

## TWO DIMENSIONAL QSC MODE SOLVERS

TWO DIMENSIONAL QSC MODE SOLVERS  
FOR  
ARBITRARY DIELECTRIC WAVEGUIDE

By  
BIN XU, B.E.

A Thesis

Submitted to the School of Graduate Studies  
in Partial Fulfillment of the Requirements  
for the Degree  
Master of Applied Science

McMaster University

© Copyright by Bin Xu, December, 2006

MASTER OF APPLIED SCIENCE (2006)

McMaster University

(Electrical and Computer Engineering)

Hamilton, Ontario

TITLE: Two Dimensional QSC Mode Solvers for Arbitrary  
Dielectric Waveguide

AUTHOR: Bin Xu, B.E. (Tsinghua University, P. R. China)

SUPERVISOR: Professor Xun Li

NUMBER OF PAGES: viii, 87

## **Abstract**

Novel scalar and full-vectorial mode solvers based on quadratic spline collocation (QSC) method have been developed in MATLAB for optical dielectric waveguide with arbitrary two-dimensional cross-section and refractive index profile.

Compared with the conventional finite difference mode solver in the literature and a commercial mode solver, the QSC mode solvers are simple and easy to implement in MATLAB without losing the accuracy of the mode solutions. The scalar mode solver is fast for solving weakly guiding waveguides. Three typical rib waveguides are calculated by the QSC scalar mode solver and compared with the numerical results of a finite difference scalar mode solver in the literature. The full-vectorial mode solver is capable of solving both weakly and strongly guiding waveguides. Typical numerical examples are calculated by the full-vectorial QSC mode solver and the solver is verified by comparing the results to a commercial mode solver.

At the end of the thesis, methods of calculating leaky and radiation modes of general dielectric waveguides and possible methods of increasing the accuracy of the QSC mode solvers are proposed.

## **Acknowledgements**

First I would like to express my extreme thanks to my supervisor, Professor Dr. Xun Li, for his full guidance and full financial support throughout my Master Degree study at McMaster University. Without his full support, the completion of the degree won't be possible.

Second I would like to appreciate Professors Dr. W. P. Huang, Dr. S. Kumar, and Dr. Reilly, whose graduate courses I have attended. Especial thanks should be given to Dr. Huang, for his two major courses and his insight advice and help for the thesis completion.

Third I would like to express my thanks to Dr. Seyed M. Sadeghi, Dimitry, and other labmates for all the useful discussion and advice. I also thank Cheryl and other staffs in the ECE department for all of their help and support.

Last but not the least, I would dedicated the thesis to my parents and girlfriend for their unconditional support.

## Table of Contents

|  |     |
|--|-----|
| Abstract .....   | iii |
| Acknowledgements .....   | iv  |
| Table of Contents .....  | v   |
| List of Illustrations and Tables .....   | vii |
| <br>   |     |
| Chapter 1. Introduction .....  | 1   |
| 1.1 Background and Overview .....  | 1   |
| 1.2 Thesis Organization .....  | 4   |
| <br>   |     |
| Chapter 2. Basic Electromagnetic Theory .....  | 5   |
| 2.1 Maxwell's Equations .....  | 5   |
| 2.2 Vector Helmholtz Wave Equation .....   | 6   |
| 2.2.1 Vector Wave Equation for Electric Field .....  | 6   |
| 2.2.2 Vector Wave Equation for Magnetic Field .....  | 8   |
| 2.3 Semi-vectorial and Scalar Approximations .....   | 10  |
| 2.4 Interface Conditions and Perfect Conductor Boundary Conditions .....                         | 11  |
| <br>   |     |
| Chapter 3. Quadratic Spline Collocation Method .....   | 14  |
| 3.1 Quadratic Spline Collocation Method General Discussion .....                                 | 14  |
| 3.1.1 Validity of Quadratic Spline Collocation Method for Solving Vector Wave<br>Equations ..... | 15  |
| 3.1.2 Promising Advantage of Piecewise Quadratic Spline Interpolant .....                        | 16  |
| 3.2 QSC Method for A Single PDE in 2D Space .....  | 17  |
| 3.2.1 Problem Description .....  | 17  |
| 3.2.2 QSC Mesh and Collocation Points .....  | 17  |
| 3.2.3 One-dimensional Basis Functions .....  | 18  |
| 3.2.4 Two-dimensional Basis Functions and Interpolant .....                                      | 20  |
| 3.2.5 Collocation Method .....   | 21  |
| 3.2.6 Dirichlet and Neumann Boundary Conditions .....  | 22  |
| 3.2.7 In Matrix Form .....   | 24  |
| 3.3 QSC Method for A System of Two PDEs .....  | 28  |
| <br>   |     |
| Chapter 4. QSC Scalar Mode Solver .....  | 30  |
| 4.1 The Governing Equation and Boundary Condition .....  | 30  |
| 4.2 The Simple MATLAB Program .....  | 32  |
| 4.2.1 Inputs .....   | 32  |
| 4.2.2 Formulation .....  | 33  |
| 4.2.3 Eigen-System Solution .....  | 35  |

|   |    |
|---|----|
| 4.2.4 Outputs .....                                   | 36 |
| 4.3 Numerical Results and Discussion .....            | 36 |
| Chapter 5.    QSC Full-vectorial Mode Solver.....     | 41 |
| 5.1 Governing Equations and Boundary Conditions.....  | 41 |
| 5.1.1 Choice of Governing Equations .....             | 41 |
| 5.1.2 Governing Equations and Boundary Condition..... | 42 |
| 5.1.3 QSC Matrix Form.....                            | 44 |
| 5.2 The Simple MATLAB Program .....                   | 46 |
| 5.2.1 Inputs.....                                     | 46 |
| 5.2.2 Formulation .....                               | 47 |
| 5.2.3 Eigen-System Solution .....                     | 48 |
| 5.2.4 Calculate E Field from H Field .....            | 48 |
| 5.2.5 Outputs .....                                   | 49 |
| 5.3 Numerical Results and Discussion .....            | 49 |
| 5.3.1 Ridge Waveguide.....                            | 49 |
| 5.3.2 Slot Waveguide.....                             | 53 |
| 5.3.3. Mesh Average Technique for QSC .....           | 55 |
| Chapter 6.    Conclusion and Future Work.....         | 62 |
| 6.1 Thesis Summary .....                              | 62 |
| 6.2 Future Work.....                                  | 64 |
| 6.2.1 Calculating Leaky and Radiation Modes .....     | 64 |
| 6.2.2 Higher Accuracy Solvers.....                    | 69 |
| 6.2.3 Extending to Propagation Simulation.....        | 70 |
| Appendix A: QSC Scalar Mode Solver Example.....       | 71 |
| Appendix B: QSC Vectorial Mode Solve Example.....     | 76 |
| Bibliography.....                                     | 86 |

## List of Illustrations and Tables

|  |    |
|--|----|
| FIGURE 3.1. THE COLLOCATION POINTS FOR $M=4, N=3$ .....  | 19 |
| FIGURE 3.2. AN EXAMPLE OF GENERAL ONE-DIMENSIONAL QUADRATIC SPLINE BASIS FUNCTIONS .....   | 19 |
| FIGURE 3.3. AN EXAMPLE OF ONE-DIMENSIONAL BASIS FUNCTIONS WITH DIRICHLET BOUNDARY<br>CONDITION.....  | 23 |
| FIGURE 3.4. THE SPARSE BI-QUADRATIC MATRIX K FOR $M=8, N=6$ , THE MAXIMUM NUMBER OF NON-ZERO<br>ELEMENT IS 352. ....   | 27 |
| FIGURE 4.1. TYPICAL SEMICONDUCTOR RIB WAVEGUIDE STRUCTURES. ....   | 38 |
| FIGURE 5.1. TYPICAL STRUCTURE OF RIDGE WAVEGUIDES.....   | 50 |
| FIGURE 5.2. THE COMPUTATION WINDOW OF SLOT WAVEGUIDE.....  | 54 |
| FIGURE 5.3. QSC ORIGINAL MESH AND LUMERICAL'S MESHES INDEX-X, INDEX-Y, INDEX-Z .....   | 58 |
| FIGURE 5.4. COLLOCATION POINTS IN THE ORIGINAL QSC MESH CANNOT WELL REFLECT THE REAL<br>INTERFACE .....  | 60 |
| FIGURE 5.5. EFFECTIVE INDEX CURVE OF SLOT WAVEGUIDE .....  | 61 |
| FIGURE 6.1. THE REFRACTIVE INDEX PROFILE AND PML PROFILE OF A TYPICAL ARROW WAVEGUIDE [9]67  | 67 |
| FIGURE 6.2. THE LOWEST LEAKY MODE OF THE ARROW WAVEGUIDE .....   | 68 |
| FIGURE 6.3. THE SECOND LOWEST LEAKY MODE OF THE ARROW WAVEGUIDE .....  | 68 |
| TABLE 4.1. THE SEMICONDUCTOR RIB WAVEGUIDE STRUCTURES BT1, BT2, AND BT3 FROM [1] .....   | 39 |
| TABLE 4.2. COMPARISON OF MODAL INDICES BETWEEN QSC SCALAR MODE SOLVER AND FINITE<br>DIFFERENCE SCALAR MODE SOLVER IN [1], WITH NEARLY IDENTICAL GRID SIZES AND PARAMETERS.39 | 39 |
| TABLE 4.3. FUNDAMENTAL SCALAR MODE 5% CONTOUR AND COMPUTATION TIME FOR BT1, BT2, AND   |    |



|   |    |
|---|----|
| BT3 .....   | 40 |
| TABLE 5.1. THE OPTICAL AND GEOMETRICAL PARAMETERS OF THE RIDGE WAVEGUIDE .....  | 50 |
| TABLE 5.2. THE SIMULATION PARAMETERS FOR BOTH LUMERICAL AND QSC .....   | 50 |
| TABLE 5.3. FUNDAMENTAL QUASI-TE AND FUNDAMENTAL QUASI-TM MODES OF RIDGE WAVEGUIDE<br>CALCULATED BY BOTH QSC FULL-VECTORIAL MODE SOLVER AND LUMERICAL MODE<br>SOLUTION. .... | 52 |
| TABLE 5.4. SECONDARY QUASI-TE AND SECONDARY QUASI-TM MODES OF RIDGE WAVEGUIDE<br>CALCULATED BY BOTH QSC FULL-VECTORIAL MODE SOLVER AND LUMERICAL MODE<br>SOLUTION. ....     | 53 |
| TABLE 5.5. THE OPTICAL AND GEOMETRICAL PARAMETERS OF THE SLOT WAVEGUIDE.....  | 55 |
| TABLE 5.6. THE SIMULATION PARAMETERS FOR BOTH LUMERICAL AND QSC .....   | 55 |
| TABLE 5.7. QSC SIMULATION RESULTS OF SLOT WAVEGUIDE.....  | 56 |
| TABLE 5.8. LUMERICAL SIMULATION RESULTS OF SLOT WAVEGUIDE .....   | 57 |
| TABLE 5.9. MODAL INDICES OF DIFFERENCE SIMULATION CONDITIONS .....  | 59 |

# Chapter 1. Introduction

## 1.1 Background and Overview

The basic building block of all key elements in modern optical communication systems is longitudinally invariant waveguide. Like the connecting “wire” of the electric circuit, optical waveguide functions as the link in optoelectronic integrated circuits (OEICs) and also forms the basis of photonic devices, including active devices and passive devices. Therefore, accurate modeling of the characteristics of the constituent waveguides is one of the most critical issues in photonics and optoelectronic simulation and design.

Even for the most fundamental waveguide, for example, the three layer slab waveguide, the analytic solution is usually out of reach. Numerical modeling and simulation are the alternatives. With the fast development of computer technology, most of the ordinary personal computers nowadays are more powerful than the super computers of a decade ago. So, even when there is an analytic solution available, most people will prefer to choose a reliable numerical tool to simulate the waveguide rather than going deep into the complicated analytical analysis. In other words, the numerical analytical tools will play a larger and larger role in the waveguide analysis, which is the basis of further advanced designs of optical devices and circuits.

For a complete modeling for the longitudinally invariant optical waveguide, mode

analysis of optical waveguides is a fundamental and important issue. In the past more than two decades, continuous and significant effort has been made to solve the full-vectorial modes of longitudinally invariant waveguide (also called as planar optical waveguide in the literature, such as rib, ridge, slot, etc.) in order to fully describe the behavior of optical waveguides. Various numerical methods and techniques have been investigated and developed.

Among the numerous numerical mode solvers in the literature, characterized by the numerical algorithms, finite difference (FD) method and finite element (FE) method are two of the major types.

The basic idea of finite difference method is to replace the partial differential operators of the vector solutions of Maxwell Solutions by difference operators [1]. The finite difference idea is straightforward and understandable, and for most of the conventional planar dielectric waveguide, like ridge waveguide and directional coupler, it is one of the most effective choices for its simplicity. However, finite difference mode solver may meet great challenges when dealing with irregular interfaces of the cross-section. Usually the finite difference mode solver is much oriented to specific categories of waveguides such as the rectangular waveguides with dielectric interfaces along the coordinates.

The finite element idea on the other hand may not be described and understood easily. And it is more complicated than the finite difference method to implement, though the finite element mode solvers are generally regarded as one of most

powerful and adaptable methods for all kinds of waveguides with various cross section geometries. The reason of the generality of the finite element solver is mainly the result from its basic idea: the basic operating blocks in the finite element method are usually triangles. However, though the traditional finite element method provides us a great possibility of finding a perfect numerical solution, the processes of finding a suitable and effective mesh for a specific problem may become overwhelming.

Besides the conventional finite element method and finite difference method, there are also other numerical approaches available in the literature. For example, D. Marcus proposed to use a complete and orthogonal set of basis functions, e.g. sinusoidal basis functions, to approximate the unknown fields, and when a proper number of basis functions are chosen, the method will result in a matrix equation with much lower rank compared with equivalent matrix forms resulting from finite difference and finite element methods [2]. The idea of using basis functions to simulate the unknown solution is also used in the conventional finite element method. But instead of using global complete and orthogonal basis, a set of linear local basis functions are usually used.

Quadratic spline collocation method is one of numerical approaches in the literature to solve boundary value problems. No particular interest in the literature has been shown in the past decades. The reason might be that it seems more complicated and less accurate. However, with the improvement of the accuracy of this method in the 1990s and 2000s [3], it is worth trying to include this method in the waveguide

solver family. In this thesis, starting from the basic idea of quadratic spline collocation method, both scalar and full-vectorial mode solvers were explored to solve two-dimensional waveguide with arbitrary cross-sections.

## **1.2 Thesis Organization**

Chapter 2 reviews the basic electromagnetic theory relevant to this work and some common techniques for the analysis of dielectric waveguides. In Chapter 3, the full coverage will be concentrated on the quadratic spline collocation method. In Chapter 4, two-dimensional scalar mode solvers are developed based on QSC method. In Chapter 5, based on the previous Chapters, a full vectorial two-dimensional mode solver is fully examined. Chapter 6 concludes the thesis and presents the future work, including the proposal of calculating leaky and radiation modes and the methods of increasing the accuracy of QSC mode solvers.

## Chapter 2. Basic Electromagnetic Theory

In this chapter the fundamental equations and concepts applied throughout the whole thesis are presented. The Maxwell's equations, the modal solutions for longitudinally invariant waveguides, and the commonly used levels of approximation to modes (scalar and semi-vectorial) are discussed in the following sections.

### 2.1 Maxwell's Equations

In complex form, time-harmonic Maxwell's equations in source-free regions are

[4]

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} = -j\omega \mathbf{B} \quad (2.1)$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} = j\omega \mathbf{D} \quad (2.2)$$

$$\nabla \cdot \mathbf{D} = 0 \quad (2.3)$$

$$\nabla \cdot \mathbf{B} = 0. \quad (2.4)$$

In these equations,  $\mathbf{E}$  and  $\mathbf{H}$  are the electric and magnetic field vectors,  $\mathbf{D}$  is the electric displacement current vector and  $\mathbf{B}$  the magnetic flux density vector. A time-harmonic factor  $e^{j\omega t}$  is assumed and suppressed from hereon, with the angular frequency  $\omega$  related to the vacuum wavelength  $\lambda$  by

$$\omega = \frac{2\pi c}{\lambda} \quad (2.5)$$

where  $c$  is the light speed in free space. In isotropic media we have the following constitutive relations

$$\mathbf{D} = \varepsilon \mathbf{E} \quad (2.6)$$

$$\mathbf{B} = \mu \mathbf{H} \quad (2.7)$$

with the permittivity  $\varepsilon$  and permeability  $\mu$  given by

$$\varepsilon = \varepsilon_0 \varepsilon_r \quad (2.8)$$

$$\mu = \mu_0 \mu_r. \quad (2.9)$$

In the two equations above,  $\varepsilon_0$  and  $\mu_0$  are the permittivity and permeability of free space;  $\varepsilon_r$  and  $\mu_r$  are the relative permittivity and relative permeability of the medium. Since most of the materials widely used in the optical waveguides are isotropic and non-magnetic ( $\mu_r = 1$ ), we restrict ourselves to these kind of dielectric materials. The other two definitions which will be frequently used throughout this work are the wave number of free space  $k_0$  defined as

$$k_0 = \omega \sqrt{\varepsilon_0 \mu_0} \quad (2.10)$$

and the refractive index  $n$  defined as

$$n = \sqrt{\varepsilon_r} \quad (2.11)$$

## 2.2 Vector Helmholtz Wave Equation

### 2.2.1 Vector Wave Equation for Electric Field

By taking the curl of the both sides of (2.1) and using (2.2), (2.6), (2.7), (2.10), and (2.11), we can obtain the following equation for electric field

$$\nabla \times (\nabla \times \mathbf{E}) = k_0^2 n^2 \mathbf{E} \quad (2.12)$$

Using the identity

$$\nabla \times (\nabla \times \mathbf{E}) = \nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E} \quad (2.13)$$

and

$$\nabla \cdot (n^2 \mathbf{E}) = \mathbf{E} \cdot \nabla n^2 + n^2 \nabla \cdot \mathbf{E} = 0 \quad (2.14)$$

by (2.3), we obtain from (2.12) the vector Helmholtz wave equation for the electric field

$$\nabla^2 \mathbf{E} + \nabla \left( \mathbf{E} \cdot \frac{\nabla n^2}{n^2} \right) + n^2 k_0^2 \mathbf{E} = 0 \quad (2.15)$$

Since we will concentrate on the longitudinally invariant waveguides, in which the refractive index depends on the two transverse coordinates only ( $n = n(x, y)$ ), the modal solution of Maxwell's equations can be written in the following form

$$\mathbf{E} = \mathbf{E}_0(x, y) \exp(-j\beta z) \quad (2.16)$$

where  $\mathbf{E}_0$  is a vector and has three field component as  $E_x(x, y)$ ,  $E_y(x, y)$ , and  $E_z(x, y)$ ;  $\beta$  is the propagation constant and its relation with the effective refractive index  $n_{eff}$ , which is a commonly used concept in the waveguide theory, is

$$\beta = n_{eff} k_0 \quad (2.17)$$

From (2.16) we can decompose the Laplace operator  $\nabla^2$  in the above equation as

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - \beta^2. \quad (2.18)$$

And, in Cartesian coordinates which will be the default coordinate system throughout the whole thesis, the vector wave equation can be decomposed into  $x$ ,  $y$ , and  $z$  components. However, since we are considering source free waveguide and have the divergence identity (2.3), the  $E_z$  field component can be expressed in terms of the



transverse components  $E_x$  and  $E_y$  as

$$E_z = \frac{-i}{n^2\beta} \left( \frac{\partial}{\partial x} (n^2 E_x) + \frac{\partial}{\partial y} (n^2 E_y) \right) \quad (2.19)$$

And from (2.1) and (2.7), together with (2.19), the three magnetic field components can be expressed in the terms of  $E_x$  and  $E_y$  as

$$H_x = \frac{-1}{\omega\mu_0\beta} \left( \frac{\partial^2 E_y}{\partial x^2} - \frac{\partial^2 E_x}{\partial x\partial y} - n^2 k^2 E_y \right) \quad (2.20)$$

$$H_y = \frac{-1}{\omega\mu_0\beta} \left( \frac{\partial^2 E_x}{\partial y^2} - \frac{\partial^2 E_y}{\partial x\partial y} - n^2 k^2 E_x \right) \quad (2.21)$$

$$H_z = \frac{i}{\omega\mu_0} \left( \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right). \quad (2.22)$$

Thus, only two of the three equations from the vector Helmholtz wave equation for electric field (2.15)

$$\frac{\partial^2 E_x}{\partial x^2} + \frac{\partial^2 E_x}{\partial y^2} + (n^2 k_0^2 - \beta^2) E_x + 2 \frac{\partial}{\partial x} \left[ E_x \frac{\partial \ln(n)}{\partial x} + E_y \frac{\partial \ln(n)}{\partial y} \right] = 0 \quad (2.23)$$

$$\frac{\partial^2 E_y}{\partial x^2} + \frac{\partial^2 E_y}{\partial y^2} + (n^2 k_0^2 - \beta^2) E_y + 2 \frac{\partial}{\partial y} \left[ E_x \frac{\partial \ln(n)}{\partial x} + E_y \frac{\partial \ln(n)}{\partial y} \right] = 0 \quad (2.24)$$

need to be solved for the total mode solutions.

## 2.2.2 Vector Wave Equation for Magnetic Field

Similarly, starting from the Maxwell's equations, we can derive the vector wave equation for magnetic field and the two desired component equations. Taking the curl of both sides of (2.2), and using (2.6), (2.8), and (2.11), we obtain

$$\nabla \times (\nabla \times \mathbf{H}) = j\omega\epsilon_0 \nabla \times (n^2 \mathbf{E}) \quad (2.25)$$

Using vector identities

$$\nabla(n^2\mathbf{E}) = n^2(\nabla \times \mathbf{E}) + (\nabla n^2) \times \mathbf{E} \quad (2.26)$$

$$\nabla \times (\nabla \times \mathbf{H}) = \nabla(\nabla \cdot \mathbf{H}) - \nabla^2 \mathbf{H} \quad (2.27)$$

and (2.1), (2.2), (2.4), (2.7), and (2.10) we further obtain

$$\nabla^2 \mathbf{H} + n^2 k_0^2 \mathbf{H} + (\nabla n^2) \times \left[ \frac{1}{n^2} (\nabla \times \mathbf{H}) \right] = 0 \quad (2.28)$$

By vector identity

$$(\nabla n^2) \times \left[ \frac{1}{n^2} (\nabla \times \mathbf{H}) \right] = \frac{1}{n^2} [(\nabla n^2) \times (\nabla \times \mathbf{H})] \quad (2.29)$$

the vector wave equation for magnetic field assumes the form

$$\nabla^2 \mathbf{H} + n^2 k^2 \mathbf{H} + \left[ \frac{\nabla n^2}{n^2} \times (\nabla \times \mathbf{H}) \right] = 0 \quad (2.30)$$

And the transverse components of (2.30) yield

$$\frac{\partial^2 H_y}{\partial x^2} + \frac{\partial^2 H_y}{\partial y^2} + (n^2 k_0^2 - \beta^2) H_y - 2 \frac{\partial \ln(n)}{\partial x} \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) = 0 \quad (2.31)$$

$$\frac{\partial^2 H_x}{\partial x^2} + \frac{\partial^2 H_x}{\partial y^2} + (n^2 k_0^2 - \beta^2) H_x + 2 \frac{\partial \ln(n)}{\partial y} \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) = 0 \quad (2.32)$$

which are the two necessary equations to be solved for the whole modal solutions.

Here, we write the  $H_y$  first to reveal a better comparison with the electric component equations (2.23) and (2.24), since for plane waves in free space,  $H_y$  and  $H_x$  have the same transverse spatial distributions as  $E_x$  and  $E_y$  [2].

And similarly, we can express the rest components of the magnetic field and electric field into combination of  $H_y$  and  $H_x$ .

$$E_z = \frac{i}{n^2 \omega \epsilon_0} \left( \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x} \right) \quad (2.33)$$

$$E_x = \frac{1}{\omega \epsilon_0 \beta} \left\{ k_0^2 H_y - \frac{\partial}{\partial x} \left[ \frac{1}{n^2} \left( \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x} \right) \right] \right\} \quad (2.34)$$

$$E_y = \frac{-1}{\omega \epsilon_0 \beta} \left\{ k_0^2 H_x + \frac{\partial}{\partial y} \left[ \frac{1}{n^2} \left( \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x} \right) \right] \right\} \quad (2.35)$$

$$H_z = \frac{i}{\omega \mu_0} \left( \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right). \quad (2.36)$$

## 2.3 Semi-vectorial and Scalar Approximations

The wave equations for electric field (2.23) and (2.24), or the wave equations for magnetic field (2.31) and (2.32) are the full-vectorial equations, the solutions of which are the exact solutions of Maxwell equations for longitudinally invariant waveguides. To get the most accurate solutions in a numerical approach, without other considerations such as memory limit and time consumption, we should always attempt to solve the full-vectorial equations. However, when we consider numerical modeling, memory limit and time consumption in most cases are two of the most important issues and can never be neglected. Thus, proper approximations, which will simplify the original problems within the allowance of accuracy, are always sought and used. Here in the waveguide problems, two major approximations are widely utilized for specific problems.

The semi-vectorial approximation arises from the physically justifiable assumption that one of the two transverse components is dominant over the other one.

The assumption that  $E_x$  or  $H_y$  is dominant will lead to the quasi-TE equation for  $E_x$  or  $H_y$  after discarding the minor component  $E_y$  or  $H_x$  from (2.23) or (2.31). And the quasi-TM equation can be obtained through the same way when we assume that  $E_y$  or  $H_x$  is the principal component and  $E_x$  or  $H_y$  can be dropped from (2.24) or (2.32) [5]. The semi-vectorial approximation can reduce the two full-vectorial coupled equations into a single equation, which greatly reduces the whole computation effort. The semi-vectorial approximation is suitable for the waveguides which well support solutions polarized along the axis, like the rectangular waveguides of low index-contrast and the rib waveguides [6].

For weakly guiding structure, namely the structure consisting of materials whose refractive index  $n$  is nearly identical in those regions where the field intensity is large, the scalar approximation can be applied [2]. Scalar approximation discards the polarization dependence and for smooth refractive index, the scalar solution satisfies the scalar Helmholtz equation

$$\nabla_T^2 \phi + (k_0^2 n^2 - \beta^2) \phi = 0 \quad (2.37)$$

at all points, where  $\phi$  can represent any component.

## 2.4 Interface Conditions and Perfect Conductor Boundary Conditions

At an interface between two media in a planar dielectric waveguide, the fields in medium 1 and medium 2 of an interface must satisfy the following interface

conditions

$$\hat{n} \times (\mathbf{E}_1 - \mathbf{E}_2) = 0 \quad (2.38)$$

$$\hat{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s \quad (2.39)$$

$$\hat{n} \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{J}_s \quad (2.40)$$

$$\hat{n} \cdot (\mathbf{B}_1 - \mathbf{B}_2) = 0 \quad (2.41)$$

where  $\rho_s$  is the surface charge,  $\mathbf{J}_s$  is the surface current density, and  $\hat{n}$  is a unit vector with the normal direction of the interface. The above four equations can be derived from the Maxwell's integral equations [5].

For a source-free passive dielectric waveguide, which means  $\rho_s = 0$  and  $\mathbf{J}_s = 0$ , by using the constitutive relations (2.6) and (2.7), the boundary conditions (2.38) ~ (2.41) are reduced to

$$E_1^{Tangential} = E_2^{Tangential} \quad (2.42)$$

$$\varepsilon_1 E_1^{normal} = \varepsilon_2 E_2^{normal} \quad (2.43)$$

$$H_1^{Tangential} = H_2^{Tangential} \quad (2.44)$$

$$\mu_1 H_1^{normal} = \mu_2 H_2^{normal} \quad (2.45)$$

where  $\varepsilon_1$ ,  $\varepsilon_2$  are the permittivity of medium 1 and medium 2, respectively,  $\mu_1$ ,  $\mu_2$  are the permeability of medium 1 and medium 2, respectively. From (2.42) ~ (2.45), we can conclude that in a passive source-free waveguide, the tangential part of electric field and magnetic field are continuous at the dielectric interfaces, and for most of the typical cases where  $\varepsilon_1 \neq \varepsilon_2$  and  $\mu_1 = \mu_2 = 0$ , the normal part of the electric field is discontinuous and the normal part of the magnetic field is continuous at the dielectric

interfaces.

For perfect electric conducting boundary conditions, which are widely applied in numerical simulations for unbounded problems, we can assume medium 1 is a perfect conductor, which means the tangential electric field  $E_1^{Tangential}$  must be zero (equal-potential of the perfect conducting surface), and the normal magnetic flux  $H_1^{normal}$  must be zero since there is no electric field inside the perfect conductor. Thus, (2.42) and (2.45) for a perfect conductor surface become

$$E_2^{Tangential} = 0 \quad (2.46)$$

$$H_2^{normal} = 0 \quad (2.47)$$

Furthermore, from Maxwell equations (2.1), (2.2) and (2.46), (2.47), we obtain the first derivatives of the electric field and the magnetic field in dielectric medium 2 at the boundary of the perfect conductor medium 1

$$\frac{\partial E_2^{normal}}{\partial n} = 0 \quad (2.48)$$

$$\frac{\partial H_2^{Tangential}}{\partial n} = 0 \quad (2.49)$$

Namely for a perfect conductor boundary, the tangential electric field is zero and the first derivative of the normal electric field is zero; the normal magnetic field is zero and the first derivative of the tangential magnetic field is zero.

# Chapter 3. Quadratic Spline Collocation Method

In this thesis, we will explore the quadratic spline collocation method for solving the vector Helmholtz wave equations. Before we start to solve our specific problems, we will first describe the basic idea of quadratic spline collocation (QSC) method. Then, QSC method for solving a single general partial differential equation (PDE) is discussed. Since we will solve the set of vector wave equations, QSC method for a set of equations will be discussed at the end of this chapter.

## 3.1 Quadratic Spline Collocation Method General Discussion

The basic idea of the QSC method is to approximate the original solution of a partial differential equation by a set of localized quadratic basis functions (quadratic spline) and to calculate the weights of each basis by forcing the residual between the exact solution and the approximate solution to be zero at a complete set of points (collocation). The piecewise polynomial collocation approximation guarantees that the values at the collocation points are accurate and the values between the collocation points are stable without oscillation, since three stable quadratic basis functions at most determine the values, which are shown straightforwardly in the one-dimensional QSC basis function example in Figure 3.2. This property makes this method

substantially different, hence more useful, from a global polynomial expansion which is capable of fitting a given set of points but fails to avoid oscillation between the given points, since all of the values are determined by all the global coefficients.

### **3.1.1 Validity of Quadratic Spline Collocation Method for Solving Vector Wave Equations**

Before we attempt to use the quadratic spline collocation method, we first prove its validity for solving the vector wave equations. Unlike the eigen-function expansion, such as the eigen-expansion of a complete orthogonal set of sine functions [2], the quadratic spline collocation method doesn't employ a complete set of orthogonal basis, but a set of piecewise second-order polynomials.

Though the subspace built on the quadratic basis functions is not complete, namely, the combination of the quadratic spline basis functions cannot represent all of the possible solutions in computation space, the QSC subspace may well represent the continuous solution with continuous first-order derivative in the second-order partial differential equations, for example, the vector Helmholtz wave equations.

The uniqueness and convergence property of the quadratic spline collocation interpolant for a second-order partial differential equation will be discussed in 3.2 when the specific basis functions are selected. QSC method is verified based on the proof of the uniqueness and the convergence property [3].



### **3.1.2 Promising Advantage of Piecewise Quadratic Spline Interpolant**

The eigen-function expansion usually adopts a complete set of functions that represent the global property of the computation window. The number of the orthogonal and complete basis functions for the eigen-function expansion is usually infinite and a probable truncation of the basis functions will be crucial in deciding the accuracy of the expansion. Usually the first dozens of basis functions will be reserved to approximate the exact solution to have a good representation the global property. If the exact solution is generally evenly and smoothly distributed throughout the entire computation domain, only a few terms of the global eigen-basis functions will be needed to gain a good result [2]. However, if the exact solution has tremendous changes or is generally centralized in a tiny region like the strongly guide waveguides we will discuss later, the global eigen-basis expansion will have to include as many basis functions as possible to well represent the exact solution, and thus dramatically lose its efficiency in simulation.

Piecewise quadratic spline basis functions or other piecewise spline basis functions, on the other hand, may have advantage in simulating the cases when the exact solution shows significant localized property. And in this sense, the piecewise spline basis functions will be more general for solving all of the problems under the assumption of its validity. To make this point much clearer, we can imagine the extreme case of piecewise localized basis functions – the set of localized delta

functions when the grid size becomes infinite small. Every point of the computation window can be represented exactly by the weights of the local delta function.

## 3.2 QSC Method for A Single PDE in 2D Space

### 3.2.1 Problem Description

We first consider quadratic spline collocation (QSC) method for solving a single linear second order partial differential equations (PDEs) in a two-dimensional rectangular region, which can be expressed as

$$\begin{aligned} \mathbf{L}u \equiv au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu = g \\ \text{in } \Omega \equiv (a_x, b_x) \times (c_y, d_y) \end{aligned} \quad (3.1)$$

subject to mixed boundary condtions

$$\mathbf{B}u \equiv \alpha u + \beta u_n = \gamma \text{ on } \partial\Omega \equiv \text{boundary of } \Omega \quad (3.2)$$

where  $u, a, b, c, d, e, f, g, \alpha, \beta, \gamma$  are functions of  $x$  and  $y$ ,  $u_n$  denotes the normal derivative of  $u$ .  $u_{xx}$  and  $u_{yy}$  denote the second-order derivative of  $u$  with respect to  $x$  and  $y$  respectively, and the subscripts of  $u_{xy}$ ,  $u_x$  and  $u_y$  hold similar derivative meaning.  $\Omega$  is the inner rectangular region, and  $\partial\Omega$  is the boundary.  $\mathbf{L}$  is the linear second order operator and  $\mathbf{B}$  is the boundary operator respectively.

### 3.2.2 QSC Mesh and Collocation Points

The first step of quadratic spline collocation method is the mesh generation and

the selection of collocation points. In our problem (3.1) and (3.2), define

$$\bar{\Omega} \equiv \Omega \cup \partial\Omega \equiv [a_x, b_x] \times [c_y, d_y] \quad (3.3)$$

as the total problem region and let

$$\Delta_x \equiv \{a_x = x_0 < x_1 < \dots < x_M = b_x\} \quad (3.4)$$

$$\Delta_y \equiv \{c_y < y_0 < y_1 < \dots < y_N = d_y\} \quad (3.5)$$

be uniform partitions of the intervals  $[a_x, b_x]$ ,  $[c_y, d_y]$  with mesh size  $h_x$ ,  $h_y$  respectively. Then we can define the induced grid partition of  $\bar{\Omega}$  as

$$\Delta \equiv \Delta_x \times \Delta_y \quad (3.6)$$

We denote the midpoints of  $\Delta_x$  and  $\Delta_y$  by  $\tau_i^x (i = 1, \dots, M)$  and  $\tau_j^y (j = 1, \dots, N)$  respectively. For convenience, we extend the notation to boundary points as  $\tau_0^x \equiv x_0$ ,  $\tau_{M+1}^x \equiv x_M$ ,  $\tau_0^y \equiv y_0$ , and  $\tau_{N+1}^y \equiv y_N$ . Thus, the set of collocation points of  $\bar{\Omega}$  can be defined as

$$\Gamma \equiv \left\{ (\tau_i^x, \tau_j^y), i = 0, \dots, M+1, j = 0, \dots, N+1 \right\} \quad (3.7)$$

To illustrate the induced grid partition and the collocation points, an example of  $5 \times 4$  grid is displayed in Figure 3.1.

### 3.2.3 One-dimensional Basis Functions

The second step is to formulate the localized quadratic basis functions, which will be used to form the quadratic spline interpolant of the true solution of our problem.

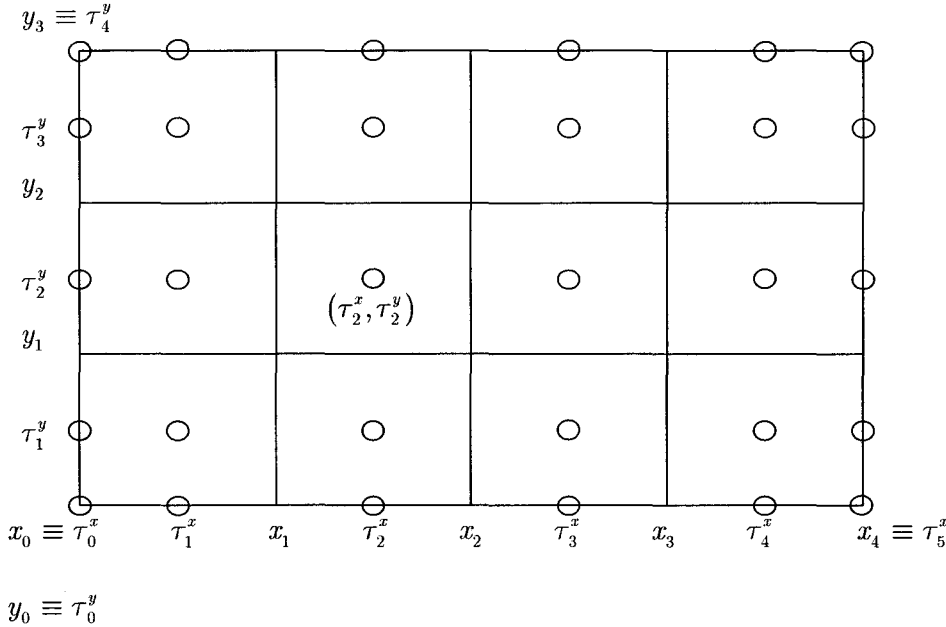


Figure 3.1. The collocation points for  $M = 4, N = 3$ .

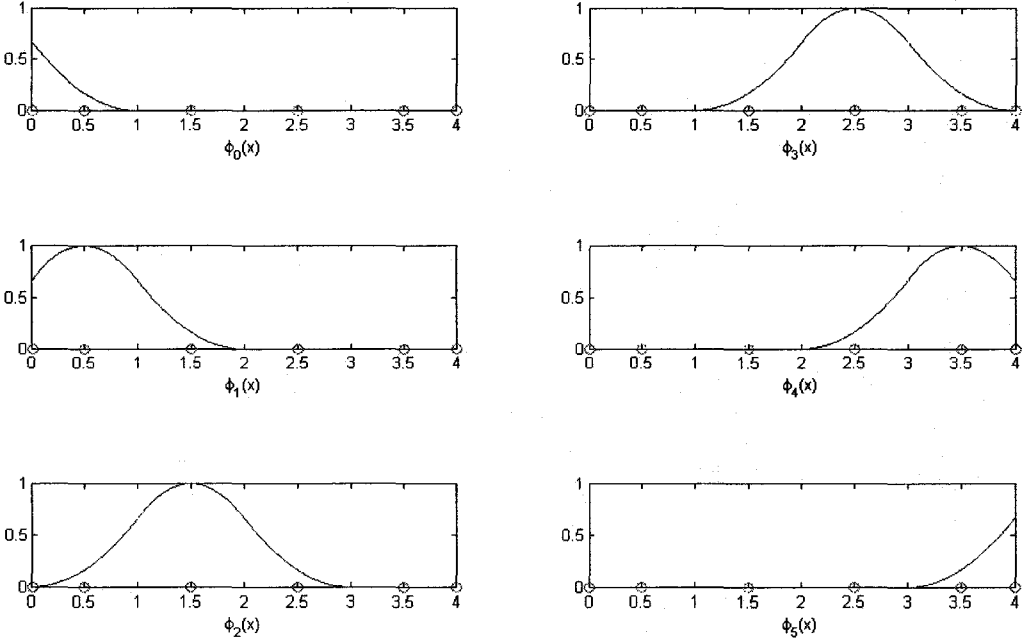


Figure 3.2. An example of general one-dimensional quadratic spline basis functions

Throughout, we denote by  $\mathbf{P}_{2,\Delta_x}$ ,  $\mathbf{P}_{2,\Delta_y}$  the one-dimensional space of piecewise quadratic polynomials with respect to partitions  $\Delta_x$ ,  $\Delta_y$  respectively, by  $S_{2,\Delta_x} \equiv \mathbf{P}_{2,\Delta_x} \cap \mathbb{C}^1([a_x, b_x])$ ,  $S_{2,\Delta_y} \equiv \mathbf{P}_{2,\Delta_y} \cap \mathbb{C}^1([c_y, d_y])$  the one-dimensional space of piecewise quadratic polynomials with continuous first-order derivative in  $\Delta_x$ ,  $\Delta_y$  respectively, by  $\mathbf{P}_{2,\Delta} \equiv \mathbf{P}_{2,\Delta_x} \times \mathbf{P}_{2,\Delta_y}$  the two-dimensional space of piecewise biquadratic polynomials with respect to partition  $\Delta$  of  $\bar{\Omega}$ , and by  $S_{2,\Delta} \equiv S_{2,\Delta_x} \times S_{2,\Delta_y}$  the two-dimensional space of piecewise biquadratic polynomials in  $\bar{\Omega}$  with continuous first-order derivative with respect to  $x$  and  $y$ .

The one-dimensional basis functions for  $S_{2,\Delta_x}$  can be chosen as [3]

$$\phi_i(x) \equiv \frac{2}{3} \psi \left( \frac{x - a_x}{h_x} - i + 2 \right) \quad i = 0, \dots, M + 1 \quad (3.8)$$

where the quadratic spline function  $\psi$  is defined by

$$\psi(x) \equiv \begin{cases} 0 & x > 3 \text{ or } x < 0 \\ x^2 & 0 \leq x \leq 1 \\ -3 + 6x - 2x^2 & 1 \leq x \leq 2 \\ 9 - 6x + x^2 & 2 \leq x \leq 3 \end{cases} \quad (3.9)$$

An example of one-dimensional basis functions for  $M = 4$ ,  $a_x = 0$ ,  $h_x = 1$  is shown in Figure 3.2. The basis functions  $\{\phi_j(y)\}_{j=0}^{N+1}$  for  $S_{2,\Delta_y}$  are constructed in a similar way.

### 3.2.4 Two-dimensional Basis Functions and Interpolant

The two-dimensional bi-quadratic basis functions for  $S_{2,\Delta}$  are constructed by forming the tensor product of the two set of one-dimensional basis functions

$$\{\phi_i(x)\}_{i=0}^{M+1} \text{ and } \{\phi_j(y)\}_{j=0}^{N+1} \quad [3].$$

Then based on the two-dimensional basis functions, we can construct the bi-quadratic spline interpolant, which will be represented as

$$S = \sum_{i=0}^{M+1} \sum_{j=0}^{N+1} \theta_{ij} \phi_i(x) \phi_j(y) \quad (3.10)$$

where  $S \in S_{2,\Delta}$  is the interpolant with continuous first-order derivative,  $\theta_{ij}$  is the unknown coefficient for bi-quadratic basis function  $\phi_i(x)\phi_j(y)$ .

### 3.2.5 Collocation Method

In quadratic spline collocation method, the way of determining the unknown coefficients  $\theta_{ij}$  is the collocation method, which requires the interpolant  $S$  to exactly match the true solution of the partial differential equation  $u$  at the collocation points  $(\tau_i^x, \tau_j^y)$ , namely,

$$S(\tau_i^x, \tau_j^y) = u(\tau_i^x, \tau_j^y) \quad 0 \leq i \leq M + 1, \quad 0 \leq j \leq N + 1. \quad (3.11)$$

The biquadratic spline collocation interpolant  $S$ , which is defined by (3.10) and satisfies (3.11), exists and is uniquely defined [3]. If  $u \in \mathbb{C}^1(\bar{\Omega})$ , the interpolation error  $e(x, y) = S(x, y) - u(x, y)$  satisfies [3]

$$\begin{aligned}
\|e\|_{\infty} &= O(h^3) \\
\|D_x e\|_{\infty} &= O(h^2) \\
\|D_y e\|_{\infty} &= O(h^2) \\
\|D_{xy} e\|_{\infty} &= O(h^2) \\
\|D_{xx} e\|_{\infty} &= O(h) \\
\|D_{yy} e\|_{\infty} &= O(h)
\end{aligned} \tag{3.12}$$

where  $h = \max(h_x, h_y)$ ,  $D_i$  means the first-order derivative with respect to axis  $i$ , and  $D_{ij}$  means the second-order derivative with respect to  $i$  and  $j$ .

### 3.2.6 Dirichlet and Neumann Boundary Conditions

Since usually only guided modes are studied and the simple zero boundary condition and perfect metal boundary condition are effective for solving the guided modes, we will first concentrate on QSC method for partial differential equation with Dirichlet boundary condition

$$u = 0 \quad \text{on} \quad \partial\Omega \tag{3.13}$$

or Neumann boundary condition

$$u_n = 0 \quad \text{on} \quad \partial\Omega \tag{3.14}$$

The perfect metal boundary condition is also applicable for the perfect matched layer absorbing boundary condition, which will be discussed in Chapter 6.

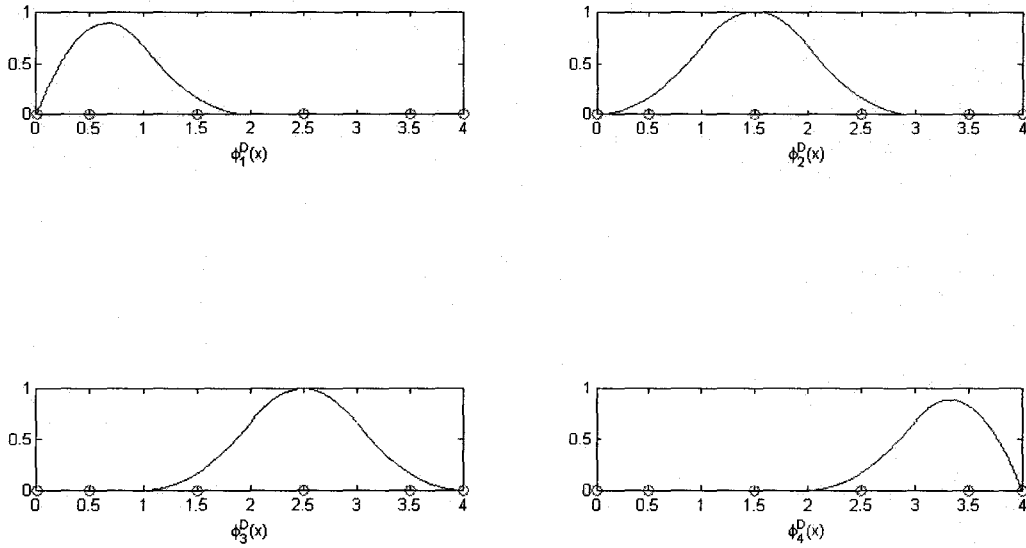


Figure 3.3. An example of one-dimensional basis functions with Dirichlet boundary condition

For the Dirichlet boundary condition, we introduce a special set of basis functions

$$\begin{aligned}
 \phi_1^D(x) &= \phi_1(x) - \phi_0(x) \\
 \phi_i^D(x) &= \phi_i(x), \quad i = 2, \dots, M-1 \\
 \phi_M^D(x) &= \phi_M(x) - \phi_{M+1}(x)
 \end{aligned} \tag{3.15}$$

which satisfies (3.13) directly. An example of one-dimensional basis functions with Dirichlet boundary condition for  $M = 4$ ,  $a_x = 0$ ,  $h_x = 1$  is shown in Figure 3.3.

For the Neumann boundary condition, similarly, we introduce



$$\begin{aligned}
\phi_1^N(x) &= \phi_1(x) + \phi_0(x) \\
\phi_i^N(x) &= \phi_i(x), \quad i = 2, \dots, M-1 \\
\phi_M^N(x) &= \phi_M(x) + \phi_{M+1}(x)
\end{aligned} \tag{3.16}$$

which satisfies (3.14) directly.

With Dirichlet boundary condition or Neumann boundary condition pre-configured, the collocation relation (3.11) can be simplified into

$$S(\tau_i^x, \tau_j^y) = u(\tau_i^x, \tau_j^y) \quad 1 \leq i \leq M, \quad 1 \leq j \leq N. \tag{3.17}$$

### 3.2.7 In Matrix Form

With the basis functions (3.15) or (3.16), and the simplified collocation relation (3.17), we can build the algebraic equations to solve for the coefficients  $\theta_{ij}$  in (3.10).

For a general partial differential equation with Dirichlet boundary condition, by using the Dirichlet basis functions (3.15), we can obtain from (3.17) the following Dirichlet discrete equations

$$\begin{aligned}
\mathbf{L} \left( \sum_{i=1}^M \sum_{j=1}^N \theta_{ij} \phi_i^D(x) \phi_j^D(y) \right) &= \mathbf{g} \quad 1 \leq i \leq M, \quad 1 \leq j \leq N \Rightarrow \\
\left\{ a \left( \frac{4}{3h_x^2} \mathbf{T}_{D,-2}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) + b \left( \frac{2}{3h_x} \mathbf{T}_{D,0}^M \right) \otimes \left( \frac{2}{3h_y} \mathbf{T}_{D,0}^N \right) + c \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{4}{3h_y^2} \mathbf{T}_{D,-2}^N \right) \right. \\
+ d \left( \frac{2}{3h_x} \mathbf{T}_{D,0}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) + e \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{2}{3h_y} \mathbf{T}_{D,0}^N \right) + f \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) \Big\} \bar{\theta} &= \bar{\mathbf{g}} \\
\Rightarrow \mathbf{K}_{DD} \bar{\theta} &\approx \bar{\mathbf{g}}
\end{aligned} \tag{3.18}$$

Where  $\otimes$  represents the tensor product of two matrices,  $\mathbf{T}_{D,-2}^M$ ,  $\mathbf{T}_{D,0}^M$ ,  $\mathbf{T}_{D,6}^M$ ,  $\mathbf{T}_{D,-2}^N$ ,  $\mathbf{T}_{D,0}^N$ , and  $\mathbf{T}_{D,6}^N$  are tridiagonal matrices. The superscripts  $M$  and  $N$  denote the order of the matrices. The matrices  $\mathbf{T}_{D,-2}^M$  and  $\mathbf{T}_{D,-2}^N$  are the second-order derivative

matrices for Dirichlet boundary condition, and are defined in terms of the generic matrix  $\mathbf{T}_{D,-2}$

$$\mathbf{T}_{D,-2} = \begin{bmatrix} -3 & 1 & & & \\ 1 & -2 & 1 & & \\ & \cdot & \cdot & \cdot & \\ & & 1 & -2 & 1 \\ & & & 1 & -3 \end{bmatrix} \quad (3.19)$$

Similarly, the matrices  $\mathbf{T}_{D,0}^M$  and  $\mathbf{T}_{D,0}^N$  are the first-order derivative matrices from the generic matrix  $\mathbf{T}_{D,0}$

$$\mathbf{T}_{D,0} = \begin{bmatrix} 1 & 1 & & & \\ -1 & 0 & 1 & & \\ & \cdot & \cdot & \cdot & \\ & & -1 & 0 & 1 \\ & & & -1 & -1 \end{bmatrix} \quad (3.20)$$

and the matrices  $\mathbf{T}_{D,6}^M$  and  $\mathbf{T}_{D,6}^N$  are the zero-order matrices from the generic matrix  $\mathbf{T}_{D,6}$

$$\mathbf{T}_{D,6} = \begin{bmatrix} 5 & 1 & & & \\ 1 & 6 & 1 & & \\ & \cdot & \cdot & \cdot & \\ & & 1 & 6 & 1 \\ & & & 1 & 5 \end{bmatrix} \quad (3.21)$$

In the same way, for a general partial differential equation with Neumann boundary condition, by using the Neumann basis functions (3.16), we can obtain from (3.17) the Neumann discrete equations

$$\begin{aligned}
& \mathbf{L} \left( \sum_{i=1}^M \sum_{j=1}^N \theta_{ij} \phi_i^N(x) \phi_j^N(y) \right) = \mathbf{g} \quad 1 \leq i \leq M, 1 \leq j \leq N \Rightarrow \\
& \left\{ a \left( \frac{4}{3h_x^2} \mathbf{T}_{N,-2}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right) + b \left( \frac{2}{3h_x} \mathbf{T}_{N,0}^M \right) \otimes \left( \frac{2}{3h_y} \mathbf{T}_{N,0}^N \right) + c \left( \frac{1}{6} \mathbf{T}_{N,6}^M \right) \otimes \left( \frac{4}{3h_y^2} \mathbf{T}_{N,-2}^N \right) \right. \\
& \left. + d \left( \frac{2}{3h_x} \mathbf{T}_{N,0}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right) + e \left( \frac{1}{6} \mathbf{T}_{N,6}^M \right) \otimes \left( \frac{2}{3h_y} \mathbf{T}_{N,0}^N \right) + f \left( \frac{1}{6} \mathbf{T}_{N,6}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right) \right\} \bar{\theta} = \bar{\mathbf{g}} \\
& \Rightarrow \mathbf{K}_{NN} \bar{\theta} \approx \bar{\mathbf{g}}
\end{aligned} \tag{3.22}$$

The similar generic matrices are the second-order derivative matrix  $\mathbf{T}_{N,-2}$

$$\mathbf{T}_{N,-2} = \begin{bmatrix} -1 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & & \cdot & \cdot & \cdot & \\ & & & 1 & -2 & 1 \\ & & & & 1 & 1 \end{bmatrix} \tag{3.23}$$

the first-order derivative matrix  $\mathbf{T}_{N,0}$

$$\mathbf{T}_{N,0} = \begin{bmatrix} -1 & 1 & & & & \\ -1 & 0 & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & & -1 & 0 & 1 \\ & & & & -1 & 1 \end{bmatrix} \tag{3.24}$$

and the zero order matrix  $\mathbf{T}_{N,6}$

$$\mathbf{T}_{N,6} = \begin{bmatrix} 7 & 1 & & & & \\ 1 & 6 & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & & 1 & 6 & 1 \\ & & & & 1 & 7 \end{bmatrix} \tag{3.25}$$

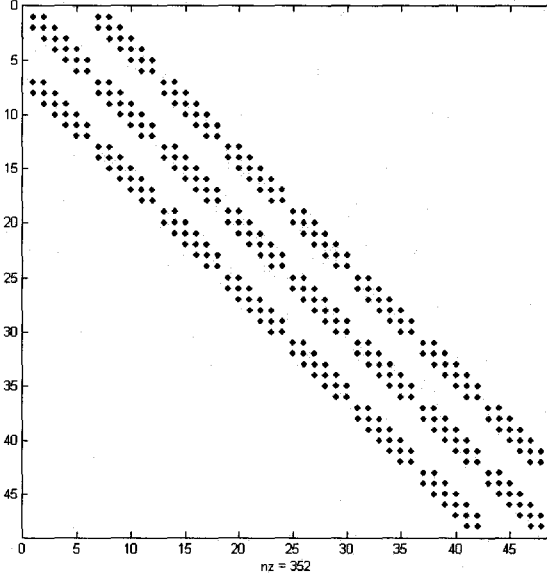


Figure 3.4. The sparse bi-quadratic matrix  $K$  for  $M=8$ ,  $N=6$ , the maximum number of non-zero element is 352.

For a general PDE subject to Dirichlet boundary condition on one direction and subject to Neumann boundary condition on the other direction, we just choose the right basis functions in the directions and choose the right generic matrices accordingly. For instance, if  $u$  is subject to Dirichlet and Neumann boundary condition in  $x$  and  $y$  respectively, the discrete equations will be expressed as

$$\begin{aligned}
 & \mathbf{L} \left( \sum_{i=1}^M \sum_{j=1}^N \theta_{ij} \tilde{\phi}_i^D(x) \tilde{\phi}_j^N(y) \right) = \mathbf{g} \quad 1 \leq i \leq M, \quad 1 \leq j \leq N \Rightarrow \\
 & \left\{ a \left( \frac{4}{3h_x^2} \mathbf{T}_{D,-2}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right) + b \left( \frac{2}{3h_x} \mathbf{T}_{D,0}^M \right) \otimes \left( \frac{2}{3h_y} \mathbf{T}_{N,0}^N \right) + c \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{4}{3h_y^2} \mathbf{T}_{N,-2}^N \right) \right. \\
 & \left. + d \left( \frac{2}{3h_x} \mathbf{T}_{D,0}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right) + e \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{2}{3h_y} \mathbf{T}_{N,0}^N \right) + f \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right) \right\} \bar{\theta} = \bar{g} \\
 & \Rightarrow \mathbf{K}_{DN} \bar{\theta} \approx \bar{g}
 \end{aligned} \tag{3.26}$$

Note that the single PDE matrix  $\mathbf{K}_{DD}$ ,  $\mathbf{K}_{NN}$  and  $\mathbf{K}_{DN}$  are  $MN \times MN$  square diagonal matrices with at most 9 non-zero elements per row. Usually  $MN$  are much larger than 9 so that  $\mathbf{K}_{DD}$ ,  $\mathbf{K}_{NN}$  and  $\mathbf{K}_{DN}$  are very sparse matrices. Figure 3.4 shows the K matrix for  $M=8$  and  $N=6$ .

### 3.3 QSC Method for A System of Two PDEs

We consider a system of two linear second-order PDEs in the same two-dimensional region as in (3.1)

$$\begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \quad (3.27)$$

where, for  $i = 1, 2$  and  $j = 1, 2$ ,  $\mathbf{L}_{ij}u \equiv a_{ij}u_{xx} + b_{ij}u_{xy} + c_{ij}u_{yy} + d_{ij}u_x + e_{ij}u_y + f_{ij}u$ ,  $a_{ij}, b_{ij}, c_{ij}, d_{ij}, e_{ij}, f_{ij}, g_i$  are given functions of  $x$  and  $y$ ,  $u$  and  $v$  are the unknown functions of  $x$  and  $y$ ,  $u$  and  $v$  are subject to either Dirichlet or Neumann boundary condition, and the induced grid partition is the same as before. The quadratic spline basis functions are built according to the boundary conditions and the bi-quadratic interpolants are used to approximate the true solutions as in (3.10)

$$S_u = \sum_{i=1}^M \sum_{j=1}^N \theta_{ij}^u \tilde{\phi}_i^{D \text{ or } N}(x) \tilde{\phi}_j^{D \text{ or } N}(y) \quad (3.28)$$

$$S_v = \sum_{i=1}^M \sum_{j=1}^N \theta_{ij}^v \tilde{\phi}_i^{D \text{ or } N}(x) \tilde{\phi}_j^{D \text{ or } N}(y) \quad (3.29)$$

After the interpolants are chosen, the collocation relations

$$\begin{aligned} S_u(\tau_i^x, \tau_j^y) &= u(\tau_i^x, \tau_j^y) \\ S_v(\tau_i^x, \tau_j^y) &= v(\tau_i^x, \tau_j^y) \end{aligned} \quad (3.30)$$

are applied into the original set of PDEs (3.27) to form the algebraic equations

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \bar{\theta}^u \\ \bar{\theta}^v \end{bmatrix} = \begin{bmatrix} \bar{g}_1 \\ \bar{g}_2 \end{bmatrix} \quad (3.31)$$

as in (3.26), where  $\mathbf{K}_{11}, \mathbf{K}_{12}, \mathbf{K}_{21}, \mathbf{K}_{22}$  are the  $MN \times MN$  known sparse matrices as  $\mathbf{K}_{DD}, \mathbf{K}_{NN}$  and  $\mathbf{K}_{DN}$  in the single PDE problem with at most 9 non-zero elements per row, and  $\bar{\theta}^u$  and  $\bar{\theta}^v$  are the unknown coefficients  $MN \times 1$  vectors.

## Chapter 4. QSC Scalar Mode Solver

In this chapter, QSC scalar mode solver is developed in MATLAB to solve the traditional weakly-guiding planar waveguides, and numerical examples of rib waveguides are calculated and compared with known results in the literature.

### 4.1 The Governing Equation and Boundary Condition

The governing partial differential equation for the QSC scalar mode solver is the scalar wave equation (2.37), and we can further express it as an eigen-system as

$$\phi_{xx} + \phi_{yy} + k_0^2 n(x,y)^2 \phi = \beta^2 \phi \quad (4.1)$$

where  $\phi_{xx}$  and  $\phi_{yy}$  are the second order derivative with respect to  $x$  and  $y$  respectively,  $k_0$  is a given constant,  $n(x,y)$  is a known refractive index profile throughout the problem region, and  $\beta$  is the propagation constant along  $z$  and the unknown eigen-values of the eigen-system.  $\phi$  represents  $E_z$  or  $H_z$ , which is continuous throughout the solution region  $\bar{\Omega}$  and has continuous first-order derivative  $\partial\phi/\partial x$  and  $\partial\phi/\partial y$ .

The zero boundary condition is applied at the computation window boundary  $\partial\Omega$  with the assumption that the computation window is sufficiently large and the field at the boundary is very close to zero. Note that, to calculate the leaky modes and radiation modes, other specific boundary conditions should be applied, such as perfectly matched layer absorbing boundary condition, which will be discussed in the

future work in Chapter 6. Since we only discuss guided mode solution in this chapter, we will use zero boundary condition.

With the boundary condition selected, the basis functions, according to Chapter 3, are chosen to be the Dirichlet basis functions and the true solution  $\phi$  is approximated by the bi-quadratic interpolant as

$$\phi = \sum_{i=1}^M \sum_{j=1}^N \theta_y \tilde{\phi}_i^D(x) \tilde{\phi}_j^D(y) + O(h^3) \equiv S^D + O(h^3) \quad (4.2)$$

and the interpolant has the collocation relation with the true solution

$$\phi(\tau_i^x, \tau_j^y) = S^D(\tau_i^x, \tau_j^y) \quad 1 \leq i \leq M, 1 \leq j \leq N \quad (4.3)$$

By applying (4.2) and (4.3) into (4.1), we obtain the algebraic equations at the collocation points and in matrix form, as in (3.18), it is

$$\left\{ \left( \frac{4}{3h_x^2} \mathbf{T}_{D,-2}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) + \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{4}{3h_y^2} \mathbf{T}_{D,-2}^N \right) + k_0^2 \mathbf{N} \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) \right\} \bar{\theta} = \beta^2 \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) \bar{\theta} \quad (4.4)$$

where  $\mathbf{N}$  is a  $MN \times MN$  refractive index matrix with repeated rows

$$\mathbf{N} = \begin{pmatrix} n(\tau_1^x, \tau_1^y) & n(\tau_1^x, \tau_2^y) & \dots & n(\tau_M^x, \tau_{N-1}^y) & n(\tau_M^x, \tau_N^y) \\ n(\tau_1^x, \tau_1^y) & n(\tau_1^x, \tau_2^y) & \dots & n(\tau_M^x, \tau_{N-1}^y) & n(\tau_M^x, \tau_N^y) \\ \dots & \dots & \dots & \dots & \dots \\ n(\tau_1^x, \tau_1^y) & n(\tau_1^x, \tau_2^y) & \dots & n(\tau_M^x, \tau_{N-1}^y) & n(\tau_M^x, \tau_N^y) \\ n(\tau_1^x, \tau_1^y) & n(\tau_1^x, \tau_2^y) & \dots & n(\tau_M^x, \tau_{N-1}^y) & n(\tau_M^x, \tau_N^y) \end{pmatrix} \quad (4.5)$$

and  $\bar{\theta}$  is the  $MN \times 1$  unknown coefficients vector or the eigen-vector



$$\bar{\theta} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \dots \\ \theta_{M(N-1)} \\ \theta_{MN} \end{bmatrix} \quad (4.6)$$

$\mathbf{T}_{D,-2}^M$ ,  $\mathbf{T}_{D,0}^M$ ,  $\mathbf{T}_{D,6}^M$ ,  $\mathbf{T}_{D,-2}^N$ ,  $\mathbf{T}_{D,0}^N$ , and  $\mathbf{T}_{D,6}^N$  are known tridiagonal matrices as defined in Chapter 3.

Thus, the QSC scalar mode solution with zero boundary condition becomes the eigen-system solution

$$\mathbf{A}\bar{\theta} = \beta^2 \mathbf{B}\bar{\theta} \quad (4.7)$$

## 4.2 The Simple MATLAB Program

The embedded sparse matrix functions and the great graphic display functions of MATLAB enable us a very simple, fast and effective QSC scalar mode solver for the traditional weakly-guiding planar waveguide. In this section, we will go through the MATLAB program of QSC scalar mode solver. An example of QSC scalar mode solver in MATLAB can be found in Appendix A. The MATLAB program can be easily transformed into other programming languages.

### 4.2.1 Inputs

The necessary inputs for solving (4.7) in MATLAB include the grid sizes  $h_x$  and  $h_y$ , the problem domain size  $M$  and  $N$ , the refractive index profile  $\mathbf{N}$ , the wave length  $\lambda$ , and the necessary constants, including the vacuum permittivity  $\epsilon_0$ , and the

vacuum permeability  $\mu_0$ .

## 4.2.2 Formulation

With the inputs  $M$  and  $N$ , the tridiagonal Dirichlet matrices  $\mathbf{T}_{D,-2}^M$ ,  $\mathbf{T}_{D,0}^M$ ,  $\mathbf{T}_{D,6}^M$ ,  $\mathbf{T}_{D,-2}^N$ ,  $\mathbf{T}_{D,0}^N$ , and  $\mathbf{T}_{D,6}^N$  are built. Since for practical problems,  $M$  and  $N$  are usually in the order of hundred, it will greatly save computer memory to store the very sparse Dirichlet matrices in Sparse Matrix Form by the MATLAB built-in function `sparse ()` in the beginning.

| MATLAB :: sparse ( )  |
|---|
| <pre>function [varargout] = sparse(varargin) % SPARSE Create sparse matrix. % S = SPARSE(X) converts a sparse or full matrix to sparse form by % squeezing out any zero elements.</pre> |

The tensor product of the sparse Dirichlet matrices can be formed by handily using the `kron ()` function.

| MATLAB :: kron ( )  |
|---|
| <pre>function K = kron(A,B) % KRON Kronecker tensor product. % KRON(X,Y) is the Kronecker tensor product of X and Y. % The result is a large matrix formed by taking all possible % products between the elements of X and those of Y. For % example, if X is 2 by 3, then KRON(X,Y) is % %      [ X(1,1)*Y  X(1,2)*Y  X(1,3)*Y %        X(2,1)*Y  X(2,2)*Y  X(2,3)*Y ] %</pre> |

```

% If either X or Y is sparse, only nonzero elements are multiplied
% in the computation, and the result is sparse.

% Previous versions by Paul Fackler, North Carolina State,
% and Jordan Rosenthal, Georgia Tech.
% Copyright 1984-2004 The MathWorks, Inc.
% $Revision: 5.17.4.1 $ $Date: 2004/01/24 09:22:29 $

```

When forming the  $\mathbf{A}$  matrix in (4.7), we will have a major matrix multiplication of the refractive index profile  $\mathbf{N}$  by  $(\mathbf{T}_{D,6}^M) \otimes (\mathbf{T}_{D,6}^N)$ . If no optimization is made here, it will exhaust the computer resource quickly when  $M$  and  $N$  increase, since both of the matrices are  $MN \times MN$  matrices. Directly use of the matrix multiplication for this step should be avoided. A simple and fast matrix multiplication is developed based on the fact that  $\mathbf{N}$  has repeated rows and  $(\mathbf{T}_{D,6}^M) \otimes (\mathbf{T}_{D,6}^N)$  is extremely sparse matrix.

The idea is to only multiply the non-zero elements in  $(\mathbf{T}_{D,6}^M) \otimes (\mathbf{T}_{D,6}^N)$  with the correct refractive index element in  $\mathbf{N}$ . The embedded function in MATLAB of finding the non-zero element in  $(\mathbf{T}_{D,6}^M) \otimes (\mathbf{T}_{D,6}^N)$  is `find ( )`. Then for each non-zero element, based on its row number  $m(i)$ , we use

$$\begin{aligned} \tau_i^x &= \text{Int}[m(i)/N] + 1 \\ \tau_j^y &= m(i) - (\tau_i^x - 1)N \end{aligned} \quad (4.8)$$

where `Int` means to get the integer of,  $\tau_i^x$  and  $\tau_j^y$  are the corresponding collocation point where the refractive index should be chosen. The MATLAB code block for doing this multiplication is

```

Code Block for Multiplication of  $\mathbf{N}$  and  $\left(\mathbf{T}_{D,6}^M\right) \otimes \left(\mathbf{T}_{D,6}^N\right)$ 
u = kron(T0xs_D,T0ys_D);
[M, N] = find(u);
lengthM = length(M);
for i = 1:lengthM
    ypos = mod(M(i),je);
    if (ypos == 0)
        ypos = je;
        xpos = M(i)/je;
    else
        xpos = (M(i)-ypos)/je+1;
    end
    u(M(i),N(i)) = cRI(xpos,ypos)*u(M(i),N(i));
end

```

The left side matrices  $\left(\frac{4}{3h_x^2} \mathbf{T}_{D,-2}^M\right) \otimes \left(\frac{1}{6} \mathbf{T}_{D,6}^N\right)$ ,  $\left(\frac{1}{6} \mathbf{T}_{D,6}^M\right) \otimes \left(\frac{4}{3h_y^2} \mathbf{T}_{D,-2}^N\right)$ , and  $k_0^2 \mathbf{N} \left(\frac{1}{6} \mathbf{T}_{D,6}^M\right) \otimes \left(\frac{1}{6} \mathbf{T}_{D,6}^N\right)$  then can be added up to form  $\mathbf{A}$  in eigen-system (4.7), and the right side matrix  $\left(\frac{1}{6} \mathbf{T}_{D,6}^M\right) \otimes \left(\frac{1}{6} \mathbf{T}_{D,6}^N\right)$  is  $\mathbf{B}$  in (4.7).

### 4.2.3 Eigen-System Solution

To solve the eigen-system (4.7), MABLAB offers the `eigs()` function, which uses ARPACK [7]

```

MATLAB :: eigs ( )
function varargout = eigs(varargin)
% EIGS Find a few eigenvalues and eigenvectors of a matrix using ARPACK.
% D = EIGS(A) returns a vector of A's 6 largest magnitude eigenvalues.
% A must be square and should be large and sparse.
%

```

```

% [V,D] = EIGS(A) returns a diagonal matrix D of A's 6 largest magnitude
% eigenvalues and a matrix V whose columns are the corresponding eigenvectors.
%
% [V,D,FLAG] = EIGS(A) also returns a convergence flag. If FLAG is 0
% then all the eigenvalues converged; otherwise not all converged.
%
% EIGS(A,B) solves the generalized eigenvalue problem  $A*V = B*V*D$ . B
must
% be symmetric (or Hermitian) positive definite and the same size as A.
% EIGS(A,[],...) indicates the standard eigenvalue problem  $A*V = V*D$ .

```

## 4.2.4 Outputs

To show the mode profile, `contour ( )`, `pcolor ( )`, etc. in MATLAB provide very good graphic display. Throughout our numerical method, the 5% contour is widely used.

```

MATLAB :: contour ( )
function [cout, hand] = contour(varargin)
%CONTOUR Contour plot.
% CONTOUR(Z) is a contour plot of matrix Z treating the values in Z
% as heights above a plane. A contour plot are the level curves
% of Z for some values V. The values V are chosen automatically.
% CONTOUR(X,Y,Z) X and Y specify the (x,y) coordinates of the
% surface as for SURF.
% CONTOUR(Z,N) and CONTOUR(X,Y,Z,N) draw N contour lines,
% overriding the automatic value.

```

## 4.3 Numerical Results and Discussion

To validate our QSC scalar mode solver, we calculate several typical weakly-

guiding planar dielectric waveguide which have various numerical results in the literature. Our results are compared to the known results in the literature.

We consider the three rib waveguides BT1, BT2, and BT3, which was analyzed by a finite difference scalar mode solver with zero boundary condition [1]. The typical semiconductor rib waveguide structure within the computation window for BT1, BT2, and BT3 is shown in Figure 4.1. The optical and geometrical parameters for the numerical computation shown in Figure 4.1 are listed in Table 4.1. The operating wavelength is  $1.55 \mu\text{m}$ . Note that, for a practical rib waveguide, the width of the substrate  $X_S$ , the thickness of the substrate  $Y_S$ , and the thickness of the cladding  $Y_C$  are much larger than the parameters in the table. Also Note that, since  $X_S$ ,  $Y_S$ , and  $Y_C$  can be chosen as needed under the assumption that the computation window is sufficient large to have nearly zero field values at the boundary, our values in Table 4.1 are slightly different from the original ones in [1].

The fundamental symmetric scalar modes are calculated in the QSC scalar mode solver. The modal indices, as well as the grid sizes  $h_x$  and  $h_y$ , are listed and compared with [1] in Table 4.2. The 5% percent contour of the fundamental

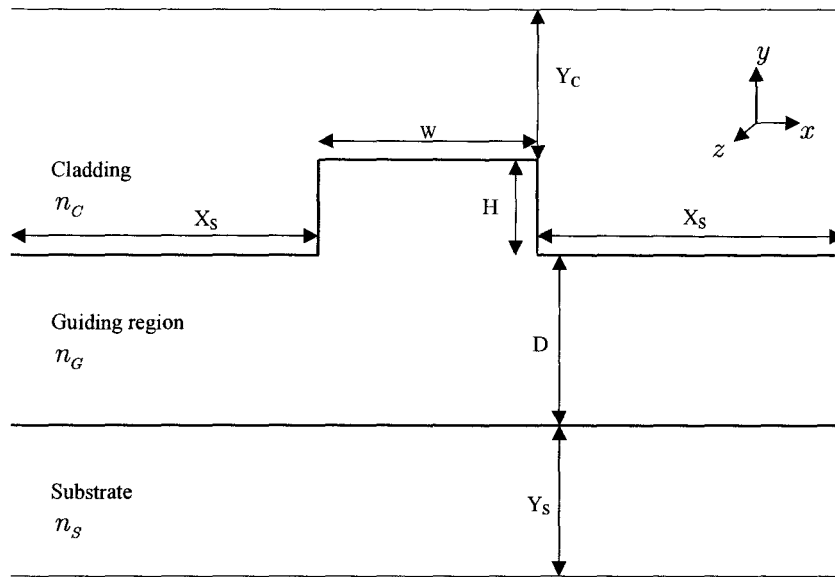


Figure 4.1. Typical semiconductor rib waveguide structures.

scalar mode profiles as well as the computation time<sup>1</sup> of our QSC scalar mode solver, which mainly consists of the time for the MATLAB eigs solver is recorded and listed in 4.3. The complete scalar mode solver in MATLAB for BT1 is listed in Appendix A.

As shown in Table 4.2 and Table 4.3, the QSC scalar mode solver can quickly calculate the fundamental scalar mode for rib waveguides BT2 and BT3 and gain a good match of the modal indices with the results from [1]. The modal index of BT1 is not matched so well. Due to the lack of a third party scalar mode solver, it is hard to tell whether QSC scalar solution is less accurate, or BT1 may be not a good example of scalar approximation.

<sup>1</sup> Computer: Intel Pentium 4 CPU 3.6GHz, 1.00GB of RAM; MATLAB: 7.0.4

|     | $n_G$ | $n_S$ | $n_C$ | W<br>( $\mu m$ ) | H<br>( $\mu m$ ) | D<br>( $\mu m$ ) | $X_S$<br>( $\mu m$ ) | $Y_S$<br>( $\mu m$ ) | $Y_C$<br>( $\mu m$ ) |
|-----|-------|-------|-------|------------------|------------------|------------------|----------------------|----------------------|----------------------|
| BT1 | 3.44  | 3.34  | 1.0   | 2                | 1.1              | 0.2              | 3.00                 | 5.00                 | 0.50                 |
| BT2 | 3.44  | 3.36  | 1.0   | 3                | 0.1              | 0.9              | 3.00                 | 5.00                 | 0.50                 |
| BT3 | 3.44  | 3.435 | 1.0   | 4                | 2.5              | 3.5              | 4.00                 | 7.50                 | 0.50                 |

Table 4.1. The semiconductor rib waveguide structures BT1, BT2, and BT3 from [1]

|     | QSC $n_{eff}$ | FD $n_{eff}$ | QSC $h_x, h_y$<br>$\mu m$ | FD $h_x, h_y$<br>$\mu m$ |
|-----|---------------|--------------|---------------------------|--------------------------|
| BT1 | 3.3815478     | 3.3901676    | 0.1000, 0.05              | 0.0952, 0.05             |
| BT2 | 3.3953972     | 3.3954801    | 0.0938, 0.05              | 0.0968, 0.05             |
| BT3 | 3.4366916     | 3.4367069    | 0.1000, 0.10              | 0.0976, 0.10             |

Table 4.2. Comparison of modal indices between QSC scalar mode solver and finite difference scalar mode solver in [1], with nearly identical grid sizes and parameters.



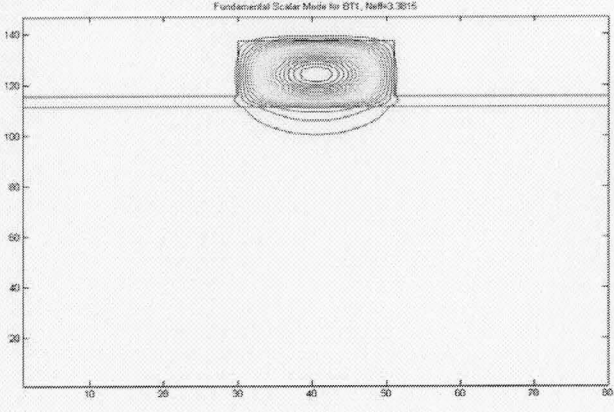
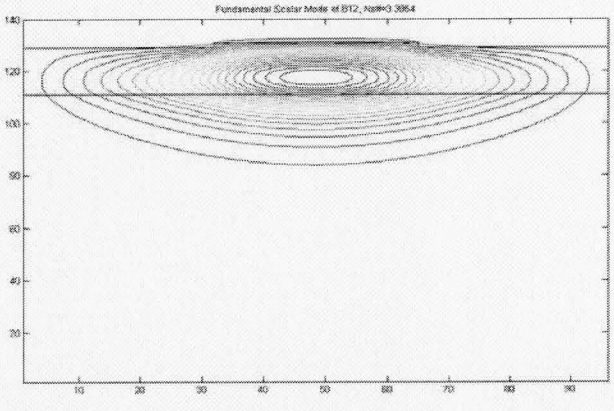
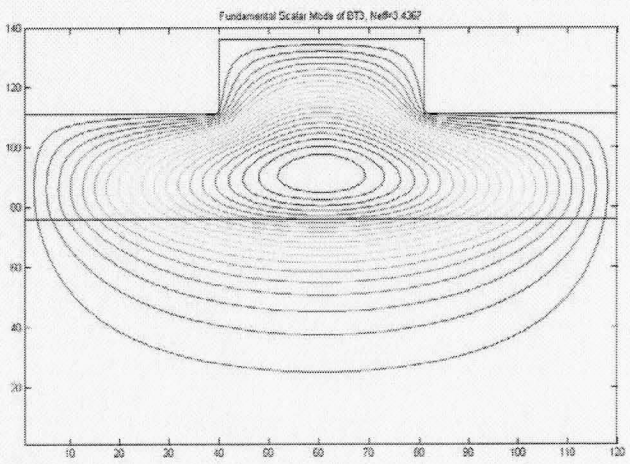
|     | 5% Contour of the Fundamental Scalar Mode  | Time           |
|-----|--|----------------|
| BT1 |    | 20.7<br>second |
| BT2 |   | 39.9<br>second |
| BT3 |  | 77.5<br>second |

Table 4.3. Fundamental Scalar Mode 5% Contour and Computation Time for BT1, BT2, and BT3

# Chapter 5. QSC Full-vectorial Mode Solver

The scalar mode solution in Chapter 4 doesn't tell us anything about the difference between polarized modes, which are very important in some optical devices. And the scalar mode solver is not applicable for strongly-guiding waveguides and its use is limited. A full-vectorial mode solver, theoretically, will solve any problem since the governing equations are exactly derived from Maxwell's equation. In this chapter, a full-vectorial mode solver based on QSC method is developed for arbitrary two-dimensional dielectric waveguides.

## 5.1 Governing Equations and Boundary Conditions

### 5.1.1 Choice of Governing Equations

Unlike the scalar mode solver, which has the same equation for electric field  $E_z$  and magnetic field  $H_z$ , the full-vectorial mode solutions have two sets of equations, the transverse electric field equations (2.23) and (2.24), and the transverse magnetic field equations (2.31) and (2.32).

From our discussion in Chapter 2, we know that the electric field and magnetic field should meet interface conditions at the dielectric interfaces. In practical cases, usually the permittivity is piecewise throughout the problem region and has sudden

changes at the interfaces. The sudden changes of the permittivity will result in the discontinuity of electric field. This discontinuity will greatly harm the accuracy of the bi-quadratic spline interpolant, since the QSC interpolant itself is continuous throughout the problem region. Thus, the electric field equations should be used carefully if the discontinuity plays a great role in the waveguide.

On the other hand, in most practical cases, the permeability behaves well and is constant throughout the waveguide cross-section, which means the magnetic field is continuous throughout and the bi-quadratic interpolant will be a good approximation.

Therefore, intuitively, it seems more reasonable to choose the magnetic equations as our governing equations for the QSC full-vectorial mode solver. However, since the equations for the electric fields must be the adjoint of the equations of the magnetic field, the QSC eigen-solution should be the same for both the transverse electric field equations and the transverse magnetic field equations [2, 14].

In this thesis, the transverse magnetic field equations are chosen as the governing equations for our QSC full-vectorial mode solution. The solution based the transverse electric field equations can be constructed in a similar way.

### **5.1.2 Governing Equations and Boundary Condition**

Our governing equations for QSC full-vectorial mode solver are selected as the magnetic vector wave equations (2.31) and (2.32), which can be rewritten as

$$\begin{aligned} (H_y)_{xx} + (H_y)_{yy} + [\ln n^{-2}(x, y)]_x (H_y)_x + n^2(x, y) k_0^2 (H_y) \\ - [\ln n^{-2}(x, y)]_x (H_x)_y = \beta^2 (H_y) \end{aligned} \quad (5.1)$$

$$\begin{aligned} (H_x)_{xx} + (H_x)_{yy} - [\ln n^{-2}(x, y)]_y (H_x)_y + n^2(x, y) k_0^2 (H_x) \\ + [\ln n^{-2}(x, y)]_y (H_y)_x = \beta^2 (H_x) \end{aligned} \quad (5.2)$$

where  $(H_y)_{xx}$  means the second-order derivative of  $H_y$  with respect to  $x$  and so forth,  $(H_x)_{xx}$  means the second-order derivative of  $H_x$  with respect to  $x$  and so forth,  $k_0$  is a given constant,  $n(x, y)$  is the known refractive index profile throughout the problem region,  $[\ln n^{-2}(x, y)]_x$  and  $[\ln n^{-2}(x, y)]_y$  are the first-order derivative of  $n(x, y)$  with respect to  $x$  and  $y$  respectively, and  $\beta$  is the propagation constant along  $z$  and the unknown eigen-values of the eigen- system.

The selection of boundary condition for our full-vectorial mode solver can be zero boundary condition or perfect metal boundary condition. We choose the perfect metal boundary condition, which gives us the choice of mixed boundary basis functions.

From the discussion in Chapter 2, for perfect metal boundary condition,  $H_y$  has Neumann boundary along  $x$  direction and has Dirichlet boundary along  $y$  direction, while  $H_x$  has Dirichlet boundary along  $x$  direction and Neumann boundary along  $y$  direction. Thus, we approximate  $H_y$  by the following bi-quadratic interpolant

$$H_y = \sum_{i=1}^M \sum_{j=1}^N \theta_{ij}^{H_y} \tilde{\phi}_i^N(x) \tilde{\phi}_j^D(y) + O(h^3) \equiv S^{H_y} + O(h^3) \quad (5.3)$$

and approximate  $H_x$  by the following bi-quadratic interpolant

$$H_x = \sum_{i=1}^M \sum_{j=1}^N \theta_{ij}^{H_x} \tilde{\phi}_i^D(x) \tilde{\phi}_j^N(y) + O(h^3) \equiv S^{H_x} + O(h^3) \quad (5.4)$$

The interpolant  $S^{H_y}$  has the collocation relation with the true solution  $H_y$  as

$$H_y(\tau_i^x, \tau_j^y) = S^{H_y}(\tau_i^x, \tau_j^y) \quad 1 \leq i \leq M, 1 \leq j \leq N \quad (5.5)$$

And the interpolant  $S^{H_x}$  has the similar collocation relation with  $H_x$  as

$$H_x(\tau_i^x, \tau_j^y) = S^{H_x}(\tau_i^x, \tau_j^y) \quad 1 \leq i \leq M, 1 \leq j \leq N \quad (5.6)$$

### 5.1.3 QSC Matrix Form

By applying (5.3), (5.4), (5.5), and (5.6) into the original magnetic wave equations

(5.1) and (5.2), we obtain the algebraic equations at the collocation points

$$\mathbf{Y}_1 \bar{\theta}^{H_y} + \mathbf{X}_1 \bar{\theta}^{H_x} = \mathbf{B}_1 \bar{\theta}^{H_y} + 0 \bar{\theta}^{H_x} \quad (5.7)$$

$$\mathbf{Y}_2 \bar{\theta}^{H_y} + \mathbf{X}_2 \bar{\theta}^{H_x} = 0 \bar{\theta}^{H_y} + \mathbf{B}_2 \bar{\theta}^{H_x} \quad (5.8)$$

where

$$\begin{aligned} \mathbf{Y}_1 &= \left( \frac{4}{3h_x^2} \mathbf{T}_{N,-2}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) + \left( \frac{1}{6} \mathbf{T}_{N,6}^M \right) \otimes \left( \frac{4}{3h_y^2} \mathbf{T}_{D,-2}^N \right) \\ &\quad + \mathbf{N}^x \left( \frac{2}{3h_x} \mathbf{T}_{N,0}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) + k_0^2 \mathbf{N} \left( \frac{1}{6} \mathbf{T}_{N,6}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) \\ \mathbf{X}_1 &= \left[ -\mathbf{N}^x \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{2}{3h_y} \mathbf{T}_{N,0}^N \right) \right] \\ \mathbf{B}_1 &= \beta^2 \left( \frac{1}{6} \mathbf{T}_{N,6}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{D,6}^N \right) \end{aligned} \quad (5.9)$$

$$\begin{aligned}
\mathbf{X}_2 &= \left( \frac{4}{3h_x^2} \mathbf{T}_{D,-2}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right) + \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{4}{3h_y^2} \mathbf{T}_{N,-2}^N \right) \\
&\quad - \mathbf{N}^y \left( \frac{2}{3h_x} \mathbf{T}_{D,0}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right) + k_0^2 \mathbf{N} \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right) \\
\mathbf{Y}_2 &= \left[ \mathbf{N}^y \left( \frac{1}{6} \mathbf{T}_{N,6}^M \right) \otimes \left( \frac{2}{3h_y} \mathbf{T}_{D,0}^N \right) \right] \\
\mathbf{B}_2 &= \beta^2 \left( \frac{1}{6} \mathbf{T}_{D,6}^M \right) \otimes \left( \frac{1}{6} \mathbf{T}_{N,6}^N \right)
\end{aligned} \tag{5.10}$$

and  $\mathbf{N}$  has the same form as in (4.5)

$$\mathbf{N} = \begin{bmatrix} n(\tau_1^x, \tau_1^y) & n(\tau_1^x, \tau_2^y) & \dots & n(\tau_M^x, \tau_{N-1}^y) & n(\tau_M^x, \tau_N^y) \\ n(\tau_1^x, \tau_1^y) & n(\tau_1^x, \tau_2^y) & \dots & n(\tau_M^x, \tau_{N-1}^y) & n(\tau_M^x, \tau_N^y) \\ \dots & \dots & \dots & \dots & \dots \\ n(\tau_1^x, \tau_1^y) & n(\tau_1^x, \tau_2^y) & \dots & n(\tau_M^x, \tau_{N-1}^y) & n(\tau_M^x, \tau_N^y) \\ n(\tau_1^x, \tau_1^y) & n(\tau_1^x, \tau_2^y) & \dots & n(\tau_M^x, \tau_{N-1}^y) & n(\tau_M^x, \tau_N^y) \end{bmatrix} \tag{5.11}$$

$\mathbf{N}^x$  and  $\mathbf{N}^y$  are the  $MN \times MN$  first-order derivative matrices of  $\ln n^{-2}(x, y)$

with respect to  $x$  and  $y$  respectively, and have the following forms

$$\mathbf{N}^x = \begin{bmatrix} \left[ \ln n^{-2}(\tau_1^x, \tau_1^y) \right]_x & \left[ \ln n^{-2}(\tau_1^x, \tau_2^y) \right]_x & \dots & \left[ \ln n(\tau_M^x, \tau_{N-1}^y) \right]_x & \left[ \ln n(\tau_M^x, \tau_N^y) \right]_x \\ \left[ \ln n^{-2}(\tau_1^x, \tau_1^y) \right]_x & \left[ \ln n^{-2}(\tau_1^x, \tau_2^y) \right]_x & \dots & \left[ \ln n(\tau_M^x, \tau_{N-1}^y) \right]_x & \left[ \ln n(\tau_M^x, \tau_N^y) \right]_x \\ \dots & \dots & \dots & \dots & \dots \\ \left[ \ln n^{-2}(\tau_1^x, \tau_1^y) \right]_x & \left[ \ln n^{-2}(\tau_1^x, \tau_2^y) \right]_x & \dots & \left[ \ln n(\tau_M^x, \tau_{N-1}^y) \right]_x & \left[ \ln n(\tau_M^x, \tau_N^y) \right]_x \\ \left[ \ln n^{-2}(\tau_1^x, \tau_1^y) \right]_x & \left[ \ln n^{-2}(\tau_1^x, \tau_2^y) \right]_x & \dots & \left[ \ln n(\tau_M^x, \tau_{N-1}^y) \right]_x & \left[ \ln n(\tau_M^x, \tau_N^y) \right]_x \end{bmatrix} \tag{5.12}$$

$$\mathbf{N}^y = \begin{bmatrix} \left[ \ln n^{-2}(\tau_1^x, \tau_1^y) \right]_y & \left[ \ln n^{-2}(\tau_1^x, \tau_2^y) \right]_y & \dots & \left[ \ln n(\tau_M^x, \tau_{N-1}^y) \right]_y & \left[ \ln n(\tau_M^x, \tau_N^y) \right]_y \\ \left[ \ln n^{-2}(\tau_1^x, \tau_1^y) \right]_y & \left[ \ln n^{-2}(\tau_1^x, \tau_2^y) \right]_y & \dots & \left[ \ln n(\tau_M^x, \tau_{N-1}^y) \right]_y & \left[ \ln n(\tau_M^x, \tau_N^y) \right]_y \\ \dots & \dots & \dots & \dots & \dots \\ \left[ \ln n^{-2}(\tau_1^x, \tau_1^y) \right]_y & \left[ \ln n^{-2}(\tau_1^x, \tau_2^y) \right]_y & \dots & \left[ \ln n(\tau_M^x, \tau_{N-1}^y) \right]_y & \left[ \ln n(\tau_M^x, \tau_N^y) \right]_y \\ \left[ \ln n^{-2}(\tau_1^x, \tau_1^y) \right]_y & \left[ \ln n^{-2}(\tau_1^x, \tau_2^y) \right]_y & \dots & \left[ \ln n(\tau_M^x, \tau_{N-1}^y) \right]_y & \left[ \ln n(\tau_M^x, \tau_N^y) \right]_y \end{bmatrix} \tag{5.13}$$

We further integrate (5.7) and (5.8) into a single matrix equation

$$\mathbf{A}\bar{\theta} = \beta^2 \mathbf{B}\bar{\theta} \quad (5.14)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{Y}_1 & \mathbf{X}_1 \\ \mathbf{Y}_2 & \mathbf{X}_2 \end{bmatrix} \quad (5.15)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & 0 \\ 0 & \mathbf{B}_2 \end{bmatrix} \quad (5.16)$$

$$\bar{\theta} = \begin{bmatrix} \bar{\theta}^{H_y} \\ \bar{\theta}^{H_x} \end{bmatrix} \quad (5.17)$$

The final matrix form (5.14) for our QSC full-vectorial mode solver is again very sparse.

## 5.2 The Simple MATLAB Program

Similar to our scalar mode solver, the embedded sparse matrix functions and the great graphic display functions of MATLAB enable us to develop a simple but effective QSC vector mode solver. In this section, we will go through the MATLAB program of QSC vector mode solver based on the governing matrix equation (5.14). An example of QSC full-vectorial mode solver in MATLAB for calculating the fundamental Quasi-TM mode and Quasi-TE mode of slot waveguide can be found in Appendix B.

### 5.2.1 Inputs

The necessary inputs for solving (5.14) in MATLAB includes the grid sizes  $h_x$  and  $h_y$ , the problem domain size  $M$  and  $N$ , the refractive index profile  $N$ , the

wave length  $\lambda$ , and the necessary constants, including the vacuum permittivity  $\epsilon_0$ , and the vacuum permeability  $\mu_0$ .

## 5.2.2 Formulation

### Step 1: Forming the one-dimensional basis matrices

With the inputs  $M$  and  $N$ , the necessary tridiagonal Dirichlet matrices and Neumann matrices in (5.9) and (5.10) are built. Like the scalar mode solver, we store the very sparse Dirichlet matrices in sparse matrix form by the MATLAB function `sparse()` in the beginning.

### Step 2: Forming the two-dimensional matrices

The tensor product of the sparse Dirichlet and Neumann matrices can be formed by handily using the `kron()` function as in our scalar mode solver.

### Step 3: Forming the coefficients matrices $N^x$ and $N^y$

These two matrices are very important for our success of vector mode solver, since they are the coefficients matrices for the coupling terms. One simple and effective wave of forming  $N^x$  is the central difference scheme correct to second order of  $h_x$

$$\left[ \ln n^{-2}(\tau_i^x, \tau_j^y) \right]_x = [\ln n^{-2}(\tau_{i+1}^x, \tau_j^y) - \ln n^{-2}(\tau_{i-1}^x, \tau_j^y)] / 2h_x + O(h_x^2) \quad (5.18)$$

or correct to fourth order of  $h_x$

$$\begin{aligned} \left[ \ln n^{-2}(\tau_i^x, \tau_j^y) \right]_x = & [-\ln n^{-2}(\tau_{i+2}^x, \tau_j^y) + 8 \ln n^{-2}(\tau_{i+1}^x, \tau_j^y) \\ & - 8 \ln n^{-2}(\tau_{i-1}^x, \tau_j^y) + \ln n^{-2}(\tau_{i-2}^x, \tau_j^y)] / 12h_x + O(h_x^4) \end{aligned} \quad (5.19)$$

Similarly, we can build  $N^y$ .



#### Step 4: Multiplication of coefficients matrices and two-dimensional matrices

When forming the  $\mathbf{A}$  matrix in (5.14), we will have several matrix multiplication, including  $\mathbf{N}^x(\mathbf{T}_{N,0}^M) \otimes (\mathbf{T}_{D,6}^N)$ ,  $\mathbf{N}(\mathbf{T}_{N,6}^M) \otimes (\mathbf{T}_{D,6}^N)$ ,  $\mathbf{N}^x(\mathbf{T}_{D,6}^M) \otimes (\mathbf{T}_{N,0}^N)$ ,  $\mathbf{N}^y(\mathbf{T}_{D,0}^M) \otimes (\mathbf{T}_{N,6}^N)$ ,  $\mathbf{N}(\mathbf{T}_{D,6}^M) \otimes (\mathbf{T}_{N,6}^N)$ ,  $\mathbf{N}^y(\mathbf{T}_{N,6}^M) \otimes (\mathbf{T}_{D,0}^N)$ . If no optimization is made here, it will exhaust the computer resource quickly when  $M$  and  $N$  increase, since both of the matrices are  $MN \times MN$  matrices. Directly use of the matrix multiplication for this step should be avoided. A simple and fast matrix multiplication is developed based on the fact that  $\mathbf{N}$  has repeated rows and  $(\mathbf{T}_{D,6}^M) \otimes (\mathbf{T}_{D,6}^N)$  is extremely sparse matrix. We use the same coding technique as in the scalar mode solver.

#### Step 5: Forming the final eigen-system

The left side matrices  $\left(\frac{4}{3h_x^2} \mathbf{T}_{D,-2}^M\right) \otimes \left(\frac{1}{6} \mathbf{T}_{D,6}^N\right)$ ,  $\left(\frac{1}{6} \mathbf{T}_{D,6}^M\right) \otimes \left(\frac{4}{3h_y^2} \mathbf{T}_{D,-2}^N\right)$ , and  $k_0^2 \mathbf{N} \left(\frac{1}{6} \mathbf{T}_{D,6}^M\right) \otimes \left(\frac{1}{6} \mathbf{T}_{D,6}^N\right)$  then can be added up to form  $\mathbf{A}$  in eigen-system (4.7), and the right side matrix  $\left(\frac{1}{6} \mathbf{T}_{D,6}^M\right) \otimes \left(\frac{1}{6} \mathbf{T}_{D,6}^N\right)$  is  $\mathbf{B}$  in (4.7).

### 5.2.3 Eigen-System Solution

The MATLAB `eigs()` can still be used for the full-vectorial QSC mode solver.

### 5.2.4 Calculate E Field from H Field

The governing equations for calculating the transverse electric fields from the

transverse magnetic field are (2.34) ~ (2.35). The central difference scheme as in forming the  $N^x$  and  $N^y$  is again applied.

### 5.2.5 Outputs

The same as the scalar mode solver, the mode profile can be shown by contour ( ) in MATLAB.

## 5.3 Numerical Results and Discussion

To validate the two-dimensional QSC full-vectorial mode solver, we calculate two typical examples and compare the QSC results to a commercial mode solver LUMERICAL<sup>2</sup>. The first example is the conventional ridge waveguide, which is weakly guided in the high index core region. The second example is the slot waveguide, which is strongly guided in the low index air region.

### 5.3.1 Ridge Waveguide

The structure of a typical ridge dielectric waveguide is shown in Figure 5.1. Compared with the rib waveguide structure of Figure 4.1, the ridge waveguide has one more layer as the core region with a higher refractive index  $n_G$ . The optical and geometrical parameters defined in Figure 5.1 for the simulation is listed in Table 5.1.

The numerical simulation parameters, including the grid sizes and boundary conditions, were set identical for both the LUMERICAL simulation and the QSC

---

<sup>2</sup> <http://www.lumerical.com/mode>

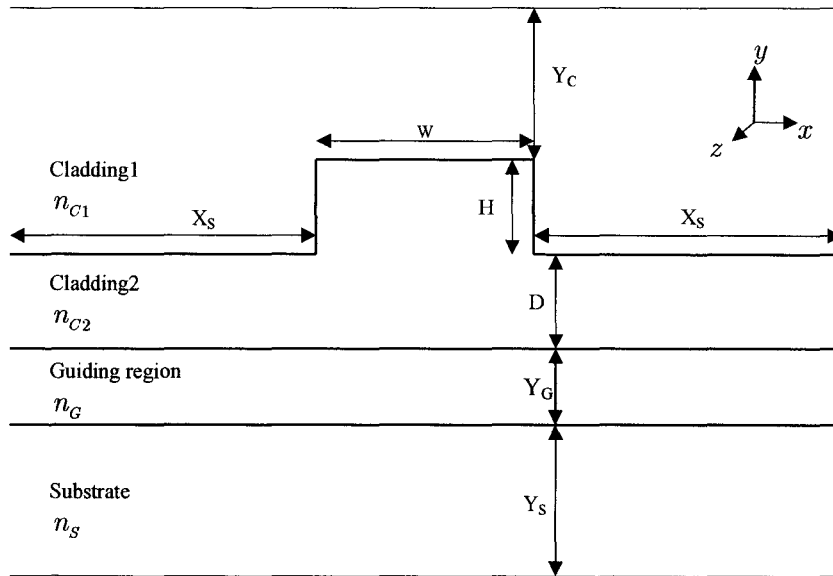


Figure 5.1. Typical structure of ridge waveguides

| $n_G$ | $n_S$ | $n_{C1}$ | $n_{C2}$ | W<br>$\mu m$ | H<br>$\mu m$ | D<br>$\mu m$ | $X_S$<br>$\mu m$ | $Y_C$<br>$\mu m$ | $Y_G$<br>$\mu m$ | $Y_S$<br>$\mu m$ |
|-------|-------|----------|----------|--------------|--------------|--------------|------------------|------------------|------------------|------------------|
| 3.45  | 3.19  | 1.0      | 3.19     | 3.0          | 0.5          | 0.1          | 4.0              | 0.5              | 0.2              | 2.5              |

Table 5.1. The optical and geometrical parameters of the ridge waveguide

| $h_x$         | $h_y$         | Wavelength  | MB  | First 4 Modes |
|---------------|---------------|-------------|-----|---------------|
| 0.125 $\mu m$ | 0.025 $\mu m$ | 1.5 $\mu m$ | Yes | Yes           |

Table 5.2. The simulation parameters for both LUMERICAL and QSC

simulation. The simulation parameters and condition are listed in Table 5.2.

The simulation results of the ridge waveguide are listed in Table 5.3. The modal indices and the 5% contour of the first four modes calculated by the QSC full-vectorial mode solver are displayed with the corresponding simulation results by

LUMERICAL in Table 5.3 and Table 5.4. The fundamental Quasi-TE mode and the fundamental Quasi-TM mode are perfectly matched, while the secondary Quasi-TE mode and the secondary Quasi-TM mode are well matched. The computation time for calculating the first four modes by our QSC full-vectorial mode solver is approximately 600 seconds while LUMERICAL mode solver took around 200 seconds to complete the simulation<sup>3</sup>.

---

<sup>3</sup> Computer: Intel Pentium 4 1.5 GHz; 524 MB of RAM

|                               | QSC Full-Vectorial Mode Solution | LUMERICAL Mode Solution |
|-------------------------------|----------------------------------|-------------------------|
| The fundamental Quasi-TE mode |                                  |                         |
| The fundamental Quasi-TM mode |                                  |                         |

Table 5.3. Fundamental quasi-TE and fundamental quasi-TM modes of ridge waveguide calculated by both QSC full-vectorial mode solver and LUMERICAL Mode Solution.

|                             | QSC Full-Vectorial Mode Solution | LUMERICAL Mode Solution |
|-----------------------------|----------------------------------|-------------------------|
| The secondary Quasi-TE mode |                                  |                         |
| The secondary Quasi-TM mode |                                  |                         |

Table 5.4. Secondary quasi-TE and secondary quasi-TM modes of ridge waveguide calculated by both QSC full-vectorial mode solver and LUMERICAL Mode Solution.

### 5.3.2 Slot Waveguide

Unlike the conventional optical waveguide, the slot waveguide, proposed and experimentally proven in [8], can confine the electric field of the fundamental

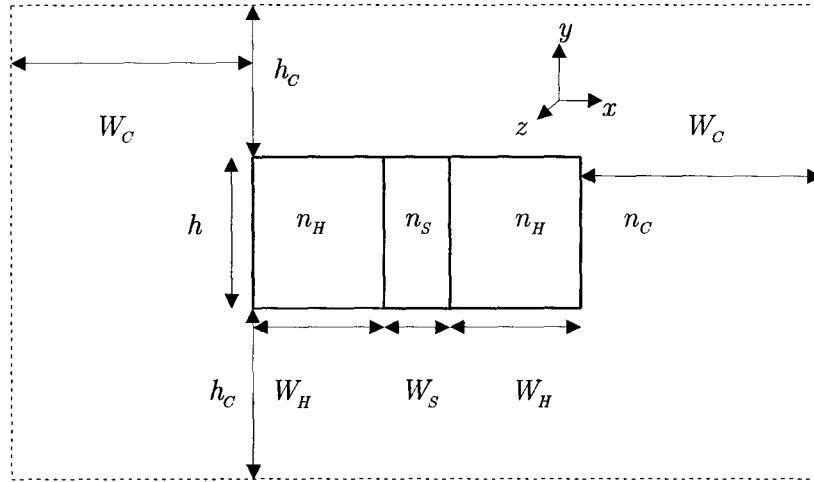


Figure 5.2. The computation window of slot waveguide

quasi-TE guided mode in the nanometer-wide low-index region. The cross-section of the slot waveguide is shown in Figure 5.2. The low index air slot is embedded between two high index silicon regions, and the whole structure is built in low index cladding background.

The optical and geometrical parameters for our simulation for QSC full-vectorial mode solver and LUMERICAL mode solution are identical and listed in Table 5.5, which are adopted from [8].

The numerical simulation parameters, including the grid size, boundary conditions, were set identical for both the LUMERICAL simulation and the QSC simulation. The simulation parameters and condition are listed in Table 5.6.

The QSC simulation results for wavelength  $1.6\ \mu m$  for the slot waveguide are listed in Table 5.7 and Table 5.8. The modal indices and the 5% contour of the first

| $W_S$   | $W_H$   | $W_C$   | $h$     | $h_C$   | $n_S$ | $n_H$ | $n_C$ |
|---------|---------|---------|---------|---------|-------|-------|-------|
| 0.10    | 0.22    | 1.00    | 0.25    | 1.00    | 1.00  | 3.48  | 1.46  |
| $\mu m$ | $\mu m$ | $\mu m$ | $\mu m$ | $\mu m$ |       |       |       |

Table 5.5. The optical and geometrical parameters of the slot waveguide

| $h_x$        | $h_y$         | Wavelength  | MB  | First 4 Modes |
|--------------|---------------|-------------|-----|---------------|
| 0.01 $\mu m$ | 0.025 $\mu m$ | 1.6 $\mu m$ | Yes | Yes           |

Table 5.6. The simulation parameters for both LUMERICAL and QSC

two modes calculated by the QSC full-vectorial mode solver are displayed in Table 5.7, and the corresponding simulation results by LUMERICAL are displayed in Table 5.8.

We can see the QSC and LUMERICAL results are fundamentally matched, though not perfectly. The electric field of the fundamental Quasi-TE mode is well confined in the air slot as in the [8]. However, while the result of the LUMERICAL shows that the electric field in the slot is almost evenly distributed, the result of the QSC shows aggregated electric field in the interfaces of silicon and air. And the modal indices from QSC and LUMERICAL are not so well matched as in the previous ridge waveguide.

### 5.3.3. Mesh Average Technique for QSC

To find out the causes of the simulation difference between QSC and



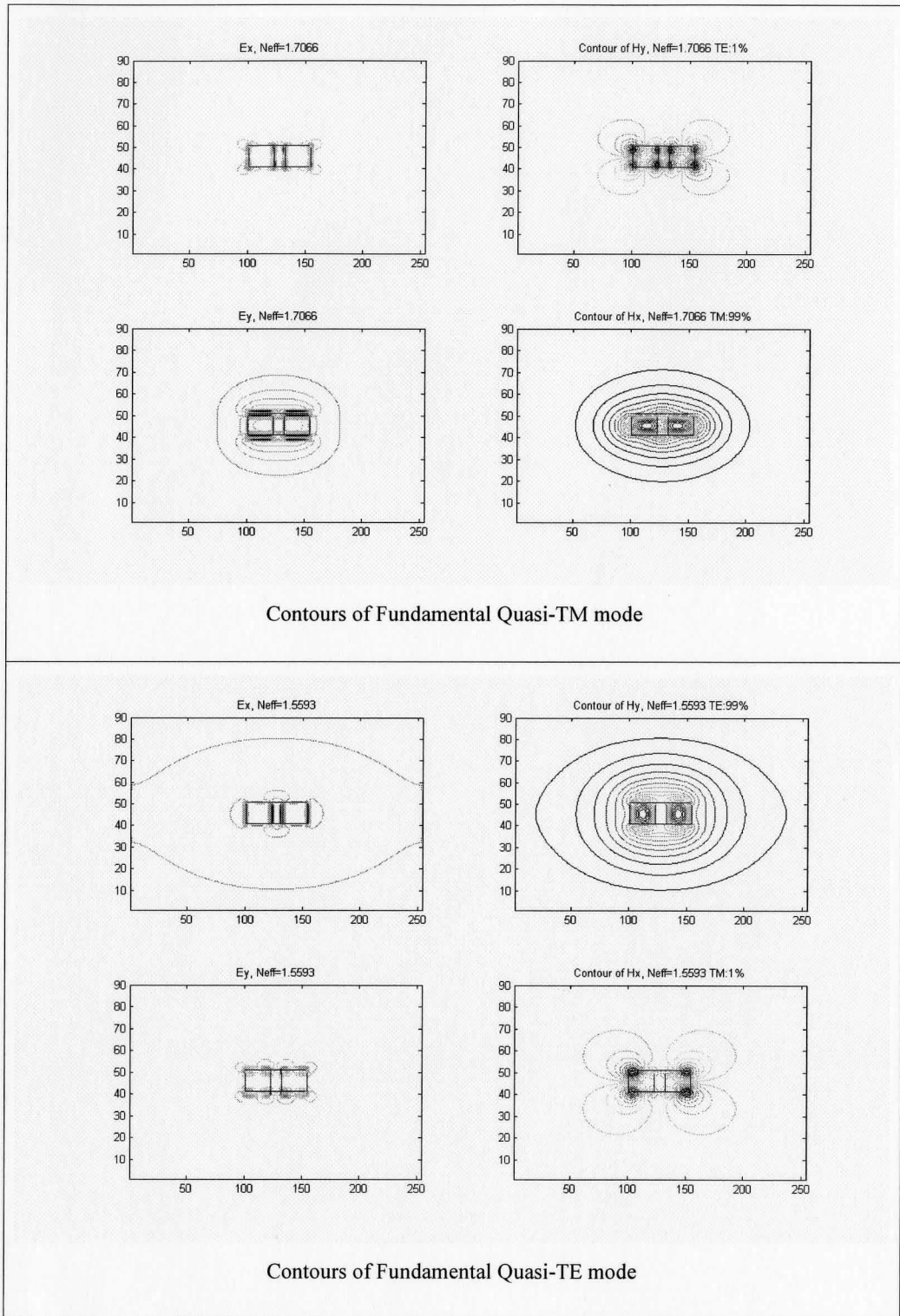


Table 5.7. QSC simulation results of slot waveguide

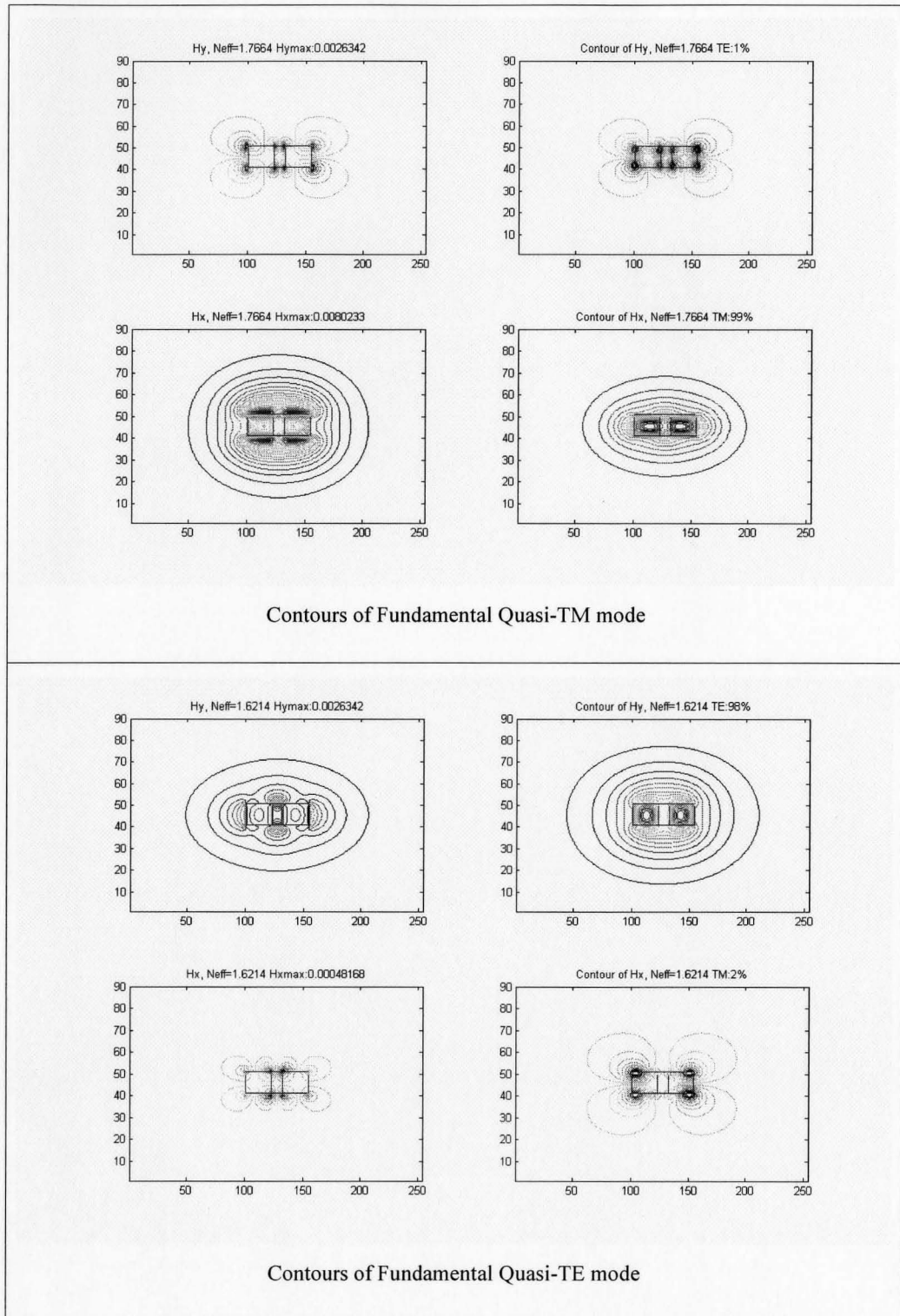


Table 5.8. LUMERICAL simulation results of slot waveguide

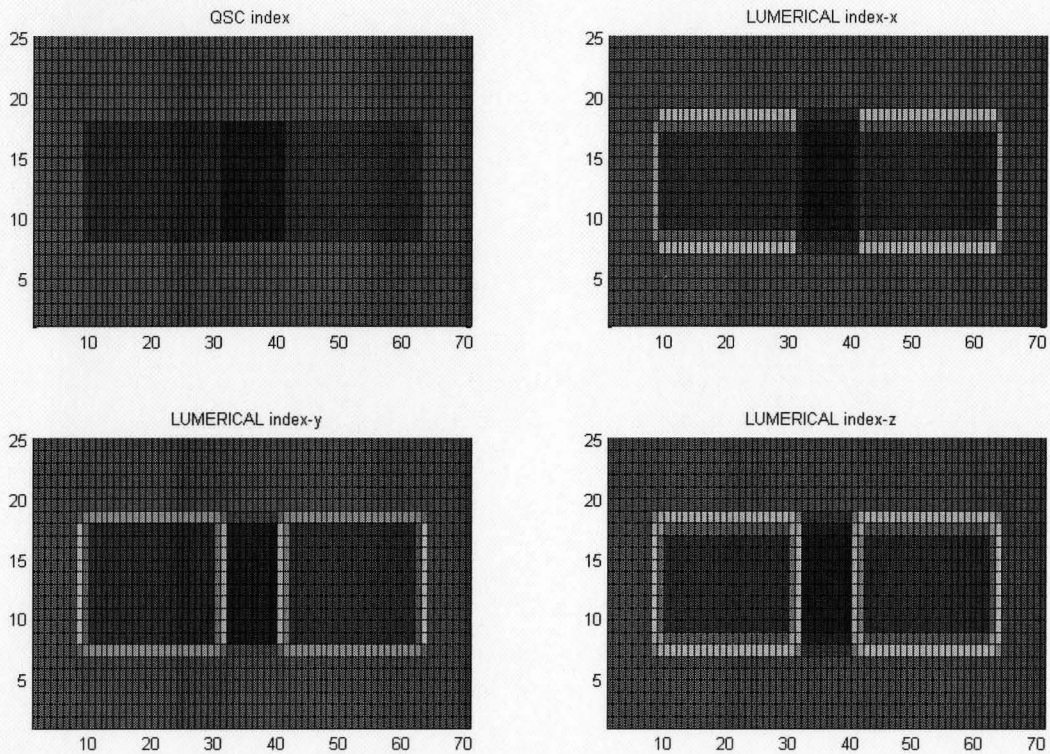


Figure 5.3. QSC original mesh and LUMERICAL's meshes index-x, index-y, index-z

LUMERICAL, and provide possible improvement, we investigated the meshes of LUMERICAL. In LUMERICAL mode solution, three different meshes are generated and named as index-x, index-y, and index-z, which are shown in Figure 5.3. We can see that the meshes used in LUMERICAL algorithms are not the original mesh as the QSC mesh in Figure 5.3, but are treated to have more smooth interfaces. To test whether the treated meshes will help reduce the difference between the simulation results of QSC and LUMERICAL, we apply the three meshes of LUMERICAL to QSC. The modal indices for the fundamental Quasi-TM and Quasi-TE modes are

|   | QSC<br>with index-x | QSC<br>with index-y | QSC<br>with index-z | QSC    | LUMERICAL |
|---|---------------------|---------------------|---------------------|--------|-----------|
| Effective Index<br>of fundamental<br>Quasi-TM | 1.7656              | 1.7340              | 1.7670              | 1.7066 | 1.7664    |
| Effective Index<br>of fundamental<br>Quasi-TM | 1.5695              | 1.6086              | 1.6142              | 1.5593 | 1.6214    |

Table 5.9. Modal indices of difference simulation conditions

listed and compared in Table 5.10. We can see that if we choose the LUMERICAL meshes, the differences become smaller, and if QSC uses the index-z mesh, the simulation results of QSC and LUMERICAL becomes close.

A closer look at the mesh and the collocation points of QSC will tell us why the LUMERICAL average meshes seems better for the simulation of the slot waveguide for QSC. Since slot waveguide is a strongly guided waveguide and heavily dependent on the high-index and low-index interfaces, we should create a mesh which should represent the real structure as close as possible. Though the original QSC mesh in Figure 5.3 absolutely reflects the refractive indices of the real structure, the interfaces of the real structure are not so well reflected in the program due to the locations of collocation points. Figure 5.4 shows why the collocation points in the original QSC mesh cannot well reflect the interface: there is a large undefined shadow area between the two collocation points and the simulation interface can be placed anywhere in the undefined area.

On the other hand, the LUMERICAL meshes sacrifice the absolute reflection of

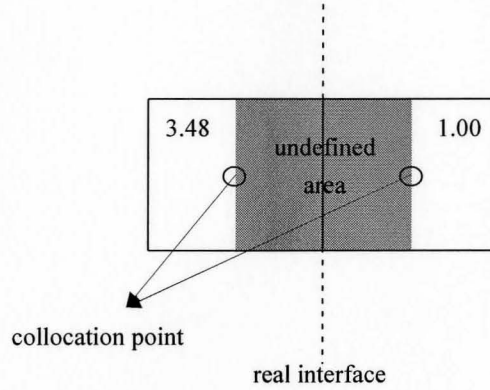


Figure 5.4. Collocation points in the original QSC mesh cannot well reflect the real interface

the refractive indices to better reflect the position of the interfaces by introducing average meshes index-x, index-y, index-z.

To address this problem and make the QSC full-vectorial mode solver more capable of simulating strongly guided waveguide, a simple scheme of mesh average technique is introduced. The idea is to set the refractive index at the collocation points the average of the surrounding four points half of the grid size away

$$RI_{average}(i, j) = \frac{1}{4} [RI(i + 0.5h_x, j) + RI(i - 0.5h_x, j) + RI(i, j + 0.5h_y) + RI(i, j - 0.5h_y)] \quad (5.20)$$

If the refractive indices in the surrounding area are not defined, a further simple average scheme can be applied as

$$\begin{aligned} RI(i \pm 0.5h_x, j) &= [RI(i \pm 1, j) + RI(i, j)]/2 \\ RI(i, j \pm 0.5h_y) &= [RI(i, j \pm 1) + RI(i, j)]/2 \end{aligned} \quad (5.21)$$

and (5.20) in this case can be rewritten as

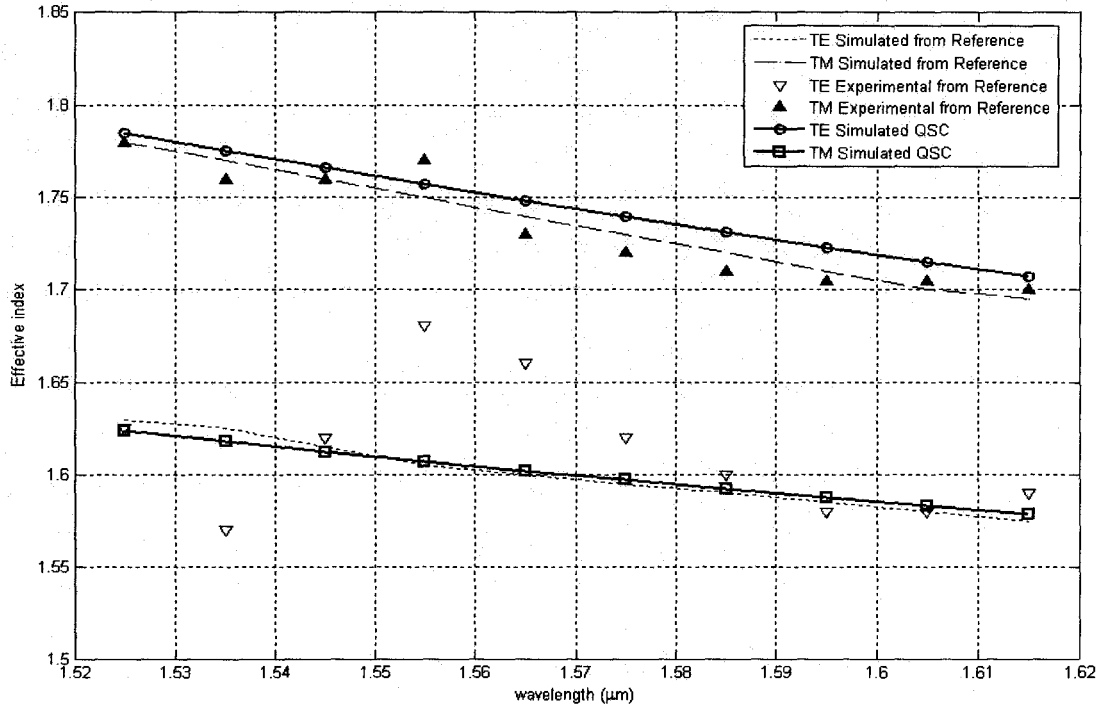


Figure 5.5. Effective index curve of slot waveguide

$$\begin{aligned}
 RI_{average}(i, j) = & [4 \cdot RI(i, j) + RI(i - 1, j) + RI(i + 1, j) \\
 & + RI(i, j - 1) + RI(i, j + 1)] / 8
 \end{aligned} \tag{5.22}$$

The effective index curve of slot waveguide by QSC full-vectorial mode solver with average mesh technique is plotted in comparison with the experimental and simulated results of [8]. The optical, geometrical, and simulation parameters are the same as before. The QSC curves shows great match to the curves adapted from [8].

# Chapter 6. Conclusion and Future Work

## 6.1 Thesis Summary

In this thesis, we apply quadratic spline collocation method to the mode solution of longitudinally invariant dielectric waveguide, which are the important fundamental building blocks for the complicated optical devices and integrated optoelectronic circuits.

Starting from the fundamental electromagnetic theory for the waveguide theory and the basic numerical ideas of quadratic spline collocation, we build both scalar mode solver and full-vectorial mode solver with Dirichlet or Neumann boundary conditions as simple MATLAB programs. Particular attention has been concentrated on the full-vectorial mode solver, since the scalar mode solver has limited use for weakly guided waveguides while the full-vectorial QSC mode solver is supposed to solve any kinds of waveguides.

The quadratic spline collocation method utilizes a set of localized quadratic basis functions to approximate the exact solution and uses the discrete equations at the collocation points to solve for the coefficients of the basis functions. Though such localized piecewise polynomial basis functions are not a complete and orthogonal basis functions of the simulation space, they meet the convergence property when the grid size goes smaller and they are essentially more suitable to represent those

solutions which will have strong local properties. The idea results in an eigen-value problem in sparse matrices, and the advantages of sparse matrix treatment and handy graphic display in MATLAB enable us to realize simple but effective MATLAB mode solvers.

Based on the scalar wave equation, the QSC scalar mode solver can quickly calculate the scalar modes of weakly guiding waveguides, such as conventional rib waveguides. The correctness and effectiveness of the QSC scalar mode solver are proven by simulating three rib waveguides and comparing the QSC results with the finite difference scalar mode solver results in [1].

The full-vectorial mode solver is based on the two wave equations of the transverse magnetic fields. To verify the solver, we apply the QSC full-vectorial mode solver to both weakly guiding ridge waveguide and the strongly guiding slot waveguide and compare the results with the simulation results of a commercial mode solver, LUMERICAL. Under the same simulation conditions, including the grid size, boundary conditions, etc., the QSC results of the ridge waveguide perfectly match the LUMERICAL results, while for the slot waveguide QSC results only fundamentally match LUMERICAL results. Mesh average technique, as in the LUMERICAL mode solution, is proposed to increase the accuracy of QSC full-vectorial mode solver simulating strongly guided waveguides, and is proven useful by calculating the effective index curve of the slot waveguide and comparing it with the original results from [9].



At appendices, simple MATLAB examples of the scalar mode solver and full-vectorial solver are listed for readers' further reference.

## **6.2 Future Work**

### **6.2.1 Calculating Leaky and Radiation Modes**

Though the simple QSC scalar mode solver and QSC full-vectorial we have developed are a handy tool for calculating the scalar or full-vectorial guided modes, the perfect metal boundary condition and zero boundary condition that we have applied will no longer be applicable for calculating leaky modes or radiation modes for an open-boundary problems. The reason is simple: neither the perfect metal nor the zero boundary conditions can properly reflect the actual mode profile at the boundaries. And since the leaky modes and radiation modes are becoming more and more important in some of the optical application, e. g., some active optical devices, it is demanding for a successful mode solver to possess the capability of dealing with all eigen modes in a waveguide, not just guided modes.

One of most popular and successful boundary conditions that have been applied in the literature for calculating the leaky and radiation modes is the perfectly matched layer (PML) absorbing boundary condition (ABC) [9, 10]. The basic idea of PML ABC is to truncate the computation window with numerical layers which will not reflect any of the incident waves. The idea was first introduced in FDTD application in [11] and later further investigated for other problems in [12].

In Cartesian coordinates, the generalized PML formulation replaces original coordinates by the following complex coordinates

$$\begin{aligned}x' &= \left(1 - j \frac{\sigma_x}{\omega \epsilon}\right)x \\y' &= \left(1 - j \frac{\sigma_y}{\omega \epsilon}\right)y \\z' &= \left(1 - j \frac{\sigma_z}{\omega \epsilon}\right)z\end{aligned}\tag{6.1}$$

where  $\sigma_x$ ,  $\sigma_y$ , and  $\sigma_z$  are pre-determined values for the perfectly matched layers [13]. With such simple replacements, the perfect non-reflectional interfaces at the perfect matched layers and the problem regions are realized [10].

Thus, to integrate the PML boundary for our scalar mode solver, we just change the original governing equation (2.37) by inserting (6.1) into

$$\frac{\partial}{\partial x'} \frac{\partial \phi}{\partial x'} + \frac{\partial}{\partial y'} \frac{\partial \phi}{\partial y'} + (k_0^2 n^2 - \beta^2) \phi = 0\tag{6.2}$$

and by using

$$\begin{aligned}\frac{\partial x}{\partial x'} &= \left(1 - j \frac{\sigma_x}{\omega \epsilon}\right)^{-1} \\ \frac{\partial y}{\partial y'} &= \left(1 - j \frac{\sigma_y}{\omega \epsilon}\right)^{-1}\end{aligned}\tag{6.3}$$

we obtain the governing equation for QSC scalar mode solver with PML ABC

$$\begin{aligned}\left(1 - j \frac{\sigma_x}{\omega \epsilon}\right)^{-1} \frac{\partial}{\partial x} \left[ \left(1 - j \frac{\sigma_x}{\omega \epsilon}\right)^{-1} \frac{\partial \phi}{\partial x} \right] + \left(1 - j \frac{\sigma_y}{\omega \epsilon}\right)^{-1} \frac{\partial}{\partial y} \left[ \left(1 - j \frac{\sigma_y}{\omega \epsilon}\right)^{-1} \frac{\partial \phi}{\partial y} \right] \\ + (k_0^2 n^2 - \beta^2) \phi = 0\end{aligned}\tag{6.4}$$

Besides changing the governing equation, we should include extra perfectly matched layers at the boundaries by introducing non-zero conductivity distribution and may use the perfect metal boundary for the outmost boundary.

And similarly, we can quickly integrate the PML ABC into our QSC full-vectorial mode solver.

Theoretically, the PML can be implemented as stated above. But due to shortage of time, not enough numerical examples have been carried out to fully verify the PML ABC in QSC for calculating leaky and radiation modes. Only a one-dimensional QSC scalar mode solution for an ARROW [9] waveguide is verified. Thus one of our future projects will be focused on developing high performance full-vectorial mode solver with PML ABC boundary conditions so that the mode solver will be capable of solving not only conventional guided modes but also leaky modes and radiation modes, which may have more and more important applications.

Here, we present the one dimensional scalar QSC mode solution of the leaky modes of the ARROW waveguide with PML ABC.

The governing equation is easily reduced from (6.4) by dropping variable  $y$

$$\left(1 - j \frac{\sigma_x}{\omega \varepsilon}\right)^{-1} \frac{\partial}{\partial x} \left[ \left(1 - j \frac{\sigma_x}{\omega \varepsilon}\right)^{-1} \frac{\partial \phi}{\partial x} \right] + k_0^2 n^2 \phi = \beta^2 \phi \quad (6.5)$$

The refractive index of the ARROW waveguide in [9] is shown in Figure 6.1. The conductivity distribution is also shown in Figure 6.1 and assumed as

$$\sigma(x) = \sigma_{\max} \left( \frac{|x - x_{PML}|}{|x_{PE} - x_{PML}|} \right)^2 \quad (6.6)$$

where  $\sigma_{\max}$  is a pre-determined value [13] and here is set at  $0.1 \Omega^{-1}(\mu m)^{-1}$ ,  $x_{PML}$  and  $x_{PE}$  are the starting position of the PML and the position of the ending perfect metal boundary, respectively. The PML boundary at both sides of the simulation

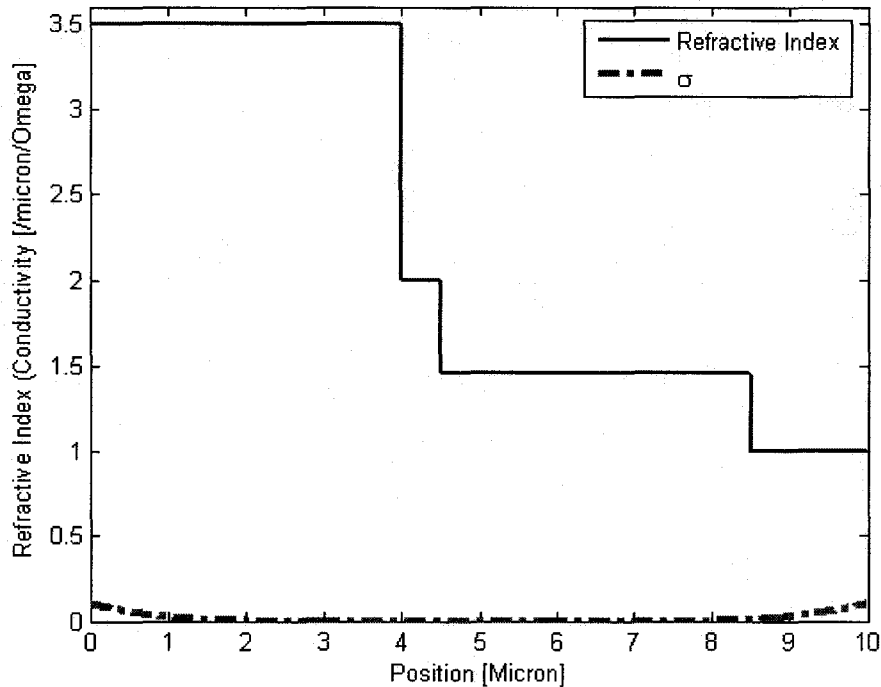


Figure 6.1. The refractive index profile and PML profile of a typical ARROW waveguide [9]

window has a width of  $2\ \mu\text{m}$ . The grid size of the QSC simulation is set at  $0.05\ \mu\text{m}$ . The wavelength is  $1.55\ \mu\text{m}$ . The total window size is  $10\ \mu\text{m}$ . In Figure 6.2, the real part of the lowest TE leaky mode is shown. To be illustrative, the simulation result without PML is also shown in Figure 6.2. We can see that with PML, the field profile in the guided region is no longer affected by the field at both ends of the computation window and thus has an accurate simulation result. The effective index is calculated as  $N = (1.448186133, -3.835 \times 10^{-4})$ , which is a good match with [9] and thus verify the one-dimensional QSC scalar mode solver with PML ABC.

Here, we cannot observe much difference between the results with PML or without PML; however, in Figure 6.3, we display the simulation results for the second

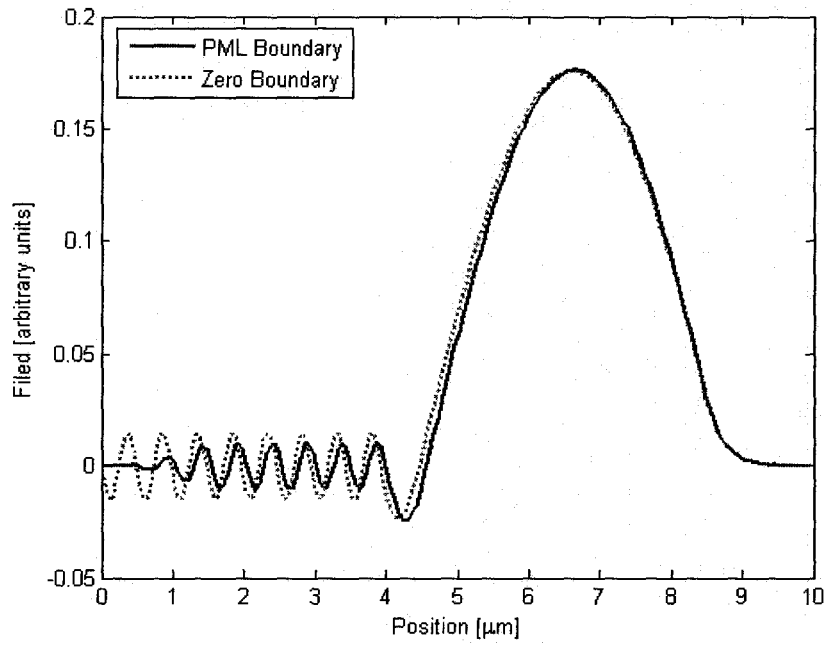


Figure 6.2. The lowest leaky mode of the ARROW waveguide

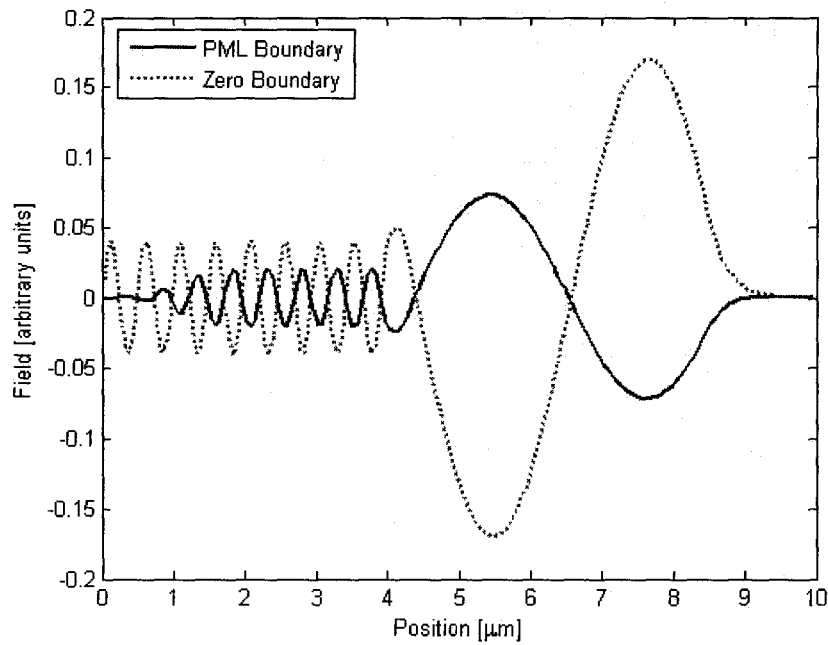


Figure 6.3. The second lowest leaky mode of the ARROW waveguide

order TE leaky mode and the difference becomes apparent: the reflected field for the simulation without PML totally change the field profile.

## 6.2.2 Higher Accuracy Solvers

The quadratic spline collocation method is not a widely-used numerical approach since its accuracy is not optimal [3]. We can see that though the QSC solution of the weakly guided ridge waveguide is quite good, the strongly guided slot waveguide is not so well calculated. One way to increase the accuracy is already introduced and implemented in Chapter 5 and called as the mesh averaging technique. However, the mesh averaging technique is not an essential solution to increase the accuracy.

Basically, there are two way of increasing the accuracy of the QSC mode solver. One general way is to use higher order polynomial basis functions. Instead of quadratic interpolant, we can choose cubic interpolant or even higher order interpolants. However, we will certainly sacrifice computation time since we can image that higher order interpolants will result in denser matrices.

The other way of increasing the accuracy is to use specific perturbation method [3]. Though it is not so obviously available to most of the potential QSC mode solver user, it is worth trying out optimal QSC method [3] if higher accuracy is required.

Another critical issue for the mode solvers in the literature is the ability of simulating the dielectric corners [14 – 16]. Since QSC mode solvers are also based on continuous basis functions, we should investigate the performance of QSC simulating

dielectric corners. No particular attention has been paid to this issue in our previous work and it should be one of our projects in the future.

### **6.2.3 Extending to Propagation Simulation**

Like the widely used finite difference method, the QSC method also has the simplicity of implementation without losing much accuracy. So, it should be handily applied to other equation system. One natural potential application is the beam propagation method based quadractic spline collocation method.

## Appendix A: QSC Scalar Mode Solver Example

```

%% 2DQSC_Scalar.m
%
% AUTHOR: Bin Xu, Photonics Research Laboratory, McMaster University
% Email: xub4@mcmaster.ca
% Created: 10.08.2006
% Modified: 10.09.2006
%
% This program is the main part of the optical waveguide scalar mode solver
% by standard quadratic spline collocation method, for BT1
%
% The Scalar Mode Solver Governing Equation is:
%  $U_{xx} + U_{yy} + n^2 k^2 H^2 = \beta^2 H^2$ 

clear all;
t0 = cputime;

%% Fundamental Parameters
cc = 3.0e8;
muo = 4.0*pi*1.0e-7;
epso = 1.0/(cc*cc*muo);
lambda = 1.55e-6; % wavelength = 1.5 [micron]
dy = 0.05e-6; % hy = 0.05 [micron]
dx = 0.1e-6; % hx = 0.1 [micron]

%% Grid parameters
iair = 3.0e-6/dx; % Xs
iw = 2.0e-6/dx; % W
ie = iair+iw+iair
jair = 0.525e-6/dy; % Yc
jcla = 1.1e-6/dy; % H
jcla2 = 0.2e-6/dy; % D
jcore = 0;
jsub = 5.0e-6/dy; % S
je = jair+jcla+jcla2+jcore+jsub+jair

```



```

%% Initialize the coefficient array and QSC coefficient matrices
% x direction
T0x = zeros(ic);
T1x = T0x;
T2x = T0x;
c1x = 2.0/(3.0*dx);
c2x = 4.0/(3.0*dx*dx);
% y direction
T0y = zeros(jc);
T1y = T0y;
T2y = T0y;
c1y = 2.0/(3.0*dy);
c2y = 4.0/(3.0*dy*dy);
% x direction coefficient matrix for Dirichlet Boundary Conditions
for i = 2:ie-1
    T0x(i,i) = 1.0;
    T0x(i,i-1) = 1.0/6;
    T0x(i,i+1) = 1.0/6;
    T1x(i,i-1) = -c1x;
    T1x(i,i+1) = c1x;
    T2x(i,i) = -2.0*c2x;
    T2x(i,i-1) = c2x;
    T2x(i,i+1) = c2x;
end
T0x(1,1) = 5.0/6.0;
T0x(1,2) = 1.0/6.0;
T0x(ie,ie) = T0x(1,1);
T0x(ie,ie-1) = T0x(1,2);
T1x(1,1) = c1x;
T1x(1,2) = c1x;
T1x(ie,ie) = -c1x;
T1x(ie,ie-1) = -c1x;
T2x(1,1) = -3.0*c2x;
T2x(1,2) = c2x;
T2x(ie,ie) = -3.0*c2x;
T2x(ie,ie-1) = c2x;
% y direction coefficient matrix for Dirichlet Boundary Conditions
for j = 2:je-1
    T0y(j,j) = 1.0;

```

```

T0y(j,j-1) = 1.0/6;
T0y(j,j+1) = 1.0/6;
T1y(j,j-1) = -c1y;
T1y(j,j+1) = c1y;
T2y(j,j) = -2.0*c2y;
T2y(j,j-1) = c2y;
T2y(j,j+1) = c2y;
end
T0y(1,1) = 5.0/6.0;
T0y(1,2) = 1.0/6.0;
T0y(je,je) = T0y(1,1);
T0y(je,je-1) = T0y(1,2);
T1y(1,1) = c1y;
T1y(1,2) = c1y;
T1y(je,je) = -c1y;
T1y(je,je-1) = -c1y;
T2y(1,1) = -3.0*c2y;
T2y(1,2) = c2y;
T2y(je,je) = -3.0*c2y;
T2y(je,je-1) = c2y;
% Save basis matrices as sparse matrices
T0xs_D = sparse(T0x);
T1xs_D = sparse(T1x);
T2xs_D = sparse(T2x);
T0ys_D = sparse(T0y);
T1ys_D = sparse(T1y);
T2ys_D = sparse(T2y);
clear T0x T1x T2x T0y T1y T2y; % Free memories

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set up the refractive index profile
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RI = ones(ie,je);
RI(iair+1:iair+iw,je+1-(jair+jcla):je+1-(jair+1)) =
3.44*RI(iair+1:iair+iw,je+1-(jair+jcla):je+1-(jair+1));
RI(:,je+1-(jair+jcla+jcla2):je+1-(jair+jcla+1)) = 3.44*RI(:,je+1-(jair+jcla+jcla2):je+1-(jair+jcla+1));
RI(:,je+1-(jair+jcla+jcla2+jcore):je+1-(jair+jcla+jcla2+1)) = ...
3.34*RI(:,je+1-(jair+jcla+jcla2+jcore):je+1-(jair+jcla+jcla2+1));
RI(:,je+1-je:je+1-(jair+jcla+jcla2+jcore+1)) = ...
3.34*RI(:,je+1-je:je+1-(jair+jcla+jcla2+jcore+1));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Form A matrix and B matrix:
% A = T2x@T0y + T0x@T2y + n^2*k^2*T0x@T0y
% B = T0x@T0y
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cRI = (2*pi*RI/lambda).^2;
u = kron(T0xs_D,T0ys_D);
[M, N] = find(u);
lengthM = length(M);
for i = 1:lengthM
    ypos = mod(M(i),je);
    if (ypos ==0)
        ypos = je;
        xpos = M(i)/je;
    else
        xpos = (M(i)-ypos)/je+1;
    end
    u(M(i),N(i)) = cRI(xpos,ypos)*u(M(i),N(i));
end
A = kron(T2xs_D,T0ys_D)+kron(T0xs_D,T2ys_D)+u;
B = kron(T0xs_D,T0ys_D);

t7 = cputime;           % Record Start Time

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve the generalized Eigenvalue Problem by MATLAB Imbedded Function
% eig(A,B)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[V,D] = eigs(A,B,2,'lr'); % solve for the largest two eigenvalues

t8 = cputime;           % Record End Time
Step8time = t8 - t7    % Matrix Solver Duration

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate the modal index and Plot the scalar mode profile
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Neff = diag(D,0).^5*lambda/2/pi
                                % Calculate the modal index

for ii = 1:2
    val = Neff(ii);
    NeffText = num2str(val);
    Hz = kron(T0xs_D,T0ys_D)*V(1:ie*je,ii);
    for i = 1:ie

```

```

    Hzz(i,:) = real(Hz((i-1)*je+1:i*je));
end
figure(ii+100)
subplot(2,1,1)
pcolor(Hzz')           % Pcolor Plot
line([1,iair],[je+1-(jair+jcla),je+1-(jair+jcla)])
line([iair,iair+iw+1],[je+1-jair,je+1-jair])
line([iair+iw+1,iair+iw+iair],[je+1-(jair+jcla),je+1-(jair+jcla)])
line([iair,iair],[je+1-(jair),je+1-(jair+jcla)])
line([iair+iw+1,iair+iw+1],[je+1-(jair),je+1-(jair+jcla)])
line([1,ie],[je+1-(jair+jcla+jcla2),je+1-(jair+jcla+jcla2)])
line([1,ie],[je+1-(jair+jcla+jcla2+jcore),je+1-(jair+jcla+jcla2+jcore)])
title(['Hz, Neff=',NeffText])
subplot(2,1,2)
contour(Hzz',20)      % 5% Contour Plot
line([1,iair],[je+1-(jair+jcla),je+1-(jair+jcla)])
line([iair,iair+iw+1],[je+1-jair,je+1-jair])
line([iair+iw+1,iair+iw+iair],[je+1-(jair+jcla),je+1-(jair+jcla)])
line([iair,iair],[je+1-(jair),je+1-(jair+jcla)])
line([iair+iw+1,iair+iw+1],[je+1-(jair),je+1-(jair+jcla)])
line([1,ie],[je+1-(jair+jcla+jcla2),je+1-(jair+jcla+jcla2)])
line([1,ie],[je+1-(jair+jcla+jcla2+jcore),je+1-(jair+jcla+jcla2+jcore)])
title(['Contour of Hz, Neff=',NeffText])
end

```

## Appendix B: QSC Vectorial Mode Solve Example

```

% Slot_Waveguide_Vectorial_Mode_Solver
%
% AUTHOR: Bin Xu, xub4@mcmaster.ca, Photonics Research Laboratory, McMaster
% University
% Created:          07.24.2006
% Last Modified:    10.29.2006
%
% This program is quadratic spline collocation full-vectorial mode solver
% for slot waveguide
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fundamental Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cc = 3.0e8;
muo = 4.0*pi*1.0e-7;
epso = 1.0/(cc*cc*muo);
lambda = 1.6e-6;      % wavelength = 1.6 [micron]
dy = 0.25e-6/10;     % hy = 0.025 [micron]
dx = 0.10e-6/10;     % hx = 0.010 [micron]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Grid parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
icla = 1.0e-6/dx;     % width of cladding in x direction
isi = 0.22e-6/dx;    % width of silicon region in x direction
islot = 0.1e-6/dx;   % width of the air slot in x direction
ie = icla+isi+islot+isi+icla
jcla = 1.0e-6/dy;    % height of cladding in y direction
jw = 0.25e-6/dy;    % height of the slot in y direction
je = jcla+jw+jcla
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialize the coefficient array and QSC coefficient matrices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% x direction
T0x = zeros(ie);
T1x = T0x;
T2x = T0x;
c1x = 2.0/(3.0*dx);
c2x = 4.0/(3.0*dx*dx);
% y direction

```

```

T0y = zeros(je);
T1y = T0y;
T2y = T0y;
c1y = 2.0/(3.0*dy);
c2y = 4.0/(3.0*dy*dy);
% x direction coefficient matrices for Dirichlet Boundary Conditions
for i = 2:ie-1
    T0x(i,i) = 1.0;
    T0x(i,i-1) = 1.0/6;
    T0x(i,i+1) = 1.0/6;
    T1x(i,i-1) = -c1x;
    T1x(i,i+1) = c1x;
    T2x(i,i) = -2.0*c2x;
    T2x(i,i-1) = c2x;
    T2x(i,i+1) = c2x;
end
T0x(1,1) = 5.0/6.0;
T0x(1,2) = 1.0/6.0;
T0x(ie,ie) = T0x(1,1);
T0x(ie,ie-1) = T0x(1,2);
T1x(1,1) = c1x;
T1x(1,2) = c1x;
T1x(ie,ie) = -c1x;
T1x(ie,ie-1) = -c1x;
T2x(1,1) = -3.0*c2x;
T2x(1,2) = c2x;
T2x(ie,ie) = -3.0*c2x;
T2x(ie,ie-1) = c2x;
% y direction coefficient matrices for Dirichlet Boundary Conditions
for j = 2:je-1
    T0y(j,j) = 1.0;
    T0y(j,j-1) = 1.0/6;
    T0y(j,j+1) = 1.0/6;
    T1y(j,j-1) = -c1y;
    T1y(j,j+1) = c1y;
    T2y(j,j) = -2.0*c2y;
    T2y(j,j-1) = c2y;
    T2y(j,j+1) = c2y;
end
T0y(1,1) = 5.0/6.0;
T0y(1,2) = 1.0/6.0;
T0y(je,je) = T0y(1,1);

```

```

T0y(je,je-1) = T0y(1,2);
T1y(1,1) = c1y;
T1y(1,2) = c1y;
T1y(je,je) = -c1y;
T1y(je,je-1) = -c1y;
T2y(1,1) = -3.0*c2y;
T2y(1,2) = c2y;
T2y(je,je) = -3.0*c2y;
T2y(je,je-1) = c2y;
T0xs_D = sparse(T0x);
T1xs_D = sparse(T1x);
T2xs_D = sparse(T2x);
T0ys_D = sparse(T0y);
T1ys_D = sparse(T1y);
T2ys_D = sparse(T2y);
% x direction coefficient matrices for Neumann Boundary Conditions
T0x(1,1) = 7.0/6.0;
T0x(1,2) = 1.0/6.0;
T0x(ie,ie) = T0x(1,1);
T0x(ie,ie-1) = T0x(1,2);
T1x(1,1) = -c1x;
T1x(1,2) = c1x;
T1x(ie,ie) = c1x;
T1x(ie,ie-1) = -c1x;
T2x(1,1) = -1.0*c2x;
T2x(1,2) = c2x;
T2x(ie,ie) = -1.0*c2x;
T2x(ie,ie-1) = c2x;
% y direction coefficient matrices for Neumann Boundary Conditions
T0y(1,1) = 7.0/6.0;
T0y(1,2) = 1.0/6.0;
T0y(je,je) = T0y(1,1);
T0y(je,je-1) = T0y(1,2);
T1y(1,1) = -c1y;
T1y(1,2) = c1y;
T1y(je,je) = c1y;
T1y(je,je-1) = -c1y;
T2y(1,1) = -1.0*c2y;
T2y(1,2) = c2y;
T2y(je,je) = -1.0*c2y;
T2y(je,je-1) = c2y;
T0xs_N = sparse(T0x);

```

```

T1xs_N = sparse(T1x);
T2xs_N = sparse(T2x);
T0ys_N = sparse(T0y);
T1ys_N = sparse(T1y);
T2ys_N = sparse(T2y);
clear T0x T1x T2x T0y T1y T2y; % Free memories
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set up the refractive index profile
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RII = 1.46 * ones(ie,je);
RIx = RII;
RIy = RII;
RI = RII;
RI_average=RII;
% The original mesh
RII(icla+1:icla+isi,jcla+1:jcla+jw) = 3.48/1.46*RII(icla+1:icla+isi,jcla+1:jcla+jw);
RII(icla+isi+1:icla+isi+islot,jcla+1:jcla+jw) = RII(icla+isi+1:icla+isi+islot,jcla+1:jcla+jw)/1.46;
RII(icla+isi+islot+1:icla+isi+islot+isi,jcla+1:jcla+jw) = ...
    3.48/1.46*RII(icla+isi+islot+1:icla+isi+islot+isi,jcla+1:jcla+jw);
% Mesh averaging
for i = 2:ie-1
    for j = 2:je-1
        RI_average(i,j) = (4*RII(i,j)+RII(i-1,j)+RII(i+1,j)+RII(i,j-1)+RII(i,j+1))/8;
    end
end
% Final mesh
for i = 1:ie
    for j = 1:je
        RI(i,j) = RI_average(i,j);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set up the log(RI)*ux and log(RI)*uy matrices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ogRI = log(RI);
logRIux = zeros(ie,je);
for i = 2:ie-1
    for j = 1:je
        logRIux(i,j) = (logRI(i+1,j)-logRI(i-1,j))/(2.0*dx);
    end
end
logRIuy = zeros(ie,je);

```



```

for i = 1:ie
    for j = 2:je-1
        logRIuy(i,j) = (logRI(i,j+1)-logRI(i,j-1))/(2.0*dy);
    end
end
end
%% Form Y1 matrix and X1 matrix :
% Y1 = T2x@T0y + T0x@T2y + n^2*k^2*T0x@T0y -[2*ln(n)*ux]*T1x@T0y
% Y1 = uxx + uyy + cRI*u - 2*logRIux*ux
%
% X1 = [2*ln(n)*ux]*T0x@T1y
% X1 = 2*logRIux*uy
%% Form n^2*k^2*T0x@T0y
cRI = (2*pi*RI/lambda).^2;
u = kron(T0xs_N,T0ys_D);
[M, N] = find(u);
lengthM = length(M);
for i = 1:lengthM
    ypos = mod(M(i),je);
    if (ypos==0)
        ypos = je;
        xpos = M(i)/je;
    else
        xpos = (M(i)-ypos)/je+1;
    end
    u(M(i),N(i)) = cRI(xpos,ypos)*u(M(i),N(i));
end
% Form T2x@T0y + T0x@T2y + n^2*k^2*T0x@T0y
Y1 = kron(T2xs_N,T0ys_D)+kron(T0xs_N,T2ys_D)+u;
clear u;
% Form [2*ln(n)*ux]*T1x@T0y
ux = kron(T1xs_N,T0ys_D);
[M, N] = find(ux);
lengthM = length(M);
for i = 1:lengthM
    ypos = mod(M(i),je);
    if (ypos==0)
        ypos = je;
        xpos = M(i)/je;
    else
        xpos = (M(i)-ypos)/je+1;
    end
end

```

```

end
ux(M(i),N(i)) = 2.0*logRlux(xpos,ypos)*ux(M(i),N(i));
end
% Form finl Y1
Y1 = Y1 - ux;
clear ux;
% Form X1
X1 = kron(T0xs_D,T1ys_N);
[M, N] = find(X1);
lengthM = length(M);
for i = 1:lengthM
ypos = mod(M(i),je);
if (ypos ==0)
ypos = je;
xpos = M(i)/je;
else
xpos = (M(i)-ypos)/je+1;
end
X1(M(i),N(i)) = 2.0*logRlux(xpos,ypos)*X1(M(i),N(i));
end
%%%%
% Form Y2 Matrix and X2 Matrix
% Y2 = [-2*ln(n)*uy]*T1x@T0y
% X2 = T2x@T0y + T0x@T2y + n^2*k^2*T0x@T0y + [2*ln(n)*uy]*T0x@T1y
%%%%
% Form Y2
Y2 = kron(T1xs_N,T0ys_D);
[M, N] = find(Y2);
lengthM = length(M);
for i = 1:lengthM
ypos = mod(M(i),je);
if (ypos ==0)
ypos = je;
xpos = M(i)/je;
else
xpos = (M(i)-ypos)/je+1;
end
Y2(M(i),N(i)) = +2.0*logRluy(xpos,ypos)*Y2(M(i),N(i));
end
% Form n^2*k^2*T0x@T0y
u = kron(T0xs_D,T0ys_N);
[M, N] = find(u);

```

```

lengthM = length(M);
for i = 1:lengthM
    ypos = mod(M(i),je);
    if (ypos ==0)
        ypos = je;
        xpos = M(i)/je;
    else
        xpos = (M(i)-ypos)/je+1;
    end
    u(M(i),N(i)) = cRI(xpos,ypos)*u(M(i),N(i));
end
% Form  $T2x @ T0y + T0x @ T2y + n^2 * k^2 * T0x @ T0y$ 
X2 = kron(T2xs_D,T0ys_N) + kron(T0xs_D,T2ys_N) + u;
clear u;
% Form  $[2 * \ln(n) * uy] * T0x @ T1y$ 
uy = kron(T0xs_D,T1ys_N);
[M, N] = find(uy);
lengthM = length(M);
for i = 1:lengthM
    ypos = mod(M(i),je);
    if (ypos ==0)
        ypos = je;
        xpos = M(i)/je;
    else
        xpos = (M(i)-ypos)/je+1;
    end
    uy(M(i),N(i)) = -2.0*logRluy(xpos,ypos)*uy(M(i),N(i));
end
% Form X2
X2 = X2 + uy;
clear uy;
% Form the right matrix B for the vectorial Helmholtz wave equation
uy = kron(T0xs_N,T0ys_D);
ux = kron(T0xs_D,T0ys_N);
zerox = sparse(zeros(je));
zeroy = sparse(zeros(je));
zeroxy = kron(zerox,zeroy);
B = [uy zeroxy; zeroxy ux];
clear uy ux zerox zeroy zeroxy;

```

```

% Form the left matrix A for the vectorial Helmholtz wave equation
%%
A = [ Y1 X1; Y2 X2];
clear Y1 X1 Y2 X2;
%%
% Solve the generalized Eigenvalue Problem by MATLAB Imbedded Function
% eig(A,B)
%%
t7 = cputime; % Record start time
[V,D] = eigs(A,B,2,'lr'); % Solve for the largest three eigenvalues
t8 = cputime; % Record end time
Step8time = t8 - t7 % Solver time
%%
% Find the eigenvalues and eigenvectors and plot the mode profile
%%
Neff = diag(D,0).^5*lambda/2/pi %Output the effective indices
for ii = 1:2
    val = Neff(ii);
    NeffText = num2str(val);
    % Calculate Ex and Ey
    Hy = kron(T0xs_N,T0ys_D)*V(1:ie*je,ii);
    Hyux = kron(T1xs_N,T0ys_D)*V(1:ie*je,ii);
    Hyuxx = kron(T2xs_N,T0ys_D)*V(1:ie*je,ii);
    Hyuxuy = kron(T1xs_N,T1ys_D)*V(1:ie*je,ii);
    Hyy = zeros(ie,je);
    Hyyux = zeros(ie,je);
    Hyyuxx = zeros(ie,je);
    Hyyuxuy = zeros(ie,je);
    for i = 1:ie
        Hyy(i,:) = Hy((i-1)*je+1:i*je);
        Hyyux(i,:) = Hyux((i-1)*je+1:i*je);
        Hyyuxx(i,:) = Hyyuxx((i-1)*je+1:i*je);
        Hyyuxuy(i,:) = Hyuxuy((i-1)*je+1:i*je);
    end
    Hx = kron(T0xs_D,T0ys_N)*V(ie*je+1:2*ie*je,ii);
    Hxuy = kron(T0xs_D,T1ys_N)*V(ie*je+1:2*ie*je,ii);
    Hxuxuy = kron(T1xs_D,T1ys_N)*V(ie*je+1:2*ie*je,ii);
    Hxuyy = kron(T0xs_D,T2ys_N)*V(ie*je+1:2*ie*je,ii);
    Hxx = zeros(ie,je);
    Hxxuy = zeros(ie,je);
    Hxxuxuy = zeros(ie,je);
    Hxxuyy = zeros(ie,je);

```

```

for i = 1:ie
    Hxx(i,:) = Hx((i-1)*je+1:i*je);
    Hxxuy(i,:) = Hxuy((i-1)*je+1:i*je);
    Hxxuxuy(i,:) = Hxuxuy((i-1)*je+1:i*je);
    Hxxuyy(i,:) = Hxuyy((i-1)*je+1:i*je);
end
DELTA = (Hxxuy - Hyyux)/(RI).^2;
DELTAux = zeros(ie,je);
for i = 2:ie-1
    for j = 1:je
        DELTAux(i,j) = (DELTA(i+1,j)-DELTA(i-1,j))/(2.0*dx);
    end
end
DELTAuy = zeros(ie,je);
for i=1:ie
    for j=2:je-1
        DELTAuy(i,j) = (DELTA(i,j+1)-DELTA(i,j-1))/(2*dy);
    end
end
Exx = Hyy*(2*pi/lambda)^2 - DELTAux;
Eyy = Hxx*(2*pi/lambda)^2 + DELTAuy;
% Simply calculate the TE and TM percent
HyyT = 0;
HxxT = 0;
for i = 1:ie
    for j = 1:je
        HyyT = HyyT + Hyy(i,j)^2;
        HxxT = HxxT + Hxx(i,j)^2;
    end
end
Hxmax = num2str(max(max(abs(Hxx))));
TEpercent = num2str(int32(100* HyyT/(HyyT+HxxT)));
TMpercent = num2str(int32(100* HxxT/(HyyT+HxxT)));
% Plot the 5% contours of Ex Ey Hy Hx
figure(ii+100)
subplot(2,2,1)
contour(real(Exx),',20)
line([icla+1,icla+1],[jcla+1,jcla+jw])
line([icla+isi+1,icla+isi+1],[jcla+1,jcla+jw])
line([icla+isi+islot+1,icla+isi+islot+1],[jcla+1,jcla+jw])
line([icla+isi+islot+isi+1,icla+isi+islot+isi+1],[jcla+1,jcla+jw])
line([icla+1,icla+isi+islot+isi],[jcla+1,jcla+1])

```

```

line([icla+1,icla+isi+islot+isi],[jcla+jw+1,jcla+jw+1])
title(['Ex, Neff=',NeffText])
subplot(2,2,2)
contour(real(Hyy),',20)
line([icla+1,icla+1],[jcla+1,jcla+jw])
line([icla+isi+1,icla+isi+1],[jcla+1,jcla+jw])
line([icla+isi+islot+1,icla+isi+islot+1],[jcla+1,jcla+jw])
line([icla+isi+islot+isi+1,icla+isi+islot+isi+1],[jcla+1,jcla+jw])
line([icla+1,icla+isi+islot+isi],[jcla+1,jcla+1])
line([icla+1,icla+isi+islot+isi],[jcla+jw+1,jcla+jw+1])
title(['Contour of Hy, Neff=',NeffText,' TE:',TEpercent,'%'])
subplot(2,2,3)
contour(real(Eyy),',20)
line([icla+1,icla+1],[jcla+1,jcla+jw])
line([icla+isi+1,icla+isi+1],[jcla+1,jcla+jw])
line([icla+isi+islot+1,icla+isi+islot+1],[jcla+1,jcla+jw])
line([icla+isi+islot+isi+1,icla+isi+islot+isi+1],[jcla+1,jcla+jw])
line([icla+1,icla+isi+islot+isi],[jcla+1,jcla+1])
line([icla+1,icla+isi+islot+isi],[jcla+jw+1,jcla+jw+1])
title(['Ey, Neff=',NeffText])
subplot(2,2,4)
contour(real(Hxx),',20)
line([icla+1,icla+1],[jcla+1,jcla+jw])
line([icla+isi+1,icla+isi+1],[jcla+1,jcla+jw])
line([icla+isi+islot+1,icla+isi+islot+1],[jcla+1,jcla+jw])
line([icla+isi+islot+isi+1,icla+isi+islot+isi+1],[jcla+1,jcla+jw])
line([icla+1,icla+isi+islot+isi],[jcla+1,jcla+1])
line([icla+1,icla+isi+islot+isi],[jcla+jw+1,jcla+jw+1])
title(['Contour of Hx, Neff=',NeffText,' TM:',TMpercent,'%'])
end

```

## Bibliography

- [1] W.-P. Huang, Ed., "Methods for modeling and simulation of guided-wave optoelectronic devices: Part I. Modes and couplings," *Progress in Electromagnetics Research*, EMW Publishing, Cambridge, MA, 1995
- [2] D. Marcuse, "Solution of the Vector Wave Equation for General Dielectric Waveguides by the Galerkin Method", *IEEE Journal of Quantum Electronics*, Vol. 28 No. 2, February 1992
- [3] C. C. Christara, "Quadratic spline collocation methods for elliptic partial differential equations", *BIT* 34 (1) (1994) 33-61
- [4] R. E. Collin, "Field theory of guided waves", IEEE/OUP, 1991
- [5] N. Thomas, "Finite-Difference Methods for the Modal Analysis of Dielectric Waveguides with Rectangular Corners", Ph.D. Thesis, University of Nottingham, 2004
- [6] M. S. Stern, "Semivectorial polarized finite difference method for optical waveguides with arbitrary index profiles", *IEE Proc. J*, 135, 56-63, 1988
- [7] Lehoucq, R.B., D.C. Sorensen, and C. Yang, "ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods", *SIAM Publications*, Philadelphia, 1998.
- [8] Xu, Q., V. R. Almeida, R. R. Panepucci, and M. Lipson, "Experimental demonstration of guiding and confining light in nanometer-size low-refractive-index material", *Optics Letters*, Vol. 29, No. 14, July 2004
- [9] Huang, W.-P., C. L. Xu, W. Lui, and K. Yokoyama, "The perfectly matched layer boundary condition for modal analysis of optical waveguides: leaky mode calculations", *IEEE Photonics Technology Letters*, Vol. 8, No. 5, May 1996
- [10] Huang, W.-P., C. L. Xu, W. Lui, and K. Yokoyama, "The perfectly matched layer (PML) boundary condition for the beam propagation method", *IEEE Photonics Technology Letters*, Vol. 8, No. 5, May 1996
- [11] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Computational Phys.*, Vol. 114, pp. 185 – 200, 1994
- [12] C. M. Rappaport, "Perfectly matched absorbing boundary conditions based on anisotropic

- lossy mapping of space”, *IEEE Microwave Guided Wave Lett.*, Vol. 5, No. 3, pp. 90 – 92, 1995
- [13] Katz, D. S., E. T. Thiele, and A. Taflove, “Validation and extension to three dimensions of the Berenger PML absorbing boundary condition for FD-TD meshes”, *IEEE Microwave Guided Wave Lett.*, Vol. 4, No. 8, pp. 268 – 270, 1994
- [14] C. Vassallo, “1993–1995 optical mode solvers,” *Opt. Quantum Electron.*, vol. 29, pp. 95–114, 1997
- [15] G. R. Hadley, “High-Accuracy Finite-Difference Equations for Dielectric Waveguide Analysis II: Dielectric Corners”, *Journal of Lightwave Technology*, Vol. 20, No. 7, July 2002
- [16] A. S. Sudbo, “Why Are Accurate Computations of Mode Fields in Rectangular Dielectric Waveguides Difficult?” *Journal of Lightwave Technology*, Vol. 10, No. 4, April 1992
- [17] G. R. Hadley, “High-Accuracy Finite-Difference Equations for Dielectric Waveguide Analysis I: Uniform Regions and Dielectric Interfaces”, *Journal of Lightwave Technology*, Vol. 20, No. 7, July 2002
- [18] S. M. Saard, “Review of Numerical Methods for the Analysis of Arbitrarily-Shaped Microwave and Optical Dielectric Waveguides”, *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-33, No. 10, October 1985