To the Graduate Council:

I am submitting herewith a thesis written by Eva W. Kam entitled "Analyses of Mode Structures in Optical Waveguides Excited by Specific Spatial Intensity Distributions." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

M. J. Roberts

Michael J. Roberts, Major Professor

We have read this thesis
and recommend its acceptance:

Herbert K. Ruff Jr.

Accepted for the Council:

Ewrinkel

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Eva Kaw

Date May, 1989

ANALYSES OF MODE STRUCTURES IN OPTICAL WAVEGUIDES EXCITED BY SPECIFIC SPATIAL INTENSITY DISTRIBUTIONS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Eva W. Kam

May 1989

ACKNOWLEDGMENTS

The author would like to thank everyone who has helped on this thesis. Special thanks to my major advisor, Dr. Roberts, for his advice, support and patience throughout my thesis work. Thanks is extended to Drs. Neff and Hoffman for serving on my committee and helping me on the thesis. Special thanks to my parents and family for their great support. Finally, the author would also like to thank Julian Chan for his support, patience and encouragement towards the completion of this thesis.

ABSTRACT

The flow of electromagnetic energy in slab dielectric waveguides is studied. Boundary conditions are fitted at one end of a waveguide which acts as the source of input power (spatial intensity distribution) to excite the waveguides. The electric field of the wave which propagates along the direction of the axis of the waveguides is studied. The study begins by considering the single slab waveguide. Several input waveforms are used and the propagation patterns along the waveguide are observed. After gaining an understanding of the waveguiding properties in the single slab waveguide, coupling effects between parallel waveguides are studied. This study should provide a better understanding of the flow of energy along waveguides being exicted by different energy sources.

TABLE OF CONTENTS

CHAPTER		PAGE
1. INTRODUCTION		**1**
1.1	Motivation	1
1.2	Description of the problem	2
2. MATHEMATICAL DEVELOPMENT		**4**
2.1	Understanding mode propagation in slab dielectric waveguides	4
2.1.1	Single slab waveguide	4
2.1.2	Parallel slab waveguides	8
2.2	Fitting boundary conditions	16
3. RESULTS		**19**
3.1	Single slab waveguide	19
3.1.1	Plane wave	19
3.1.2	Pencil beam	23
3.1.3	Focused wave	32
3.2	Parallel slab waveguides	43
3.2.1	Plane wave	43
3.2.2	Focused wave	47
4. CONCLUSION		**53**
BIBLIOGRAPHY		**56**
APPENDIX		**58**

CHAPTER PAGE

VITA 169

LIST OF FIGURES

FIGURE		PAGE
1.1	Single slab dielectric waveguide.	3
1.2	Parallel slab dielectric waveguides.	3
2.1	Symmetrical slab dielectric waveguide with an input wave coming from $z < 0$.	5
2.2	Parallel slab dielectric waveguides with an input wave coming from $z < 0$.	8
3.1	A unit amplitude plane wave entering the single slab waveguide.	20
3.2	Electric field of guided modes for the single slab waveguide being excited by plane wave.	21
3.3	Electric field of guided and radiation modes for the single slab waveguide being excited by plane wave.	22
3.4	Electric fields at different z for the single slab waveguide being excited by plane wave.	24
3.5	Electric fields at $z = 0$ for the single slab waveguide being excited by plane wave with different number of radiation modes.	25
3.6	A unit amplitude plane wave entering the single slab waveguide at an angle.	26
3.7	Electric field at $z = 0$ for the single slab waveguide being excited by pencil beam.	28
3.8	Phase of the electric field at $z = 0$ for the single slab waveguide being excited by pencil beam.	29
3.9	Electric fields at different z for the single slab waveguide being excited by pencil beam.	30

FIGURE PAGE

3.10 Electric field at $z = 0$ and $\alpha = 0$ for the single slab waveguide being excited by pencil beam. 31

3.11 Lens against the single slab waveguide. 32

3.12 Magnitude and phase of the electric field at $z = 0$ for the single slab waveguide with lens against the slab. 34

3.13 Electric fields at different z for the single slab waveguide with lens against the slab. 35

3.14 Electric fields at different z, and $f = 0.1m$ for the single slab waveguide with lens against the slab. 36

3.15 Focused wave entering the single slab waveguide. 37

3.16 Electric field at $z' = 3\mu$ for the single slab waveguide being excited by focused wave. 38

3.17 Electric field at $z' = f$ for the single slab waveguide being excited by focused wave. 40

3.18 Electric fields at different z for the single slab waveguide being excited by focused wave. 41

3.19 Electric field at $z = 0$ and $f = 0.02m$ for the single slab waveguide being excited by focused wave. 42

3.20 A unit plane wave entering the parallel slab waveguides. 44

3.21 Electric field at $z = 0$ for the parallel waveguides being excited by plane wave. 45

3.22 Electric fields at different z for the parallel waveguides being excited by plane wave and the waveguides are separated by a distance of 4μ. 46

3.23 Electric fields at different z for the parallel waveguides being excited by plane wave and the waveguides are separated by a distance of 1μ. 48

FIGURE PAGE

3.24 Focused wave entering the parallel slab waveguides. 49

3.25 Electric field at $z = 0$ for the parallel waveguides being excited by focused wave. 51

3.26 Electric fields at different z for the parallel waveguides being excited by focused wave and the waveguides are separated by a distance of 1μ. . . . 52

CHAPTER 1

INTRODUCTION

1.1 Motivation

For the past twenty years, optical communications has been growing in importance in the field of communication. Optical waveguides are being used for the transmission of information. With low-loss optical fibres, information can be transmitted further than cables made with copper without much distortion or loss of information during transmission. Besides optical fibres, dielectric optical waveguides play an important role in the rapid growth of integrated optics.

Many books and papers have discussed guiding mechanisms, and transmission properties for different kinds of optical waveguides. They help in the understanding of the waveguiding properties of optical waveguides. But for anyone who is interested in optical communication system, it will also be very helpful and interesting to know how waves are propagated along the optical waveguides as well. In order to begin with the observation of wave propagation in optical waveguides, it will be best to start with the dielectric slab waveguides, the simplest optical waveguides. In this study, waves propagating along single and parallel dielectric waveguides are being studied. The wave being observed is excited by specific spatial intensity distribution set at the end of the slab waveguides.

1.2 Description of the problem

In order to obtain an understanding of the flow of electromagnetic energy guided by dielectric waveguides, this study begins by studying the waveguiding properties of symmetric single slab waveguide.

The single slab waveguide has the geometry shown in Figure 1.1. The assumption has been made that it is a two dimensional problem and the dielectric materials are infinite in the y direction. The objective is to have an input wave entering the end of the slab waveguide and observe how the wave propagates along medium two. Reflectors are included at some distances above and below the waveguide. The reason for including the reflectors will be explained later when fitting the boundary conditions.

After gaining an understanding of wave propagation along the single slab waveguide, it will be a logical move to study the parallel waveguides. The geometry for the parallel waveguides is shown in Figure 1.2. The objective is to have an input wave entering medium two and/or four, and observe how the wave propagates along the direction of the axis of the wavguides. In this way, the coupling effect between waveguides is studied also.

Chapter 2 is mainly devoted to the finding of the forms of electromagnetic waves that propagate along the single and parallel dielectric slab waveguides. Also, the method for fitting boundary conditions at the end of the waveguide is presented. Chapter 3 presents and analyzes the results obtained with different input waves entering the single and parallel slab waveguides. Chapter 4 discusses the work and gives suggestions for future research. Finally, the appendix will include all the simulation programs.

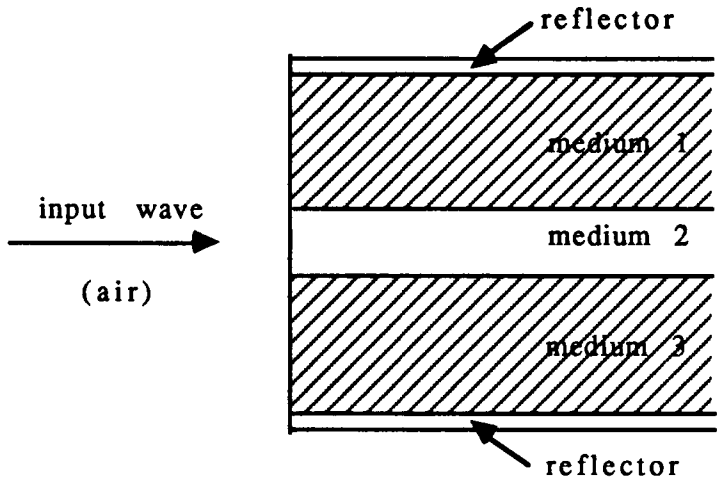


Figure 1.1: Single slab dielectric waveguide.

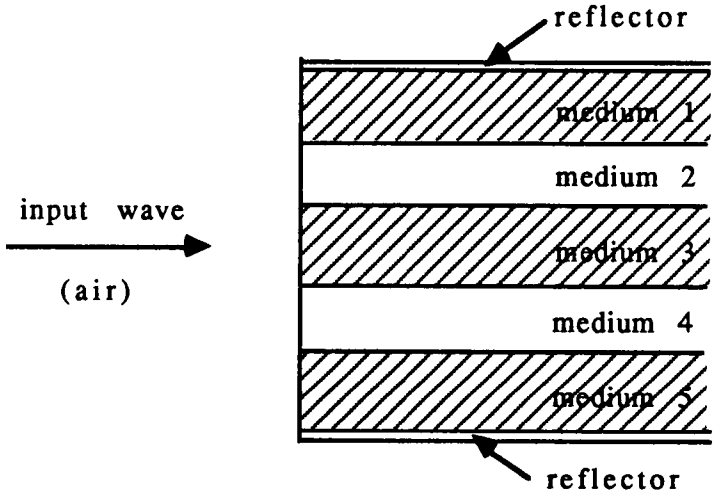


Figure 1.2: Parallel slab dielectric waveguides.

CHAPTER 2

MATHEMATICAL DEVELOPMENT

2.1 Understanding mode propagation in slab dielectric waveguides

In order to study how waves propagate in the slab waveguides, we need a knowledge of the properties of the modes which can exist in slab dielectric waveguides. Mainly, there are two different kind of modes, guided and radiation modes. Guided modes consist of waves confined mostly inside the slab , and radiation modes exist both inside and outside of the slab.

According to Yariv in chapter 13 [1], each mode of a dielectric waveguide is a solution of the wave equation

$$\frac{\partial}{\partial x^2}E(x,z) + (k_0^2 n^2 - \beta^2)E(x,z) = 0$$

where $E(x,z)$ represents the electric field, β is the propagation constant, n is the refractive index of the dielectric waveguide, $k_0 = (\frac{2\pi}{\lambda_0})$ and λ_0 is the free space wavelength.

In our study, the slab waveguides are to be infinite in the y direction. We will study wave propagation (along the axis of the waveguides) by looking at the electric field, E_y (that is, normal to the direction of propagation) only.

2.1.1 Single slab waveguide

A symmetrical single slab waveguide is studied and it has the geometry shown in Figure 2.1.

First, the guided modes are considered. There are two kinds of guided modes, even

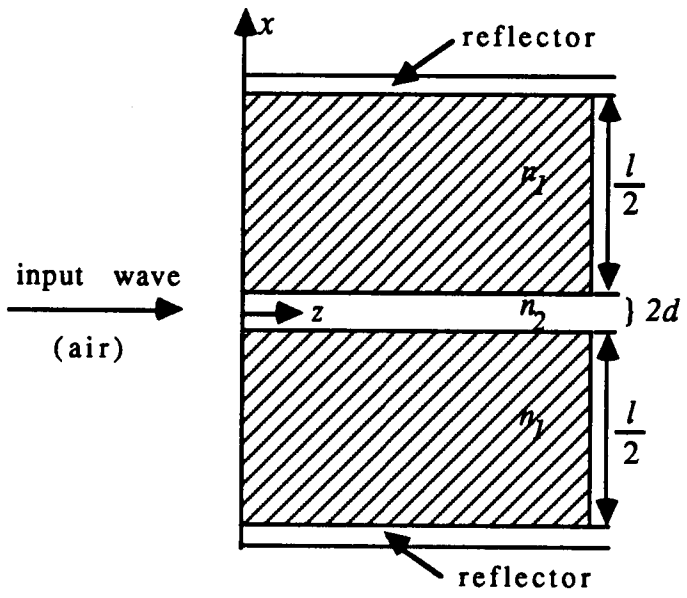


Figure 2.1: Symmetrical slab dielectric waveguide with an input wave coming from $z < 0$.

and odd. Yariv gives the form of the electric field (E_y) for the even and odd modes for the case where the reflectors are placed at infinity or no reflectors at all. Although in our case we have a finite value for l, we can still use the form of the electric field given by Yariv because we are going to choose l such that the electric field is extremely small (close to zero) when it reaches the reflectors.

In the case of even modes, the electric field has the form

$$E_y(x,z) = \begin{cases} Ae^{-p(|x|-d)-jk_z z} & |x| \geq d \\ B\cos(hx)e^{-jk_z z} & |x| \leq d \end{cases} \tag{2.1}$$

where A, B are arbitrary constants and the relationships between A, B, h, p and k_z are given below

$$A = B\cos(hd) \tag{2.2}$$

$$pd = hd\tan(hd) \tag{2.3}$$

$$k_z^2 = (\frac{2\pi}{\lambda_0})^2 n_2^2 - h^2 \tag{2.4}$$

$$(pd)^2 + (hd)^2 = (n_2^2 - n_1^2)(\frac{2\pi}{\lambda_0})^2 d^2 \tag{2.5}$$

where B is chosen to be unity for our calculation.

In order to calculate how many and what kind of even modes exist, we need to solve equations 2.3 through 2.5 to find a set of p and h values for each even mode exists. After that, the electric field at different x and z can be calculated according to equation 2.1.

In the case of odd modes, the electric field has the form

$$E_y(x,z) = \begin{cases} Ae^{-p(|x|-d)-jk_z z} & |x| \geq d \\ B\sin(hx)e^{-jk_z z} & |x| \leq d \end{cases} \tag{2.6}$$

where k_z is the same as for the even modes. Similarly, the relationships between A, B, h, p and k_z are

$$A = B\sin(hd) \tag{2.7}$$

$$pd = -hd\cot(hd) \tag{2.8}$$

and equation 2.5 also applys to the odd modes. Thus, for calculating the electric field for the odd modes at different x and z, we need to solve equations 2.5, 2.6 and 2.8.

After considering the guided modes, the radiation modes are studied. There are also even and odd radiation modes. Lee, in chapter 4 [2], gave the electric field (E_y) solutions for the radiation modes where $x > 0$. The electric field for $x \leq 0$ can be obtained by symmetry. The electric field has the form

$$E_y(x,z) = \begin{cases} A_2\{-\frac{k_{2x}}{k_{1x}}\sin(k_{2x}d - \psi)\sin[k_{1x}(x-d)] + \\ \quad \cos(k_{2x}d - \psi)\cos[k_{1x}(x-d)]\}e^{-jk_z z} & d < x < d + \frac{l}{2} \\ A_2\cos(k_{2x}x - \psi)e^{-jk_z z} & |x| \leq d \\ A_2\{\frac{k_{2x}}{k_{1x}}\sin(k_{2x}d + \psi)\sin[k_{1x}(x+d)] + \\ \quad \cos(k_{2x}d + \psi)\cos[k_{1x}(x+d)]\}e^{-jk_z z} & -(d + \frac{l}{2}) < x < -d \end{cases} \tag{2.9}$$

where A_2 is chosen to be unity for our calculation, and relationships between k_{1x}, k_{2x}, and k_z are given below

$$\Delta k^2 = \frac{2\pi}{\lambda_0}\sqrt{n_2^2 - n_1^2}$$

$$k_{2x}^2 = k_{1x}^2 + \Delta k^2 \tag{2.10}$$

$$k_z^2 = (\frac{2\pi n_2}{\lambda_0})^2 - k_{2x}^2 \tag{2.11}$$

For even and odd modes, $\psi = 0$ and π respectively, and the following condition must be met

$$x \cot(\frac{xl}{2d}) = \sqrt{x^2 + \nu^2} \tan(\sqrt{x^2 + \nu^2} - \psi) \tag{2.12}$$

where

$$x = k_{1x}d$$

$$\nu = \Delta k d$$

When equation 2.12 is solved, k_{1x}, k_{2x}, k_z can be calculated with equations 2.10 and 2.11. Once all these values are found, $E_y(x, z)$ at different x and z can be calculated.

From equation 2.12, we can see the difficulty in solving for x when l/d is getting larger and larger, since solutions for x are getting closer together. As l approaches infinity, a solution exists for every x. This will create problems when fitting the boundary condition, and that is the reason for using the reflectors. This problem will be discussed in detail when methods are presented for fitting the boundary condition.

2.1.2 Parallel slab waveguides

A symmetrical parallel slab waveguide problem is studied and it has a geometry shown in Figure 2.2. The assumption is made that n_1 and n_5 are smaller than n_2, n_3 and n_4. If n_1 and n_5 are larger than n_2, n_3 and n_4, leaky waves occur, which is beyond the scope

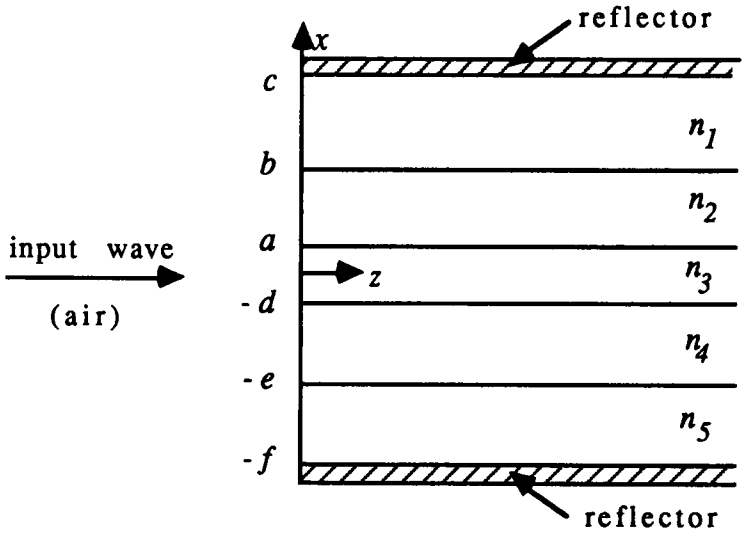


Figure 2.2: Parallel slab dielectric waveguides with an input wave coming from $z < 0$.

of this study because we are interested in studying the wave propagation guided along the slab waveguides in medium two or medium four.

In comparing to the single slab waveguide, the parallel slab waveguide has more possible forms of electric field due to the different combinations of the refractive indices.

First, the guided modes are considered. The electric field can be expressed as a combination of five different forms. Ideas for finding the forms of the electric field came from Lee, Kapany, Burke and Marcuse [2][3][4]. Each possible form of the electric field exists for $-\infty < x < \infty$, but with the same reason as for the single slab case, the values for c and f (location of the reflectors) will be chosen such that the electric field is approximately zero at the reflectors. Different forms for the electric field will be explained in detail as follows.

When n_1, n_5 are smaller than n_2, n_3 and n_4, the electric field has the form

$$E_y(x,z) = \begin{cases} A_1 e^{-h_1(x-b)} e^{-j\beta z} & x \geq b \\ A_2 \cos(h_2 x - \psi_2) e^{-j\beta z} & a \leq x \leq b \\ A_3 \cos(h_3 x - \psi_3) e^{-j\beta z} & -d \leq x \leq a \\ A_4 \cos(h_4 x - \psi_4) e^{-j\beta z} & -e \leq x \leq -d \\ A_5 e^{h_5(x+e)} e^{-j\beta z} & x \leq -e \end{cases} \tag{2.13}$$

where A_1 through A_5 are arbitrary constants, h's are the propagation constants of a given mode. A_1 is chosen to be unity in our calculation. The relationships between β, h's and A's are given below

$$\beta^2 = k_0^2 n_1^2 + h_1^2 \tag{2.14}$$

$$\beta^2 = k_0^2 n_2^2 - h_2^2 \tag{2.15}$$

$$\beta^2 = k_0^2 n_3^2 - h_3^2 \tag{2.16}$$

$$\beta^2 = k_0^2 n_4^2 - h_4^2 \tag{2.17}$$

$$\beta^2 = k_0^2 n_5^2 + h_5^2 \tag{2.18}$$

$$A_2 = \frac{A_1}{\cos(h_2 b - \psi_2)} \tag{2.19}$$

$$A_3 = \frac{A_2 \cos(h_2 a - \psi_2)}{\cos(h_3 a - \psi_3)} \tag{2.20}$$

$$A_4 = \frac{A_3 \cos(h_3 d + \psi_3)}{\cos(h_4 d + \psi_4)} \tag{2.21}$$

$$A_5 = A_4 \cos(h_4 e + \psi_4) \tag{2.22}$$

In order to solve for all the unknowns, the matching of the electric and magnetic fields at interfaces is needed. As a result of the matching, the following condition must be met. Let

$$L = \tan^{-1}\{\frac{h_2}{h_3} \tan[h_2(a-b) + \tan^{-1}(\frac{h_1}{h_2})]\}$$

and

$$\tan[h_3(a+d)-L] = \frac{h_4}{h_3}\tan[h_4(d-e)+\tan^{-1}(\frac{h_5}{h_4})] \tag{2.23}$$

With equations 2.23 and 2.14 through 2.18, all h's can be calculated. After that, all the A's can be calculated with equations 2.19 through 2.22. Once all these values are found, $E_y(x,z)$ can easily be found with equation 2.13. With this form of electric field, the parallel slab waveguides can be modeled as an asymmetric single slab waveguide by using same material for medium two, three, four and different materials for media one and five.

When $n_2 = n_4$ and larger than n_1, n_3 and n_5, the electric field has the forms

$$E_y(x,z) = \begin{cases} A_1 e^{-h_1(x-b)}e^{-j\beta z} & x \geq b \\ A_2\cos(h_2x-\psi_2)e^{-j\beta z} & a \leq x \leq b \\ A_3\cosh(h_3x)e^{-j\beta z} & -d \leq x \leq a \\ A_4\cos(h_4x-\psi_4)e^{-j\beta z} & -e \leq x \leq -d \\ A_5 e^{h_5(x+e)}e^{-j\beta z} & x \leq -e \end{cases} \tag{2.24}$$

and

$$E_y(x,z) = \begin{cases} A_1 e^{-h_1(x-b)}e^{-j\beta z} & x \geq b \\ A_2\cos(h_2x-\psi_2)e^{-j\beta z} & a \leq x \leq b \\ A_3\sinh(h_3x)e^{-j\beta z} & -d \leq x \leq a \\ A_4\cos(h_4x-\psi_4)e^{-j\beta z} & -e \leq x \leq -d \\ A_5 e^{h_5(x+e)}e^{-j\beta z} & x \leq -e \end{cases} \tag{2.25}$$

where A_1 is chosen to be unity as before and the relationships between β and h's for both cases are given below

$$\beta^2 = k_0^2 n_1^2 + h_1^2 \tag{2.26}$$

$$\beta^2 = k_0^2 n_2^2 - h_2^2 \tag{2.27}$$

$$\beta^2 = k_0^2 n_3^2 + h_3^2 \tag{2.28}$$

$$\beta^2 = k_0^2 n_4^2 - h_4^2 \tag{2.29}$$

$$\beta^2 = k_0^2 n_5^2 + h_5^2 \tag{2.30}$$

For the case with a hyperbolic cosine function in medium three, the A's and ψ's are related to the h's as follows

$$\psi_2 = h_2 b - \tan^{-1}(\frac{h_1}{h_2}) \tag{2.31}$$

$$\psi_4 = \tan^{-1}(\frac{h_5}{h_4}) - h_4 e \tag{2.32}$$

$$A_2 = \frac{A_1}{\cos(h_2 b - \psi_2)} \tag{2.33}$$

$$A_3 = \frac{A_2 \cos(h_2 a - \psi_2)}{\cosh(h_3 a)} \tag{2.34}$$

$$A_4 = \frac{A_3 \cosh(h_3 d)}{\cos(h_4 d + \psi_4)} \tag{2.35}$$

$$A_5 = A_4 \cos(h_4 e + \psi_4) \tag{2.36}$$

In order to solve for all the unknowns, matching of the electric and magnetic fields is needed. As a result of the matching, the following condition must be met.

$$h_3 \tanh(h_3 a) + h_2 \tan(h_2 a - \psi_2) =$$

$$h_3 \tanh(h_3 d) + h_4 \tan(h_4 d + \psi_4) \tag{2.37}$$

With equations 2.26 through 2.37, all the unknowns for each mode can be found and $E_y(x, z)$ in equation 2.24 can be found.

For the case with a hyperbolic sine function in medium three, equations 2.36 and 2.31 through 2.33 still hold. A_3 and A_4 will be given as follow

$$A_3 = \frac{A_2 \cos(h_2 a - \psi_2)}{\sinh(h_3 a)} \tag{2.38}$$

$$A_4 = -\frac{A_3 \sinh(h_3 d)}{\cos(h_4 d + \psi_4)} \tag{2.39}$$

As the result of matching the electric and magnetic fields, the following condition must be met.

$$h_3 \coth(h_3 a) + h_2 \tan(h_2 a - \psi_2) =$$

$$h_3 \coth(h_3 d) + h_4 \tan(h_4 d + \psi_4) \tag{2.40}$$

By similar procedures, all unknowns can be found by using equations 2.38 through 2.40 and 2.26 through 2.33. After that, $E_y(x,z)$ can easily be found for each mode.

When n_3 is larger than n_1, n_2, n_4 and n_5, the electric field has the form

$$E_y(x,z) = \begin{cases} A_1 e^{-h_1(x-b)} e^{-j\beta z} & x \geq b \\ A_2 e^{-h_2(x-a)} e^{-j\beta z} & a \leq x \leq b \\ (A_3 \cos(h_3 x) + B_3 \sin(h_3 x)) e^{-j\beta z} & -d \leq x \leq a \\ A_4 e^{h_4(x+d)} e^{-j\beta z} & -e \leq x \leq -d \\ A_5 e^{h_5(x+e)} e^{-j\beta z} & x \leq -e \end{cases} \tag{2.41}$$

where A_3 is chosen to be unity for our calculation and the relationships between β, B_3, h's and A's are given below

$$\beta^2 = k_0^2 n_1^2 + h_1^2 \tag{2.42}$$

$$\beta^2 = k_0^2 n_2^2 + h_2^2 \tag{2.43}$$

$$\beta^2 = k_0^2 n_3^2 - h_3^2 \tag{2.44}$$

$$\beta^2 = k_0^2 n_4^2 + h_4^2 \tag{2.45}$$

$$\beta^2 = k_0^2 n_5^2 + h_5^2 \tag{2.46}$$

$$A_1 = [A_3 \cos(h_3 a) + B_3 \sin(h_3 a)] e^{-h_2(b-a)} \tag{2.47}$$

$$A_2 = A_3 \cos(h_3 a) + B_3 \sin(h_3 a) \tag{2.48}$$

$$A_4 = A_3\cos(h_3 d) - B_3\sin(h_3 d) \tag{2.49}$$

$$A_5 = [A_3\cos(h_3 d) - B_3\sin(h_3 d)]e^{h_4(d-e)} \tag{2.50}$$

$$B_3 = \frac{-h_2\cos(h_3 a) + h_3\sin(h_3 a)}{h_3\cos(h_3 a) + h_2\sin(h_3 a)} \tag{2.51}$$

After matching the electric and magnetic fields at interfaces, the following condition must be met.

$$h_3[\sin(h_3 d) + B_3\cos(h_2 d)] = h_4[\cos(h_3 d) - B_3\sin(h_3 d)] \tag{2.52}$$

By similar procedures, all unknowns for each mode can be found with equations 2.42 through 2.52, and $E_y(x,z)$ can be found afterwards.

When n_2 is larger than n_1, n_3, n_4 and n_5, the electric field has the form

$$E_y(x,z) = \begin{cases} (A\cos(h_2 b) + B\sin(h_2 b))e^{-h_1(x-b)}e^{-j\beta z} & x \geq b \\ (A\cos(h_2 x) + B\sin(h_2 x))e^{-j\beta z} & a \leq x \leq b \\ (A\cos(h_2 a) + B\sin(h_2 a))e^{h_3(x-a)}e^{-j\beta z} & -d \leq x \leq a \\ (A\cos(h_2 a) + B\sin(h_2 a))e^{-h_3(a+d)}e^{h_4(x+d)}e^{-j\beta z} & -e \leq x \leq -d \\ (A\cos(h_2 a) + B\sin(h_2 a))e^{-h_3(a+d)}e^{h_4(d-e)}e^{-j\beta z} & x \leq -e \end{cases} \tag{2.53}$$

where A is chosen to be unity for our calculation and the relationships between β, B_3 and the h's are given below

$$\beta^2 = k_0^2 n_1^2 + h_1^2 \tag{2.54}$$

$$\beta^2 = k_0^2 n_2^2 - h_2^2 \tag{2.55}$$

$$\beta^2 = k_0^2 n_3^2 + h_3^2 \tag{2.56}$$

$$\beta^2 = k_0^2 n_4^2 + h_4^2 \tag{2.57}$$

$$\beta^2 = k_0^2 n_5^2 + h_5^2 \tag{2.58}$$

$$B = \frac{h_2\sin(h_2 b) - h_1\cos(h_2 b)}{h_1\sin(h_2 b) + h_2\cos(h_2 b)} \tag{2.59}$$

After matching the electric and magnetic fields, the following condition must be met.

$$-h_2[\sin(h_2a) - B\cos(h_2a)] = h_3[\cos(h_2a) + B\sin(h_2a)] \tag{2.60}$$

By similar procedures, all unknowns for each mode can be found with equations 2.54 through 2.60, and $E_y(x,z)$ can be found afterwards.

When n_4 is larger than n_1, n_2, n_3 and n_5, the electric field has the form

$$E_y(x,z) = \begin{cases} (A\cos(h_4d) - B\sin(h_4d))e^{-h_3(a+d)}e^{-h_2(b-a)}e^{-h_1(x-b)}e^{-j\beta z} & x \geq b \\ (A\cos(h_4d) - B\sin(h_4d))e^{-h_3(a+d)}e^{-h_2(x-a)}e^{-j\beta z} & a \leq x \leq b \\ (A\cos(h_4d) - B\sin(h_4d))e^{-h_3(x+d)}e^{-j\beta z} & -d \leq x \leq a \\ (A\cos(h_4x) + B\sin(h_4x))e^{-j\beta z} & -e \leq x \leq -d \\ (A\cos(h_4e) - B\sin(h_4e))e^{h_5(x+e)}e^{-j\beta z} & x \leq -e \end{cases} \tag{2.61}$$

where A is chosen to be unity for our calculation and the relationships between β, B_3 and the h's are given below

$$\beta^2 = k_0^2 n_1^2 + h_1^2 \tag{2.62}$$

$$\beta^2 = k_0^2 n_2^2 + h_2^2 \tag{2.63}$$

$$\beta^2 = k_0^2 n_3^2 + h_3^2 \tag{2.64}$$

$$\beta^2 = k_0^2 n_4^2 - h_4^2 \tag{2.65}$$

$$\beta^2 = k_0^2 n_5^2 + h_5^2 \tag{2.66}$$

$$B = \frac{h_4\sin(h_4d) + h_3\cos(h_4d)}{h_3\sin(h_4d) - h_4\cos(h_4d)} \tag{2.67}$$

After matching of the electric and magnetic fields, the following condition must be met.

$$h_4[\sin(h_4e) + B\cos(h_4e)] = h_5[\cos(h_4e) - B\sin(h_4e)] \tag{2.68}$$

By similar procedures, all unknowns for each mode can be found with equations 2.62 through 2.68, and $E_y(x,z)$ can be found afterwards.

After considering all the possible guided modes, radiation modes are studied. The form of electric field is very similar to the one shown in equation 2.13 except sinusoidal waveforms exist in media one and five instead of an exponential decaying waveform. The electric field has the form

$$E_y(x,z) = \begin{cases} A_1 \sin[h_1(x-b) - \psi_1]e^{-j\beta z} & x \geq b \\ A_2 \cos(h_2 x - \psi_2)e^{-j\beta z} & a \leq x \leq b \\ A_3 \cos(h_3 x - \psi_3)e^{-j\beta z} & -d \leq x \leq a \\ A_4 \cos(h_4 x - \psi_4)e^{-j\beta z} & -e \leq x \leq -d \\ A_5 \sin[h_5(x+e) + \psi_5]e^{-j\beta z} & x \leq -e \end{cases} \tag{2.69}$$

where A_1 is assumed to be unity for our calculation. The relationships between β, h's, ψ's and A's are given below

$$\beta^2 = k_0^2 n_1^2 - h_1^2 \tag{2.70}$$

$$\beta^2 = k_0^2 n_2^2 - h_2^2 \tag{2.71}$$

$$\beta^2 = k_0^2 n_3^2 - h_3^2 \tag{2.72}$$

$$\beta^2 = k_0^2 n_4^2 - h_4^2 \tag{2.73}$$

$$\beta^2 = k_0^2 n_5^2 - h_5^2 \tag{2.74}$$

$$A_2 = -\frac{A_1 \sin\psi_1}{\cos(h_2 b - \psi_2)} \tag{2.75}$$

$$A_3 = \frac{A_2 \cos(h_2 a - \psi_2)}{\cos(h_3 a - \psi_3)} \tag{2.76}$$

$$A_4 = \frac{A_3 \cos(h_3 d + \psi_3)}{\cos(h_4 d + \psi_4)} \tag{2.77}$$

$$A_5 = \frac{A_4 \cos(h_4 e + \psi_4)}{\sin\psi_5} \tag{2.78}$$

$$\psi_1 = h_1(c-b) \tag{2.79}$$

$$\psi_5 = h_5(f-e) \tag{2.80}$$

$$\psi_2 = h_2 b - \tan^{-1}[\frac{h_1}{h_2}\cot\psi_1] \tag{2.81}$$

$$\psi_4 = \tan^{-1}[\frac{h_5}{h_4}\cot\psi_5] - h_4 e \tag{2.82}$$

$$\psi_3 = h_3 a - \tan^{-1}[\frac{h_2}{h_3}\tan(h_2 a - \psi_2)] \tag{2.83}$$

When the electric and magnetic fields are matched, the following condition must be met.

$$h_4\tan[h_4(d-e) + \tan^{-1}(\frac{h_5}{h_4}\cot\psi_5)] = \tag{2.84}$$

$$h_3\tan\{h_3(a+d) - \tan^{-1}[\frac{h_2}{h_3}\tan(h_2 a - \psi_2)]\} \tag{2.85}$$

With the use of equations 2.70 through 2.85, all the unknowns for each mode can be found.

2.2 Fitting boundary conditions

After gaining an understanding of mode propagation in slab dielectric waveguides, study is made to fit the boundary condition which sets up by a known input wave (spatial intensity distribution) at the end of the slab dielectric waveguides. Refer to Figures 2.1 and 2.2. An input wave enters the slab and sets a boundary condition for the electric field at $z = 0$. The electric field, E_y (combination of weighted guided and radiation modes), at different z is calculated in order to observe how the wave propagates in the slab waveguides.

To find out the weights for each guided and radiation modes, we apply the idea of orthogonal functions as shown by Kreyszig in chapter 4 [5]. We can use the idea of orthogonal functions because the set of guided and radiation modes solutions is an

orthogonal set. Properties for orthogonal functions state that an arbitrary function, $f(x)$, can be written as a sum of weighted orthogonal functions $g_n(x)$ as follows,

$$f(x) = \sum_{n=1}^{\infty} c_n g_n(x) \tag{2.86}$$

In our case, $f(x)$ will be the boundary condition at $z = 0$, $g_n(x)$ will be the electric field $E_y(x, z)$ at $z = 0$ for the guided and radiation modes, and c_n are the constants which determine the weight for each guided and radiation mode that exists in order to satisfy the boundary condition. The constants, c_n, are given by

$$c_n = \frac{1}{\|g_m\|^2} \int_{a'}^{b'} f(x) g_m(x) dx \tag{2.87}$$

where $a' \leq x \leq b'$ is the interval where $f(x)$ exists, and

$$\|g_m\|^2 = \int_{c'}^{d'} g_m^2(x) dx$$

where $c' = -\infty$, $d' = \infty$ for guided modes and $c' = -\frac{l}{2}$, $d' = \frac{l}{2}$ for radiation modes.

There is one important point to be noticed. In order to fit the boundary condition, a finite number of guided modes exist but infinitely many radiation modes exist. Thus numerically, it is impossible to include all the radiation modes. In our study, a reasonable finite number of radiation modes is chosen to fit the boundary condition.

After c_n for each guided and radiation mode is calculated, the weighted sum should approximately equal to $f(x)$, the boundary condition at $z = 0$. Since n in equation 2.86 cannot go to infinity in actual calculations (as explained above), an exact solution cannot be obtained. When the boundary condition is satisfied within a reasonable range, we can study the wave propagation by looking at the total electric field $E_{y\mathbf{T}}(x, z)$, and it is given as

$$E_{y\mathbf{T}}(x, z) = \sum_{n=1}^{M} c_n g_n(x, z)$$

where M means the total number of guided and radiation modes used, $g_n(x,z)$ is the electric field for each guided and radiation mode and c_n is unchanged , they are still those constants calculated by equation 2.87 at $z = 0$.

CHAPTER 3

RESULTS

When techniques for finding the existing guided and radiation modes and fitting boundary condition are learned, several cases of spatial intensity distributions exciting the single slab and parallel slab waveguides are studied and results are observed.

3.1 Single slab waveguide

For the single slab waveguide, three different kinds of input wave entering the slab are studied and each of which will be explained in detail in the following.

3.1.1 Plane wave

A unit amplitude plane wave entering the slab. The geometry is shown in Figure 3.1 and the boundary condition is given as

$$f(x) = \begin{cases} 1 & |x| \leq d \\ 0 & \text{otherwise} \end{cases}$$

Assume at $z = 0$, all the wave is absorbed except between $x = -d$ and d.

Now, consider the case when $n_2 = 2$, $n_1 = 1.5$, $l = 19\mu$ and $d = 2\mu$. The magnitude of the electric field is plotted at different z. Figure 3.2 shows the electric field of all the weighted guided modes that exist, and Figure 3.3 shows the electric field with guided and radiation modes. In comparing these two figures, they show that most of the electric field strength is contained in the guided modes. When the radiation modes are added,

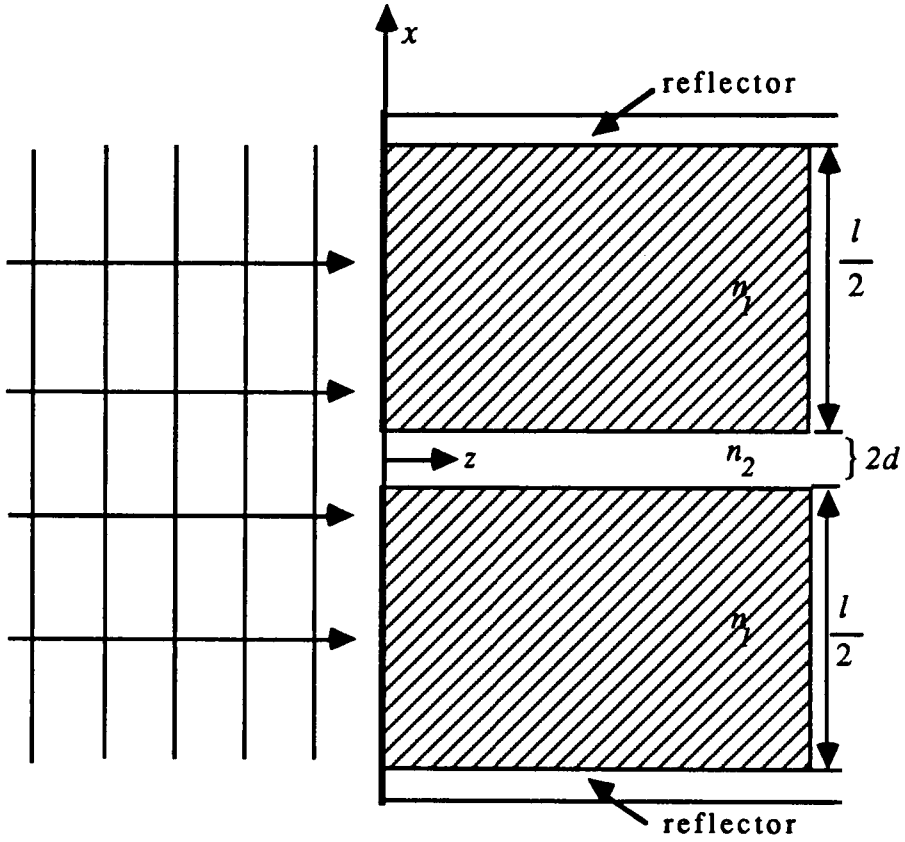


Figure 3.1: A unit amplitude plane wave entering the single slab waveguide.

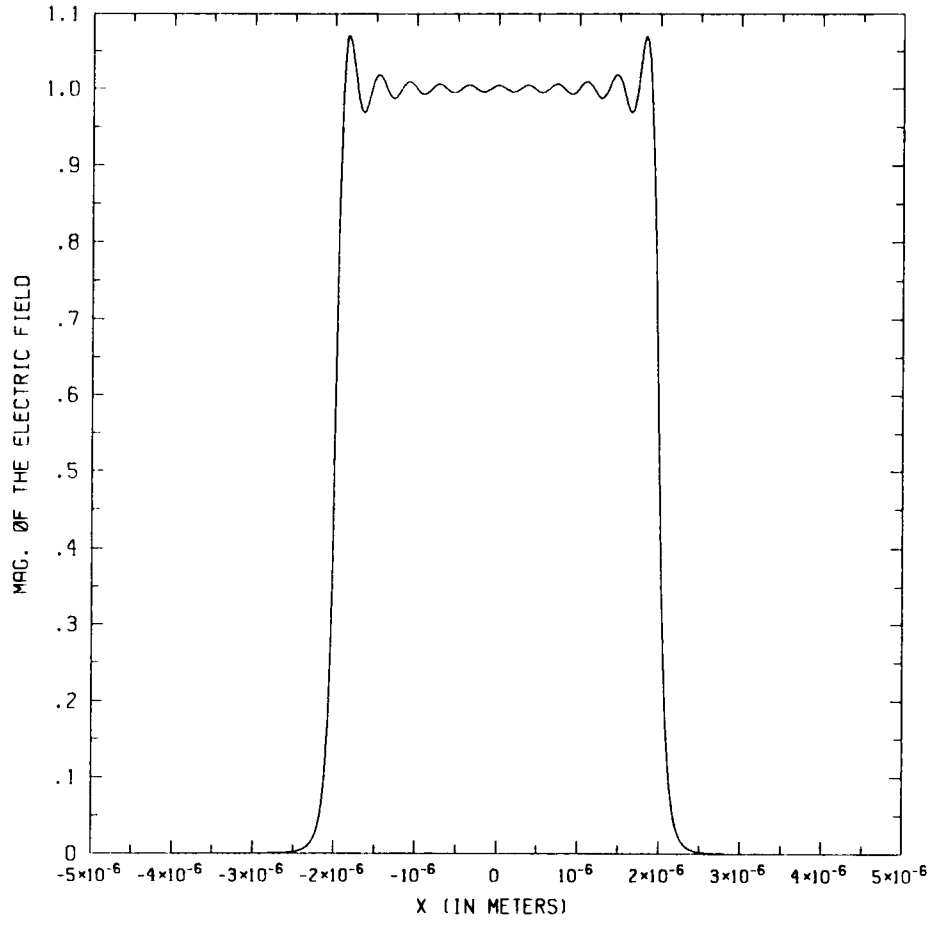


Figure 3.2: Electric field of guided modes for the single slab waveguide being excited by plane wave.

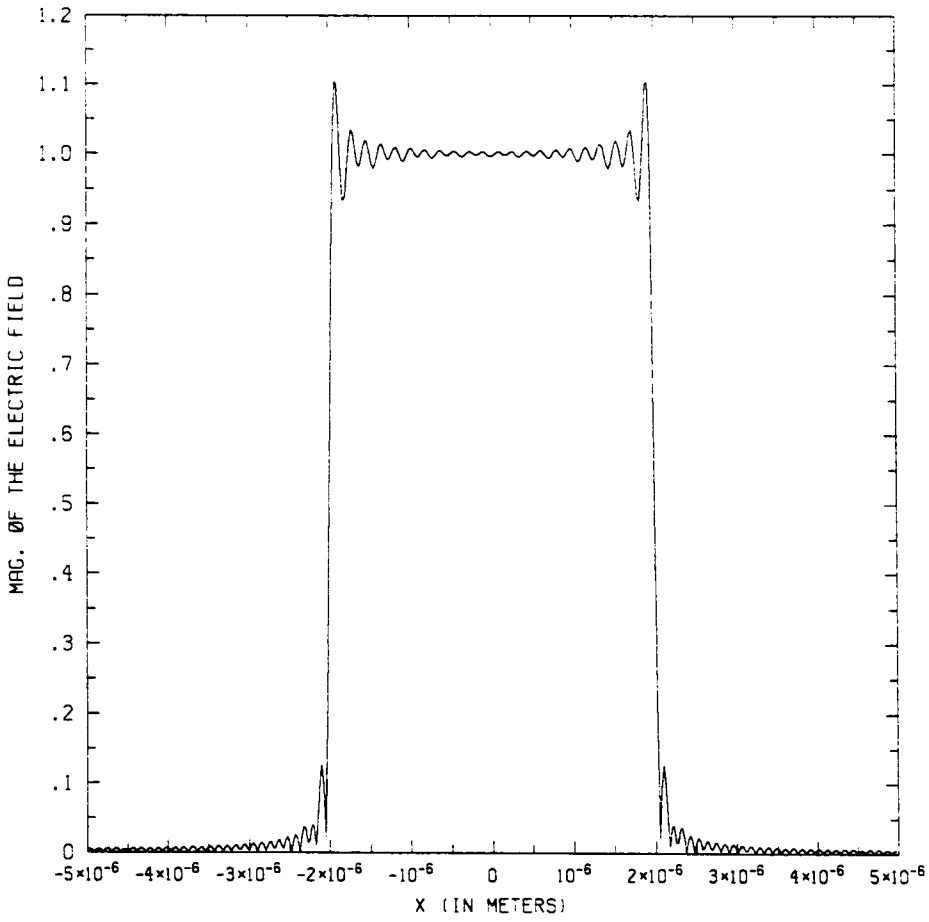


Figure 3.3: Electric field of guided and radiation modes for the single slab waveguide being excited by plane wave.

the boundary condition is fitted very well except some overshoots around the boundaries between different media. The propagation pattern of the wave is shown in Figure 3.4. As z increases, the strength of the electric field begins to concentrate at the centre of the core, but when z increases further, the electric field strength is concentrated at the boundaries between the core and the surrounding medium. The reason for this is because each guided mode travels at a different speed and thus the electric fields are added up in phase or out of phase at different z.

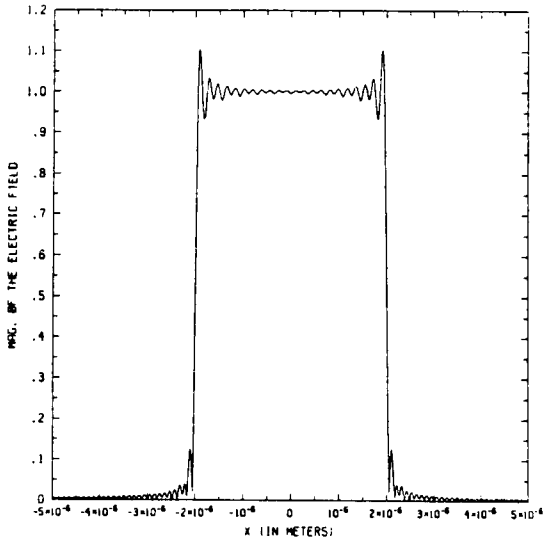
Next, a case is considered when $n_2 = 2, n_1 = 1.9999, l = 16\mu$, $d = 2\mu$. Only one even guided mode exists. This implies that most of the electric field strength is contained in the radiation modes and Figure 3.5 can prove this. The figure shows that more radiation modes are needed in order to fit the boundary condition, which in turn means that the computing time required is increased and creates a disadvantage in using the orthogonal function approximation. But, since an analytical solution cannot be obtained, using this approximation can obtain a quite reasonable result except at certain situations such as the one discussed above.

3.1.2 Pencil beam

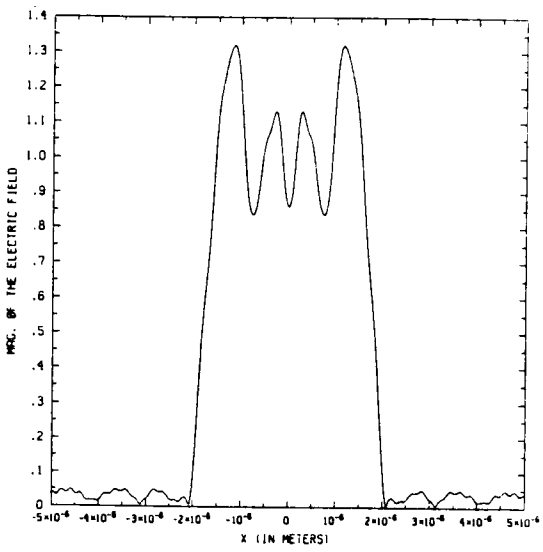
A unit amplitude plane wave entering the slab at an angle, α. The geometry is shown in Figure 3.6 and the boundary condition is

$$f(x) = \begin{cases} e^{j\frac{2\pi \sin\alpha}{\lambda_0}x} & a' \leq x \leq b' \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

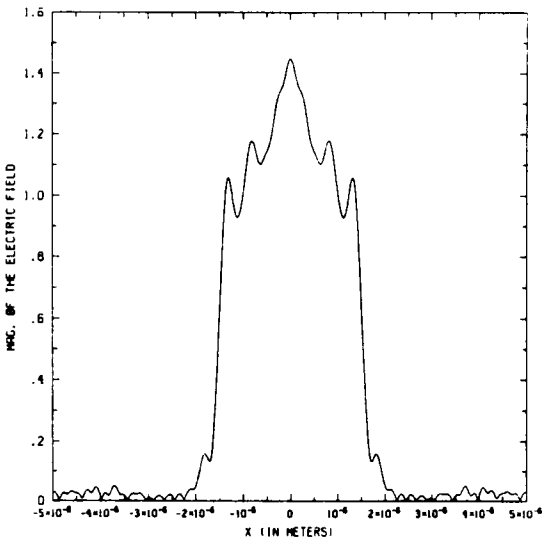
A case is studied when $a' = -20\mu$, $b' = -16\mu$, $n_2 = 2$, $n_1 = 1.5$, $l = 168\mu$, $d = 20\mu$, and α is chosen to be -40^0 (greater than the critical angle). The reason for choosing this set of values is to create a pencil beam to enter the slab and the wave propagating in the slab should be total internally reflected. In order for the wave to be total internally



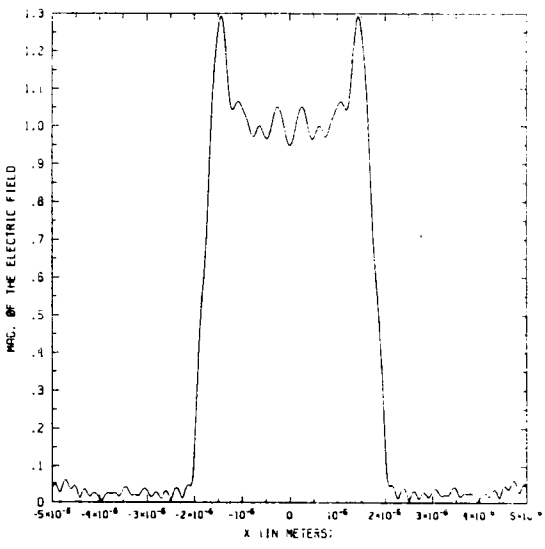
(a) z = 0 μ



(b) z = 4 μ

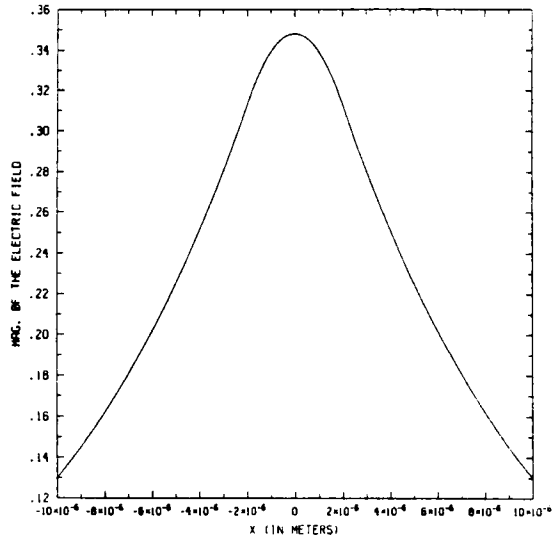


(c) z = 16 μ

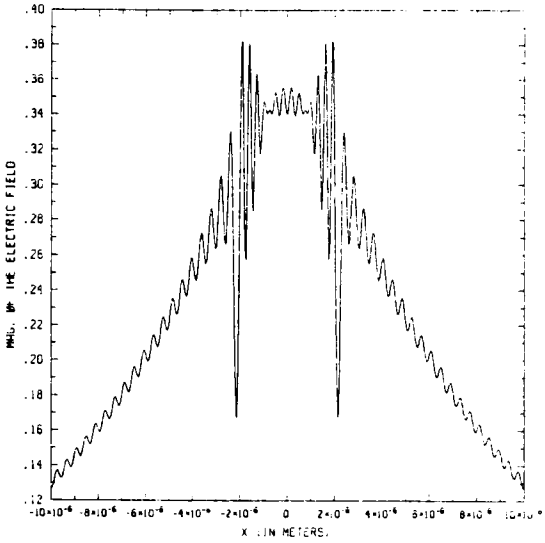


(d) z = 64 μ

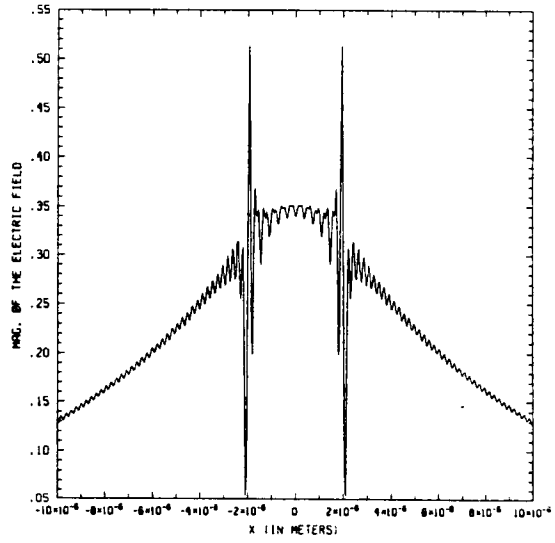
Figure 3.4: Electric fields at different z for the single slab waveguide being excited by plane wave.



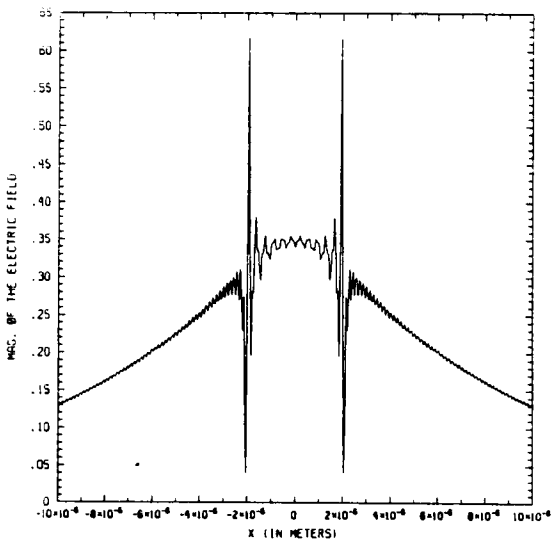
(a) Guided modes only



(b) 100 radiation modes



(c) 200 radiation modes



(d) 300 radiation modes

Figure 3.5: Electric fields at $z = 0$ for the single slab waveguide being excited by plane wave with different number of radiation modes.

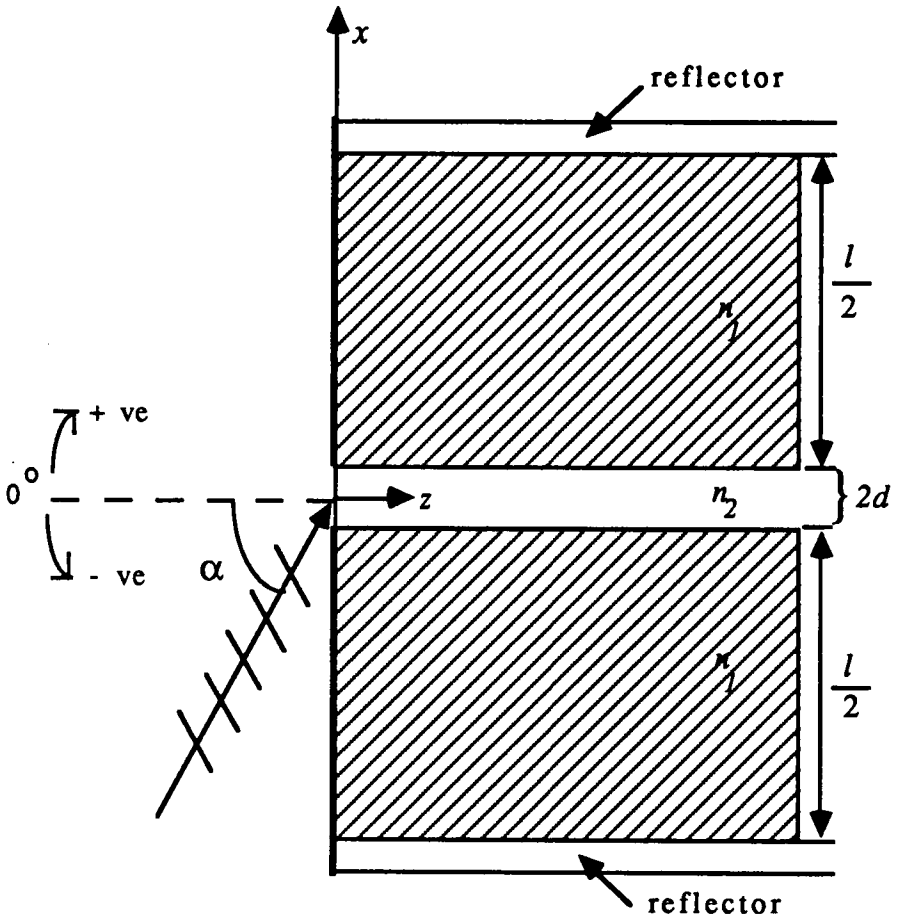


Figure 3.6: A unit amplitude plane wave entering the single slab waveguide at an angle.

reflected, the value for the critical angle is needed. The critical angle, θ_c, is calculated by using Snell's law

$$\theta_c = \sin^{-1}(\frac{n_1}{n_2})$$

Due to the geometry of this set up shown in Figure 3.6, α has to be less than $(90^0 - \theta_c)$ in order for the wave to be total internal reflected.

In Figure 3.7, the electric field at $z = 0$ is plotted and it shows that the boundary condition is satisfied reasonably in magnitude. The phase should be linear for $a' \leq x \leq b'$ as shown by $f(x)$ in equation 3.1 and it can be proved by Figure 3.8. The wave propagating behaviour is observed by calculating electric fields at different z, and they are shown in Figure 3.9. From these figures, the electric field strength shifted in position across the slab and which agrees with the prediction that the wave bounces back and forth between the boundaries of the core. But, one should notice that the electric field disperses. The reason for the dispersion may due to the fact that the radiation modes are bounced back at the reflectors and the electric field for each mode added up in phase or out of phase. Although dispersion occurs, the electric field strength shifted in position across the slab as predicted.

When $\alpha = 0^0$, it is just like having a unit amplitude wave entering the slab as the case studied in section 3.1. In order to prove that, a plane wave entering the slab at $\alpha = 0^0$ is compared with the previous case when a plane wave is entering the slab perpendicularly under the same conditions except the angle of incidence is not considered. This argument is proved by comparing Figures 3.10 and 3.3, they appear to be identical.

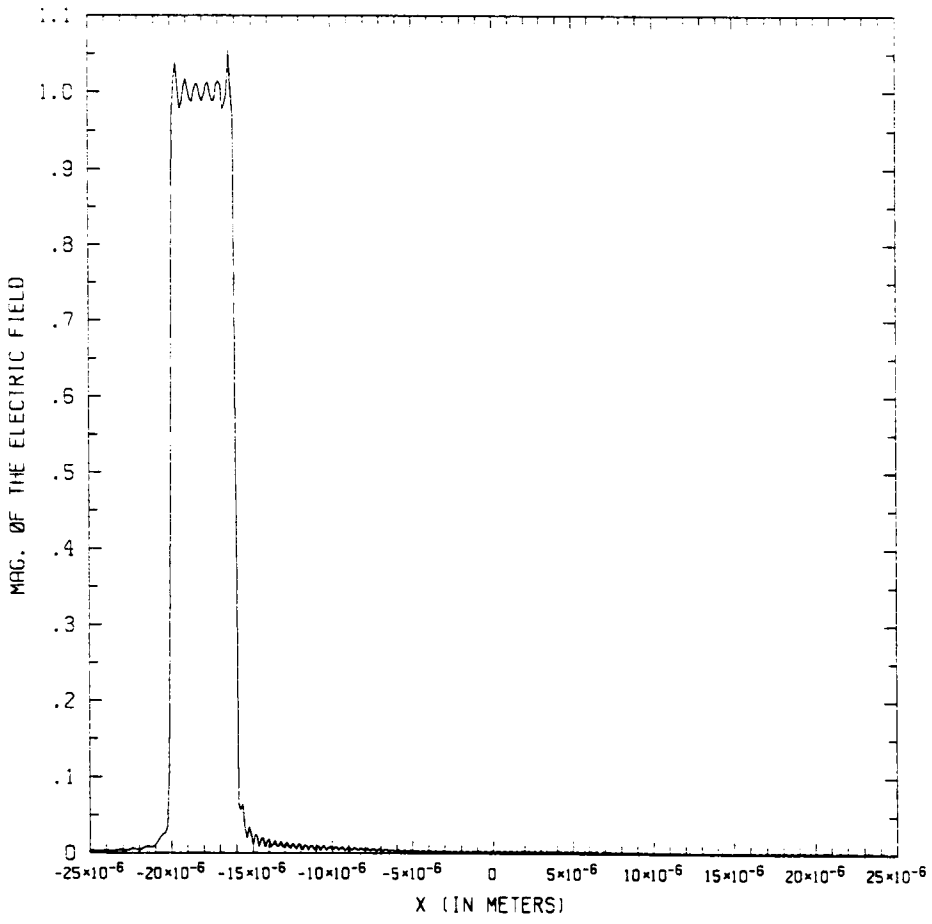


Figure 3.7: Electric field at $z = 0$ for the single slab waveguide being excited by pencil beam.

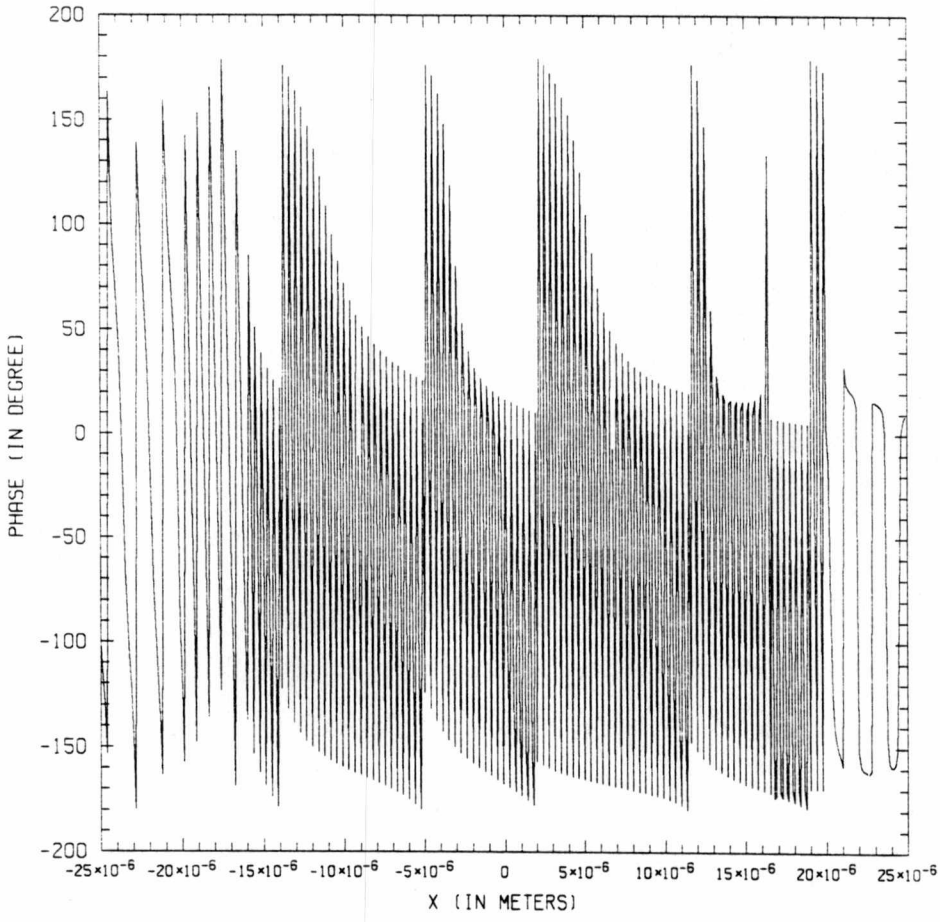


Figure 3.8: Phase of the electric field at $z = 0$ for the single slab waveguide being excited by pencil beam.

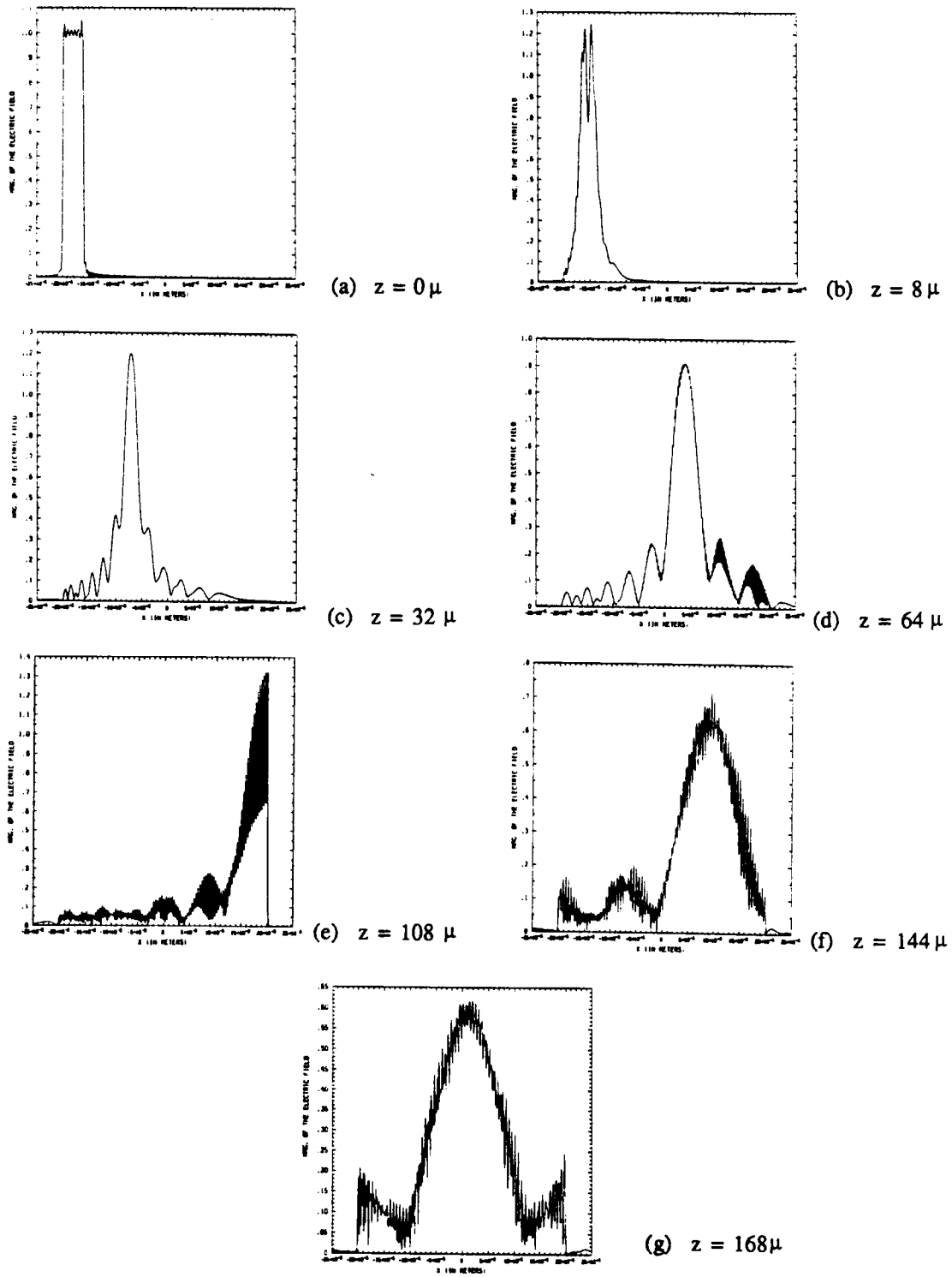


Figure 3.9: Electric fields at different z for the single slab waveguide being excited by pencil beam.

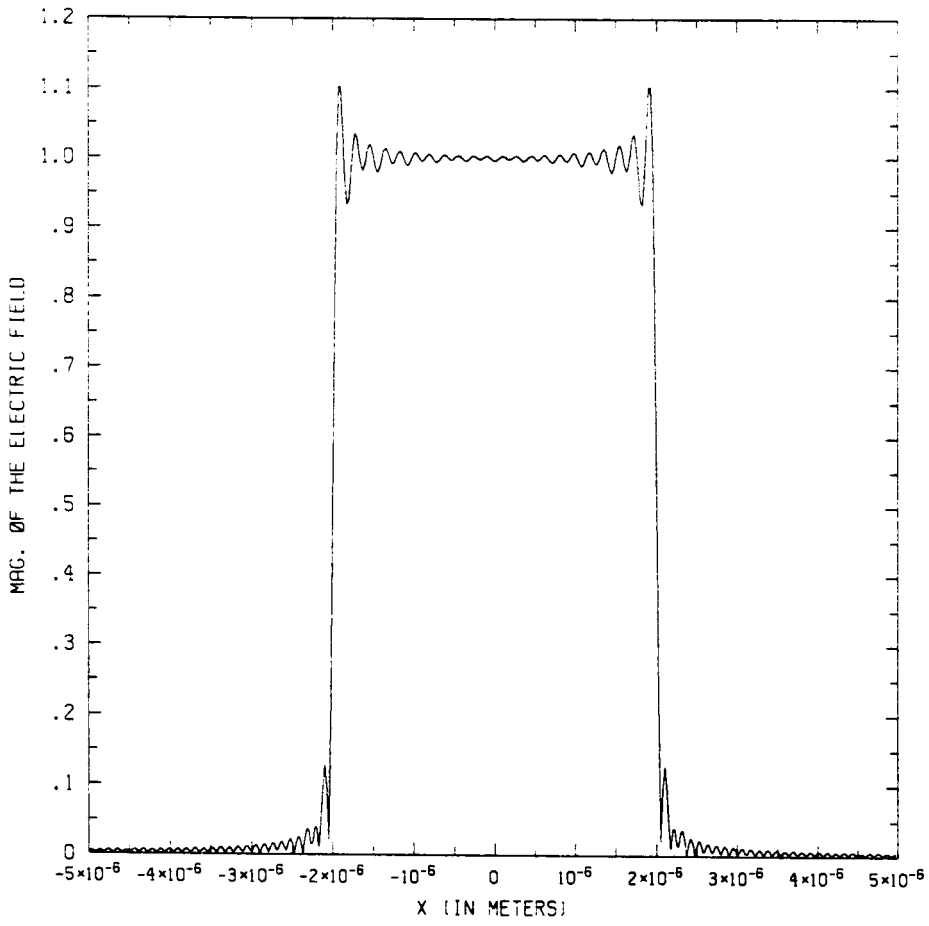


Figure 3.10: Electric field at $z = 0$ and $\alpha = 0$ for the single slab waveguide being excited by pencil beam.

3.1.3 Focused wave

To begin, we studied a case where a lens is placed against the slab. The geometry is shown in Figure 3.11, and the boundary condition is given as

$$f(x) = \begin{cases} e^{-j\frac{k}{2f}x^2} & |x| \leq a' \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where f is the focal length of the lens and k is the wave number ($= \frac{2\pi}{\lambda_0}$). The lens is finite in the x direction and infinite in the y direction. f is chosen to be 10μ. The electric field, E_y, at certain z values is calculated and plotted.

Before looking at the electric field plots, one may predict what the electric field should look like as the wave propagates in the slab. At $z = 0$, it should be obvious that the magnitude of the electric field is approximately equal to 1 , and the phase should be

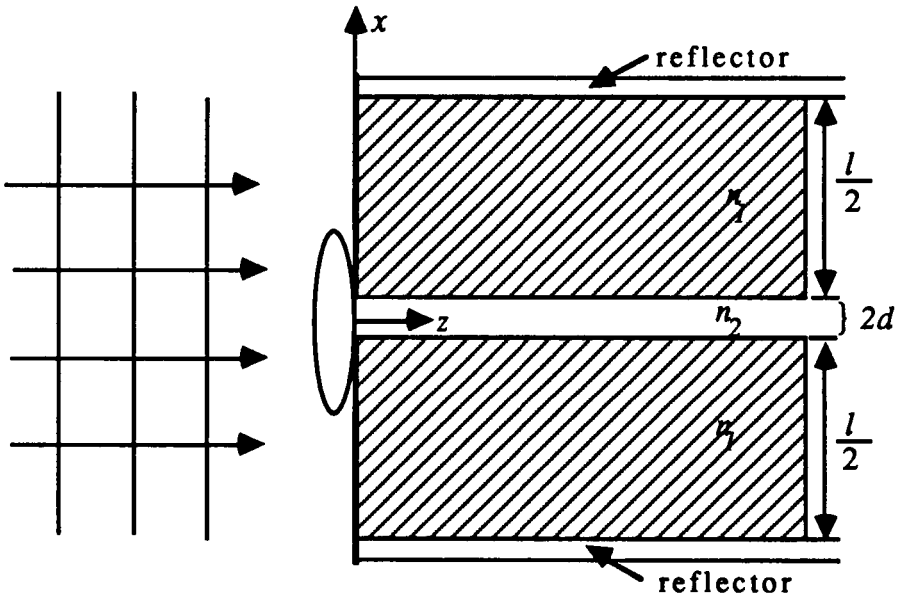


Figure 3.11: Lens against the single slab waveguide.

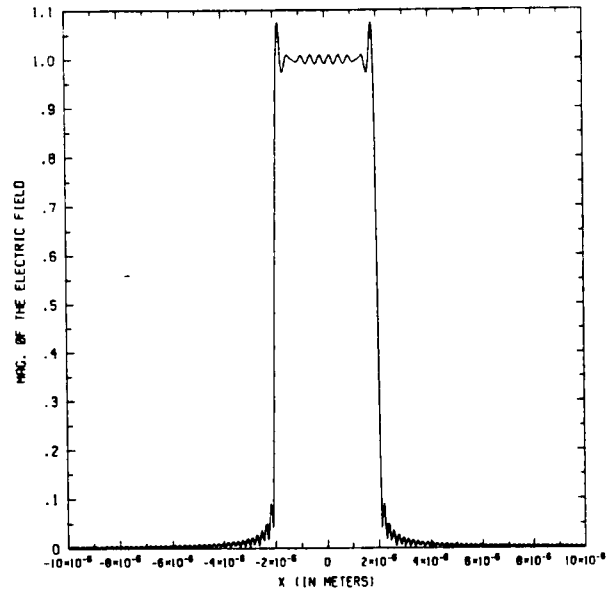
quadratic in x for $|x| \leq a'$. Now, if the slab has the same refractive index as air, the electric field should converge when $z = f$, but since $n_2 > n_{air}$, the apparent focal length should be larger and this can be calculated by Snell's law. In the case, $n_2 = 2$, $n_1 = 1.5$, $l = 200\mu$, $d = 25\mu$, $a' = 2\mu$, the apparent focal length is calculated to be approximately 20.2978μ. Thus the electric field is suspected to converge at $z = 20.2978\mu$. Now, plots of the electric field are studied to see if they agree with the predictions.

Figure 3.12 shows that the magnitude and phase agree with the predicted behaviour of the wave at $z = 0$ and thus the boundary condition is fitted quite well. When z increases from 0 to 60μ as shown in Figure 3.13 , the electric field does converge but it is not shown specifically clear at part (d), $z = 20.2978\mu$, where high concentrated electric field is expected. Also, the electric field strength is spreading out to the surrounding medium as z increases.

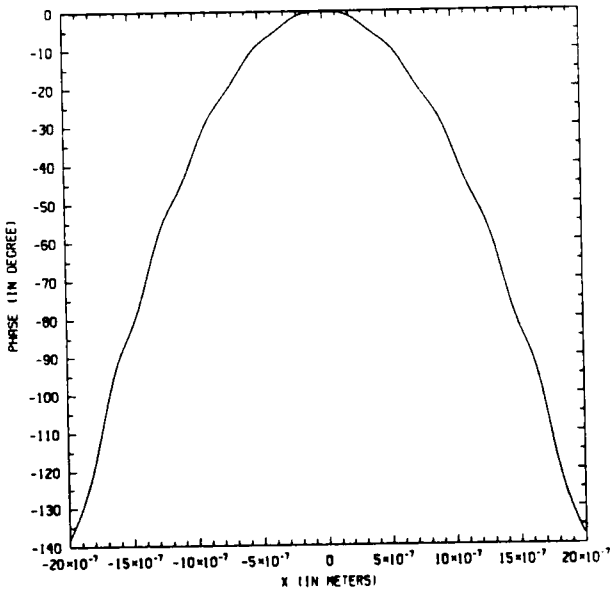
Next, f is considered to be very large ($\sim \infty$), we can see from equation 3.2 that $f(x)$ will be equal to 1 within a certain specified region in x, and it is just the case considered in section 3.1.1. The only difference is the phase for the boundary condition. Assume all the conditions are the same except one has a lens placed in front of the slab, the magnitude of the electric field at the same z in both cases should be very similar. Take $f = 0.1m$ and the results of the two cases are compared with the electric field plots shown in Figures 3.4 and 3.14. The electric fields at the same z distances into the slab are very similar in both cases.

In order to consider a more practical case, a lens is placed at a distance in front of the slab, and thus a converging wave is entering the slab (Fresnel diffraction). The geometry is shown in Figure 3.15 and the boundary condition is shown as below. Let

$$K = \frac{e^{jkz'}}{j\lambda_0 z'} e^{j\frac{k}{2z'}x^2}$$

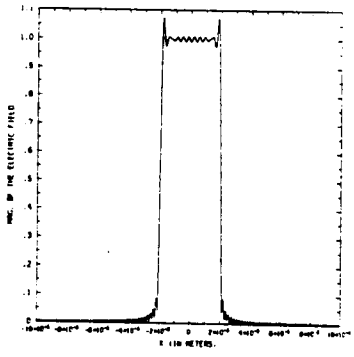


(a) Magnitude

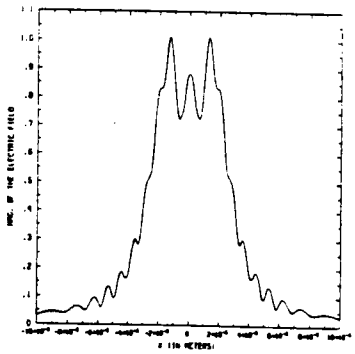


(b) Phase

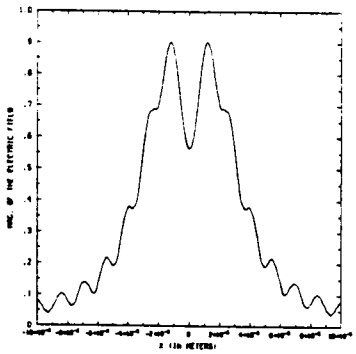
Figure 3.12: Magnitude and phase of the electric field at $z = 0$ for the single slab waveguide with lens against the slab.



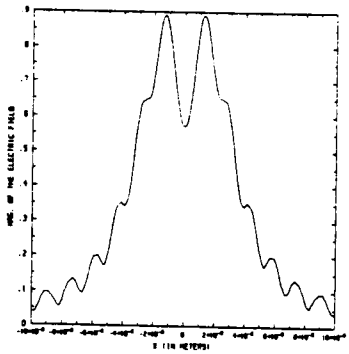
(a) z = 0 μ



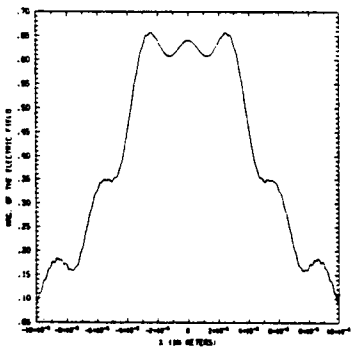
(b) z = 10 μ



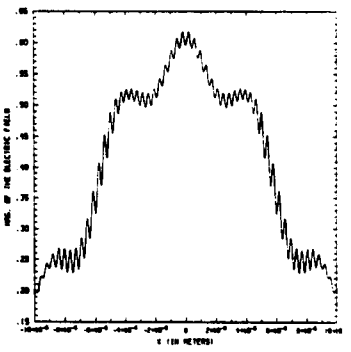
(c) z = 19 μ



(d) z = 20.2978 μ

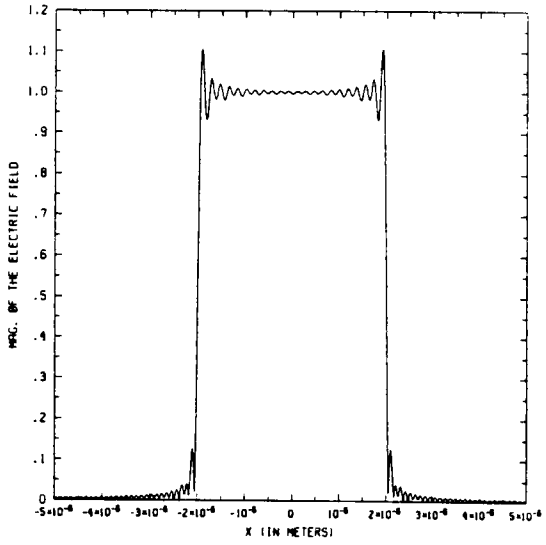


(e) z = 40 μ

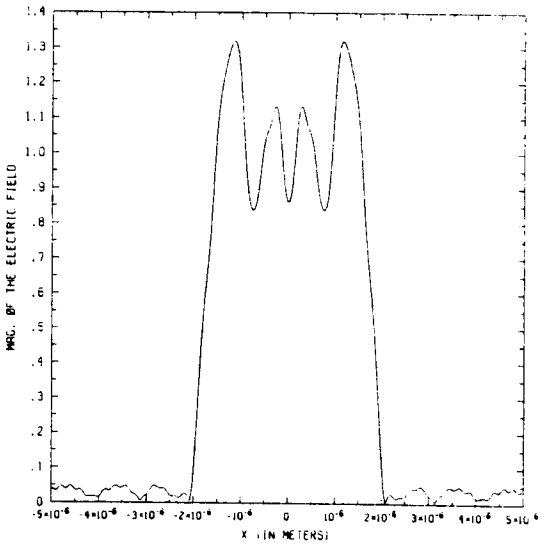


(f) z = 60 μ

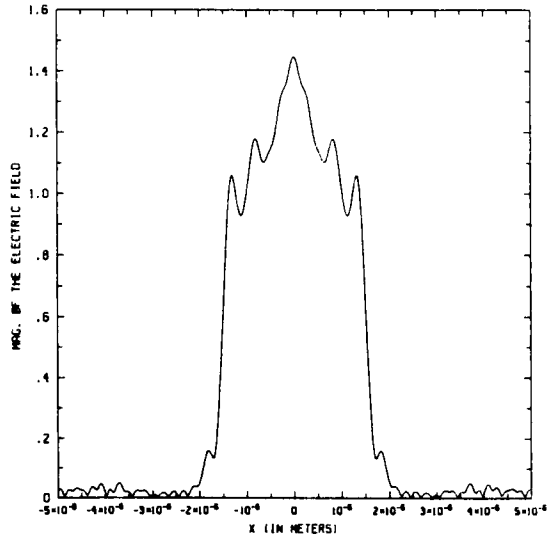
Figure 3.13: Electric fields at different z for the single slab waveguide with lens against the slab.



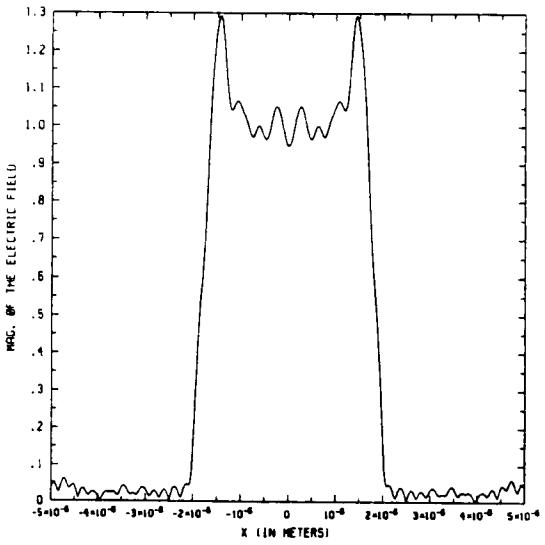
(a) z = 0 μ



(b) z = 4 μ



(c) z = 16 μ



(d) z = 64 μ

Figure 3.14: Electric fields at different z, and $f = 0.1m$ for the single slab waveguide with lens against the slab.

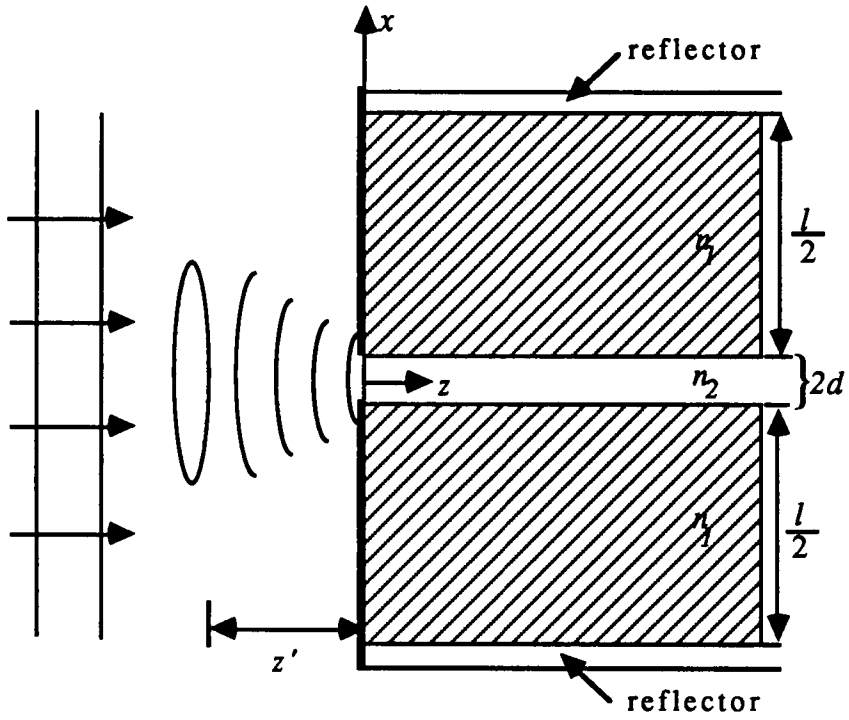


Figure 3.15: Focused wave entering the single slab waveguide.

and

$$f(x) = \begin{cases} K \int_{a'}^{b'} e^{j\frac{k}{2}(x')^2(\frac{1}{z'}-\frac{1}{f})} e^{-j\frac{k}{z'}xx'} dx' & |x| \leq d \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where f is the focal length of the lens, k is the wave number ($= \frac{2\pi}{\lambda_0}$), z' is the distance between the lens and the slab, and the lens is infinite in the y direction and finite in the x direction ($a' \leq x \leq b'$).

For all the calculations below (unless specially specified), the lens is assumed to have a focal length of 10μ, $a' = -5\mu$, $b' = 5\mu$, $d = 2\mu$, $l = 19\mu$, $n_2 = 2$ and $n_1 = 1.5$.

For a small z' (that is, less than the focal length and the slab is close to the lens), the wave entering the slab will be very similar to a plane wave and this can be proved by observing the electric field at $z = 0$ shown in Figure 3.16.

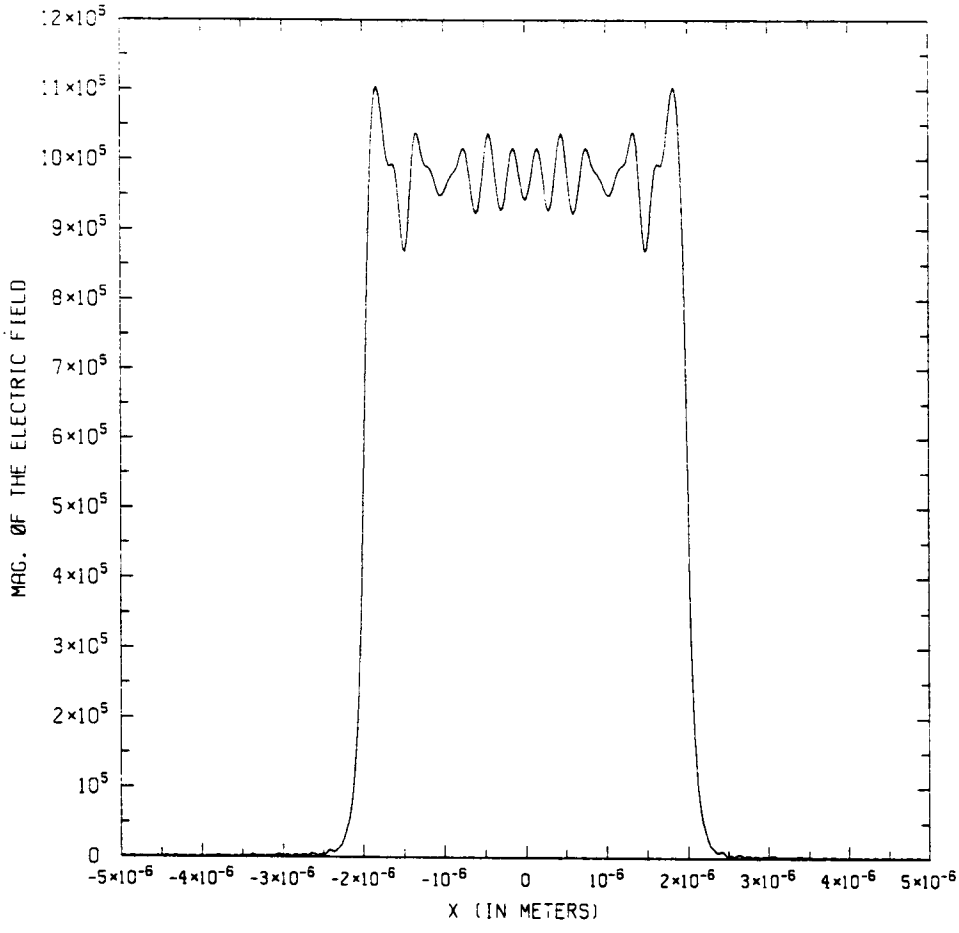


Figure 3.16: Electric field at $z' = 3\mu$ for the single slab waveguide being excited by focused wave.

When $z' = f$, magnitude of $f(x)$ should take the form as a sinc function because it is just like having a Fraunhofer diffraction with an aperture. In our case, the aperture is set by the size of the lens. The magnitude of the electric field at $z = 0$ is shown in Figure 3.17 and it is a sinc function as predicted. With this value for z', electric fields at different z are calculated and shown in Figure 3.18. These plots show that the electric field is spreading within the core as z increases but will converge into a shape similar to a sinc function (the boundary condition) when z increases further. After that, the electric field is again spreading and appears that the cycle of the electric field strength spreading and converging within the core is repeated.

When f is very large compare to the size of the slab and $z' = f$, the input wave sets a boundary condition which appears to be coming from a plane wave. It does not mean that a sinc function is not formed at $z = 0$. A sinc function still occurs but the distance between the first nulls is so much larger than the size of the slab that it appears to have a constant magnitude for $f(x)$ within the core. This can be shown by the electric field at $z = 0$ with $f = 0.02m$ in Figure 3.19.

From all the plots for the electric field, we can observe that the magnitudes are extremely high and we may suspect that they should not be greater than unity since the wave is coming from a unit amplitude plane wave. The reason is that we are considering a one-dimensional problem, not a two-dimensional problem.

Take the case where $z' = f$, and $f(x)$ should be a sinc function. It is the same as having a Fraunhofer diffraction pattern of an aperture set by the size of the lens. For the two-dimensional case with an aperture of size l_x by l_y. The magnitude of the Fraunhofer diffraction pattern is given as

$$\frac{l_x l_y}{\lambda_0 z'}$$

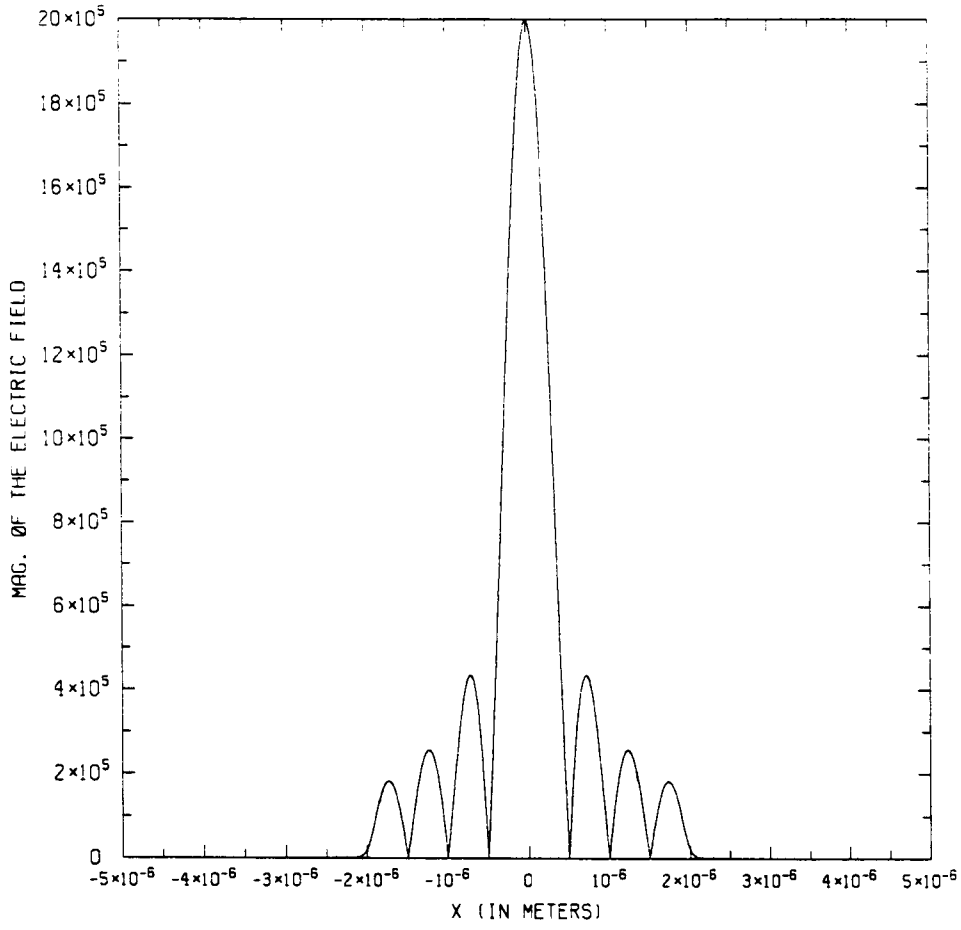


Figure 3.17: Electric field at $z' = f$ for the single slab waveguide being excited by focused wave.

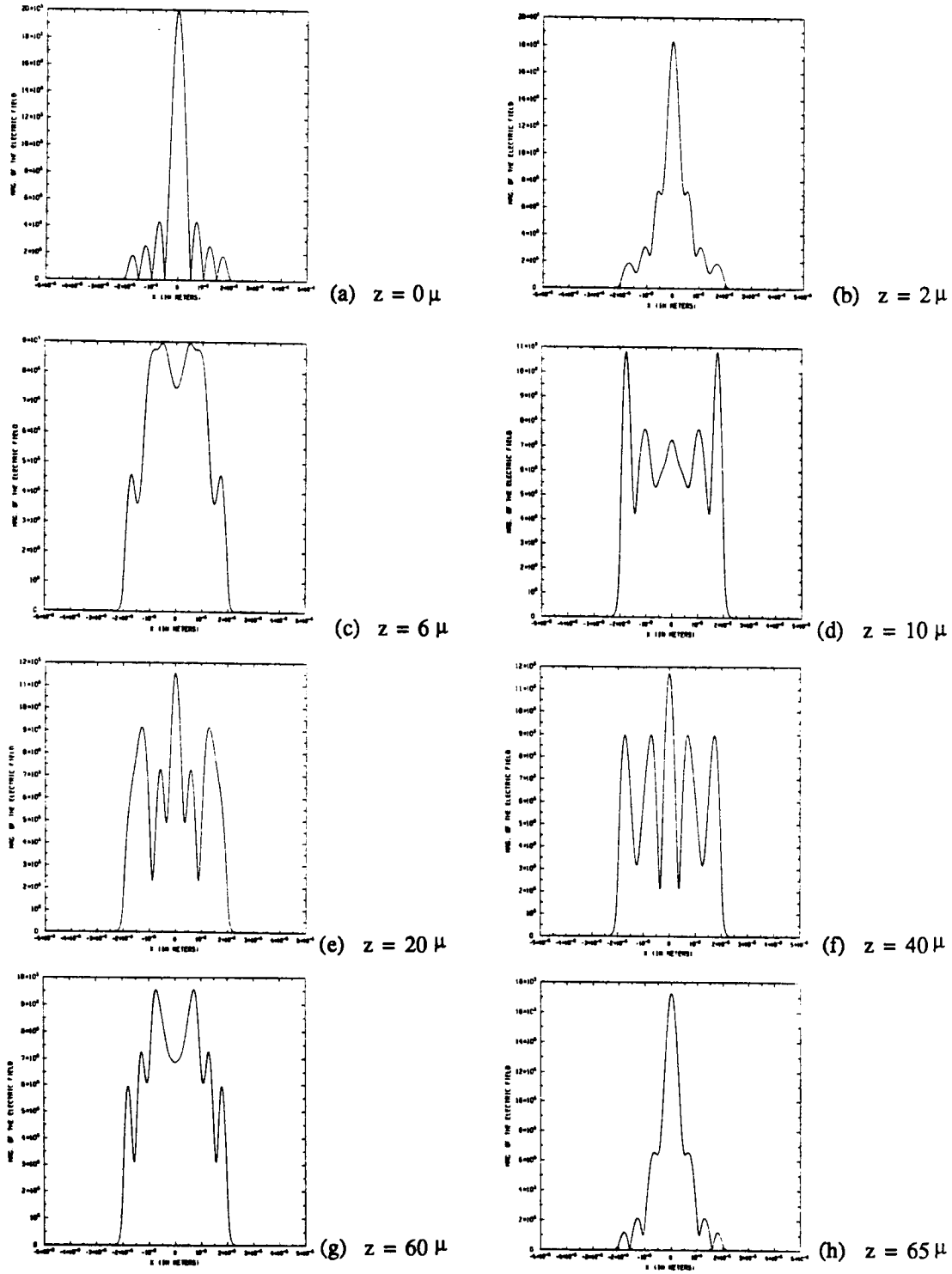


Figure 3.18: Electric fields at different z for the single slab waveguide being excited by focused wave.

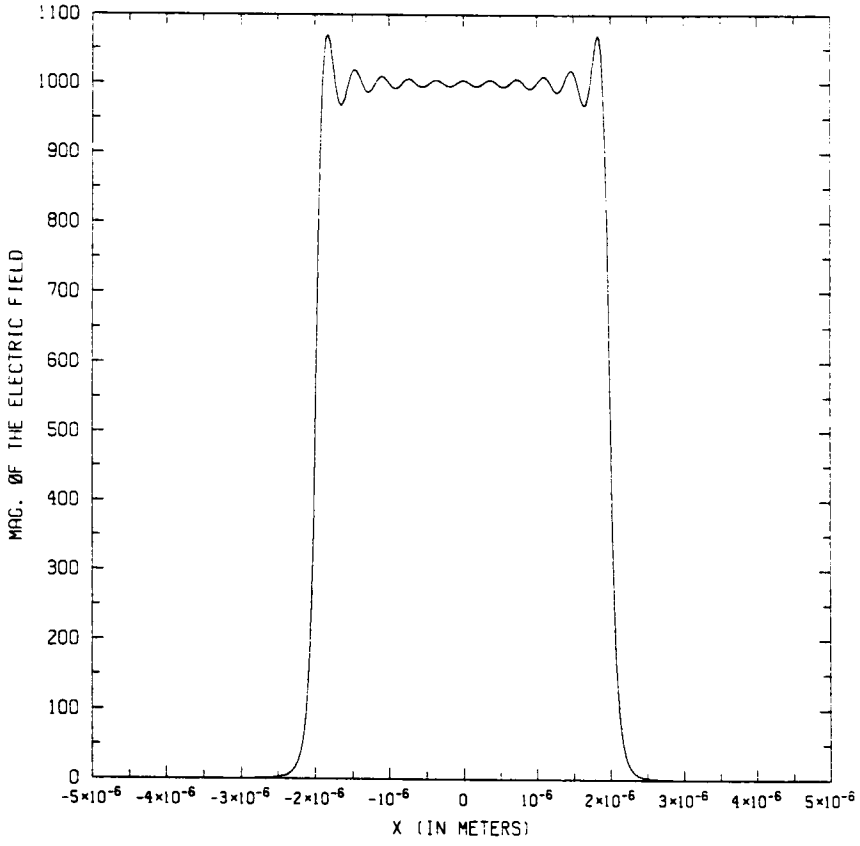


Figure 3.19: Electric field at $z = 0$ and $f = 0.02m$ for the single slab waveguide being excited by focused wave.

This is shown by J. Goodman in chapter 4 [6]. For the one-dimensional case we are working on, it is just like having an aperture which is between $x = a'$ and b' and infinite in the y direction. The magnitude is calculated by the same method as for the two-dimensional case but with $l_y = 1$. Although the same method is used, the meaning of the magnitude calculated in both cases is different. For the two-dimensional case, the magnitude describes the strength of the electric field and has a unit of V/m, but for the one-dimensional case, the magnitude describes the strength of the electric field per meter in the y direction and the unit is $V/m/m$. This explains why the magnitude calculated is so large.

3.2 Parallel slab waveguides

After observing the wave propagating behaviour of several spatial intensity distributions in the single slab waveguide, the wave propagation in the parallel waveguides is studied. Our main objective is to study the coupling effect in the parallel waveguides. Spatial intensity distributions of plane wave and focused wave are used to excite the parallel slab waveguides. Results are presented and explained.

3.2.1 Plane wave

A unit amplitude plane wave enters the parallel waveguides. The geometry is shown in Figure 3.20 and the boundary condition is given as

$$f(x) = \begin{cases} 1 & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

Only medium two is excited. In our study, we assume $n_2 = n_4 = 1.5$, $n_1 = n_3 = n_5 = 1.498$, $a = d = 2.4\mu$, $b = e = 0.5\mu$, $c = f = 400\mu$ and study the coupling between

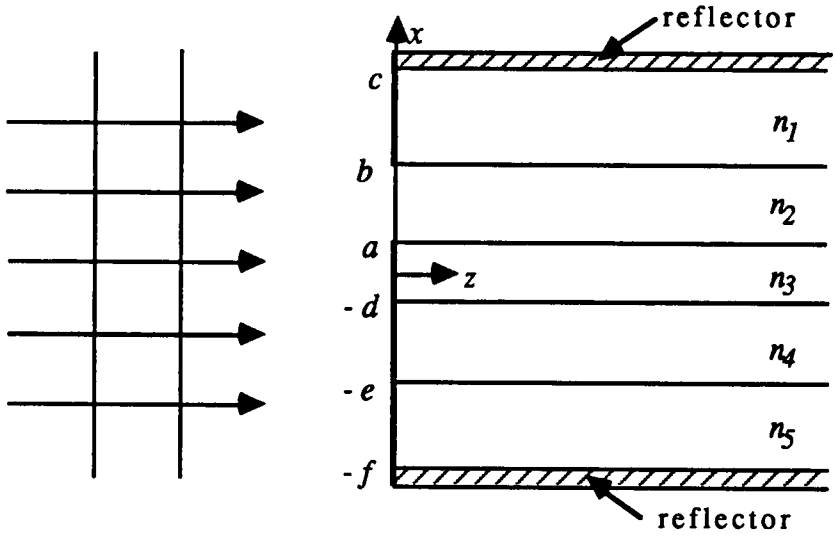


Figure 3.20: A unit plane wave entering the parallel slab waveguides.

medium two and medium four.

First, we let $a = d = 2\mu$, $b = e = 3.9\mu$, $c = f = 400\mu$, such that medium two and medium four is separated by a distance of 4μ. The electric field (with all the existed guided modes and one thousand five hundred radiation modes) at $z = 0$ is shown in Figure 3.21, and the boundary condition is fitted quite well. Due to the small difference between the refractive indices of the waveguide, most of the electric field strength is contained in the radiation modes. Thus, the boundary condition can be fitted much better if more radiation modes are used but the computing time will be increased at the same time. In our case, the boundary condition is fitted acceptably. After the boundary condition is fitted, electric fields at different values of z into the waveguides are plotted and shown in Figure 3.22. From Figure 3.22, we notice that most of the energy is confined inside medium two as the wave propagates along the parallel waveguides and a

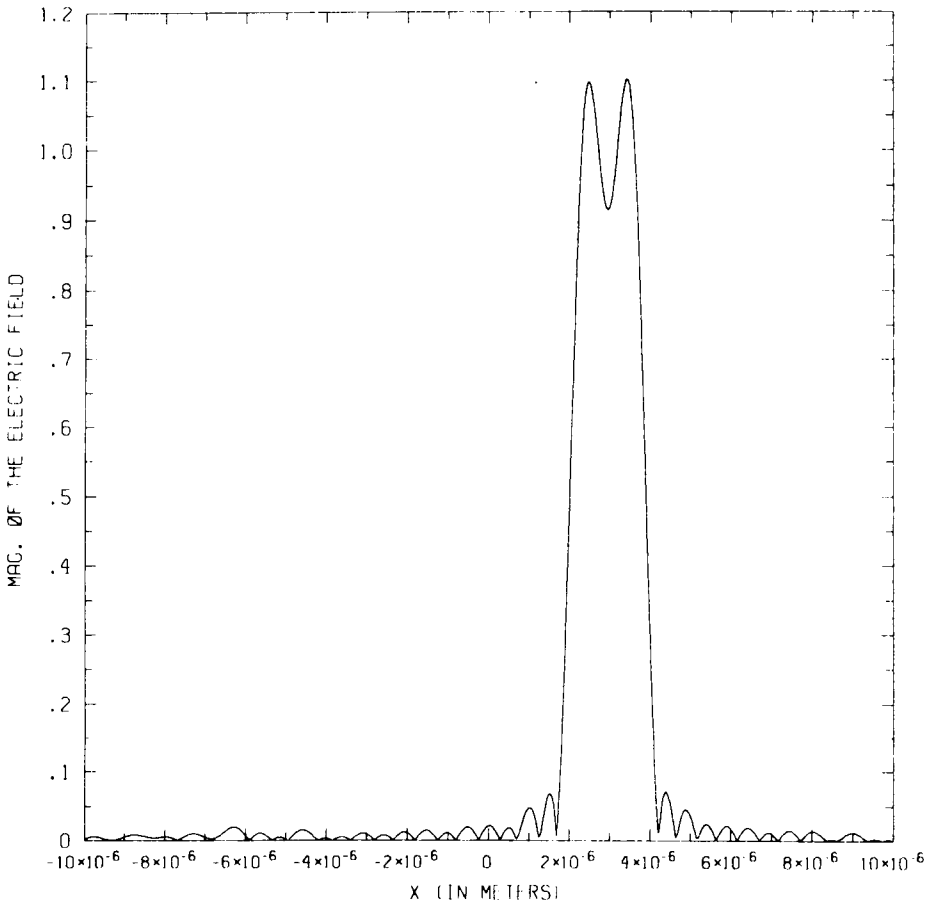


Figure 3.21: Electric field at $z = 0$ for the parallel waveguides being excited by plane wave.

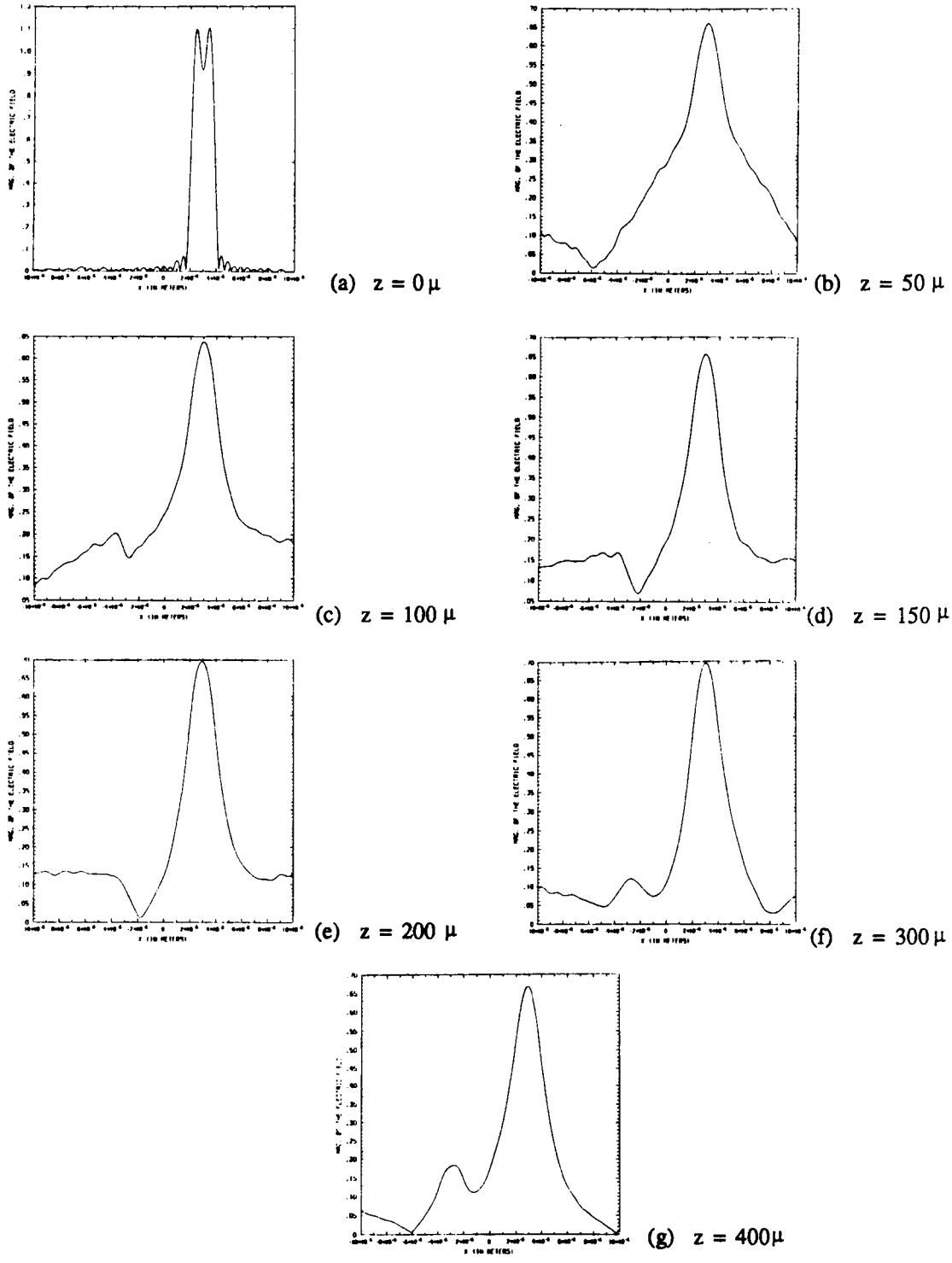


Figure 3.22: Electric fields at different z for the parallel waveguides being excited by plane wave and the waveguides are separated by a distance of 4μ.

very small amount of energy is being coupled into the adjacent waveguide.

Next, the distance between the waveguides is reduced to 1μ and that is $a = d = 2.4\mu$,$b = e = 0.5\mu$. We may predict that since the waveguides are closer together, coupling between them should increase. In order to prove that, the electric fields at diffferent z are plotted and shown in Figure 3.23. From these figures, we can see that energy is being coupled from the waveguide being excited to the adjacent waveguide as the wave propagates down the z axis. Thus, when we compare the above two cases, we can show that the coupling power between waveguides increases as the distance between the waveguides decreases.

3.2.2 Focused wave

The case of a focused wave entering one of the waveguides and the coupling effect is studied. The geometry is shown in Figure 3.24 and the boundary condition is given as below. Let

$$K = \frac{e^{jkz'}}{j\lambda_0 z'} e^{j\frac{k}{2z'}x^2}$$

and

$$f(x) = \begin{cases} K \int_{a'}^{b'} e^{j\frac{k}{2}(x')^2(\frac{1}{z'}-\frac{1}{f})} e^{-j\frac{k}{z'}xx'} dx' & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

where f is the focal length of the lens, k is the wave number ($= \frac{2\pi}{\lambda_0}$), z' is the distance between the lens and the slab, and the lens is infinite in the y direction and finite in the x direction ($a' \leq x \leq b'$).

In our case, $f = 10\mu$, $z' = 10\mu$, $a' = 3.95\mu$, $b' = 6.05\mu$, $n_2 = n_4 = 1.5$, $n_1 = n_3 = n_5 = 1.498$, $a = d = 2.4\mu$, $b = e = 0.5\mu$ and $c = f = 400\mu$. It is the same as the case mentioned in the last section where the separation between the waveguides is 1μ except now a focused wave is used. Since z' is chosen to be equal to f, a sinc function should

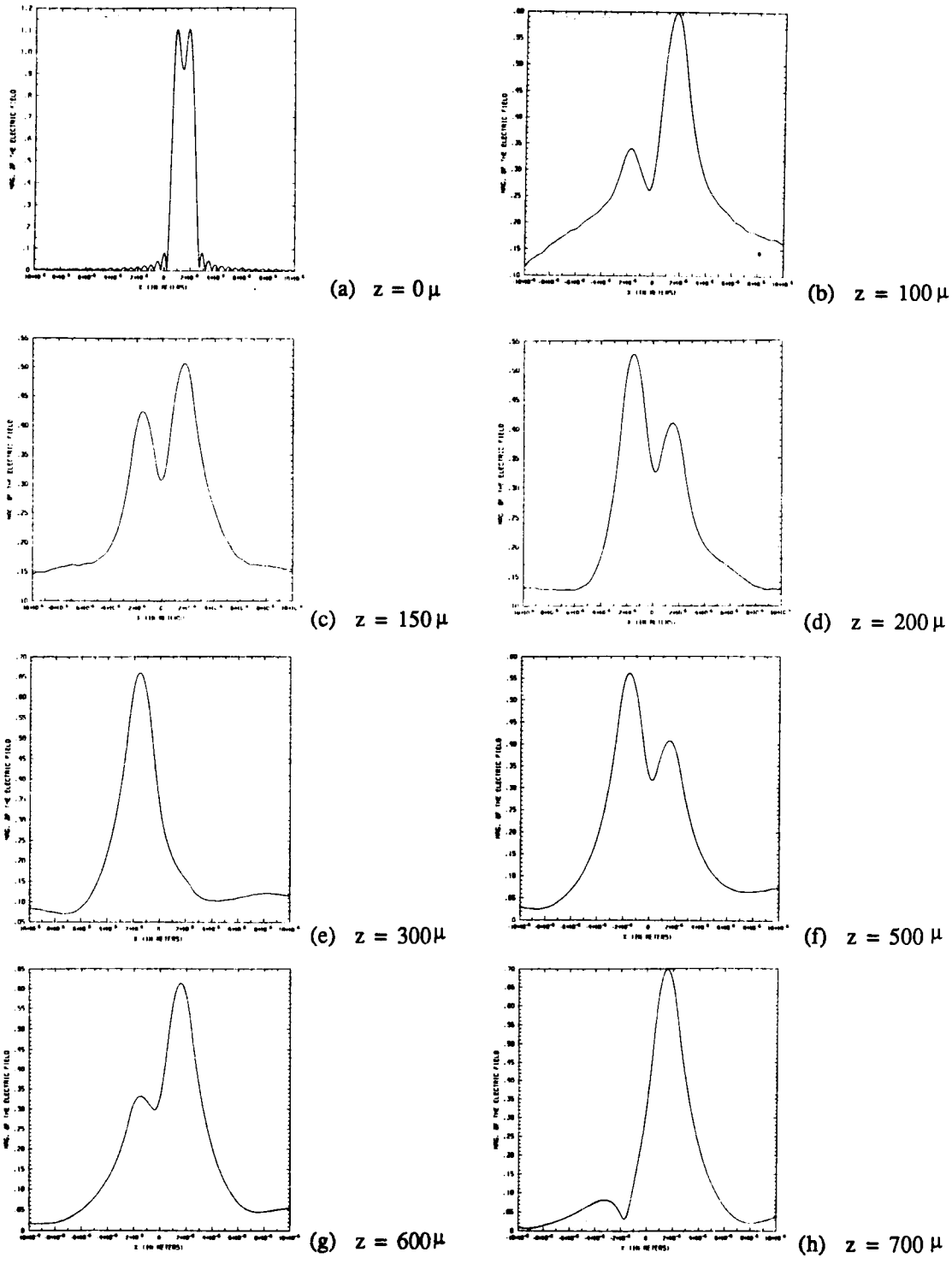


Figure 3.23: Electric fields at different z for the parallel waveguides being excited by plane wave and the waveguides are separated by a distance of 1μ.

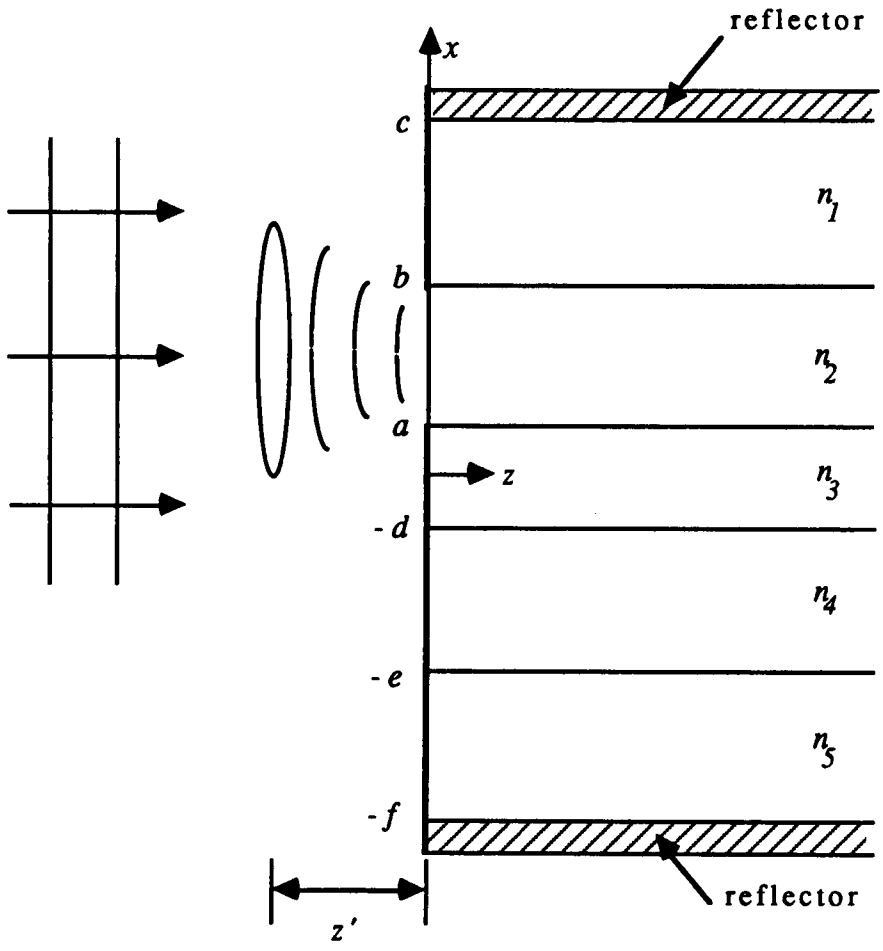


Figure 3.24: Focused wave entering the parallel slab waveguides.

occur at $z = 0$ and this is proved by the electric field plot (with all existing guided modes and one thousand five hundred radiations modes) shown in Figure 3.25. Again, the boundary condition can be fitted better if more radiation modes are used. Electric fields at different value of z are plotted and shown in Figure 3.26. In comparing Figures 3.23 and 3.26, we notice that the coupling behaviour for the two cases is very similar except the magnitude in the second case is very large. The reason for such large values in the magnitude for the electric field is the same as explained in the results for the single slab waveguide being excited by focused wave.

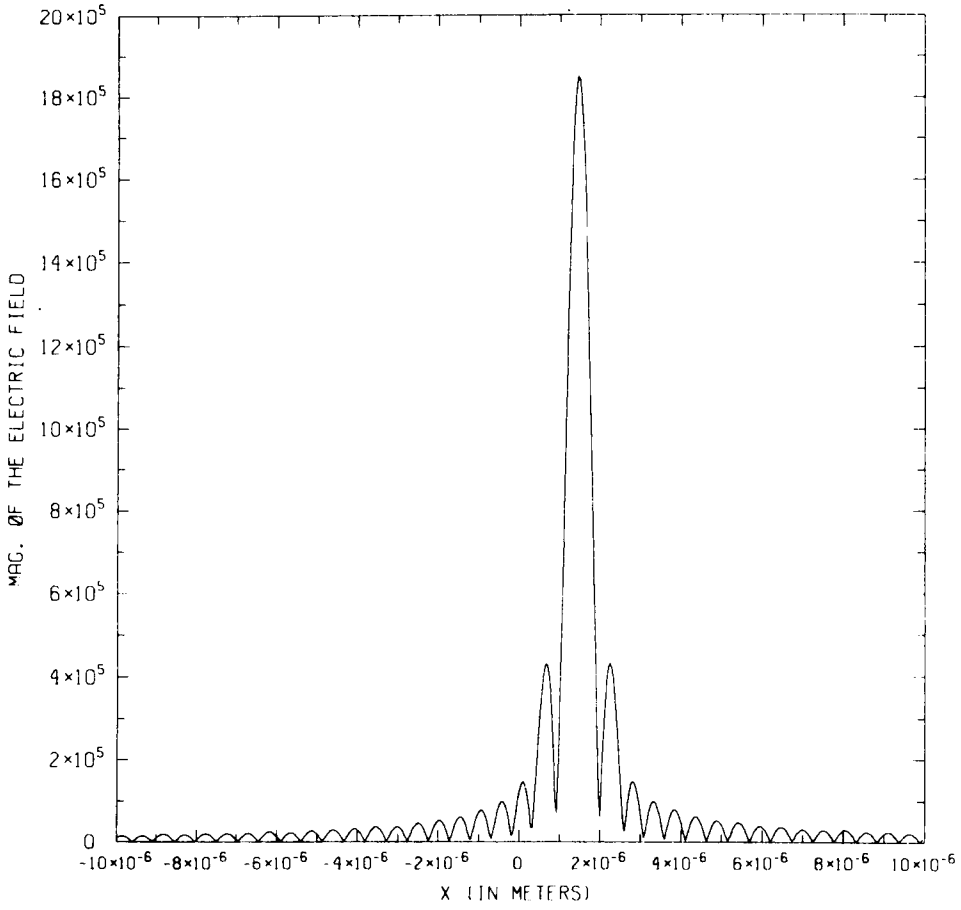


Figure 3.25: Electric field at $z = 0$ for the parallel waveguides being excited by focused wave.

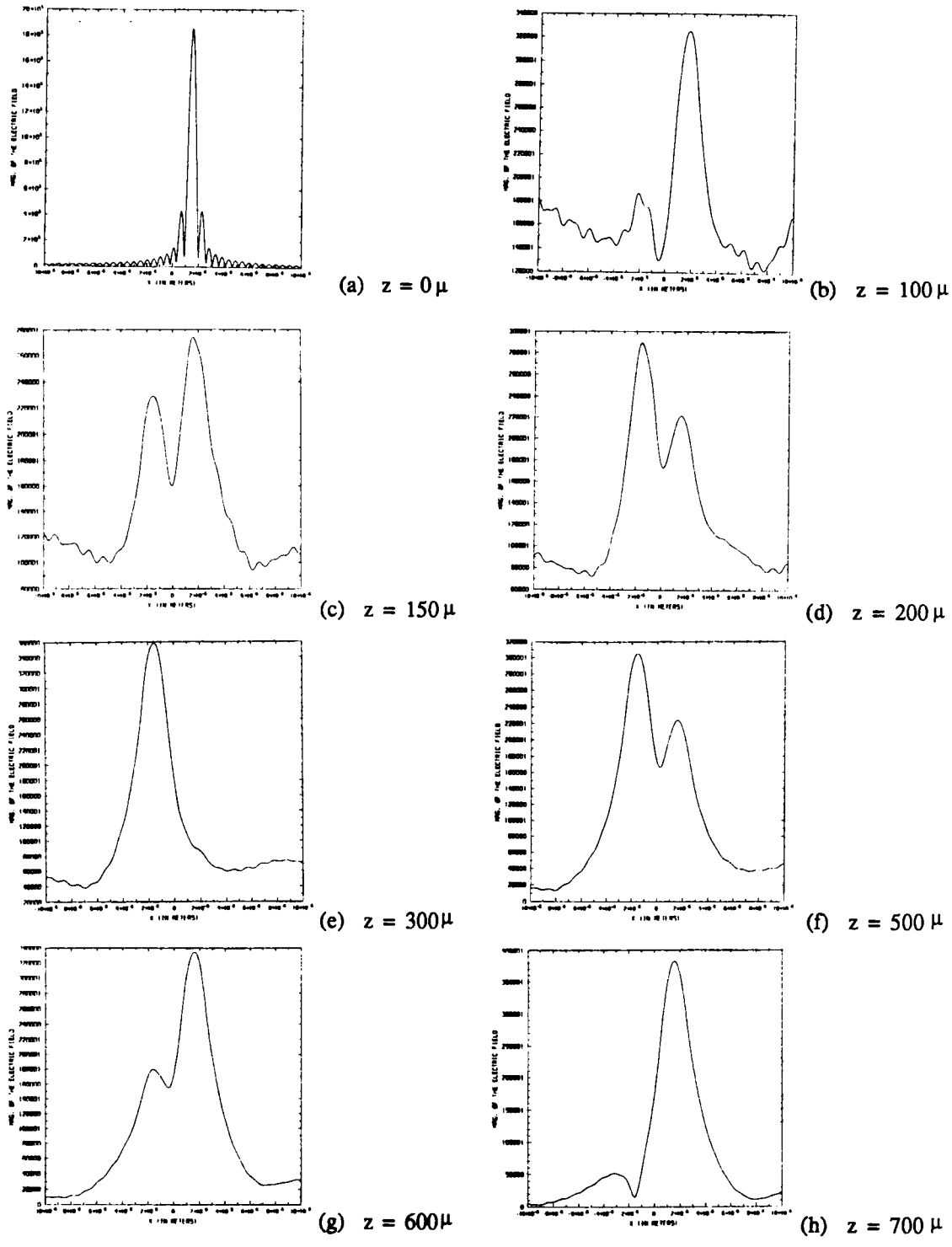


Figure 3.26: Electric fields at different z for the parallel waveguides being excited by focused wave and the waveguides are separated by a distance of 1μ.

CHAPTER 4

CONCLUSION

This thesis studies wave propagation in one-dimensional single slab waveguides and parallel waveguides. The waveguides are excited by specific intensity distributions and boundary conditions are set up on the xy plane. The z axis is the axis of wave propagation. Methods for finding the mode structures existing in the waveguides are presented and shown to be accurate and successful. Also the methods of orthogonal functions are used to find the weighting factor for each guided and radiation mode. The weighting factors are needed in order to fit the boundary condition set up by the spatial intensity distribution. From the various results shown in chapter 3, the use of the orthogonal functions to find the weighting factors is correct.

For the single slab waveguide, wave propagation of plane wave, pencil beam and focused wave are studied. As for the plane wave, we can observe from the results and understand that the energy is spreading out or recombining in the core as the wave is propagated along the z axis. This effect can be explained by the fact that the phase of each guided mode changes at different value of z. Energy is being added in phase or out of phase for different value of z. Also, one of the results for the plane wave propagation showed that as the difference of the refractive indices between the core and the surrounding medium decreases, more and more energy exists in the radiation modes instead of the guided modes.

Next, when a pencil beam is used to excite the single slab waveguide, the wave bounces back and forth inside the core. This phenomenon agreed with what we have

expected for the wave to be totally internal reflected within the core. For the case with the focused wave exciting the single slab waveguide, it is a practical problem because there are situations which wave is focused onto the waveguide to be transmitted. The behaviour of the wave propagation is similar to that of the the plane wave. The energy is spread out of the core and recombined in the core as the value of z changes. But, there is one important point to be learned from the focused wave case. That point is that the method we used to calculate the electric field applys to a two-dimensional problem and the cases we considered are one-dimensional. Thus, the electric field we are actually calculating is the electric field per meter in the y direction. This is the reason why such large values for the magnitude of the electric field occurred.

For the parallel slab waveguides, wave propagation of plane wave and focused wave are studied. The coupling effect between the waveguides is studied by learning the relationship between the separation of the waveguides and the energy coupled between the waveguides. A case is studied in which the waveguides are separated by a certain distance, and the electric field calculated showed that most of the energy is confined in the waveguide which is being excited. Then the waveguides are placed closer together and the coupling of the energy between the waveguides is studied. The results showed that energy is transferred from the waveguide being excited to the adjacent waveguide, and then the energy is transferred back to the other waveguide. Thus, the coupling of the energy between the waveguides is related to the separation between them. As the separation between the waveguides increases, the coupling between them decreases.

In conclusion, this thesis successfully shows the wave propagation pattern for specific intensity distribution in single slab waveguide and parallel slab waveguides. Methods for finding the mode structures and fitting boundary conditions set by the spatial intensity distribution are presented and they proved from the results that they are successful.

After gaining an understanding how a wave propagates in slab dielectric waveguide, studying wave propagation in optical fibres is the next logical move and it is suggested for future research. Due to the geometry of optical fibres, a three dimensional problem will be encountered instead of the two dimensional problem studied in this thesis. Thus, a more complicated problem is expected, but it will be very challenging too!

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] A. Yariv, *Optical Electronics.*
CBS College Publishing, 1985.

[2] D. L. Lee, *Electromagnetic Principles of Integrated Optics.*
John Wiley and Sons, Inc., 1986.

[3] N. S. Kapany and J. J. Burke, *Optical Waveguides.*
Academic Press, Inc., 1972.

[4] D. Marcuse, *Theory of Dielectric Optical Waveguide.*
Academic Press, Inc., 1974.

[5] E. Kreyszig, *Advaned Engineering Mathematics.*
John Wiley and Sons, Inc., 1983.

[6] J. W. Goodman, *Introduction to Fourier Optics.*
McGraw-hill, Inc., 1968.

APPENDIX

```
C  PROGRAM : To determine all the guided modes exist for the single
C           slab waveguide being excited by plane wave.  Also, the
C           weighting factor for each guided mode is calculated.
C
      CHARACTER*3 SIGN2, SIGN4
      CHARACTER*4 MODE
      CHARACTER*5 REPEAT
      COMPLEX*8   FIELD
      REAL*8      PI, U2, HD, PD, K0, D2, D, N2, N1, LOW, HIGH, L, A2,
     $            LPT, HPT, P1, P2, P3, P4, P5, FP1, FP2, FP4, FP5, P,
     $            DEN01, DEN02, DEN04, DEN05, H, LAMBDA0, DUM, DUM0,
     $            EVENP(100), ODDP(100), EVENH(100), ODDH(100),EC(100)
      INTEGER     DUMMY, NUM, M, I, COUNT, COUNT1, COUNT2, ENUM, ONUM
C
      OPEN (UNIT=2,FILE='PH.DAT',STATUS='NEW')
      OPEN (UNIT=3,FILE='GECOEFF.DAT',STATUS='NEW')
      OPEN (UNIT=4,FILE='GOCOEFF.DAT',STATUS='NEW')
C
C  N2 : Refractive index of the slab
C  N1 : Refractive index of the surrounding medium
C  D2  : Thickness of the slab in meters
C  EC : Coefficients for the even guided modes
C  OC : Coefficients for the odd guided modes
C  ENUM : Number of even guided modes
C  ONUM : Number of odd guided modes
C  LAMBDA0 : Free-space wavelength of light
C  PH : File contains propagation constant for each guided mode
C  GECOEFF : File contains weighting factor for each even guided mode
C  GOCOEFF : File contains weighting factor for each odd guided mode
C
C  Obtain information from the user
C
      WRITE(*,*) ' Thickness of the slab (in meters) = ?'
      READ(*,*) D2
      WRITE(*,*) ' Distance betwen the reflectors (minus the slab) ='
      READ(*,*) L
      WRITE(*,*) ' Refractive index of the slab = ?'
      READ(*,*) N2
      WRITE(*,*) ' Refractive index of the surrounding medium = ?'
      READ(*,*) N1
C
C  Set up constants for calculations
C
      PI = 3.141592654
      LAMBDA0 = 0.5E-6
      D = D2/2
      K0 = 2*PI/LAMBDA0
      U2 = (N2**2 - N1**2)*(K0**2)*(D**2)
      COUNT = INT((SQRT(U2))*2/PI)
      WRITE(2,*) D, N2, LAMBDA0
C
C  To find the existed guided modes
C
C     Find the number of existed guided modes (even/odd)
C
      ENUM = 0
      ONUM = 0
      DO M = 0, COUNT
         LOW = M/2.0
```

```
            HIGH = LOW + 0.5
            LPT = LOW*PI
            HPT = HIGH*PI
            DUMMY = INT(LOW)
            IF ((DUMMY-LOW) .EQ. 0) THEN
               MODE = 'EVEN'
               ENUM = ENUM + 1
            ELSE
               MODE = 'ODD '
               ONUM = ONUM + 1
            ENDIF
            NUM = DUMMY + 1
            WRITE(*,*) MODE,'#',NUM
C
C           Find the propagation constant for each guided mode
C
            P1 = LPT + 1E-20
            P5 = HPT
            REPEAT = 'TRUE '
            DO WHILE (REPEAT .EQ. 'TRUE ')
               P3 = (P5-P1)/2 + P1
               P2 = (P3-P1)/2 + P1
               P4 = (P5-P3)/2 + P3
               IF (MODE .EQ. 'EVEN') THEN
                  DEN02 = DTAN(P2)**2
                  DEN04 = DTAN(P4)**2
               ELSE
                  DEN02 = 1/(DTAN(P2)**2)
                  DEN04 = 1/(DTAN(P4)**2)
               ENDIF
               FP2 = U2 - (P2**2)*(1+DEN02)
               FP4 = U2 - (P4**2)*(1+DEN04)
               IF (FP2 .GT. 0) THEN
                  SIGN2 = 'POS'
               ELSE
                  SIGN2 = 'NEG'
               ENDIF
               IF (FP4 .GT. 0) THEN
                  SIGN4 = 'POS'
               ELSE
                  SIGN4 = 'NEG'
               ENDIF
               IF (SIGN2 .EQ. SIGN4) THEN
                 IF (FP2 .GT. FP4) THEN
                   IF (SIGN2 .EQ. 'POS') THEN
                      P1 = P4
                      P5 = P5
                   ELSE
                      P1 = P1
                      P5 = P2
                   ENDIF
                 ELSE
                   IF (SIGN .EQ. 'POS') THEN
                      P1 = P1
                      P5 = P2
                   ELSE
                      P1 = P4
                      P5 = P5
                   ENDIF
```

```
                ENDIF
              ELSE
                P1 = P2
                P5 = P4
              ENDIF
              IF (MODE .EQ. 'EVEN') THEN
                 DEN01 = DTAN(P1)**2
                 DEN05 = DTAN(P5)**2
              ELSE
                 DEN01 = 1/(DTAN(P1)**2)
                 DEN05 = 1/(DTAN(P5)**2)
              ENDIF
              FP1 = U2 - (P1**2)*(1+DEN01)
              FP5 = U2 - (P5**2)*(1+DEN05)
              IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
                 IF (ABS(FP1) .LT. ABS(FP5)) THEN
                    HD = P1
                 ELSE
                    HD = P5
                 ENDIF
                 REPEAT = 'FALSE'
              ELSE
                 REPEAT = 'TRUE '
              ENDIF
            END DO
            PD = SQRT(U2-HD**2)
            P = PD/D
            H = HD/D
            IF (MODE .EQ. 'EVEN') THEN
              EVENP(NUM) = P
              EVENH(NUM) = H
            ELSE
              ODDP(NUM) = P
              ODDH(NUM) = H
            ENDIF
         END DO
C
C  Write the number of odd and even guided modes and their correspnding
C  propgation constant to PH.DAT
C
         COUNT1 = NUM
         COUNT2 = COUNT1
         IF (MODE .EQ. 'EVEN') THEN
            COUNT2 = NUM-1
         ENDIF
         DO M = 1, COUNT1
            WRITE(2,*) EVENP(M), EVENH(M)
         END DO
         DO M = 1, COUNT2
            WRITE(2,*) ODDP(M), ODDH(M)
         END DO
         WRITE(3,*) ENUM
         WRITE(4,*) 0
C
C  Compute the weighting factor for the even gudied modes, the weighting
C  factors for odd guided modes are not calculated because no odd guided
C  modes exist for the symmetrical slab waveguide excited by plane wave.
C
         DO I = 1, ENUM
```

```
            DUM= EVENH(I) * D
            A = DCOS(DUM)
            A2 = A**2
            EC(I) = 4*DSIN(DUM)/(2*DUM+DSIN(2*DUM)+A2*2*EVENH(I)
     $              /EVENP(I))
            WRITE(3,*) EC(I)
         END DO
C
         CLOSE (UNIT=2)
         CLOSE (UNIT=3)
         CLOSE (UNIT=4)
         STOP
         END
```

```
C  PROGRAM : To determine the radiation modes for the single slab
C           waveguide being excited by plane wave.  Also, the weight-
C           ing factor for each radiation mode used is calculated.
C
      CHARACTER*3 SIGN1, SIGN2, SIGN4, SIGNL, SIGNR, LSLOPE, RSLOPE
      CHARACTER*4 MODE, SIGN
      CHARACTER*5 REPEAT
      COMPLEX*8   RFIELD
      REAL*8 EVENX(1000), ODDX(1000), V, DELK, D2, D, L, N2, N1,
     $       COTAN(1000), ETAN(1000), OTAN(1000), LAMBDA0, START,
     $       ELEFT(1000), ERIGHT(1000), OLEFT(1000), ORIGHT(1000),
     $       P1, P2, P3, P4, P5, FP1, FP2, FP4, FP5, DUM1, DUM2,
     $       DUM, SLOPE, XL1, XL2, XR1, XR2, YL1, YL2, YR1, YR2,
     $       K1X, K2X, EC(1000), OC(1000), PART1, PART2, PART3,
     $       PART4, PART5, PART6, PART7, PART8, PART9, PART10,
     $       PART11, PART12, PART13, PART14, NUMER, DEN0, DUMMY,
     $       DUM0,K1EVEN(1000),K1ODD(1000),K2EVEN(1000),K2ODD(1000)
      INTEGER I, M ,N, J, K, COUNT, NUM, ENUM, ONUM
C
      OPEN (UNIT=2,FILE='K1K2.DAT',STATUS='NEW')
      OPEN (UNIT=3,FILE='RECOEFF.DAT',STATUS='NEW')
      OPEN (UNIT=4,FILE='ROCOEFF.DAT',STATUS='NEW')
C
C  D2 : Thickness of the slab in meters
C  N2 : Refractive index of the slab
C  N1 : Refractive index of the surrounding medium
C  ENUM : Number of even radiation modes used
C  ONUM : Number of odd radiation modes used
C  LAMBDA : Free-space wavelength of light
C  K1K2 : File contains propagation constant for each radiation mode
C  RECOEFF : File contains weighting factor for each even radiation mode
C  ROCOEFF : File contains weighting factor for each odd radiation mode
C
C  Obtain information from the user
C
      TYPE *,' Thickness of the slab (in meters)= ?'
      READ(*,*) D2
      TYPE *,' Distance between the reflectors (minus the slab) = ?'
      READ(*,*) L
      TYPE *,' Refractive index of the slab = ?'
      READ(*,*) N2
      TYPE *,' Refractive index of the surrounding medium = ?'
      READ(*,*) N1
      TYPE *,' No. of even modes to be used (<1000) = ?'
      READ(*,*) ENUM
      TYPE *,' No. of odd modes to be used (<1000) = ?'
      READ(*,*) ONUM
C
C  Set up constants for calculations
C
      D = D2/2
      PI = 3.141592654
      DUM0 = 0.0
      LAMBDA0 = 0.5E-6
      DELK = (2*PI/LAMBDA0)*(SQRT(N2**2-N1**2))
      V = DELK*D
      WRITE(2,*) D, N2, LAMBDA0
C
C  Finding X at which X*COTAN(X*L/2D) goes to infinity :
```

```
C
      START = 0.0
      DO I = 1, 200
         COTAN(I) = START + (2*D*PI/L)*(I-1)
      END DO
C
C  Finding X at which SQRT(X**2+V**2)*TAN(DQRT(X**2+V**2))
C  goes to infinty :
C
C  A)  Even radiation modes :
C
      IF (V .LE. PI/2) THEN
         START = PI/2
      ELSE
         START = INT((V-PI/2)/PI)*PI + 3*PI/2
      ENDIF
      DO I = 1, 200
         ETAN(I) = SQRT((START + PI*(I-1))**2 - V**2)
      END DO
C
C  B)  Odd radiation modes :
C
      IF (V .LE. PI) THEN
         START = PI/2
      ELSE
         START = (INT(V/PI)+1)*PI - PI/2
      ENDIF
      DO I = 1, 200
         OTAN(I) = SQRT(((START + PI*(I-1))+PI/2)**2 - V**2)
      END DO
C
C  Sorting out ranges where solutions for X exist (even modes)
C
      ELEFT(1) = COTAN(1)
      M = 2
      N = 1
      COUNT = 1
      DO WHILE (COUNT .LE. 160)
         IF (ETAN(N) .LT. COTAN(M)) THEN
            XL1 = ELEFT(COUNT) + (ETAN(N)-ELEFT(COUNT))/(500*L/D)
            CALL EDIFF (XL1,L,V,D,YL1)
            IF (YL1 .GT. 0) THEN
               SIGNL = 'POS'
            ELSE
               SIGNL = 'NEG'
            ENDIF
            XR1 = ETAN(N) - (ETAN(N)-ELEFT(COUNT))/(500*L/D)
            CALL EDIFF (XR1,L,V,D,YR1)
            IF (YR1 .GT. 0.0) THEN
               SIGNR = 'POS'
            ELSE
               SIGNR = 'NEG'
            ENDIF
            IF (SIGNL .NE. SIGNR) THEN
               ERIGHT(COUNT) = ETAN(N)
               COUNT = COUNT + 1
               ELEFT(COUNT) = ERIGHT(COUNT-1)
            ELSE
               ELEFT(COUNT) = ETAN(N)
```

```
               ENDIF
               N = N + 1
            ELSE
               XL1 = ELEFT(COUNT) + (COTAN(M)-ELEFT(COUNT))/(500*L/D)
               CALL EDIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0.0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = COTAN(M) - (COTAN(M)-ELEFT(COUNT))/(500*L/D)
               CALL EDIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ERIGHT(COUNT) = COTAN(M)
                  COUNT = COUNT + 1
                  ELEFT(COUNT) = ERIGHT(COUNT-1)
               ELSE
                  ELEFT(COUNT) = COTAN(M)
               ENDIF
               M = M + 1
            ENDIF
         END DO
C
C  Sorting out ranges where solutions for X exist (odd modes)
C
         OLEFT(1) = COTAN(1)
         J = 2
         K = 1
         COUNT = 1
         DO WHILE (COUNT .LE. 160)
            IF (OTAN(K) .LT. COTAN(J)) THEN
               XL1 = OLEFT(COUNT) + (OTAN(K)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = OTAN(K) - (OTAN(K)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ORIGHT(COUNT) = OTAN(K)
                  COUNT = COUNT + 1
                  OLEFT(COUNT) = ORIGHT(COUNT-1)
               ELSE
                  OLEFT(COUNT) = OTAN(K)
               ENDIF
               K = K + 1
            ELSE
```

```
               XL1 = OLEFT(COUNT) + (COTAN(J)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0.0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = COTAN(J) - (COTAN(J)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ORIGHT(COUNT) = COTAN(J)
                  COUNT = COUNT + 1
                  OLEFT(COUNT) = ORIGHT(COUNT-1)
               ELSE
                  OLEFT(COUNT) = COTAN(J)
               ENDIF
               J  = J + 1
            ENDIF
         END DO
C
C  Calculate the value of X's for the even radiation modes :
C
         DO I = 1, ENUM
            P1 = ELEFT(I) + 1E-10
            P5 = ERIGHT(I) - 1E-10
            REPEAT = 'TRUE '
            DO WHILE (REPEAT .EQ. 'TRUE ')
               P3 = (P5-P1)/2 + P1
               P2 = (P3-P1)/2 + P1
               P4 = (P5-P3)/2 + P3
               CALL EDIFF (P1,L,V,D,FP1)
               IF (FP1 .GT. 0) THEN
                  SIGN1 = 'POS'
               ELSE
                  SIGN1 = 'NEG'
               ENDIF
               CALL EDIFF (P2,L,V,D,FP2)
               IF (FP2 .GT. 0) THEN
                  SIGN2 = 'POS'
               ELSE
                  SIGN2 = 'NEG'
               ENDIF
               CALL EDIFF (P4,L,V,D,FP4)
               IF (FP4 .GT. 0) THEN
                  SIGN4 = 'POS'
               ELSE
                  SIGN4 = 'NEG'
               ENDIF
               IF (SIGN2 .EQ. SIGN4) THEN
                  IF (FP2 .GT. FP4) THEN
                     IF (SIGN2 .EQ. 'POS') THEN
                        IF (SIGN1 .EQ. SIGN2) THEN
                           P1 = P4
                        ELSE
```

```
                        P5 = P2
                     ENDIF
                  ELSE IF (SIGN1 .NE. SIGN2) THEN
                           P5 = P2
                        ELSE
                           P1 = P4
                  ENDIF
               ELSE
                  IF (SIGN2 .EQ. 'POS') THEN
                     IF (SIGN1 .NE. SIGN2)THEN
                        P5 = P2
                     ELSE
                        P1 = P4
                     ENDIF
                  ELSE IF (SIGN1 .NE. SIGN2) THEN
                           P5 = P2
                        ELSE
                           P1 = P4
                  ENDIF
               ENDIF
            ELSE
               P1 = P2
               P5 = P4
            ENDIF
            CALL EDIFF (P1,L,V,D,FP1)
            CALL EDIFF (P5,L,V,D,FP5)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  EVENX(I) = P1
               ELSE
                  EVENX(I) = P5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
      END DO
C
C  Calculate the value of X's for the odd radiation modes :
C
      DO I = 1, ONUM
         P1 = OLEFT(I) + 1E-10
         P5 = ORIGHT(I) - 1E-10
         REPEAT = 'TRUE '
         DO WHILE (REPEAT .EQ. 'TRUE ')
            P3 = (P5-P1)/2 + P1
            P2 = (P3-P1)/2 + P1
            P4 = (P5-P3)/2 + P3
            CALL ODIFF (P1,L,V,D,FP1)
            IF (FP1 .GT. 0) THEN
               SIGN1 = 'POS'
            ELSE
               SIGN1 = 'NEG'
            ENDIF
            CALL ODIFF (P2,L,V,D,FP2)
            IF (FP2 .GT. 0) THEN
               SIGN2 = 'POS'
            ELSE
```

```
               SIGN2 = 'NEG'
            ENDIF
            CALL ODIFF (P4,L,V,D,FP4)
            IF (FP4 .GT. 0) THEN
               SIGN4 = 'POS'
            ELSE
               SIGN4 = 'NEG'
            ENDIF
            IF (SIGN2 .EQ. SIGN4) THEN
               IF (FP2 .GT. FP4) THEN
                  IF (SIGN2 .EQ. 'POS') THEN
                     IF (SIGN1 .EQ. SIGN2) THEN
                        P1 = P4
                     ELSE
                        P5 = P2
                     ENDIF
                  ELSE IF (SIGN1 .NE. SIGN2) THEN
                           P5 = P2
                       ELSE
                           P1 = P4
                  ENDIF
               ELSE
                  IF (SIGN2 .EQ. 'POS') THEN
                     IF (SIGN1 .NE. SIGN2) THEN
                        P5 = P2
                     ELSE
                        P1 = P4
                     ENDIF
                  ELSE IF (SIGN1 .NE. SIGN2) THEN
                           P5 = P2
                       ELSE
                          P1 = P4
                  ENDIF
               ENDIF
            ELSE
               P1 = P2
               P5 = P4
            ENDIF
            CALL ODIFF (P1,L,V,D,FP1)
            CALL ODIFF (P5,L,V,D,FP5)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  ODDX(I) = P1
               ELSE
                  ODDX(I) = P5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
      END DO
C
C  Compute the weighting factor for even and odd radiation modes,
C  assume A2 = 1
C
      DO K = 1, 2
         IF (K .EQ. 1) THEN
            PHASE = 0.0
```

```
            NUM = ENUM
            WRITE(2,*) ENUM
         ELSE
            PHASE = PI/2
            NUM = ONUM
            WRITE(2,*) ONUM
         ENDIF
         DO I = 1, NUM
            IF (K .EQ. 1) THEN
               K1X = EVENX(I)/D
               K2X = SQRT(K1X**2+DELK**2)
               K1EVEN(I) = K1X
               K2EVEN(I) = K2X
               WRITE(2,*) K1EVEN(I), K2EVEN(I)
            ELSE
               K1X = ODDX(I)/D
               K2X = SQRT(K1X**2+DELK**2)
               K1ODD(I) = K1X
               K2ODD(I) = K2X
               WRITE(2,*) K1ODD(I), K2ODD(I)
            ENDIF
            PART1 = ((K2X/K1X)**2)*(DSIN(K2X*D+PHASE)**2)/2
            PART2 = L/2 - D + DSIN(2*K1X*(D-L/2))/(2*K1X)
            PART3 = (DCOS(K2X*D+PHASE)**2)/2
            PART4 = L/2 - D - DSIN(2*K1X*(D-L/2))/(2*K1X)
            PART5 = K2X/(4*K1X**2)*DSIN(2*(K2X*D+PHASE))
            PART6 = DCOS(2*K1X*(D-L/2)) - 1
            PART7 = (DSIN(2*(K2X*D-PHASE))+DSIN(2*(K2X*D+PHASE)))
     $              /(2*K2X)
            PART8 = (2*D + PART7)/2
            PART9 = ((K2X/K1X)**2)*(DSIN(K2X*D-PHASE)**2)/2
            PART10 = L/2 - D - DSIN(2*K1X*(L/2-D))/(2*K1X)
            PART11 = (DCOS(K2X*D-PHASE)**2)/2
            PART12 = L/2 - D + DSIN(2*K1X*(L/2-D))/(2*K1X)
            PART13 = K2X/(4*K1X**2)*DSIN(2*(K2X*D-PHASE))
            PART14 = 1 - DCOS(2*K1X*(L/2-D))
            DENO  = PART1*PART2 + PART3*PART4 + PART5*PART6 + PART8 +
     $              PART9*PART10 + PART11*PART12 - PART13*PART14
            NUMER = (DSIN(K2X*D-PHASE) + DSIN(K2X*D+PHASE))/K2X
            IF (K .EQ. 1) THEN
               EC(I) = NUMER/DENO
               WRITE(3,*) EC(I)
            ELSE
               OC(I) = NUMER/DENO
               WRITE(4,*) OC(I)
            ENDIF
         END DO
      END DO
C
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)
      CLOSE (UNIT=4)
      STOP
      END
C
      SUBROUTINE EDIFF (DUM,L,V,D,DIFF)
C
C  Solve for the difference of DUM*(COTAN(DUM*L/(2*D)) and
C  SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2))  (Even modes)
```

```
C
      REAL*8  DUM, DUM1, DUM2, DUM3, DIFF, D, L, V
C
C  D : Half of the thickness of the slab
C  L : Distance between the reflectors (mius the slab)
C  V : Delta K * D
C
      DUM3 = DTAN(DUM*L/(2*D))
      IF (DUM3 .EQ. 0) THEN
         DUM3 = DUM3 + 1E-20
      ENDIF
      DUM1 = DUM*(1/DUM3)
      DUM2 = SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2))
      DIFF = DUM1 - DUM2
C
      RETURN
      END
C
      SUBROUTINE ODIFF (DUM,L,V,D,DIFF)
C
C  Solve for the difference of DUM*(COTAN(DUM*L/(2*D)) and
C  SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2)-PI/2)  (Odd modes)
C
      REAL*8  DUM, DUM1, DUM2, DUM3, DIFF, D, L, V
C
C  D : Half of the thickness of the slab
C  L : Distance between the reflectors (mius the slab)
C  V : Delta K * D
C
      PI = 3.141592654
      DUM3 = DTAN(DUM*L/(2*D))
      IF (DUM3 .EQ. 0) THEN
         DUM3 = DUM3 + 1E-20
      ENDIF
      DUM1 = DUM*(1/DUM3)
      DUM2 = SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2)-PI/2)
      DIFF = DUM1 - DUM2
C
      RETURN
      END
```

```
C  PROGRAM : To determine all the guided modes exist for the single
C           slab waveguide being excited by pencil beam.  Also, the
C           weighting factor for each guided mode is calculated.
C
      CHARACTER*3 SIGN2, SIGN4
      CHARACTER*4 MODE
      CHARACTER*5 REPEAT
      COMPLEX*8   EC(1000), OC(1000), CDUM1, CDUM2, FIELD, NUMER,
     $            DEN0, CDUM3, CDUM4, EF(10000), OF(10000)
      REAL*8      PI, U2, HD, PD, K0, D2, D, N2, N1, LOW, HIGH, L,
     $            LPT, HPT, P1, P2, P3, P4, P5, FP1, FP2, FP4, FP5,
     $            DEN01, DEN02, DEN04, DEN05, LAMBDA0, UPLIM, LOLIM,
     $            EVENP(1000), ODDP(1000), EVENH(1000), ODDH(1000),
     $            DUM1, DUM2, DUM3, DUM4, DUM5, Z, KZ, A2, P, H,
     $            ALPHA, DUM0
      INTEGER     DUMMY, NUM, M, COUNT, COUNT1, COUNT2, ENUM,
     $            ONUM, I
C
      OPEN (UNIT=2,FILE='PH.DAT',STATUS='NEW')
      OPEN (UNIT=3,FILE='GECOEFF.DAT',STATUS='NEW')
      OPEN (UNIT=4,FILE='GOCOEFF.DAT',STATUS='NEW')
C
C  N2 : Refractive index of the slab
C  N1 : Refractive index of the surrounding medium
C  D2 : Thickness of the slab in meters
C  EC : Coefficients for even guided modes
C  OC : Coefficients for odd guided modes
C  ENUM : Number of even guided modes
C  ONUM : Number of odd guided modes
C  LAMBDA0 : Free-space wavelength of light
C  ALPHA : Incidence angle
C  UPLIM, LOLIM : Range where f(x) exists
C  EVENP, EVENH : Contain propagation constants for even modes
C  ODDP, ODDH  : Contain propagation constants for odd modes
C  PH : File contains propagation constant for each guided mode
C  GECOEFF : File contains weighting factor for each even mode
C  GOCOEFF : File contains weighting factor for each odd mode
C
C  Obtain information from the user
C
      TYPE *,' Thickness of the slab (in meters) = ?'
      READ(*,*) D2
      TYPE *,' Distance between the reflectors (minus the slab)?'
      READ(*,*) L
      TYPE *,' Refractive index of the slab ?'
      READ(*,*) N2
      TYPE *,' Refractive index of the surrounding medium?'
      READ(*,*) N1
      TYPE *,' Upper limit of f(X) (in meters) ?'
      READ(*,*) UPLIM
      TYPE *,' Lower limit of f(X) (in meters) ?'
      READ(*,*) LOLIM
      TYPE *,' What is the incidence angle (in degree) ?'
      READ(*,*) ALPHA
C
C  Set up constants for calculations
C
      PI = 3.141592654
```

```
      LAMBDA0 = 0.5E-6
      ALPHA = ALPHA*PI/180
      D = D2/2
      K0 = 2*PI/LAMBDA0
      U2 = (N2**2 - N1**2)*(K0**2)*(D**2)
      COUNT = INT((SQRT(U2))*2/PI)
      WRITE(2,*) D, N2, LAMBDA0
C
C  To find the existed guided modes
C
C       Find the number of existed guided modes (even/odd)
C
      ENUM = 0
      ONUM = 0
      DO M = 0, COUNT
         LOW = M/2.0
         HIGH = LOW + 0.5
         LPT = LOW*PI
         HPT = HIGH*PI
         DUMMY = INT(LOW)
         IF ((DUMMY-LOW) .EQ. 0) THEN
            MODE = 'EVEN'
            ENUM = ENUM + 1
         ELSE
            MODE = 'ODD '
            ONUM = ONUM + 1
         ENDIF
         NUM = DUMMY + 1
C
C          Find the propagation constants for each guided mode
C
         P1 = LPT + 1E-20
         P5 = HPT
         REPEAT = 'TRUE '
         DO WHILE (REPEAT .EQ. 'TRUE ')
            P3 = (P5-P1)/2 + P1
            P2 = (P3-P1)/2 + P1
            P4 = (P5-P3)/2 + P3
            IF (MODE .EQ. 'EVEN') THEN
               DEN02 = DTAN(P2)**2
               DEN04 = DTAN(P4)**2
            ELSE
               DEN02 = 1/(DTAN(P2)**2)
               DEN04 = 1/(DTAN(P4)**2)
            ENDIF
            FP2 = U2 - (P2**2)*(1+DEN02)
            FP4 = U2 - (P4**2)*(1+DEN04)
            IF (FP2 .GT. 0) THEN
               SIGN2 = 'POS'
            ELSE
               SIGN2 = 'NEG'
            ENDIF
            IF (FP4 .GT. 0) THEN
               SIGN4 = 'POS'
            ELSE
               SIGN4 = 'NEG'
            ENDIF
            IF (SIGN2 .EQ. SIGN4) THEN
              IF (FP2 .GT. FP4) THEN
```

```
                  IF (SIGN2 .EQ. 'POS') THEN
                     P1 = P4
                  ELSE
                     P5 = P2
                  ENDIF
               ELSE
                  IF (SIGN .EQ. 'POS') THEN
                     P5 = P2
                  ELSE
                     P1 = P4
                  ENDIF
               ENDIF
            ELSE
               P1 = P2
               P5 = P4
            ENDIF
            IF (MODE .EQ. 'EVEN') THEN
               DEN01 = DTAN(P1)**2
               DEN05 = DTAN(P5)**2
            ELSE
               DEN01 = 1/(DTAN(P1)**2)
               DEN05 = 1/(DTAN(P5)**2)
            ENDIF
            FP1 = U2 - (P1**2)*(1+DEN01)
            FP5 = U2 - (P5**2)*(1+DEN05)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  HD = P1
               ELSE
                  HD = P5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
         PD = SQRT(U2-HD**2)
         P = PD/D
         H = HD/D
         IF (MODE .EQ. 'EVEN') THEN
            EVENP(NUM) = P
            EVENH(NUM) = H
         ELSE
            ODDP(NUM) = P
            ODDH(NUM) = H
         ENDIF
      END DO
C
C  Write the number of odd and even guided modes and their corresponding
C  propagation constants to PH.DAT
C
      COUNT1 = NUM
      COUNT2 = COUNT1
      IF (MODE .EQ. 'EVEN') THEN
         COUNT2 = NUM-1
      ENDIF
      DO M = 1, COUNT1
         WRITE(2,*) EVENP(M), EVENH(M)
      END DO
```

```
      DO 50 M = 1, COUNT2
         WRITE(2,*) ODDP(M), ODDH(M)
      END DO
C
C  Compute the weighting factor for the even guided modes
C
      WRITE(3,*) ENUM
      DO I = 1, ENUM
         DUM1 = EVENH(I) * D
         A = DCOS(EVENH(I)*D)
         A2 = A**2
         DUM2 = A2/EVENP(I) + D + DSIN(2*EVENH(I)*D)/(2*EVENH(I))
         DUM3 = 2*PI*DSIN(ALPHA)/LAMBDA0
         DEN0 = CMPLX((EVENH(I)**2 - DUM3**2)*DUM2)
         DUM4 = EVENH(I)*UPLIM
         DUM5 = EVENH(I)*LOLIM
         CDUM1 = CMPLX(EVENH(I)*DSIN(DUM4),DUM3*DCOS(DUM4))
         CDUM2 = CMPLX(EVENH(I)*DSIN(DUM5),DUM3*DCOS(DUM5))
         CDUM3 = CEXP(CMPLX(DUM0,DUM3*UPLIM))*CDUM1
         CDUM4 = CEXP(CMPLX(DUM0,DUM3*LOLIM))*CDUM2
         NUMER = CDUM3 - CDUM4
         EC(I) = NUMER/DEN0
         WRITE(3,*) EC(I)
      END DO
C
C  Compute the weighting factor for the odd guided modes
C
      WRITE(4,*) ONUM
      DO I = 1, ONUM
         DUM1 = ODDH(I) * D
         A = DSIN(ODDH(I)*D)
         A2 = A**2
         DUM2 = A2/ODDP(I) + D - DSIN(2*ODDH(I)*D)/(2*ODDH(I))
         DUM3 = 2*PI*DSIN(ALPHA)/LAMBDA0
         DEN0 = CMPLX((ODDH(I)**2 - DUM3**2)*DUM2)
         DUM4 = ODDH(I)*UPLIM
         DUM5 = ODDH(I)*LOLIM
         CDUM1 = CMPLX(-1*ODDH(I)*DCOS(DUM4),DUM3*DSIN(DUM4))
         CDUM2 = CMPLX(-1*ODDH(I)*DCOS(DUM5),DUM3*DSIN(DUM5))
         CDUM3 = CEXP(CMPLX(DUM0,DUM3*UPLIM))*CDUM1
         CDUM4 = CEXP(CMPLX(DUM0,DUM3*LOLIM))*CDUM2
         NUMER = CDUM3 - CDUM4
         OC(I) = NUMER/DEN0
         WRITE(4,*) OC(I)
      END DO
C
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)
      CLOSE (UNIT=4)
      STOP
      END
```

```
C  PROGRAM : To determine the radiation modes for the single slab
C           waveguide being excited by pencil beam. Also,the weight-
C           ing factor for each radiation mode used is calculated.
C
      CHARACTER*3 SIGN1, SIGN2, SIGN4, SIGNL, SIGNR, LSLOPE, RSLOPE
      CHARACTER*4 MODE, SIGN
      CHARACTER*5 REPEAT
      COMPLEX*8   EC(1000), OC(1000), RFIELD, CDUM, CDUM1, CDUM2,
     $            CDUM3, CDUM4, CDUM5, CDUM6, NUMER, DENO
      REAL*8      V, DELK, D2, D, L, N2, N1, LAMBDA0, START,
     $            EVENX(1000), ODDX(1000), COTAN(1000), ETAN(1000),
     $            OTAN(1000), ELEFT(1000), ERIGHT(1000), OLEFT(1000),
     $            ORIGHT(1000), P1, P2, P3, P4, P5, FP1, FP2, FP4,
     $            FP5, DUM1, DUM2, DUM, SLOPE, XL1, XL2, XR1, XR2,
     $            YL1, YL2, YR1, YR2, PART1, PART2, PART3, PART4,
     $            PART5, PART6, PART7, PART8, PART9, PART10, PART11,
     $            PART12, PART13, PART14, K1EVEN(1000), K1ODD(1000),
     $            K2EVEN(1000), K2ODD(1000), DUMMY, DUM0, DUM3, DUM4,
     $            ALPHA, UPLIM, LOLIM, K1X, K2X, PHASE
      INTEGER     I, M ,N, J, K, COUNT, NUM, ENUM, ONUM
C
      OPEN (UNIT=2,FILE='K1K2.DAT',STATUS='NEW')
      OPEN (UNIT=3,FILE='RECOEFF.DAT',STATUS='NEW')
      OPEN (UNIT=4,FILE='ROCOEFF.DAT',STATUS='NEW')
C
C  D2 : Thickness of the slab in meters
C  N2 : Refractive index of the slab
C  N1 : Refractive index of the surrounding medium
C  ENUM : Number of even radiation modes used
C  ONUM : Number of odd radiation modes used
C  ALPHA : Incidence angle
C  LAMBDA0 : Free-space wavelength of light
C  UPLIM, LOLIM : Range where f(x) exists
C  K1K2 : File contains propagation constant for each radiation mode
C  RECOEFF : File contains weighting factor for each even radiaton mode
C  ROCOEFF : File contains weighting factor for each odd radiaton mode
C
C  Obtain information from user
C
      TYPE *,' Thickness of the slab (in meter) ?'
      READ(*,*) D2
      TYPE *,' Distance between the reflectors (minus the slab) ?'
      READ(*,*) L
      TYPE *,' Refractive index of the slab ?'
      READ(*,*) N2
      TYPE *,' Refractive index of the surrounding medium ?'
      READ(*,*) N1
      TYPE *,' Upper limit of f(X) (in meter) ?'
      READ(*,*) UPLIM
      TYPE *,' Lower limit of f(X) (in meter) ?'
      READ(*,*) LOLIM
      TYPE *,' What is the incidence angle (in degree) ?'
      READ(*,*) ALPHA
      TYPE *,' Number of even modes to be used (<1000) = ?'
      READ(*,*) ENUM
      TYPE *,' Number of odd modes to be used (<1000) = ?'
      READ(*,*) ONUM
C
```

```
C  Set up constants for calculations
C
      PI = 3.141592654
      DUM0 = 0.0
      D = D2/2
      ALPHA = ALPHA*PI/180
      LAMBDA0 = 0.5E-6
      DELK = (2*PI/LAMBDA0)*(SQRT(N2**2-N1**2))
      V = DELK*D
      WRITE(2,*) D, N2, LAMBDA0
C
C  Finding X at which X*COTAN(X*L/2D) goes to infinity :
C
      START = 0.0
      DO I = 1, 190
         COTAN(I) = START + (2*D*PI/L)*(I-1)
      END DO
C
C  Finding X at which SQRT(X**2+V**2)*TAN(SQRT(X**2+V**2))
C  goes to infinity:
C
C  A)  Even radiation modes :
C
      IF (V .LE. PI/2) THEN
         START = PI/2
      ELSE
         START = INT((V-PI/2)/PI)*PI + 3*PI/2
      ENDIF
      DO I = 1, 190
         ETAN(I) = SQRT((START + PI*(I-1))**2 - V**2)
      END DO
C
C  B)  Odd radiation modes :
C
      IF (V .LE. PI) THEN
         START = PI/2
      ELSE
         START = (INT(V/PI)+1)*PI - PI/2
      ENDIF
      DO I = 1, 190
         OTAN(I) = SQRT(((START + PI*(I-1))+PI/2)**2 - V**2)
      END DO
C
C  Sorting out ranges where solutions for X exist (Even modes)
C
      ELEFT(1) = COTAN(1)
      M = 2
      N = 1
      COUNT = 1
      DO WHILE (COUNT .LE. 160)
         IF (ETAN(N) .LT. COTAN(M)) THEN
            XL1 = ELEFT(COUNT) + (ETAN(N)-ELEFT(COUNT))/(500*L/D)
            CALL EDIFF (XL1,L,V,D,YL1)
            IF (YL1 .GT. 0) THEN
               SIGNL = 'POS'
            ELSE
               SIGNL = 'NEG'
            ENDIF
            XR1 = ETAN(N) - (ETAN(N)-ELEFT(COUNT))/(500*L/D)
```

```
               CALL EDIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ERIGHT(COUNT) = ETAN(N)
                  COUNT = COUNT + 1
                  ELEFT(COUNT) = ERIGHT(COUNT-1)
               ELSE
                  ELEFT(COUNT) = ETAN(N)
               ENDIF
               N = N + 1
            ELSE
               XL1 = ELEFT(COUNT) + (COTAN(M)-ELEFT(COUNT))/(500*L/D)
               CALL EDIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0.0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = COTAN(M) - (COTAN(M)-ELEFT(COUNT))/(500*L/D)
               CALL EDIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ERIGHT(COUNT) = COTAN(M)
                  COUNT = COUNT + 1
                  ELEFT(COUNT) = ERIGHT(COUNT-1)
               ELSE
                  ELEFT(COUNT) = COTAN(M)
               ENDIF
               M = M + 1
            ENDIF
         END DO
C
C  Sorting out ranges where solutions for X exist (Odd modes)
C
         OLEFT(1) = COTAN(1)
         J = 2
         K = 1
         COUNT = 1
         DO WHILE (COUNT .LE. 160)
            IF (OTAN(K) .LT. COTAN(J)) THEN
               XL1 = OLEFT(COUNT) + (OTAN(K)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = OTAN(K) - (OTAN(K)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
```

```
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ORIGHT(COUNT) = OTAN(K)
                  COUNT = COUNT + 1
                  OLEFT(COUNT) = ORIGHT(COUNT-1)
               ELSE
                  OLEFT(COUNT) = OTAN(K)
               ENDIF
               K = K + 1
            ELSE
               XL1 = OLEFT(COUNT) + (COTAN(J)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0.0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = COTAN(J) - (COTAN(J)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ORIGHT(COUNT) = COTAN(J)
                  COUNT = COUNT + 1
                  OLEFT(COUNT) = ORIGHT(COUNT-1)
               ELSE
                  OLEFT(COUNT) = COTAN(J)
               ENDIF
               J  = J + 1
            ENDIF
         END DO
C
C  Calculate the value of X's for the even radiation modes :
C
         DO I = 1, ENUM
            P1 = ELEFT(I) + 1E-10
            P5 = ERIGHT(I) - 1E-10
            REPEAT = 'TRUE '
            DO WHILE (REPEAT .EQ. 'TRUE ')
               P3 = (P5-P1)/2 + P1
               P2 = (P3-P1)/2 + P1
               P4 = (P5-P3)/2 + P3
               CALL EDIFF (P2,L,V,D,FP2)
               IF (FP2 .GT. 0) THEN
                  SIGN2 = 'POS'
               ELSE
                  SIGN2 = 'NEG'
               ENDIF
               CALL EDIFF (P4,L,V,D,FP4)
               IF (FP4 .GT. 0) THEN
                  SIGN4 = 'POS'
               ELSE
                  SIGN4 = 'NEG'
               ENDIF
```

```
            IF (SIGN2 .EQ. SIGN4) THEN
               IF (FP2 .GT. FP4) THEN
                  IF (SIGN2 .EQ. 'POS') THEN
                     P1 = P4
                  ELSE
                     P5 = P2
                  ENDIF
               ELSE
                  IF (SIGN2 .EQ. 'POS') THEN
                     P5 = P2
                  ELSE
                     P1 = P4
                  ENDIF
               ENDIF
            ELSE
               P1 = P2
               P5 = P4
            ENDIF
            CALL EDIFF (P1,L,V,D,FP1)
            CALL EDIFF (P5,L,V,D,FP5)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  EVENX(I) = P1
               ELSE
                  EVENX(I) = P5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
      END DO
C
C  Calculate the value of X's for the odd radiation modes :
C
      DO I = 1, ONUM
         P1 = OLEFT(I) + 1E-10
         P5 = ORIGHT(I) - 1E-10
         REPEAT = 'TRUE '
         DO WHILE (REPEAT .EQ. 'TRUE ')
            P3 = (P5-P1)/2 + P1
            P2 = (P3-P1)/2 + P1
            P4 = (P5-P3)/2 + P3
            CALL ODIFF (P2,L,V,D,FP2)
            IF (FP2 .GT. 0) THEN
               SIGN2 = 'POS'
            ELSE
               SIGN2 = 'NEG'
            ENDIF
            CALL ODIFF (P4,L,V,D,FP4)
            IF (FP4 .GT. 0) THEN
               SIGN4 = 'POS'
            ELSE
               SIGN4 = 'NEG'
            ENDIF
            IF (SIGN2 .EQ. SIGN4) THEN
               IF (FP2 .GT. FP4) THEN
                  IF (SIGN2 .EQ. 'POS') THEN
                     P1 = P4
```

```
                  ELSE
                     P5 = P2
                  ENDIF
               ELSE
                  IF (SIGN2 .EQ. 'POS') THEN
                     P5 = P2
                  ELSE
                     P1 = P4
                  ENDIF
               ENDIF
            ELSE
               P1 = P2
               P5 = P4
            ENDIF
            CALL ODIFF (P1,L,V,D,FP1)
            CALL ODIFF (P5,L,V,D,FP5)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  ODDX(I) = P1
               ELSE
                  ODDX(I) = P5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
      END DO
C
C  Compute the weighting factor for even and odd radiation modes,
C  assume A2 = 1
C
      DO K = 1, 2
         IF (K .EQ. 1) THEN
            PHASE = 0.0
            NUM = ENUM
            WRITE(2,*) ENUM
         ELSE
            PHASE = PI/2
            NUM = ONUM
            WRITE(2,*) ONUM
         ENDIF
         DO I = 1, NUM
            IF (K .EQ. 1) THEN
               K1X = EVENX(I)/D
               K2X = SQRT(K1X**2+DELK**2)
               K1EVEN(I) = K1X
               K2EVEN(I) = K2X
               WRITE(2,*) K1EVEN(I), K2EVEN(I)
            ELSE
               K1X = ODDX(I)/D
               K2X = SQRT(K1X**2+DELK**2)
               K1ODD(I) = K1X
               K2ODD(I) = K2X
               WRITE(2,*) K1ODD(I), K2ODD(I)
            ENDIF
            PART1 = ((K2X/K1X)**2)*(DSIN(K2X*D+PHASE)**2)/2
            PART2 = L/2 - D + DSIN(2*K1X*(D-L/2))/(2*K1X)
            PART3 = (DCOS(K2X*D+PHASE)**2)/2
```

```
               PART4 = L/2 - D - DSIN(2*K1X*(D-L/2))/(2*K1X)
               PART5 = K2X/(4*K1X**2)*DSIN(2*(K2X*D+PHASE))
               PART6 = DCOS(2*K1X*(D-L/2)) - 1
               PART7 = (DSIN(2*(K2X*D-PHASE))+DSIN(2*(K2X*D+PHASE)))
     $                 /(2*K2X)
               PART8 = (2*D + PART7)/2
               PART9 = ((K2X/K1X)**2)*(DSIN(K2X*D-PHASE)**2)/2
               PART10 = L/2 - D - DSIN(2*K1X*(L/2-D))/(2*K1X)
               PART11 = (DCOS(K2X*D-PHASE)**2)/2
               PART12 = L/2 - D + DSIN(2*K1X*(L/2-D))/(2*K1X)
               PART13 = K2X/(4*K1X**2)*DSIN(2*(K2X*D-PHASE))
               PART14 = 1 - DCOS(2*K1X*(L/2-D))
               DEN0 = CMPLX(PART1*PART2 + PART3*PART4 + PART5*PART6 +
     $                PART8+PART9*PART10+PART11*PART12-PART13*PART14)
               DUM1 = 2*PI*DSIN(ALPHA)/LAMBDA0
               DUM2 = K2X**2 - DUM1**2
               DUM3 = K2X*UPLIM
               DUM4 = K2X*LOLIM
               CDUM1 = CMPLX(K2X*DSIN(DUM3),DUM1*DCOS(DUM3))
               CDUM2 = CMPLX(K2X*DSIN(DUM4),DUM1*DCOS(DUM4))
               CDUM3 = CEXP(CMPLX(DUM0,DUM1*UPLIM))*CDUM1
               CDUM4 = CEXP(CMPLX(DUM0,DUM1*LOLIM))*CDUM2
               CDUM5 = CMPLX(DCOS(PHASE)/DUM2)*(CDUM3 - CDUM4)
               CDUM1 = CMPLX(-1*K2X*DCOS(DUM3),DUM1*DSIN(DUM3))
               CDUM2 = CMPLX(-1*K2X*DCOS(DUM4),DUM1*DSIN(DUM4))
               CDUM3 = CEXP(CMPLX(DUM0,DUM1*UPLIM))*CDUM1
               CDUM4 = CEXP(CMPLX(DUM0,DUM1*LOLIM))*CDUM2
               CDUM6 = CMPLX(DSIN(PHASE)/DUM2)*(CDUM3 - CDUM4)
               NUMER = CDUM5 + CDUM6
               IF (K .EQ. 1) THEN
                  EC(I) = NUMER/DEN0
                  WRITE(3,*) EC(I)
               ELSE
                  OC(I) = NUMER/DEN0
                  WRITE(4,*) OC(I)
               ENDIF
            END DO
         END DO
C
         CLOSE (UNIT=2)
         CLOSE (UNIT=3)
         CLOSE (UNIT=4)
         STOP
         END
C
         SUBROUTINE EDIFF (DUM,L,V,D,DIFF)
C
C  Solve for the difference of DUM*(COTAN(DUM*L/(2*D)) and
C  SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2)) (Even modes)
C
         REAL*8  DUM, DUM1, DUM2, DUM3, DIFF, D, L, V
C
C  D : Half of the thickness of the slab
C  L : Distance between the reflectors (minus the slab)
C  V : Delta K * D
C
         DUM3 = DTAN(DUM*L/(2*D))
         IF (DUM3 .EQ. 0) THEN
            DUM3 = DUM3 + 1E-20
```

```
      ENDIF
      DUM1 = DUM*(1/DUM3)
      DUM2 = SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2))
      DIFF = DUM1 - DUM2
C
      RETURN
      END
C
      SUBROUTINE ODIFF (DUM,L,V,D,DIFF)
C
C  Solve for the difference of DUM*(COTAN(DUM*L/(2*D)) and
C  SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2-PI/2)) (Odd modes)
C
      REAL*8  DUM, DUM1, DUM2, DUM3, DIFF, D, L, V
C
C  D : Half of the thickness of the slab
C  L : Distance between the reflectors (minus the slab)
C  V : Delta K * D
C
      PI = 3.141592654
      DUM3 = DTAN(DUM*L/(2*D))
      IF (DUM3 .EQ. 0) THEN
         DUM3 = DUM3 + 1E-20
      ENDIF
      DUM1 = DUM*(1/DUM3)
      DUM2 = SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2)-PI/2)
      DIFF = DUM1 - DUM2
C
      RETURN
      END
```

```
C  PROGRAM : To determine all the guided modes exist for the single
C           slab waveguide being excited by plane wave and have a
C           lens placed against the slab waveguide at z = 0.  Also,
C           the weighting factor for each guided mode is calculated.
C
      CHARACTER*3  SIGN2, SIGN4
      CHARACTER*4  MODE, MORE
      COMPLEX*8    EC(1000), OC(1000), CDUM, FIELD, ENUMER, EDENO,
     $             ONUMER, ODENO, EF(10000), OF(10000)
      CHARACTER*5  REPEAT
     $             DUMMY, M , COUNT, COUNT1, COUNT2
      REAL*8       PI, U2, HD, PD, KO, D2, D, N2, N1, LOW, HIGH, L,
     $             LPT, HPT, P1, P2, P3, P4, P5, FP1, FP2, FP4, FP5,
     $             DENO1, DENO2, DENO4, DENO5, LAMBDAO, UPLIM, LOLIM,
     $             EVENP(1000), ODDP(1000), EVENH(1000), ODDH(1000),
     $             STEP, Z, K, KZ, A2, X, LRANGE, RRANGE, A, B, P,
     $             SEP, DUMO, ESO, ES1, ES2, OSO, OS1, OS2, F, H,
     $             EAREA1, EAREA2, OAREA1, OAREA2
      INTEGER      ENUM, ONUM, MODNUM, I, J, N, NUM, POINTS
C
      OPEN (UNIT=2,FILE='PH.DAT',STATUS='NEW')
      OPEN (UNIT=3,FILE='GECOEFF.DAT',STATUS='NEW')
      OPEN (UNIT=4,FILE='GOCOEFF.DAT',STATUS='NEW')
C
C  F : Focal length of the lens
C  N2 : Refractive index of the slab
C  N1 : Refractive index of the surrounding medium
C  D2 : Thickness of the slab in meters
C  EC : Coefficients for even guided modes
C  OC : Coefficients for even guided modes
C  LAMBDAO : Free-space wavelength of light
C  UPLIM, LOLIM : Range where f(X) exists
C  EVENP, EVENH : Contain propagation constants for even modes
C  ODDP,  ODDH  : Contain propagation constants for odd modes
C  GECOEFF : File contains propagation constant for each even mode
C  GOCOEFF : File contains propagation constant for each odd mode
C
C  Obtain information from the user
C
      TYPE *,' Thickness of the slab (in meters) = ?'
      READ(*,*) D2
      TYPE *,' Distance between the reflectors (minus the slab)? '
      READ(*,*) L
      TYPE *,' Refractive index of the slab ?'
      READ(*,*) N2
      TYPE *,' Refractive index of the surrounding medium ?'
      READ(*,*) N1
      TYPE *,' Upper limit of f(X) (in meters) ?'
      READ(*,*) UPLIM
      TYPE *,' Lower limit of f(X) (in meters) ?'
      READ(*,*) LOLIM
      TYPE *,' What is focal length of the lens (in meters) ?'
      READ(*,*) F
C
C  Set up constants for calculations
C
      PI = 3.141592654
      LAMBDAO = 0.5E-6
```

```
      DUMO = 0.0
      D = D2/2
      KO = 2*PI/LAMBDAO
      U2 = (N2**2 - N1**2)*(KO**2)*(D**2)
      COUNT = INT((SQRT(U2))*2/PI)
      WRITE(2,*) D, N2, LAMBDAO
C
C  To find the existed guided modes
C
C     Find the number of existed guided modes (even/odd)
C
      ENUM = 0
      ONUM = 0
      DO M = 0, COUNT
         LOW = M/2.0
         HIGH = LOW + 0.5
         LPT = LOW*PI
         HPT = HIGH*PI
         DUMMY = INT(LOW)
         IF ((DUMMY-LOW) .EQ. 0) THEN
            MODE = 'EVEN'
            ENUM = ENUM + 1
         ELSE
            MODE = 'ODD '
            ONUM = ONUM + 1
         ENDIF
         NUM = DUMMY + 1
C
C        Find the propagation constants for each guided mode
C
         P1 = LPT + 1E-20
         P5 = HPT
         REPEAT = 'TRUE '
         DO WHILE (REPEAT .EQ. 'TRUE ')
            P3 = (P5-P1)/2 + P1
            P2 = (P3-P1)/2 + P1
            P4 = (P5-P3)/2 + P3
            IF (MODE .EQ. 'EVEN') THEN
               DEN02 = DTAN(P2)**2
               DEN04 = DTAN(P4)**2
            ELSE
               DEN02 = 1/(DTAN(P2)**2)
               DEN04 = 1/(DTAN(P4)**2)
            ENDIF
            FP2 = U2 - (P2**2)*(1+DEN02)
            FP4 = U2 - (P4**2)*(1+DEN04)
            IF (FP2 .GT. 0) THEN
               SIGN2 = 'POS'
            ELSE
               SIGN2 = 'NEG'
            ENDIF
            IF (FP4 .GT. 0) THEN
               SIGN4 = 'POS'
            ELSE
               SIGN4 = 'NEG'
            ENDIF
            IF (SIGN2 .EQ. SIGN4) THEN
              IF (FP2 .GT. FP4) THEN
                IF (SIGN2 .EQ. 'POS') THEN
```

```
                  P1 = P4
                ELSE
                  P5 = P2
                ENDIF
              ELSE
                IF (SIGN .EQ. 'POS') THEN
                  P5 = P2
                ELSE
                  P1 = P4
                ENDIF
              ENDIF
            ELSE
              P1 = P2
              P5 = P4
            ENDIF
            IF (MODE .EQ. 'EVEN') THEN
               DEN01 = DTAN(P1)**2
               DEN05 = DTAN(P5)**2
            ELSE
               DEN01 = 1/(DTAN(P1)**2)
               DEN05 = 1/(DTAN(P5)**2)
            ENDIF
            FP1 = U2 - (P1**2)*(1+DEN01)
            FP5 = U2 - (P5**2)*(1+DEN05)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  HD = P1
               ELSE
                  HD = P5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
         PD = SQRT(U2-HD**2)
         P = PD/D
         H = HD/D
         IF (MODE .EQ. 'EVEN') THEN
            EVENP(NUM) = P
            EVENH(NUM) = H
         ELSE
            ODDP(NUM) = P
            ODDH(NUM) = H
         ENDIF
      END DO
C
C  Write the number of odd and even guided modes and their
C  corresponding propagation constants to PH.DAT
C
      COUNT1 = NUM
      COUNT2 = COUNT1
      IF (MODE .EQ. 'EVEN') THEN
         COUNT2 = NUM-1
      ENDIF
      DO M = 1, COUNT1
         WRITE(2,*) EVENP(M), EVENH(M)
      END DO
      DO M = 1, COUNT2
```

```
            WRITE(2,*) ODDP(M), ODDH(M)
         END DO
C
C  Compute the weighting factor for guided modes
C
         K = 2*PI/LAMBDA0
         NUM = 25
         SEP = (UPLIM-LOLIM)/(2*NUM)
         POINTS = 400
         STEP = (RRANGE-LRANGE)/POINTS
         WRITE(3,*) ENUM
         WRITE(4,*) ONUM
         IF (ENUM .EQ. ONUM) THEN
            MODNUM = ENUM
            MORE = 'NO  '
         ELSE
            MODNUM = ONUM
            MORE = 'YES '
         ENDIF
         DO I = 1, MODNUM
            A = DCOS(EVENH(I)*D)
            A2 = A**2
            EDEN0 = CMPLX(A2/EVENP(I) + D + DSIN(2*EVENH(I)*D)/
     $              (2*EVENH(I)))
            A = DSIN(ODDH(I)*D)
            A2 = A**2
            ODEN0 = CMPLX(A2/ODDP(I) + D - DSIN(2*ODDH(I)*D)/
     $              (2*ODDH(I)))
            ES0 = DCOS(K*(LOLIM**2)/(2*F))*DCOS(EVENH(I)*LOLIM) +
     $            DCOS(K*(UPLIM**2)/(2*F))*DCOS(EVENH(I)*UPLIM)
            ES1 = 0
            ES2 = 0
            OS0 = DCOS(K*(LOLIM**2)/(2*F))*DSIN(ODDH(I)*LOLIM) +
     $            DCOS(K*(UPLIM**2)/(2*F))*DSIN(ODDH(I)*UPLIM)
            OS1 = 0
            OS2 = 0
            DO J = 1, 2*NUM-1, 2
               X = LOLIM + J*SEP
               ES1 = DCOS(K*(X**2)/(2*F))*DCOS(EVENH(I)*X) + ES1
               OS1 = DCOS(K*(X**2)/(2*F))*DSIN(ODDH(I)*X) + OS1
            END DO
            DO J = 2, 2*NUM-2, 2
               X = LOLIM + J*SEP
               ES2 = DCOS(K*(X**2)/(2*F))*DCOS(EVENH(I)*X) + ES2
               OS2 = DCOS(K*(X**2)/(2*F))*DSIN(ODDH(I)*X) + OS2
            END DO
            EAREA1 = (SEP/3)*(ES0 + 4*ES1 + 2*ES2)
            OAREA1 = (SEP/3)*(OS0 + 4*OS1 + 2*OS2)
            ES0 = DSIN(K*(LOLIM**2)/(2*F))*DCOS(EVENH(I)*LOLIM) +
     $            DSIN(K*(UPLIM**2)/(2*F))*DCOS(EVENH(I)*UPLIM)
            ES1 = 0
            ES2 = 0
            OS0 = DSIN(K*(LOLIM**2)/(2*F))*DSIN(ODDH(I)*LOLIM) +
     $            DSIN(K*(UPLIM**2)/(2*F))*DSIN(ODDH(I)*UPLIM)
            OS1 = 0
            OS2 = 0
            DO J = 1, 2*NUM-1, 2
               X = LOLIM + J*SEP
               ES1 = DSIN(K*(X**2)/(2*F))*DCOS(EVENH(I)*X) + ES1
```

```
            OS1 = DSIN(K*(X**2)/(2*F))*DSIN(ODDH(I)*X) + OS1
         END DO
         DO J = 2, 2*NUM-2, 2
            X = LOLIM + J*SEP
            ES2 = DSIN(K*(X**2)/(2*F))*DCOS(EVENH(I)*X) + ES2
            OS2 = DSIN(K*(X**2)/(2*F))*DSIN(ODDH(I)*X) + OS2
         END DO
         EAREA2 = (SEP/3)*(ES0 + 4*ES1 + 2*ES2)
         OAREA2 = (SEP/3)*(OS0 + 4*OS1 + 2*OS2)
         ENUMER = CMPLX(EAREA1,-1*EAREA2)
         ONUMER = CMPLX(OAREA1,-1*OAREA2)
         EC(I) = ENUMER/EDEN0
         OC(I) = ONUMER/ODEN0
         WRITE(3,*) EC(I)
         WRITE(4,*) OC(I)
      END DO
C
      IF (MORE .EQ. 'YES ') THEN
        DO I = MODNUM, ENUM
         A = DCOS(EVENH(I)*D)
         A2 = A**2
         EDEN0 = CMPLX(A2/EVENP(I)+D+DSIN(2*EVENH(I)*D)/(2*EVENH(I)))
         ES0 = DCOS(K*(LOLIM**2)/(2*F))*DCOS(EVENH(I)*LOLIM) +
     $         DCOS(K*(UPLIM**2)/(2*F))*DCOS(EVENH(I)*UPLIM)
         ES1 = 0
         ES2 = 0
         DO J = 1, 2*NUM-1, 2
            X = LOLIM + J*SEP
            ES1 = DCOS(K*(X**2)/(2*F))*DCOS(EVENH(I)*X) + ES1
         END DO
         DO J = 2, 2*NUM-2, 2
            X = LOLIM + J*SEP
            ES2 = DCOS(K*(X**2)/(2*F))*DCOS(EVENH(I)*X) + ES2
         END DO
         EAREA1 = (SEP/3)*(ES0 + 4*ES1 + 2*ES2)
         ES0 = DSIN(K*(LOLIM**2)/(2*F))*DCOS(EVENH(I)*LOLIM) +
     $         DSIN(K*(UPLIM**2)/(2*F))*DCOS(EVENH(I)*UPLIM)
         ES1 = 0
         ES2 = 0
         DO J = 1, 2*NUM-1, 2
            X = LOLIM + J*SEP
            ES1 = DSIN(K*(X**2)/(2*F))*DCOS(EVENH(I)*X) + ES1
         END DO
         DO J = 2, 2*NUM-2, 2
            X = LOLIM + J*SEP
            ES2 = DSIN(K*(X**2)/(2*F))*DCOS(EVENH(I)*X) + ES2
         END DO
         EAREA2 = (SEP/3)*(ES0 + 4*ES1 + 2*ES2)
         ENUMER = CMPLX(EAREA1,-1*EAREA2)
         EC(I) = ENUMER/EDEN0
         WRITE(3,*) EC(I)
        END DO
      ENDIF
C
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)
      CLOSE (UNIT=4)
      STOP
      END
```

```
C  PROGRAM : To determine the radiation modes for the single slab
C           waveguide being excited by plane wave and a lens is
C           placed against the slab waveguide at z=0.  Also, the
C           weighting factor for each radiation mode used is
C           calculated.
C
      CHARACTER*3 SIGN1, SIGN2, SIGN4, SIGNL, SIGNR, LSLOPE, RSLOPE
      CHARACTER*4 MODE, SIGN
      CHARACTER*5 REPEAT
      COMPLEX*8   EC(1000), OC(1000), RFIELD, CDUM, NUMER, DENO
      REAL*8  EVENX(1000), ODDX(1000), V, DELK, D2, D, L, N2, N1,
     $        COTAN(1000), ETAN(1000), OTAN(1000), LAMBDA0, START,
     $        ELEFT(1000), ERIGHT(1000), OLEFT(1000), ORIGHT(1000),
     $        P1, P2, P3, P4, P5, FP1, FP2, FP4, FP5, DUM1, DUM2,
     $        DUM, SLOPE, XL1, XL2, XR1, XR2, YL1, YL2, YR1, YR2,
     $        UPLIM, LOLIM, F, K1X, K2X, PHASE, S0, S1, S2, SEP,
     $        AREA1, AREA2, AREA3, AREA4, DUMMY, DUM0, K, PART1,
     $        PART2, PART3, PART4, PART5, PART6, PART7, PART8,
     $        PART9, PART10, PART11, PART12, PART13, PART14,
     $        K1EVEN(1000), K1ODD(1000), K2EVEN(1000), K2ODD(1000)
      INTEGER I, M ,N, J, KK, COUNT, NUM, ENUM, ONUM
C
      OPEN (UNIT=2,FILE='K1K2.DAT',STATUS='NEW')
      OPEN (UNIT=3,FILE='RECOEFF.DAT',STATUS='NEW')
      OPEN (UNIT=4,FILE='ROCOEFF.DAT',STATUS='NEW')
C
C  D2 : Thickness of the slab
C  N2 : Refractive index of the slab
C  N1 : Refractive index of the surrounding medium
C  F : Focal length of the lens
C  LAMBDA0 : Free-space wavelength of light
C  UPLIM, LOLIM : Range where f(X) exists
C  K1K2 : File contains propagation constants for each radiation mode
C  RECOEFF : File contains weighting factor for each even radiation mode
C  ROCOEFF : File contains weighting factor for each odd radiation mode
C
C  Obtain information from user
C
      TYPE *,' Thickness of the slab (in meter) ?'
      READ(*,*) D2
      D = D2/2
      TYPE *,' Distance between the relectors (minus the slab) ?'
      READ(*,*) L
      TYPE *,' Refractive index of the slab ?'
      READ(*,*) N2
      TYPE *,' Refractive index of the surrounding medium ?'
      READ(*,*) N1
      TYPE *,' Upper limit of f(X) (in meter) ?'
      READ(*,*) UPLIM
      TYPE *,' Lower limit of f(X) (in meter) ?'
      READ(*,*) LOLIM
      TYPE *,' What is the focal length of the lens (in meters)?'
      READ(*,*) F
      TYPE *,' Nnumber of even modes used (<1000) = ?'
      READ(*,*) ENUM
      TYPE *,' Nnumber of odd modes used (<1000) = ?'
      READ(*,*) ONUM
C
```

```
C  Set up constants for calculations
C
      PI = 3.141592654
      LAMBDA0 = 0.5E-6
      DUM0 = 0.0
      DELK = (2*PI/LAMBDA0)*(SQRT(N2**2-N1**2))
      V = DELK*D
      WRITE(2,*) D, N2, LAMBDA0
C
C  Finding X at which X*COTAN(X*L/2D) goes to infinity :
C
      START = 0.0
      DO I = 1, 190
         COTAN(I) = START + (2*D*PI/L)*(I-1)
      END DO
C
C  Finding X at which SQRT(X**2+V**2)*TAN(SQRT(X**2+V**2))
C  goes to infinity :
C
C  A)  Even radiation modes :
C
      IF (V .LE. PI/2) THEN
         START = PI/2
      ELSE
         START = INT((V-PI/2)/PI)*PI + 3*PI/2
      ENDIF
      DO I = 1, 190
         ETAN(I) = SQRT((START + PI*(I-1))**2 - V**2)
      END DO
C
C  B)  Odd radiation modes :
C
      IF (V .LE. PI) THEN
         START = PI/2
      ELSE
         START = (INT(V/PI)+1)*PI - PI/2
      ENDIF
      DO I = 1, 190
         OTAN(I) = SQRT(((START + PI*(I-1))+PI/2)**2 - V**2)
      END DO
C
C  Sorting out ranges where solutions for X exist (Even modes)
C
      ELEFT(1) = COTAN(1)
      M = 2
      N = 1
      COUNT = 1
      DO WHILE (COUNT .LE. 160)
         IF (ETAN(N) .LT. COTAN(M)) THEN
            XL1 = ELEFT(COUNT) + (ETAN(N)-ELEFT(COUNT))/(500*L/D)
            CALL EDIFF (XL1,L,V,D,YL1)
            IF (YL1 .GT. 0) THEN
               SIGNL = 'POS'
            ELSE
               SIGNL = 'NEG'
            ENDIF
            XR1 = ETAN(N) - (ETAN(N)-ELEFT(COUNT))/(500*L/D)
            CALL EDIFF (XR1,L,V,D,YR1)
            IF (YR1 .GT. 0.0) THEN
```

```
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ERIGHT(COUNT) = ETAN(N)
                  COUNT = COUNT + 1
                  ELEFT(COUNT) = ERIGHT(COUNT-1)
               ELSE
                  ELEFT(COUNT) = ETAN(N)
               ENDIF
               N = N + 1
            ELSE
               XL1 = ELEFT(COUNT) + (COTAN(M)-ELEFT(COUNT))/(500*L/D)
               CALL EDIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0.0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = COTAN(M) - (COTAN(M)-ELEFT(COUNT))/(500*L/D)
               CALL EDIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ERIGHT(COUNT) = COTAN(M)
                  COUNT = COUNT + 1
                  ELEFT(COUNT) = ERIGHT(COUNT-1)
               ELSE
                  ELEFT(COUNT) = COTAN(M)
               ENDIF
               M = M + 1
            ENDIF
         END DO
C
C  Sorting out ranges where solutions for X exist (Odd modes)
C
         OLEFT(1) = COTAN(1)
         J = 2
         KK = 1
         COUNT = 1
         DO WHILE (COUNT .LE. 160)
            IF (OTAN(KK) .LT. COTAN(J)) THEN
               XL1 = OLEFT(COUNT) + (OTAN(KK)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = OTAN(KK) - (OTAN(KK)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
```

```
                ENDIF
                IF (SIGNL .NE. SIGNR) THEN
                   ORIGHT(COUNT) = OTAN(KK)
                   COUNT = COUNT + 1
                   OLEFT(COUNT) = ORIGHT(COUNT-1)
                ELSE
                   OLEFT(COUNT) = OTAN(KK)
                ENDIF
                KK = KK + 1
             ELSE
                XL1 = OLEFT(COUNT) + (COTAN(J)-OLEFT(COUNT))/(500*L/D)
                CALL ODIFF (XL1,L,V,D,YL1)
                IF (YL1 .GT. 0.0) THEN
                   SIGNL = 'POS'
                ELSE
                   SIGNL = 'NEG'
                ENDIF
                XR1 = COTAN(J) - (COTAN(J)-OLEFT(COUNT))/(500*L/D)
                CALL ODIFF (XR1,L,V,D,YR1)
                IF (YR1 .GT. 0.0) THEN
                   SIGNR = 'POS'
                ELSE
                   SIGNR = 'NEG'
                ENDIF
                IF (SIGNL .NE. SIGNR) THEN
                   ORIGHT(COUNT) = COTAN(J)
                   COUNT = COUNT + 1
                   OLEFT(COUNT) = ORIGHT(COUNT-1)
                ELSE
                   OLEFT(COUNT) = COTAN(J)
                ENDIF
                J  = J + 1
             ENDIF
          END DO
C
C  Calculate the value of X's for the even radiation modes :
C
          DO I = 1, ENUM
             P1 = ELEFT(I) + 1E-10
             P5 = ERIGHT(I) - 1E-10
             REPEAT = 'TRUE '
             DO WHILE (REPEAT .EQ. 'TRUE ')
                P3 = (P5-P1)/2 + P1
                P2 = (P3-P1)/2 + P1
                P4 = (P5-P3)/2 + P3
                CALL EDIFF (P2,L,V,D,FP2)
                IF (FP2 .GT. 0) THEN
                   SIGN2 = 'POS'
                ELSE
                   SIGN2 = 'NEG'
                ENDIF
                CALL EDIFF (P4,L,V,D,FP4)
                IF (FP4 .GT. 0) THEN
                   SIGN4 = 'POS'
                ELSE
                   SIGN4 = 'NEG'
                ENDIF
                IF (SIGN2 .EQ. SIGN4) THEN
                   IF (FP2 .GT. FP4) THEN
```

```
                  IF (SIGN2 .EQ. 'POS') THEN
                     P1 = P4
                  ELSE
                     P5 = P2
                  ENDIF
               ELSE
                  IF (SIGN2 .EQ. 'POS') THEN
                     P5 = P2
                  ELSE
                     P1 = P4
                  ENDIF
               ENDIF
            ELSE
               P1 = P2
               P5 = P4
            ENDIF
            CALL EDIFF (P1,L,V,D,FP1)
            CALL EDIFF (P5,L,V,D,FP5)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  EVENX(I) = P1
               ELSE
                  EVENX(I) = P5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
      END DO
C
C  Calculate the value of X's for the odd radiation modes :
C
      DO I = 1, ONUM
         P1 = OLEFT(I) + 1E-10
         P5 = ORIGHT(I) - 1E-10
         REPEAT = 'TRUE '
         DO WHILE (REPEAT .EQ. 'TRUE ')
            P3 = (P5-P1)/2 + P1
            P2 = (P3-P1)/2 + P1
            P4 = (P5-P3)/2 + P3
            CALL ODIFF (P2,L,V,D,FP2)
            IF (FP2 .GT. 0) THEN
               SIGN2 = 'POS'
            ELSE
               SIGN2 = 'NEG'
            ENDIF
            CALL ODIFF (P4,L,V,D,FP4)
            IF (FP4 .GT. 0) THEN
               SIGN4 = 'POS'
            ELSE
               SIGN4 = 'NEG'
            ENDIF
            IF (SIGN2 .EQ. SIGN4) THEN
               IF (FP2 .GT. FP4) THEN
                  IF (SIGN2 .EQ. 'POS') THEN
                     P1 = P4
                  ELSE
                     P5 = P2
```

```
                  ENDIF
               ELSE
                  IF (SIGN2 .EQ. 'POS') THEN
                     P5 = P2
                  ELSE
                     P1 = P4
                  ENDIF
               ENDIF
            ELSE
               P1 = P2
               P5 = P4
            ENDIF
            CALL ODIFF (P1,L,V,D,FP1)
            CALL ODIFF (P5,L,V,D,FP5)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  ODDX(I) = P1
               ELSE
                  ODDX(I) = P5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
      END DO
C
C  Compute the weighting factor for even and odd radiation modes,
C  assume A2 = 1
C
      N = 50
      SEP = (UPLIM-LOLIM)/(2*N)
      K = 2*PI/LAMBDA0
      DO M = 1, 2
         IF (M .EQ. 1) THEN
            PHASE = 0.0
            NUM = ENUM
            WRITE(2,*) ENUM
         ELSE
            PHASE = PI/2
            NUM = ONUM
            WRITE(2,*) ONUM
         ENDIF
         DO I = 1, NUM
            IF (M .EQ. 1) THEN
               K1X = EVENX(I)/D
               K2X = SQRT(K1X**2+DELK**2)
               K1EVEN(I) = K1X
               K2EVEN(I) = K2X
               WRITE(2,*) K1EVEN(I), K2EVEN(I)
            ELSE
               K1X = ODDX(I)/D
               K2X = SQRT(K1X**2+DELK**2)
               K1ODD(I) = K1X
               K2ODD(I) = K2X
               WRITE(2,*) K1ODD(I), K2ODD(I)
            ENDIF
            PART1 = ((K2X/K1X)**2)*(DSIN(K2X*D+PHASE)**2)/2
            PART2 = L/2 - D + DSIN(2*K1X*(D-L/2))/(2*K1X)
```

```
      PART3 = (DCOS(K2X*D+PHASE)**2)/2
      PART4 = L/2 - D - DSIN(2*K1X*(D-L/2))/(2*K1X)
      PART5 = K2X/(4*K1X**2)*DSIN(2*(K2X*D+PHASE))
      PART6 = DCOS(2*K1X*(D-L/2)) - 1
      PART7 = (DSIN(2*(K2X*D-PHASE))+DSIN(2*(K2X*D+PHASE)))
     $        /(2*K2X)
      PART8 = (2*D + PART7)/2
      PART9 = ((K2X/K1X)**2)*(DSIN(K2X*D-PHASE)**2)/2
      PART10 = L/2 - D - DSIN(2*K1X*(L/2-D))/(2*K1X)
      PART11 = (DCOS(K2X*D-PHASE)**2)/2
      PART12 = L/2 - D + DSIN(2*K1X*(L/2-D))/(2*K1X)
      PART13 = K2X/(4*K1X**2)*DSIN(2*(K2X*D-PHASE))
      PART14 = 1 - DCOS(2*K1X*(L/2-D))
      DEN0  = CMPLX(PART1*PART2+PART3*PART4+PART5*PART6+PART8
     $         + PART9*PART10 + PART11*PART12 - PART13*PART14)
      S0 = DCOS(K*(LOLIM**2)/(2*F))*DCOS(K2X*LOLIM) +
     $     DCOS(K*(UPLIM**2)/(2*F))*DCOS(K2X*UPLIM)
      S1 = 0
      S2 = 0
      DO J = 1, 2*N-1, 2
         X = LOLIM + J*SEP
         S1 = DCOS(K*(X**2)/(2*F))*DCOS(K2X*X) + S1
      END DO
      DO J = 2, 2*N-2, 2
         X = LOLIM + J*SEP
         S2 = DCOS(K*(X**2)/(2*F))*DCOS(K2X*X) + S2
      END DO
      AREA1 = (SEP/3)*(S0 + 4*S1 + 2*S2)
      S0 = DSIN(K*(LOLIM**2)/(2*F))*DCOS(K2X*LOLIM) +
     $     DSIN(K*(UPLIM**2)/(2*F))*DCOS(K2X*UPLIM)
      S1 = 0
      S2 = 0
      DO J = 1, 2*N-1, 2
         X = LOLIM + J*SEP
         S1 = DSIN(K*(X**2)/(2*F))*DCOS(K2X*X) + S1
      END DO
      DO J = 2, 2*N-2, 2
         X = LOLIM + J*SEP
         S2 = DSIN(K*(X**2)/(2*F))*DCOS(K2X*X) + S2
      END DO
      AREA2 = (SEP/3)*(S0 + 4*S1 + 2*S2)
      S0 = DCOS(K*(LOLIM**2)/(2*F))*DSIN(K2X*LOLIM) +
     $     DCOS(K*(UPLIM**2)/(2*F))*DSIN(K2X*UPLIM)
      S1 = 0
      S2 = 0
      DO J = 1, 2*N-1, 2
         X = LOLIM + J*SEP
         S1 = DCOS(K*(X**2)/(2*F))*DSIN(K2X*X) + S1
      END DO
      DO J = 2, 2*N-2, 2
         X = LOLIM + J*SEP
         S2 = DCOS(K*(X**2)/(2*F))*DSIN(K2X*X) + S2
      END DO
      AREA3 = (SEP/3)*(S0 + 4*S1 + 2*S2)
      S0 = DSIN(K*(LOLIM**2)/(2*F))*DSIN(K2X*LOLIM) +
     $     DSIN(K*(UPLIM**2)/(2*F))*DSIN(K2X*UPLIM)
      S1 = 0
      S2 = 0
      DO J = 1, 2*N-1, 2
```

```
                  X = LOLIM + J*SEP
                  S1 = DSIN(K*(X**2)/(2*F))*DSIN(K2X*X) + S1
               END DO
               DO J = 2, 2*N-2, 2
                  X = LOLIM + J*SEP
                  S2 = DSIN(K*(X**2)/(2*F))*DSIN(K2X*X) + S2
               END DO
               AREA4 = (SEP/3)*(S0 + 4*S1 + 2*S2)
               NUMER = CMPLX(DCOS(PHASE)*AREA1,-1*DCOS(PHASE)*AREA2) +
     $                 CMPLX(DSIN(PHASE)*AREA3,-1*DSIN(PHASE)*AREA4)
               IF (M .EQ. 1) THEN
                  EC(I) = NUMER/DEN0
                  WRITE(3,*) EC(I)
               ELSE
                  OC(I) = NUMER/DEN0
                  WRITE(4,*) OC(I)
               ENDIF
            END DO
         END DO
C
         CLOSE (UNIT=2)
         CLOSE (UNIT=3)
         CLOSE (UNIT=4)
         STOP
         END
C
         SUBROUTINE EDIFF (DUM,L,V,D,DIFF)
C
C  Solve for the difference of DUM*(COTAN(DUM*L/(2*D)) and
C  SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2))   (Even modes)
C
         REAL*8  DUM, DUM1, DUM2, DUM3, DIFF, D, L, V
C
C  D : Half of the thickness of the slab
C  L : Distance between the reflectors (minus the slab)
C  V : Delta K * D
C
         DUM3 = DTAN(DUM*L/(2*D))
         IF (DUM3 .EQ. 0) THEN
            DUM3 = DUM3 + 1E-20
         ENDIF
         DUM1 = DUM*(1/DUM3)
         DUM2 = SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2))
         DIFF = DUM1 - DUM2
C
         RETURN
         END
C
         SUBROUTINE ODIFF (DUM,L,V,D,DIFF)
C
C  Solve for the difference of DUM*(COTAN(DUM*L/(2*D)) and
C  SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2-PI/2))  (Odd modes)
C
         REAL*8  DUM, DUM1, DUM2, DUM3, DIFF, D, L, V
C
C  D : Half of the thickness of the slab
C  L : Distance between the reflectors (minus the slab)
C  V : Delta K * D
C
```

```
      PI = 3.141592654
      DUM3 = DTAN(DUM*L/(2*D))
      IF (DUM3 .EQ. 0) THEN
         DUM3 = DUM3 + 1E-20
      ENDIF
      DUM1 = DUM*(1/DUM3)
      DUM2 = SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2)-PI/2)
      DIFF = DUM1 - DUM2
C
      RETURN
      END
```

```
C  PROGRAM : To determine all the guided modes exist for the single
C           slab waveguide being excited by focused wave.  Also, the
C           weighting factor for each guided mode is calculated.
C
      CHARACTER*3  SIGN2, SIGN4
      CHARACTER*4  MODE, MORE
      CHARACTER*5  REPEAT
      COMPLEX*8    EC(1000), OC(1000), CDUM, FIELD, ENUMER, ONUMER,
     $             EDEN0, ODEN0, EF(10000), OF(10000), U(0:10000),
     $             ES0, ES1, ES2, OS0, OS1, OS2
      REAL*8       PI, U2, HD, PD, K0, D2, D, N2, N1, LOW, HIGH, L,
     $             LPT, HPT, P1, P2, P3, P4, P5, FP1, FP2, FP4, FP5,
     $             DEN01, DEN02, DEN04, DEN05, LAMBDA0, UPLIM, LOLIM,
     $             EVENP(1000), ODDP(1000), EVENH(1000), ODDH(1000),
     $             UPL, LOL, ZPRIM, A, B, STEP, Z, F, KZ, A2, X, P,
     $             DUM0, LRANGE, RRANGE, SEP, K, H
      INTEGER      DUMMY, NUM, M , COUNT, COUNT1, COUNT2,
     $             ENUM, ONUM, MODNUM, I, J, N, POINTS
C
      OPEN (UNIT=2,FILE='PH.DAT',STATUS='NEW')
      OPEN (UNIT=3,FILE='GECOEFF.DAT',STATUS='NEW')
      OPEN (UNIT=4,FILE='GOCOEFF.DAT',STATUS='NEW')
C
C  K : Wavenumber
C  F : Focal length of the lens
C  N2 : Refractive index of the slab
C  N1 : Refractive index of the surrounding medium
C  D2 : Thickness of the slab
C  LAMBDA0 : Free-space wavelength of light
C  UPLIM, LOLIM : Range where f(X) exists
C  EVENP, EVENH : Contains propagation constants for even modes
C  ODDP, ODDH  : Contains propagation constants for odd modes
C  GECOEFF : File contains propagation constant for each even mode
C  GOCOEFF : File contains propagation constant for each odd mode
C
C  Obtain information from user
C
      TYPE *,' Thickness of the slab (in meters) ?'
      READ(*,*) D2
      TYPE *,' Distance between the reflectors (minus the slab) ?'
      READ(*,*) L
      TYPE *,' Refractive index of the slab ?'
      READ(*,*) N2
      TYPE *,' Refractive index of the surrounding medium ?'
      READ(*,*) N1
      TYPE *,' Upper limit of f(X) (in meters) ?'
      READ(*,*) UPLIM
      TYPE *,' Lower limit of f(X) (in meters) ?'
      READ(*,*) LOLIM
      TYPE *,' What is the focal length of the lens (in meter) ?'
      READ(*,*) F
      TYPE *,'What is the distance between the lens and the slab',
     $       ' waveguide ? (in meters)'
      READ(*,*) ZPRIM
      TYPE *,'What is the size of the lens ? (between X=UPL & X=LOL)'
      TYPE *,'UPL = ?'
      READ(*,*) UPL
      TYPE *,'LOL = ?'
```

```
      READ(*,*) LOL
C
C  Set up constants for calculations
C
      PI = 3.141592654
      DUMO = 0.0
      LAMBDAO = 0.5E-6
      D = D2/2
      KO = 2*PI/LAMBDAO
      U2 = (N2**2 - N1**2)*(KO**2)*(D**2)
      COUNT = INT((SQRT(U2))*2/PI)
      WRITE(2,*) D, N2, LAMBDAO
C
C  To find the existed guided modes
C
C     Find the number of existed guided modes (even/odd)
C
      ENUM = 0
      ONUM = 0
      DO M = 0, COUNT
         LOW = M/2.0
         HIGH = LOW + 0.5
         LPT = LOW*PI
         HPT = HIGH*PI
         DUMMY = INT(LOW)
         IF ((DUMMY-LOW) .EQ. 0) THEN
            MODE = 'EVEN'
            ENUM = ENUM + 1
         ELSE
            MODE = 'ODD '
            ONUM = ONUM + 1
         ENDIF
         NUM = DUMMY + 1
C
C        Find the propagation constants for each guided mode
C
         P1 = LPT + 1E-20
         P5 = HPT
         REPEAT = 'TRUE '
         DO WHILE (REPEAT .EQ. 'TRUE ')
            P3 = (P5-P1)/2 + P1
            P2 = (P3-P1)/2 + P1
            P4 = (P5-P3)/2 + P3
            IF (MODE .EQ. 'EVEN') THEN
               DEN02 = DTAN(P2)**2
               DEN04 = DTAN(P4)**2
            ELSE
               DEN02 = 1/(DTAN(P2)**2)
               DEN04 = 1/(DTAN(P4)**2)
            ENDIF
            FP2 = U2 - (P2**2)*(1+DEN02)
            FP4 = U2 - (P4**2)*(1+DEN04)
            IF (FP2 .GT. 0) THEN
               SIGN2 = 'POS'
            ELSE
               SIGN2 = 'NEG'
            ENDIF
            IF (FP4 .GT. 0) THEN
               SIGN4 = 'POS'
```

```
            ELSE
               SIGN4 = 'NEG'
            ENDIF
            IF (SIGN2 .EQ. SIGN4) THEN
              IF (FP2 .GT. FP4) THEN
                IF (SIGN2 .EQ. 'POS') THEN
                   P1 = P4
                ELSE
                   P5 = P2
                ENDIF
              ELSE
                IF (SIGN .EQ. 'POS') THEN
                   P5 = P2
                ELSE
                   P1 = P4
                ENDIF
              ENDIF
            ELSE
              P1 = P2
              P5 = P4
            ENDIF
            IF (MODE .EQ. 'EVEN') THEN
               DEN01 = DTAN(P1)**2
               DEN05 = DTAN(P5)**2
            ELSE
               DEN01 = 1/(DTAN(P1)**2)
               DEN05 = 1/(DTAN(P5)**2)
            ENDIF
            FP1 = U2 - (P1**2)*(1+DEN01)
            FP5 = U2 - (P5**2)*(1+DEN05)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  HD = P1
               ELSE
                  HD = P5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
         PD = SQRT(U2-HD**2)
         P = PD/D
         H = HD/D
         IF (MODE .EQ. 'EVEN') THEN
            EVENP(NUM) = P
            EVENH(NUM) = H
         ELSE
            ODDP(NUM) = P
            ODDH(NUM) = H
         ENDIF
      END DO
      COUNT1 = NUM
      COUNT2 = COUNT1
      IF (MODE .EQ. 'EVEN') THEN
         COUNT2 = NUM-1
      ENDIF
      DO M = 1, COUNT1
         WRITE(2,*) EVENP(M), EVENH(M)
```

```
      END DO
      DO M = 1, COUNT2
         WRITE(2,*) ODDP(M), ODDH(M)
      END DO
C
C  Compute thd electric field of the input wave at z=0
C
      K = 2*PI/LAMBDA0
      NUM = 25
      SEP = (UPLIM-LOLIM)/(2*NUM)
      POINTS = 400
      STEP = (RRANGE-LRANGE)/POINTS
      CALL UFIELD (LOLIM,F,K,LAMBDA0,ZPRIM,UPL,LOL,U(0))
      CALL UFIELD (UPLIM,F,K,LAMBDA0,ZPRIM,UPL,LOL,U(2*NUM))
      DO J = 1, 2*NUM-1
         X = LOLIM + J*SEP
         CALL UFIELD (X,F,K,LAMBDA0,ZPRIM,UPL,LOL,U(J))
      END DO
C
C  Compute the weighting factor for guided modes
C
      WRITE(3,*) ENUM
      WRITE(4,*) ONUM
      IF (ENUM .EQ. ONUM) THEN
          MODNUM = ENUM
          MORE = 'NO  '
      ELSE
          MODNUM = ONUM
          MORE = 'YES '
      ENDIF
      DO I = 1, MODNUM
         A = DCOS(EVENH(I)*D)
         A2 = A**2
         EDENO = CMPLX(A2/EVENP(I)+D+DSIN(2*EVENH(I)*D)/(2*EVENH(I)))
         A = DSIN(ODDH(I)*D)
         A2 = A**2
         ODENO = CMPLX(A2/ODDP(I)+D-DSIN(2*ODDH(I)*D)/(2*ODDH(I)))
         ESO = U(0)*CMPLX(DCOS(EVENH(I)*LOLIM)) +
     $         U(2*NUM)*CMPLX(DCOS(EVENH(I)*UPLIM))
         ES1 = CMPLX(0)
         ES2 = CMPLX(0)
         OSO = U(0)*CMPLX(DSIN(ODDH(I)*LOLIM)) +
     $         U(2*NUM)*CMPLX(DSIN(ODDH(I)*UPLIM))
         OS1 = CMPLX(0)
         OS2 = CMPLX(0)
         DO J = 1, 2*NUM-1, 2
            X = LOLIM + J*SEP
            ES1 = U(J)*CMPLX(DCOS(EVENH(I)*X)) + ES1
            OS1 = U(J)*CMPLX(DSIN(ODDH(I)*X)) + OS1
         END DO
         DO J = 2, 2*NUM-2, 2
            X = LOLIM + J*SEP
            ES2 = U(J)*CMPLX(DCOS(EVENH(I)*X)) + ES2
            OS2 = U(J)*CMPLX(DSIN(ODDH(I)*X)) + OS2
         END DO
         ENUMER = CMPLX(SEP/3)*(ESO + 4*ES1 + 2*ES2)
         EC(I) = ENUMER/EDENO
         WRITE(3,*) EC(I)
         ONUMER = CMPLX(SEP/3)*(OSO + 4*OS1 + 2*OS2)
```

```
            OC(I) = ONUMER/ODENO
            WRITE(4,*) OC(I)
         END DO
         IF (MORE .EQ. 'YES ') THEN
            DO I = MODNUM+1, ENUM
               A = DCOS(EVENH(I)*D)
               A2 = A**2
               EDENO = CMPLX(A2/EVENP(I)+D+DSIN(2*EVENH(I)*D)/
     $                  (2*EVENH(I)))
               ES0 = U(0)*CMPLX(DCOS(EVENH(I)*LOLIM)) +
     $                U(2*NUM)*CMPLX(DCOS(EVENH(I)*UPLIM))
               ES1 = CMPLX(0)
               ES2 = CMPLX(0)
               DO J = 1, 2*NUM-1, 2
                 X = LOLIM + J*SEP
                 ES1 = U(J)*CMPLX(DCOS(EVENH(I)*X)) + ES1
               END DO
               DO J = 2, 2*NUM-2, 2
                 X = LOLIM + J*SEP
                 ES2 = U(J)*CMPLX(DCOS(EVENH(I)*X)) + ES2
               END DO
               ENUMER = (SEP/3)*(ES0 + 4*ES1 + 2*ES2)
               EC(I) = ENUMER/EDENO
               WRITE(3,*) EC(I)
            END DO
         ENDIF
C
         CLOSE (UNIT=2)
         CLOSE (UNIT=3)
         CLOSE (UNIT=4)
         STOP
         END
C
         SUBROUTINE UFIELD (X0,F,K,LAMBDA0,Z,UP,L0,AREA)
C
C  To compute the electric field of the input wave at z=0 and X=X0
C
         COMPLEX*8 S0, S1, S2, C1, C2, F0, F2N, AREA
         REAL*8    X, X0, F, K, LAMBDA0, Z, UP, L0, H, DUM0
         INTEGER   N, I
C
C  N  : Half of the number of partitions used for integration
C  UP : Upper limit for the integration
C  L0 : Lower limit for the integration
C  H  : Distance of the partition
C
         PI = 3.141592654
         N = 500
         DUM0 = 0.0
         H = (UP-L0)/(2*N)
         C1 = CEXP(CMPLX(DUM0,K*L0**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUM0,-1*K*X0*L0/Z))
         F0 = C1*C2
         C1 = CEXP(CMPLX(DUM0,K*UP**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUM0,-1*K*X0*UP/Z))
         F2N = C1*C2
         S0 = F0 + F2N
         S1 = CMPLX(0)
         S2 = CMPLX(0)
```

```
      DO I = 1, 2*N-1, 2
         X = LO + I*H
         C1 = CEXP(CMPLX(DUMO,K*X**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUMO,-1*K*XO*X/Z))
         S1 = C1*C2 + S1
      END DO
      DO I = 2, 2*N-2, 2
         X = LO + I*H
         C1 = CEXP(CMPLX(DUMO,K*X**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUMO,-1*K*XO*X/Z))
         S2 = C1*C2 + S2
      END DO
      C1 = CEXP(CMPLX(DUMO,K*Z))*CEXP(CMPLX(DUMO,K*XO**2/(2*Z)))
      C2 = CMPLX(DUMO,LAMBDAO*Z)
      AREA = (C1/C2)*((H/3)*(SO + 4*S1 + 2*S2))
C
      RETURN
      END
```

```
C  PROGRAM : To determine the radiation modes for the single slab
C           waveguide being excited by focused wave.  Also, the
C           weighting factor for each radiation modes is calculated.
C
      CHARACTER*3 SIGN1, SIGN2, SIGN4, SIGNL, SIGNR, LSLOPE, RSLOPE
      CHARACTER*4 MODE, SIGN
      CHARACTER*5 REPEAT
      COMPLEX*8   EC(1000), OC(1000), RFIELD, CDUM, NUMER, DENO,
     $            SOL, S1L, S2L, SOR, S1R, S2R, U(0:10000)
      REAL*8      EVENX(1000), ODDX(1000), V, DELK, D2, D, L, N2,
     $            N1, COTAN(1000), ETAN(1000), OTAN(1000), LAMBDAO,
     $            START, ELEFT(1000), ERIGHT(1000), OLEFT(1000),
     $            ORIGHT(1000), P1, P2, P3, P4, P5, FP1, FP2, FP4,
     $            FP5, DUM1, DUM2, DUM, SLOPE, XL1, XL2, XR1, XR2,
     $            YL1, YL2, YR1, YR2, UPLIM, LOLIM, F, ZPRIM, UPL,
     $            LOL, K1X, K2X, PHASE, SEP, AREA1, AREA2, DUMMY,
     $            DUMO, K, PART1, PART2, PART3, PART4, PART5, PART6,
     $            PART7, PART8, PART9, PART10, PART11, PART12,
     $            PART13, PART14, K1EVEN(1000), K1ODD(1000),
     $            K2EVEN(1000), K2ODD(1000)
      INTEGER     I, M ,N, J, KK, ENUM, ONUM, COUNT, NUM
C
      OPEN (UNIT=2,FILE='K1K2.DAT',STATUS='NEW')
      OPEN (UNIT=3,FILE='RECOEFF.DAT',STATUS='NEW')
      OPEN (UNIT=4,FILE='ROCOEFF.DAT',STATUS='NEW')
C
C  K : Wavenumber
C  N : Half of the number of points used for integration
C  F : Focal length of the lens
C  D2 : Thickness of the slab
C  N2 : Refractive index of the slab
C  N1 : Refractive index of the surrounding medium
C  LAMBDAO : Free-space wavelength of light
C  UPLIM, LOLIM : Range where f(X) exists
C  SEP : Distance between partitions for integration
C
      TYPE *,' Thickness of the slab (in meters) ?'
      READ(*,*) D2
      TYPE *,' Distance between reflectors (minus the slab) ?'
      READ(*,*) L
      TYPE *,' Refractive index of the slab ?'
      READ(*,*) N2
      TYPE *,' Refractive index of the surrounding medium ?'
      READ(*,*) N1
      TYPE *,' Upper limit of f(X) (in meters) ?'
      READ(*,*) UPLIM
      TYPE *,' Lower limit of f(X) (in meters) ?'
      READ(*,*) LOLIM
      TYPE *,' What is the focal length of the lens (in meters)?'
      READ(*,*) F
      TYPE *,'What is the distance between the lens and the',
     $       ' slab waveguide ? (in meters)'
      READ(*,*) ZPRIM
      TYPE *,'What is the size of thelens ? (between X=UPL & X=LOL)'
      TYPE *,'UPL = ?'
      READ(*,*) UPL
      TYPE *,'LOL = ?'
      READ(*,*) LOL
```

```
      TYPE *,' Number of even modes to be used (<1000) = ?'
      READ(*,*) ENUM
      TYPE *,' Number of odd modes to be used (<1000) = ?'
      READ(*,*) ONUM
C
C  Set up constants for calculations
C
      PI = 3.141592654
      DUMO = 0.0
      D = D2/2
      LAMBDAO = 0.5E-6
      DELK = (2*PI/LAMBDAO)*(SQRT(N2**2-N1**2))
      V = DELK*D
      WRITE(2,*) D, N2, LAMBDAO
C
C  Finding X at which X*COTAN(X*L/2D) goes to infinity :
C
      START = 0.0
      DO I = 1, 190
         COTAN(I) = START + (2*D*PI/L)*(I-1)
      END DO
C
C  Finding X at which SQRT(X**2+V**2)*TAN(SQRT(X**2+V**2))
C  goes to infinity :
C
C  A)  Even radiation modes :
C
      IF (V .LE. PI/2) THEN
         START = PI/2
      ELSE
         START = INT((V-PI/2)/PI)*PI + 3*PI/2
      ENDIF
      DO I = 1, 190
         ETAN(I) = SQRT((START + PI*(I-1))**2 - V**2)
      END DO
C
C  B)  Odd radiation modes :
C
      IF (V .LE. PI) THEN
         START = PI/2
      ELSE
         START = (INT(V/PI)+1)*PI - PI/2
      ENDIF
      DO I = 1, 190
         OTAN(I) = SQRT(((START + PI*(I-1))+PI/2)**2 - V**2)
      END DO
C
C  Sorting out ranges where solutions for X exist (Even modes)
C
      ELEFT(1) = COTAN(1)
      M = 2
      N = 1
      COUNT = 1
      DO WHILE (COUNT .LE. 160)
         IF (ETAN(N) .LT. COTAN(M)) THEN
            XL1 = ELEFT(COUNT) + (ETAN(N)-ELEFT(COUNT))/(500*L/D)
            CALL EDIFF (XL1,L,V,D,YL1)
            IF (YL1 .GT. 0) THEN
               SIGNL = 'POS'
```

```
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = ETAN(N) - (ETAN(N)-ELEFT(COUNT))/(500*L/D)
               CALL EDIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ERIGHT(COUNT) = ETAN(N)
                  COUNT = COUNT + 1
                  ELEFT(COUNT) = ERIGHT(COUNT-1)
               ELSE
                  ELEFT(COUNT) = ETAN(N)
               ENDIF
               N = N + 1
            ELSE
               XL1 = ELEFT(COUNT) + (COTAN(M)-ELEFT(COUNT))/(500*L/D)
               CALL EDIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0.0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
               XR1 = COTAN(M) - (COTAN(M)-ELEFT(COUNT))/(500*L/D)
               CALL EDIFF (XR1,L,V,D,YR1)
               IF (YR1 .GT. 0.0) THEN
                  SIGNR = 'POS'
               ELSE
                  SIGNR = 'NEG'
               ENDIF
               IF (SIGNL .NE. SIGNR) THEN
                  ERIGHT(COUNT) = COTAN(M)
                  COUNT = COUNT + 1
                  ELEFT(COUNT) = ERIGHT(COUNT-1)
               ELSE
                  ELEFT(COUNT) = COTAN(M)
               ENDIF
               M = M + 1
            ENDIF
         END DO
C
C  Sorting out ranges where solutions for X exist (Odd modes)
C
         OLEFT(1) = COTAN(1)
         J = 2
         KK = 1
         COUNT = 1
         DO WHILE (COUNT .LE. 160)
            IF (OTAN(KK) .LT. COTAN(J)) THEN
               XL1 = OLEFT(COUNT) + (OTAN(KK)-OLEFT(COUNT))/(500*L/D)
               CALL ODIFF (XL1,L,V,D,YL1)
               IF (YL1 .GT. 0) THEN
                  SIGNL = 'POS'
               ELSE
                  SIGNL = 'NEG'
               ENDIF
```

```
            XR1 = OTAN(KK) - (OTAN(KK)-OLEFT(COUNT))/(500*L/D)
            CALL ODIFF (XR1,L,V,D,YR1)
            IF (YR1 .GT. 0.0) THEN
               SIGNR = 'POS'
            ELSE
               SIGNR = 'NEG'
            ENDIF
            IF (SIGNL .NE. SIGNR) THEN
               ORIGHT(COUNT) = OTAN(KK)
               COUNT = COUNT + 1
               OLEFT(COUNT) = ORIGHT(COUNT-1)
            ELSE
               OLEFT(COUNT) = OTAN(KK)
            ENDIF
            KK = KK + 1
         ELSE
            XL1 = OLEFT(COUNT) + (COTAN(J)-OLEFT(COUNT))/(500*L/D)
            CALL ODIFF (XL1,L,V,D,YL1)
            IF (YL1 .GT. 0.0) THEN
               SIGNL = 'POS'
            ELSE
               SIGNL = 'NEG'
            ENDIF
            XR1 = COTAN(J) - (COTAN(J)-OLEFT(COUNT))/(500*L/D)
            CALL ODIFF (XR1,L,V,D,YR1)
            IF (YR1 .GT. 0.0) THEN
               SIGNR = 'POS'
            ELSE
               SIGNR = 'NEG'
            ENDIF
            IF (SIGNL .NE. SIGNR) THEN
               ORIGHT(COUNT) = COTAN(J)
               COUNT = COUNT + 1
               OLEFT(COUNT) = ORIGHT(COUNT-1)
            ELSE
               OLEFT(COUNT) = COTAN(J)
            ENDIF
            J  = J + 1
         ENDIF
      END DO
C
C  Calculate the value of X's for the even radiation modes :
C
      DO I = 1, ENUM
         P1 = ELEFT(I) + 1E-10
         P5 = ERIGHT(I) - 1E-10
         REPEAT = 'TRUE '
         DO WHILE (REPEAT .EQ. 'TRUE ')
            P3 = (P5-P1)/2 + P1
            P2 = (P3-P1)/2 + P1
            P4 = (P5-P3)/2 + P3
            CALL EDIFF (P2,L,V,D,FP2)
            IF (FP2 .GT. 0) THEN
               SIGN2 = 'POS'
            ELSE
               SIGN2 = 'NEG'
            ENDIF
            CALL EDIFF (P4,L,V,D,FP4)
            IF (FP4 .GT. 0) THEN
```

```
                  SIGN4 = 'POS'
               ELSE
                  SIGN4 = 'NEG'
               ENDIF
               IF (SIGN2 .EQ. SIGN4) THEN
                  IF (FP2 .GT. FP4) THEN
                     IF (SIGN2 .EQ. 'POS') THEN
                        P1 = P4
                     ELSE
                        P5 = P2
                     ENDIF
                  ELSE
                     IF (SIGN2 .EQ. 'POS') THEN
                        P5 = P2
                     ELSE
                        P1 = P4
                     ENDIF
                  ENDIF
               ELSE
                  P1 = P2
                  P5 = P4
               ENDIF
               CALL EDIFF (P1,L,V,D,FP1)
               CALL EDIFF (P5,L,V,D,FP5)
               IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
                  IF (ABS(FP1) .LT. ABS(FP5)) THEN
                     EVENX(I) = P1
                  ELSE
                     EVENX(I) = P5
                  ENDIF
                  REPEAT = 'FALSE'
               ELSE
                  REPEAT = 'TRUE '
               ENDIF
            END DO
         END DO
C
C  Calculate the value of X's for the odd radiation modes :
C
         DO I = 1, ONUM
            P1 = OLEFT(I) + 1E-10
            P5 = ORIGHT(I) - 1E-10
            REPEAT = 'TRUE '
            DO WHILE (REPEAT .EQ. 'TRUE ')
               P3 = (P5-P1)/2 + P1
               P2 = (P3-P1)/2 + P1
               P4 = (P5-P3)/2 + P3
               CALL ODIFF (P2,L,V,D,FP2)
               IF (FP2 .GT. 0) THEN
                  SIGN2 = 'POS'
               ELSE
                  SIGN2 = 'NEG'
               ENDIF
               CALL ODIFF (P4,L,V,D,FP4)
               IF (FP4 .GT. 0) THEN
                  SIGN4 = 'POS'
               ELSE
                  SIGN4 = 'NEG'
               ENDIF
```

```
               IF (SIGN2 .EQ. SIGN4) THEN
                  IF (FP2 .GT. FP4) THEN
                     IF (SIGN2 .EQ. 'POS') THEN
                        P1 = P4
                     ELSE
                        P5 = P2
                     ENDIF
                  ELSE
                     IF (SIGN2 .EQ. 'POS') THEN
                        P5 = P2
                     ELSE
                        P1 = P4
                     ENDIF
                  ENDIF
               ELSE
                  P1 = P2
                  P5 = P4
               ENDIF
               CALL ODIFF (P1,L,V,D,FP1)
               CALL ODIFF (P5,L,V,D,FP5)
               IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
                  IF (ABS(FP1) .LT. ABS(FP5)) THEN
                     ODDX(I) = P1
                  ELSE
                     ODDX(I) = P5
                  ENDIF
                  REPEAT = 'FALSE'
               ELSE
                  REPEAT = 'TRUE '
               ENDIF
            END DO
         END DO
C
C  Compute the electric field of the input wave at z=0
C
         N = 50
         SEP = (UPLIM-LOLIM)/(2*N)
         K = 2*PI/LAMBDA0
         CALL UFIELD (LOLIM,F,K,LAMBDA0,ZPRIM,UPL,LOL,U(0))
         CALL UFIELD (UPLIM,F,K,LAMBDA0,ZPRIM,UPL,LOL,U(2*N))
         DO J = 1, 2*N-1
            X = LOLIM + J*SEP
            CALL UFIELD (X,F,K,LAMBDA0,ZPRIM,UPL,LOL,U(J))
         END DO
C
C  Compute the weighting factor for even and odd radiation modes,
C  assume A2 = 1
C
         DO M = 1, 2
            IF (M .EQ. 1) THEN
               PHASE = 0.0
               NUM = ENUM
               WRITE(2,*) ENUM
            ELSE
               PHASE = PI/2
               NUM = ONUM
               WRITE(2,*) ONUM
            ENDIF
            DO I = 1, NUM
```

```
            IF (M .EQ. 1) THEN
               K1X = EVENX(I)/D
               K2X = SQRT(K1X**2+DELK**2)
               K1EVEN(I) = K1X
               K2EVEN(I) = K2X
               WRITE(2,*) K1EVEN(I), K2EVEN(I)
            ELSE
               K1X = ODDX(I)/D
               K2X = SQRT(K1X**2+DELK**2)
               K1ODD(I) = K1X
               K2ODD(I) = K2X
               WRITE(2,*) K1ODD(I), K2ODD(I)
            ENDIF
            PART1 = ((K2X/K1X)**2)*(DSIN(K2X*D+PHASE)**2)/2
            PART2 = L/2 - D + DSIN(2*K1X*(D-L/2))/(2*K1X)
            PART3 = (DCOS(K2X*D+PHASE)**2)/2
            PART4 = L/2 - D - DSIN(2*K1X*(D-L/2))/(2*K1X)
            PART5 = K2X/(4*K1X**2)*DSIN(2*(K2X*D+PHASE))
            PART6 = DCOS(2*K1X*(D-L/2)) - 1
            PART7 = (DSIN(2*(K2X*D-PHASE))+DSIN(2*(K2X*D+PHASE)))
     $              /(2*K2X)
            PART8 = (2*D + PART7)/2
            PART9 = ((K2X/K1X)**2)*(DSIN(K2X*D-PHASE)**2)/2
            PART10 = L/2 - D - DSIN(2*K1X*(L/2-D))/(2*K1X)
            PART11 = (DCOS(K2X*D-PHASE)**2)/2
            PART12 = L/2 - D + DSIN(2*K1X*(L/2-D))/(2*K1X)
            PART13 = K2X/(4*K1X**2)*DSIN(2*(K2X*D-PHASE))
            PART14 = 1 - DCOS(2*K1X*(L/2-D))
            DEN0  = CMPLX(PART1*PART2+PART3*PART4+PART5*PART6+PART8
     $              + PART9*PART10 + PART11*PART12 - PART13*PART14)
            SOL = U(0)*CMPLX(DCOS(K2X*LOLIM)) +
     $            U(2*N)*CMPLX(DCOS(K2X*UPLIM))
            SOR = U(0)*CMPLX(DSIN(K2X*LOLIM)) +
     $            U(2*N)*CMPLX(DSIN(K2X*UPLIM))
            S1L = CMPLX(0)
            S2L = CMPLX(0)
            S1R = CMPLX(0)
            S2R = CMPLX(0)
            DO J = 1, 2*N-1, 2
               X = LOLIM + J*SEP
               S1L = U(J)*CMPLX(DCOS(K2X*X)) + S1L
               S1R = U(J)*CMPLX(DSIN(K2X*X)) + S1R
            END DO
            DO J = 2, 2*N-2, 2
               X = LOLIM + J*SEP
               S2L = U(J)*CMPLX(DCOS(K2X*X)) + S2L
               S2R = U(J)*CMPLX(DSIN(K2X*X)) + S2R
            END DO
            AREA1 = CMPLX(DCOS(PHASE))*((SEP/3)*(SOL+4*S1L+2*S2L))
            AREA2 = CMPLX(DSIN(PHASE))*((SEP/3)*(SOR+4*S1R+2*S2R))
            NUMER = AREA1 + AREA2
            IF (M .EQ. 1) THEN
               EC(I) = NUMER/DEN0
               WRITE(3,*) EC(I)
            ELSE
               OC(I) = NUMER/DEN0
               WRITE(4,*) OC(I)
            ENDIF
         END DO
```

```
      END DO
C
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)
      CLOSE (UNIT=4)
      STOP
      END
C
      SUBROUTINE EDIFF (DUM,L,V,D,DIFF)
C
C  Solve for the difference of DUM*(COTAN(DUM*L/(2*D)) and
C  SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2))  (Even modes)
C
      REAL*8  DUM, DUM1, DUM2, DUM3, DIFF, D, L, V
C
C  D : Half of the thickness of the slab
C  L : Distance between the reflectors (minus the slab)
C  V : Delta K * D
C
      DUM3 = DTAN(DUM*L/(2*D))
      IF (DUM3 .EQ. 0) THEN
         DUM3 = DUM3 + 1E-20
      ENDIF
      DUM1 = DUM*(1/DUM3)
      DUM2 = SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2))
      DIFF = DUM1 - DUM2
C
      RETURN
      END
C
      SUBROUTINE ODIFF (DUM,L,V,D,DIFF)
C
C  Solve for the difference of DUM*(COTAN(DUM*L/(2*D)) and
C  SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2-PI/2))  (Odd modes)
C
      REAL*8  DUM, DUM1, DUM2, DUM3, DIFF, D, L, V
C
C  D : Half of the thickness of the slab
C  L : Distance between the reflectors (minus the slab)
C  V : Delta K * D
C
      PI = 3.141592654
      DUM3 = DTAN(DUM*L/(2*D))
      IF (DUM3 .EQ. 0) THEN
         DUM3 = DUM3 + 1E-20
      ENDIF
      DUM1 = DUM*(1/DUM3)
      DUM2 = SQRT(DUM**2+V**2)*DTAN(SQRT(DUM**2+V**2)-PI/2)
      DIFF = DUM1 - DUM2
C
      RETURN
      END
C
      SUBROUTINE UFIELD (X0,F,K,LAMBDA0,Z,UP,L0,AREA)
C
C  To compute the electric field of the input wave at z=0 and X=X0
C
      COMPLEX*8 S0, S1, S2, C1, C2, F0, F2N, AREA
      REAL*8    X, X0, F, K, LAMBDA0, Z, UP, L0, H, DUM0
```

```
      INTEGER   N, I
C
C  N  : Half of the number of partitions used for integration
C  UP : Upper limit for the integration
C  LO : Lower limit for the integration
C  H  : Distance of the partition
C
      PI = 3.141592654
      N = 500
      DUMO = 0.0
      H = (UP-LO)/(2*N)
      C1 = CEXP(CMPLX(DUMO,K*LO**2*(1/Z-1/F)/2))
      C2 = CEXP(CMPLX(DUMO,-1*K*XO*LO/Z))
      FO = C1*C2
      C1 = CEXP(CMPLX(DUMO,K*UP**2*(1/Z-1/F)/2))
      C2 = CEXP(CMPLX(DUMO,-1*K*XO*UP/Z))
      F2N = C1*C2
      SO = FO + F2N
      S1 = CMPLX(0)
      S2 = CMPLX(0)
      DO I = 1, 2*N-1, 2
         X = LO + I*H
         C1 = CEXP(CMPLX(DUMO,K*X**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUMO,-1*K*XO*X/Z))
         S1 = C1*C2 + S1
      END DO
      DO I = 2, 2*N-2, 2
         X = LO + I*H
         C1 = CEXP(CMPLX(DUMO,K*X**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUMO,-1*K*XO*X/Z))
         S2 = C1*C2 + S2
      END DO
      C1 = CEXP(CMPLX(DUMO,K*Z))*CEXP(CMPLX(DUMO,K*XO**2/(2*Z)))
      C2 = CMPLX(DUMO,LAMBDAO*Z)
      AREA = (C1/C2)*((H/3)*(SO + 4*S1 + 2*S2))
C
      RETURN
      END
```

```
C  PROGRAM : To find the electric field for all the existed guided modes
C           in the single slab waveguide.
C
      CHARACTER*4 MODE, MORE
      COMPLEX*8   EC(1000), OC(1000), CDUM, FIELD, ENUMER, ONUMER,
     $            EDENO, ODENO, EF(10000), OF(10000), ES0, ES1, ES2,
     $            OS0, OS1, OS2
      REAL*8      EVENP(1000), EVENH(1000), ODDP(1000), ODDH(1000),
     $            B, STEP, Z, F, KZ, A2, X, N2, LAMBDA0, D, UPLIM,
     $            DUM0, LOLIM, LRANGE, RRANGE, ZPRIM, UPL, LOL, A
      INTEGER     ENUM, ONUM, MODNUM, I, J, N, NUM, POINTS
C
      OPEN (UNIT=2,FILE='PH.DAT',STATUS='OLD')
      OPEN (UNIT=3,FILE='GECOEFF.DAT',STATUS='OLD')
      OPEN (UNIT=4,FILE='GOCOEFF.DAT',STATUS='OLD')
      OPEN (UNIT=5,FILE='GFIELD.DAT',STATUS='NEW')
C
C  STEP : Step size
C  F : Focal length of the lens
C  D : Half of the thickness of the slab
C  RANGE : Range of X to be plotted
C  POINTS : Number of points to be plotted
C  A,B : Coefficients for the electric field at different modes
C        B is assumed to be 1
C  GFIELD : File contains the electric field values at z for
C           different values of x
C
C  Obtain information form the user
C
      TYPE *,' What is the value of z ?'
      READ(*,*)  Z
      TYPE *,' Range of X to be plotted (between X=LEFT & X=RIGHT) ?'
      TYPE *,' LEFT = ?'
      READ(*,*)  LRANGE
      TYPE *, ' RIGHT = ?'
      READ(*,*)  RRANGE
      WRITE(5,*) LRANGE, RRANGE
C
C  Set up constants for calculations
C
      PI = 3.141592654
      DUM0 = 0.0
      POINTS = 400
      STEP = (RRANGE-LRANGE)/POINTS
      READ(2,*) D, N2, LAMBDA0
      READ(3,*) ENUM
      READ(4,*) ONUM
      DO I = 1, ENUM
         READ(2,*) EVENP(I), EVENH(I)
      END DO
      DO I = 1, ONUM
         READ(2,*) ODDP(I),ODDH(I)
      END DO
      DO I = 1, ENUM
         READ(3,*) EC(I)
      END DO
      DO I = 1, ONUM
         READ(4,*) OC(I)
      END DO
```

```
C
C  Compute the electric for the even guided modes
C
      N = 1
      DO X = LRANGE, RRANGE, STEP
         FIELD = CMPLX(0.0)
         DO I = 1, ENUM
            A = DCOS(EVENH(I)*D)
            KZ = SQRT((2*PI*N2/LAMBDA0)**2 - EVENH(I)**2)
            IF (X .GT. D) THEN
               CDUM = CMPLX(A*DEXP(-1*EVENP(I)*(X-D)))
            ELSE IF (X .LT. -1*D) THEN
                  CDUM = CMPLX(A*DEXP(-1*EVENP(I)*(-1*X-D)))
               ELSE
                  CDUM = CMPLX(B*DCOS(EVENH(I)*X))
            ENDIF
            FIELD = FIELD + EC(I)*(CDUM*CEXP(CMPLX(DUM0,-1*KZ*Z)))
         END DO
         EF(N) = FIELD
         N = N + 1
      END DO
C
C  Compute the electric for the odd guided modes
C
      N = 1
      DO X = LRANGE, RRANGE, STEP
         FIELD = CMPLX(0.0)
         DO I = 1, ONUM
            A = DSIN(ODDH(I)*D)
            KZ = SQRT((2*PI*N2/LAMBDA0)**2 - ODDH(I)**2)
            IF (X .GT. D) THEN
               CDUM = CMPLX(A*DEXP(-1*ODDP(I)*(X-D)))
            ELSE IF (X .LT. -1*D) THEN
                  CDUM = CMPLX(-1*A*DEXP(-1*ODDP(I)*(-1*X-D)))
               ELSE
                  CDUM = CMPLX(B*DSIN(ODDH(I)*X))
            ENDIF
            FIELD = FIELD + OC(I)*(CDUM*CEXP(CMPLX(DUM0,-1*KZ*Z)))
         END DO
         OF(N) = FIELD
         N = N + 1
      END DO
C
C  Compute the electric field for the guided modes
C
      WRITE(5,*) POINTS
      N = 1
      DO X = LRANGE, RRANGE, STEP
         WRITE(5,*) X, EF(N)+OF(N)
         N = N + 1
      END DO
C
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)
      CLOSE (UNIT=4)
      CLOSE (UNIT=5)
      STOP
      END
```

```
C  PROGRAM : To find the electric field of the specified number of
C           radiaton modes in the single slab waveguide.
C
      CHARACTER*4 MODE, SIGN
      COMPLEX*8   EC(1000), OC(1000), RFIELD, CDUM
      REAL*8      K1X, K2X, DELK, EVENX(1000), ODDX(1000), PHASE,
     $            ZPRIM, UPL, LOL, F, D2, D, Z, X, LAMBDA0, LRANGE,
     $            RRANGE, STEP, PART1, PART2, DUMMY, N2, N1, KZ,
     $            DUM0, L, V, UPLIM, LOLIM, K, K1EVEN(1000),
     $            K1ODD(1000), K2EVEN(1000), K2ODD(1000)
      INTEGER     POINTS, M, N, I, J, NUM, ENUM, ONUM
C
      OPEN (UNIT=5,FILE='K1K2.DAT',STATUS='OLD')
      OPEN (UNIT=6,FILE='RECOEFF.DAT',STATUS='OLD')
      OPEN (UNIT=7,FILE='ROCOEFF.DAT',STATUS='OLD')
      OPEN (UNIT=8,FILE='REFIELD.DAT',STATUS='NEW')
      OPEN (UNIT=11,FILE='ROFIELD.DAT',STATUS='NEW')
C
C  K : Wavenumber
C  F : Focal length of the lens
C  N : Half of the number of points used for integration
C  ENUM : Number of even radiation modes used
C  ONUM : Number of odd radiation modes used
C  REFIELD : File contains the electric field for the even radiation
C            modes at z for different values of x
C  ROFIELD : File contains the electric field for the odd radiation
C            modes at z for different values of x
C
C  Obtain information from user
C
      TYPE *,' What is the value of z ?'
      READ(*,*) Z
      TYPE *,' Range of X to be plotted (between X=LEFT & X=RIGHT) ?'
      TYPE *,' LEFT = ?'
      READ(*,*) LRANGE
      TYPE *,' RIGHT = ?'
      READ(*,*) RRANGE
C
C  Set up constants for calculations
C
      PI = 3.141592654
      DUM0 = 0.0
      READ(5,*) D, N2, LAMBDA0
      READ(5,*) ENUM
      DO I = 1, ENUM
         READ(5,*) K1EVEN(I), K2EVEN(I)
      END DO
      READ(5,*) ONUM
      DO I = 1, ONUM
         READ(5,*) K1ODD(I), K2ODD(I)
      END DO
      DO I = 1, ENUM
         READ(6,*) EC(I)
      END DO
      DO I = 1, ONUM
         READ(7,*) OC(I)
      END DO
C
C  Compute the electric field for the even and odd radiation modes
```

```
C
      POINTS = 400
      STEP = (RRANGE-LRANGE)/POINTS
      DO M = 1, 2
         IF (M .EQ. 1) THEN
            PHASE = 0.
            NUM = ENUM
         ELSE
            PHASE = PI/2
            NUM = ONUM
         ENDIF
         DO 20 X = LRANGE, RRANGE, STEP
            RFIELD = 0.0
            DO 15 I = 1, NUM
               IF (M .EQ. 1) THEN
                  K1X = K1EVEN(I)
                  K2X = K2EVEN(I)
                  DUMMY = (2*PI*N2/LAMBDA0)**2 - K2X**2
                  IF (DUMMY .LT. 0) THEN
                     KZ = SQRT(-1*DUMMY)
                     SIGN = 'IMAG'
                  ELSE
                     KZ = SQRT(DUMMY)
                     SIGN = 'REAL'
                  ENDIF
               ELSE
                  K1X = K1ODD(I)
                  K2X = K2ODD(I)
                  DUMMY = (2*PI*N2/LAMBDA0)**2 - K2X**2
                  IF (DUMMY .LT. 0) THEN
                     KZ = SQRT(-1*DUMMY)
                     SIGN = 'IMAG'
                  ELSE
                     KZ = SQRT(DUMMY)
                     SIGN = 'REAL'
                  ENDIF
               ENDIF
               IF (X .LT. -1*D) THEN
                  PART1 = (K2X/K1X)*DSIN(K2X*D+PHASE)*DSIN(K1X*(X+D))
                  PART2 = DCOS(K2X*D+PHASE)*DCOS(K1X*(X+D))
                  CDUM = CMPLX(PART1+PART2)
               ELSE IF (X .GT. D) THEN
                     PART1 = (-1*K2X/K1X)*DSIN(K2X*D-PHASE)*
     $                       DSIN(K1X*(X-D))
                     PART2 = DCOS(K2X*D-PHASE)*DCOS(K1X*(X-D))
                     CDUM = CMPLX(PART1+PART2)
                  ELSE
                     CDUM = CMPLX(DCOS(K2X*X-PHASE))
               ENDIF
               IF (M .EQ. 1) THEN
                  IF (SIGN .EQ. 'IMAG') THEN
                    RFIELD = RFIELD + EC(I)*CDUM*CEXP(CMPLX(-1*KZ*Z))
                  ELSE
                    RFIELD = RFIELD + EC(I)*CDUM*
     $                       CEXP(CMPLX(DUM0,-1*KZ*Z))
                  ENDIF
               ELSE
                  IF (SIGN .EQ. 'IMAG') THEN
                    RFIELD = RFIELD + 0C(I)*CDUM*CEXP(CMPLX(-1*KZ*Z))
```

```
                  ELSE
                    RFIELD = RFIELD + OC(I)*CDUM*
     $                       CEXP(CMPLX(DUMO,-1*KZ*Z))
                  ENDIF
               ENDIF
            END DO
            IF (M .EQ. 1) THEN
               WRITE(8,*) X, RFIELD
            ELSE
               WRITE(11,*) X, RFIELD
            ENDIF
         END DO
      END DO
C
      CLOSE (UNIT=5)
      CLOSE (UNIT=6)
      CLOSE (UNIT=7)
      CLOSE (UNIT=8)
      CLOSE (UNIT=11)
      STOP
      END
```

```
C  PROGRAM : To plot the total electric field of the guided and
C           radiation modes for the single slab waveguide.
C
      COMPLEX*8 TOTAL, GUIDE, RADEVEN, RADODD
      REAL*8  X, L, LRANGE, RRANGE, STEP, DUM, DUM1, DUM2, PHASE
      INTEGER POINTS
C
      OPEN (UNIT=10,FILE='GFIELD.DAT',STATUS='OLD')
      OPEN (UNIT=11,FILE='REFIELD.DAT',STATUS='OLD')
      OPEN (UNIT=12,FILE='ROFIELD.DAT',STATUS='OLD')
      OPEN (UNIT=9,FILE='TOTAL.DAT',STATUS='NEW')
      OPEN (UNIT=13,FILE='PHASE.DAT',STATUS='NEW')
C
C  TOTAL : File contains the magnitude of the electric field at z
C          for different values of x
C  PHASE : File contains the phase of the electric field at z
C          for different values of x
C
      READ(10,*) LRANGE, RRANGE
      READ(10,*) POINTS
      WRITE(9,*) POINTS
      WRITE(13,*) POINTS
      STEP = (RRANGE-LRANGE)/POINTS
      DO X = LRANGE, LRANGE+POINTS*STEP, STEP
         READ(10,*) DUM, GUIDE
         READ(11,*) DUM, RADEVEN
         READ(12,*) DUM, RADODD
         TOTAL = GUIDE + RADEVEN + RADODD
         DUM1 = AIMAG(TOTAL)
         DUM2 = REAL(TOTAL)
         PHASE = DATAN2D(DUM1,DUM2)
         WRITE(9,*) X, CABS(TOTAL)
         WRITE(13,*) X, PHASE
      END DO
C
      CLOSE (UNIT=10)
      CLOSE (UNIT=11)
      CLOSE (UNIT=12)
      CLOSE (UNIT=9)
      STOP
      END
```

```
C  PROGRAM : To find all the existed guided modes for the case
C           of the parallel slab dielectric waveguides being
C           excited by an input wave at z=0.
C
      CHARACTER*3 SIGNB, SIGNA, SIGN2, SIGN4, FLAG1, FLAG2
      CHARACTER*5 REPEAT
      REAL*8  A, B, C, D, E, F, N1, N2, N3, N4, N5, K0, PI,
     $        LAMBDA0, K1, K2, K3, K4, K5, CONST, HA, HB,
     $        LRANGE(100), RRANGE(100), P1, P2, L1, L2, TH,
     $        P3, P4, P5, FP1, FP2, FP4, FP5, LCON, RCON,
     $        H1(100), H2(100), H3(100), H4(100), H5(100),
     $        A1(100), A2(100), A3(100), A4(100), A5(100),
     $        SI2(100), SI3(100), SI4(100), X, STEP, DUM,
     $        DUM1, DUM2, DUM3, DUM4, OLD, NOW, AFT, CON
      INTEGER I, J, GNUM, NUM, CODE
C
      OPEN (UNIT=1,FILE='COPG.DAT',STATUS='NEW')
C
C  A-F : Define the locations of the dielectric waveguides
C        and the reflectors
C  GNUM : Number of existed guided modes
C  N1-N5 : Refractive indices for each medium
C  H1-H5 : Propagation constants for each guide mode at
C          different medium
C  A1-A5 : Arbitrary constants for the electric field
C  SI2-SI4 : Phase shift in the electric field
C  LAMBDA0 : Free-space wavelength
C  COPG : File contains propagation constants for each guided
C         mode and arbitrary constants for the electric field
C
C  Obtain information from the user
C
      WRITE(*,*) 'N1 = ?'
      READ(*,*) N1
      WRITE(*,*) 'N2 = ?'
      READ(*,*) N2
      WRITE(*,*) 'N3 = ?'
      READ(*,*) N3
      WRITE(*,*) 'N4 = ?'
      READ(*,*) N4
      WRITE(*,*) 'N5 = ?'
      READ(*,*) N5
      WRITE(*,*) 'A = ?'
      READ(*,*) A
      WRITE(*,*) 'B = ?'
      READ(*,*) B
      WRITE(*,*) 'C = ?'
      READ(*,*) C
      WRITE(*,*) 'D = ?'
      READ(*,*) D
      WRITE(*,*) 'E = ?'
      READ(*,*) E
      WRITE(*,*) 'F = ?'
      READ(*,*) F
C
C  Set up constants for calculations
C
      PI = 3.141592654
      LAMBDA0 = 0.5E-6
```

```
      KO = 2*PI/LAMBDA0
      K1 = (KO**2)*(N1**2)
      K2 = (KO**2)*(N2**2)
      K3 = (KO**2)*(N3**2)
      K4 = (KO**2)*(N4**2)
      K5 = (KO**2)*(N5**2)
      WRITE(1,*) N1, N2, N3, N4, N5, KO
      WRITE(1,*) A, B, C, D, E, F
C
C  Compute all the existed guided modes with sinusoidal behaviour
C  in medium 2,3,4 and exponential behaviour in medium 1,5.
C
      DUM1 = K2 - K1
      DUM2 = K3 - K1
      DUM3 = K4 - K1
      IF (DUM1 .LT. 0 .OR. DUM2 .LT. 0 .OR. DUM3 .LT. 0) THEN
         CONST = -1.
      ELSE
       IF (DUM1 .LT. DUM2) THEN
         IF (DUM1 .LT. DUM3) THEN
            RCON = SQRT(DUM1)
         ELSE
            RCON = SQRT(DUM3)
         ENDIF
       ELSE
         IF (DUM2 .LT. DUM3) THEN
            RCON = SQRT(DUM2)
         ELSE
            RCON = SQRT(DUM3)
         ENDIF
       ENDIF
       DUM1 = K1 - K5
       IF (DUM1 .LT. 0) THEN
         LCON = SQRT(ABS(K1-K5))
       ELSE
         LCON = 0.
       ENDIF
       CONST = RCON - LCON
      ENDIF
      STEP = CONST/2000
C
C     Compute the number of existed guided modes
C
      IF (CONST .GT. 0) THEN
       J = 1
       DO I = 0, 2000
         IF (I .EQ. 2000) THEN
            TH = RCON - 1E-9
         ELSE
            TH = I*STEP + LCON
         ENDIF
         CALL EQNHS (TH,K1,K2,K3,K4,K5,A,B,D,E,DIFF)
         IF (I .EQ. 0) THEN
            HB = TH
            IF (DIFF .GE. 0) THEN
               SIGNB = 'POS'
            ELSE
               SIGNB = 'NEG'
            ENDIF
```

```
            ENDIF
            HA = TH
            IF (DIFF .GE. 0) THEN
               SIGNA = 'POS'
            ELSE
               SIGNA = 'NEG'
            ENDIF
            IF (SIGNA .NE. SIGNB .AND. SIGNB .EQ. 'POS') THEN
               LRANGE(J) = HB
               RRANGE(J) = HA
               J = J + 1
            ENDIF
            HB = HA
            SIGNB = SIGNA
          END DO
          GNUM = J - 1
          WRITE(1,*) GNUM
C
C         Compute the H's, A's and SI's for each guided mode
C
          DO I = 1, GNUM
            A1(I) = 1.0
            P1 = LRANGE(I)
            P5 = RRANGE(I)
            REPEAT = 'TRUE '
            DO WHILE (REPEAT .EQ. 'TRUE ')
               P3 = (P5-P1)/2 + P1
               P2 = (P3-P1)/2 + P1
               P4 = (P5-P3)/2 + P3
               CALL EQNHS (P2,K1,K2,K3,K4,K5,A,B,D,E,FP2)
               IF (FP2 .GT. 0) THEN
                  SIGN2 = 'POS'
               ELSE
                  SIGN2 = 'NEG'
               ENDIF
               CALL EQNHS (P4,K1,K2,K3,K4,K5,A,B,D,E,FP4)
               IF (FP4 .GT. 0) THEN
                  SIGN4 = 'POS'
               ELSE
                  SIGN4 = 'NEG'
               ENDIF
               IF (SIGN2 .EQ. SIGN4) THEN
                  IF (FP2 .GT. FP4) THEN
                     IF (SIGN2 .EQ. 'POS') THEN
                        P1 = P4
                     ELSE
                        P5 = P2
                     ENDIF
                  ELSE
                     IF (SIGN2 .EQ. 'POS') THEN
                        P5 = P2
                     ELSE
                        P1 = P4
                     ENDIF
                  ENDIF
               ELSE
                  P1 = P2
                  P5 = P4
               ENDIF
```

```
               CALL EQNHS (P1,K1,K2,K3,K4,K5,A,B,D,E,FP1)
               CALL EQNHS (P5,K1,K2,K3,K4,K5,A,B,D,E,FP5)
               IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
                  IF (ABS(FP1) .LT. ABS(FP5)) THEN
                     H1(I) = P1
                     DUM = FP1
                  ELSE
                     H1(I) = P5
                     DUM = FP5
                  ENDIF
                  REPEAT = 'FALSE'
               ELSE
                  REPEAT = 'TRUE '
               ENDIF
            END DO
            DUM = H1(I)**2
            H2(I) = SQRT(K2 - K1 - DUM)
            H3(I) = SQRT(K3 - K1 - DUM)
            H4(I) = SQRT(K4 - K1 - DUM)
            H5(I) = SQRT(K1 - K5 + DUM)
            WRITE(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
            SI2(I) = H2(I)*B - DATAN(H1(I)/H2(I))
            SI4(I) = DATAN(H5(I)/H4(I)) - H4(I)*E
            SI3(I) = H3(I)*A-DATAN((H2(I)/H3(I))*DTAN(H2(I)
     $                  *(A-B)+DATAN(H1(I)/H2(I))))
            A2(I) = A1(I)/DCOS(H2(I)*B-SI2(I))
            A3(I) = A2(I)*DCOS(H2(I)*A-SI2(I))/DCOS(H3(I)*A-SI3(I))
            A4(I) = A3(I)*DCOS(H3(I)*D+SI3(I))/DCOS(H4(I)*D+SI4(I))
            A5(I) = A4(I)*DCOS(H4(I)*E+SI4(I))
            WRITE(1,*) A1(I), A2(I), A3(I), A4(I), A5(I)
            WRITE(1,*) SI2(I), SI3(I), SI4(I)
          END DO
         ELSE
          GNUM = 0
          WRITE(1,*) GNUM
         ENDIF
C
C  Compute all the existed guided modes with exponential behaviour
C  in medium 1,2,3 and sinusoidal behaviour in medium 2,4.
C
         DUM1 = K1 - K3
         DUM2 = K1 - K5
         IF (DUM1 .LT. 0)THEN
            L1 = SQRT(ABS(DUM1))
         ELSE
            L1 = 0.
         ENDIF
         IF (DUM2 .LT. 0)THEN
            L2 = SQRT(ABS(DUM2))
         ELSE
            L2 = 0.
         ENDIF
         IF (L1 .GT. L2) THEN
           LCON = L1 + 1E-9
         ELSE
           LCON = L2 + 1E-9
         ENDIF
         DUM1 = K2 - K1
         DUM2 = K4 - K1
```

```
      IF (DUM1 .LT. 0 .OR. DUM2 .LT. 0) THEN
        RCON = -1.
      ELSE
        IF (DUM1 .LT. DUM2) THEN
          RCON = SQRT(DUM1) - 1E-9
        ELSE
          RCON = SQRT(DUM2) - 1E-9
        ENDIF
      ENDIF
      CONST = RCON - LCON
      STEP = CONST/4000
C
C     Compute the number of existed guided modes
C
      IF (CONST .GT. 0) THEN
       DO CODE = 1, 2
        J = 1
        DO I = 0, 4000
         IF (I .EQ. 4000) THEN
            TH = RCON
         ELSE
            TH = I*STEP + LCON
         ENDIF
         CALL EQNH (CODE,TH,K1,K2,K3,K4,K5,A,B,D,E,DIFF)
         IF (I .EQ. 0) THEN
            HB = TH
            IF (DIFF .GE. 0) THEN
               SIGNB = 'POS'
            ELSE
               SIGNB = 'NEG'
            ENDIF
            OLD = DIFF
         ENDIF
         HA = TH
         IF (DIFF .GE. 0) THEN
            SIGNA = 'POS'
         ELSE
            SIGNA = 'NEG'
         ENDIF
         NOW = DIFF
         IF (I .LT. 4000-1) THEN
\           CALL EQNH (CODE,TH+STEP,K1,K2,K3,K4,K5,A,B,D,E,AFT)
         ELSE
            AFT = NOW
         ENDIF
         IF (OLD-NOW .GT. 0) THEN
            FLAG1 = 'POS'
         ELSE
            FLAG1 = 'NEG'
         ENDIF
         IF (NOW-AFT .GT. 0) THEN
            FLAG2 = 'POS'
         ELSE
            FLAG2 = 'NEG'
         ENDIF
         OLD = NOW
         IF (SIGNA .NE. SIGNB .AND. FLAG1 .EQ. FLAG2) THEN
            LRANGE(J) = HB
            RRANGE(J) = HA
```

```
            J = J + 1
          ENDIF
          HB = HA
          SIGNB = SIGNA
        END DO
        NUM = J - 1
        WRITE(1,*) NUM
C
C       Compute the H's, A's and SI's for each guided mode
C
        DO I = 1, NUM
          A1(I) = 1.0
          P1 = LRANGE(I)
          P5 = RRANGE(I)
          REPEAT = 'TRUE '
          DO WHILE (REPEAT .EQ. 'TRUE ')
            P3 = (P5-P1)/2 + P1
            P2 = (P3-P1)/2 + P1
            P4 = (P5-P3)/2 + P3
            CALL EQNH (CODE,P2,K1,K2,K3,K4,K5,A,B,D,E,FP2)
            IF (FP2 .GT. 0) THEN
              SIGN2 = 'POS'
            ELSE
              SIGN2 = 'NEG'
            ENDIF
            CALL EQNH (CODE,P4,K1,K2,K3,K4,K5,A,B,D,E,FP4)
            IF (FP4 .GT. 0) THEN
              SIGN4 = 'POS'
            ELSE
              SIGN4 = 'NEG'
            ENDIF
            IF (SIGN2 .EQ. SIGN4) THEN
              IF (FP2 .GT. FP4) THEN
                IF (SIGN2 .EQ. 'POS') THEN
                  P1 = P4
                ELSE
                  P5 = P2
                ENDIF
              ELSE
                IF (SIGN2 .EQ. 'POS') THEN
                  P5 = P2
                ELSE
                  P1 = P4
                ENDIF
              ENDIF
            ELSE
              P1 = P2
              P5 = P4
            ENDIF
            CALL EQNH (CODE,P1,K1,K2,K3,K4,K5,A,B,D,E,FP1)
            CALL EQNH (CODE,P5,K1,K2,K3,K4,K5,A,B,D,E,FP5)
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5).LT. 1E-5) THEN
              IF (ABS(FP1) .LT. ABS(FP5)) THEN
                H1(I) = P1
                DUM = FP1
              ELSE
                H1(I) = P5
                DUM = FP5
              ENDIF
```

```
                REPEAT = 'FALSE'
              ELSE
                REPEAT = 'TRUE '
              ENDIF
            END DO
            DUM = H1(I)**2
            H2(I) = SQRT(K2 - K1 - DUM)
            H3(I) = SQRT(K1 - K3 + DUM)
            H4(I) = SQRT(K4 - K1 - DUM)
            H5(I) = SQRT(K1 - K5 + DUM)
            WRITE(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
            SI2(I) = H2(I)*B - DATAN(H1(I)/H2(I))
            SI4(I) = DATAN(H5(I)/H4(I)) - H4(I)*E
            A2(I) = A1(I)/DCOS(H2(I)*B-SI2(I))
            IF (CODE .EQ. 1) THEN
              A3(I) = A2(I)*DCOS(H2(I)*A-SI2(I))/DSINH(H3(I)*A)
              A4(I) = -1*A3(I)*DSINH(H3(I)*D)/DCOS(H4(I)*D+SI4(I))
              A5(I) = A4(I)*DCOS(H4(I)*E+SI4(I))
            ELSE
              A3(I) = A2(I)*DCOS(H2(I)*A-SI2(I))/DCOSH(H3(I)*A)
              A4(I) = A3(I)*DCOSH(H3(I)*D)/DCOS(H4(I)*D+SI4(I))
              A5(I) = A4(I)*DCOS(H4(I)*E+SI4(I))
            ENDIF
            WRITE(1,*) A1(I), A2(I), A3(I), A4(I), A5(I)
            WRITE(1,*) SI2(I), SI4(I)
           END DO
          END DO
         ELSE
          NUM = 0
          WRITE(1,*) NUM
         ENDIF
C
C  Compute all the existed guided modes with exponential behaviour
C  in medium 1,2,4,5 (or 1,3,4,5) [or 1,2,3,5] and sinusoidal
C  behaviour in medium 3 (or 2) [or 4]
C
         DO CODE = 1, 3
          IF (CODE .EQ. 1) THEN
           DUM1 = K1 - K5
           DUM2 = K1 - K2
           DUM3 = K1 - K4
           DUM4 = ABS(K3 - K1)
          ELSE IF (CODE .EQ. 2) THEN
                 DUM1 = K1 - K5
                 DUM2 = K1 - K3
                 DUM3 = K1 - K4
                 DUM4 = ABS(K2 - K1)
              ELSE
                 DUM1 = K1 - K5
                 DUM2 = K1 - K3
                 DUM3 = K1 - K2
                 DUM4 = ABS(K4 - K1)
          ENDIF
          IF (DUM1 .GT. 0 .AND. DUM2 .GT. 0 .AND. DUM3 .GT. 0) THEN
            LCON = 0.
          ELSE
            DUM1 = ABS(DUM1)
            DUM2 = ABS(DUM2)
            DUM3 = ABS(DUM3)
```

```
            IF (DUM1 .GT. DUM2) THEN
             IF (DUM1 .GT. DUM3) THEN
               LCON = SQRT(DUM1) + 1E-9
             ELSE
               LCON = SQRT(DUM3) + 1E-9
             ENDIF
            ELSE
             IF (DUM2 .GT. DUM3) THEN
               LCON = SQRT(DUM2) + 1E-9
             ELSE
               LCON = SQRT(DUM3) + 1E-9
             ENDIF
            ENDIF
           ENDIF
           RCON = SQRT(DUM4) - 1E-9
           CONST = RCON - LCON
           STEP = CONST/10000
C
C       Compute the number of existed guided modes
C
           IF (CONST .GT. 0) THEN
            J = 1
            DO I = 0, 10000
             IF (I .EQ. 10000) THEN
                TH = RCON
             ELSE
                TH = I*STEP + LCON
             ENDIF
             CALL EQN (CODE,TH,K1,K2,K3,K4,K5,A,B,D,E,DIFF)
             IF (I .EQ. 0) THEN
                HB = TH
                IF (DIFF .GE. 0) THEN
                   SIGNB = 'POS'
                ELSE
                   SIGNB = 'NEG'
                ENDIF
                OLD = DIFF
             ENDIF
             HA = TH
             IF (DIFF .GE. 0) THEN
                SIGNA = 'POS'
             ELSE
                SIGNA = 'NEG'
             ENDIF
             NOW = DIFF
             IF (I .LT. 10000-1) THEN
                CALL EQN (CODE,TH+STEP,K1,K2,K3,K4,K5,A,B,D,E,AFT)
             ELSE
                AFT = NOW
             ENDIF
             IF (OLD-NOW .GT. 0) THEN
                FLAG1 = 'POS'
             ELSE
                FLAG1 = 'NEG'
             ENDIF
             IF (NOW-AFT .GT. 0) THEN
                FLAG2 = 'POS'
             ELSE
                FLAG2 = 'NEG'
```

```
            ENDIF
            IF (SIGNA .NE. SIGNB .AND. FLAG1 .EQ. FLAG2) THEN
               LRANGE(J) = HB
               RRANGE(J) = HA
               J = J + 1
            ENDIF
            OLD = NOW
            HB = HA
            SIGNB = SIGNA
           END DO
           NUM = J - 1
           WRITE(1,*) NUM
C
C          Compute the H's, A's and SI's for each guided mode
C
           DO I = 1, NUM
            A1(I) = 1.0
            P1 = LRANGE(I)
            P5 = RRANGE(I)
            REPEAT = 'TRUE '
            DO WHILE (REPEAT .EQ. 'TRUE ')
               P3 = (P5-P1)/2 + P1
               P2 = (P3-P1)/2 + P1
               P4 = (P5-P3)/2 + P3
               CALL EQN (CODE,P2,K1,K2,K3,K4,K5,A,B,D,E,FP2)
               IF (FP2 .GT. 0) THEN
                  SIGN2 = 'POS'
               ELSE
                  SIGN2 = 'NEG'
               ENDIF
               CALL EQN (CODE,P4,K1,K2,K3,K4,K5,A,B,D,E,FP4)
               IF (FP4 .GT. 0) THEN
                  SIGN4 = 'POS'
               ELSE
                  SIGN4 = 'NEG'
               ENDIF
               IF (SIGN2 .EQ. SIGN4) THEN
                  IF (FP2 .GT. FP4) THEN
                     IF (SIGN2 .EQ. 'POS') THEN
                        P1 = P4
                     ELSE
                        P5 = P2
                     ENDIF
                  ELSE
                     IF (SIGN2 .EQ. 'POS') THEN
                        P5 = P2
                     ELSE
                        P1 = P4
                     ENDIF
                  ENDIF
               ELSE
                  P1 = P2
                  P5 = P4
               ENDIF
               CALL EQN (CODE,P1,K1,K2,K3,K4,K5,A,B,D,E,FP1)
               CALL EQN (CODE,P5,K1,K2,K3,K4,K5,A,B,D,E,FP5)
               IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5) THEN
                  IF (ABS(FP1) .LT. ABS(FP5)) THEN
                     H1(I) = P1
```

```
                  DUM = FP1
               ELSE
                  H1(I) = P5
                  DUM = FP5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
         DUM = H1(I)**2
         IF (CODE .EQ. 1) THEN
           H2(I) = SQRT(K1 - K2 + DUM)
           H3(I) = SQRT(K3 - K1 - DUM)
           H4(I) = SQRT(K1 - K4 + DUM)
           H5(I) = SQRT(K1 - K5 + DUM)
           CON = (-1*H2(I)*DCOS(H3(I)*A)+H3(I)*DSIN(H3(I)*A))/
     $           (H3(I)*DCOS(H3(I)*A)+H2(I)*DSIN(H3(I)*A))
         ENDIF
         IF (CODE .EQ. 2) THEN
           H2(I) = SQRT(K2 - K1 - DUM)
           H3(I) = SQRT(K1 - K3 + DUM)
           H4(I) = SQRT(K1 - K4 + DUM)
           H5(I) = SQRT(K1 - K5 + DUM)
          CON = (H2(I)*DSIN(H2(I)*B)-H1(I)*DCOS(H2(I)*B))/
     $           (H1(I)*DSIN(H2(I)*B)+H2(I)*DCOS(H2(I)*B))
         ENDIF
         IF (CODE .EQ. 3) THEN
           H2(I) = SQRT(K1 - K2 + DUM)
           H3(I) = SQRT(K1 - K3 + DUM)
           H4(I) = SQRT(K4 - K1 - DUM)
           H5(I) = SQRT(K1 - K5 + DUM)
           CON = (H4(I)*DSIN(H4(I)*D)+H3(I)*DCOS(H4(I)*D))/
     $           (H3(I)*DSIN(H4(I)*D)-H4(I)*DCOS(H4(I)*D))
         ENDIF
         WRITE(1,*) H1(I), H2(I), H3(I), H4(I), H5(I), CON
        END DO
       ELSE
        NUM = 0
        WRITE(1,*) NUM
       ENDIF
      END DO
C
      CLOSE (UNIT=1)
      STOP
      END
C
      SUBROUTINE EQNHS (H,K1,K2,K3,K4,K5,A,B,D,E,DIFF)
C
C  Solve for H1 in each guided mode with sinusoidal behaviour
C  in medium 2,3,4 and exponential behaviour in medium 1,5.
C
      REAL*8  H1, H2, H3, H4, H5, A, B, D, E, DIFF, LEFT,
     $        RIGHT, K1, K2, K3, K4, K5, H
C
      H1 = H
      H2 = SQRT(K2 - K1 - H1**2)
      H3 = SQRT(K3 - K1 - H1**2)
      H4 = SQRT(K4 - K1 - H1**2)
```

```
      H5 = SQRT(K1 - K5 + H1**2)
      LEFT = DTAN(H3*(A+D)-DATAN((H2/H3)*
     $       DTAN(H2*(A-B)+DATAN(H1/H2))))
      RIGHT = (H4/H3)*DTAN(H4*(D-E)+DATAN(H5/H4))
      DIFF = LEFT - RIGHT
C
      RETURN
      END
C
      SUBROUTINE EQNH (CODE,H,K1,K2,K3,K4,K5,A,B,D,E,DIFF)
C
C  Solve for H1 in each guided mode with sinusoidal behaviour
C  in medium 2,4 and exponential behaviour in medium 1,3,5.
C
      REAL*8  H1, H2, H3, H4, H5, A, B, D, E, DIFF, LEFT,
     $        RIGHT, K1, K2, K3, K4, K5, H
      INTEGER CODE
C
      H1 = H
      H2 = SQRT(K2 - K1 - H1**2)
      H3 = SQRT(K1 - K3 + H1**2)
      H4 = SQRT(K4 - K1 - H1**2)
      H5 = SQRT(K1 - K5 + H1**2)
      IF (CODE .EQ. 1) THEN
        LEFT = H3/DTANH(H3*A) + H3/DTANH(H3*D)
        RIGHT = -1*H2*DTAN(H2*(A-B) + DATAN(H1/H2)) -
     $          H4*DTAN(H4*(D-E) + DATAN(H5/H4))
      ELSE
        LEFT = H3*DTANH(H3*A) + H3*DTANH(H3*D)
        RIGHT =  -1*H2*DTAN(H2*(A-B) + DATAN(H1/H2)) -
     $           H4*DTAN(H4*(D-E) + DATAN(H5/H4))
      ENDIF
      DIFF = LEFT - RIGHT
C
      RETURN
      END
C
      SUBROUTINE EQN (CODE,H,K1,K2,K3,K4,K5,A,B,D,E,DIFF)
C
C  Solve for H1 in each guided mode with sinusoidal behaviour
C  in medium 2 (or 3) (or 4) and exponential behaviour in
C  medium 1,3,4,5 (or 1,2,4,5) (or 1,2,3,5).
C
      REAL*8  H1, H2, H3, H4, H5, A, B, D, E, DIFF, LEFT,
     $        RIGHT, K1, K2, K3, K4, K5, H, DUM1, DUM2,
     $        DUM3, CON
      INTEGER CODE
C
      H1 = H
      IF (CODE .EQ. 1) THEN
        H2 = SQRT(K1 - K2 + H1**2)
        H3 = SQRT(K3 - K1 - H1**2)
        H4 = SQRT(K1 - K4 + H1**2)
        H5 = SQRT(K1 - K5 + H1**2)
        CON = (-1*H2*DCOS(H3*A)+H3*DSIN(H3*A))/
     $        (H3*DCOS(H3*A)+H2*DSIN(H3*A))
        LEFT = H3*(DSIN(H3*D) + CON*DCOS(H3*D))
        RIGHT = H4*(DCOS(H3*D) - CON*DSIN(H3*D))
      ENDIF
```

```
      IF (CODE .EQ. 2) THEN
        H2 = SQRT(K2 - K1 - H1**2)
        H3 = SQRT(K1 - K3 + H1**2)
        H4 = SQRT(K1 - K4 + H1**2)
        H5 = SQRT(K1 - K5 + H1**2)
        CON = (H2*DSIN(H2*B)-H1*DCOS(H2*B))/
     $        (H1*DSIN(H2*B)+H2*DCOS(H2*B))
        LEFT = -1*H2*(DSIN(H2*A) - CON*DCOS(H2*A))
        RIGHT = H3*(DCOS(H2*A) + CON*DSIN(H2*A))
      ENDIF
      IF (CODE .EQ. 3) THEN
        H2 = SQRT(K1 - K2 + H1**2)
        H3 = SQRT(K1 - K3 + H1**2)
        H4 = SQRT(K4 - K1 - H1**2)
        H5 = SQRT(K1 - K5 + H1**2)
        CON = (H4*DSIN(H4*D)+H3*DCOS(H4*D))/
     $        (H3*DSIN(H4*D)-H4*DCOS(H4*D))
        LEFT = H4*(DSIN(H4*E) + CON*DCOS(H4*E))
        RIGHT = H5*(DCOS(H4*E) - CON*DSIN(H4*E))
      ENDIF
      DIFF = LEFT - RIGHT
C
      RETURN
      END
```

```
C  PROGRAM : To find the weighting factor for each existed guided
C           mode of the parallel slab waveguides being excited
C           by plane wave.
C
      COMPLEX*8 COE(100), NUMER, DENO, DUM1, DUM2, DUM3
      REAL*8    N1, N2, N3, N4, N5, KO, PI, UPLIM, LOLIM,
     $          H1(100), H2(100), H3(100), H4(100), H5(100),
     $          A1(100), A2(100), A3(100), A4(100), A5(100),
     $          SI2(100), SI3(100), SI4(100), A, B, C, D, E,
     $          F, DUMO, CONS, CON(100)
      INTEGER I, J, ENUM, ONUM, NUM
C
      OPEN (UNIT=1,FILE='COPG.DAT',STATUS='OLD')
      OPEN (UNIT=2,FILE='COPGCOE.DAT',STATUS='NEW')
C
C  A-F : Define the location of the dielectric waveguides
C        and the reflectors
C  COE : Weighting factor for each guided mode
C  N1-N5 : Refractive indices for each medium
C  UPLIM : Upper limit where f(x) exists
C  LOLIM : Lower limit where f(x) exists
C  COPGCOE : File contains the weighting factor for each guided mode
C
C  Obtain information from the user and previous old file
C
      WRITE(*,*) 'What is the upper limit for f(x) (in meters)?'
      READ(*,*) UPLIM
      WRITE(*,*) 'What is the lower limit for f(x) (in meters)?'
      READ(*,*) LOLIM
      READ(1,*) N1, N2, N3, N4, N5, KO
      READ(1,*) A, B, C, D, E, F
      PI = 3.141592654
C
C  Compute the weights for all existed guided modes with sinusoidal
C  behaviour in medium 2,3,4 and exponential behaviour in medium 1,5.
C
      READ(1,*) NUM
      DO I = 1, NUM
         READ(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
         READ(1,*) A1(I), A2(I), A3(I), A4(I), A5(I)
         READ(1,*) SI2(I), SI3(I), SI4(I)
      END DO
      DO I = 1, NUM
         IF (UPLIM .GT. A) THEN
           IF (LOLIM .GE. A) THEN
              CALL NUMR (UPLIM,LOLIM,A2(I),H2(I),SI2(I),NUMER)
           ELSE IF (LOLIM .GE. -1*D) THEN
                   CALL NUMR (UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                   CALL NUMR (A,LOLIM,A3(I),H3(I),SI3(I),DUM2)
                   NUMER = DUM1 + DUM2
                ELSE IF (LOLIM .GE. -1*E) THEN
                        CALL NUMR (UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                        CALL NUMR (A,-1*D,A3(I),H3(I),SI3(I),DUM2)
                        CALL NUMR (-1*D,LOLIM,A4(I),H4(I),SI4(I),DUM3)
                        NUMER = DUM1 + DUM2 + DUM3
           ENDIF
         ELSE IF (UPLIM .GT. -1*D) THEN
                IF (LOLIM .GE. -1*D) THEN
                   CALL NUMR (UPLIM,LOLIM,A3(I),H3(I),SI3(I),NUMER)
```

```
                ELSE IF (LOLIM .GE. -1*E) THEN
                      CALL NUMR (UPLIM,-1*D,A3(I),H3(I),SI3(I),DUM1)
                      CALL NUMR (-1*D,LOLIM,A4(I),H4(I),SI4(I),DUM2)
                      NUMER = DUM1 + DUM2
                ENDIF
              ELSE IF (LOLIM .GE. -1*E) THEN
                   CALL NUMR (UPLIM,LOLIM,A4(I),H4(I),SI4(I),NUMER)
          ENDIF
          CALL DENOMINS (A1(I),A2(I),A3(I),A4(I),A5(I),H1(I),H2(I),
     $                    H3(I),H4(I),H5(I),A,B,D,E,SI2(I),SI3(I),
     $                    SI4(I),DENO)
          COE(I) = NUMER/DENO
          WRITE(2,*) COE(I)
        END DO
C
C  Compute the weights for all the existed guided modes
C  with sinusoidal behaviour in medium 2,4 and exponential
C  behaviour in medium 1,3,5.
C
        DO J = 1, 2
         READ(1,*) NUM
         DO I = 1, NUM
           READ(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
           READ(1,*) A1(I), A2(I), A3(I), A4(I), A5(I)
           READ(1,*) SI2(I), SI4(I)
         END DO
         DO I = 1, NUM
           IF (UPLIM .GT. A) THEN
             IF (LOLIM .GE. A) THEN
                CALL NUMR24 (UPLIM,LOLIM,A2(I),H2(I),SI2(I),NUMER)
             ELSE IF (LOLIM .GE. -1*D) THEN
                     CALL NUMR24 (UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                     CALL NUMR3 (J,A,LOLIM,A3(I),H3(I),DUM2)
                     NUMER = DUM1 + DUM2
                ELSE IF (LOLIM .GE. -1*E) THEN
                      CALL NUMR24 (UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                      CALL NUMR3 (J,A,-1*D,A3(I),H3(I),DUM2)
                      CALL NUMR24 (-1*D,LOLIM,A4(I),H4(I),SI4(I),DUM3)
                      NUMER = DUM1 + DUM2 + DUM3
             ENDIF
           ELSE IF (UPLIM .GT. -1*D) THEN
                  IF (LOLIM .GE. -1*D) THEN
                     CALL NUMR3 (J,UPLIM,LOLIM,A3(I),H3(I),NUMER)
                  ELSE IF (LOLIM .GE. -1*E) THEN
                        CALL NUMR3 (J,UPLIM,-1*D,A3(I),H3(I),DUM1)
                        CALL NUMR24 (-1*D,LOLIM,A4(I),H4(I),SI4(I),DUM2)
                        NUMER = DUM1 + DUM2
                  ENDIF
                ELSE IF (LOLIM .GE. -1*E) THEN
                     CALL NUMR24 (UPLIM,LOLIM,A4(I),H4(I),SI4(I),NUMER)
           ENDIF
           CALL DENOMIN (J,A1(I),A2(I),A3(I),A4(I),A5(I),H1(I),H2(I),
     $                    H3(I),H4(I),H5(I),SI2(I),SI4(I),A,B,D,E,DENO)
           COE(I) = NUMER/DENO
           WRITE(2,*) COE(I)
         END DO
        END DO
C
C  Compute the weights for all the existed guided modes with
```

```
C  sinusoidal behaviour in medium 2 (or 3 or 4) and exponential
C  behaviour in medium 1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
      DO J = 1, 3
       READ(1,*) NUM
       DO I = 1, NUM
         READ(1,*) H1(I),H2(I),H3(I),H4(I),H5(I),CON(I)
       END DO
       DO I = 1, NUM
         IF (UPLIM .GT. A) THEN
           IF (LOLIM .GE. A) THEN
              IF (J .EQ. 1) THEN
                CONS = (DCOS(H3(I)*A)+CON(I)*
     $                  DSIN(H3(I)*A))*DEXP(H2(I)*A)
              ELSE IF (J .EQ. 2) THEN
                    CONS = CON(I)
                   ELSE
                    CONS = (DCOS(H4(I)*D)-CON(I)*DSIN(H4(I)*D))*
     $                     DEXP(-1*H3(I)*(A+D))*DEXP(H2(I)*A)
              ENDIF
              CALL REG2 (J,UPLIM,LOLIM,H2(I),CONS,NUMER)
           ELSE IF (LOLIM .GE. -1*D) THEN
                  IF (J .EQ. 1) THEN
                    CONS = (DCOS(H3(I)*A)+CON(I)*
     $                     DSIN(H3(I)*A))*DEXP(H2(I)*A)
                  ELSE IF (J .EQ. 2) THEN
                        CONS = CON(I)
                       ELSE
                        CONS = (DCOS(H4(I)*D)-CON(I)*DSIN(H4(I)*D))
     $                         *DEXP(-1*H3(I)*(A+D))*DEXP(H2(I)*A)
                  ENDIF
                  CALL REG2 (J,UPLIM,A,H2(I),CONS,DUM1)
                  IF (J .EQ. 1) THEN
                     CONS = CON(I)
                  ELSE IF (J .EQ. 2) THEN
                         CONS = (DCOS(H2(I)*A)+CON(I)*
     $                           DSIN(H2(I)*A))*DEXP(-1*H3(I)*A)
                       ELSE
                         CONS = (DCOS(H4(I)*D)-CON(I)*
     $                           DSIN(H4(I)*D))*DEXP(-1*H3(I)*D)
                  ENDIF
                  CALL REG3 (J,A,LOLIM,H3(I),CONS,DUM2)
                  NUMER = DUM1 + DUM2
                ELSE IF (LOLIM .GE. -1*E) THEN
                      IF (J .EQ. 1) THEN
                        CONS = (DCOS(H3(I)*A)+CON(I)*
     $                          DSIN(H3(I)*A))*DEXP(H2(I)*A)
                      ELSE IF (J .EQ. 2) THEN
                             CONS = CON(I)
                           ELSE
                             CONS = (DCOS(H4(I)*D)-CON(I)*
     $                               DSIN(H4(I)*D))*DEXP(-1*
     $                               H3(I)*(A+D))*DEXP(H2(I)*A)
                      ENDIF
                      CALL REG2 (J,UPLIM,A,H2(I),CONS,DUM1)
                      IF (J .EQ. 1) THEN
                        CONS = CON(I)
                      ELSE IF (J .EQ. 2) THEN
                             CONS = (DCOS(H2(I)*A)+CON(I)*
```

```
$                                DSIN(H2(I)*A))*DEXP(-1*H3(I)*A)
                          ELSE
                            CONS = (DCOS(H4(I)*D)-CON(I)*
$                                DSIN(H4(I)*D))*DEXP(-1*H3(I)*D)
                        ENDIF
                        CALL REG3 (J,A,-1*D,H3(I),CONS,DUM2)
                        IF (J .EQ. 1) THEN
                          CONS = (DCOS(H3(I)*D)-CON(I)*
$                              DSIN(H3(I)*D))*DEXP(H4(I)*D)
                        ELSE IF (J .EQ. 2) THEN
                               CONS = (DCOS(H2(I)*A)+CON(I)*
$                                  DSIN(H2(I)*A))*DEXP(-1*
$                                  H3(I)*(A+D))*DEXP(H4(I)*D)
                          ELSE
                            CONS = CON(I)
                        ENDIF
                        CALL REG4 (J,-1*D,LOLIM,H4(I),CONS,DUM3)
                        NUMER = DUM1 + DUM2 + DUM3
          ENDIF
        ELSE IF (UPLIM .GT. -1*D) THEN
                IF (LOLIM .GE. -1*D) THEN
                   IF (J .EQ. 1) THEN
                     CONS = CON(I)
                   ELSE IF (J .EQ. 2) THEN
                          CONS = (DCOS(H2(I)*A)+CON(I)*
$                                DSIN(H2(I)*A))*DEXP(-1*H3(I)*A)
                      ELSE
                        CONS = (DCOS(H4(I)*D)-CON(I)*
$                               DSIN(H4(I)*D))*DEXP(-1*H3(I)*D)
                   ENDIF
                   CALL REG3 (J,UPLIM,LOLIM,H3(I),CONS,NUMER)
                ELSE IF (LOLIM .GE. -1*E) THEN
                      IF (J .EQ. 1) THEN
                        CONS = CON(I)
                      ELSE IF (J .EQ. 2) THEN
                             CONS = (DCOS(H2(I)*A)+CON(I)*
$                                  DSIN(H2(I)*A))*DEXP(-1*H3(I)*A)
                         ELSE
                           CONS = (DCOS(H4(I)*D)-CON(I)*
$                                  DSIN(H4(I)*D))*DEXP(-1*H3(I)*D)
                      ENDIF
                      CALL REG3 (J,UPLIM,-1*D,H3(I),CONS,DUM1)
                      IF (J .EQ. 1) THEN
                          CONS = (DCOS(H3(I)*D)-CON(I)*
$                                DSIN(H3(I)*D))*DEXP(H4(I)*D)
                      ELSE IF (J .EQ. 2) THEN
                             CONS = (DCOS(H2(I)*A)+CON(I)*
$                                   DSIN(H2(I)*A))*DEXP(-1*H3(I)*
$                                   (A+D))*DEXP(H4(I)*D)
                         ELSE
                           CONS = CON(I)
                      ENDIF
                      CALL REG4 (J,-1*D,LOLIM,H4(I),CONS,DUM2)
                      NUMER = DUM1 + DUM2
                ENDIF
              ELSE IF (LOLIM .GE. -1*E) THEN
                     IF (J .EQ. 1) THEN
                         CONS = (DCOS(H3(I)*D)-CON(I)*
$                              DSIN(H3(I)*D))*DEXP(H4(I)*D)
```

```
                        ELSE IF (J .EQ. 2) THEN
                                CONS = (DCOS(H2(I)*A)+CON(I)*
     $                                 DSIN(H2(I)*A))*DEXP(-1*H3(I)*
     $                                 (A+D))*DEXP(H4(I)*D)
                             ELSE
                                CONS = CON(I)
                        ENDIF
                        CALL REG4 (J,UPLIM,LOLIM,H4(I),CONS,NUMER)
              ENDIF
              CALL DENOM (J,H1(I),H2(I),H3(I),H4(I),H5(I),CON(I),A,B,D,
     $                    E,DENO)
              COE(I) = NUMER/DENO
              WRITE(2,*) COE(I)
           END DO
        END DO
C
        CLOSE (UNIT=1)
        CLOSE (UNIT=2)
        STOP
        END
C
        SUBROUTINE NUMR (UP,LO,A,H,SI,ANSWER)
C
C Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C For UP and LO exists in medium 2,3 or 4 with Ey(x) having sinusoidal
C behaviour in medium 2,3,4 and exponential behaviour in medium 1,5)
C
        COMPLEX*8 ANSWER
        REAL*8    UP, LO, A, H, SI
C
        ANSWER = CMPLX(A*(DSIN(H*UP-SI) - DSIN(H*LO-SI))/H)
C
        RETURN
        END
C
        SUBROUTINE DENOMINS (A1,A2,A3,A4,A5,H1,H2,H3,H4,H5,
     $                       A,B,D,E,SI2,SI3,SI4,ANS)
C
C  Solve for ||Ey(x)**2|| between x = negative infinity and x =
C  positive infinity. (For Ey(x) having sinusoidal behaviour in
C  medium 2,3,4 and exponential behaviour in medium 1,5)
C
        COMPLEX*8 ANS
        REAL*8    A1, A2, A3, A4, A5, H1, H2, H3, H4, H5, A, B,
     $            D, E, DUM, DUM1, DUM2, DUM3, DUM4, SI2, SI3, SI4
C
        DUM1 = A1**2/(2*H1) + A5**2/(2*H5)
        DUM = (DSIN(2*(H2*B-SI2)) - DSIN(2*(H2*A-SI2)))/(2*H2)
        DUM2 = (A2**2)*(B - A + DUM)/2
        DUM = (DSIN(2*(H3*A-SI3)) + DSIN(2*(H3*D+SI3)))/(2*H3)
        DUM3 = (A3**2)*(A + D + DUM)/2
        DUM = (DSIN(2*(H4*D+SI4)) - DSIN(2*(H4*E+SI4)))/(2*H4)
        DUM4 = (A4**2)*(E - D - DUM)/2
        ANS = CMPLX(DUM1 + DUM2 + DUM3 + DUM4)
C
        RETURN
        END
C
```

```
      SUBROUTINE NUMR24 (UP,LO,A,H,SI,ANSWER)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C  For UP and LO exists in medium 2 or 4 with Ey(x) having sinusoidal
C  behaviour in medium 2,4 and exponential behaviour in medium 1,3,5.
C
      COMPLEX*8 ANSWER
      REAL*8    UP, LO, A, H, SI
C
      ANSWER = CMPLX(A*(DSIN(H*UP-SI) - DSIN(H*LO-SI))/H)
C
      RETURN
      END
C
      SUBROUTINE NUMR3 (J,UP,LO,A3,H3,ANSWER)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C  For Up and LO exists in medium 3 with Ey(x) having sinusoidal
C  behaviour in medium 2,4 and exponential behaviour in medium 1,3,5.
C
      COMPLEX*8 ANSWER
      REAL*8    UP, LO, A3, H3
      INTEGER   J
C
      IF (J .EQ. 1) THEN
         ANSWER = CMPLX(A3*(DCOSH(H3*UP) - DCOSH(H3*LO))/H3)
      ELSE
         ANSWER = CMPLX(A3*(DSINH(H3*UP) - DSINH(H3*LO))/H3)
      ENDIF
C
      RETURN
      END
C
      SUBROUTINE DENOMIN (J,A1,A2,A3,A4,A5,H1,H2,H3,H4,H5,
     $                    SI2,SI4,A,B,D,E,ANS)
C
C  Solve for ||Ey(x)**2|| between x = negative infinity and x =
C  positive infinity. (For Ey(x) having sinusoidal behaviour in
C  medium 2,4 and exponential behaviour in medium 1,3,5)
C
      COMPLEX*8 ANS
      REAL*8    A1, A2, A3, A4, A5, H1, H2, H3, H4, H5, A, B,
     $          D, E, DUM, DUM1, DUM2, DUM3, DUM4, SI2, SI4
      INTEGER   J
C
      DUM1 = A1**2/(2*H1) + A5**2/(2*H5)
      DUM = (DSIN(2*(H2*B-SI2)) - DSIN(2*(H2*A-SI2)))/(2*H2)
      DUM2 = (A2**2)*(B - A + DUM)/2
      DUM = (DSIN(2*(H4*D+SI4)) - DSIN(2*(H4*E+SI4)))/(2*H4)
      DUM4 = (A4**2)*(E - D - DUM)/2
      DUM = DEXP(2*H3*A)+DEXP(2*H3*D)-
     $      DEXP(-1*2*H3*A)-DEXP(-1*2*H3*D)
      IF (J .EQ. 1) THEN
         DUM3 = (A3**2)*(DUM/(2*H3) - 2*(A+D))/4.
      ELSE
         DUM3 = (A3**2)*(DUM/(2*H3) + 2*(A+D))/4.
      ENDIF
      ANS = CMPLX(DUM1 + DUM2 + DUM3 + DUM4)
```

```
C
      RETURN
      END
C
      SUBROUTINE REG2 (J,UP,LO,H,CS,ANS)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C  For UP and LO exists in medium 2 with Ey(x) having sinusoidal
C  behaviour in medium 2 (or 3 or 4) and exponential behaviour in
C  medium 1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
      COMPLEX*8 ANS
      REAL*8    UP, LO, H, CS
      INTEGER   J
C
      IF (J .EQ. 1 .OR. J .EQ. 3) THEN
         ANS = CMPLX(CS)*CMPLX((DEXP(-1*H*LO)-DEXP(-1*H*UP))/H)
      ELSE IF (J .EQ. 2) THEN
            ANS = CMPLX(((DSIN(H*UP)-CS*DCOS(H*UP)) -
     $              (DSIN(H*LO)-CS*DCOS(H*LO)))/H)
      ENDIF
C
      RETURN
      END
C
      SUBROUTINE REG3 (J,UP,LO,H,CONS,ANS)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C  For UP and LO exists in medium 3 with Ey(x) having sinusoidal
C  behaviour in medium 2 (or 3 or 4) and exponential behaviour in
C  medium 1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
      COMPLEX*8 ANS
      REAL*8    UP, LO, A, H, CONS, DUM1, DUM2
      INTEGER   J
C
      IF (J .EQ. 1) THEN
         DUM1 = DSIN(H*UP) - CONS*DCOS(H*UP)
         DUM2 = DSIN(H*LO) - CONS*DCOS(H*LO)
         ANS = CMPLX((DUM1-DUM2)/H)
      ENDIF
      IF (J .EQ. 2) THEN
         ANS = CMPLX(CONS*(DEXP(H*UP)-DEXP(H*LO))/H)
      ENDIF
C
      RETURN
      END
C
      SUBROUTINE REG4 (J,UP,LO,H,CS,ANS)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C  For UP and LO exists in medium 4 with Ey(x) having sinusoidal
C  behaviour in medium 2 (or 3 or 4) and exponential behaviour in
C  medium 1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
      COMPLEX*8 ANS
      REAL*8    UP, LO, H, CS, DUM1, DUM2
      INTEGER   J
```

```
C
      IF (J .EQ. 1 .OR. J .EQ. 2) THEN
        ANS = CMPLX(CS)*CMPLX((DEXP(H*UP)-DEXP(H*LO))/H)
      ELSE
        DUM1 = DSIN(H*UP) - CS*DCOS(H*UP)
        DUM2 = DSIN(H*LO) - CS*DCOS(H*LO)
        ANS = CMPLX((DUM1-DUM2)/H)
      ENDIF
C
      RETURN
      END
C
      SUBROUTINE DENOM (J,H1,H2,H3,H4,H5,CS,A,B,D,E,ANS)
C
C  Solve for ||Ey(x)**2|| between x = negative infinity and x =
C  positive infinity. For Ey(x) having sinusoidal behaviour in
C  medium 2 (or 3 or 4) and exponential behaviour in medium
C  1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
      COMPLEX*8   ANS
      REAL*8      H1, H2, H3, H4, H5, A, B, D, E,
     $            DUM, DUM1, DUM2, DUM3, DUM4, CS
      INTEGER     J
C
      IF (J .EQ. 1) THEN
         DUM = (DCOS(H3*A) + CS*DSIN(H3*A))*DEXP(-1*H2*(B-A))
         DUM1 = (DUM**2)/(2*H1)
         DUM2 = ((DUM/DEXP(-1*H2*(B-A)))**2)*
     $          (1.-DEXP(-1*2*H2*(B-A)))/(2*H2)
         DUM = (DCOS(H3*D) - CS*DSIN(H3*D))*DEXP(H4*(D-E))
         DUM1 = DUM1 + (DUM**2)/(2*H5)
         DUM4 = ((DUM/DEXP(H4*(D-E)))**2)*
     $          (1.-DEXP(2*H4*(D-E)))/(2*H4)
         DUM3 = (1+CS**2)*(A+D)/2. + (1-CS**2)*(DSIN(2*H3*A)+
     $          DSIN(2*H3*D))/(4*H3) + CS*(DCOS(2*H3*D)-
     $          DCOS(2*H3*A))/(2*H3)
      ENDIF
      IF (J .EQ. 2) THEN
         DUM = DCOS(H2*B) + CS*DSIN(H2*B)
         DUM1 = (DUM**2)/(2*H1)
         DUM = (DCOS(H2*A)+CS*DSIN(H2*A))*DEXP(-1*H3*(A+D))*
     $         DEXP(H4*(D-E))
         DUM1 = DUM1 + (DUM**2)/(2*H5)
         DUM2 = (1+CS**2)*(B-A)/2. +
     $          (1-CS**2)*(DSIN(2*H2*B)-DSIN(2*H2*A))/(4*H2) -
     $          CS*(DCOS(2*H2*B)-DCOS(2*H2*A))/(2*H2)
         DUM = DCOS(H2*A) + CS*DSIN(H2*A)
         DUM3 = (DUM**2)*(1.- DEXP(-2*H3*(A+D)))/(2*H3)
         DUM4 = ((DUM*DEXP(-1*H3*(A+D)))**2)*
     $          (1.- DEXP(2*H4*(D-E)))/(2*H4)
      ENDIF
      IF (J .EQ. 3) THEN
         DUM = DCOS(H4*E) - CS*DSIN(H4*E)
         DUM1 = (DUM**2)/(2*H5)
         DUM = (DCOS(H4*D)-CS*DSIN(H4*D))*DEXP(-1*H3*(A+D))*
     $         DEXP(-1*H2*(B-A))
         DUM1 = DUM1 + (DUM**2)/(2*H1)
         DUM = DCOS(H4*D) - CS*DSIN(H4*D)
```

```
            DUM3 = (DUM**2)*(1.- DEXP(-2*H3*(A+D)))/(2*H3)
            DUM2 = ((DUM*DEXP(-1*H3*(A+D)))**2)*
     $             (1.- DEXP(-2*H2*(B-A)))/(2*H2)
            DUM4 = (1+CS**2)*(E-D)/2. +
     $             (1-CS**2)*(DSIN(2*H4*E)-DSIN(2*H4*D))/(4*H4) -
     $             CS*(DCOS(2*H4*D)-DCOS(2*H4*E))/(2*H4)
         ENDIF
         ANS = CMPLX(DUM1 + DUM2 + DUM3 + DUM4)
C
         RETURN
         END
```

```
C  PROGRAM : To find the weighting factor for each existed guided mode
C           of the parallel waveguides being excited by focused wave.
C
      COMPLEX*8 COE(100), NUMER, DENO, DUM1, DUM2, DUM3, U(0:10000)
      REAL*8    N1, N2, N3, N4, N5, KO, PI, UPLIM, LOLIM, H1(100),
     $          H2(100), H3(100), H4(100), H5(100), A1(100), A2(100),
     $          A3(100), A4(100), A5(100), SI2(100), SI3(100),
     $          SI4(100), A, B, C, D, E, F, DUMO, CONS, CON(100),
     $          FOCAL, ZPRIM, SEP, X, UU, LL, UPL, LOL
      INTEGER I, J, ENUM, ONUM, NUM, UC
C
C  U : Electric field of the input wave at z=0
C  A-F : Define the location of the dielectric waveguides
C        and the reflectors
C  COE : Weighting factor for each guided mode
C  FOCAL : Focal length of the lens
C  UPLIM : Upper limit where f(x) exists
C  LOLIM : Lower limit where f(x) exists
C  N1-N5 : Refractive indices for each medium
C  ZPRIM : Distance between the lens and the slab waveguides
C  UPL,LOL : Location of the lens
C  COPGCOE : File contains the weighting factor for each guided mode
C
      OPEN (UNIT=1,FILE='COPG.DAT',STATUS='OLD')
      OPEN (UNIT=2,FILE='COPGCOE.DAT',STATUS='NEW')
C
C  Obtain information from the user
C
      WRITE(*,*) 'What is the upper limit for f(x) (in meters)?'
      READ(*,*) UPLIM
      WRITE(*,*) 'What is the lower limit for f(x) (in meters)?'
      READ(*,*) LOLIM
      WRITE(*,*) 'What is the focal length of the lens (in meters)?'
      READ(*,*) FOCAL
      WRITE(*,*) 'What is the location of the lens ?'
      WRITE(*,*) '(between x = UPL and x = LOL)'
      WRITE(*,*) 'UPL = ?'
      READ(*,*) UPL
      WRITE(*,*) 'LOL = ?'
      READ(*,*) LOL
      WRITE(*,*) 'What is the distance between the lens and the',
     $           ' slab waveguides (in meters) ?'
      READ(*,*) ZPRIM
      READ(1,*) N1, N2, N3, N4, N5, KO
      READ(1,*) A, B, C, D, E, F
      PI = 3.141592654
C
C  Compute the electric field of the input wave at z = 0.
C
      UC = 50
      SEP = (UPLIM-LOLIM)/(2*UC)
      UU = (UPLIM-LOLIM)/2
      LL = -1*UU
      CALL UFIELD (LL,FOCAL,KO,ZPRIM,UPL,LOL,U(0))
      CALL UFIELD (UU,FOCAL,KO,ZPRIM,UPL,LOL,U(2*UC))
      DO J = 1, 2*UC-1
         X = LL + J*SEP
         CALL UFIELD (X,FOCAL,KO,ZPRIM,UPL,LOL,U(J))
```

```
      END DO
C
C  Compute the weights for all existed guided modes with sinusodial
C  behaviour in medium 2,3,4 and exponential behaviour in medium 1,5.
C
      READ(1,*) NUM
      DO I = 1, NUM
         READ(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
         READ(1,*) A1(I), A2(I), A3(I), A4(I), A5(I)
         READ(1,*) SI2(I), SI3(I), SI4(I)
      END DO
      DO I = 1, NUM
         IF (UPLIM .GT. A) THEN
           IF (LOLIM .GE. A) THEN
              CALL NUMR (U,UC,UPLIM,LOLIM,A2(I),H2(I),SI2(I),NUMER)
           ELSE IF (LOLIM .GE. -1*D) THEN
                  CALL NUMR (U,UC,UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                  CALL NUMR (U,UC,A,LOLIM,A3(I),H3(I),SI3(I),DUM2)
                  NUMER = DUM1 + DUM2
                ELSE IF (LOLIM .GE. -1*E) THEN
                       CALL NUMR (U,UC,UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                       CALL NUMR (U,UC,A,-1*D,A3(I),H3(I),SI3(I),DUM2)
                       CALL NUMR (U,UC,-1*D,LOLIM,A4(I),H4(I),
     $                                  SI4(I),DUM3)
                       NUMER = DUM1 + DUM2 + DUM3
           ENDIF
         ELSE IF (UPLIM .GT. -1*D) THEN
                IF (LOLIM .GE. -1*D) THEN
                 CALL NUMR (U,UC,UPLIM,LOLIM,A3(I),H3(I),SI3(I),NUMER)
                ELSE IF (LOLIM .GE. -1*E) THEN
                       CALL NUMR (U,UC,UPLIM,-1*D,A3(I),H3(I),
     $                                  SI3(I),DUM1)
                       CALL NUMR (U,UC,-1*D,LOLIM,A4(I),H4(I),
     $                                  SI4(I),DUM2)
                       NUMER = DUM1 + DUM2
                ENDIF
              ELSE IF (LOLIM .GE. -1*E) THEN
                     CALL NUMR (U,UC,UPLIM,LOLIM,A4(I),H4(I),
     $                                SI4(I),NUMER)
         ENDIF
         CALL DENOMINS (A1(I),A2(I),A3(I),A4(I),A5(I),H1(I),H2(I),
     $                   H3(I),H4(I),H5(I),A,B,D,E,SI2(I),SI3(I),
     $                   SI4(I),DENO)
         COE(I) = NUMER/DENO
         WRITE(2,*) COE(I)
      END DO
C
C  Compute the weights for all existed guided modes with sinusoidal
C  behaviour in medium 2,4 and exponential behaviour in medium 1,3,5.
C
      DO J = 1, 2
       READ(1,*) NUM
       DO I = 1, NUM
         READ(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
         READ(1,*) A1(I), A2(I), A3(I), A4(I), A5(I)
         READ(1,*) SI2(I), SI4(I)
       END DO
       DO I = 1, NUM
         IF (UPLIM .GT. A) THEN
```

```
            IF (LOLIM .GE. A) THEN
             CALL NUMR24 (U,UC,UPLIM,LOLIM,A2(I),H2(I),SI2(I),NUMER)
            ELSE IF (LOLIM .GE. -1*D) THEN
                  CALL NUMR24 (U,UC,UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                  CALL NUMR3 (U,UC,J,A,LOLIM,A3(I),H3(I),DUM2)
                  NUMER = DUM1 + DUM2
                 ELSE IF (LOLIM .GE. -1*E) THEN
                        CALL NUMR24 (U,UC,UPLIM,A,A2(I),H2(I),
     $                                SI2(I),DUM1)
                        CALL NUMR3 (U,UC,J,A,-1*D,A3(I),H3(I),DUM2)
                        CALL NUMR24 (U,UC,-1*D,LOLIM,A4(I),H4(I),
     $                                SI4(I),DUM3)
                        NUMER = DUM1 + DUM2 + DUM3
            ENDIF
          ELSE IF (UPLIM .GT. -1*D) THEN
                IF (LOLIM .GE. -1*D) THEN
                  CALL NUMR3 (U,UC,J,UPLIM,LOLIM,A3(I),H3(I),NUMER)
                 ELSE IF (LOLIM .GE. -1*E) THEN
                       CALL NUMR3 (U,UC,J,UPLIM,-1*D,A3(I),H3(I),DUM1)
                       CALL NUMR24 (U,UC,-1*D,LOLIM,A4(I),H4(I),
     $                             SI4(I),DUM2)
                       NUMER = DUM1 + DUM2
                 ENDIF
                ELSE IF (LOLIM .GE. -1*E) THEN
                       CALL NUMR24 (U,UC,UPLIM,LOLIM,A4(I),H4(I),
     $                             SI4(I),NUMER)
          ENDIF
          CALL DENOMIN (J,A1(I),A2(I),A3(I),A4(I),A5(I),H1(I),H2(I),
     $                  H3(I),H4(I),H5(I),SI2(I),SI4(I),A,B,D,E,DENO)
          COE(I) = NUMER/DENO
          WRITE(2,*) COE(I)
        END DO
      END DO
C
C  Compute the weights for all existed guided modes with sinusoidal
C  behaviour in medium 2 (or 3 or 4) and exponential behaviour in
C  medium 1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
      DO J = 1, 3
       READ(1,*) NUM
       DO I = 1, NUM
         READ(1,*) H1(I),H2(I),H3(I),H4(I),H5(I),CON(I)
       END DO
       DO I = 1, NUM
         IF (UPLIM .GT. A) THEN
           IF (LOLIM .GE. A) THEN
              IF (J .EQ. 1) THEN
                CONS = (DCOS(H3(I)*A)+CON(I)*
     $                  DSIN(H3(I)*A))*DEXP(H2(I)*A)
              ELSE IF (J .EQ. 2) THEN
                    CONS = CON(I)
                   ELSE
                    CONS = (DCOS(H4(I)*D)-CON(I)*DSIN(H4(I)*D))*
     $                      DEXP(-1*H3(I)*(A+D))*DEXP(H2(I)*A)
              ENDIF
              CALL REG2 (U,UC,J,UPLIM,LOLIM,H2(I),CONS,NUMER)
           ELSE IF (LOLIM .GE. -1*D) THEN
                  IF (J .EQ. 1) THEN
                    CONS = (DCOS(H3(I)*A)+CON(I)*
```

```
$                    DSIN(H3(I)*A))*DEXP(H2(I)*A)
              ELSE IF (J .EQ. 2) THEN
                    CONS = CON(I)
                   ELSE
                    CONS = (DCOS(H4(I)*D)-CON(I)*DSIN(H4(I)*D))
$                          *DEXP(-1*H3(I)*(A+D))*DEXP(H2(I)*A)
              ENDIF
              CALL REG2 (U,UC,J,UPLIM,A,H2(I),CONS,DUM1)
              IF (J .EQ. 1) THEN
                CONS = CON(I)
              ELSE IF (J .EQ. 2) THEN
                    CONS = (DCOS(H2(I)*A)+CON(I)*
$                           DSIN(H2(I)*A))*DEXP(-1*H3(I)*A)
                   ELSE
                    CONS = (DCOS(H4(I)*D)-CON(I)*
$                           DSIN(H4(I)*D))*DEXP(-1*H3(I)*D)
              ENDIF
              CALL REG3 (U,UC,J,A,LOLIM,H3(I),CONS,DUM2)
              NUMER = DUM1 + DUM2
            ELSE IF (LOLIM .GE. -1*E) THEN
                 IF (J .EQ. 1) THEN
                   CONS = (DCOS(H3(I)*A)+CON(I)*
$                          DSIN(H3(I)*A))*DEXP(H2(I)*A)
                 ELSE IF (J .EQ. 2) THEN
                        CONS = CON(I)
                      ELSE
                        CONS = (DCOS(H4(I)*D)-CON(I)*
$                               DSIN(H4(I)*D))*DEXP(-1*
$                               H3(I)*(A+D))*DEXP(H2(I)*A)
                 ENDIF
                 CALL REG2 (U,UC,J,UPLIM,A,H2(I),CONS,DUM1)
                 IF (J .EQ. 1) THEN
                   CONS = CON(I)
                 ELSE IF (J .EQ. 2) THEN
                        CONS = (DCOS(H2(I)*A)+CON(I)*
$                              DSIN(H2(I)*A))*DEXP(-1*H3(I)*A)
                      ELSE
                        CONS = (DCOS(H4(I)*D)-CON(I)*
$                              DSIN(H4(I)*D))*DEXP(-1*H3(I)*D)
                 ENDIF
                 CALL REG3 (U,UC,J,A,-1*D,H3(I),CONS,DUM2)
                 IF (J .EQ. 1) THEN
                   CONS = (DCOS(H3(I)*D)-CON(I)*
$                         DSIN(H3(I)*D))*DEXP(H4(I)*D)
                 ELSE IF (J .EQ. 2) THEN
                        CONS = (DCOS(H2(I)*A)+CON(I)*
$                               DSIN(H2(I)*A))*DEXP(-1*
$                               H3(I)*(A+D))*DEXP(H4(I)*D)
                      ELSE
                        CONS = CON(I)
                 ENDIF
                 CALL REG4 (U,UC,J,-1*D,LOLIM,H4(I),CONS,DUM3)
                 NUMER = DUM1 + DUM2 + DUM3
        ENDIF
      ELSE IF (UPLIM .GT. -1*D) THEN
             IF (LOLIM .GE. -1*D) THEN
                IF (J .EQ. 1) THEN
                  CONS = CON(I)
                ELSE IF (J .EQ. 2) THEN
```

```
                        CONS = (DCOS(H2(I)*A)+CON(I)*
     $                         DSIN(H2(I)*A))*DEXP(-1*H3(I)*A)
                      ELSE
                        CONS = (DCOS(H4(I)*D)-CON(I)*
     $                         DSIN(H4(I)*D))*DEXP(-1*H3(I)*D)
                   ENDIF
                   CALL REG3 (U,UC,J,UPLIM,LOLIM,H3(I),CONS,NUMER)
                ELSE IF (LOLIM .GE. -1*E) THEN
                      IF (J .EQ. 1) THEN
                        CONS = CON(I)
                      ELSE IF (J .EQ. 2) THEN
                            CONS = (DCOS(H2(I)*A)+CON(I)*
     $                              DSIN(H2(I)*A))*DEXP(-1*H3(I)*A)
                          ELSE
                            CONS = (DCOS(H4(I)*D)-CON(I)*
     $                              DSIN(H4(I)*D))*DEXP(-1*H3(I)*D)
                      ENDIF
                      CALL REG3 (U,UC,J,UPLIM,-1*D,H3(I),CONS,DUM1)
                      IF (J .EQ. 1) THEN
                          CONS = (DCOS(H3(I)*D)-CON(I)*
     $                           DSIN(H3(I)*D))*DEXP(H4(I)*D)
                      ELSE IF (J .EQ. 2) THEN
                             CONS = (DCOS(H2(I)*A)+CON(I)*
     $                               DSIN(H2(I)*A))*DEXP(-1*H3(I)*
     $                               (A+D))*DEXP(H4(I)*D)
                          ELSE
                            CONS = CON(I)
                      ENDIF
                      CALL REG4 (U,UC,J,-1*D,LOLIM,H4(I),CONS,DUM2)
                      NUMER = DUM1 + DUM2
                ENDIF
             ELSE IF (LOLIM .GE. -1*E) THEN
                    IF (J .EQ. 1) THEN
                       CONS = (DCOS(H3(I)*D)-CON(I)*
     $                        DSIN(H3(I)*D))*DEXP(H4(I)*D)
                    ELSE IF (J .EQ. 2) THEN
                           CONS = (DCOS(H2(I)*A)+CON(I)*
     $                             DSIN(H2(I)*A))*DEXP(-1*H3(I)*
     $                             (A+D))*DEXP(H4(I)*D)
                        ELSE
                           CONS = CON(I)
                    ENDIF
                    CALL REG4 (U,UC,J,UPLIM,LOLIM,H4(I),CONS,NUMER)
          ENDIF
          CALL DENOM (J,H1(I),H2(I),H3(I),H4(I),H5(I),CON(I),A,B,
     $                D,E,DENO)
          COE(I) = NUMER/DENO
          WRITE(2,*) COE(I)
        END DO
      END DO
C
      CLOSE (UNIT=1)
      CLOSE (UNIT=2)
      STOP
      END
C
      SUBROUTINE NUMR (U,NUM,UP,LO,A,H,SI,ANSWER)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
```

```
C  For UP and LO exists in medium 2,3 or 4 with Ey(x) having sinusoidal
C  behaviour in medium 2,3,4 and exponential behaviour in medium 1,5)
C
      COMPLEX*8 ANSWER, U(0:10000), S0, S1, S2
      REAL*8    UP, LO, A, H, SI, SEP, X
      INTEGER   NUM, I
C
      SEP = (UP-LO)/(2*NUM)
      S0 = U(0)*CMPLX(A*DCOS(H*LO-SI))+U(2*NUM)*CMPLX(A*DCOS(H*UP-SI))
      S1 = CMPLX(0.)
      S2 = CMPLX(0.)
      DO I = 1, 2*NUM-1, 2
         X = LO + I*SEP
         S1 = U(I)*CMPLX(A*DCOS(H*X-SI)) + S1
      END DO
      DO I = 2, 2*NUM-2, 2
         X = LO + I*SEP
         S2 = U(I)*CMPLX(A*DCOS(H*X-SI)) + S2
      END DO
      ANSWER = CMPLX(SEP/3)*(S0 + 4*S1 + 2*S2)
C
      RETURN
      END
C
      SUBROUTINE DENOMINS (A1,A2,A3,A4,A5,H1,H2,H3,H4,H5,A,B,D,E,
     $                     SI2,SI3,SI4,ANS)
C
C  Solve for ||Ey(x)**2|| between x = negative infinity and x =
C  positive infinity. (For Ey(x) having sinusoidal behaviour in
C  medium 2,3,4 and exponential behaviour in medium 1,5)
C
      COMPLEX*8 ANS
      REAL*8    A1, A2, A3, A4, A5, H1, H2, H3, H4, H5, A, B, D,
     $          E, DUM, DUM1, DUM2, DUM3, DUM4, SI2, SI3, SI4
C
      DUM1 = A1**2/(2*H1) + A5**2/(2*H5)
      DUM = (DSIN(2*(H2*B-SI2)) - DSIN(2*(H2*A-SI2)))/(2*H2)
      DUM2 = (A2**2)*(B - A + DUM)/2
      DUM = (DSIN(2*(H3*A-SI3)) + DSIN(2*(H3*D+SI3)))/(2*H3)
      DUM3 = (A3**2)*(A + D + DUM)/2
      DUM = (DSIN(2*(H4*D+SI4)) - DSIN(2*(H4*E+SI4)))/(2*H4)
      DUM4 = (A4**2)*(E - D - DUM)/2
      ANS = CMPLX(DUM1 + DUM2 + DUM3 + DUM4)
C
      RETURN
      END
C
      SUBROUTINE NUMR24 (U,NUM,UP,LO,A,H,SI,ANSWER)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C  For UP and LO exists in medium 2 or 4 with Ey(x) having sinusoidal
C  behaviour in medium 2,4 and exponential behaviour in medium 1,3,5.
C
      COMPLEX*8 ANSWER, U(0:10000), S0, S1, S2
      REAL*8    UP, LO, A, H, SI, SEP, X
      INTEGER   NUM, I
```

```
C
      SEP = (UP-LO)/(2*NUM)
      SO = U(0)*CMPLX(A*DCOS(H*LO-SI)) +
     $     U(2*NUM)*CMPLX(A*DCOS(H*UP-SI))
      S1 = CMPLX(0.)
      S2 = CMPLX(0.)
      DO I = 1, 2*NUM-1, 2
         X = LO + I*SEP
         S1 = U(I)*CMPLX(A*DCOS(H*X-SI)) + S1
      END DO
      DO I = 2, 2*NUM-2, 2
         X = LO + I*SEP
         S2 = U(I)*CMPLX(A*DCOS(H*X-SI)) + S2
      END DO
      ANSWER = CMPLX(SEP/3)*(SO + 4*S1 + 2*S2)
C
      RETURN
      END
C
      SUBROUTINE NUMR3 (U,NUM,J,UP,LO,A3,H3,ANSWER)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C  For UP and LO exists in medium 3 with Ey(x) having sinusoidal
C  behaviour in medium 2,4 and exponential behaviour in medium 1,3,5.
C
      COMPLEX*8 ANSWER, U(0:10000), SO, S1, S2
      REAL*8    UP, LO, A3, H3, SEP, X
      INTEGER   J, I, NUM
C
      SEP = (UP-LO)/(2*NUM)
      IF (J .EQ. 1) THEN
         SO = U(0)*CMPLX(A3*DSINH(H3*LO)) +
     $        U(2*NUM)*CMPLX(A3*DSINH(H3*UP))
         S1 = CMPLX(0.)
         S2 = CMPLX(0.)
         DO I = 1, 2*NUM-1, 2
            X = LO + I*SEP
            S1 = U(I)*CMPLX(A3*DSINH(H3*X)) + S1
         END DO
         DO I = 2, 2*NUM-2, 2
            X = LO + I*SEP
            S2 = U(I)*CMPLX(A3*DSINH(H3*X)) + S2
         END DO
      ELSE
         SO = U(0)*CMPLX(A3*DCOSH(H3*LO)) +
     $        U(2*NUM)*CMPLX(A3*DCOSH(H3*UP))
         S1 = CMPLX(0.)
         S2 = CMPLX(0.)
         DO I = 1, 2*NUM-1, 2
            X = LO + I*SEP
            S1 = U(I)*CMPLX(A3*DCOSH(H3*X)) + S1
         END DO
         DO I = 2, 2*NUM-2, 2
            X = LO + I*SEP
            S2 = U(I)*CMPLX(A3*DCOSH(H3*X)) + S2
         END DO
      ENDIF
      ANSWER = CMPLX(SEP/3)*(SO + 4*S1 + 2*S2)
C
```

```
      RETURN
      END
C
      SUBROUTINE DENOMIN (J,A1,A2,A3,A4,A5,H1,H2,H3,H4,H5,
     $                    SI2,SI4,A,B,D,E,ANS)
C
C  Solve for ||Ey(x)**2|| between x = negative infinity and x =
C  positive infinity. (For Ey(x) having sinusoidal behaviour in
C  medium 2,4 and exponential behaviour in medium 1,3,5)
C
      COMPLEX*8 ANS
      REAL*8    A1, A2, A3, A4, A5, H1, H2, H3, H4, H5, A, B, D,
     $          E, DUM, DUM1, DUM2, DUM3, DUM4, SI2, SI4
      INTEGER   J
C
      DUM1 = A1**2/(2*H1) + A5**2/(2*H5)
      DUM = (DSIN(2*(H2*B-SI2)) - DSIN(2*(H2*A-SI2)))/(2*H2)
      DUM2 = (A2**2)*(B - A + DUM)/2
      DUM = (DSIN(2*(H4*D+SI4)) - DSIN(2*(H4*E+SI4)))/(2*H4)
      DUM4 = (A4**2)*(E - D - DUM)/2
      DUM = DEXP(2*H3*A)+DEXP(2*H3*D)-
     $      DEXP(-1*2*H3*A)-DEXP(-1*2*H3*D)
      IF (J .EQ. 1) THEN
         DUM3 = (A3**2)*(DUM/(2*H3) - 2*(A+D))/4.
      ELSE
         DUM3 = (A3**2)*(DUM/(2*H3) + 2*(A+D))/4.
      ENDIF
      ANS = CMPLX(DUM1 + DUM2 + DUM3 + DUM4)
C
      RETURN
      END
C
      SUBROUTINE REG2 (U,NUM,J,UP,LO,H,CS,ANS)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C  For UP and LO exists in medium 2 with Ey(x) having sinusoidal
C  behaviour in medium 2 (or 3 or 4) and exponential behaviour in
C  medium 1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
      COMPLEX*8 ANS, U(0:10000), S0, S1, S2
      REAL*8    UP, LO, H, CS, SEP, X
      INTEGER   J, I, NUM
C
      SEP = (UP-LO)/(2*NUM)
      IF (J .EQ. 1 .OR. J .EQ. 3) THEN
         S0 = U(0)*CMPLX(CS*DEXP(-1*H*LO)) +
     $        U(2*NUM)*CMPLX(CS*DEXP(-1*H*UP))
         S1 = CMPLX(0.)
         S2 = CMPLX(0.)
         DO I = 1, 2*NUM-1, 2
            X = LO + I*SEP
            S1 = U(I)*CMPLX(CS*DEXP(-1*H*X)) + S1
         END DO
         DO I = 2, 2*NUM-2, 2
            X = LO + I*SEP
            S2 = U(I)*CMPLX(CS*DEXP(-1*H*X)) + S2
         END DO
      ELSE IF (J .EQ. 2) THEN
```

```
            S0 = U(0)*CMPLX(DCOS(H*L0)+CS*DSIN(H*L0)) +
     $           U(2*NUM)*CMPLX(DCOS(H*UP)+CS*DSIN(H*UP))
            S1 = CMPLX(0.)
            S2 = CMPLX(0.)
            DO I = 1, 2*NUM-1, 2
               X = L0 + I*SEP
               S1 = U(I)*CMPLX(DCOS(H*X)+CS*DSIN(H*X)) + S1
            END DO
            DO I = 2, 2*NUM-2, 2
               X = L0 + I*SEP
               S2 = U(I)*CMPLX(DCOS(H*X)+CS*DSIN(H*X)) + S2
            END DO
         ENDIF
         ANS = CMPLX(SEP/3)*(S0 + 4*S1 + 2*S2)
C
         RETURN
         END
C
         SUBROUTINE REG3 (U,NUM,J,UP,L0,H,CONS,ANS)
C
C  Solve the integral f(x)*Ey(x) between UP and L0 with respect to x.
C  For UP and L0 exists in medium 3 with Ey(x) having sinusoidal
C  behaviour in medium 2 (or 3 or 4) and exponential behaviour in
C  medium 1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
         COMPLEX*8 ANS, U(0:10000), S0, S1, S2
         REAL*8    UP, L0, A, H, CONS, SEP, X
         INTEGER   J, I, NUM
C
         SEP = (UP-L0)/(2*NUM)
         IF (J .EQ. 1) THEN
            S0 = U(0)*CMPLX(DCOS(H*L0)+CS*DSIN(H*L0)) +
     $           U(2*NUM)*CMPLX(DCOS(H*UP)+CS*DSIN(H*UP))
            S1 = CMPLX(0.)
            S2 = CMPLX(0.)
            DO I = 1, 2*NUM-1, 2
               X = L0 + I*SEP
               S1 = U(I)*CMPLX(DCOS(H*X)+CS*DSIN(H*X)) + S1
            END DO
            DO I = 2, 2*NUM-2, 2
               X = L0 + I*SEP
               S2 = U(I)*CMPLX(DCOS(H*X)+CS*DSIN(H*X)) + S2
            END DO
         ENDIF
         IF (J .EQ. 2) THEN
            S0 = U(0)*CMPLX(CS*DEXP(H*L0)) +
     $           U(2*NUM)*CMPLX(CS*DEXP(H*UP))
            S1 = CMPLX(0.)
            S2 = CMPLX(0.)
            DO I = 1, 2*NUM-1, 2
               X = L0 + I*SEP
               S1 = U(I)*CMPLX(CS*DEXP(H*X)) + S1
            END DO
            DO I = 2, 2*NUM-2, 2
               X = L0 + I*SEP
               S2 = U(I)*CMPLX(CS*DEXP(H*X)) + S2
            END DO
         ENDIF
```

```
      IF (J .EQ. 3) THEN
         S0 = U(0)*CMPLX(CS*DEXP(-1*H*LO)) +
     $        U(2*NUM)*CMPLX(CS*DEXP(-1*H*UP))
         S1 = CMPLX(0.)
         S2 = CMPLX(0.)
         DO I = 1, 2*NUM-1, 2
            X = LO + I*SEP
            S1 = U(I)*CMPLX(CS*DEXP(-1*H*X)) + S1
         END DO
         DO I = 2, 2*NUM-2, 2
            X = LO + I*SEP
            S2 = U(I)*CMPLX(CS*DEXP(-1*H*X)) + S2
         END DO
      ENDIF
      ANS = CMPLX(SEP/3)*(S0 + 4*S1 + 2*S2)
C
      RETURN
      END
C
      SUBROUTINE REG4 (U,NUM,J,UP,LO,H,CS,ANS)
C
C  Solve the integral f(x)*Ey(x) between UP and LO with respect to x.
C  For UP and LO exists in medium 4 with Ey(x) having sinusoidal
C  behaviour in medium 2 (or 3 or 4) and exponential behaviour in
C  medium 1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
      COMPLEX*8 ANS, U(0:10000), S0, S1, S2
      REAL*8    UP, LO, H, CS, SEP, X
      INTEGER   J, I, NUM
C
      SEP = (UP-LO)/(2*NUM)
      IF (J .EQ. 1 .OR. J .EQ. 2) THEN
         S0 = U(0)*CMPLX(CS*DEXP(H*LO)) +
     $        U(2*NUM)*CMPLX(CS*DEXP(H*UP))
         S1 = CMPLX(0.)
         S2 = CMPLX(0.)
         DO I = 1, 2*NUM-1, 2
            X = LO + I*SEP
            S1 = U(I)*CMPLX(CS*DEXP(H*X)) + S1
         END DO
         DO I = 2, 2*NUM-2, 2
            X = LO + I*SEP
            S2 = U(I)*CMPLX(CS*DEXP(H*X)) + S2
         END DO
      ELSE IF (J .EQ. 3) THEN
         S0 = U(0)*CMPLX(DCOS(H*LO)+CS*DSIN(H*LO)) +
     $        U(2*NUM)*CMPLX(DCOS(H*UP)+CS*DSIN(H*UP))
         S1 = CMPLX(0.)
         S2 = CMPLX(0.)
         DO I = 1, 2*NUM-1, 2
            X = LO + I*SEP
            S1 = U(I)*CMPLX(DCOS(H*X)+CS*DSIN(H*X)) + S1
         END DO
         DO I = 2, 2*NUM-2, 2
            X = LO + I*SEP
            S2 = U(I)*CMPLX(DCOS(H*X)+CS*DSIN(H*X)) + S2
         END DO
      ENDIF
```

```
      ANS = CMPLX(SEP/3)*(S0 + 4*S1 + 2*S2)
C
      RETURN
      END
C
      SUBROUTINE DENOM (J,H1,H2,H3,H4,H5,CS,A,B,D,E,ANS)
C
C  Solve for ||Ey(x)**2|| between x = negative infinity and x =
C  positive infinity. For Ey(x) having sinusoidal behaviour in
C  medium 2 (or 3 or 4) and exponential behaviour in medium
C  1,3,4,5 (or 1,2,4,5 or 1,2,3,5).
C
      COMPLEX*8   ANS
      REAL*8      H1, H2, H3, H4, H5, A, B, D, E,
     $            DUM, DUM1, DUM2, DUM3, DUM4, CS
      INTEGER     J
C
      IF (J .EQ. 1) THEN
         DUM = (DCOS(H3*A) + CS*DSIN(H3*A))*DEXP(-1*H2*(B-A))
         DUM1 = (DUM**2)/(2*H1)
         DUM2 = ((DUM/DEXP(-1*H2*(B-A)))**2)*
     $          (1.-DEXP(-1*2*H2*(B-A)))/(2*H2)
         DUM = (DCOS(H3*D) - CS*DSIN(H3*D))*DEXP(H4*(D-E))
         DUM1 = DUM1 + (DUM**2)/(2*H5)
         DUM4 = ((DUM/DEXP(H4*(D-E)))**2)*
     $          (1.-DEXP(2*H4*(D-E)))/(2*H4)
         DUM3 = (1+CS**2)*(A+D)/2. + (1-CS**2)*(DSIN(2*H3*A)+
     $          DSIN(2*H3*D))/(4*H3) + CS*(DCOS(2*H3*D)-
     $          DCOS(2*H3*A))/(2*H3)
      ENDIF
      IF (J .EQ. 2) THEN
         DUM = DCOS(H2*B) + CS*DSIN(H2*B)
         DUM1 = (DUM**2)/(2*H1)
         DUM = (DCOS(H2*A)+CS*DSIN(H2*A))*DEXP(-1*H3*(A+D))*
     $          DEXP(H4*(D-E))
         DUM1 = DUM1 + (DUM**2)/(2*H5)
         DUM2 = (1+CS**2)*(B-A)/2. +
     $          (1-CS**2)*(DSIN(2*H2*B)-DSIN(2*H2*A))/(4*H2) -
     $          CS*(DCOS(2*H2*B)-DCOS(2*H2*A))/(2*H2)
         DUM = DCOS(H2*A) + CS*DSIN(H2*A)
         DUM3 = (DUM**2)*(1.- DEXP(-2*H3*(A+D)))/(2*H3)
         DUM4 = ((DUM*DEXP(-1*H3*(A+D)))**2)*
     $          (1.- DEXP(2*H4*(D-E)))/(2*H4)
      ENDIF
      IF (J .EQ. 3) THEN
         DUM = DCOS(H4*E) - CS*DSIN(H4*E)
         DUM1 = (DUM**2)/(2*H5)
         DUM = (DCOS(H4*D)-CS*DSIN(H4*D))*DEXP(-1*H3*(A+D))*
     $          DEXP(-1*H2*(B-A))
         DUM1 = DUM1 + (DUM**2)/(2*H1)
         DUM = DCOS(H4*D) - CS*DSIN(H4*D)
         DUM3 = (DUM**2)*(1.- DEXP(-2*H3*(A+D)))/(2*H3)
         DUM2 = ((DUM*DEXP(-1*H3*(A+D)))**2)*
     $          (1.- DEXP(-2*H2*(B-A)))/(2*H2)
         DUM4 = (1+CS**2)*(E-D)/2. +
     $          (1-CS**2)*(DSIN(2*H4*E)-DSIN(2*H4*D))/(4*H4) -
     $          CS*(DCOS(2*H4*D)-DCOS(2*H4*E))/(2*H4)
      ENDIF
```

```
      ANS = CMPLX(DUM1 + DUM2 + DUM3 + DUM4)
C
      RETURN
      END
C
      SUBROUTINE UFIELD (XO,F,K,Z,UPP,LOO,AREA)
C
C  Compute the electric field of a diffraction pattern at a
C  distance z behind the lens. (at X = XO)
C
      COMPLEX*8  SO, S1, S2, C1, C2, FO, F2N, AREA
      REAL*8     X, XO, F, K, LAMBDAO, Z, UPP, LOO, H,
     $           UP, LO, DUMO, PI
      INTEGER    N, I
C
C  N : Half of the number of partitions used for integration
C  H : Distance between partitions
C  UPP : Upper limit for the integration
C  LOO : Lower limit for the integration
C
      PI = 3.141592654
      N = 500
      DUMO = 0.0
      LAMBDAO = 2*PI/K
      H = (UPP-LOO)/(2*N)
      UP = (UPP-LOO)/2
      LO = -1*UP
      C1 = CEXP(CMPLX(DUMO,K*LO**2*(1/Z-1/F)/2))
      C2 = CEXP(CMPLX(DUMO,-1*K*XO*LO/Z))
      FO = C1*C2
      C1 = CEXP(CMPLX(DUMO,K*UP**2*(1/Z-1/F)/2))
      C2 = CEXP(CMPLX(DUMO,-1*K*XO*UP/Z))
      F2N = C1*C2
      SO = FO + F2N
      S1 = CMPLX(0.)
      S2 = CMPLX(0.)
      DO I = 1, 2*N-1, 2
         X = LO + I*H
         C1 = CEXP(CMPLX(DUMO,K*X**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUMO,-1*K*XO*X/Z))
         S1 = C1*C2 + S1
      END DO
      DO I = 2, 2*N-2, 2
         X = LO + I*H
         C1 = CEXP(CMPLX(DUMO,K*X**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUMO,-1*K*XO*X/Z))
         S2 = C1*C2 + S2
      END DO
      C1 = CEXP(CMPLX(DUMO,K*Z))*CEXP(CMPLX(DUMO,K*XO**2/(2*Z)))
      C2 = CMPLX(DUMO,LAMBDAO*Z)
      AREA = (C1/C2)*((H/3)*(SO + 4*S1 + 2*S2))
C
      RETURN
      END
```

```
C  PROGRAM : To find the specified number of radiation modes for the
C           case of the parallel slab dielectric waveguides being
C           excited by an input wave at z=0.
C
      CHARACTER*3 SIGNB, SIGNA, SIGN2, SIGN4, FLAG1, FLAG2
      CHARACTER*5 REPEAT
      REAL*8  A, B, C, D, E, F, N1, N2, N3, N4, N5, K0, LAMBDA0, PI,
     $        TH, K1, K2, K3, K4, K5, CONST, HA, HB, LRANGE(1500),
     $        RRANGE(1500), P1, P2, P3, P4, P5, FP1, FP2, FP4, FP5,
     $        H1(1500), H2(1500), H3(1500), H4(1500), H5(1500),
     $        A1(1500), A2(1500), A3(1500), A4(1500), A5(1500),
     $        SI1(1500), SI2(1500), SI3(1500), SI4(1500), SI5(1500),
     $        LCONST, RCONST, DUM, STEP, X, OLD1, OLD5, NEW1, NEW5
      INTEGER I, J, NUM, M, RNUM, SOL, COUNT, COUNT2
C
C  B-A : Thickness for slab define as medium 2 (B>A)
C  E-D : Thickness for slab define as medium 4 (E>D)
C  A+D : Distance between the two slabs
C  C,F : Distance of the reflectors from the z axis
C        (+ve and -ve X direction respectively)
C  N1-N5 : Refractive indices for the five media
C  LAMBDA0 : Free-space wavelength of light
C  NUM : Nnumber of radiation modes to be used
C  COPR : File contains propagation constants for each radiation
C         mode and arbitrary constants for the electric field
C
      OPEN (UNIT=1,FILE='COPR.DAT',STATUS='NEW')
C
C  Obtain information from the user
C
      WRITE(*,*) 'N1 = ?'
      READ(*,*) N1
      WRITE(*,*) 'N2 = ?'
      READ(*,*) N2
      WRITE(*,*) 'N3 = ?'
      READ(*,*) N3
      WRITE(*,*) 'N4 = ?'
      READ(*,*) N4
      WRITE(*,*) 'N5 = ?'
      READ(*,*) N5
      WRITE(*,*) 'A = '
      READ(*,*) A
      WRITE(*,*) 'B = '
      READ(*,*) B
      WRITE(*,*) 'C = '
      READ(*,*) C
      WRITE(*,*) 'D = '
      READ(*,*) D
      WRITE(*,*) 'E = '
      READ(*,*) E
      WRITE(*,*) 'F = '
      READ(*,*) F
      WRITE(*,*) 'Number of radiation modes to be used ?'
      READ(*,*) NUM
C
C  Set up constants for calculations
C
      PI = 3.141592654
      LAMBDA0 = 0.5E-6
```

```
      K0 = 2*PI/LAMBDA0
      K1 = (K0**2)*(N1**2)
      K2 = (K0**2)*(N2**2)
      K3 = (K0**2)*(N3**2)
      K4 = (K0**2)*(N4**2)
      K5 = (K0**2)*(N5**2)
      WRITE(1,*) N1, N2, N3, N4, N5, K0
      WRITE(1,*) A, B, C, D, E, F
C
C  Decide which refractive index (N1,N3,N5) is larger
C
      IF (N3 .GT. N1) THEN
         IF (N5 .GE. N1) THEN
            SOL = 1
         ELSE
            SOL = 5
         ENDIF
      ELSE IF (N5 .LT. N3) THEN
              SOL = 5
           ELSE
              SOL = 3
      ENDIF
C
C  Compute the propagation constants for each radiation mode
C
      WRITE(1,*) NUM
      LCONST = -1*30000.
      CONST = 30000.
      STEP = CONST/500
      M = 1
      DO WHILE (M .LE. NUM)
        LCONST = LCONST + 30000.
        J = 1
        DO I = 0, 499
         IF (I .EQ. 0) THEN
            TH = 1E-10 + LCONST
         ELSE
            TH = I*STEP + LCONST
         ENDIF
         CALL EQNH (SOL,TH,K1,K2,K3,K4,K5,A,B,C,D,E,F,DIFF)
         IF (I .EQ. 0) THEN
            HB = TH
            IF (DIFF .GE. 0) THEN
               SIGNB = 'POS'
            ELSE
               SIGNB = 'NEG'
            ENDIF
         ENDIF
         HA = TH
         IF (DIFF .GE. 0) THEN
            SIGNA = 'POS'
         ELSE
            SIGNA = 'NEG'
         ENDIF
         IF (SIGNA .NE. SIGNB .AND. SIGNB .EQ. 'NEG') THEN
            LRANGE(J) = HB
            RRANGE(J) = HA
            J = J + 1
         ENDIF
```

```
            HB = HA
            SIGNB = SIGNA
          END DO
          RNUM = J - 1
          DO I = 1, RNUM
           P1 = LRANGE(I)
           P5 = RRANGE(I)
           REPEAT = 'TRUE '
           COUNT = 0
           COUNT2 = 0
           DO WHILE (REPEAT .EQ. 'TRUE ')
              P3 = (P5-P1)/2 + P1
              P2 = (P3-P1)/2 + P1
              P4 = (P5-P3)/2 + P3
              CALL EQNH (SOL,P2,K1,K2,K3,K4,K5,A,B,C,D,E,F,FP2)
              IF (FP2 .GT. 0) THEN
                 SIGN2 = 'POS'
              ELSE
                 SIGN2 = 'NEG'
              ENDIF
              CALL EQNH (SOL,P4,K1,K2,K3,K4,K5,A,B,C,D,E,F,FP4)
              IF (FP4 .GT. 0) THEN
                 SIGN4 = 'POS'
              ELSE
                 SIGN4 = 'NEG'
              ENDIF
              IF (FP1 .EQ. FP2 .AND. FP4 .EQ. FP5 .AND.
     $             FP1 .LT. 1E-2) THEN
                  COUNT = COUNT + 1
              ENDIF
              IF (SIGN2 .EQ. SIGN4) THEN
                 IF (FP2 .GT. FP4) THEN
                    IF (SIGN2 .EQ. 'POS') THEN
                       P1 = P4
                    ELSE
                       P5 = P2
                    ENDIF
                 ELSE
                    IF (SIGN2 .EQ. 'POS') THEN
                       P5 = P2
                    ELSE
                       P1 = P4
                    ENDIF
                 ENDIF
              ELSE
                 P1 = P2
                 P5 = P4
              ENDIF
C
              CALL EQNH (SOL,P1,K1,K2,K3,K4,K5,A,B,C,D,E,F,FP1)
              CALL EQNH (SOL,P5,K1,K2,K3,K4,K5,A,B,C,D,E,F,FP5)
              NEW1 = FP1
              NEW5 = FP5
              IF (OLD1 .EQ. NEW1 .AND. OLD5 .EQ. NEW5 .AND.
     $             OLD1 .LT. 1E-2) THEN
                     COUNT 2 = COUNT2 + 1
              ENDIF
              OLD1 = NEW1
              OLD5 = NEW5
```

```
            IF (ABS(FP1) .LT. 1E-5 .OR. ABS(FP5) .LT. 1E-5 .OR.
     $           COUNT .GT. 10 .OR. COUNT2 .GT. 10) THEN
               IF (ABS(FP1) .LT. ABS(FP5)) THEN
                  IF (SOL .EQ. 1) THEN
                     H1(M) = P1
                  ELSE IF (SOL .EQ. 3) THEN
                         H3(M) = P1
                       ELSE
                         H5(M) = P1
                  ENDIF
                  DUM = FP1
               ELSE
                  IF (SOL .EQ. 1) THEN
                     H1(M) = P5
                  ELSE IF (SOL .EQ. 3) THEN
                         H3(M) = P5
                       ELSE
                         H5(M) = P5
                  ENDIF
                  DUM = FP5
               ENDIF
               REPEAT = 'FALSE'
            ELSE
               REPEAT = 'TRUE '
            ENDIF
         END DO
         IF (SOL .EQ. 1) THEN
            DUM = H1(M)**2
            H2(M) = SQRT(K2 - K1 + DUM)
            H3(M) = SQRT(K3 - K1 + DUM)
            H4(M) = SQRT(K4 - K1 + DUM)
            H5(M) = SQRT(K5 - K1 + DUM)
         ELSE IF (SOL .EQ. 3) THEN
                DUM = H3(M)**2
                H1(M) = SQRT(K1 - K3 + DUM)
                H2(M) = SQRT(K2 - K3 + DUM)
                H4(M) = SQRT(K4 - K3 + DUM)
                H5(M) = SQRT(K5 - K3 + DUM)
              ELSE
                DUM = H5(M)**2
                H1(M) = SQRT(K1 - K5 + DUM)
                H2(M) = SQRT(K2 - K5 + DUM)
                H3(M) = SQRT(K3 - K5 + DUM)
                H4(M) = SQRT(K4 - K5 + DUM)
         ENDIF
         WRITE(1,*) H1(M), H2(M), H3(M), H4(M), H5(M)
         M = M + 1
       END DO
     END DO
C
      CLOSE (UNIT=1)
      CLOSE (UNIT=2)
      STOP
      END
C
      SUBROUTINE EQNH (SOL,H,K1,K2,K3,K4,K5,A,B,C,D,E,F,DIFF)
C
C  Solve the equation for H1 with H1,H2,H3,H4,H5 given
C
```

```
      CHARACTER*4 MODE
      REAL*8  H1, H2, H3, H4, H5, A, B, C, D, E, F, DIFF, LEFT,
     $        RIGHT, K1, K2, K3, K4, K5, H, SI1, SI5
      INTEGER SOL
C
      IF (SOL .EQ. 1) THEN
         H1 = H
         H2 = SQRT(K2 - K1 + H**2)
         H3 = SQRT(K3 - K1 + H**2)
         H4 = SQRT(K4 - K1 + H**2)
         H5 = SQRT(K5 - K1 + H**2)
      ELSE IF (SOL .EQ. 3) THEN
            H3 = H
            H1 = SQRT(K1 - K3 + H**2)
            H2 = SQRT(K2 - K3 + H**2)
            H4 = SQRT(K4 - K3 + H**2)
            H5 = SQRT(K5 - K3 + H**2)
        ELSE
            H5 = H
            H1 = SQRT(K1 - K5 + H**2)
            H2 = SQRT(K2 - K5 + H**2)
            H3 = SQRT(K3 - K5 + H**2)
            H4 = SQRT(K4 - K5 + H**2)
      ENDIF
      SI1 = H1*(C-B)
      SI5 = H5*(F-E)
      LEFT = H3*DTAN(H3*(D+A)-DATAN((H2/H3)*DTAN(H2*(A-B)+
     $       DATAN(H1/(H2*DTAN(SI1))))))
      RIGHT = H4*DTAN(H4*(D-E)+DATAN(H5/(H4*DTAN(SI5))))
      DIFF = LEFT - RIGHT
C
      RETURN
      END
```

```
C  PROGRAM : To find the weighting factors for radiation modes used
C           of the parallel waveguides being excited by plane wave.
C
      COMPLEX*8 COE(1500), NUMER, DENO, DUM1, DUM2, DUM3
      REAL*8 N1, N2, N3, N4, N5, A, B, C, D, E, F, KO, PI, LOLIM,
     $       H1(1500), H2(1500), H3(1500), H4(1500), H5(1500), UPLIM,
     $       A1(1500), A2(1500), A3(1500), A4(1500), A5(1500),
     $       SI1(1500), SI2(1500), SI3(1500), SI4(1500), SI5(1500)
      INTEGER I, J, NUM
C
      OPEN (UNIT=1,FILE='COPR.DAT',STATUS='OLD')
      OPEN (UNIT=2,FILE='COPRCOE.DAT',STATUS='NEW')
C
C  COE : Weighting factor for each radiation mode
C  UPLIM,LOLIM : Upper and lower limit where f(x) exists
C  COPRCOE : File contains weighting factor for each radiation mode
C
      WRITE(*,*) 'What is the upper limit for f(x) ? (in meters)'
      READ(*,*) UPLIM
      WRITE(*,*) 'What is the lower limit for f(x) ? (in meters)'
      READ(*,*) LOLIM
      READ(1,*) N1, N2, N3, N4, N5, KO
      READ(1,*) A, B, C, D, E, F
      PI = 3.141592654
C
C  Compute the arbitrary constants for the electric field
C
       READ(1,*) NUM
       DO I = 1, NUM
         READ(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
         SI1(I) = H1(I)*(C-B)
         SI5(I) = H5(I)*(F-E)
         SI2(I) = H2(I)*B-DATAN((H1(I)/H2(I))/DTAN(SI1(I)))
         SI4(I) = DATAN((H5(I)/H4(I))/DTAN(SI5(I)))-H4(I)*E
         SI3(I) = H3(I)*A-DATAN((H2(I)/H3(I))*DTAN(H2(I)*A-SI2(I)))
         A1(I) = 1.0
         A2(I) = -1*A1(I)*DSIN(SI1(I))/DCOS(H2(I)*B-SI2(I))
         A3(I) = A2(I)*DCOS(H2(I)*A-SI2(I))/DCOS(H3(I)*A-SI3(I))
         A4(I) = A3(I)*DCOS(H3(I)*D+SI3(I))/DCOS(H4(I)*D+SI4(I))
         A5(I) = A4(I)*DCOS(H4(I)*E+SI4(I))/DSIN(SI5(I))
       END DO
C
C  Compute the weighting factor for each radiation mode used
C
       DO I = 1, NUM
         IF (UPLIM .GT. A) THEN
           IF (LOLIM .GE. A) THEN
             CALL NUMR (UPLIM,LOLIM,A2(I),H2(I),SI2(I),NUMER)
           ELSE IF (LOLIM .GE. -1*D) THEN
                  CALL NUMR (UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                  CALL NUMR (A,LOLIM,A3(I),H3(I),SI3(I),DUM2)
                  NUMER = DUM1 + DUM2
                ELSE IF (LOLIM .GE. -1*E) THEN
                       CALL NUMR (UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                       CALL NUMR (A,-1*D,A3(I),H3(I),SI3(I),DUM2)
                       CALL NUMR (-1*D,LOLIM,A4(I),H4(I),SI4(I),DUM3)
                       NUMER = DUM1 + DUM2 + DUM3
           ENDIF
         ELSE IF (UPLIM .GT. -1*D) THEN
```

```
                  IF (LOLIM .GE. -1*D) THEN
                    CALL NUMR (UPLIM,LOLIM,A3(I),H3(I),SI3(I),NUMER)
                  ELSE IF (LOLIM .GE. -1*E) THEN
                        CALL NUMR (UPLIM,-1*D,A3(I),H3(I),SI3(I),DUM1)
                        CALL NUMR (-1*D,LOLIM,A4(I),H4(I),SI4(I),DUM2)
                        NUMER = DUM1 + DUM2
                  ENDIF
                ELSE IF (LOLIM .GE. -1*E) THEN
                     CALL NUMR (UPLIM,LOLIM,A4(I),H4(I),SI4(I),NUMER)
           ENDIF
           CALL DENOMIN (A1(I),A2(I),A3(I),A4(I),A5(I),H1(I),H2(I),
     $                 H3(I),H4(I),H5(I),A,B,C,D,E,F,SI1(I),SI2(I),
     $                 SI3(I),SI4(I),SI5(I),DENO)
           COE(I) = NUMER/DENO
           WRITE(2,*) COE(I)
        END DO
C
        CLOSE (UNIT=1)
        CLOSE (UNIT=2)
        STOP
        END
C
        SUBROUTINE NUMR (UP,LO,A,H,SI,ANSWER)
C
C  Solve the integral f(x)*gm(x) between UP and LO with respect to x.
C  (For both even and odd radiation modes in medium 2, 3 or 4)
C
        COMPLEX*8 ANSWER
        REAL*8    UP, LO, A, H, SI
C
        ANSWER = A*(DSIN(H*UP-SI) - DSIN(H*LO-SI))/H
C
        RETURN
        END
C
        SUBROUTINE DENOMIN (A1,A2,A3,A4,A5,H1,H2,H3,H4,H5,A,B,C,D,E,F,
     $                      SI1,SI2,SI3,SI4,SI5,ANS)
C
C  Solve ||gm(x)||**2 for each radiation mode
C
        COMPLEX*8 ANS
        REAL*8    A1, A2, A3, A4, A5, H1, H2, H3, H4, H5, A, B, C,
     $            D, E, F, SI1, SI2, SI3, SI4, SI5, DUM, DUM1,
     $            DUM2, DUM3, DUM4
C
        DUM1 = (A1**2)*(C-B-(DSIN(2*SI1))/(2*H1))/2. +
     $         (A5**2)*(F-E-(DSIN(2*SI5))/(2*H5))/2.
        DUM = (DSIN(2*(H2*B-SI2)) - DSIN(2*(H2*A-SI2)))/(2*H2)
        DUM2 = (A2**2)*(B - A + DUM)/2
        DUM = (DSIN(2*(H3*A-SI3)) + DSIN(2*(H3*D+SI3)))/(2*H3)
        DUM3 = (A3**2)*(A + D + DUM)/2
        DUM = (DSIN(2*(H4*D+SI4)) - DSIN(2*(H4*E+SI4)))/(2*H4)
        DUM4 = (A4**2)*(E - D - DUM)/2
        ANS = DUM1 + DUM2 + DUM3 + DUM4
C
        RETURN
        END
```

```
C  PROGRAM : To find the weighting factor for each radiation mode used
C           of the parallel waveguides being excited by focused wave.
C
      COMPLEX*8 COE(1500), NUMER, DENO, DUM1, DUM2, DUM3, U(0:10000)
      REAL*8 N1, N2, N3, N4, N5, KO, PI, UPLIM, LOLIM, UPL ,LOL,
     $       H1(1500), H2(1500), H3(1500), H4(1500), H5(1500),
     $       A1(1500), A2(1500), A3(1500), A4(1500), A5(1500),
     $       SI1(1500), SI2(1500), SI3(1500), SI4(1500), SI5(1500),
     $       A, B, C, D, E, F, FOCAL, ZPRIM, SEP, X, UU, LL
      INTEGER I, J, NUM, CNT
C
      OPEN (UNIT=1,FILE='COPR.DAT',STATUS='OLD')
      OPEN (UNIT=2,FILE='COPRCOE.DAT',STATUS='NEW')
C
C COE : Weighting factor for each radiation mode used
C FOCAL : Focal length of the lens
C ZPRIM : Distance between the lens and the slab waveguides
C UPLIM,LOLIM : Upper and lower limit where f(x) exists
C COPRCOE : File contains the weighting factor for each radiation mode
C
      WRITE(*,*) 'What is the upper limit for f(x) ? (in meters)'
      READ(*,*) UPLIM
      WRITE(*,*) 'What is the lower limit for f(x) ? (in meters)'
      READ(*,*) LOLIM
      WRITE(*,*) 'What is the focal length of the lens ? (in meters)'
      READ(*,*) FOCAL
      WRITE(*,*) 'What is the location of the lens ?'
      WRITE(*,*) '(between X = UPL and X = LOL)'
      WRITE(*,*) 'UPL = ?'
      READ(*,*) UPL
      WRITE(*,*) 'LOL = ?'
      READ(*,*) LOL
      WRITE(*,*) 'What is the distance between the lens and the',
     $           ' parallel slab waveguides ? (in meters)'
      READ(*,*) ZPRIM
      READ(1,*) N1, N2, N3, N4, N5, KO
      READ(1,*) A, B, C, D, E, F
C
C  Compute the electric field for the input wave at z=0
C
      PI = 3.141592654
      CNT = 25
      SEP = (UPLIM-LOLIM)/(2*CNT)
      UU = (UPLIM-LOLIM)/2
      LL = -1*UU
      CALL UFIELD (LL,FOCAL,KO,ZPRIM,UPL,LOL,U(0))
      CALL UFIELD (UU,FOCAL,KO,ZPRIM,UPL,LOL,U(2*CNT))
      DO J = 1, 2*CNT-1
         X = LL + J*SEP
        CALL UFIELD (X,FOCAL,KO,ZPRIM,UPL,LOL,U(J))
      END DO
C
C  Compute the arbitrary constants for the electric field
C
       READ(1,*) NUM
       DO I = 1, NUM
         READ(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
         SI1(I) = H1(I)*(C-B)
```

```
            SI5(I) = H5(I)*(F-E)
            SI2(I) = H2(I)*B-DATAN((H1(I)/H2(I))/DTAN(SI1(I)))
            SI4(I) = DATAN((H5(I)/H4(I))/DTAN(SI5(I)))-H4(I)*E
            SI3(I) = H3(I)*A-DATAN((H2(I)/H3(I))*DTAN(H2(I)*A-SI2(I)))
            A1(I) = 1.0
            A2(I) = -1*A1(I)*DSIN(SI1(I))/DCOS(H2(I)*B-SI2(I))
            A3(I) = A2(I)*DCOS(H2(I)*A-SI2(I))/DCOS(H3(I)*A-SI3(I))
            A4(I) = A3(I)*DCOS(H3(I)*D+SI3(I))/DCOS(H4(I)*D+SI4(I))
            A5(I) = A4(I)*DCOS(H4(I)*E+SI4(I))/DSIN(SI5(I))
          END DO
C
C  Compute the propagation constant for each radiation mode used
C
          DO I = 1, NUM
            IF (UPLIM .GT. A) THEN
              IF (LOLIM .GE. A) THEN
                CALL NUMR (U,CNT,UPLIM,LOLIM,A2(I),H2(I),SI2(I),NUMER)
              ELSE IF (LOLIM .GE. -1*D) THEN
                     CALL NUMR (U,CNT,UPLIM,A,A2(I),H2(I),SI2(I),DUM1)
                     CALL NUMR (U,CNT,A,LOLIM,A3(I),H3(I),SI3(I),DUM2)
                     NUMER = DUM1 + DUM2
                   ELSE IF (LOLIM .GE. -1*E) THEN
                          CALL NUMR (U,CNT,UPLIM,A,A2(I),H2(I),
     $                                SI2(I),DUM1)
                          CALL NUMR (U,CNT,A,-1*D,A3(I),H3(I),
     $                                SI3(I),DUM2)
                          CALL NUMR (U,CNT,-1*D,LOLIM,A4(I),H4(I),
     $                                SI4(I),DUM3)
                          NUMER = DM1 + DUM2 + DUM3
              ENDIF
             ELSE IF (UPLIM .GT. -1*D) THEN
                   IF (LOLIM .GE. -1*D) THEN
                    CALL NUMR (U,CNT,UPLIM,LOLIM,A3(I),H3(I),
     $                         SI3(I),NUMER)
                   ELSE IF (LOLIM .GE. -1*E) THEN
                          CALL NUMR (U,CNT,UPLIM,-1*D,A3(I),H3(I),SI3(I),
     $                                DUM1)
                          CALL NUMR (U,CNT,-1*D,LOLIM,A4(I),H4(I),SI4(I),
     $                                DUM2)
                          NUMER = DUM1 + DUM2
                  ENDIF
                ELSE IF (LOLIM .GE. -1*E) THEN
                       CALL NUMR (U,CNT,UPLIM,LOLIM,A4(I),H4(I),SI4(I),
     $                            NUMER)
            ENDIF
            CALL DENOMIN (A1(I),A2(I),A3(I),A4(I),A5(I),H1(I),H2(I),
     $                    H3(I),H4(I),H5(I),A,B,C,D,E,F,SI1(I),SI2(I),
     $                    SI3(I),SI4(I),SI5(I),DENO)
            COE(I) = NUMER/DENO
            WRITE(2,*) COE(I)
          END DO
C
          CLOSE (UNIT=1)
          CLOSE (UNIT=2)
          STOP
          END
C
          SUBROUTINE NUMR (U,NUM,UP,LO,A,H,SI,ANSWER)
C
```

```
C  Solve the integral f(x)*gm(x) between UP and LO with respect to x.
C  (For medium 2, 3, OR, 4)
C
      COMPLEX*8 ANSWER, U(0:10000), S0, S1, S2
      REAL*8    UP, LO, A, H, SI, SEP, X
      INTEGER   NUM, I
C
      SEP = (UP-LO)/(2*NUM)
      S0 = U(0)*CMPLX(A*DCOS(H*LO-SI)) +
     $     U(2*NUM)*CMPLX(A*DCOS(H*UP-SI))
      S1 = CMPLX(0.)
      S2 = CMPLX(0.)
      DO I = 1, 2*NUM-1, 2
         X = LO + I*SEP
         S1 = U(I)*CMPLX(A*DCOS(H*X-SI)) + S1
      END DO
      DO I = 2, 2*NUM-2, 2
         X = LO + I*SEP
         S2 = U(I)*CMPLX(A*DCOS(H*X-SI)) + S2
      END DO
      ANSWER = CMPLX(SEP/3)*(S0 + 4*S1 + 2*S2)
C
      RETURN
      END
C
      SUBROUTINE DENOMIN (A1,A2,A3,A4,A5,H1,H2,H3,H4,H5,A,B,C,D,E,F,
     $                    SI1,SI2,SI3,SI4,SI5,ANS)
C
C  Solve ||gm(x)||**2 for the each radiation mode
C
      COMPLEX*8 ANS
      REAL*8    A1, A2, A3, A4, A5, H1, H2, H3, H4, H5, A, B, C,
     $          D, E, F, SI1, SI2, SI3, SI4, SI5, DUM, DUM1,
     $          DUM2, DUM3, DUM4
      DUM1 = (A1**2)*(C-B-(DSIN(2*SI1))/(2*H1))/2. +
     $       (A5**2)*(F-E-(DSIN(2*SI5))/(2*H5))/2.
      DUM = (DSIN(2*(H2*B-SI2)) - DSIN(2*(H2*A-SI2)))/(2*H2)
      DUM2 = (A2**2)*(B - A + DUM)/2
      DUM = (DSIN(2*(H3*A-SI3)) + DSIN(2*(H3*D+SI3)))/(2*H3)
      DUM3 = (A3**2)*(A + D + DUM)/2
      DUM = (DSIN(2*(H4*D+SI4)) - DSIN(2*(H4*E+SI4)))/(2*H4)
      DUM4 = (A4**2)*(E - D - DUM)/2
      ANS = DUM1 + DUM2 + DUM3 + DUM4
C
      RETURN
      END
C
      SUBROUTINE UFIELD (X0,F,K,Z,UPP,LOO,AREA)
C
C  To compute the electric field of a diffraction pattern at a
C  distance z behind the lens. (At X = X0)
C
      COMPLEX*8 S0, S1, S2, C1, C2, F0, F2N, AREA
      REAL*8    X, X0, F, K, LAMBDA0, Z, UPP, LOO, H,
     $          UP, LO, DUM0, SHIFT
      INTEGER   N, I
C
C  N : Half of the number of partitions used for integration
C  H : Distance between partitions
```

```
C  UP : Upper limit for the integration
C  LO : Lower limit for the integration
C
      PI = 3.141592654
      N = 500
      DUMO = 0.0
      LAMBDAO = 2*PI/K
      H = (UPP-LOO)/(2*N)
      UP = (UPP-LOO)/2
      LO = -1*UP
      C1 = CEXP(CMPLX(DUMO,K*LO**2*(1/Z-1/F)/2))
      C2 = CEXP(CMPLX(DUMO,-1*K*XO*LO/Z))
      FO = C1*C2
      C1 = CEXP(CMPLX(DUMO,K*UP**2*(1/Z-1/F)/2))
      C2 = CEXP(CMPLX(DUMO,-1*K*XO*UP/Z))
      F2N = C1*C2
      SO = FO + F2N
      S1 = CMPLX(O.)
      S2 = CMPLX(O.)
      DO I = 1, 2*N-1, 2
         X = LO + I*H
         C1 = CEXP(CMPLX(DUMO,K*X**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUMO,-1*K*XO*X/Z))
         S1 = C1*C2 + S1
      END DO
      DO I = 2, 2*N-2, 2
         X = LO + I*H
         C1 = CEXP(CMPLX(DUMO,K*X**2*(1/Z-1/F)/2))
         C2 = CEXP(CMPLX(DUMO,-1*K*XO*X/Z))
         S2 = C1*C2 + S2
      END DO
      C1 = CEXP(CMPLX(DUMO,K*Z))*CEXP(CMPLX(DUMO,K*XO**2/(2*Z)))
      C2 = CMPLX(DUMO,LAMBDAO*Z)
      AREA = (C1/C2)*((H/3)*(SO + 4*S1 + 2*S2))
C
      RETURN
      END
```

```
C  PROGRAM : To find the electric field for all the guided modes
C           used for the parallel slab waveguides at z.
C
      COMPLEX*8 FIELD(800), EFIELD(800), OFIELD(800),
     $          ECOE(500), OCOE(500), COE(500), SFIELD(800)
      REAL*8  H1(500), H2(500), H3(500), H4(500), H5(500), A1(500),
     $        A2(500), A3(500), A4(500), A5(500), SI2(500), SI3(500),
     $        SI4(500), A, B, C, D, E, F, EH1(500), EH2(500),
     $        EH3(500), EH4(500), EH5(500), EA1(500), EA2(500),
     $        EA3(500), EA4(500), EA5(500), OH1(500), OH2(500),
     $        OH3(500), OH4(500), OH5(500), OA1(500), OA2(500),
     $        OA3(500), OA4(500), OA5(500), ESI2(500), ESI4(500),
     $        OSI2(500), OSI4(500), KO, PI, N1, N2, N3, N4, N5, DUM,
     $        X, LEFT, RIGHT, Z, DUMO, BETA, CON(500)
      INTEGER I, J, ENUM, ONUM, NUM, CODE
C
      OPEN (UNIT=1,FILE='COPG.DAT',STATUS='OLD')
      OPEN (UNIT=2,FILE='COPGCOE.DAT',STATUS='OLD')
      OPEN (UNIT=3,FILE='COPG.PLOT',STATUS='NEW')
C
C  Obtain information from the user
C
      WRITE(*,*) 'What is  z ? (in meters)'
      READ(*,*) Z
      READ(1,*) N1, N2, N3, N4, N5, KO
      READ(1,*) A, B, C, D, E, F
      READ(1,*) NUM
      WRITE(3,*) 500
      TYPE *,'Range of X to be plotted (between X=LEFT & X=RIGHT)?'
      WRITE(*,*) 'LEFT = '
      READ(*,*) LEFT
      WRITE(*,*) 'RIGHT = '
      READ(*,*) RIGHT
      PI = 3.141592654
      DUMO = 0.0
C
C  Compute the electric field with sinusoidal behaviour in
C  medium 2,3,4 and exponential behaviour in medium 1,5.
C
      DO I = 1, NUM
         READ(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
         READ(1,*) A1(I), A2(I), A3(I), A4(I), A5(I)
         READ(1,*) SI2(I), SI3(I), SI4(I)
         READ(2,*) COE(I)
      END DO
      J = 1
      DO X = LEFT, RIGHT, (RIGHT-LEFT)/500
        FIELD(J)  = CMPLX(0.)
        DO I = 1, NUM
         IF (X .GT. B) THEN
           DUM = A1(I)*DEXP(-1*H1(I)*(X-B))
         ELSE IF (X .GT. A .AND. X .LE. B) THEN
                DUM = A2(I)*DCOS(H2(I)*X-SI2(I))
              ELSE IF (X .GE. -1*D .AND. X .LE. A) THEN
                     DUM = A3(I)*DCOS(H3(I)*X-SI3(I))
                   ELSE IF (X .GE. -1*E .AND. X .LT. -1*D) THEN
                          DUM = A4(I)*DCOS(H4(I)*X-SI4(I))
                        ELSE
                          DUM = A5(I)*DEXP(H5(I)*(X+E))
```

```
            ENDIF
            BETA = SQRT(KO**2*N1**2 + H1(I)**2)
            FIELD(J) = COE(I)*CMPLX(DUM)*CEXP(CMPLX(DUMO,-1*BETA*Z)) +
     $                 FIELD(J)
          END DO
          J = J + 1
        END DO
C
C  Compute the electric field with sinusoidal behaviour in
C  medium 2,4 and exponential behaviour in medium 1,3,5.
C
        READ(1,*) ENUM
        DO I = 1, ENUM
           READ(1,*) EH1(I), EH2(I), EH3(I), EH4(I), EH5(I)
           READ(1,*) EA1(I), EA2(I), EA3(I), EA4(I), EA5(I)
           READ(1,*) ESI2(I), ESI4(I)
           READ(2,*) ECOE(I)
        END DO
        READ(1,*) ONUM
        DO I = 1, ONUM
           READ(1,*) OH1(I), OH2(I), OH3(I), OH4(I), OH5(I)
           READ(1,*) OA1(I), OA2(I), OA3(I), OA4(I), OA5(I)
           READ(1,*) OSI2(I), OSI4(I)
           READ(2,*) OCOE(I)
        END DO
        J = 1
        DO X = LEFT, RIGHT, (RIGHT-LEFT)/500
          EFIELD(J)  = CMPLX(0.)
          DO I = 1, ENUM
           IF (X .GT. B) THEN
             DUM = EA1(I)*DEXP(-1*EH1(I)*(X-B))
           ELSE IF (X .GT. A .AND. X .LE. B) THEN
                  DUM = EA2(I)*DCOS(EH2(I)*X-ESI2(I))
                ELSE IF (X .GE. -1*D .AND. X .LE. A) THEN
                       DUM = EA3(I)*DSINH(EH3(I)*X)
                     ELSE IF (X .GE. -1*E .AND. X .LT. -1*D) THEN
                            DUM = EA4(I)*DCOS(EH4(I)*X-ESI4(I))
                          ELSE
                            DUM = EA5(I)*DEXP(EH5(I)*(X+E))
           ENDIF
           BETA = SQRT(KO**2*N1**2 + EH1(I)**2)
           EFIELD(J) = EFIELD(J)+ECOE(I)*CMPLX(DUM)*
     $                 CEXP(CMPLX(DUMO,-1*BETA*Z))
          END DO
          J = J + 1
        END DO
        J = 1
        DO X = LEFT, RIGHT, (RIGHT-LEFT)/500
          OFIELD(J) = CMPLX(0.)
          DO I = 1, ONUM
           IF (X .GT. B) THEN
             DUM = OA1(I)*DEXP(-1*OH1(I)*(X-B))
           ELSE IF (X .GT. A .AND. X .LE. B) THEN
                  DUM = OA2(I)*DCOS(OH2(I)*X-OSI2(I))
                ELSE IF (X .GE. -1*D .AND. X .LE. A) THEN
                       DUM = OA3(I)*DCOSH(OH3(I)*X)
                     ELSE IF (X .GE. -1*E .AND. X .LT. -1*D) THEN
                            DUM = OA4(I)*DCOS(OH4(I)*X-OSI4(I))
                          ELSE
```

```
                          DUM = OA5(I)*DEXP(OH5(I)*(X+E))
            ENDIF
            BETA = SQRT(KO**2*N1**2 + OH1(I)**2)
            OFIELD(J) = OFIELD(J)+OCOE(I)*CMPLX(DUM)*
     $                  CEXP(CMPLX(DUMO,-1*BETA*Z))
          END DO
          J = J + 1
        END DO
C
C  Compute the electric field with sinusoidal behaviour in medium 2
C  (or 3 or 4) and exponential behaviour in medium 1,3,4,5
C  (or 1,2,4,5 or 1,2,3,5).
C
        DO I = 1, 600
          SFIELD(I) = CMPLX(0.)
        END DO
        DO CODE = 1, 3
          READ(1,*) NUM
          DO I = 1, NUM
            READ(1,*) H1(I), H2(I), H3(I), H4(I), H5(I), CON(I)
            READ(2,*) COE(I)
          END DO
          J = 1
          DO X = LEFT, RIGHT, (RIGHT-LEFT)/500
           DO I = 1, NUM
            IF (X .GT. B) THEN
             IF (CODE .EQ. 1) THEN
               DUM = (DCOS(H3(I)*A)+CON(I)*DSIN(H3(I)*A))*
     $               DEXP(-1*H2(I)*(B-A))*DEXP(-1*H1(I)*(X-B))
             ELSE IF (CODE .EQ. 2) THEN
                    DUM = (DCOS(H2(I)*B)+CON(I)*DSIN(H2(I)*B))*
     $                     DEXP(-1*H1(I)*(X-B))
                  ELSE
                    DUM = (DCOS(H4(I)*D)-CON(I)*DSIN(H4(I)*D))*
     $                    DEXP(-1*H3(I)*(A+D))*DEXP(-1*H2(I)*
     $                    (B-A))*DEXP(-1*H1(I)*(X-B))
             ENDIF
            ELSE IF (X .GT. A .AND. X .LE. B) THEN
                    IF (CODE .EQ. 1) THEN
                      DUM = (DCOS(H3(I)*A)+CON(I)*DSIN(H3(I)*A))*
     $                       DEXP(-1*H2(I)*(X-A))
                    ELSE IF (CODE .EQ. 2) THEN
                          DUM = DCOS(H2(I)*X)+CON(I)*DSIN(H2(I)*X)
                         ELSE
                          DUM = (DCOS(H4(I)*D)-CON(I)*
     $                           DSIN(H4(I)*D))*DEXP(-1*H3(I)*
     $                           (A+D))*DEXP(-1*H2(I)*(X-A))
                    ENDIF
               ELSE IF (X .GE. -1*D .AND. X .LE. A) THEN
                       IF (CODE .EQ. 1) THEN
                          DUM = DCOS(H3(I)*X)+CON(I)*DSIN(H3(I)*X)
                       ELSE IF (CODE .EQ. 2) THEN
                              DUM = (DCOS(H2(I)*A)+CON(I)*DSIN(H2(I)*
     $                                A))*DEXP(H3(I)*(X-A))
                            ELSE
                              DUM = (DCOS(H4(I)*D)-CON(I)*DSIN(H4(I)*
     $                                D))*DEXP(-1*H3(I)*(X+D))
                       ENDIF
                    ELSE IF (X .GE. -1*E .AND. X .LT. -1*D) THEN
```

```
                              IF (CODE .EQ. 1) THEN
                               DUM = (DCOS(H3(I)*D)-CON(I)*DSIN(H3(I)*D))
     $                               *DEXP(H4(I)*(X+D))
                              ELSE IF (CODE .EQ. 2) THEN
                                    DUM = (DCOS(H2(I)*A)+CON(I)*
     $                                     DSIN(H2(I)*A))*DEXP(-1*H3(I)*
     $                                     (A+D))*DEXP(H4(I)*(X+D))
                                   ELSE
                                    DUM = DCOS(H4(I)*X)+
     $                                    CON(I)*DSIN(H4(I)*X)
                              ENDIF
                             ELSE
                              IF (CODE .EQ. 1) THEN
                                DUM = (DCOS(H3(I)*D)-CON(I)*DSIN(H3(I)*D))
     $                               *DEXP(H4(I)*(D-E))*DEXP(H5(I)*(X+E))
                              ELSE IF (CODE .EQ. 2) THEN
                                     DUM = (DCOS(H2(I)*A)+CON(I)*
     $                                      DSIN(H2(I)*A))*DEXP(-1*H3(I)*
     $                                      (A+D))*DEXP(H4(I)*(D-E))*
     $                                      DEXP(H5(I)*(X+E))
                                    ELSE
                                     DUM = (DCOS(H4(I)*E)-CON(I)*DSIN(H4(I)
     $                                      *E))*DEXP(H5(I)*(X+E))
                              ENDIF
            ENDIF
            BETA = SQRT(KO**2*N1**2 + H1(I)**2)
            SFIELD(J) = SFIELD(J)+COE(I)*CMPLX(DUM)*
     $                  CEXP(CMPLX(DUMO,-1*BETA*Z))
          END DO
          J = J + 1
        END DO
      END DO
C
C  Compute the total electric field at z for all the guided modes
C
      I = 1
      DO X = LEFT, RIGHT, (RIGHT-LEFT)/500
        WRITE(3,*) X, FIELD(I)+ OFIELD(I) + EFIELD(I) + SFIELD(I)
        I = I + 1
      END DO
C
      CLOSE (UNIT=1)
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)
      STOP
      END
```

```
C  PROGRAM : To find the electric field of the specified number of
C           radiation modes of parallel slab waveguides at z.
C
      CHARACTER*4 SIGN
      COMPLEX*8 FIELD(1500), COE(1500)
      REAL*8  H1(1500), H2(1500), H3(1500), H4(1500), H5(1500),
     $        A1(1500), A2(1500), A3(1500), A4(1500), A5(1500),
     $        SI1(1500), SI2(1500), SI3(1500), SI4(1500), SI5(1500),
     $        A, B, C, D, E, F, KO, PI, N1, N2, N3, N4, N5,
     $        DUM, X, LEFT, RIGHT, Z, DUMO, BETA, DUM1
      INTEGER I, J, NUM
C
      OPEN (UNIT=1,FILE='COPR.DAT',STATUS='OLD')
      OPEN (UNIT=2,FILE='COPRCOE.DAT',STATUS='OLD')
      OPEN (UNIT=3,FILE='COPR.PLOT',STATUS='NEW')
C
C  Obtain information from user and old files
C
      WRITE(*,*) 'What is  z ? (in meters)'
      READ(*,*) Z
      WRITE(3,*) 500
      TYPE *,'Range of X to be plotted (between X=LEFT & X=RIGHT)?'
      WRITE(*,*) 'LEFT = ?'
      READ(*,*) LEFT
      WRITE(*,*) 'RIGHT = ?'
      READ(*,*) RIGHT
      READ(1,*) N1, N2, N3, N4, N5, KO
      READ(1,*) A, B, C, D, E, F
      READ(1,*) NUM
      DO I = 1, NUM
         READ(1,*) H1(I), H2(I), H3(I), H4(I), H5(I)
         SI1(I) = H1(I)*(C-B)
         SI5(I) = H5(I)*(F-E)
         SI2(I) = H2(I)*B-DATAN((H1(I)/H2(I))/DTAN(SI1(I)))
         SI4(I) = DATAN((H5(I)/H4(I))/DTAN(SI5(I)))-H4(I)*E
         SI3(I) = H3(I)*A-DATAN((H2(I)/H3(I))*DTAN(H2(I)*A-SI2(I)))
         A1(I) = 1.0
         A2(I) = -1*A1(I)*DSIN(SI1(I))/DCOS(H2(I)*B-SI2(I))
         A3(I) = A2(I)*DCOS(H2(I)*A-SI2(I))/DCOS(H3(I)*A-SI3(I))
         A4(I) = A3(I)*DCOS(H3(I)*D+SI3(I))/DCOS(H4(I)*D+SI4(I))
         A5(I) = A4(I)*DCOS(H4(I)*E+SI4(I))/DSIN(SI5(I))
         READ(2,*) COE(I)
      END DO
      PI = 3.141592654
      DUMO = 0.0
C
C  Comopute the electric field for the radiation modes used at z
C
      J = 1
      DO X = LEFT, RIGHT, (RIGHT-LEFT)/500
        FIELD(J)  = CMPLX(0.)
        DO I = 1, NUM
         IF (X .GT. B) THEN
           DUM = A1(I)*DSIN(H1(I)*(X-B)-SI1(I))
         ELSE IF (X .GT. A .AND. X .LE. B) THEN
                DUM = A2(I)*DCOS(H2(I)*X-SI2(I))
              ELSE IF (X .GE. -1*D .AND. X .LE. A) THEN
                     DUM = A3(I)*DCOS(H3(I)*X-SI3(I))
                   ELSE IF (X .GE. -1*E .AND. X .LT. -1*D) THEN
```

```
                        DUM = A4(I)*DCOS(H4(I)*X-SI4(I))
                      ELSE
                        DUM = A5(I)*DSIN(H5(I)*(X+E)+SI5(I))
          ENDIF
          DUM1 = K0**2*N1**2 - H1(I)**2
          IF (DUM1 .LT. 0) THEN
             BETA = SQRT(-1*DUM1)
             SIGN = 'IMAG'
          ELSE
             BETA = SQRT(DUM1)
             SIGN = 'REAL'
          ENDIF
          IF (SIGN .EQ. 'REAL') THEN
             FIELD(J) = COE(I)*CMPLX(DUM)*CEXP(CMPLX(DUM0,-1*BETA*Z))+
     $                  FIELD(J)
          ELSE
             FIELD(J) = COE(I)*CMPLX(DUM)*CEXP(CMPLX(-1*BETA*Z)) +
     $                  FIELD(J)
          ENDIF
        END DO
        WRITE(3,*) X, FIELD(J)
        J = J + 1
      END DO
C
      CLOSE (UNIT=1)
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)
      STOP
      END
```

```
C  PROGRAM : To plot the electric field of the guided and radiation
C           modes for the parallel slab waveguide at z.
C
      CHARACTER*20  TAB1, TAB2
      CHARACTER*1   PLOT, CHOICE
      COMPLEX*8     DUM1, DUM2
      REAL*8        X
      INTEGER       I, NUM
C
      WRITE(*,*) 'TABLE 1 = ?'
      READ(*,10) TAB1
10    FORMAT(A20)
      WRITE(*,*) 'TABLE 2 = ?'
      READ(*,10) TAB2
C
      OPEN (UNIT=1,FILE=TAB1,STATUS='OLD')
      OPEN (UNIT=2,FILE=TAB2,STATUS='OLD')
      OPEN (UNIT=3,FILE='COPPLOT.DAT',STATUS='NEW')
C
C  File contains the output fo the electric field
C
      TYPE *,'Plot: A) Electric field for the first file'
      TYPE *,'    : B) Electric field for the second file'
      TYPE *,'    : C) Electric field for the sum of two files'
      READ(*,20) CHOICE
      IF (CHOICE .EQ. 'A') THEN
         READ(1,*) NUM
         WRITE(3,*) NUM
         DO I = 1, NUM
            READ(1,*) X, DUM1
            WRITE(3,*) X, CABS(DUM1)
         END DO
      ELSE IF (CHOICE .EQ. 'B') THEN
              READ(2,*) NUM
              WRITE(3,*) NUM
              DO I = 1, NUM
                READ(2,*) X, DUM1
                WRITE(3,*) X, CABS(DUM1)
              END DO
           ELSE
              WRITE(*,*) 'Plot complex(C) or absolute(S) field ?'
              READ(*,20) PLOT
              READ(1,*) NUM
              READ(2,*) NUM
              WRITE(3,*) NUM
              DO I = 1, NUM
                READ(1,*) X, DUM1
                READ(2,*) X, DUM2
                IF (PLOT .EQ. 'C') THEN
                  WRITE(3,*) X, DUM1+DUM2
                ELSE
                  WRITE(3,*) X, CABS(DUM1+DUM2)
                ENDIF
              END DO
      ENDIF
20    FORMAT(A1)
C
      STOP
      END
```

VITA

Eva Wai-Ling Kam was born in Hong Kong on September 23, 1965. She attended secondary school in Hong Kong and was graduated from St. Paul's Secondary School in May 1982. The following September, she went to the United States and entered Webb School of Knoxville as a senior of high school. She was graduated from that high school in June 1983.

In the fall of 1983, she entered University of Tennessee, Knoxville and she received a Bachelor of Science degree in Electrical Engineering in August 1986. In the following September, she accepted a graduate assistantship at the same university and began her Master's degree study. The author is going to Canada after graduation.