

A Finite-Element Mode Solver for Microcavity and Step-Index Fiber

by

Jiawen Li

B.Eng., Beihang University, Beijing, 2018

A Project Report in Partial Fulfillment of the
Requirements for the Degree of

Master of Engineering

in the Department of Electrical and Computer Engineering

© Jiawen Li, 2021

University of Victoria

All rights reserved. This report may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.

A Finite-Element Mode Solver for Microcavity and Step-Index Fiber

by

Jiawen Li

B.Eng., Beihang University, Beijing, 2018

Supervisory Committee

Dr. Tao Lu, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Wusheng Lu, Departmental Member

(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Tao Lu, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Wusheng Lu, Departmental Member

(Department of Electrical and Computer Engineering)

ABSTRACT

Simple Finite Elements in python (Sfepy) is an open-source framework which is dedicated to solving partial differential equation (PDE) problems with the help of vectorized operation in Numpy package. And Gmsh is an open-source 3D mesh generator combining a build-in CAD engine which provide a speedy, light and easy implementation meshing tool.

I build a finite element mode solver based on Sfepy and Gmsh for fiber mode and whispering gallery mode (WGM) analysis. In this report, we will simulate different type of step-index fiber to research their power contribution in terms of fiber's core radius, numerical aperture and V number, and we will also simulate a microcavity to solve WGM in terms of its resonance wavelength, quality factor and electromagnetic distribution.

Keywords: Finite elements, Sfepy, fiber mode, whispering gallery mode.

Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	iv
List of Tables	vii
List of Figures	viii
Acknowledgements	x
Dedication	xi
1 Introduction	1
1.1 Background and Motivation	2
1.2 Outline	5
2 Theory	7
2.1 Maxwell's Equations	7
2.2 Vector Wave Equations	8
2.3 Mode Equations	10
2.4 Scalar Mode	14
3 Finite Element Method	16
3.1 Boundary Value Problem	16

3.2	Variational Methods	18
3.2.1	Ritz Method	18
3.2.2	Galerkin Method	20
3.3	Scalar FEM	21
4	Sfepy	25
4.1	Software Requirement	25
4.2	Brief Description	26
4.3	Problem Description Files	28
4.4	Weak Formulations	31
4.5	Boundary Conditions	32
5	Gmsh	34
5.1	Introduction	34
5.2	Geometry	34
5.3	Finite Element Mesh	36
5.4	Mesh Size and Quality Measurement	37
6	Fiber Mode Simulation	41
6.1	Structure	41
6.2	Simulation	43
6.3	Results and Discussion	44
7	Whispering Gallery Mode Simulation	53
7.1	Structure	53
7.2	Simulation	55
7.3	Result and Discussion	57
8	Conclusion	65
A	Problem Description File	66

B Notation	69
References	70

List of Tables

Table 6.1	Waveguide parameter in different step-index fibers	46
Table 6.2	Power contribution of fiber type 1	48
Table 6.3	Power contribution of fiber type 2	48
Table 6.4	Power contribution of fiber type 3	50
Table 7.1	Whispering gallery mode simulation for microtoroid	61

List of Figures

Figure 3.1	The hybrid of Dirichlet and Neumann conditions	17
Figure 3.2	Linear triangular element	22
Figure 5.1	This is a mesh on the cross section of fiber with cladding using Gmsh	35
Figure 5.2	This is the mesh entity illustration, (0,i) represents vertex entity ,(1,i) represents edge entity and (2,i) represents face entity . . .	37
Figure 5.3	Two dimensional finite element meshes	37
Figure 5.4	Three dimensional finite element meshes	38
Figure 5.5	Mesh and EM field profile of a fiber cross section	39
Figure 6.1	Step Index Fiber, Optical Fiber Communication[1]	42
Figure 6.2	Fiber Core Diameter	43
Figure 6.3	5 variations of the lowest order step-index fiber mode	45
Figure 6.4	Relative Intensity profiles of different modes of fiber 1 and 2 . .	47
Figure 6.5	Sfepy generated mode profiles of different step-index fiber . . .	49
Figure 6.6	Relation between core power and cladding power over increasing core radius	50
Figure 6.7	Relation between fundamental mode and other mode over increasing core radius	51
Figure 7.1	The upper left (a) is the illustration of acoustic whispering gallery, the upper right (b) is optical whispering gallery, and (c) is an ultrahigh-Q optical microcavity resonator structure[2]	54

Figure 7.2	The black line indicates the physical edge of the microcavity. Note that the left figure is the cross section profile of the microcavity and the right figure is the horizontal plane of the microcavity[3].	58
Figure 7.3	Relative error of resonant wavelength for 340 th fundamental WGM in microtoroid as mesh size is halved constantly. For analytical purpose, we take $\lambda_{ref} = 780.91nm$ as the reference.	59
Figure 7.4	Cross section of a microtoroid which is submerged in the water with wavelength $\lambda = 633nm$	60
Figure 7.5	Fundamental TE mode for microtoroid submerged in water, where $m = 636$, $\lambda = 632.784$ nm, (a) is the real part of z-directed electric field, (b) is the real part of ρ -directed electric field, (c) is the whole electric field squared $ E ^2$ and (d) is the real part of ϕ -directed electric field.	63
Figure 7.6	Fundamental TE mode for microtoroid submerged in water, where $m = 636$, $\lambda = 632.4$ nm, (a) is the real part of z-directed electric field, (b) is the real part of ρ -directed electric field, (c) is the whole electric field squared $ E ^2$ and (d) is the real part of ϕ -directed electric field.	64

ACKNOWLEDGEMENTS

I would like to convey my sincere gratitude to:

Prof. Tao Lu for his enlightening advice and precious guidance through my program.

Prof. Wusheng Lu for his valuable expert advice and support.

my family and friends, for their steadfast support on my trek to explore the truth
of science and world.

I think, therefore I am.

Je pense, donc je suis

Cogito, ergo sum

DEDICATION

To my parents!

Chapter 1

Introduction

This report will address the construction of a 2D finite elements mode solver using Sfepy and Gmsh. I will present a detailed walkthrough of solving a fiber mode and a whispering gallery mode using the mode solver.

Simple Finite Elements in Python (Sfepy) is an open-source multi-platform framework designed to solve partial differential equations (PDEs). It supports 1D, 2D and 3D, combining other Python package Sfepy can be implemented in the various application. For example, it has been used for multiscale biomechanical modelling [4], photonic band gaps [5], acoustic transmission coefficients [6], etc.

Gmsh is an open-source 3D finite element meshing tool that provides various algorithms to generate finite element mesh: Frontal, MMG3D, MeshAdapt. Meantime, it's capable of creating ordered and hybrid meshes, including which are with the difference in linear dimensions of the smallest and the largest finite elements by 3-5 orders within a single mesh [7].

In the following chapters, I will detail Sfepy and Gmsh's use and the simulation results of my finite elements mode solver.

1.1 Background and Motivation

The finite element method (FEM) is a popular method using numerical algorithms to solve engineering problems with even complicated phenomena. It's significant to understand how FEM developed with its historical perspective.

The method of dividing an entire domain into discrete elements has a long history that can be traced back to the estimation of π when ancient mathematicians used polygon to divide the circle to get an accurate approximation. Shellback, in 1851, first used the finite difference method to divide a domain into triangles to calculate the minimum surface bounded by a closed curve [8]. And the modern research on the finite element method started in the 1940s when aircrafts came into people's lives. Aircrafts related structural engineering, such as fuselage and wings, incited scientists to discretize the whole structure into one dimension elements (bars or beams) to solver continuous solid, and this method was called framework method, which is considered as the precursor of FEM [9].

Courant [10], who proposed the modern prototype of FEM in 1943, divided the cross section of a hollow shaft into triangular subdomain using piecewise function, and solved the torsional stiffness by calculating the piecewise function over each subdomain at known nets or nodes, which we later consider these known nodes or nets as boundary values. And Levy proposed a stiffness method in 1953 [11] as an alternative in aircraft structure engineering, which didn't attract too much attention until the invention of the fast processing computer. In the early 1960s, Clough [12] first uses the term "finite elements" to solve the aircraft stress problem. And Martin, in 1961, extended FEM from two dimensions to three dimensions by discretizing the domain into small tetrahedrals [13]. Worth mentioning that all the above works handled small-scale problems; nevertheless, works related to large-scale problems were also carried out in the 1960s, such as thermal and deflection analysis [14], distributed-mass systems [15], nonlinear material and viscoelasticity [16]. Meanwhile, Melosh [17] made another achievement of FEM –the invention of variation formulation

(also known as weak formulation), which is the core to modern FEM. Melosh made FEM possible to solve a more complicated problem: fluid flow, torsion problem, the electromagnetic field in the waveguide, and heat conduction. However, many applications were limited due to the complexity of equations with thousands of degrees of freedom (DOF), as those equations were far beyond human calculation capacity by hand. And this predicament has been improved in the late 1970s with the development of computer graphics, and more and more FEM software commenced after that. And I will take about more detail about commercial software later. With the development of computation capacity, graphic processing, finite element software became more and more powerful; however, a retrospect of history and theory of FEM can help us get a better understanding behind this software and get a more reliable and accurate result.

The general definition of finite element method (FEM) is using a numerical method to divide a domain into small finite elements, and partial differential equations and approximation methods are applied over each parts using the finite difference method, and combine the results of each element to get final solution [18][19][20].

In the actual application, we can break down the process into several steps: First, identify and categorize the problem. Usually, there are some types of FEM problems and their corresponding solutions; it's essential to select the correct one. Second, derive a mathematical model as simple as possible from the physical concepts of real problems. Then perform FEM with the aid of software, and check the results. Worth to mention that the FEM is iterative, which means the results need to be interpreted carefully even though the results "ideally" match our expectation.

Nowadays, the potential of FEM seems infinite. With the development of communication technologies, our lives are more convenient than ever before. From browsing your phone to cloud-based service, you can easily run a complex simulation on your mobile device with the aid of cloud service. And newly developed numerical tactics realize the significant reduction of inherent approximation errors of FEM. And more and more commercial FEM software has become reliable, accurate, fast, and flexible.

With the development of mathematics and physics, partial differential equations (PDEs) played a more significant role in research. However, people found out that lots of PDEs cannot be solved analytically. Fortunately, with the assistance of computers, people designed various methods to solve PDEs numerically. The problems pertaining to PDEs that always come with specific boundary value problems (BVPs) have covered a wide range of fields, such as electromagnetic (EM), mechanical, thermal, and so forth.

Generally, obtaining an analytical solution satisfying both the PDEs and BVPs in the given domain is usually tricky even though feasible in most cases, especially in our modern research, which always has numerous variables to solve in one PDE problem. In this case, we seek an alternative way to solve such an impossible problem—using software to obtain numerical solutions. And the standard method is to expand the solution or discretize the equations under a certain algorithm to approximate the PDEs. In this report, I'll focus on the eigenvalue problem pertained to mode solver in 2D scalar optical fiber and microcavity.

The finite elements method (FEM) has been developed over 40 years. People specialized in different areas show increasing interest in this method. Like other computer-assisted numerical techniques, the FEM also divides the entire domain into small subdomains to get an approximate solution in each subdomain that satisfies the adjacent subdomains boundary conditions [21]. Therefore within this small subdomain, FEM can easily derive an approximate solution compared to the whole domain [22]. With the development of technology, more and more FEM software comes to our sights. Currently, the popular FEM-based software includes MARC, COMSOL, ADINA, ANSYS, ABAQUS. And the nonlinear finite element analysis software MSC/MARC has become the most powerful finite element software globally. COMSOL provides a cross-platform finite element analysis and multiphysics simulation solution. ANSYS is dedicated to the analysis of couples of fields such as fluid, heat, electromagnetic [23]. ABAQUS is best at nonlinear finite element calculation capabilities such as geometric nonlinearity, material nonlinearity, and state nonlinearity. ADINA, which MIT develops, shows its

powerful capability in nonlinear finite element calculation and fluid-structure coupling analysis [24].

Globalization brings enormous competition to the market. The traditional engineering companies and manufacturers desperately seek to reduce their development expense to accelerate their productiveness. Sometimes, those companies may confront embarrassment and catastrophic disasters when trying to take the "shortcut." FEM software transforms the traditional facilities and their testing equipment into their virtual counterparts [25]. Therefore, compared to creating actual prototypes, FEM can make it possible to design and test actual applications in digital form. In this case, the engineering industry will pay much less for their invention and testing, saving a significant budget [26].

Hence, no matter if you are designing an auto, aircraft, building, or anything else, you will understand how they respond to certain conditions, such as heat, fluids, electromagnetic, or other external impacts, before they become touchable entities. FEM software provides you a possibility of testing your products with real-world stressors in virtual. With FEM software, there is no more testing cost, and all design and test processes become more convenient with the easiness of changing parameters.

1.2 Outline

This report contains the following chapters:

Chapter 2 provides essential theoretical background about Maxwell equations and Standard Wave equations following by a detailed explanation of wave mode.

Chapter 3 gives an introduction of boundary value problem and common numerical methods using in FEM.

Chapter 4 introduces the software SfePy and its construction following by a brief description about weak formulations and boundary conditions.

Chapter 5 introduces the software Gmsh and the generation of finite element mesh.

Chapter 6 demonstrates the finite-element-method simulation of step-index fibers.

The result of different step-index fibers have been compared.

Chapter 7 demonstrates the finite-element-method simulation of WGM. A simple microcavity has been simulated, and the results have been presented.

Chapter 8 concludes the report.

Chapter 2

Theory

Maxwell's equations are the most important equations in electromagnetic fields. This FEM mode solver de facto is to solve Helmholtz wave equation which is derived from Maxwell's equations. Therefore before I discuss the FEM mode solver, I will first consider Maxwell's equations and the Helmholtz equation.

2.1 Maxwell's Equations

In our project, Maxwell equations is governing equations, so let's consider the differential form of Maxwell equations[27]:

$$\nabla \cdot \mathbf{D} = \rho \quad (2.1)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.2)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (2.3)$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}_e \quad (2.4)$$

where \mathbf{E} and \mathbf{H} is the electric and magnetic field respectively, $\mathbf{D} = \epsilon\mathbf{E}$ is the electric flux density, $\mathbf{B} = \mu\mathbf{H}$ is the magnetic flux density. Sometimes there is no need to calculate all three dimension component of \mathbf{E} and \mathbf{H} . For example, in the waveguide,

the wave either has no z-component of the electric field or no z-component of the magnetic field, so that in the waveguide, as well as in the optical fiber, there is a particular electromagnetic (E or H) which radiation plane is always perpendicular to its radiation propagation direction. We call these electromagnetic waves Transverse Electric (TE) modes or Transverse Magnetic (TM) modes. And TE and TM modes are also known as semi-vector modes, which I will discuss in the following section. And there is a third mode in waveguide and fiber, which has non-zero z-component of both electric and magnetic field, for waveguide, it refers to Transverse Electromagnetic (TEM) mode, and for fiber, it refers to hybrid mode (also known as HE/EH mode, which depends on the dominance of longitudinal field).

2.2 Vector Wave Equations

Before we talk about the semi-vector mode, let's first examine vector mode(also known as full vector mode). Generally, we denote time-harmonic electromagnetic waves as:

$$\begin{cases} \mathbf{E}(x, y, z, t) = \mathbf{E}(x, y, z)e^{i\omega t} \\ \mathbf{H}(x, y, z, t) = \mathbf{H}(x, y, z)e^{i\omega t} \end{cases} \quad (2.5)$$

In dielectric media, we often denote the dielectric permittivity as $\varepsilon(r) = n^2(r)\varepsilon_0$, and magnetic permeability as μ , and $n(r)$ is the refractive index and ε_0 is the free space dielectric permittivity. Since the magnetism in the material is so weak that we only consider the permeability is the same as of free space, hence $\mu = \mu_0$.

We can rewrite the Maxwell's equation (2.1)-(2.4) in phasor form as:

$$\begin{cases} \nabla \cdot (\varepsilon_0 n^2 \mathbf{E}) = \rho \\ \nabla \cdot (\mu \mathbf{H}) = 0 \\ \nabla \times \mathbf{E} = -\mu_0 \frac{\partial \mathbf{H}}{\partial t} = -i\omega \mu_0 \mathbf{H} \\ \nabla \times \mathbf{H} = \varepsilon_0 n^2 \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J}_e = i\omega \varepsilon_0 n^2 \mathbf{E} + \mathbf{J}_e \end{cases} \quad (2.6)$$

here we consider the speed of light in free space as c , the wavelength as λ , angular frequency as ω , wave number as k , hence

$$\omega = 2\pi \frac{c}{\lambda} = \frac{k}{\sqrt{\varepsilon_0 \mu_0}} \quad (2.7)$$

Now let's derive the vector mode for electric field. Take the curl operator over the third equation in(2.6), and we get

$$\begin{aligned} \nabla \times (\nabla \times \mathbf{E}) &= -i\omega \mu_0 \nabla \times \mathbf{H} = -i\sqrt{\frac{\mu_0}{\varepsilon_0}} k \nabla \times \mathbf{H} \\ &= -i\sqrt{\frac{\mu_0}{\varepsilon_0}} k \mathbf{J}_e + k^2 n^2 \mathbf{E} \end{aligned} \quad (2.8)$$

and consider the follow vector identities

$$\begin{cases} \nabla \times (\nabla \times \mathbf{A}) = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A} \\ \nabla(\nabla \times \mathbf{A}) = 0 \end{cases} \quad (2.9)$$

where we denote that $\nabla^2 = \Delta = \sum_{i=1}^3 \frac{\partial^2}{\partial x_i^2}$, then we get

$$\begin{aligned} \nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E} &= -i\sqrt{\frac{\mu_0}{\varepsilon_0}} k \mathbf{J}_e + k^2 n^2 \mathbf{E} \\ (\nabla^2 + k^2 n^2) \mathbf{E} &= \nabla(\nabla \cdot \mathbf{E}) + i\sqrt{\frac{\mu_0}{\varepsilon_0}} k \mathbf{J}_e \end{aligned} \quad (2.10)$$

Apply divergence to the fourth equation in (2.6), and based on the second equation in (2.9), we get

$$\nabla \cdot (\nabla \times \mathbf{H}) = 0$$

$$\nabla \cdot \mathbf{J}_e + i\omega\varepsilon_0 \nabla \cdot (n^2 \mathbf{E}) = \nabla \cdot \mathbf{J}_e + i\omega\varepsilon_0 \mathbf{E} \nabla(n^2) + i\omega\varepsilon_0 n^2 \nabla \cdot \mathbf{E} = 0 \quad (2.11)$$

so after some calculation, we get

$$\begin{aligned} \nabla \cdot \mathbf{E} &= -\frac{i}{\omega\varepsilon_0} \frac{\nabla \cdot \mathbf{J}_e}{n^2} - \frac{\nabla n^2}{n^2} \mathbf{E} \\ &= -\frac{i}{\omega\varepsilon_0} \frac{\nabla \cdot \mathbf{J}_e}{n^2} - \mathbf{E} \nabla \ln n^2 \end{aligned} \quad (2.12)$$

substitute (2.12) into (2.10),

$$(\nabla^2 + k^2 n^2) \mathbf{E} = -\nabla(\mathbf{E} \nabla \ln n^2) - \frac{i}{\omega\varepsilon_0} \nabla \left(\frac{\nabla \cdot \mathbf{J}_e}{n^2} \right) + i \sqrt{\frac{\mu_0}{\varepsilon_0}} k \mathbf{J}_e \quad (2.13)$$

this is the vector mode for electric field, and similarly we can get the vector mode for magnetic field as:

$$(\nabla^2 + k^2 n^2) \mathbf{H} = (\nabla \times \mathbf{H}) \nabla \ln n^2 - \nabla \times \mathbf{J}_e - \mathbf{J}_e \times \nabla \ln n^2 \quad (2.14)$$

this two equations are vector wave equations.

2.3 Mode Equations

To obtain an more concise equations since the original Maxwell equation requires three dimensions differential, we need to reduce Maxwell equations into two dimensions. In order to reduce the dimension, let's consider the scalar form of Maxwell equations.

Denote that \mathbf{E} and \mathbf{H} each one has three component called (E_x, E_y, E_z) , (H_x, H_y, H_z) respectively, let's rewrite the Maxwell equations(2.1)-(2.4) and neglect the conduction

current \mathbf{J}_e :

$$\left\{ \begin{array}{l} i\omega\varepsilon E_x = \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right) \\ i\omega\varepsilon E_y = \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x}\right) \\ i\omega\varepsilon E_z = \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right) \\ i\omega\mu H_x = -\left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z}\right) \\ i\omega\mu H_y = -\left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x}\right) \\ i\omega\mu H_z = -\left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}\right) \end{array} \right. \quad (2.15)$$

Since we expand the curl operator to get above equations, we can easily set given component to zero if that component doesn't exist in certain dimension.

Let's consider a detailed process of reducing Maxwell from 3D to 2D, using TE mode as an example (TM mode is similar as TE mode), let's set $H_x = H_y = 0, E_z = 0$ and $\frac{d}{dz} = 0$, then we can get:

$$\nabla \times \mathbf{E} = \varepsilon_{ijk} \hat{e}_i \partial_j E_k = \varepsilon_{312} \hat{e}_3 \partial_1 E_2 - \varepsilon_{321} \hat{e}_3 \partial_2 E_1 = \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}\right) \hat{e}_z \quad (2.16)$$

and,

$$\nabla \times \mathbf{H} = \varepsilon_{ijk} \hat{e}_i \partial_j H_k = \varepsilon_{123} \partial_2 H_3 \hat{e}_1 - \varepsilon_{213} \partial_1 H_3 \hat{e}_2 = \frac{\partial H_z}{\partial y} \hat{e}_x - \frac{\partial H_z}{\partial x} \hat{e}_y \quad (2.17)$$

where $\hat{e}_x, \hat{e}_y, \hat{e}_z$ are the unit directional vector in x,y,z respectively, here we take Einstein notation, 1,2,3 represent x,y,z respectively. Then substitute (2.16) and (2.17) into (2.3) and (2.4) we will get:

$$\left\{ \begin{array}{l} i\omega\varepsilon \mathbf{E} = \frac{\partial H_z}{\partial y} \hat{e}_x - \frac{\partial H_z}{\partial x} \hat{e}_y \\ i\omega\mu \mathbf{H} = \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}\right) \hat{e}_z \end{array} \right. \quad (2.18)$$

in equation (2.15), $\mathbf{E} = (E_x, E_y, 0)$, $\mathbf{H} = (0, 0, H_z)$, we can rewrite these equations into scalar form:

$$\begin{cases} i\omega\varepsilon E_x = \frac{\partial H_z}{\partial y} \\ i\omega\varepsilon E_y = -\frac{\partial H_z}{\partial x} \\ i\omega\mu H_z = \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}\right) \end{cases} \quad (2.19)$$

this is the TE mode after dimension reduction. Similarly, we can also get the TM mode after dimension reduction as :

$$\begin{cases} i\omega\mu H_x = \frac{\partial E_z}{\partial y} \\ i\omega\mu H_y = -\frac{\partial E_z}{\partial x} \\ i\omega\varepsilon E_z = \left(\frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x}\right) \end{cases} \quad (2.20)$$

Let's further take a look at TM mode equations (2.19). We can do some transformation to evade calculating H_x, H_y components, and use one partial differential equation (PDE) to express E_z .

$$\begin{cases} i\omega\mu \frac{\partial}{\partial y} H_x = \frac{\partial^2 E_z}{\partial y^2} \\ i\omega\mu \frac{\partial}{\partial x} H_y = -\frac{\partial^2 E_z}{\partial x^2} \\ -\omega^2\varepsilon\mu E_z = i\omega\mu \left(\frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x}\right) \end{cases} \quad (2.21)$$

Substitute the first two equations into the last one in(2.20), we will get the wave equation (Helmholtz equations):

$$\begin{cases} \nabla^2 \mathbf{E} = -\omega^2\varepsilon\mu \mathbf{E} = -k^2 \mathbf{E} \\ \nabla^2 \mathbf{H} = -\omega^2\varepsilon\mu \mathbf{H} = -k^2 \mathbf{H} \end{cases} \quad (2.22)$$

And in optical fiber, refractive index is usually z-independent since the light propagates along the z-direction, we rewrite the envelop part of electromagnetic field in (2.5) as

$$\begin{cases} \mathbf{E}(x, y, z) = \mathbf{E}(x, y)\mathbf{e}(z) \\ \mathbf{H}(x, y, z) = \mathbf{H}(x, y)\mathbf{h}(z) \end{cases} \quad (2.23)$$

substitute eq.(2.23) into eq.(2.22) we can obtain,

$$e(z)(\nabla^2 + k^2)\mathbf{E}_z = \mathbf{E}_z\nabla^2 e(z) \quad (2.24)$$

worth to note that $\mathbf{E}(x, y)$ is x,y-dependent, $\mathbf{e}(z)$ is only z-dependent, therefore the solution of eq.(2.24) can only be in the form as:

$$\mathbf{e}(z) = e^{i\beta z}$$

and the gradient operator ∇ can be rewritten as $\nabla = \vec{\nabla}_\perp + \hat{\mathbf{z}}\frac{\partial}{\partial z}$, we only consider the z-component of \mathbf{E} field for TM mode, so

$$\nabla^2 E_z = \vec{\nabla}_\perp^2 E_z + \frac{\partial E_z}{\partial z^2} = \vec{\nabla}_\perp^2 E_z - \beta^2 E_z \quad (2.25)$$

and since we neglect the conduction current $J = 0$, the substitute (2.24) into (2.13),

$$(\vec{\nabla}_\perp^2 + k^2 n^2 - \beta^2)E_z = -(\vec{\nabla}_\perp + i\beta\hat{\mathbf{z}})(E_z\nabla_\perp \ln n^2) \quad (2.26)$$

this is the TM mode, and similarly we can get TE mode as:

$$(\vec{\nabla}_\perp^2 + k^2 n^2 - \beta^2)H_z = ((\vec{\nabla}_\perp + i\beta\hat{\mathbf{z}}) \times H_z) \times \vec{\nabla}_\perp \ln n^2 \quad (2.27)$$

And for most cases, the refractive index are step functions, therefore, (2.25) and (2.26) can be written as:

$$\begin{cases} \vec{\nabla}_{\perp}^2 E_z + (k^2 n^2 - \beta^2) E_z = 0 \\ \vec{\nabla}_{\perp}^2 H_z + (k^2 n^2 - \beta^2) H_z = 0 \end{cases} \quad (2.28)$$

For semi-vector mode, we only need to consider only one of (2.27) since TE or TM mode must satisfy $E_z = 0$ (TE mode) or $H_z = 0$ (TM mode).

2.4 Scalar Mode

For the full vector mode equations (2.27), the solutions are the exact same as Maxwell's equations. If we want to get a more accurate solution with a numerical method without considering any other computational issue, we should always compute the full vector mode equations. However, in practice, we have to confront numerous problems when using the numerical method. And memory consumption and computation time are two major issues that we have to consider. The more accurate solution always means more memory consumption and more computation time. Sometimes it's not worth consuming the whole day to get an exact solution for simple Maxwell's equations. Therefore it's necessary to find some approximation methods to simplify the initial problems and to get a little less precise solution but within the tolerance. Semi-vector mode is one of the approximation methods for the transverse mode (TE and TM mode) since it reduces two coupled full vector modes into one simple equation. However, for some certain waveguide or fiber with extremely small refractive index, i.e., the refractive index is almost the same in the domains where the field intensity is significant, we can apply another approximation—scalar approximation. With scalar approximation (scalar mode), we neglect the electromagnetic polarization, and the scalar mode can be expressed by scalar Helmholtz equation:

$$\nabla \Phi + (k_0^2 n^2 - \beta^2) \Phi = 0 \quad (2.29)$$

and Φ can represent an electric field or magnetic field at all points.

Chapter 3

Finite Element Method

There are two main types of FEM in applications: the Scalar FEM and the Vectorial FEM. Each of them has its weaknesses and advantages. In our project, we chose the first one since our implementation has no spurious problem and the eigenvalue matrixes are symmetrical; in this case, Scalar FEM can provide a more efficient and faster simulation.

3.1 Boundary Value Problem

Before talking about FEM, let's first take a look at the Boundary Value Problem (BVP). Consider a typical BVP: a governing equation in differential form is defined in the domain with some boundary conditions as 3.1.

$$\Delta\Phi = f \tag{3.1}$$

where

$$\Delta = \nabla \cdot \nabla = \sum_{i=1}^3 \frac{\partial^2}{\partial x_i^2}$$

is the Laplacian operator and f is the source, Φ is the unknown function. And generally there are three boundary conditions (BCs):

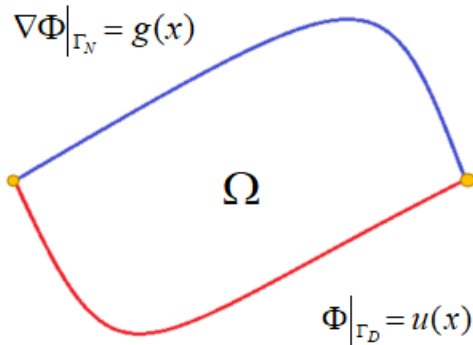


Figure 3.1: The hybrid of Dirichlet and Neumann conditions

1. First-type BC (Dirichlet BC)(3.2): u is a known function defined on the boundary Γ_D . It specifies the value of the unknown function along the boundary.

$$\Phi|_{\Gamma_D} = u(x), x \in \Gamma_D \quad (3.2)$$

2. Second-type BC (Neumann BC)(3.3): g is a known function defined on the boundary Γ_N . It specifies the value where the derivative of the unknown function is applied along the boundary.

$$\hat{\mathbf{n}} \cdot \nabla\Phi = g(x), x \in \Gamma_N \quad (3.3)$$

3. Third-type BC (Robin condition)(3.4): this is an impedance condition. It specifies the value of linear combination of the unknown function and its derivative.

$$a\Phi + \hat{\mathbf{n}} \cdot b\nabla\Phi = g(x), x \in \Gamma_R \quad (3.4)$$

However, the hybrid boundary conditions is more general in applications, such as the hybrid of Dirichlet and Neumann conditions shown in Figure (3.1).

3.2 Variational Methods

Among the all methods in FEM, there are two popular methods used in FEM to seek the solution: the Ritz Method and the Galerkin Method.

3.2.1 Ritz Method

The Ritz method (as known as Rayleigh-Ritz method) is a variational approximation method where BVP has an alternative representation called functional $F(\Phi)$ (eq. (3.6)). And to seek the solution, therefore, is to minimize the functional related to the governing equations and corresponding boundary conditions.

Just like most approximation method, Ritz method also assume the solution has such form:

$$\tilde{\Phi} = \sum_{k=1}^M \alpha_k \phi_k \quad (3.5)$$

where α_k is the unknown parameter and $\{\phi_k\}$ is the basis function in computational domain, and Ritz method is to minimize the functional $F(\tilde{\Phi})$ corresponding to α_k ,

$$F(\tilde{\Phi}) = \langle \Delta \tilde{\Phi}, \tilde{\Phi} \rangle - 2 \langle \tilde{\Phi}, f \rangle \quad (3.6)$$

where f is the known function (see eq.(3.1)), and since $\{\phi_k\}$ are orthogonal to each other, we can get the stationary solution $\tilde{\Phi}$ when

$$\frac{\partial F(\tilde{\Phi})}{\partial \alpha_k} = 0, k = 1, 2, \dots, M \quad (3.7)$$

let rewrite eq.(3.5) in vector form as

$$\tilde{\Phi} = \sum_{k=1}^M \alpha_k \phi_k = \vec{\alpha}^T \vec{\phi} \quad (3.8)$$

so substitute eq.(3.5) into eq.(3.6), we get:

$$F(\tilde{\Phi}) = \vec{\alpha}^T \int_{\Omega} \vec{\phi} \vec{\phi}^T d\Omega \cdot \vec{\alpha} - 2\vec{\alpha}^T \int_{\Omega} \vec{\phi} f \cdot d\Omega \quad (3.9)$$

In order to minimize the functional $F(\tilde{\Phi})$, using the optimization condition (3.7), we can get:

$$\begin{aligned} \frac{\partial F(\tilde{\Phi})}{\partial \alpha_k} &= \int_{\Omega} \vec{\phi}_j \vec{\phi}_j^T d\Omega \cdot \vec{\alpha} + \vec{\alpha}^T \int_{\Omega} \vec{\phi} \vec{\phi}_j^T d\Omega - 2 \int_{\Omega} \vec{\phi}_j f d\Omega \\ &= 0 \quad j = 1, 2, \dots, M \end{aligned} \quad (3.10)$$

and after some transformation,

$$\sum_{k=1}^M \alpha_k \int_{\Omega} (\vec{\phi}_k \vec{\phi}_j^T + \vec{\phi}_j \vec{\phi}_k^T) d\Omega - 2 \int_{\Omega} \vec{\phi}_j f d\Omega = 0 \quad j = 1, 2, \dots, M \quad (3.11)$$

hence the equation(3.11) can be rewritten in matrix form as:

$$D \cdot \vec{\phi} = \vec{b} \quad (3.12)$$

and the matrix \mathbf{D} is given by

$$D_{ij} = \int_{\Omega} \vec{\phi}_i \vec{\phi}_j^T d\Omega \quad (3.13)$$

therefore we can solve the approximation problem by obtaining $\{\alpha_k\}$ using matrix equation.

We can generalize the basic steps of FEM using Ritz method:

- Divide the whole domain into several subdomains $\Omega_l, l = 1, 2, \dots, N$
- Over each subdomain, calculate the unknown function Φ as the interpolation of values on each element $\tilde{\Phi}^l = \sum_{k=1}^M \alpha_k^l \phi_k^l$,
- Reformulate the functional over each subdomain related to unknown parameters:

$$F(\tilde{\Phi}) = \sum_{l=1}^N F(\tilde{\Phi}^l),$$

- Implementing the optimization condition (3.7) $\frac{\partial F(\tilde{\Phi})}{\partial \alpha^k}$ to minimize the functional $F(\tilde{\Phi})$
- Solve the governing equations using Ritz method.

3.2.2 Galerkin Method

Other than the Ritz method, there is another popular method – Galerkin Method. Galerkin method specifies the residuals of the differential equation and weights the residuals to seek the solution of BVP. Just same as the Ritz method, we still assume the approximated solution of eq. (3.1) is $\tilde{\Phi}$, so the residual is defined as:

$$r = \Delta\tilde{\Phi} - f \quad (3.14)$$

and take the coordinate function as the weight ϕ_i , we seek to get the least residual over the whole domain Ω with the weight,

$$R_i = \int_{\Omega} \phi_i r d\Omega = 0, \quad i = 1, 2, \dots, M \quad (3.15)$$

substitute (3.14) into (3.15), we get:

$$R_i = \int_{\Omega} (\phi_i \Delta\tilde{\Phi} - \phi_i f) d\Omega = 0, \quad i = 1, 2, \dots, M \quad (3.16)$$

therefore (3.16) is similar as (3.11), it also can be solved by matrix equation in (3.13). Unlike the Ritz method minimizing the functional(functional has the form of energy), Galerkin seeks to minimize the error between known function and approximate function.

3.3 Scalar FEM

As mentioned in chapter 2, there are three types of mode solver, full-vector mode, semi-vector mode, and scalar mode. And in this project, we are dedicated to solving fiber mode and whispering gallery mode, which both have less refractive index changes over the computation region. In this case, Scalar FEM provides an efficient method for solving these problems.

Compared to the full-vector mode and semi-vector mode, even though they always obtain more accurate results, the scalar mode will cost much less computation time and obtain a relatively accurate result which usually satisfies the requirement of calculation accuracy when the simulation material is inhomogeneous and isotropic, which has little refractive index changes over computation region.

The scalar wave equation of isotropic material can be expressed as:

$$\Delta\Phi + k^2\Phi = 0 \quad (3.17)$$

where k is the eigenvalue and Φ is the unknown wave function. From (3.9) we can obtain the functional $F(\Phi)$ as:

$$F(\Phi) = \int_{\Omega} (\Delta\Phi - 2k^2\Phi) ds \quad (3.18)$$

where Ω is the computational region of waveguide.

In the numerical calculation, the computational region is always divided into small subdomain (usually triangle, this step is called mesh, which will be addressed in chapter 5), and therefore we divide our functional(3.18) into as:

$$F(\Phi) = \sum_{l=1}^N \int_{\Omega_l} (\Delta\Phi^l - 2k^2\Phi^l) \quad (3.19)$$

as mentioned in section Ritz Method, l is the subdomain number, N is the total number of subdomain, and Φ^l is the approximation of three vertices of the subdomain

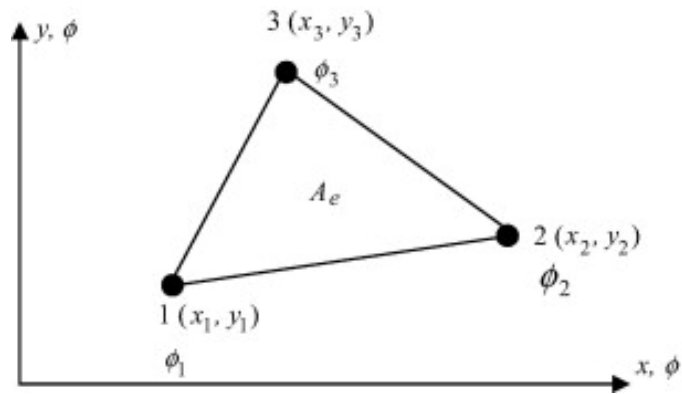


Figure 3.2: Linear triangular element

shown in figure: where A_e is the subdomain, and we try to express any points inside A_e using unknown function φ , since the element is rather small, we can approximate φ linearly using the following :

$$\begin{aligned}\varphi &= ax + by + c \\ \varphi(x, y) &= [M(x, y)]^T \varphi_l\end{aligned}\tag{3.20}$$

where $M(x, y)$ and φ_l are:

$$\begin{aligned}\varphi_l &= [\varphi_1 \quad \varphi_2 \quad \varphi_3]^T \\ M(x, y) &= \frac{1}{2D} \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\end{aligned}\tag{3.21}$$

and $2D$ is the determinant given by:

$$2D = \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}\tag{3.22}$$

and

$$\begin{aligned}
a_r &= y_s - y_t \\
b_r &= x_t - x_s \\
c_r &= x_s y_t - x_t y_s
\end{aligned} \tag{3.23}$$

the x_r, y_r are the Cartesian coordinate of point $\varphi_1, \varphi_2, \varphi_3$, and the subscription r, s, t are 1,2,3; 3,1,2; 2,3,1, therefore we have:

$$\begin{aligned}
\frac{\partial \varphi_l(x, y)}{\partial x} &= a \cdot \Phi_l \\
\frac{\partial \varphi_l(x, y)}{\partial y} &= b \cdot \Phi_l
\end{aligned} \tag{3.24}$$

and where $a = (a_1, a_2, a_3), b = (b_1, b_2, b_3)$, and we denote $\omega = [a_1 x + b_1 y + c_1, a_2 x + b_2 y + c_2, a_3 x + b_3 y + c_3]$, therefore we rewrite the functional $F(\Phi)$ (3.18) as

$$\begin{aligned}
F(\Phi) &= \int_{\Omega} \left(\frac{\partial \Phi_l}{\partial x} + \frac{\partial \Phi_l}{\partial y} - 2k^2 \Phi_l \right)^2 \\
&= \sum_{l=1}^N \int_{\Omega} (a \cdot \Phi_l)^T (a \cdot \Phi_l) + (b \cdot \Phi_l)^T (b \cdot \Phi_l) - k^2 (\omega \Phi_l)^T (\omega \Phi_l) \\
&= \sum_{l=1}^N \Phi_l^T \cdot P_l \cdot \Phi_l - k^2 \cdot \Phi_l \cdot Q_l \cdot \Phi_l
\end{aligned} \tag{3.25}$$

where

$$\begin{aligned}
P_l &= D[a^T \cdot a + b^T \cdot b] \\
Q_l &= \frac{D}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}
\end{aligned} \tag{3.26}$$

Worth noting that the subscripts l in P_l, Q_l are subdomain matrices, and $\sum_{l=1}^N$ are assemble process of subdomain matrices into the whole domain matrix, therefore, we

donte P_w, Q_w as the whole domain matrices, and we obtain functional $F(\Phi)$ (3.18) as:

$$F(\Phi) = \Phi_l^T \cdot P_w \cdot \Phi_l - k^2 \cdot \Phi_l \cdot Q_w \cdot \Phi_l \quad (3.27)$$

therefore we can get FEM solution by minimizing the functional when the partial primes of functional are equal to zero:

$$\frac{\partial F(\Phi)}{\partial \Phi_i} = 0, i = 1, 2, \dots, N \quad (3.28)$$

where N is number of nodes.

Chapter 4

Sfepy

So far, the most powerful software that can process finite elements analysis is commercial, such as COMSOL Multiphysics, Matlab, Advance Design, etc. Sfepy provides a free and open-source option for us to solve finite elements problems.

4.1 Software Requirement

Sfepy is built mainly in Python and requires all the following prerequisites and python packages to successfully run:

1. a C compiler suite,
2. Python 3.x,
3. *NumPy*,
4. *Cython*,
5. *Pyparsing*,
6. *SciPy*,
7. *meshio* for writing and reading mesh files,
8. *scikit-umfpack* for umfpack solver,

9. *Matplotlib* for various plots,
10. *PyTables* for storing results,
11. *SymPy* for some functions,
12. *Mayavi* for results visualization,
13. *igakit* for simple IGA domain generator,
14. *petsc4py* and *mpi4py* for using parallel solvers from PETSc,
15. *slepc4py* for eigenvalue problem solvers from SLEPc,
16. *pymetis* for mesh partitioning using Metis,
17. *wxPython* for better IPython integration.

4.2 Brief Description

Sfepy is mostly written in Python. It depends on the "vectorized operation" from Numpy arrays for speed[5] with mostly the use of "index trick" and advanced broadcasting features. For the places where "vectorized operation" is not deployable, C and Cython are written if the code is unreadable. Other scientific python packages are also used, such as Scipy solvers and algorithms, Mayavi for 2D/3D plots which are used primarily in my mode solver for plots although it supports other plot option such as Matplotlib, and SymPy for the symbolic operation, etc.

Sfepy provides a flexible definition of various problems due to its fundamental structure of the code. Sfepy requires the weak formulation of the PDEs to define components according to the problem's mathematical counterpart. In Sfepy, we define our problems in the problem description file, which includes solution domain and regions, fields corresponding to certain regions, variables defined on the certain fields, equations (weak formulation), boundary conditions (Dirichlet boundary conditions and Neumann boundary conditions) and material parameters, etc. The term is the crucial concept

in Sfepy, which we will use to build PDEs by the linear combination of the terms. Sfepy has 105 terms; by combining these terms, we can create many common PDEs in mechanics, biomechanics, acoustic etc, with a significant exception of electromagnetism which Sfepy still work in progress. SfePy uses Galerkin approximation to discretize PDEs with finite elements. Although it provides basic polynomial functions of an order greater than 10, the orders greater than four are de facto unusable specifically in 3D, since it depends on Numpy vectorized operation, which will evaluate all the element matrices and add them to the global matrix at once - this will cause a really high memory consumption when polynomial orders are high.

It supports various kinds of solvers: linear, nonlinear, optimization, eigenvalue, and time stepping. Besides some external solvers (UMFPACK, PETSc, Pysparse, and Scipy) that can interact with Sfepy, it also provides some internal solvers which we can use directly in Sfepy:

- Nonlinear and optimization solvers:
 1. *Newton solver with backtracking line search.* It use Newton solver to solve linear and nonlinear problems for simplicity since linear problem will converge in the single nonlinear iteration.
 2. *Steepest descent optimization solver with backtracking line search.* It will be used when more complicated solvers fails.
- Time-stepping-related solvers:
 1. *Stationary solver,* which is for time-independent problems.
 2. *Equation sequence solver,* which is a stationary solver for analyzing the dependencies between equations and solves.
 3. *Simple implicit time stepping solver,* which is for time dependent problems with a fixed time step.
 4. *Adaptive implicit time stepping solver,* which provide a changing time step using a user defined function.

4.3 Problem Description Files

Problems description files (PDFs) is the key part of Sfepy since we need to write PDFs to describe our own problems. PDFs are in fact the Python modules which contain components defining the problems, such as fields, boundary conditions, regions, mesh, variables, solvers, etc. Here the following components should be specified in PDF files:

- Finite Element (FE) Mesh:

Sfepy supports various FE mesh formats, and in this report I will use medit mesh file format(.mesh) which is widely used and can be create by Gmsh. The mesh generation is a key part of mode solver, I will address a detailed process of mesh generation in the next chapter. In PDFs, the following syntax is required to quote the mesh:

```
filename_mesh = 'meshes_name.mesh'
```

- Regions:

Regions is used to choose the computational domain by FE mesh entities. And they are also used to define boundaries. Region definition includes *entity selection* and *set like operators*. The entity selection serves topological selection of certain mesh entity, such as the vertices of certain group. There are also other kind of entity: *cell, edge, facet, face*. And set like operators can be used to define intermediate regions (union, difference and intersection of two regions). In PDFs, the following syntax is required to create regions:

```
regions = {<region_name> :
(<entity_selection>,[<entity_kind>], [<option>])}
```

- Fields:

Fields(thermal field, electromagnetic fields and etc.) should be defined on certain region which we define above using following syntax:

```

fields = {
<name> : (<data_type>, <shape>, <region_name>, <approx_order>)
}

```

where `data_type` is either 'real' or 'complex', `shape` represents the number of dimensions of freedom (DOFs), in this sense, a scalar field can be represented as '1' or 'scalar', a vector field can be represented as '2', '3' or 'vector', `region_name` is the region where fields are defined, it refers to the regions we define above. And `approx_order` represents finite elements order, i.e '0', '1', '2'.

- Variables:

In this part, we define the variables on the given fields which we define above using following syntax:

```

variables = {
<name> : (<kind>, <field_name>, <spec>, [<option>])
}

```

where `kind` represents either 'unknown field', which is the field that we want to calculate, or 'test field', which is an auxiliary field used in weak form PDEs. `spec` is a specification, for primary variable i.e.'unknown field', `spec` is the order of unknowns, for dual variables, `spec` is the name of primary variables.

- Integrals:

Integrals defines the order of integrals and quadrature rules. This part is optional since we can directly define order of integrals in equations below.

```

        integrals = {
<name> : <order>
}

```

where name must begin with 'i', such as 'i2' for second order integral.

- Essential Boundary Conditions:

Dirichlet Boundary Conditions and *Periodic Boundary Conditions* will be implemented on a given region using following syntax:

```

        <type of bc> = {
<name> : (<region_name>, [<times_specification>],
{<dof_specification> : <value>[,
<dof_specification> : <value>, ...]})
}

```

where 'type of bc' is either 'ebcs' for Dirichlet boundary conditions or 'epbcs' for periodic boundary conditions. The detail of essential boundary conditions will be addressed in the following section.

- Materials:

Materials serves to define constitutional parameters and non-field arguments, such as refractive index, permeability or so. Depending on the situation, the parameter can be constant or functions defined over FM meshes.

- Equations:

Equations must be created by combining terms in [28].For example,

Laplace equation:

$$c\nabla^2 T = 0 :$$

```

equations = {
'Temperature' : ""dw_laplace.2.Omega( coef.val, t, s ) = 0""
}

```

4.4 Weak Formulations

In Sfepy, we use finite elements method(FEM) to solve partial differential equations(PDEs). After being developed so many years, FEM has a really wild range of application, such as electromagnetic fields, structural analysis, auto industry etc. FEM is built on a completed mathematical theory of weak PDE formulation. I'll give an brief description about this basic concept of FEM.

First, let's start with a fiber mode equation,

$$\nabla_{\perp}^2 E_z + (k^2 - \beta^2)E_z = f(x), x \in \Omega \quad (4.1)$$

$$E_z = u(x), x \in \Gamma_D \quad (4.2)$$

$$\nabla E_z \cdot \mathbf{n} = g(x), x \in \Gamma_N \quad (4.3)$$

where $E_z(x)$ is the unknown variable to be solved, $f(x), g(x)$ and $u(x)$ are given functions, $\Omega \in \mathbf{R}$ is the computational domain with the boundary $\partial\Omega \in \Gamma_D + \Gamma_N$, Γ_D is part of the boundary where applying Dirichlet boundary conditions and Γ_N is part of the boundary where applying Newmann boundary conditions.

The weak formulation of 4.1 is given by:

$$\int_{\Omega} (\nabla_{\perp}^2 E_z + (k^2 - \beta^2)E_z - f) \cdot s d\Omega = 0 \quad (4.4)$$

where s is a test function, and separate the integral we get:

$$\begin{aligned} 0 &= \int_{\Omega} \nabla_{\perp} \cdot (\nabla_{\perp} E_z) \cdot s d\Omega + \int_{\Omega} (k^2 - \beta^2) E_z \cdot s d\Omega - \int_{\Omega} f \cdot s d\Omega \\ &= - \int_{\Omega} \nabla_{\perp} E_z \cdot \nabla s d\Omega + \int_{\Omega} \nabla \cdot (\nabla_{\perp} E_z \cdot \mathbf{n}) d\Gamma + \int_{\Omega} (k^2 - \beta^2) E_z \cdot s d\Omega - \int_{\Omega} f \cdot s d\Omega \end{aligned}$$

by using Gaussian theorem, then we get:

$$0 = - \int_{\Omega} \nabla_{\perp} E_z \cdot \nabla s d\Omega + \int_{\Gamma_D \cup \Gamma_N} s \cdot (\nabla_{\perp} E_z \cdot \mathbf{n}) d\Gamma + \int_{\Omega} (k^2 - \beta^2) E_z \cdot s d\Omega - \int_{\Omega} f \cdot s d\Omega$$

Then we can split the surface integral into two parts: one is Dirichlet boundary and the other is Newmann boundary:

$$\int_{\Omega} \nabla_{\perp} E_z \cdot \nabla s d\Omega = \int_{\Gamma_D} s \cdot (\nabla_{\perp} E_z \cdot \mathbf{n}) d\Gamma + \int_{\Gamma_N} s \cdot (\nabla_{\perp} E_z \cdot \mathbf{n}) d\Gamma + \int_{\Omega} (k^2 - \beta^2) E_z \cdot s d\Omega - \int_{\Omega} f \cdot s d\Omega \quad (4.5)$$

Hence the equation 4.5 is the original weak formulation of fiber mode equation. Then we have to take a close look at boundary conditions as we cannot handle surface integral in Sfe.py.

4.5 Boundary Conditions

There are two types of boundary conditions in FEM: essential boundary condition and natural boundary condition. For essential boundary conditions, we always refer to the Dirichlet boundary condition. Since the test function \mathbf{s} can be in arbitrary form, let's consider a domain $H(\Omega)$ and a subdomain $H_0(\Omega)$, where

$$H(\Omega) = \{h(x) \in \Omega\},$$

$$H_0(\Omega) = \{h(x) \in \Omega; h(x) = 0, x \in \Gamma_D\}$$

Let's consider the unknown function $T \in H(\Omega)$ and the test function $s \in H_0(\Omega)$. Since the unknown function T must be continuous along the gradient in the domain

Ω , and the test function \mathbf{s} can be in an arbitrary form, take \mathbf{s} as a function which is also continuous along the gradient and \mathbf{s} is always zero on the boundary of domain Ω based on the above definition. In this case, the weak formulation will vanish into the following equation:

$$\int_{\Omega} \nabla_{\perp} E_z \cdot \nabla s d\Omega = \int_{\Gamma_N} s \cdot (\nabla_{\perp} E_z \cdot \mathbf{n}) + \int_{\Omega} (k^2 - \beta^2) E_z \cdot s d\Omega - \int_{\Omega} f \cdot s d\Omega \quad (4.6)$$

$$T(x) = u(x), x \in \Gamma$$

And the natural boundary condition always refers to the Newmann boundary condition, which is the surface flux in equation 4.3, so the integral term of equation 4.5 over surface Γ_N can be written as $\int g \cdot s$, we can rewrite the weak formulation as:

$$\int_{\Omega} \nabla_{\perp} E_z \cdot \nabla s d\Omega = \int_{\Gamma_N} s \cdot g + \int_{\Omega} (k^2 - \beta^2) E_z \cdot s d\Omega - \int_{\Omega} f \cdot s d\Omega$$

In this report, we only consider zero surface flux, so the weak formulation of fiber mode vanishes to

$$\int_{\Omega} \nabla_{\perp} E_z \cdot \nabla s d\Omega = \int_{\Gamma_N} s \cdot g d\omega + \int_{\Omega} (k^2 - \beta^2) E_z \cdot s d\Omega - \int_{\Omega} f \cdot s d\Omega \quad (4.7)$$

$$T(x) = u(x), x \in \Gamma$$

The equation 4.7 is the final form we are looking for, and we will use it for the simulation.

Chapter 5

Gmsh

Sfepy provides several mesh examples which can be simply modified inside Sfepy. But for fiber and toroid, Sfepy cannot generate the ideal mesh, since Sfepy support various mesh file format, we use Gmsh– an open source software that can generate finite elements mesh.

5.1 Introduction

Gmsh is a 3D finite element mesh generator, which architecture is based on four modules: Geometry, Mesh, Solver, and Post-processing. In this simulation, we only use the first two modules to generate a medit mesh file(.mesh). In the following section, I will show the detailed process of how to mesh a fiber profile.

5.2 Geometry

Gmsh provides a built-in CAD kernel for defining geometry. In Gmsh, it defines model entities using Boundary Representation(BRep). There are four main topological items: vertices, edges, faces, and volumes. An edge/curve is bounded by two points, and a face is bounded by a set of edges; a volume is bounded by a series of faces[29]. Gmsh uses a bottom-up manner to built geometry. To generate a mesh showing

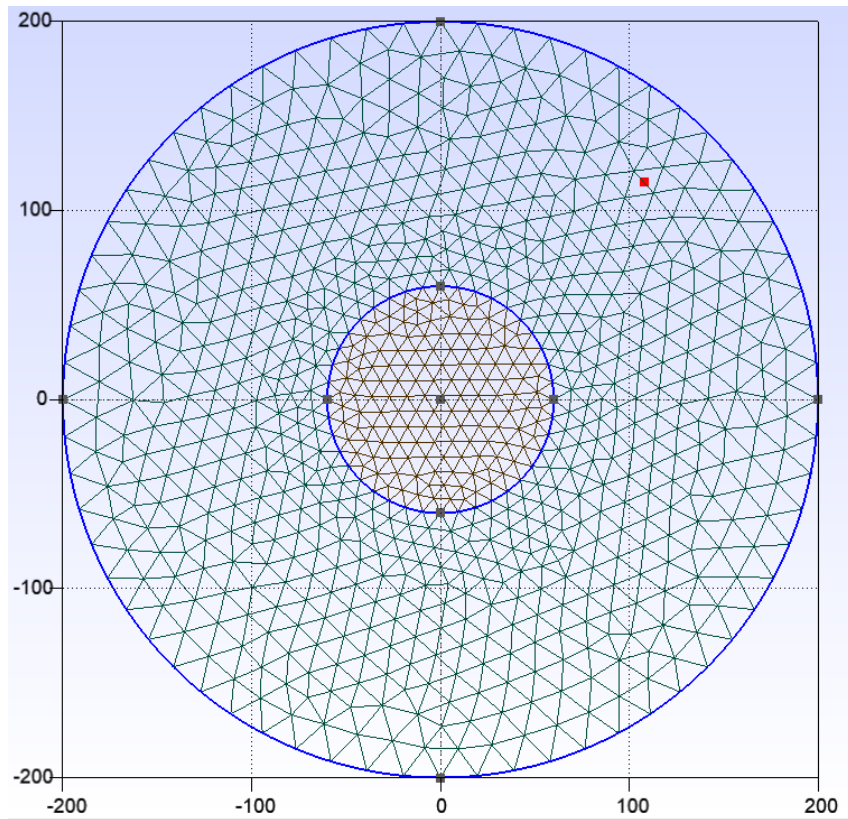


Figure 5.1: This is a mesh on the cross section of fiber with cladding using Gmsh

below, we need to use Gmsh script language:

1. Create all the points we need along the circumference of the inner and outer circle and the origin (generally we need 3-4 points to create the circle) using the following syntax.

$$\text{Point}(\text{number}) = \{x, y, z, \text{mesh_size}\}$$

Specifically we need to claim the mesh size in the point definition. For example, if we need a more accurate calculation in the fiber core, we need to set a larger mesh size.

2. After defining the points, then we connect two points by circle arc to form the circle using:

$$\text{Circle}(\text{number}) = \{\text{start}, \text{center}, \text{end}, \langle \dots \rangle\}$$

where 'start' and 'end' represent the number of start point and end point which we define above.

3. Then extrude the line and connect the lines to form a surface:

$$\text{Plane surface}(\text{number}) = \{\text{expression_list}\}$$

where expression_list is the list of number of which lines we need.

After defining the geometry, there is one important thing we need to finish—define physical groups. Unlike geometry, physical groups are the combination of physical entities (surfaces or volumes). The physical groups are essential parts in boundary condition definition. We must claim all the physical groups which we want to apply boundary condition using:

$$\text{Physical Surface}(\text{"name"}, \text{numerical ID}) = \{\text{expression_list}\};$$

Where 'numerical ID' is the user defined number and SfePy will use numerical ID to implement boundary conditions.

5.3 Finite Element Mesh

Mesh is the key component of any software for solving partial differential equations. Generally, we represent the mesh using the following concepts: mesh entity, topology, geometry, connectivity. Mesh topology means how the mesh entities are composed; mesh geometry represents how the mesh is inserted in the space. Mesh topology can be defined by the combination of mesh entities and connectivity. Mesh entity (Fig 3.2) is usually an index pair (d, i), where d represents the dimension, like 0 for the vertices, 1 for the edge, 2 for the face, etc. After defining the physical group, the Gmsh will only output mesh that belongs to at least one physical group. So, in this case, physical groups cannot overlap with each other; otherwise, there will be multiple meshes on certain faces. There are two types of meshes in FEM, 2D meshes and 3D meshes. The topology of 2D meshes is usually 2-3 and 2-4, i.e., quadrilaterals

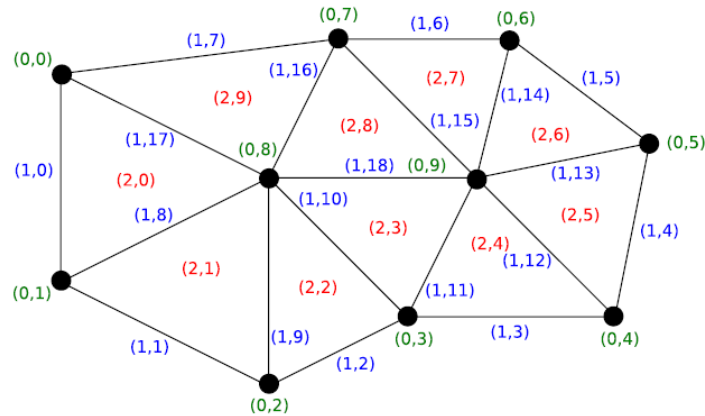


Figure 5.2: This is the mesh entity illustration, $(0,i)$ represents vertex entity, $(1,i)$ represents edge entity and $(2,i)$ represents face entity

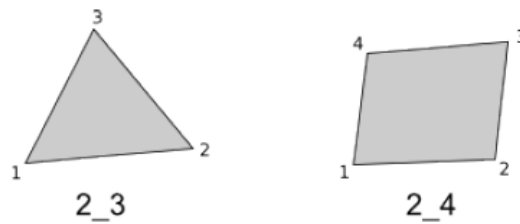


Figure 5.3: Two dimensional finite element meshes

and triangles shown in figure(3.2). And the topology of 3D meshes is generally 3-4 and 3-8, i.e. hexahedra, tetrahedra shown in figure(3.3). I will detail the meshes in the next since our mode solver only supports these four mesh topologies. Our mode solver only focuses on 2D scalar FEM, so in this report, we selected triangle meshes that achieved better discretization than quadrilaterals.

Gmsh provides a powerful engine for mesh generation. Let's consider some key parameters of the mesh generator.

5.4 Mesh Size and Quality Measurement

Mesh generation is the key component of FEM. A good mesh can save much computation time and guarantee better simulation results. Let's take an optical fiber as an example. Figure 5.5 is the electromagnetic field simulation of a fiber cross-section.

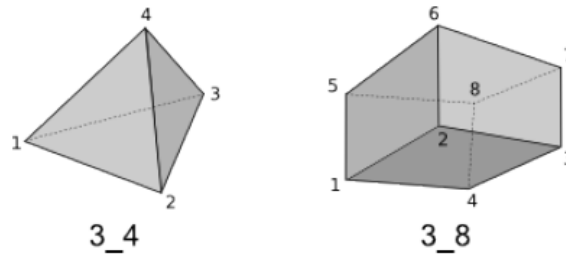


Figure 5.4: Three dimensional finite element meshes

We notice that the energy almost concentrates on the core of fiber; there is hardly any energy inside the fiber cladding. So, in this case, we know the field inside the fiber cladding is zero almost everywhere, and if we put a fine mesh in the cladding part, the computer will waste a huge amount of time only to get the same results. If we put a loose mesh on it for the fiber core, our mode solver will get inaccurate results; it even cannot distinguish the energy difference inside the core. Therefore an appropriate mesh needs to be generated to get the best result.

In order to achieve this, let's define a function called mesh size field $\theta(x, y, z)$, which defines the target mesh-size at the point. Currently, Gmsh has four methods to define the size field[30]:

- mesh size which is prescribed at vertices and interpolated on edges,
- mesh size which is prescribed by mesh grading on the edges,
- mesh size which depends on other mesh (for example, the background mesh)
- mesh size, which adapts to the primary curvature of entities.

The size fields can be carried out by several functionals, which probably depend on user predefined functions or distance between entities. If several size fields are carried out, Gmsh will select the minimum amongst all the size fields. And now let's denote the edge of the mesh e , and we define the length of edge l_e in term of size field θ as:

$$l_e = \int_e \frac{1}{\theta(x, y, z)} dl \quad (5.1)$$

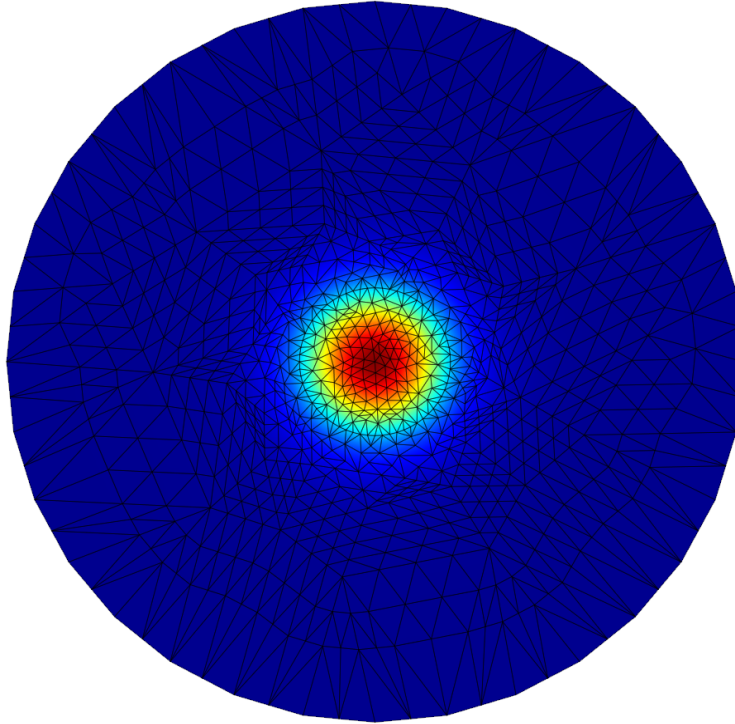


Figure 5.5: Mesh and EM field profile of a fiber cross section

And the goal of mesh generation is:

- Generate the mesh that satisfies each length of edge is approximate equal to $l_e = 1$,
- Generate the mesh that satisfies each element K is well-shaped.

In nutshell, the general goal of mesh generation is to create a good quality mesh in terms of mesh size fields. Therefore, we need define an efficiency parameter ϵ which can evaluate the relation between mesh and size field θ :

$$\epsilon = \exp\left(\frac{1}{N} \sum_{e=1}^N \epsilon_e\right) \quad (5.2)$$

where $\epsilon_e = l_e - 1$ if $l_e < 1$ and if $l_e \geq 1$, $\epsilon_e = \frac{1}{l_e} - 1$, therefore the efficiency parameter ϵ_e should range between $\epsilon_e \in [0, 1]$, and the efficiency parameter should be as close as

possible to 1. And to measure the quality of each element, several methods have been proposed in lectures[31][32], here we choose the method using element radius ratios, for examples:

- If element M is a triangle, we define the element radius ratio as:

$$\alpha_M = \frac{4 \sin a \sin b \sin c}{\sin a + \sin b + \sin c} \quad (5.3)$$

where a,b,c are three inner angles of triangle, and if M is an equilateral triangle, $\alpha_M = 1$, if M is a degenerated triangle, $\alpha_M = 0$.

- If element M is a tetrahedron, we define the element radius ratio as:

$$\alpha_M = \frac{6\sqrt{6}V_M}{(\sum_{i=1}^4 s(f_i)) \max_{i=1,2,\dots,6} l_i} \quad (5.4)$$

where V_M is the volume of tetrahedron M, $s(f_i)$ is the surface area of i^{th} face, and l_i is the length of i^{th} edge of tetrahedron M.

Chapter 6

Fiber Mode Simulation

In this chapter, I will present a detailed fiber mode simulation using the finite element mode solver. Fiber as a popular medium for communication has been developed decades, and for now, it has been wildly used in our lives; many cities use fibers for internet transmission.

6.1 Structure

The most used fiber in our life is Step-Index fiber. As shown in Fig 4.1, the step-index fiber comprises core fiber and cladding, and sometimes it's covered with buffer coating. The core and cladding have different refractive index, and core refractive index n_{core} is little greater than cladding refractive index $n_{cladding}$. So when the light incident at a certain angle, it will get total internal refraction between the interface of core-cladding and cladding-air; therefore, the light will transmit along with the fiber with zero attenuation.

There are two types of step-index fiber wildly used, one is single mode fiber, the other is multimode fiber. Before we talk about this two fiber, there is a key parameter to be claim: V number. Let's first define the index contrast:

$$\Delta = \frac{n_{core} - n_{clad}}{n_{core}}$$

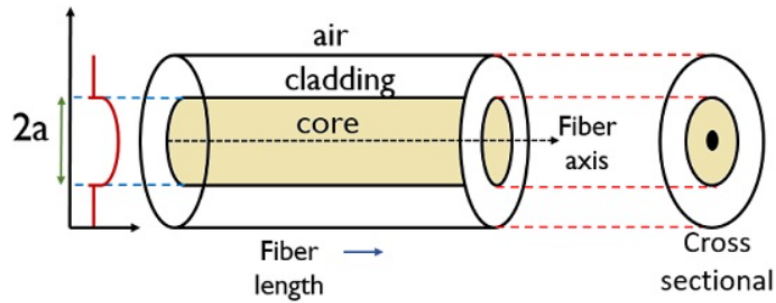


Figure 6.1: Step Index Fiber, Optical Fiber Communication[1]

then the V number can be defined as:

$$\begin{aligned}
 V &= \frac{2\pi a}{\lambda} \sqrt{n_{core}^2 - n_{clad}^2} \\
 &\approx \frac{2\pi a}{\lambda} n_{core} \sqrt{2\Delta}
 \end{aligned}
 \tag{6.1}$$

where λ is the free space wavelength, and a is the radius of fiber core. With V number, we can define the number of mode can transmit along the fiber simultaneously:

$$M \approx \frac{V^2}{2}
 \tag{6.2}$$

So we can see that with different core radius, a different number of modes can propagate inside the fiber. Based on this background, now we define two types of step-index fiber: with $V \leq 2.405$, the fiber is the single-mode fiber, which only allows one mode to propagate at one time, and with $V \geq 2.405$, the fiber is multimode, which allows several modes to propagate at the same time. And we can see that single-mode fiber has a smaller core and multimode fiber has a greater core, shown in Fig 4.2 In our case, we will simulate a multimode fiber with *core diameter* = 80um, *cladding diameter* = 400um, $n_{core} = 1.46$ and $n_{clad} = 1.44$.

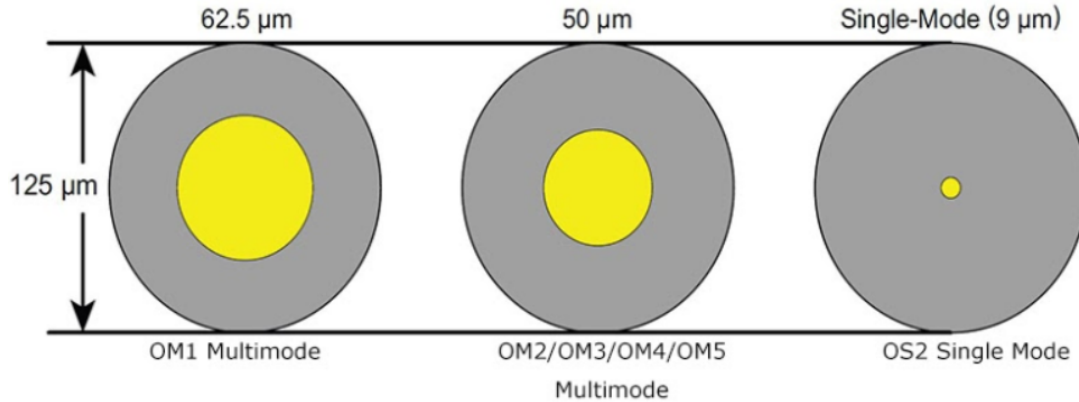


Figure 6.2: Fiber Core Diameter

6.2 Simulation

As we have derived the weak form of fiber mode equation and, for simplicity, we ignore the Neumann boundary condition and only consider Dirichlet boundary condition. For ideal condition, there is no light leakage from fiber cladding, hence we can consider the Dirichlet boundary condition as $T(x) = 0, x \in \Gamma$. For now we can vanish the equation 4.7 to the following form:

$$\int_{\Omega} \nabla_{\perp} E_z \cdot \nabla s d\Omega = \int_{\Omega} (k^2 - \beta^2) E_z \cdot s d\Omega - \int_{\Omega} f \cdot s d\Omega \quad (6.3)$$

$$T(x) = 0, x \in \Gamma$$

Using the mesh we generate in chapter 3, we need to finish the problem description file (see Appendix A) (PDF) to solve the PDEs.

In 'regions' section, 'Omega' represents the whole computational region, 'Edge' is the fiber cladding edge.

In 'materials' section, 'm' and 'k0' is constants for code simplicity, 'mat_v' is the refractive index, here I defined a step function 'fun_v' to assign 'mat_v'.

In 'fields' section, we define 'field_Ez' over domain 'Omega', and specify that 'field_Ez' is a scalar field with complex value.

Then we can define variables on the fields. In our case, we only need to define the

unknown field 'Ez' and test field 'v' on the 'field.Ez'. After defining the variables, we need to specify the boundary conditions. In fiber mode simulation, we only consider Dirichlet boundary conditions, which is in 'ebcs' section. Since we always consider there is no light leakage from fiber cladding, in other words, there should not be any light transmitting outside the fiber, the electromagnetic field outside the fiber is zero. Hence we define the boundary condition as 'ZeroSurface' on the 'Edge' with value 0. The 'equations' section corresponds to the weak form of fiber mode (Eq.4.7), and it's the central part of PDF. There we specify the Laplacian of 'Ez' in the weak form and 'i' for integral order over the specific domain 'Omega'. In sfepy, each operator has an exclusive syntax, For example, the volume integral over dot product must be written in such syntax 'dw_volume_dot.i.Omega'.

6.3 Results and Discussion

Before simulating the mode profiles, the mesh is a key component for the calculation accuracy. A more accurate mesh always means more simulation time. As mentioned in chapter 3, we slightly increase the mesh accuracy over the core region. Sfepy only supports mayavi as its visualization toolkit, using mayavi we get the following results: Figure 4.3 shows the 5 lowest order fiber modes, from the left to right, they are LP_{01} , LP_{11a} , LP_{11b} , LP_{21a} , LP_{21b} , the V number is 47. We can see that the simulated mode profiles using sfepy match fiber mode theory quite well. And the results generated by Sfepy match quite satisfying with [33], therefore the open-source mode solver achieves desirable results.

For analysis purposes, multiple simulations have been done in this section. Similarly, different step-index fiber simulations have been finished, and Sfepy calculates the effective refractive index (n_{eff}) for all modes using FEM. The effective refractive index is the main parameter of analysis since each mode has a unique effective refractive

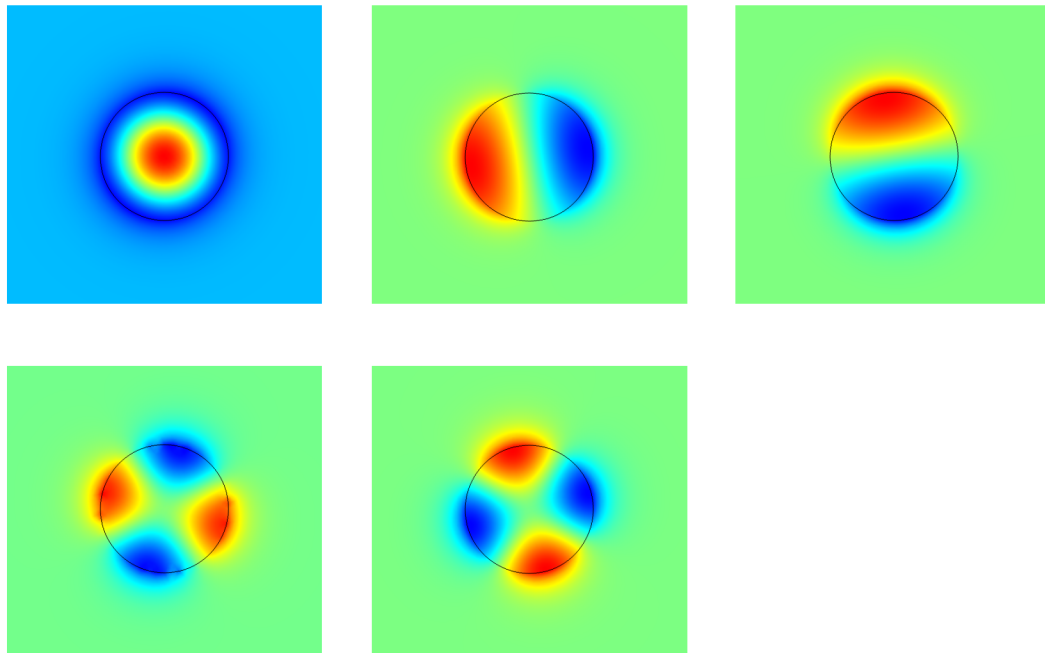


Figure 6.3: 5 variations of the lowest order step-index fiber mode

index, and we here define normalized propagation constant (b) as:

$$b = \frac{n_{eff}^2 - n_{clad}^2}{n_{core}^2 - n_{clad}^2} \quad (6.4)$$

therefore the normalized propagation constants are unique for each specific fiber mode. The step-index fibers are simulated at 1550nm, which is the standard optical communication wavelength, and below I show three types of fibers, and each fiber is simulated in SfePy and analyzed in Matlab.

Table 6.1: Waveguide parameter in different step-index fibers

Fiber type	Core radius (μm)	Core refractive index	NA	V number
1	2.5	1.466	0.2	2.029
2	6.25	1.462	0.15	3.800
3	12.5	1.462	0.15	7.601

The three different fibers with different core radius are chosen in this section. Which we can see that fiber 1 is a single-mode fiber and fiber 2,3 are multimode fibers.

Power contribution is the key parameter in fiber mode analysis. Most of the energy will concentrate on the fiber core for single-mode fiber, and it'll also have some evanescent field in fiber's cladding. For practice purposes, we desire the energy to concentrate on the core as much as possible to obtain the maximum power transmission ratio. And for multimode fiber, each mode has its own energy percentage. To analyze power contribution, let's define some parameters:

$$\begin{aligned} V &= \frac{2\pi}{\lambda} \cdot a \cdot NA \\ U &= V\sqrt{1-b} \\ W &= V\sqrt{b} \end{aligned} \quad (6.5)$$

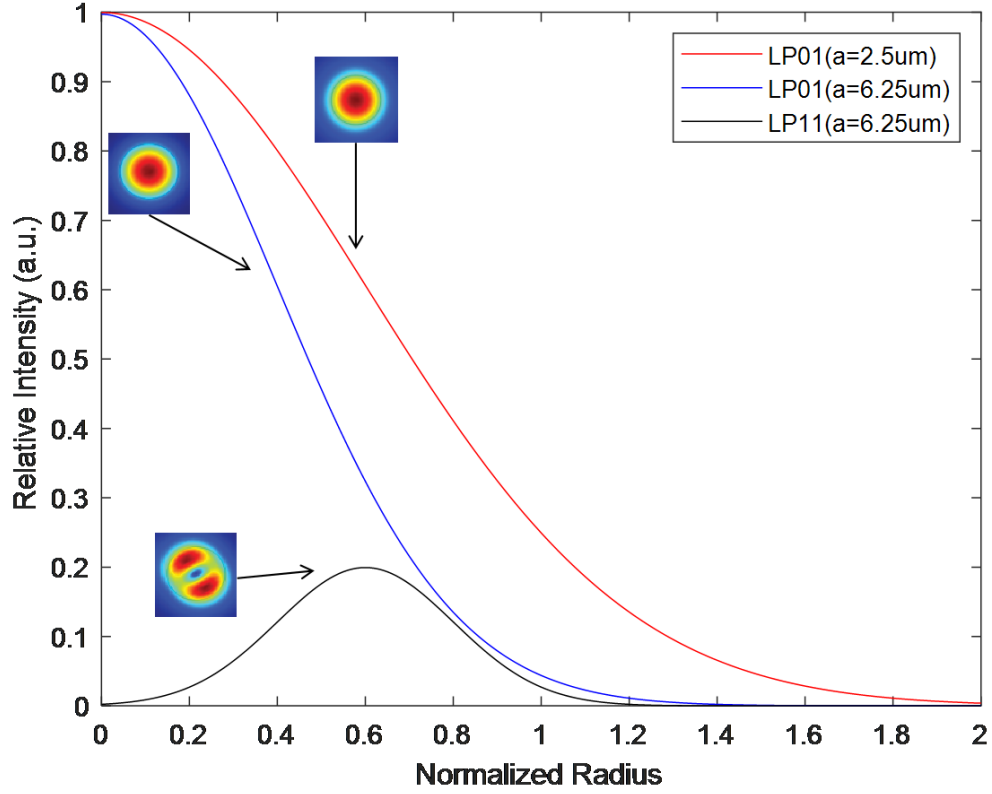


Figure 6.4: Relative Intensity profiles of different modes of fiber 1 and 2

And the mode inside the core is linearly polarized (LP) mode, which is usually referred to as LP mode. From [34], the power inside the core and cladding is given by:

$$\begin{aligned}
 P_{core} &= C\pi a^2 \left[1 - \frac{J_{l-1}(U)J_{l+1}(U)}{J_l^2(U)} \right] \\
 P_{clad} &= C\pi a^2 \left[\frac{K_{l-1}(W)K_{l+1}(W)}{K_l^2(W)} - 1 \right]
 \end{aligned} \tag{6.6}$$

where $J(U)$ and $K(W)$ are Bessel function, C is a constant. From the above formulations, we can calculate the power contribution easily. In figure 6.5, the relative intensity of different modes of fiber 1 and 2 are plotted in terms of normalized radius (distance from core center), which implies that the core-cladding is the value 1 at x-axis. From figure 6.5, there is only the fundamental mode LP_{01} presented for fiber 1 since the

fiber 1 is the single-mode fiber, only fundamental mode can propagate inside the fiber, and for fiber 2 a higher order mode LP_{11} is presented with the fundamental mode LP_{01} . And SfePy generated mode profiles are also presented with the intensity profile. And we can see that the evanescent field (cladding power) in fiber 1 is much higher than fiber 2 since fiber 1 is weakly guided. And therefore fiber 2 has more power concentrated inside the core since it's strong guidance. However fiber 1 has a higher beam quality than fiber 2 due to the higher order mode LP_{11} of fiber 2.

In table 6.2, fiber 1 has only fundamental mode propagating along the fiber. And the fundamental mode has 77.53% power in the core, the rest 22.47% power is in the cladding in the form of evanescent field.

Table 6.2: Power contribution of fiber type 1

Mode	Core Power (Watt)	Core Power (in percentage)
LP_{01}	0.7753	77.53%

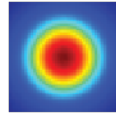
And table 6.3 and 6.4 are fiber 2 and 3. From table 6.3, we can see that 89.79% of the whole power is in the core where 65.90% of the whole power is the distribution of mode LP_{01} and 23.89% is the distribution of mode LP_{11} , and the rest 10.21% is in the cladding in the form of evanescent field.

Table 6.3: Power contribution of fiber type 2

Mode	Core Power (Watt)	Core Power (in percentage)
LP_{01}	0.6590	65.90
LP_{11}	0.2389	23.89

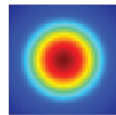
From table 6.4, we can see that 95.3% of the whole power is inside the core, and the only 4.7% is in the cladding in the form of an evanescent field. Figure 6.6 lists all

Fiber 1

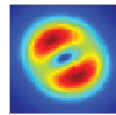


LP01

Fiber 2

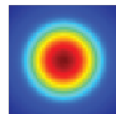


LP01

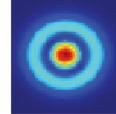


LP11

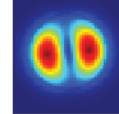
Fiber 3



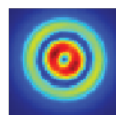
LP01



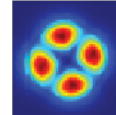
LP02



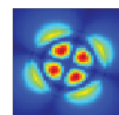
LP11



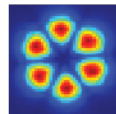
LP12



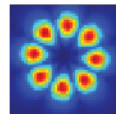
LP21



LP22



LP31



LP41

Figure 6.5: Sfevy generated mode profiles of different step-index fiber

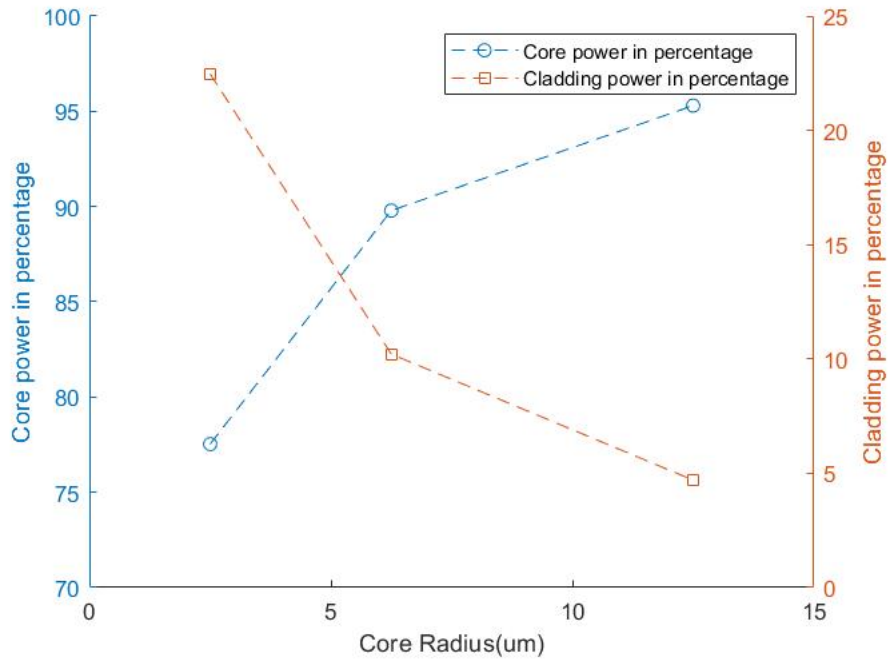


Figure 6.6: Relation between core power and cladding power over increasing core radius

mode profiles for different types of step-index fiber.

Table 6.4: Power contribution of fiber type 3

Mode	Core Power (Watt)	Core Power (in percentage)
LP_{01}	0.5653	56.53%
LP_{11}	0.1379	13.79%
LP_{21}	0.0674	6.74%
LP_{02}	0.0577	5.77%
LP_{31}	0.0408	4.08%
LP_{12}	0.0329	3.29%
LP_{41}	0.0283	2.83%
LP_{22}	0.0227	2.27%

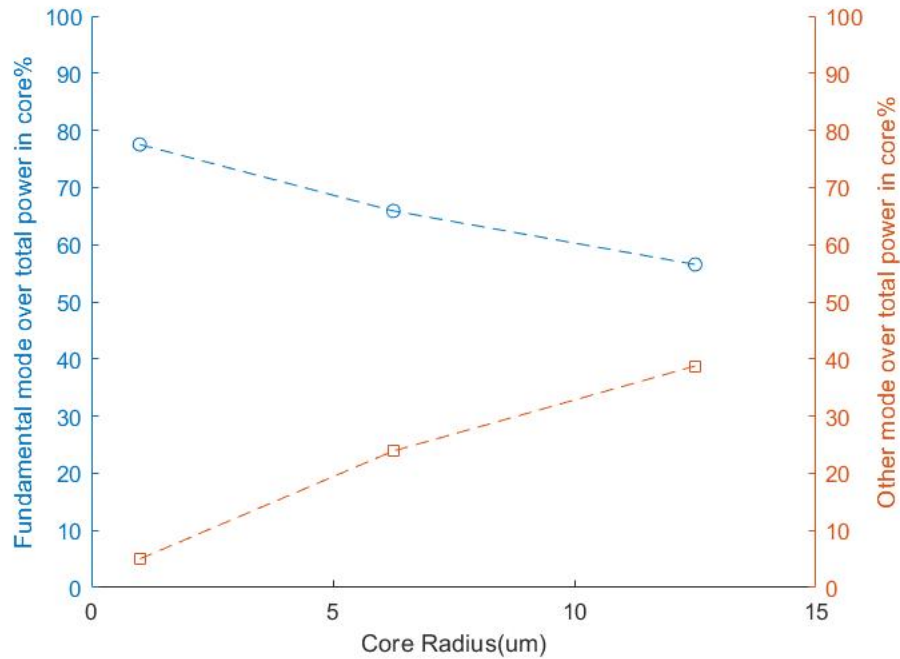


Figure 6.7: Relation between fundamental mode and other mode over increasing core radius

Based on table 6.3 and table 6.4, we can notice that for a specific fiber, the fundamental mode LP_{01} governs the main power over higher-order modes, and with the increasing core radius, every type of fiber has more power confined in its core and less power in its cladding. For fiber 1, only 77.53% of total power is confined in the core compared to the strongly guided fiber (type 2 and 3) with 89.79% and 95.3% of total power respectively confined in their cores (Figure 6.7).

We can see that due to the increasing core radius, for a fixed wavelength, the number of modes for multimode fiber is increased. This is because, for a specific wavelength, the increasing core radius will result in an increasing V number. Therefore, the number of modes (M) which we mentioned in section 6.1 will increase. Since there are more modes in the core, the fundamental mode will dominate less power over total power (shown in figure 6.8). In conclusion, the strongly guided fiber has more power confined in the core while it's multimoded; the weakly guided fiber has less power

confined in the core with a more evanescent field in its cladding. Therefore to increase the power efficiency of the fiber, it's important to increase the core area, which we usually call this type of fiber as Large Mode Area (LMA) fiber.

Chapter 7

Whispering Gallery Mode Simulation

The famous whispering gallery wave phenomenon was first discovered in St Paul's Cathedral by Lord Rayleigh in 1878. And this phenomenon attracted more and more attention with the development of science. In this chapter, we will discuss the process of simulating whispering gallery mode (WGM) using our finite element solver.

7.1 Structure

With growing interests in whispering gallery phenomena, modern science has developed a broad range of applications, especially optical communication. A widely used structure—optical microcavity (Figure 5.1(c)), also known as toroid, has been well-studied for many years. As showing in Figure 5.1, Toroid microcavities confine the incident light into a small cavity, using the total internal reflection over the toroid surface that satisfies the resonance condition (the incident light returns to the original phase and position after one round trip and interfere cohesively with itself to form standing waves). Therefore a little power of incident light can generate high power output due to resonant recirculation inside the toroid.

Like optical microcavity's acoustic analog, the whispering gallery also has a diameter-dependent

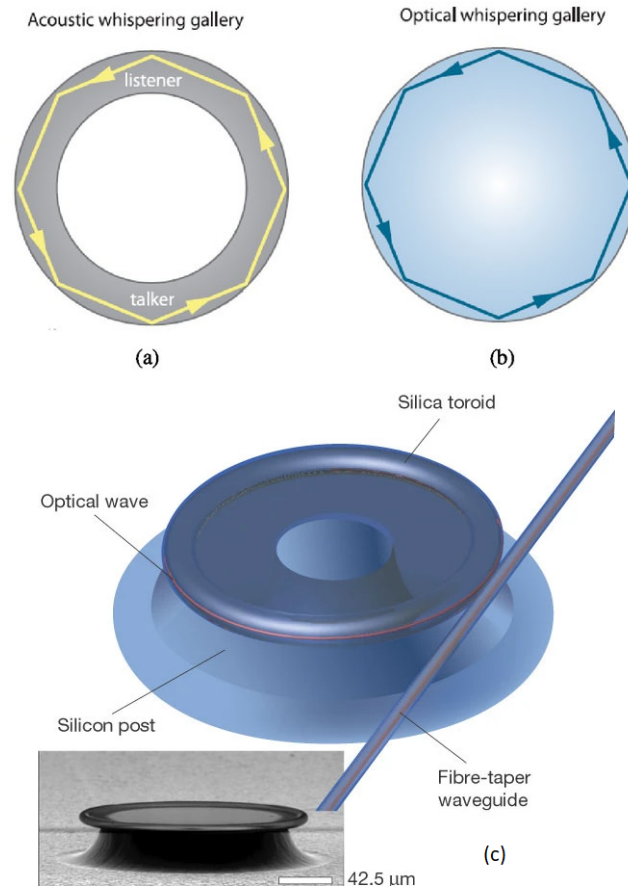


Figure 7.1: The upper left (a) is the illustration of acoustic whispering gallery, the upper right (b) is optical whispering gallery, and (c) is an ultrahigh-Q optical microcavity resonator structure[2]

frequency spectrum of resonance. Microscopic-size volume guarantees the resonant frequencies can be more sparsely scattered along the spectrum than its corresponding macroscopic-size resonator[35].

The study of optical microcavities is really active and high interest in applying WGM and resonator in the research because of their special characteristic—ultrahigh Q-factor, the capacity to operate on optical communication frequencies, the convenience of on-chip integration and fabrication. In the following section, I will discuss the theory of whispering gallery mode and the simulation results using our finite element mode solver.

7.2 Simulation

Before we discuss the simulation results, let's first take a look at the theory of whispering gallery mode and the derivation of weak form.

The basic scalar wave equation can be expressed by:

$$\nabla^2 u - \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} = f(\vec{r}, t) \quad (7.1)$$

where

$$u$$

is the wave field amplitude, $c(\vec{r})$ is the wave speed and f is the wave field sources. By introducing ansatz method:

$$u(\vec{r}, t) = \psi(\vec{r})e^{-i\omega_\zeta t} = \psi(\vec{r})e^{-iRe(\omega_\zeta)t} \cdot e^{Im(\omega_\zeta)t}$$

since $Re(\omega_\zeta) \geq 0, Im(\omega_\zeta) \neq 0$, we notice that the wave dissipates in time, and $Im(\omega_\zeta) \ll Re(\omega_\zeta)$, so the wave will dissipate quickly into free wave equation[36], i.e. $f(\vec{r}, t) = 0$, thus we can obtain the Helmholtz equation:

$$\nabla^2 \psi_\zeta + \frac{\omega_\zeta^2}{c^2} \psi_\zeta = 0 \quad (7.2)$$

As we know, the Helmholtz equation is an eigenvalue problem with the eigenvalue ω_ζ and the eigenfunction ψ_ζ . Based on the specific boundary conditions the Helmholtz equation solution can be expressed by an set of orthogonal eigenfunction ψ_ζ and their corresponding eigenvalues ω_ζ , sometimes we consider $\omega_\zeta \in \mathbf{C}$ as the eigenvalues, and ζ as mode number.

Then let's take a look at 2D scalar waves in the whispering gallery microcavity, i.e., 2D cylindrical microcavity with radius a . Assuming only the speed of wave changes on the edge of cavity, so that $c(\vec{r})$ inside the microcavity equals to c_1 and $c(\vec{r})$ outside the microcavity equals to c_0 , and therefore we denote $n = c_0/c_1$ as refractive index.

Then we try to solve the eigenfrequency and eigenmode of the microcavity.

First let's denote the wavenumber $k_{n,\zeta} = \omega_\zeta/c_n$ and $r = |\vec{r}|$, then we get

$$\begin{aligned} \nabla^2 \psi_\zeta + k_{1,\zeta}^2 \psi_\zeta &= 0, r < a \\ \nabla^2 \psi_\zeta + k_{0,\zeta}^2 \psi_\zeta &= 0, r > a \end{aligned} \quad (7.3)$$

and we need to impose some boundary conditions on these PDEs in order to get specific solutions,

1. ψ_ζ is the amplitude of the wave, it should be finite everywhere, and since we consider the space outside the cavity is free space, there should be no wave at the infinite boundary of free space, so that $r \rightarrow \infty, \psi_\zeta \rightarrow 0$.
2. ψ_ζ and $\nabla^2 \psi_\zeta$ should satisfy continuity on the boundary of microcavity at $r = a$

Based on the boundary conditions (1), and the method of separation of variables in the cylindrical coordinate, using Bessel function J_m and Hankel functions $H_m^{(1)}$, where $H_m^{(1)}(kr) \frac{1}{\sqrt{kr}} e^{ikr}$, we can get the solution of PDEs 5.3:

$$\psi_\zeta(\vec{r}) = e^{im\phi} \cdot \begin{cases} A_\zeta \cdot J_{|m|}(n, k_{0,\zeta}, r), r < a \\ B_\zeta \cdot J_{|m|}(n, k_{1,\zeta}, r), r > a \end{cases} \quad (7.4)$$

where A_ζ and B_ζ can be determined by boundary condition (2):

$$\frac{B_\zeta}{A_\zeta} = \frac{J_{|m|}(nk_{0,\zeta}a)}{H_{|m|}^{(1)}(k_{0,\zeta}a)}, \frac{J'_{|m|}(nk_{0,\zeta}a)}{J_{|m|}(nk_{0,\zeta}a)} = \frac{H_{|m|}^{(1)'}(k_{0,\zeta}a)}{nH_{|m|}^{(1)}(k_{0,\zeta}a)}$$

where the prime represents the ordinary differentiation and we can see that this is a transcendental function and can be only solved numerically. The solution consists of infinite roots for each $m \in \mathbf{Z}$. Since our task is to simulate whispering gallery mode profile using our mode solver, we will omit some analysis of prominent characteristics of microcavity, and focus on how to built our problem discription files.

Similar as configuration of fiber mode in chapter 4, we still consider a simple boundary

condition, that is the EM wave will vanish in the infinity ($r \rightarrow \infty, u(\vec{r}, t) \rightarrow 0$). And we still need to build the whispering gallery mode.

For simplicity, Let's consider the EM wave equation in the cylindrical coordinate:

$$\left[\frac{1}{\rho} \frac{\partial}{\partial \rho} \left(\rho \frac{\partial}{\partial \rho} \right) + \frac{1}{\rho^2} \frac{\partial^2}{\partial \Phi^2} + \frac{\partial^2}{\partial z^2} + k_0^2 n^2(\rho, z) \right] \vec{E}(\rho, z, \Phi) = 0 \quad (7.5)$$

here we still using the method of separation of variables, we denote:

$$\vec{E}(\rho, z, \Phi) = \vec{E}(\rho, z) e^{im\Phi}, m = m_r + im_i, m_r \text{ is an integer.}$$

and we can get the mode equation of whispering gallery as:

$$\left[\frac{\partial^2}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial}{\partial \rho} + \frac{\partial^2}{\partial z^2} + (k_0^2 n^2(\rho, z) - \frac{m^2}{\rho^2}) \right] \vec{E}(\rho, z) = 0 \quad (7.6)$$

similar as we mentioned in weak form of fiber mode equation, we consider a test function $e(\rho, z)$, and multiply the two side of equation 5.6 by the test function, we can get the weak form of equation that can be solved by our mode solver:

$$\int_{\Omega} \left(\frac{\partial^2}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial}{\partial \rho} + \frac{\partial^2}{\partial z^2} \right) \vec{E} \cdot e d\Omega + \int_{\Omega} \left(k_0^2 n^2(\rho, z) - \frac{m^2}{\rho^2} \right) \vec{E} \cdot e d\Omega = 0 \quad (7.7)$$

In the next section, I will show the simulation results of whispering gallery mode.

7.3 Result and Discussion

Let's take a look at the simulation results.

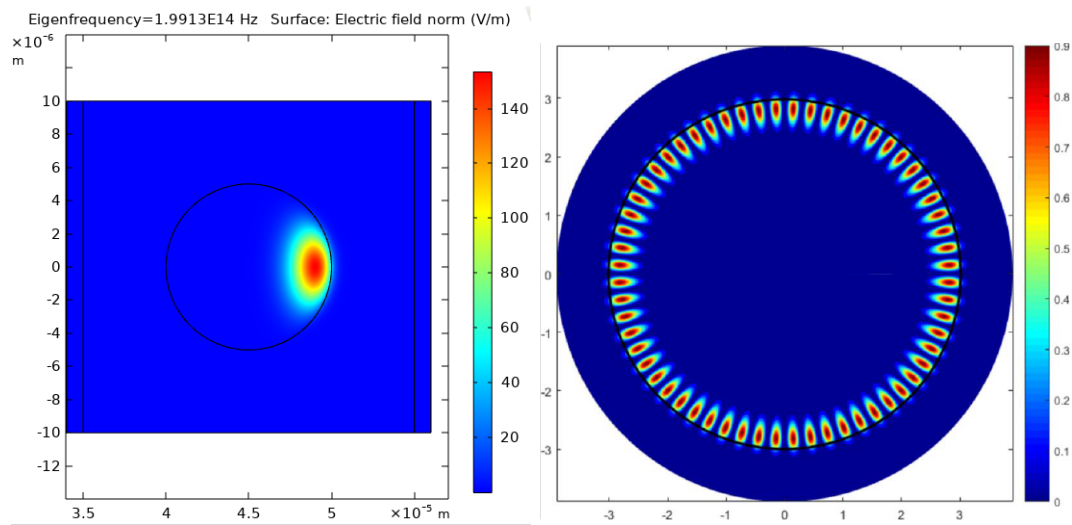


Figure 7.2: The black line indicates the physical edge of the microcavity. Note that the left figure is the cross section profile of the microcavity and the right figure is the horizontal plane of the microcavity[3].

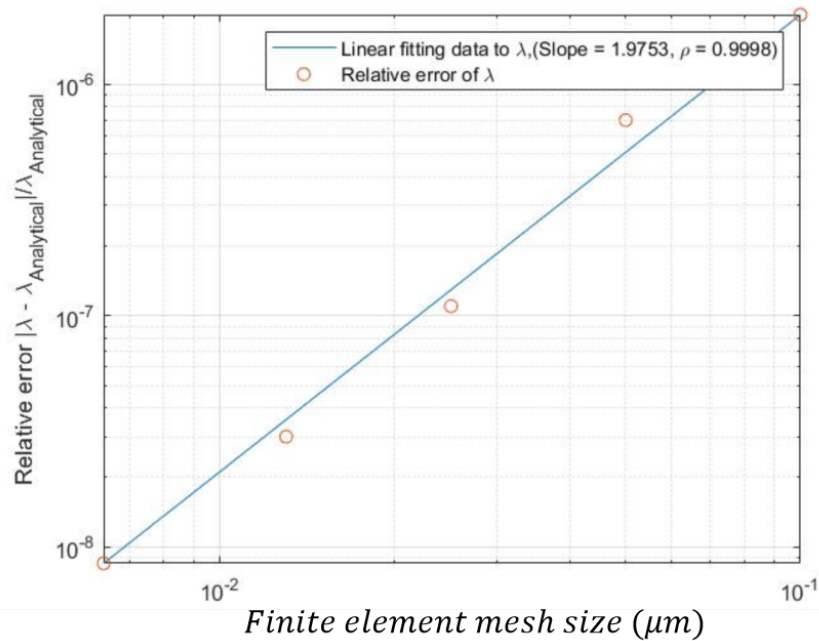


Figure 7.3: Relative error of resonant wavelength for 340th fundamental WGM in microtoroid as mesh size is halved constantly. For analytical purpose, we take $\lambda_{ref} = 780.91\text{nm}$ as the reference.

We can see that from figure 7.2, the simulation results satisfy our analysis of whispering gallery mode. Let's make a comparison with other people's simulation results. By comparing the results with [37], we can see that our simulation result works quite well compared to other's simulation.

Since SfePy and Gmsh can only support solving partial differential equations, we cannot calculate some parameters of microcavities, such as quality factor. Here we need the assistant of Matlab and COMSOL for further analysis.

Although SfePy simulation results show it matches the theory graphically, we still need to do further analysis to test if the result is correct analytically. Therefore for WGM verification, the ordinary WGM microtoroid is always a good choice. Figure 7.3 is an example of the convergence of Eigen-wavelengths for a microtoroid that is submerged in water with a radius of $30 \mu\text{m}$ and $m = 340$ [38]. And all the detail of this microgrid will be elaborated later. And there is an inherent parameter for comparing the eigenmodes of finite element mode solver to the theoretical field. Similar to

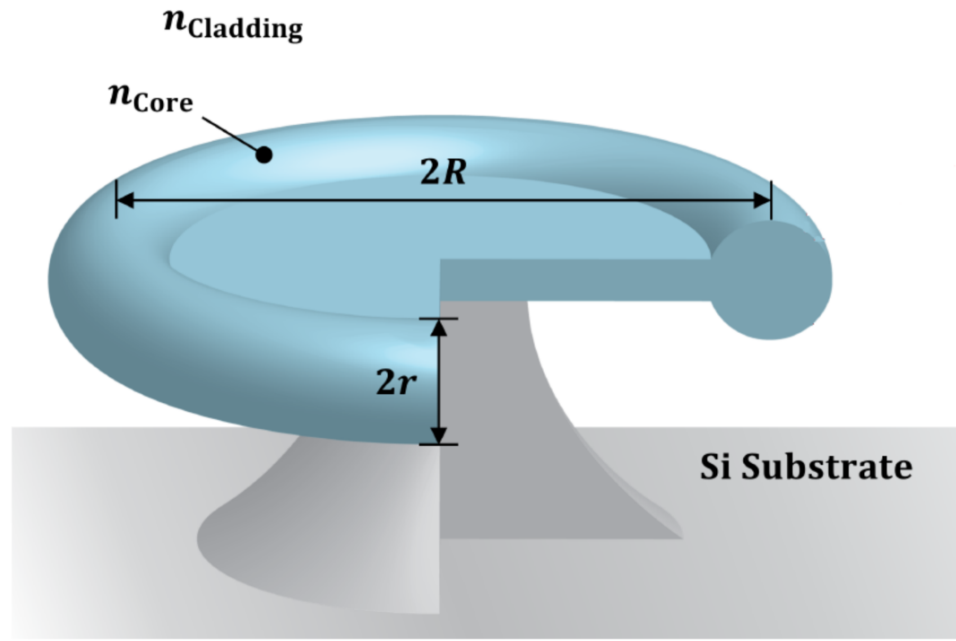


Figure 7.4: Cross section of a microtoroid which is submerged in the water with wavelength $\lambda = 633nm$

optical fiber, microtoroid also has leakage from the core to cladding (in this case, the cladding is water) comparable to the wave-decaying function over the quantum potential barrier, which is usually used to explain the particle inside a finite quantum well, where the potential field is expressed by a product of spacial function and coefficient over the in and out of the sphere.

Figure 7.4 is the illustration of a microtoroid submerged in the water with a very high degree of azimuthal and radius confinement[39] which is equivalent to the ratio between major radius and minor radius (R/r). And furthermore, by imposing a unified grid on the $z\rho$ plane, a node-by-node refractive index profile can be portrayed. With this data and simulation wavelength, we can retrieve the exact eigenvalue-eigenvector pairs.

Excitation Laser	
Working wavelength	$\lambda = 633 \text{ nm}$
Geometry	
Major radius	$R = 40 \text{ }\mu\text{m}$
Minor radius	$r = 5 \text{ }\mu\text{m}$
Aspect ratio	$\frac{R}{r} = 8$
Materials	
Core/Microtoroid: dioxide	Silicon $n_{core} = 1.457 + i(6.95 \times 10^{-11})$
Cladding/Dielectric: water	$n_{clad} = 1.332 + i(1.47 \times 10^{-8})$
Resonance	
Azimuthal mode order	$m=636$

Table 7.1: Whispering gallery mode simulation for microtoroid

We always use Helium-neon gas laser in the lab which work between the red light in the visible spectrum and high gain $633nm$. By comparing the free space wavelength with the corespondent parameter in table 7.1, we can notice that the microtoroid always resonances close to their own wavelength and using ARPACK[40] we can search some eigenvalue and eigenvector pairs with largest real part of a sparse matrix X.

And Q factor is also an important parameter in whispering gallery mode. From [41], we know that the Q-factor corresponds with wave attenuation per cycle. Denote the absorption index κ and resonance wavelength $\tilde{\lambda}$ in complex form, a primary calculation of Q is given by:

$$Q = \frac{Re(\tilde{\lambda})}{2Im(\tilde{\lambda})} \quad (7.8)$$

Choosing $R > 15\mu m$ will result absorption dominating over wave leakage [42], however smaller radius of microcavity will need the assistance of perfect match layer(PML). Figure 7.5 and 7.6 are the microtoroid submerged in water with ($R = 40\mu m$ and $r = 5\mu m$) and mesh size with 5 nm. Those figures are the indicator of correctness of simulation results of transverse electric (TE) and transverse magnetic (TM) mode in microtoroid.

In most simulations, mode solver iterations are support 2D matrix optimization. And the finite element method also do matrix decomposition to find determinants. In our simulation, it spends 51 seconds and 4 GB memory with 2.20GHz processor and 16GB RAM.

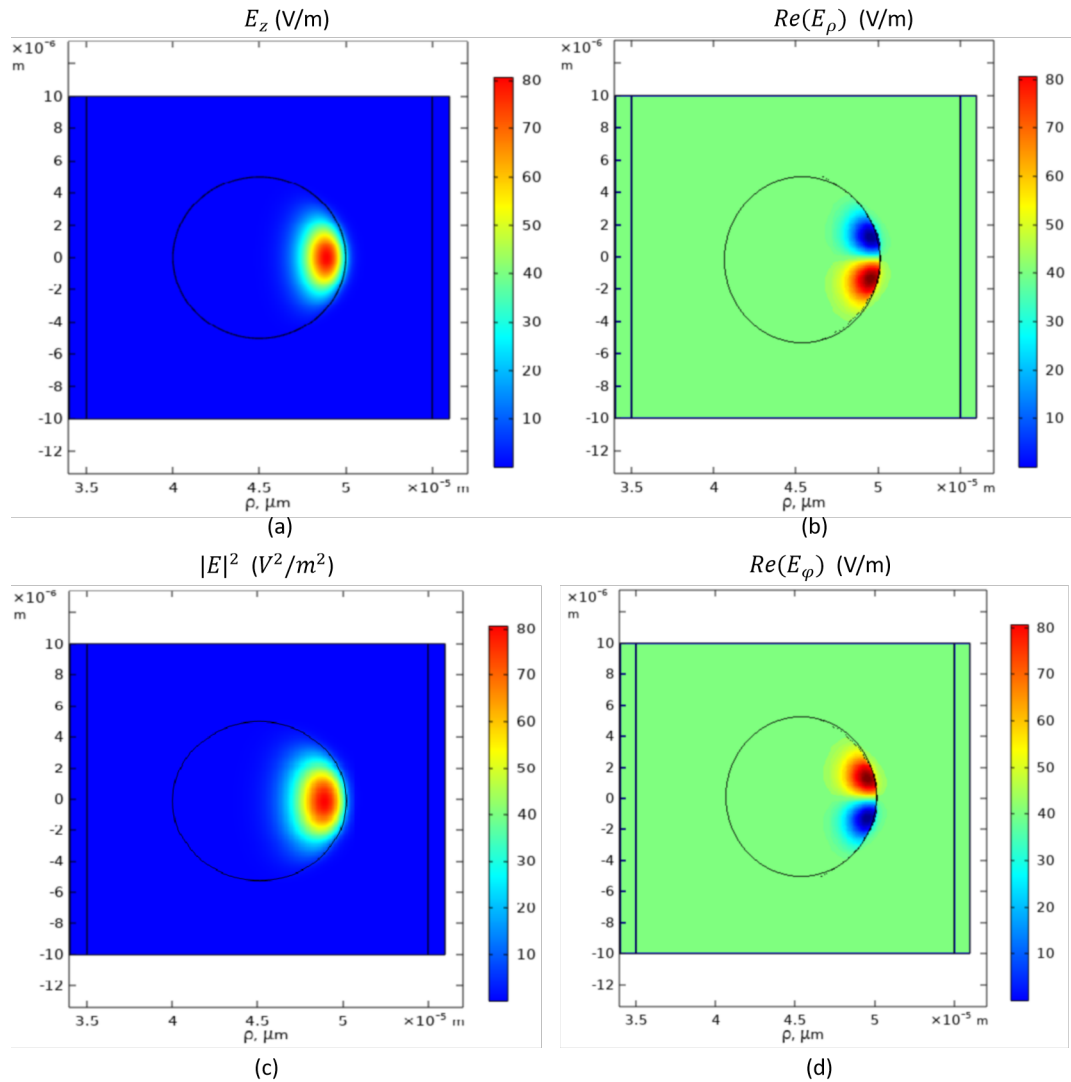


Figure 7.5: Fundamental TE mode for microtoroid submerged in water, where $m = 636$, $\lambda = 632.784$ nm, (a) is the real part of z-directed electric field, (b) is the real part of ρ -directed electric field, (c) is the whole electric field squared $|E|^2$ and (d) is the real part of ϕ -directed electric field.

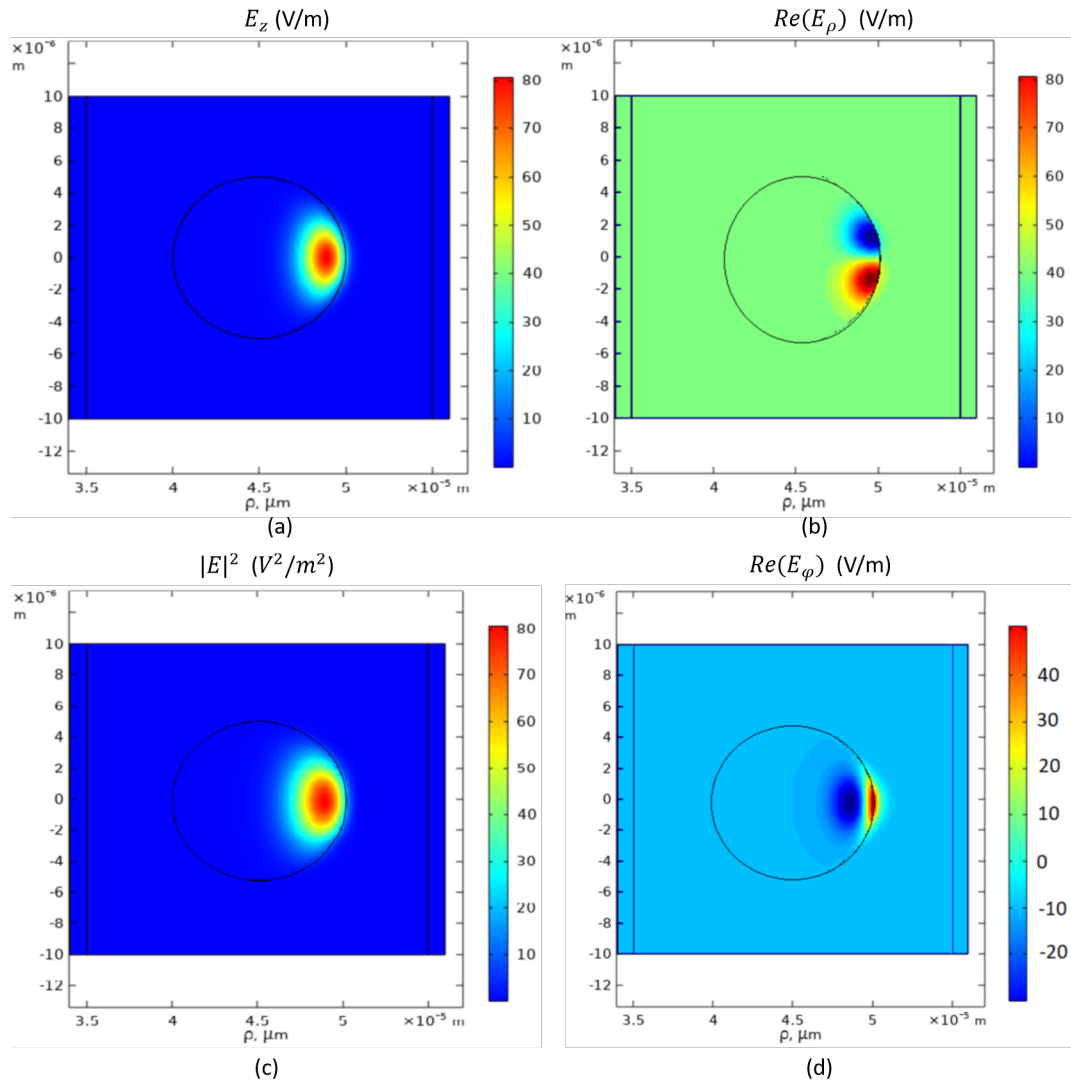


Figure 7.6: Fundamental TE mode for microtoroid submerged in water, where $m = 636$, $\lambda = 632.4$ nm, (a) is the real part of z-directed electric field, (b) is the real part of ρ -directed electric field, (c) is the whole electric field squared $|E|^2$ and (d) is the real part of ϕ -directed electric field.

Chapter 8

Conclusion

In this report, I mainly talked about using Sfepy and Gmsh to build our finite element mode solver. However, Sfepy is still in development which makes it very unstable to complete our simulation. I now realized the simulation of fiber mode and whispering gallery mode, although Sfepy may unintentionally come up with some unknown errors. Comparing to the commercial mathematical software (COMSOL, Matlab, or so), Sfepy still has a long way to go. However, since Sfepy is open-source software for the purpose of mode solver, we can develop our application using Sfepy. For most partial differential equations, Sfepy is capable of handling that theoretically.

Appendix A

Problem Description File

Here is the problem description file of fiber mode simulation.

```
region = {
    'Omega' : 'all',
    'Edge' : ('vertices_of_surface' , 'facet'),
}

material = {
    'm' : ({'val' : 1.0},),
    'mat_v' : 'fun_v',
    'k0' : ({'val' : -k0**2},),
}

functions = {
    'fun_v' : (fun_v,),
}

approx_order = 2
```

```

field = {
    'field_Ez' : ('complex' , 'scalar' , 'Omega' , approx_order),
}

variable = {
    'Ez' : ('unknown_field' , 'field_Ez' , 0),
    'v' : ('test_field' , 'field_Ez' , 'Ez'),
}

ebcs = {
    'ZeroSurface' : ('Edge' , {'Ez.0' : 0.0}),
}

integrals = {
    'i' : 2*approx_order,
}

equations = {
    'lhs' : ""dw_div_grad.i.Omega(m.val , v , Ez)
            + %s * dw_volume_dot.i.Omega(mal_v.V , v , Ez)""
            % ( - k0 ** 2 ),
    'rhs' : ""%s * dw_volume_dot.i.Omega(k0.val , v , Ez)""
            % ( - k0 ** 2 ),
}

solvers = {
    'eig' : ('eig.matlab' , {
        'method' : 'eigs' ,
        'which' : 'LM',
    })
}

```

}),
}

Appendix B

Notation

For the clarity of description, I have list all notation in this appendix and their definition.

norm vector: $\hat{\mathbf{n}}$

Gradient of f: $\nabla f = \sum_{i=1}^3 \frac{\partial f}{\partial x_i} \hat{\mathbf{x}}_i$

Laplacian operator: $\Delta = \nabla^2 = \sum_{i=1}^3 \frac{\partial^2}{\partial x_i^2}$

inner product: $\langle a, b \rangle = \int_{\Omega} ab \, d\Omega$

Curl of A: $\nabla \times \mathbf{A} = \varepsilon_{ijk} \hat{e}_i \partial_j A_k$

Divergence of A: $\nabla \cdot \mathbf{A} = \sum_{i=1}^3 \frac{\partial A_i}{\partial x_i}$

Bibliography

- [1] T. Lu and D. Yevick, “A vectorial boundary element method analysis of integrated optical waveguides,” *Journal of lightwave technology*, vol. 21, no. 8, p. 1793, 2003.
- [2] M. Oxborrow, “How to simulate the whispering-gallery modes of dielectric microresonators in femlab/comsol,” in *Laser Resonators and Beam Control IX*, vol. 6452. International Society for Optics and Photonics, 2007, p. 64520J.
- [3] A. Kaplan, M. Tomes, T. Carmon, M. Kozlov, O. Cohen, G. Bartal, and H. Schwefel, “Finite element simulation of a perturbed axial-symmetric whispering-gallery mode and its use for intensity enhancement with a nanoparticle coupled to a microtoroid.” *Optics express*, vol. 21, no. 12, pp. 14 169–80, 2013.
- [4] R. Cimrman and E. Rohan, “Two-scale modeling of tissue perfusion problem using homogenization of dual porous media,” *International Journal for Multiscale Computational Engineering*, vol. 8, no. 1, pp. 81–102, 2010.
- [5] E. Rohan, B. Miara, and F. Seifrt, “Numerical simulation of acoustic band gaps in homogenized elastic composites,” *International Journal of Engineering Science*, vol. 47, no. 4, pp. 573–594, 2009.
- [6] E. Rohan and V. Lukeš, “Homogenization of the acoustic transmission through a perforated layer,” *Journal of Computational and Applied Mathematics*, vol. 234, no. 6, pp. 1876–1885, 2010.

- [7] D. A. Shchuchkin, A. S. Khoroshev, and A. V. Pavlenko, “Application of gmsh + getdp software suite for calculation of non-stationary electromagnetic fields of electrical devices,” *2017 International Conference on Industrial Engineering, Applications and Manufacturing*, pp. 1–4, 2017.
- [8] R. L. Williamson, B. H. Rabin, and J. T. Drake, “Finite element analysis of thermal residual stresses at graded ceramic-metal interfaces. part i. model description and geometrical effects,” *Journal of Applied Physics*, vol. 74, no. 2, pp. 1310–1320, 1993.
- [9] G. Dhatt, E. Lefrançois, and G. Touzot, *Finite element method*. John Wiley & Sons, 2012.
- [10] G. Pelosi, “The finite-element method, part i: Rl courant [historical corner],” *IEEE Antennas and Propagation Magazine*, vol. 49, no. 2, pp. 180–182, 2007.
- [11] N. Levy, P. Marcal, W. Ostergren, and J. Rice, “Small scale yielding near a crack in plane strain: a finite element analysis,” *International Journal of Fracture Mechanics*, vol. 7, no. 2, pp. 143–156, 1971.
- [12] R. W. Clough, “The finite element method in plane stress analysis,” in *Proceedings of 2nd ASCE Conference on Electronic Computation, Pittsburgh Pa., Sept. 8 and 9, 1960*, 1960.
- [13] R. Kraft, P. Mckee, A. Dagro, and S. Grafton, “Combining the finite element method with structural connectome-based analysis for modeling neurotrauma: connectome neurotrauma mechanics,” *PLoS Comput Biol*, vol. 8, no. 8, p. e1002619, 2012.
- [14] Q. Liu, F. Liu, I. Turner, and V. Anh, “Finite element approximation for a modified anomalous subdiffusion equation,” *Applied mathematical modelling*, vol. 35, no. 8, pp. 4103–4116, 2011.

- [15] G. Archer, G. Fenves, and C. Thewalt, “A new object-oriented finite element analysis program architecture,” *Computers & structures*, vol. 70, no. 1, pp. 63–75, 1999.
- [16] O. C. Zienkiewicz and R. L. Taylor, *The finite element method, vol. 2*. Butterworth-Heinemann, 2000.
- [17] H. Melosh and A. Raefsky, “A simple and efficient method for introducing faults into finite element computations,” *Bulletin of the Seismological Society of America*, vol. 71, no. 5, pp. 1391–1400, 1981.
- [18] J. N. Reddy and D. K. Gartling, *The finite element method in heat transfer and fluid dynamics*. CRC press, 2010.
- [19] M. Zako, Y. Uetsuji, and T. Kurashiki, “Finite element analysis of damaged woven fabric composite materials,” *Composites science and technology*, vol. 63, no. 3-4, pp. 507–516, 2003.
- [20] K. Saitoh and M. Koshiba, “Single-polarization single-mode photonic crystal fibers,” *IEEE Photonics Technology Letters*, vol. 15, no. 10, pp. 1384–1386, 2003.
- [21] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*, ser. The Finite Element Method. Elsevier Science, 2013. [Online]. Available: <https://books.google.ca/books?id=7UL5Ls9hOF8C>
- [22] M. R. Foreman, J. D. Swaim, and F. Vollmer, “Whispering gallery mode sensors,” *Advances in optics and photonics*, vol. 7, no. 2, pp. 168–240, 2015.
- [23] C. Klusmann, J. Oppermann, P. Forster, C. Rockstuhl, and H. Kalt, “Identification of dielectric, plasmonic, and hybrid modes in metal-coated whispering-gallery-mode resonators,” *ACS Photonics*, vol. 5, no. 6, pp. 2365–2373, 2018.

- [24] Y. Yang, S. Saurabh, J. Ward, and S. N. Chormaic, “Coupled-mode-induced transparency in aerostatically tuned microbubble whispering-gallery resonators,” *Optics letters*, vol. 40, no. 8, pp. 1834–1837, 2015.
- [25] S. Yuan, L. Chen, Z. Wang, R. Wang, X. Wu, and X. Zhang, “Mode coupling in a terahertz multi-mode whispering-gallery-mode resonator,” *Optics letters*, vol. 44, no. 8, pp. 2020–2023, 2019.
- [26] V. Duong, R. Chen, L. Ma, Y. Jun, and D. Han, “Whispering gallery mode microlasers and refractive index sensing based on single polymer fiber,” *Laser & Photonics Reviews*, vol. 7, no. 1, pp. 133–139, 2013.
- [27] D. K. Cheng, *Fields and Waves of Electromagnetic*. Addison Wesley, second-edition, 1989.
- [28] R. Cirman, “Enhancing sfepey with isogeometric analysis,” *arXiv preprint arXiv:1412.6407*, 2014.
- [29] C. Geuzaine and J. F. Remacle, “Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities,” *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, vol. 79, pp. 1309–1331, 2009.
- [30] C. Geuzaine and J. F. Remacle, *Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities*. Grid Generation for Numerical Computations, Tetrahedron, 2010.
- [31] P. P. Pebay and T. J. Baker, “Analysis of triangle quality measures,” *Mathematics of Computation*, vol. 72, no. 244, pp. 1817–1839, 2003.
- [32] A. Liu and B. Joe, “Relationship between tetrahedron shape measures,” *BIT Numerical Mathematics*, vol. 34, no. 2, pp. 268–287, 1994.

- [33] J. X. Shi, K. X. Chen, and K. S. Chiang, “Analysis of mode-selective coupling between few-mode fibers and waveguides with lateral misalignment,” *International Society for Optics and Photonics*, vol. 10814, 2018.
- [34] T. Ghatak, *Introduction to Fiber Optics*, ser. South Asia edition. Cambridge university press, 1999.
- [35] X. Du, S. Vincent, and T. Lu, “Full-vectorial whispering-gallery-mode cavity analysis,” *Optics express*, vol. 21, no. 19, pp. 22 012–22 022, 2013.
- [36] E. Franchimon. (2010) Modelling circular optical microresonators using whispering gallery modes. [Online]. Available: <http://essay.utwente.nl/59791/>
- [37] M. Oxborrow, “Traceable 2-d finite-element simulation of the whispering-gallery modes of axisymmetric electromagnetic resonators,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 55, no. 6, pp. 1209–1218, 2007.
- [38] M. D. Baaske, M. R. Foreman, and F. Vollmer, “Single-molecule nucleic acid interactions monitored on a label-free microcavity biosensor platform,” *Nature Nanotechnology*, vol. 9, no. 11, pp. 933–939, 2014.
- [39] D. K. Armani, T. J. Kippenberg, S. M. Spillane, and K. J. Vahala, “Ultra-high-q toroid microcavity on a chip,” *Nature*, vol. 421, no. 6926, pp. 925–928, 2003.
- [40] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Method*. Philadelphia: SIAM, 1998.
- [41] K. Srinivasan, M. Borselli, O. Painter, A. Stintz, and S. Krishna, “Cavity q, mode volume, and lasing threshold in small diameter algaas microdisks with embedded quantum dots,” *Optical Express*, vol. 14, no. 3, pp. 1094–1105, 2006.
- [42] T. J. Ippenber, *Nonlinear Optics in Ultra-high-Q Whispering-Gallery Optical Microcavities*. California Institute of Technology, 2004.