IIT Delhi

PYD411: B Tech. Project Part I

Progress Report

Finite Element Analysis of Optical Waveguides

Author:
Nilaksh Agarwal
Nayantara Mudur

Supervisor:
Dr. Anurag Sharma

Abstract

Finite Element Analysis is a numerical method for solving partial differential equations across a variety of domain areas, most notably solid mechanics and structural analysis. The method includes subdividing the domain area into smaller elements, and calculating the field values at these vertices, then interpolating these values to obtain the field at all points within the space. The partial differential equations characterizing electromagnetic wave propagation thus lend themselves to an application of this method. In this project, we implemented a finite element solver for electromagnetics and applied this problem to computing the modal solutions for a different waveguides.

1 Introduction

In several problems involving electromagnetic wave propagation, exact analytical field solutions cannot be derived. In such cases, numerical methods such as the finite element method and the finite difference method allow one to segment the problem into smaller parts and to obtain a numerical solution in each of these parts. Some numerical methods typically used in photonics problems include the finite difference method and finite element analysis.

In the finite element approach (eg. [4], [3]), the domain is broken up in a structured fashion into several smaller elements and the field in each element is expressed in terms of a unique functional form of the field at a discrete number of points over the domain. The equations for each element are then assembled into a system of algebraic equations in terms of the unknowns. The FEM approach confers several advantages, including its ability to handle complex geometries and boundaries, accommodate for greater accuracy in field approximations and offer the scope for local element refinement.

In this report, we have applied this method to calculating the modal field solutions of a waveguide, considering different meshing and refractive index cases. The report is organized as follows: Section 2 covers the theory corresponding to the finite element formulation and the equations specific to the electromagnetic wave propagation context, Section 3 outlines the code for each segment, the mesh plots and the field solutions obtained for each field. Section 4 outlines future directions that would be of interest.

2 Theory

2.1 Finite Element Formulation

In the finite element scheme, the geometry of the waveguide must be broken up into several elements and the field in each element is expressed as a function of the value at the nodes of the element. The Delaunay Triangulation scheme is one such method of dividing a surface into triangles such that the minimum angle of a triangle is maximized, reducing the 'skewness' of the triangles. A unique Delaunay scheme necessarily exists for any set of points. It also possesses the property that no other point will belong to the circumcircle of any triangle.

The field at any point in an element can be represented in the form

$$N_i(x,y) = \frac{a_i + b_i x + c_i y}{2A} \tag{1}$$

$$\phi^e(x,y) = \sum_{I=1}^{n} N_I(x,y)\phi_I^e \tag{2}$$

where I runs over the nodes of the element and n=3 for a triangular mesh. The $N_I(x,y)$ is termed the shape function. One choice of the shape function is the Lagrange shape function.

The final finite element scheme results in a generalized eigenvalue problem of the form

$$[K]\overrightarrow{\phi} = \lambda[b]\overrightarrow{\phi}$$

This is of the form of a generalized eigenvalue problem. Here, $[K]$ and $[b]$ are termed the stiffness and mass matrices respectively.

2.2 Electromagnetic Wave Propagation in Waveguides

Maxwell's Equations can be decoupled into the vector wave equations:

$$\nabla^2\mathbf{E} + \omega^2\varepsilon\mu E = -\nabla(ln\varepsilon)\cdot\mathbf{E}$$

$$\nabla^2\mathbf{H} + \omega^2\varepsilon\mu H = -\nabla(ln\varepsilon)\times\nabla\times\mathbf{H}$$

Under the weakly guiding approximation and for small index contrasts, the scalar formulation can be applied, in which only the Transverse Electric (TE) or the quasi-TE modes are analyzed.

$$\nabla^2\mathbf{E} = \omega^2\varepsilon\mu E$$

$$\nabla^2\mathbf{H} = \omega^2\varepsilon\mu H$$

The rigorous implementation of the electromagnetic formulation would require the storage of six field components (three each for the electric and magnetic fields) which may not be very efficient. In certain cases, where the index contrast is sufficiently low the weakly guiding approximation is applicable [5]. In the modal solutions of waveguides, for TM modes, we solve for E_z subject to the Dirichlet Boundary condition of E=0 at the boundary. For TE modes, the field H_z is calculated with a Neumann Boundary condition of $\partial_z H_z = 0$. The Dirichlet boundary condition is implemented by zeroing out the rows of the stiffness and mass matrices ([K] and [b]) corresponding to the nodes which lie on the boundary.

For the scalar formulation, a variational L is minimized over the entire domain.

$$\mathbf{L} = \iint\limits_{\Omega} \nabla^2 E_x^2 - k_o n^2 E_x^2 + \beta^2 E_x^2 \tag{3}$$

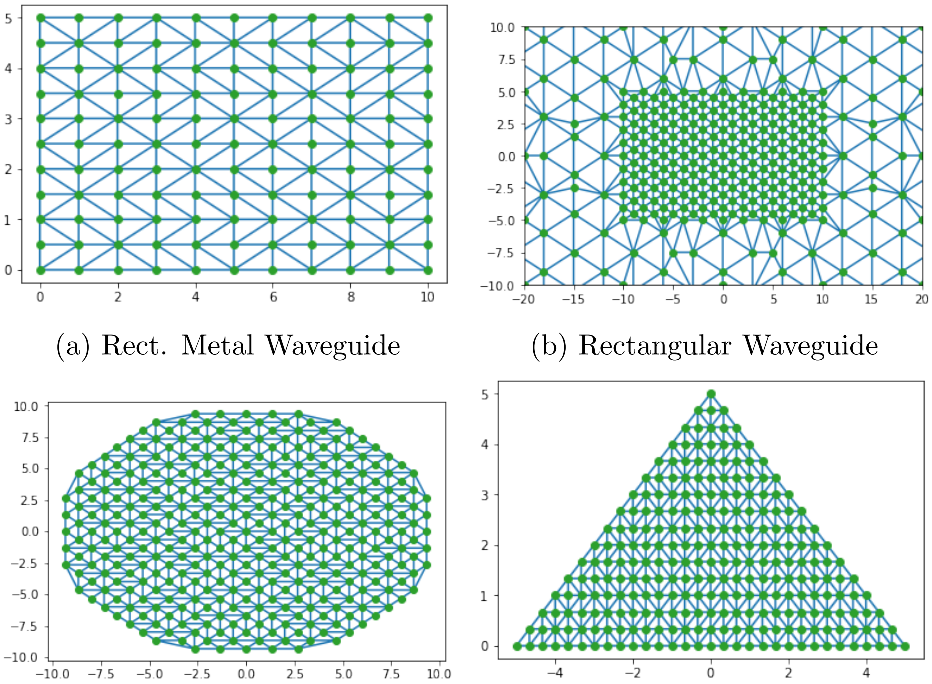
$$\mathbf{K} = \iint_{\Omega} \nabla^2 N_i^e N_j^e + \beta^2 N_i^e N_j^e \tag{4}$$

$$\mathbf{b} = \iint_{\Omega}^{e} k_o^2 n_o^2 N_i^e N_j^e \tag{5}$$

3 Code Walkthrough

3.1 Meshing

```
import numpy as np
import matplotlib.pyplot as plt

a_dim = 10.0
b_dim = 5.0

pt1 = np.array([0.0,0.0])
pt2 = np.array([a_dim, 0.0])
pt3 = np.array([a_dim, b_dim])
pt4 = np.array([0.0, b_dim])

points = []
d = 30

for i1 in range(0, d-1):
    for i2 in range(0, (d-i1)+1):
        for i3 in range(0, (d-(i1+i2)+1)):
            for i4 in range(0, (d-(i1+i2+i3)+1)):
                x = (pt1*i1 + pt2*i2 + pt3*i3 + pt4*i4)/d
                points.append(tuple(x))

points=np.array(list(set(points)))

from scipy.spatial import Delaunay
tri = Delaunay(points)

N = len(points)
M = len(tri.simplices)
```



(a) Rect. Metal Waveguide

(b) Rectangular Waveguide

(c) Circular Metal Waveguide

(d) Optical Metal Waveguide

Figure 1: Meshing Results

3.2 Generation of Mass & Stiffness Matrices

```
n     = 1.5
beta  = 2
Area  = np.zeros(M)
K_det = np.zeros((M,3,3))
b_det = np.zeros((M,3,3))
for i in range(0,M):
    temp = tri.simplices[i]
    p = [list(points[temp[0]]),
        list(points[temp[1]]),
        list(points[temp[2]])]
    a = [0,0,0]
    b = [0,0,0]
    c = [0,0,0]
    center = [0,0]
    for j in range(0,3):
        a[j] = p[(j+1)
        b[j] = p[(j+1)
        c[j] = p[(j+2)
        center[0] = center[0] + p[j][0]/3
        center[1] = center[1] + p[j][1]/3
        Area[i] = Area[i] + p[j][0]*(p[(j+1)

    Area[i] = abs(Area[i])*0.5

    for i1 in range(0,3):
        for i2 in range(0,3):
            cons = 1
            if i1==i2:
                cons = 2
            K_det[i][i1][i2] = (b[i1]*b[i2] +
                                c[i1]*c[i2])/4*Area[i]+
                                beta*beta*cons*Area[i]/12

            b_det[i][i1][i2] = n*n*cons*Area[i]/12
```

3.3 Scattering to create Global Matrices

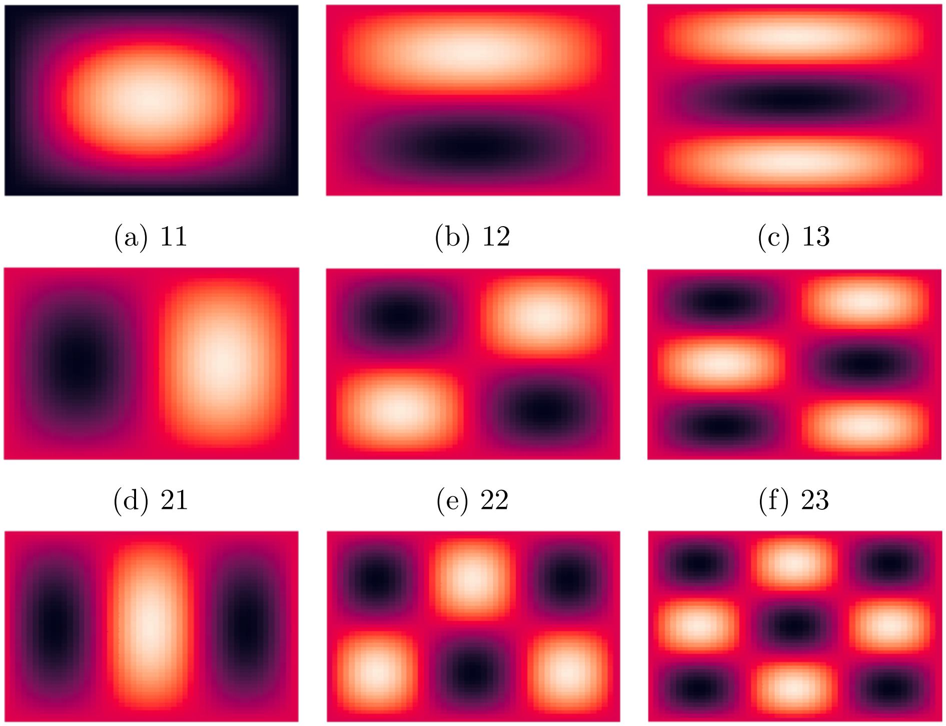
```
def assemble(element, K_loc, b_loc):
    K_globe, b_globe=np.zeros((N,N)), np.zeros((N,N))

    for el in range(len(element)):
        enodes=element[el]
        for i in range(len(enodes)):
            for j in range(len(enodes)):
                K_globe[enodes[i], enodes[j]]+=K_loc[el][i][j]
                b_globe[enodes[i], enodes[j]]+=b_loc[el][i][j]

    return K_globe, b_globe

K_globe, b_globe=assemble(tri.simplices, K_det, b_det)

bdmask=[True]*N



for p in range(len(points)):
    pt=points[p]
    if (pt[0]==0.0) or (pt[1]==0.0) or (pt[0]==a_dim) or (pt[1]==b_dim)
        bdmask[p]=False

K_globe_bc=K_globe[bdmask, :]
K_globe_bc=K_globe_bc[:, bdmask]

b_globe_bc=b_globe[bdmask, :]
b_globe_bc=b_globe_bc[:, bdmask]
```

3.4 Solving the Eigenvalue Equation

```
from sympy import Matrix
from numpy import linalg as la
K_mat=Matrix(K_globe)
b_mat=Matrix(b_globe)
#print (K_mat.det(), K_mat.is_symmetric())
#print (b_mat.det(), b_mat.is_symmetric())

binv=la.matrix_power(b_globe_bc, -1)
A=np.matmul(binv, K_globe_bc)
k_2, potvecs_bc=la.eig(A)

ko = np.sqrt(k_2)
print (np.sort(ko))

potvecs_new=np.zeros((N, N))
i, j=0, 0
for ind1 in range(N):
    j=0
    if (bdmask[ind1]):
        for ind2 in range(N):
            if (bdmask[ind2]):
                potvecs_new[ind1, ind2]=potvecs_bc[i, j]*potvecs_bc[i,j
                j=j+1
        i=i+1

import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns
X = points[:,0]
Y = points[:, 1]
for i in range(365, len(potvecs_bc)):
    data = pd.DataFrame({’X’: X, ’Y’: Y, ’Z’: potvecs_new[:,i]})
    data_pivoted = data.pivot(”X”, ”Y”, ”Z”)
    ax = sns.heatmap(data_pivoted)
    plt.show()
```

4 Results

4.1 Rectangular Metal Waveguide



(a) 11

(b) 12

(c) 13

(d) 21

(e) 22

(f) 23

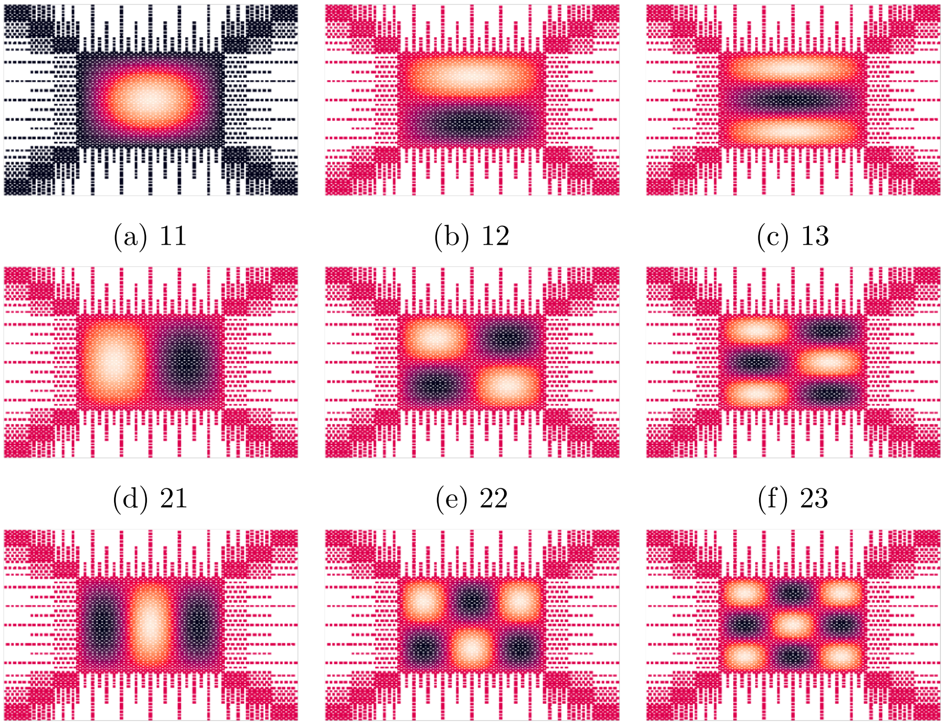
(g) 31

(h) 32

(i) 33

Figure 2: Field plots of (20x10) rectangular metal waveguide (2600 points, 5000 Mesh Elements)

4.2 Rectangular Normal Waveguide



(a) 11

(b) 12

(c) 13

(d) 21

(e) 22

(f) 23

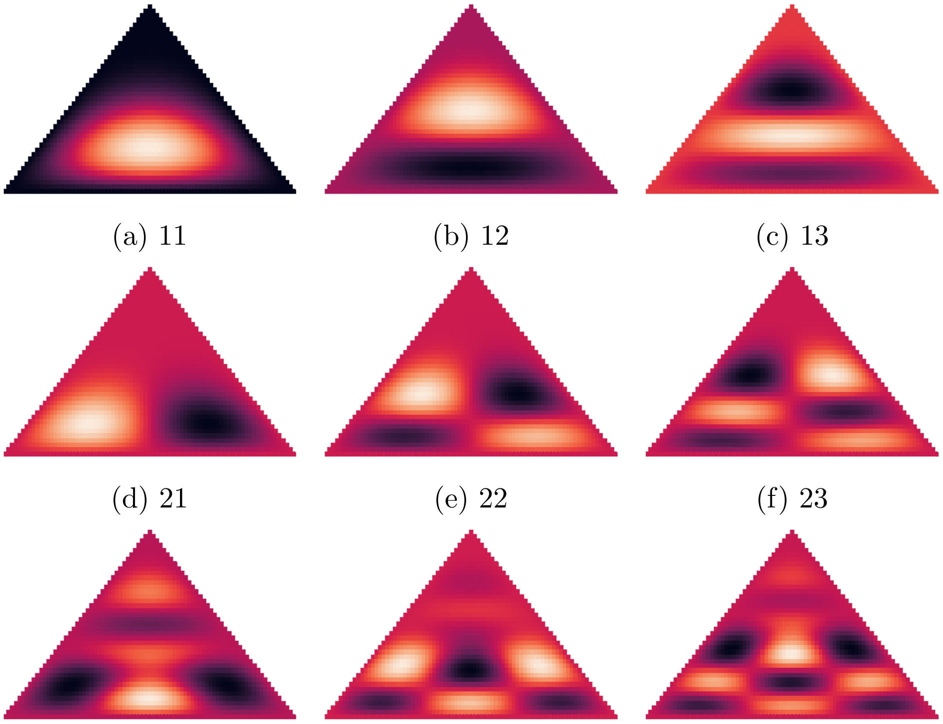
(g) 31

(h) 32

(i) 33

Figure 3: Field plots of (20x10) rectangular waveguide with a step index, placed in pseudo-infinite medium (3700 points, 7300 Mesh Elements)

4.3 Triangular Metal Waveguide



(a) 11

(b) 12

(c) 13

(d) 21

(e) 22

(f) 23

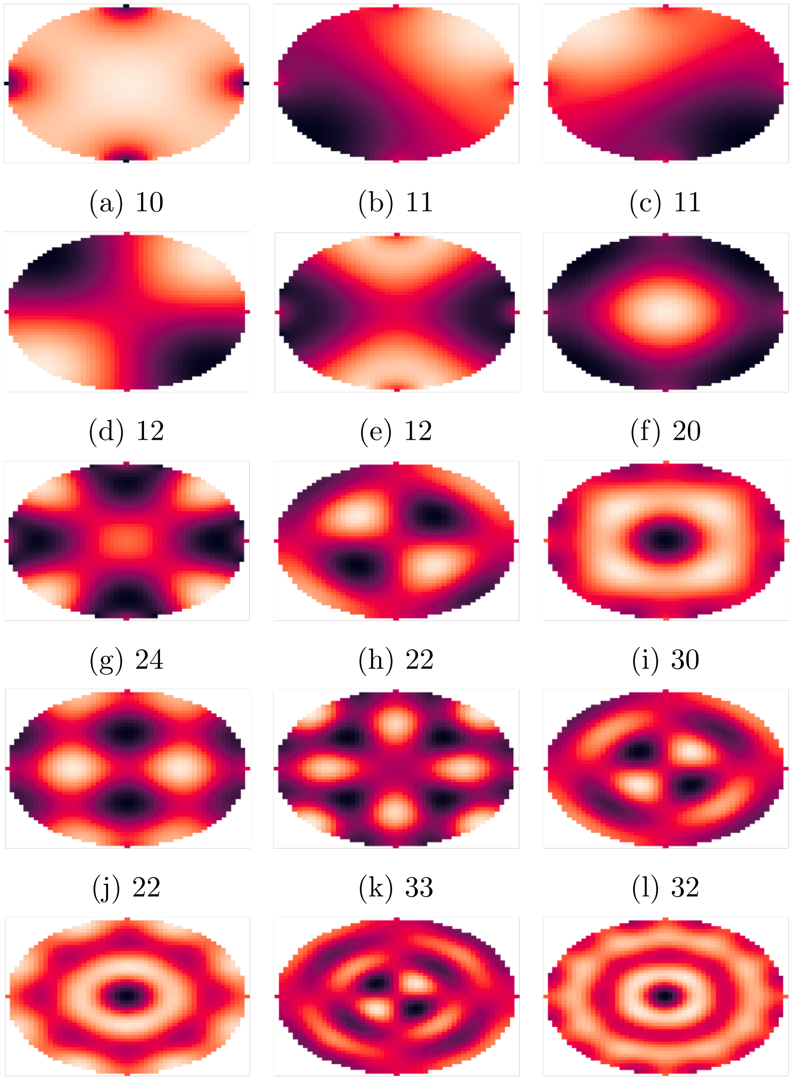
(g) 31

(h) 32

(i) 33

Figure 4: Field plots of (base:10, height:5) isosceles triangular metal waveguide
(1700 points, 3200 Mesh Elements)

4.4 Circular Metal Waveguide



(a) 10 (b) 11 (c) 11

(d) 12 (e) 12 (f) 20

(g) 24 (h) 22 (i) 30

(j) 22 (k) 33 (l) 32

(m) 40 (n) 42 (o) 50

Figure 5: Field plots of (5x5) circular metal waveguide (2000 points, 3800 Mesh Elements)

5 Future Scope

We have calculated the modal field solutions of a rectangular, circular and triangular waveguides. The field plots we have obtained are also consistent with the theoretical modal solutions to be expected for these contexts. A particularly interesting direction would be to implement the edge element formulation (Nedelec elements) in order to account for the vector behaviour of the field. Moreover, findings in literature suggest that these can be used to obtain better convergence of the eigenvalues than in the Lagrangian case (eg. Khankhoje 2014, Boffi et al. 1999[2], [1]). Moreover, it would be interesting to develop an automated scheme to eliminate spurious modes obtained as solutions of the N dimensional eigenvalue problem. Beyond the development of the finite element analysis solver, we would like to explore and implement mesh refinement techniques such as adaptive mesh refinement, in which we locally refine the mesh in regions with greater variation in the field or with geometric corners. This tecnique

References

[1] D. Boffi, R. G. Duran, and L. Gastaldi. A remark on spurious eigenvalues in a square. *Applied mathematics letters*, 12(3):107–114, 1999.

[2] U. K. Khankhoje. A primer on the 2d vector finite element method. 2014.

[3] A. Logg, K.-A. Mardal, G. N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

[4] R. H. Macneal and R. L. Harder. A proposed standard set of problems to test finite element accuracy. *Finite Elements in Analysis and Design*, 1(1):3 – 20, 1985.

[5] B. A. Rahman and A. Agrawal. *Finite element modeling methods for photonics*. Artech House, 2013.