER ISUMX(300),ISUMY(300),IBOUND(300),NB1
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      INTEGER MARGIN,NOUT,IJ,I,J,imark,ijmin,lmark
      COMPLEX*16 ETA,ETAB,XI,XIB,AS,ASB,W
      COMPLEX*16 ED,EDB,EDE,EDBEB
      COMPLEX*16 Z
      COMPLEX*16 ELE,ELEB,ELD,IM
      INTRINSIC COS,SIN,DSQRT,ACOS
      LOGICAL LSYM
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/BDOP/OX,OY,IRX,IRY,ISUMX,ISUMY,IBOUND,NB1,NB2
      COMMON/RKCOM/RK
      COMMON/RACOM/RC,AA,EN
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/CMZ/Z(0:4000)
      COMMON/SIGC/SIG(0:4000)
      COMMON/IMARKC/IMARK
      COMMON/cpijm/ijmax
      COMMON/t12min/ijmin
      EXTERNAL J06YAF,J06VJF,J06YGF
      EXTERNAL X01AAF,J06VAF,J06VCF
      EXTERNAL J06WAF,J06WBF,J06WCF,J06XFF,J06WZF
      EXTERNAL J06ABF,J06YAF,J06YCF,XXXXXX
      PARAMETER(MARGIN=1,NOUT=6,ZERO=0.0D0,
     + ONE=1.0D0,TWO=2.0D0,IM=(0.0D0,1.0D0))
C    ---------------------
       lmark=1
       LSYM=.FALSE.
          OPEN(UNIT=35,FILE='mint12.ps3')
      CALL J06VAF(1,NOUT)
      CALL XXXXXX
      CALL J06VCF(1,35)
      DX=0.1
      DY=0.1
        DATA XMIN,XMAX,YMIN,YMAX/0.1,0.5,-0.2,0.2/
        DATA UMIN,UMAX,VMIN,VMAX/0.0,0.7,0.0,0.7/
      CALL J06WAF
      CALL J06WBF(XMIN,XMAX,YMIN,YMAX,MARGIN)
      CALL J06WCF(UMIN,UMAX,VMIN,VMAX)
      CALL J06ABF(DX,DY)
      CALL J06XFF(2)
        SIZE=ABS(XMAX-XMIN)/100.0e0
        call J06YJF(size)
      PPP=0.0d0
c    for the symmetry
      do 220 ll=1,2
      NB=NB1
      DO 150 ID=1,NB
       CALL BDER(ID,XL,YL)
       IJ=IBOUND(ID)
      ED=Z(IJ)
      EDB=CONJG(ED)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      EDBEB=CONJG(EDE)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.ZERO)then
      write(6,*)'****in gra, dsqrt(pp),pp is negative.***',ij,pp
C     stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      EX=EXP(SIG(IJ))
      KL=-EX*((EN+ONE)*ETA2+EN-ONE)
      K=AA+KL
      W=TWO*ETA*EX+K/XIB
      rw=dreal(w)
      if(ll.eq.2)w=-w
      rwb=dreal(-im*w)
         A1=RW
 B1=RWB
  CALL J06YAF(A1,B1)
          CALL J06YGF(imark)
150   CONTINUE
c      write(6,*)'ijwmax=', ijwmax, '    wbmax', wbmax
220   CONTINUE
      do 320 ii=1,2
      DO 300 J=0,NY-1
      do 300 I=IMIN(J)+1,IMAX(J)-1
           if(ii.eq.2)then
            IJ=IR(I,-J)
           else
            IJ=IR(I,J)
           endif
      CALL INDER(IJ,LSYM,XL,YL,XXL,YYL,XYL)
      ED=Z(IJ)
      EDB=CONJG(ED)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      EDBEB=CONJG(EDE)
      ELD=(XL-IM*YL)/TWO
      ELE=EDE*ELD
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.ZERO)then
      write(6,*)'****in gra, dsqrt(pp),pp is negative.***',ij,pp
C     stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      EX=EXP(SIG(IJ))
      KL=-EX*((EN+ONE)*ETA2+EN-ONE)
      K=AA+KL
      W=TWO*ETA*EX+K/XIB
      rw=real(w)
      rwb=real(-im*w)
        X1=RW
        Y1=RWB
        CALL J06YAF(X1,Y1)
          if(ij.eq.ijmin)then
           CALL J06YGF(lmark)
          else
           CALL J06YGF(imark)
         endif
300   continue
320   continue
       CALL J06WZF
       CLOSE(UNIT=35)
      return
      END
c    ****************************************************
      SUBROUTINE GPD
C     general power density  analytica sol. w1,wnnx/(wnx-w1)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION ETA2,XI2,XI4,XI4K,AA,A3,BB,B3,A31,G
      DOUBLE PRECISION K,KL,V1,V2,V3,V,T1,T11,P2,Q2,QP2,EI,EN
      DOUBLE PRECISION E,E1,E2,E3,YC,PT1,PT2,PT ,XMI,XMA,YMI,YMA
      DOUBLE PRECISION L,PP,SQ,EX,ASS,DOMIN,GOMEGA(0:4000),W2(0:4000)
      DOUBLE PRECISION WX1,WX2,GY1,GY2,WMID,GMAX,DW
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70),nx
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70),IAPEX
C     ----------------------------------------
      COMPLEX*16 XC,XCB,CP,CPB,CQ,CQB
      COMPLEX*16 SEE,SEEB,SEBEB
      COMPLEX*16 EPS,EPSB,A1,A2,B1,B2
      COMPLEX*16 AS,ASB
      COMPLEX*16 KE,KEB
      COMPLEX*16 ELE,ELEB,ELEE,ELEBEB
      COMPLEX*16 P,PB,Q,QB
      COMPLEX*16 ETA,ETAB,XI,XIB
      COMPLEX*16 ED,EDB,EDE,EDBEB
      COMPLEX*16 IM,Z
      COMPLEX*16 W(0:4000),WB(0:4000)
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/NGC/NG,NG1,NG2
      COMMON/INPT/G,TETAC,PH,EI,E0
      COMMON/RACOM/R,A,EN
      COMMON/SIGC/SIG(0:4000)
      COMMON/CMZ/Z(0:4000)
      COMMON/wcom2/w,wb
      INTRINSIC COS,SIN,EXP
      EXTERNAL  J06VAF,J06VCF,J06VBF,J06YRF
      EXTERNAL J06WAF,J06WBF,J06WCF,J06XFF,J06WZF
      EXTERNAL J06ABF,J06YAF,J06YCF,XXXXXX
C     ----------------------------------------
      PARAMETER(ZERO=0.0D0,ONE=1.0D0,TWO=2.0D0,FOUR=4.0D0,
     + IM=(0.0D0,1.0D0),NOUT=6,MARGIN=1)
C     ----------------------
      OPEN(UNIT=483,FILE='powerdg.ps1')
      CALL J06VAF(1,NOUT)
      CALL XXXXXX
      CALL J06VCF(1,483)
      DO 100 IJ=0,NG2
      ED=Z(IJ)
      EDB=CONJG(ED)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      EDBEB=CONJG(EDE)
       ELE=(EI*SIN(G)-ETAB*(ONE+EI*COS(G)))/
     + (ONE-EI*COS(G)-EI*ETAP*SIN(G)+ETA2*(ONE+EI*COS(G)))
      ELEB=CONJG(ELE)
      ELEE=ELE*ELE
      ELEEB=-(ONE+EI*COS(G))/
     + (ONE-EI*COS(G)-EI*ETAP*SIN(G)+ETA2*(ONE+EI*COS(G)))+ELE*ELEB
      ELEBEB=ELEB*ELEB
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
       if(pp.lt.ZERO)then
       write(6,*)'****in gra, dsqrt(pp),pp is negative.***',ij,pp
C      stop
       endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      XI2=XI*XIB
      E1=EN*(ONE+ETA2)*(ONE+XI2)
      E2=(ONE+ETA*XIB)*(ONE+ETAB*XI)
      E3=(XI-ETA)*(XIB-ETAB)
      E=E1-E2+E3
      L=SIG(IJ)
      EX=EXP(L)
      KL=-EX*((EN+ONE)*ETA2+EN-ONE)
      K=A+KL
      W(ij)=TWO*ETA*EX+K/XIB
      WB(ij)=CONJG(W)
      KE=-TWO*ETAB*EXP(L)/(ONE+ETA2)
      KEB=CONJG(KE)
      SEE=ELE*ELE
      SEEB=(ONE-EN*EN)*(ONE+XI2)**2/E**2
      SEBEB=CONJG(SEE)
      CP=TWO*EXP(L)*(ONE+ETA*ELE)+KE/XIB
      CPB=TWO*EXP(L)*(ONE+ETAB*ELEB)+KEB/XI
      CQ=TWO*ETA*ELEB*EXP(L)+KEB/XIB
      CQB=TWO*ETAB*ELE*EXP(L)+KE/XI
      P=-TWO*(EN+ONE)*(XIB-ETAB)**2/(E*E)
      PB=-TWO*(EN+ONE)*(XI-ETA)**2/(E*E)
      Q=TWO*(EN-ONE)*(ONE+XI*ETAB)**2/(E*E)
      QB=TWO*(EN-ONE)*(ONE+XIB*ETA)**2/(E*E)
      EPS=-IM*(ONE+E0*ETAB)/(ONE+E0*ETA)
      EPSB=CONJG(EPS)
      A1=(XIB-ETAB)*(ONE+XI*ETAB)
      A2=(XI-ETA)*(ONE+XIB*ETA)
      A31=(XI-ETA)*(XIB-ETAB)*(ONE+XIB*ETA)*(ONE+XI*ETAB)
      A3=TWO*DSQRT(A31)
      AA=IM*(A1*EPSB-A2*EPS)/A3
      B1=(XI-ETA)*(ONE+XIB*ETA)
      B2=(XIB-ETAB)*(ONE+XI*ETAB)
      B3=A3
      BB=(EPS*B1+EPSB*B2)/B3
      PT1=TWO*(ETA*XIB+ETAB*XI)+(ETA2-ONE)*(XI2-ONE)
      PT2=(ETA2+ONE)*(XI2+ONE)
      PT=PT1/PT2
      T1=TWO*EN*(EN-PT)/(EN*EN-ONE)
      T11=T1/PT
      XI4=XI2*XI2
      XI4K=XI4/K**2
      P2=P*PB
      Q2=Q*QB
      QP2=Q2-P2
      V1=FOUR*(EN*PT-ONE)*(AA*AA*T1*T1+BB*BB*T11*T11)
      V2=QP2*XI4K
      V3=EN*(EN-PT)*(ONE+ETA2)**2
      V=V1*V2/V3
      XC=(P*CQB*XI**2+Q*CP*XIB**2)/K
      XCB=(PB*CQ*XIB**2+QB*CPB*XI**2)/K
      YC=(P*CPB*XI**2+Q*CQ*XIB**2)/K
      DOMIN=(ELEE-SEE-XC)*(ELEBEB-SEBEB-XCB)-(ELEEB-SEEB-YC)**2
      GOMEGA(IJ)=-V/DOMIN
      W2(IJ)=real(w(ij))
100   CONTINUE
      close(UNIT=13)
      GMAX=GOMEGA(IAPEX)
      WMID=(W2(NX)+W2(0))/TWO
      DW=W2(NX)-W2(0)
      WX1=-(W2(IAPEX)-W2(0))/DW
      GY1=GOMEGA(0)

      DX=0.5
      DY=0.5
      XMI=WX1
      XMA=-(W2(IAPEX)-W2(NX))/DW
      YMI=GY1/GMAX
      YMA=GOMEGA(NX)/GMAX
      XMIN=-0.8
      XMAX=0.8
       YMIN=-0.05
      YMAX=1.55
      DATA UMIN,UMAX,VMIN,VMAX/0.0,0.8,0.0,0.8/
      CALL J06WAF
      CALL J06WBF(XMIN,XMAX,YMIN,YMAX,MARGIN)
      CALL J06WCF(UMIN,UMAX,VMIN,VMAX)
      CALL J06ABF(DX,DY)
      CALL J06XFF(2)
      OPEN(UNIT=13,FILE='powerd.d1')
      DO 200 ID=1,NX
       GY2=GOMEGA(ID)
       WX2=-(W2(IAPEX)-W2(ID))/DW
       CALL J06YAF(WX1,GY1/GMAX)
       CALL J06YCF(WX2,GY2/GMAX)
       WX1=WX2
       GY1=GY2
       WRITE(13,10)id,WX1,GY1
10     format(i3,' &  ',f7.4,' &  ',f7.4,' &  \\hline')
200   CONTINUE
        CALL J06YRF(2)
        CALL J06YAF(XMI,zero)
        CALL J06YCF(XMI,YMI)
        CALL J06YAF(XMA,zero)
        CALL J06YCF(XMA,YMA)
        CALL J06WZF
        CLOSE(UNIT=482)
        CLOSE(UNIT=13)
      RETURN
      END
c    ********************************************
      SUBROUTINE ANAPERG
c  analytical aperture bound garph
      IMPLICIT DOUBLE  PRECISION (A-H,O-Z)
      DOUBLE PRECISION OX(7,300),OY(7,300)
      DOUBLE PRECISION K,KL,EX,SIG,RW,RWB,AA
      DOUBLE PRECISION UMIN,UMAX,VMIN,VMAX,XMIN,XMAX,YMIN,YMAX
      DOUBLE PRECISION SQ,ETAP,ETA2
      DOUBLE PRECISION size
      DOUBLE PRECISION W2NX,W20,WIA,DW
      DOUBLE PRECISION A1,B1,ASS
      INTEGER IRX(7,300),IRY(7,300)
      INTEGER ISUMX(300),ISUMY(300),IBOUND(300),NB1
      INTEGER IMIN(0:70),IMAX(0:70),IR(0:70,-40:40),JMAX(70)
      INTEGER IB(0:70),IRB(0:70),IBIND(0:70),IRBM(0:70)
      INTEGER MARGIN,NOUT,IJ,imark
      COMPLEX*16 ETA,ETAB,XI,XIB,AS,ASB,W(0:4000)
      COMPLEX*16 ED,EDB,EDE,EDBEB
      COMPLEX*16 Z
      COMPLEX*16 ELE,ELEB,IM
      INTRINSIC COS,SIN,DSQRT,ACOS
      COMMON/CMGRID/NX,NY,IMIN,IMAX,JMAX,IR,IAPEX,IBMAX,IB,IRB,IRBM,
     1        IBIND
      COMMON/BDOP/OX,OY,IRX,IRY,ISUMX,ISUMY,IBOUND,NB1,NB2
      COMMON/RKCOM/RK
      COMMON/RACOM/RC,AA,EN
      COMMON/INPT/G,TETAC,PH,E,E0
      COMMON/CMZ/Z(0:4000)
      COMMON/SIGC/SIG(0:4000)
      COMMON/IMARKC/IMARK
      EXTERNAL J06YAF,J06VJF,J06YGF
      EXTERNAL  J06VAF,J06VCF
      EXTERNAL J06WAF,J06WBF,J06WCF,J06XFF,J06WZF
      EXTERNAL J06ABF,J06YAF,J06YCF,XXXXXX
      PARAMETER(MARGIN=1,NOUT=6,ZERO=0.0D0,
     + ONE=1.0D0,TWO=2.0D0,IM=(0.0D0,1.0D0))
C    -----------------------
          OPEN(UNIT=117,FILE='anaper.ps1')
      CALL J06VAF(1,NOUT)
      CALL XXXXXX
      CALL J06VCF(1,117)
      DX=0.1
      DY=0.1
      DATA XMIN,XMAX,YMIN,YMAX/-0.1,1.5,-0.8,0.8/
      DATA UMIN,UMAX,VMIN,VMAX/0.0,0.8,0.0,0.8/
      CALL J06WAF
      CALL J06WBF(XMIN,XMAX,YMIN,YMAX,MARGIN)
      CALL J06WCF(UMIN,UMAX,VMIN,VMAX)
      CALL J06XFF(2)
      SIZE=ABS(XMAX-XMIN)/100.0d0
      call J06YJF(size)
      PPP=0.0d0
      DO 150 ID=1,NB2
      IJ=IBOUND(ID)
      ED=Z(IJ)
      EDB=CONJG(ED)
      ETA=(E0-ED)/(ONE+E0*ED)
      ETAB=CONJG(ETA)
      ETAP=ETA+ETAB
      ETA2=ETA*ETAB
      EDE=-(ONE+ED*E0)**2/(ONE+E0*E0)
      EDBEB=CONJG(EDE)
        ELE=(E*SIN(G)-ETAB*(ONE+E*COS(G)))/
     +  (ONE-E*COS(G)-E*ETAP*SIN(G)+ETA2*(ONE+E*COS(G)))
```

```
      ELEB=CONJG(ELE)
      AS=ETA+(ONE+ETA2)*ELEB
      ASB=CONJG(AS)
      ASS=AS*ASB
      PP=ONE-(EN*EN-ONE)*ASS
      if(pp.lt.ZERO)then
      write(6,*)'****in gra, dsqrt(pp),pp is negative.***',ij,pp
      stop
      endif
      SQ=DSQRT(PP)
      XI=((EN+ONE)*ETA*ELE-(EN-ONE)*ETAB*ELEB+ONE-SQ)/
     +((EN+ONE)*ELE+(EN-ONE)*ETAB*(ONE+ETAB*ELEB))
      XIB=CONJG(XI)
      EX=EXP(SIG(IJ))
      KL=-EX*((EN+ONE)*ETA2+EN-ONE)
      K=AA+KL
      W(IJ)=TWO*ETA*EX+K/XIB
150   CONTINUE
      w2nx=real(w(nx))
      w20=real(w(0))
      wia=real(w(iapex))
      DW=W2NX-W20
      DO 160 ID=1,NB2
      IJ=IBOUND(ID)
      rw=real(w(ij))
      rwb=real(-im*w(ij))
         A1=-(wia-RW)/dw
 B1=RWB/dw
 CALL J06YAF(A1,B1)
         CALL J06YGF(imark)
160   CONTINUE
       CALL J06WZF
       CLOSE(UNIT=117)
      return
      END
c    *************************
```

References

Ames, W. F., 1965, *Nonlinear partial differential equations in engineering* (Academic Press, New York).

Brickell, F., Marder, L., and Westcott, B. S., 1977, The geometrical optics design of reflectors using complex coordinates *J. Phys. A: Math. Gen.* **10**, pp. 245-260.

Brickell, F., and Westcott, B. S., 1978, Phase and power density distributions on plane aperture of reflector antenna, *J. Phys. A: Math. Gen.* **11**, pp. 777-789.

Deschamps, G. A., 1972, Ray techniques in electromagnetics, *Proc. IEEE*, **60**, pp 1022-1035.

Galindo-Israel, V., Imbriale, W. and Mittra, R., 1987, On the theory of the synthesis of single and dual offset shaped reflector antennas, *IEEE Trans. Antennas and Proagation*, **AP-35**, pp.887-896.

Galindo-Israel, V. and Mittra, R., 1984, Synthesis of offset dual shaped subreflector antennas for control of Cassegrain aperture distributions, *IEEE Trans. Antennas and Proagation*, **AP-32**, pp.86-92.

von Hoerner, S., 1978, Minimum noise maximum gain telescopes and relaxation method for shaped asymmetric surfaces, *IEEE Trans. Antennas and*

Proagation, **AP-26**, pp.464-471.

Kildal, P. S., 1990, Synthesis of multireflector antennas by kinematic and dynamic ray tracing, *IEEE Trans. Antennas and Propagation*, **AP-38**, pp.1587-1599.

Lee, J. J., Parad, L. I. and Chu, R. S., 1979, A shaped offset-fed dual reflector antenna, *IEEE Trans. Antennas and General*, **AP-26**, pp.165-171.

Lier, E. 1990, Broad-band elliptical-beammshape horn with low cross-polarisation, *IEEE Trans. Antennas Propagat.*, **vol 38**, pp. 800-805.

Lier, E. and Aas, J. A., 1985, Simple hybried-mode horn feed loaded with a dielectric cone, *Electron. Lett.*, **vol. 21**, pp.563-564.

Lier, E., Rahmat-Samii, Y. and Rengarajan, 1991, Application of rectangular and ellitical feed horns to elliptical reflector antennas, *IEEE Trans. Antennas Propagat.*, **vol 39**, pp. 1592-1597.

Lier, E., E., and Skyttemyr, S. A., 1991, A compact shaped single-reflector offset antenna with low cross-polarization, *IEEE, APS*, pp. 682-685.

Marder, L., 1981, Uniqueness in reflector mappings and the Monge-Ampère equation, *Proc. Roy. Soc. London*, **A378**, pp. 529-537.

Mehler, M., 1986, and Tun, S., Adatia, N., Direct far-field GO synthesis of shaped- beam reflector antenna, *IEE Proceedings*, **133, Pt. H**, pp.213-220.

Olver, A. D., Clarricoats, P. J. B. and Voglis, E. E., 1984, Study of dual-offset reflector antenna with dielectric cone feed, *IEE Proceedings*, **131, Pt. H**, pp. 69-76.

Voglis, E. E. and Olver, A. D., 1985, Shaped dual-offset antenna with dielectric cone feed for DBS reception, *IEE Proc.*, Part H, **132**, pp. 110-114.

Rall, L. B., 1969, *Computational solution of nonlinear operator equations* (Wiley, New York).

Rusch, W. V. T., and Potter, P. D., 1970, *Analysis of Reflector Antennas*, (Academic press, New York, London).

Schruben, J. S.,1972, Formulation of a reflector-design problem for a lighting fixture, *J. Opt.Soc Amer.*, **62**, pp.1498-1501.

Silver, S., 1949, *Microwave Antenna Theory and Design*, (New York: MacGraw-Hill, Radiation Lab. Series).

Westcott, B. S., 1983, *Shaped reflector antenna design* (Letchworth: Research Studies Press).

Westcott, B. S., 1993, On the synthesis of a single offset reflector with dielectric cone feed *J. Modern Optics*, **40**, pp. 1161-1173.

Westcott, B. S., and Brickell, F., 1986, General dielectric lens shaping using complex coordinates *IEE Proceedings*, **133**, pp. 122-126.

Westcott, B. S., Brickell, F. and Wolton, I. C., 1987, Synthesis and design of shaped dual reflector antennas from specified aperture amplitud and phase distributions, *preprint series*, **174**, University of Southampton, Faculty of Mathematical Studies.

Westcott, B. S., Graham, R. K., and Wolton, I. C., 1986, Synthesis of offset dual offset shaped reflectors for arbitrary aperture shapes using continuous domain deformation, *IEE Proc.*, **133**, pp. 57-64.

Westcott, B. S., Norris, A. P., 1975, Reflector synthesis for generalized farfields, *J. Phys. A: Math. Gen.*, **8**, pp. 521-532.

Westcott, B. S., Stevens, F. A., and Brickell, F., 1981, GO synthesis of offset dual reflector, *IEE Proc. H, Microwaves, Optics and Antennas*, **128**, pp. 11-18.