finite element method notes

Jonathan Lin

September 2022

1 Introduction

The finite element method solves differential equations by discretizing space into a mesh of primitive elements (usually triangles). An ansatz is constructed in piecewise continuous fashion: on each element, the ansatz is defined as a linear combination of "shape" (or "interpolation") functions. The coefficients of the linear combination are termed "nodal" weights, each of which correspond to a point in the primitive element. Next, the ansatz is subject to some constraint, providing a system of N equations. Sometimes the resulting system is an ordinary linear system, other times a generalized eigenvalue problem. Finally, boundary conditions are applied and the solution is found using some numerical algorithm.

The simplest finite element is the "linear triangle", where each element is defined by three nodes and the variation of the field is assumed to be planar in the triangular region bound by the nodes. The next highest order element is the quadratic triangle, which is defined by 6 nodes (3 vertices and 3 edge midpoints). Inside the quadratic triangle, the field varies in the form $ax^2 + bxy + cy^2$.

In this document, we will apply the finite element method in the context of waveguide analysis. In particular, we will use FEM (with quadratic triangles) to solve for the eigenmodes of a weakly guiding waveguide with some refractive index field $n(x, y)$. The corresponding differential equation that must be solved is the paraxial wave equation

$$\nabla^2 u + \left[k^2 n^2(x, y) - \beta^2\right] u(x, y) = 0. \tag{1}$$

Note that the above equation assumes u is an eigenmode, which is why no partial derivative with respect to the longitudinal direction z appears.

We first provide an overview of the quadratic triangle (QT) finite element and the corresponding shape functions. We then apply FEM to solve equation 1 for a single QT element. Our constraint, which allows us to go from a 6-node QT element to a system of 6 equations, will follow Galerkin's method. Next, we apply FEM to a mesh of QT elements, introduce perfectly-matched layer (PML) boundary conditions, and construct the corresponding generalized eigenvalue problem. Finally, we consider prospects for automatic differentiation, so that perturbations in the refractive index distribution may be easily propagated to changes in eigenmode structure.

2 Quadratic triangle

The quadratic triangle (Figure 1) consists of 6 nodes: 3 vertices and 3 edge midpoints. Within this region, the field u is expanded in terms of 6 shape functions; these functions can be thought of as a basis for u over the triangle. Each shape function corresponds to a node, such that it evaluates to 1 at that specific node and 0 at all other nodes. We additionally enforce that each shape function varies like $ax^2 + bxy + cy^2$ over the triangle. These can be divided into vertex and edge shape functions, which have the following forms.

$$\begin{aligned} N_i(x, y) &= \frac{\left[x_{jk}(y - y_k) - y_{jk}(x - x_k)\right]\left[x_{ln}(y - y_n) - y_{ln}(x - x_n)\right]}{(x_{jk}y_{ik} - y_{jk}x_{ik})(x_{ln}y_{in} - y_{ln}x_{in})} \\ N_l(x, y) &= \frac{\left[x_{ki}(y - y_i) - y_{ki}(x - x_i)\right]\left[x_{jk}(y - y_k) - y_{jk}(x - x_k)\right]}{(x_{ki}y_{li} - y_{ki}x_{li})(x_{jk}y_{lk} - y_{jk}x_{lk})} \end{aligned} \tag{2}$$

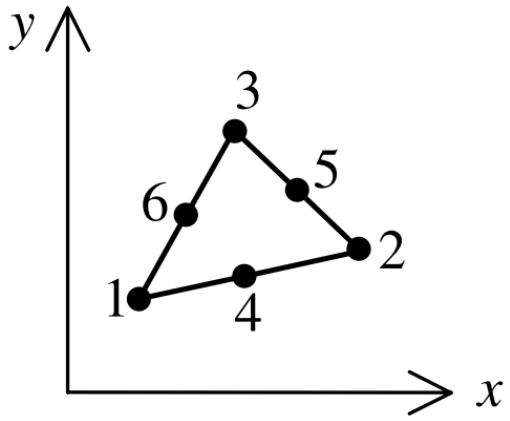

Figure 1: The quadratic triangle finite element. Numbers reflect how the nodes are indexed.

In the above, the indices i, j, k index the vertices in counterclockwise order while l, m, n do the same for edges. x_{ij} is shorthand for $x_i - x_j$, and similarly for y_{ij}. The above formulas can be extended for other edges and vertices by simultaneously cyclically permuting (i, j, k) and (l, m, n).

We now derive the partial derivatives of the shape functions for the QT element, and related area integrals over the triangle. These will later be used when we apply Galerkin FEM. It is first useful to make an affine transformation so that our coordinate axes lie along two of the edges of the triangle. Define the transformed coordinates u, v such that

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_{21} & x_{31} \\ y_{21} & y_{31} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \tag{3}$$

The inverse is

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{x_{21}y_{31} - x_{31}y_{21}} \begin{bmatrix} y_{31} & -x_{31} \\ -y_{21} & x_{21} \end{bmatrix} \begin{bmatrix} x - x_1 \\ y - y_1 \end{bmatrix} \tag{4}$$

This affine transformation sets (x_1, y_1) as the origin and edges 4 and 6 along the coordinate axes. This simplifies the basis functions immensely:

$$\begin{aligned} N_1 &= 2(1 - u - v)(1/2 - u - v) \\ N_2 &= 2u(u - 1/2) \\ N_3 &= 2v(v - 1/2) \\ N_4 &= 4u(1 - u - v) \\ N_5 &= 4uv \\ N_6 &= 4v(1 - u - v) \end{aligned} \tag{5}$$

2.1 Partial derivatives

We next compute partial derivatives with respect to x and y, expressed in the transformed basis. We expand out the math for N_1 specifically as an example. Denote $d \equiv x_{21}y_{31} - x_{31}y_{21}$ for brevity. I derive the following

(NOT CROSS-CHECKED).

$$
\begin{aligned}
\frac{\partial N_1}{\partial x} &= \frac{\partial N_1}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial N_1}{\partial v}\frac{\partial v}{\partial x} \\
&= (-3 + 4u + 4v)\frac{y_{32}}{d} \\
\frac{\partial N_1}{\partial y} &= \frac{\partial N_1}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial N_1}{\partial v}\frac{\partial v}{\partial y} \\
&= (-3 + 4u + 4v)\frac{x_{23}}{d} \\
\frac{\partial N_2}{\partial x} &= (4u - 1)\frac{y_{31}}{d} \\
\frac{\partial N_2}{\partial y} &= (4u - 1)\frac{-x_{31}}{d} \\
\frac{\partial N_3}{\partial x} &= (4v - 1)\frac{-y_{21}}{d} \\
\frac{\partial N_3}{\partial y} &= (4v - 1)\frac{x_{21}}{d} \\
\frac{\partial N_4}{\partial x} &= (4 - 8u - 4v)\frac{y_{31}}{d} + 4u\frac{y_{21}}{d} \\
\frac{\partial N_4}{\partial y} &= (4 - 8u - 4v)\frac{-x_{31}}{d} - 4u\frac{x_{21}}{d} \\
\frac{\partial N_5}{\partial x} &= 4u\frac{y_{31}}{d} - 4v\frac{y_{21}}{d} \\
\frac{\partial N_5}{\partial y} &= -4u\frac{x_{31}}{d} + 4v\frac{x_{21}}{d} \\
\frac{\partial N_6}{\partial x} &= (4 - 8v - 4u)\frac{y_{31}}{d} + 4v\frac{y_{21}}{d} \\
\frac{\partial N_6}{\partial y} &= (4 - 8v - 4u)\frac{-x_{31}}{d} - 4v\frac{x_{21}}{d}
\end{aligned}
\tag{6}
$$

We now consider area integrals over bounds of the QT element, for integrands of the form $N_i N_j$ and $\vec{\nabla} N_i \cdot \vec{\nabla} N_j$. The integrating bounds are

$$
\iint_{\Delta} dydx = d\int_0^1 \int_0^{1-u} dvdu \tag{7}
$$

where $|J| \equiv d$, J being the Jacobian of the affine transformation. This can be read from equation 3.

2.2 Integrals

The following integrals have been cross-checked through SymPy, and numerical testing. The following expressions are split into different categories and can be generalized (within the category).

vertex-same vertex

$$
\iint_{\Delta} N_1 N_1 dydx = d\int_0^1 \int_0^{1-u} N_1 N_1 dvdu = \frac{d}{60} \tag{8}
$$

vertex-other vertex

$$
\iint_{\Delta} N_1 N_2 dydx = -\frac{d}{360} \tag{9}
$$

vertex-adjacent edge

$$
\iint_{\Delta} N_1 N_4 dydx = \iint_{\Delta} N_1 N_6 dydx = 0 \tag{10}
$$

vertex-opposite edge

$$\iint_{\Delta} N_1 N_5 dydx = -\frac{d}{90} \tag{11}$$

edge-same edge

$$\iint_{\Delta} N_4 N_4 dydx = \frac{4d}{45} \tag{12}$$

edge-other edge

$$\iint_{\Delta} N_4 N_5 dydx = \frac{2d}{45} \tag{13}$$

All other relations within a category can be computed by simultaneously permuting vertices $1 \to 2 \to 3 \to 1$ and edges $4 \to 5 \to 6 \to 4$. Note that in such a procedure, $|J| \equiv d$ will not change, since $1/2|J|$ is always the area of the triangle. Clearly, the shape functions are not an orthonormal basis.

Next we compute integrals of the dot product of gradients, which are considerably more complicated.

vertex-same vertex

$$\iint_{\Delta} \vec{\nabla} N_1 \cdot \vec{\nabla} N_1 dydx = \frac{y_{32}^2 + x_{23}^2}{2d} \tag{14}$$

vertex-other vertex

$$\iint_{\Delta} \vec{\nabla} N_1 \cdot \vec{\nabla} N_2 dydx = \frac{y_{32}y_{31} + x_{32}x_{31}}{6d} \tag{15}$$

vertex-CCW adjacent edge

$$\iint_{\Delta} \vec{\nabla} N_1 \cdot \vec{\nabla} N_4 dydx = \frac{-2(y_{32}y_{31} + x_{32}x_{31})}{3d} \tag{16}$$

vertex-CW adjacent edge

$$\iint_{\Delta} \vec{\nabla} N_1 \cdot \vec{\nabla} N_6 dydx = \frac{2(y_{21}y_{32} + x_{21}x_{32})}{3d} \tag{17}$$

vertex-opposite edge

$$\iint_{\Delta} \vec{\nabla} N_1 \cdot \vec{\nabla} N_5 dydx = 0 \tag{18}$$

edge-same edge

$$\iint_{\Delta} \vec{\nabla} N_4 \cdot \vec{\nabla} N_4 dydx = \frac{4(y_{32}^2 + y_{31}y_{21} + x_{32}^2 + x_{31}x_{21})}{3d} \tag{19}$$

edge-other edge

$$\iint_{\Delta} \vec{\nabla} N_4 \cdot \vec{\nabla} N_5 dydx = \frac{4(y_{21}y_{32} + x_{21}x_{32})}{3d} \tag{20}$$

3 Galerkin FEM on a single QT element

The Galerkin method is a special case of a "weighted-residual" method for approximating the solution to a differential equation. Suppose we have an equation of the form $D[u] = 0$, where $D[]$ is some linear operator that involves differentiation. In the weighted residual method, we multiply the LHS with some weighting function $w(x, y)$ and integrate (average) over some area Ω. Setting this integral to 0 converts the differential equation into an integral equation (known as the "weak form" of the equation).

$$\iint_{\Omega} w(x,y) D[u] \, d\Omega = 0. \tag{21}$$

If we plug in an approximate solution $\tilde{u}$ to the above, $D[\tilde{u}]$ will not quite be 0 everywhere in the domain S – instead, we are left with a non-zero residual (multiplied by some weight function) which goes to 0 as the approximation becomes closer to the true solution.

In the Galerkin method, we solve equation 21 exactly over some simplified domain Ω. The shape of Ω is usually chosen to be some sort of polygon, parameterized by n nodes. For the case of the QT element, $n = 6$. Over Ω, we expand $\tilde{u}$ in terms of a basis, the shape functions N_i of Ω. We then use each shape function as a weighting function, yielding n integral equations. Each integral can be interpreted as decomposing the residual of $\tilde{u}$ in terms of the shape functions, then finding the values of $\tilde{u}$ so that each of the n components of the residual average to 0 over Ω. In this way, we solve the weak form of the differential equation $D[u]$ over the element Ω. "Global" solutions are created by subdividing the global domain into multiple simpler elements, and simultaneously solving for the local solution on each element.

We now explicitly apply the Galerkin method to solve the weak form of the paraxial wave equation 1 on a single QT element, denoted as Δ. As per the Galerkin method, we multiply equation 1 by the shape functions N_i, integrate over the triangle, and set the result to 0.

$$\iint_{\Delta} dydx \, N_i \left[\nabla^2 u + \left(k^2 n^2 - \beta^2 \right) u \right] = 0. \tag{22}$$

Next we integrate (the first portion of) the above by parts to get rid of the Laplacian. Recall the product rule, and divergence theorem for a domain Ω and closed bounding curve Γ:

$$\begin{aligned} &\int_{\Omega} \vec{\nabla} \cdot (f \vec{\nabla} g) d\Omega \\ &= \oint_{\Gamma} f \vec{\nabla} g \cdot \hat{n} d\Gamma \\ &= \iint_{\Omega} \vec{\nabla} f \cdot \vec{\nabla} g d\Omega + \iint_{\Omega} f \nabla^2 g d\Omega. \end{aligned} \tag{23}$$

The second line is the divergence theorem while the third is the product rule. Equating the second and third lines gives us the formula for integration by parts. Applying this to equation 22 gives

$$\oint_{\Gamma_\Delta} N_i \vec{\nabla} u \cdot \hat{n} d\Gamma - \iint_{\Omega_\Delta} \vec{\nabla} N_i \cdot \vec{\nabla} \tilde{u} d\Omega + \iint_{\Omega_\Delta} N_i \left[k^2 n^2 - \beta^2 \right] \tilde{u} d\Omega = 0. \tag{24}$$

The first term in equation 24 is our boundary condition, while the latter two terms set the residual of our approximate solution $\tilde{u}$ to 0. Note that in the boundary term, we do *not* approximate u. This is because boundary conditions are assumed to be known (also, boundary conditions stemming from shared edges in the mesh will cancel out, as we will later see).

We then expand $\tilde{u}$ in terms of the shape functions: $\tilde{u} = \sum_j \tilde{u}_j N_j$, where $\tilde{u}_j$ is the value of our $\tilde{u}$ at

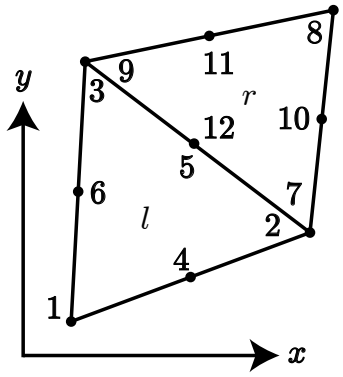

Figure 2: Two-QT mesh.

node j, giving

$$\begin{aligned}
&\oint_{\Gamma_\Delta} N_i \vec{\nabla} u \cdot \hat{n} d\Gamma - \iint_{\Omega_\Delta} \vec{\nabla} N_i \cdot \vec{\nabla} \sum_j \tilde{u}_j N_j d\Omega + \iint_{\Omega_\Delta} N_i \left[k^2 n^2 - \beta^2\right] \sum_j \tilde{u}_j N_j d\Omega = 0 \\
&\oint_{\Gamma_\Delta} N_i \vec{\nabla} u \cdot \hat{n} d\Gamma = \sum_j \tilde{u}_j \iint_{\Omega_\Delta} \vec{\nabla} N_i \cdot \vec{\nabla} N_j d\Omega - \left[k^2 n^2 - \beta^2\right] \sum_j \tilde{u}_j \iint_{\Omega_\Delta} N_i N_j d\Omega \\
&\oint_{\Gamma_\Delta} N_i \vec{\nabla} u \cdot \hat{n} d\Gamma = \sum_j \tilde{u}_j \left[\iint_{\Omega_\Delta} \vec{\nabla} N_i \cdot \vec{\nabla} N_j d\Omega - k^2 n^2 \iint_{\Omega_\Delta} N_i N_j d\Omega\right] + \beta^2 \sum_j \tilde{u}_j \iint_{\Omega_\Delta} N_i N_j d\Omega
\end{aligned} \tag{25}$$

In the above, we have assumed that the refractive index profile n does not vary over the QT element (which is a good approximation if the element is small enough and/or if the refractive index profile is piecewise-constant). Next, we make the following definitions:

$$\begin{aligned}
A_{ij} &\equiv - \iint_{\Omega_\Delta} \vec{\nabla} N_i \cdot \vec{\nabla} N_j d\Omega + k^2 n^2 \iint_{\Omega_\Delta} N_i N_j d\Omega \\
B_{ij} &\equiv \iint_{\Omega_\Delta} N_i N_j d\Omega \\
c_i &\equiv \oint_{\Gamma_\Delta} N_i \vec{\nabla} u \cdot \hat{n} d\Gamma
\end{aligned} \tag{26}$$

Equation 25 can equivalently be expressed as

$$A\tilde{u} + \vec{c} = \beta^2 B \tilde{u}. \tag{27}$$

For $\vec{c} = 0$, the above reduces to the "generalized eigenvalue" problem, which can be solved numerically. We additionally note that the contour integrals in $\vec{c}$ can often be simplified because the shape functions are 0 along certain edges. For instance, denoting the three vertices of the triangle as $1, 2, 3$:

$$\begin{aligned}
c_1 &= \int_{1\to 2} N_1 \vec{\nabla} u \cdot \hat{n} d\Gamma + \int_{3\to 1} N_1 \vec{\nabla} u \cdot \hat{n} d\Gamma \\
c_4 &= \int_{1\to 2} N_4 \vec{\nabla} u \cdot \hat{n} d\Gamma
\end{aligned} \tag{28}$$

4 Multiple elements and boundary conditions

To extend the Galerkin method over a mesh of QT elements, we first make the observation that the contour integrals which compose of the boundary condition vector $\vec{c}$ partially cancel out: specifically along edges

shared by two QT elements. To make this clear, we consider the resulting system for a two-element QT mesh (Figure 2). This mesh has 9 nodes; however, it will be useful to consider the mesh as two separate QTs, each with 6 nodes, for a total of 12 nodes. The field $\tilde{u}$ is expanded in a basis of 12 shape functions. Following Section 3, we construct the matrices A_l, A_r, B_l, B_r and the boundary condition vectors $\vec{c}_l, \vec{c}_r$, where the subscript denotes the leftmost or rightmost triangle. The overall system of equations corresponding to both QT elements can be written in the following block form:

$$\begin{bmatrix} A_l & 0 \\ 0 & A_r \end{bmatrix} \tilde{u} = \lambda \begin{bmatrix} B_l & 0 \\ 0 & B_r \end{bmatrix} \tilde{u} + \begin{bmatrix} \vec{c}_l \\ \vec{c}_r \end{bmatrix} \tag{29}$$

We now repeatedly "contract" the system of equations to remove redundant equations. To make this process clear, consider the following two equations, which correspond to rows 5 and 12 of equation 29. These are:

$$\begin{aligned} \sum_{i=1}^{6} A_{l,5i}\tilde{u}_i &= \lambda \sum_{i=1}^{6} B_{l,5i}\tilde{u}_i + c_{l,5} \\ \sum_{i=7}^{12} A_{r,12i}\tilde{u}_i &= \lambda \sum_{i=7}^{12} B_{r,12i}\tilde{u}_i + c_{r,12} \end{aligned} \tag{30}$$

Using the fact that $\tilde{u}_5 = \tilde{u}_{12}$, we add the above two equations:

$$\begin{aligned} &\sum_{i=1,\neq 5}^{6} A_{l,5i}\tilde{u}_i + \sum_{i=7}^{11} A_{r,12i}\tilde{u}_i + (A_{l,55} + A_{r,12\,12})\tilde{u}_5 \\ &= \lambda \left[\sum_{i=1,\neq 5}^{6} B_{l,5i}\tilde{u}_i + \sum_{i=7}^{11} B_{r,12i}\tilde{u}_i + (B_{l,55} + B_{r,12\,12})\tilde{u}_5 \right] + c_{l,5} + c_{r,12} \end{aligned} \tag{31}$$

By combining equations 5 and 12 like above, we have reduced the dimensionality of the problem by 1, removing the redundant equation that stemmed from labelling the shared edge node in the QT mesh twice. Next, we consider the new boundary condition term, $c_{l,5} + c_{r,5}$. This is two closed integrals, both of which traverse the central edge of the mesh. The integral for the left triangle traverses the edge in the $2 \to 3$ direction, while the right triangle contour integral traverses in the opposite direction. Furthermore, the integrands are *the same*.[1] Thus, $c_{l,5} + c_{r,5}$ reduces to a contour integral that traverses the *bounding quadrilateral* of the two triangles, in the counterclockwise direction. We further note that in the particular case of a shared edge node (node 5 aka node 12), the shape functions evaluate to 0 along the entire bounding quadrilateral. So the boundary condition term c_5 disappears. This is not necessarily the case for shared vertex nodes (nodes 2,3 aka 7,9), unless we choose appropriate boundary conditions.

The above process of combining equations can be performed two more times, to combine equations 3 and 9, and equations 2 and 7. This process reduces the dimensions of A and B to 9×9; furthermore, $\vec{c}$ will only contain contour integrals along the bounding curve of the overall mesh. The overall process can be extended to an arbitrary triangular mesh. In practice, this entails initializing empty "master" A and B matrices with dimensions $n \times n$ (for n mesh nodes). Then, we compute "element-wise" 6×6 A and B matrices for each individual triangle/set of nodes, and then add those element-wise matrices to the 6×6 slices of the master matrices corresponding to each triangle's set of 6 nodes.

We also note a subtlety with how the refractive index profile must be represented. At first, it seems most intuitive to simply discretely define n at the locations of the nodes in the mesh. However, we note that this will lead to inconsistency: if two connected nodes have differing refractive indices, n cannot be constant within the triangular element.[2] As such, n must instead be discretized at the triangle faces. For instance, given some profile $n(x, y)$ and QT mesh, the centroids of each triangle can be computed. Then the index of the entire triangle is set to be the index at the centroid.

[1] This hinges on the shape functions N_5 and N_{12} being identical when evaluated on the $2 \to 3$ edge.

[2] Another option is to define the refractive index in terms of the same shape function basis as $\tilde{u}$ ... hmm

4.1 Boundary conditions: homogeneous Neumann

As mentioned in the previous section, boundary conditions enter through the elements of the $\vec{c}$ vector, whose elements are closed contour integrals. Due to the nature of the shape functions, the only non-zero contribution of these contour integrals occurs for nodes at the mesh boundary. Furthermore, the contour integrals are only non-zero when integrated along the boundary: contributions from either internal edges cancel out, or are zeroed out by the properties of the shape functions.

The simplest boundary condition is to assume homogeneous Neumann boundary conditions. For some contour C, this condition can be written as

$$\left.\frac{\partial u}{\partial n}\right|_{x\in C} = 0 \tag{32}$$

where differentiation is with respect to the normal vector $\hat{n}$ of the contour C. In this case, all boundary conditions can simply be ignored, since they all go to 0.

5 Differentiation wrt refractive index

For brevity, we define $\kappa = k^2 n^2$. We will attempt to find the derivatives of $\tilde{u}_i$ with respect to κ, which can then be used to compute the derivative with respect to n. We will index κ as κ_k, where the k subscript denotes a specific triangular element (recall that we assume $n(x, y)$ to be piecewise constant over the triangles). Suppose we have solved for an initial refractive index profile $n(x, y)$; that is, we have solved

$$\begin{aligned} A\tilde{u} &= \beta^2 B\tilde{u} \\ \sum_k \left[-\iint_{S_\Delta} \vec{\nabla} N_i \cdot \vec{\nabla} N_j dS + \kappa_k \iint_{S_\Delta} N_i N_j dS \right] \tilde{u} &= \beta^2 \sum_k \left[\iint_{S_\Delta} N_i N_j dS \right] \tilde{u} \end{aligned} \tag{33}$$

where A and B are *global* matrices of dimension $M \times M$, where M is the total number of nodes. Summation over k represents the global assembly of the A and B matrices from the elemental 6×6 matrices corresponding to each individual triangular element. We now implicitly differentiate with respect to κ_i. The result is a similar equation that operates on $\partial\tilde{u}/\partial\kappa_k$ instead of $\tilde{u}$, with an extra term due to the product rule:

$$\begin{aligned} \sum_k \left[-\iint_{S_\Delta} \vec{\nabla} N_i \cdot \vec{\nabla} N_j dS + \kappa_k \iint_{S_\Delta} N_i N_j dS \right] \frac{\partial \tilde{u}}{\partial \kappa_k} + \left[\iint_{S_{\Delta,k}} N_i N_j dS \right] \tilde{u} &= \beta^2 \sum_k \left[\iint_{S_\Delta} N_i N_j dS \right] \frac{\partial \tilde{u}}{\partial \kappa_k} \\ A\frac{\partial \tilde{u}}{\partial \kappa_k} + B_{\Delta,k}\tilde{u} &= \beta^2 B \frac{\partial \tilde{u}}{\partial \kappa_k}. \end{aligned} \tag{34}$$

In the above, $B_{\Delta,k}$ is an $M \times M$ matrix whose only non-zero elements correspond to the 6×6 local B matrix corresponding to triangle k. Critically, we also assume that β is roughly unchanged with respect to refractive index (I believe this approximation is acceptable in the weak-guidance regime). Since we have assumed that equation 33 was solved, $\tilde{u}$ and β are known. At first glance, it seems it should be possible to isolate the derivatives, through use of the pseudo-inverse:

$$\frac{\partial \tilde{u}}{\partial \kappa_k} = \left[\beta^2 B - A\right]^+ B_{\Delta,k}\tilde{u} \tag{35}$$

Note that $\beta^2 B - A$ has no true inverse because according to the first line of equation 33, the matrix $\beta^2 B - A$ cannot be full-rank.

6 Overlap integrals

Suppose we have two fields $u(x, y)$ and $v(x, y)$, discretized over the same triangular mesh as u_i^n and v_i^n, where the n superscript indexes a triangle in the mesh and $i = 1, ..., 6$ iterates through the points in the triangle.

The fields over triangle n are

$$\begin{aligned} u^n(x,y) &= \sum_i u_i^n N_i(x,y) \\ v^n(x,y) &= \sum_i v_i^n N_i(x,y). \end{aligned} \tag{36}$$

The overlap integral A^n over triangle n is

$$\begin{aligned} A^n &= \int_{\Delta_n} dxdy \ \left[\sum_i u_i^n N_i(x,y)\right]\left[\sum_j v_j^n N_j(x,y)\right] \\ &= \sum_{i,j} u_i^n v_j^n \int_{\Delta_n} dxdy \ N_i(x,y) N_j(x,y) \\ &\equiv \sum_{i,j} u_i^n v_j^n B_{ij}^n. \end{aligned} \tag{37}$$

The total overlap integral A is

$$A = \sum_n A^n = \sum_n \sum_{ij} u_i^n v_j^n B_{ij}^n \, ? = \mathbf{u}^T B \mathbf{v} \tag{38}$$

6.1 Operator in matrix form

Suppose we have a field $f(x,y)$ and a modal basis $u_i(x,y)$, and that we want to compute the matrix F_{ij}:

$$F_{ij} \equiv \int u_i \, f \, u_j \, dx \, dy. \tag{39}$$

Upon discretization over a finite element mesh, we further assume that $f(x,y)$ is piecewise-constant over each finite element. The overlap over triangle n is

$$F_{ij}^n = \int_{\Delta_n} dxdy \ f^n \left[\sum_k u_{i,k}^n N_k(x,y)\right]\left[\sum_l v_{j,l}^n N_l(x,y)\right] \tag{40}$$

where $u_{i,k}^n$ represents the FE discretization of the basis mode u_i: n iterates through each triangle, and k iterates through the points composing each triangle. The total overlap is

$$F_{ij} \approx \sum_n F_{ij}^n = \sum_n f^n \tag{41}$$

7 Vector modes

The purpose of this section is to work out the integrals needed for the vectorial formulation of the finite-element method for modesolving. We will use linear triangle elements. The vertex weighting functions, in affine coordinates, are

$$\begin{aligned} N_1 &= 1 - u - v \\ N_2 &= u \\ N_3 &= v. \end{aligned} \tag{42}$$

In the vectorial formulations, the tranverse component of the electric field is expanded in terms of edge weighting functions, which are constructed from vertex weighting functions:

$$\boldsymbol{N}_{ij} = \left[N_i \nabla N_j - N_j \nabla N_i\right] l_{ij} \tag{43}$$

where the indices i and j correspond to the triangle vertices and l_{ij} is the signed edge length (note that the initial choice of which direction is positive doesn't matter, we just need to be consistent). The gradient of the vertex shape functions is

$$\begin{aligned}\nabla N_1 &= \frac{(-y_{31}+y_{21}, x_{31}-x_{21})}{d} \\ \nabla N_2 &= \frac{(y_{31}, -x_{31})}{d} \\ \nabla N_3 &= \frac{(-y_{21}, x_{21})}{d}.\end{aligned} \tag{44}$$

The required integrals to form the finite element matrices in the vectorial formulation are presented below.

7.1 edge - edge

$$\begin{aligned}\iint \boldsymbol{N}_{12}\cdot\boldsymbol{N}_{12} &= l_{12}^2\frac{l_{12}^2-3x_{21}x_{31}+3l_{31}^2-3y_{21}y_{31}}{12d} \\ \iint \boldsymbol{N}_{23}\cdot\boldsymbol{N}_{23} &= l_{23}^2\frac{l_{12}^2+x_{21}x_{31}+l_{31}^2+y_{21}y_{31}}{12d} \\ \iint \boldsymbol{N}_{31}\cdot\boldsymbol{N}_{31} &= l_{31}^2\frac{3l_{12}^2-3x_{21}x_{31}+l_{31}^2-3y_{21}y_{31}}{12d} \\ \iint \boldsymbol{N}_{12}\cdot\boldsymbol{N}_{23} &= l_{12}l_{23}\frac{l_{12}^2-x_{21}x_{31}-l_{31}^2-y_{21}y_{31}}{12d} \\ \iint \boldsymbol{N}_{23}\cdot\boldsymbol{N}_{31} &= l_{23}l_{31}\frac{-l_{12}^2-x_{21}x_{31}+l_{31}^2-y_{21}y_{31}}{12d} \\ \iint \boldsymbol{N}_{31}\cdot\boldsymbol{N}_{12} &= l_{31}l_{12}\frac{-l_{12}^2+3x_{21}x_{31}-l_{31}^2+3y_{21}y_{31}}{12d}\end{aligned} \tag{45}$$

7.2 curl edge - curl edge

I find:

$$\nabla\times\boldsymbol{N}_{ij} = \frac{2l_{ij}}{d}\hat{\boldsymbol{z}}. \tag{46}$$

The integrals are

$$\iint(\nabla\times\boldsymbol{N}_{ij})\cdot(\nabla\times\boldsymbol{N}_{mn}) = \frac{2l_{ij}l_{mn}}{d}. \tag{47}$$

7.3 edge - vertex

$$
\begin{aligned}
\iint \boldsymbol{N}_{12} \cdot \nabla N_1 &= l_{12} \frac{-l_{12}^2 + 3x_{21}x_{31} - 2l_{31}^2 + 3y_{21}y_{31}}{6d} \\
\iint \boldsymbol{N}_{23} \cdot \nabla N_2 &= l_{23} \frac{-x_{21}x_{31} - l_{31}^2 - y_{21}y_{31}}{6d} \\
\iint \boldsymbol{N}_{31} \cdot \nabla N_3 &= l_{31} \frac{-2l_{12}^2 + x_{21}x_{31} + y_{21}y_{31}}{6d} \\
\iint \boldsymbol{N}_{12} \cdot \nabla N_2 &= l_{12} \frac{-x_{21}x_{31} + 2l_{31}^2 - y_{21}y_{31}}{6d} \\
\iint \boldsymbol{N}_{23} \cdot \nabla N_1 &= l_{23} \frac{-l_{12}^2 + l_{31}^2}{6d} \\
\iint \boldsymbol{N}_{23} \cdot \nabla N_3 &= l_{23} \frac{l_{12}^2 + x_{21}x_{31} + y_{21}y_{31}}{6d} \\
\iint \boldsymbol{N}_{31} \cdot \nabla N_2 &= l_{31} \frac{2x_{21}x_{31} + 2y_{21}y_{31} - l_{31}^2}{6d} \\
\iint \boldsymbol{N}_{31} \cdot \nabla N_1 &= l_{31} \frac{2l_{12}^2 - 3x_{21}x_{31} + l_{31}^2 - 3y_{21}y_{31}}{6d} \\
\iint \boldsymbol{N}_{12} \cdot \nabla N_3 &= l_{12} \frac{l_{12}^2 - 2x_{21}x_{31} - 2y_{21}y_{31}}{6d}
\end{aligned}
\tag{48}
$$

7.4 vertex - vertex

$$
\begin{aligned}
\iint N_i N_i &= \frac{d}{12} \\
\iint N_i N_j &= \frac{d}{24} \quad ; \quad i \neq j
\end{aligned}
\tag{49}
$$

$$
\begin{aligned}
\iint \nabla N_1 \cdot \nabla N_1 &= \frac{l_{12}^2 + l_{31}^2 - 2x_{21}x_{31} - 2y_{21}y_{31}}{2d} \\
\iint \nabla N_2 \cdot \nabla N_2 &= \frac{l_{31}^2}{2d} \\
\iint \nabla N_3 \cdot \nabla N_3 &= \frac{l_{12}^2}{2d} \\
\iint \nabla N_1 \cdot \nabla N_2 &= \frac{-l_{31}^2 + x_{21}x_{31} + y_{21}y_{31}}{2d} \\
\iint \nabla N_2 \cdot \nabla N_3 &= \frac{-x_{21}x_{31} - y_{21}y_{31}}{2d} \\
\iint \nabla N_3 \cdot \nabla N_1 &= \frac{-l_{21}^2 + x_{21}x_{31} + y_{21}y_{31}}{2d}
\end{aligned}
\tag{50}
$$

The generalized eigenvalue problem is defined through the above sets of matrices, each of which corresponds to a 3×3 submatrix. (For brevity the index pair corresponding to the vertices for each of the edge weighting

functions are replaced by a single index which labels the edges).

$$\begin{aligned}
A_{tt,ij} &= \iint \left[k^2 n^2 \boldsymbol{N}_i \cdot \boldsymbol{N}_j - (\nabla \times \boldsymbol{N}_i) \cdot (\nabla \times \boldsymbol{N}_j)\right] \\
B_{tt,ij} &= \iint \boldsymbol{N}_i \cdot \boldsymbol{N}_j \\
B_{tz,ij} &= \iint \boldsymbol{N}_i \cdot \nabla N_j \\
B_{zt,ij} &= B_{tz,ji} \\
B_{zz,ij} &= \iint \left[\nabla N_i \cdot \nabla N_j - k^2 n^2 N_i N_j\right].
\end{aligned} \tag{51}$$

The overall generalized eigenproblem is

$$\begin{bmatrix} A_{tt} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_t \\ \boldsymbol{e}_z \end{bmatrix} = \beta^2 \begin{bmatrix} B_{tt} & B_{tz} \\ B_{zt} & B_{zz} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_t \\ \boldsymbol{e}_z \end{bmatrix} \tag{52}$$

where the vectors $\boldsymbol{e}_t$ and $\boldsymbol{e}_z$ correspond to the transverse and longitudinal components of the electric field. Note that the matrix on the right-hand side may not be positive-definite, which can cause issues for sparse solvers. In this case, you need to use a workaround. Denote the generalized eigenvalue problem as

$$A\boldsymbol{e} = \beta^2 B\boldsymbol{e}. \tag{53}$$

Find C such that

$$A = BC \tag{54}$$

using a sparse linear solver, such as `scipy.sparse.linalg.spsolve`. Then transform to the standard eigenvalue problem

$$C\boldsymbol{e} = \beta^2 \boldsymbol{e}. \tag{55}$$

You can solve the above with a method such as `scipy.sparse.linalg.eigs`. Note that C might not be real symmetric even if A and B are.