# Finite element method

Computational Photonics 2023

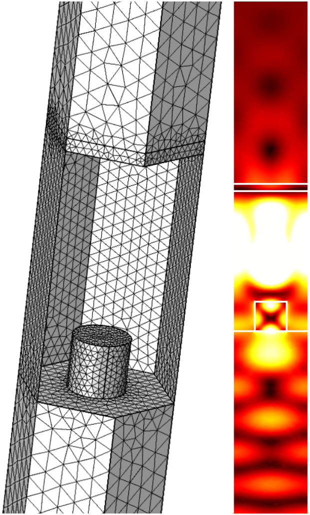
# Motivation and idea

General-purpose partial differential equation discretization method that allows for:

- Spatially varying resolution
- Discretization that conforms to geometrical shapes
- Arbitrarily shaped computational domain

**Concepts**

- Finite element: a discrete piece of space
  - 1D: interval
  - 2D: triangle, rectangle, ...
  - 3D: tetrahedron, cube, ...
- Node: a point, e.g., a vertex of a triangle
- Mesh: collection of nodes and elements

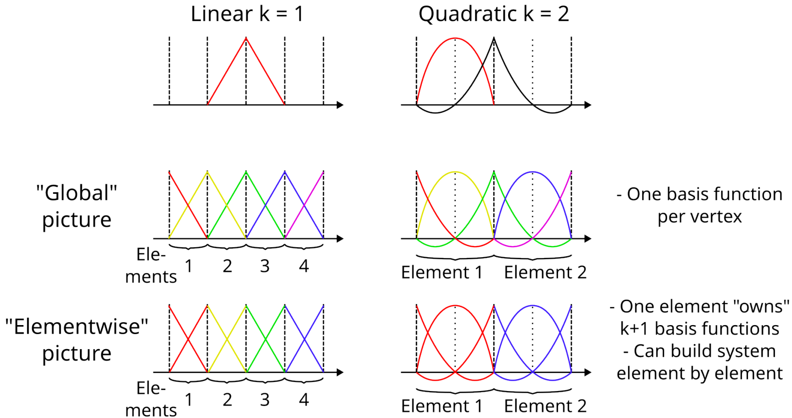


# Basis functions and finite elements

From now on, let us consider 1D problems for simplicity.
A field $u(x)$ in terms of basis functions $\phi_n(x)$:

$$u(x) = \sum_n u_n \phi_n(x)$$

In FEM, basis functions are *localized* and usually piecewise polynomial.
Here, "Lagrangian" or "nodal" elements



# The weak form

Wave equation eigenvalue problem (1D waveguide, TE polarization)

$$\frac{\partial^2 \hat{u}(x,y)}{\partial x^2} + \frac{\partial^2 \hat{u}(x,y)}{\partial y^2} + k_0^2 \epsilon(x) \hat{u}(x,y) = 0, \qquad \hat{u}(x,y) = u(x) e^{i\beta z}$$

$$\frac{\partial^2 u(x)}{\partial x^2} + k_0^2 \epsilon(x) u(x) = \beta^2 u(x)$$

Expansion of $u(x)$ into sum of basis functions

$$\sum_n u_n \frac{\partial^2 \phi_n(x)}{\partial x^2} + k_0^2 \sum_n u_n \epsilon(x) \phi_n(x) = \beta^2 \sum_n u_n \phi_n(x)$$

Multiply by one of the basis functions ("test function", $\phi_m(x)$) and integrate (inner product)

$$\sum_n u_n \int \phi_m(x) \frac{\partial^2 \phi_n(x)}{\partial x^2} dx + k_0^2 \sum_n u_n \int \phi_m(x) \epsilon(x) \phi_n(x) dx$$
$$= \beta^2 \sum_n u_n \int \phi_m(x) \phi_n(x) dx$$

# Weak form to matrix equation

Multiply with each different $\phi_m(x)$: $N$ equations ($N$ is the number of basis functions). Then,

$$(\mathbf{S} + \mathbf{W})\mathbf{u} = \beta^2 \mathbf{M}\mathbf{u}$$

where

$$\begin{aligned}
S_{m,n} &= \int \phi_m(x) \frac{\partial^2 \phi_n(x)}{\partial x^2} dx \\
W_{m,n} &= k_0^2 \int \phi_m(x) \epsilon(x) \phi_n(x) dx \\
M_{m,n} &= \int \phi_m(x) \phi_n(x) dx
\end{aligned}$$

Generalized eigenvalue problem with system matrix $\mathbf{S} + \mathbf{W}$ and mass matrix $\mathbf{M}$.

# Evaluating the integrals

The integrals can be done element by element. E.g. for the element $x \in [x_1, x_2]$:
Second derivative term

$$\sum_n u_n \int_{x_1}^{x_2} \phi_m(x) \frac{\partial^2 \phi_n(x)}{\partial x^2} dx =$$

$$\sum_n u_n \phi_m \frac{\partial \phi_n(x)}{\partial x}\Big|_{x_1}^{x_2} - \sum_n u_n \int_{x_1}^{x_2} \frac{\partial \phi_m(x)}{\partial x} \frac{\partial \phi_n(x)}{\partial x} dx$$

Wave number term

$$k_0^2 \sum_n u_n \int_{x_1}^{x_2} \phi_m(x) \epsilon(x) \phi_n(x) dx$$

Eigenvalue term

$$\beta^2 \sum_n u_n \int_{x_1}^{x_2} \phi_m(x) \phi_n(x) dx$$

In the sums, only the basis functions *on this element* need to be considered: convenience of the "elementwise" picture.

# With piecewise linear basis functions

Prototype element $x \in [0, 1]$, only two prototype basis functions on it:

$$\phi_0(x) = 1 - x$$
$$\phi_1(x) = x$$

Normalization: peak value 1.

$$\int_0^1 \phi_0(x)\phi_0(x)dx = \frac{1}{3}, \qquad \int_0^1 \phi_0(x)\phi_1(x)dx = \frac{1}{6}$$
$$\int_0^1 \frac{\partial\phi_0(x)}{\partial x}\frac{\partial\phi_0(x)}{\partial x}dx = 1, \qquad \int_0^1 \frac{\partial\phi_0(x)}{\partial x}\frac{\partial\phi_1(x)}{\partial x}dx = -1$$

For doing the integral $\int_0^1 \phi_j(x)\epsilon(x)\phi_i(x)dx$:

- If $\epsilon(x)$ is constant on each element (very common), it becomes just a prefactor.
- Numerical integration: e.g. Gaussian quadrature rules.

# Integrals on an arbitrary element

Transformation from arbitrary element to prototype element (notation: element's untransformed basis functions $\phi_E$, prototype basis functions $\phi$)

$$\int_{x_a}^{x_b} \phi_{E,m}(x)\phi_{E,n}(x)dx = (x_b - x_a)\int_0^1 \phi_m(x')\phi_n(x')dx'$$

$$\int_{x_a}^{x_b} \frac{\partial\phi_{E,m}(x)}{\partial x}\frac{\partial\phi_{E,n}(x)}{\partial x}dx = \frac{x_b - x_a}{(x_b - x_a)^2}\int_0^1 \frac{\partial\phi_m(x')}{\partial x'}\frac{\partial\phi_n(x')}{\partial x'}dx'$$

On the RHS, the integrals are exactly the same as the ones on the previous slide, and the prefactors reflect the fact that different elements (intervals) can have different sizes.

# Practical implementation of 1D FEM

System to build and solve

$$(\mathbf{S} + \mathbf{W})\mathbf{u} = \beta^2 \mathbf{M}\mathbf{u}$$

where

$$\begin{aligned}
S_{m,n} &= \int \phi_m(x) \frac{\partial^2 \phi_n(x)}{\partial x^2} dx \\
&= -\int \frac{\partial \phi_m(x)}{\partial x} \frac{\partial \phi_n(x)}{\partial x} dx \\
W_{m,n} &= k_0^2 \int \phi_m(x) \epsilon(x) \phi_n(x) dx \\
M_{m,n} &= \int \phi_m(x) \phi_n(x) dx
\end{aligned}$$

# Practical implementation of 1D FEM

Defining the mesh and distribution of $\epsilon$:

- Define nodes $[x_0, x_1, x_2, \ldots]$.
- Element $n$ is an interval between nodes $x_n$ and $x_{n+1}$.
- Let $\epsilon$ be constant on each element: $[\epsilon_0, \epsilon_1, \ldots]$

Constructing the system

- Create empty sparse matrices **S**, **W** and **M**.
- For each element $n$ with length $L_n = x_{n+1} - x_n$, and corresponding field unknowns $u_n$ and $u_{n+1}$:
  - Add to second derivative operator:
    - $S_{n,n} += -1/L_n$, this is $-1/L_n \int \partial_x \phi_0(x') \partial_x \phi_0(x') dx'$
    - $S_{n+1,n+1} += -1/L_n$, this is $-1/L_n \int \partial_x \phi_1(x') \partial_x \phi_1(x') dx'$
    - $S_{n,n+1} += 1/L_n$, this is $-1/L_n \int \partial_x \phi_0(x') \partial_x \phi_1(x') dx'$
    - $S_{n+1,n} += 1/L_n$, this is $-1/L_n \int \partial_x \phi_1(x') \partial_x \phi_0(x') dx'$
  - Add to wave number operator:
    - $W_{n,n} += k_0^2 \epsilon_n L_n \frac{1}{3}$, this is $k_0^2 \epsilon_n L_n \int \phi_0(x') \phi_0(x') dx'$
    - $W_{n+1,n+1} += k_0^2 \epsilon_n L_n \frac{1}{3}$
    - $W_{n,n+1} += k_0^2 \epsilon_n L_n \frac{1}{6}$
    - $W_{n+1,n} += k_0^2 \epsilon_n L_n \frac{1}{6}$
  - Add to mass matrix: same as wave number operator but without $k_0^2 \epsilon_n$.

# Example system

Let $x = [0, 1, 2, 3]$ and $\epsilon = 1$ everywhere. Then,

$$\mathbf{S} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

which has a boundary condition $\partial u / \partial x = 0$.

$$\mathbf{W} = k_0^2 \begin{bmatrix} 1/3 & 1/6 & 0 & 0 \\ 1/6 & 2/3 & 1/6 & 0 \\ 0 & 1/6 & 2/3 & 1/6 \\ 0 & 0 & 1/6 & 1/3 \end{bmatrix}$$

is not just a diagonal matrix.
$\mathbf{M}$ is like $\mathbf{W}$ but without the $k_0^2$.

# Boundary conditions

- Metallic / perfect electric conductor / Dirichlet boundary condition: set field unknowns $u_n$ for all $n$ on boundary to zero, remove them from the unknown vector, and remove all rows / columns corresponding to these $n$ from the matrices.
- Field continuity / Neumann boundary condition: implemented through the system matrix's boundary term

$$\sum_n u_n \phi_m \frac{\partial \phi_n(x)}{\partial x}\Big|_{x_1}^{x_2}$$

  If we do not implement this term, we set $\partial u / \partial x = 0$ as in the example.
- Periodic boundary condition: constraint equations

# FEM in two dimensions

Scalar wave equation eigenvalue problem

$$\frac{\partial^2 \hat{u}(x,y,z)}{\partial x^2} + \frac{\partial^2 \hat{u}(x,y,z)}{\partial y^2} + \frac{\partial^2 \hat{u}(x,y,z)}{\partial z^2} + k_0^2 \epsilon(x,y)\hat{u}(x,y,z) = 0$$

$$\hat{u}(x,y,z) = u(x,y)e^{i\beta z}$$

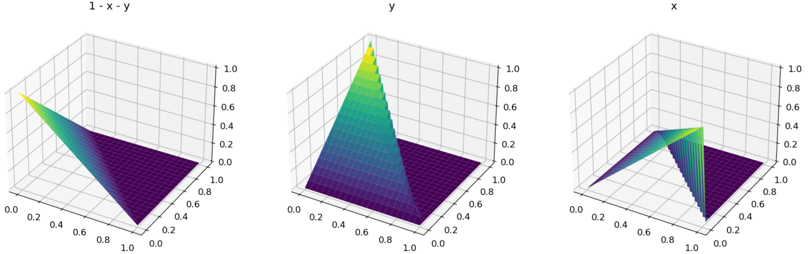
$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} + k_0^2 \epsilon(x,y)u(x,y) = \beta^2 u(x,y)$$

Basis functions should be two-dimensional as well

$$u(x,y) = \sum_n u_n \phi_n(x,y)$$

# 2D basis functions

Example: triangle elements, linear basis functions
Prototype element: a triangle with the vertices (0,0), (1,0), (0,1).




$$\int_0^1 \int_0^{1-x} (1-x-y)(1-x-y)dydx = \frac{1}{12}$$

$$\int_0^1 \int_0^{1-x} (1-x-y)xdydx = \frac{1}{24}$$

$\dots$

# Transformation to/from prototype element in 2D

Mapping the triangle element $E$ defined by points $(x, y) = (x_0, y_0)$, $(x_1, y_1)$, $(x_2, y_2)$ to $(x', y') = (0, 0)$, $(1, 0)$, $(0, 1)$ can be done with an affine transformation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{A}(\begin{bmatrix} x \\ y \end{bmatrix} - \mathbf{b})$$

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \mathbf{b} = \begin{bmatrix} x_0 & y_0 \end{bmatrix}, \mathbf{A}^{-1} = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{bmatrix}$$

Basis function on the arbitrary element defined from the prototype basis function

$$\phi_E(x, y) = \phi(ax + by, cx + dy) = \phi(x', y')$$

Gradients of the basis functions of an arbitrary element

$$\nabla \phi_E(x, y) = \mathbf{A}^T \nabla' \phi(x', y')$$

where $\nabla = (\partial/\partial x, \partial/\partial y)$, $\nabla' = (\partial/\partial x', \partial/\partial y')$.

# Transformation of integrals in 2D

All integrals pick up a prefactor from the change of *dxdy*:

$$\iint_E \phi_{E,m}(x,y)\phi_{E,n}(x,y)dxdy = (\det \mathbf{A}^{-1}) \iint_{\text{proto}} \phi_m(x',y')\phi_n(x',y')dx'dy'$$

With linear elements ($\nabla\phi_E(x,y)$ constant) the gradient integrals are simple:

$$\begin{aligned}\iint_E \nabla\phi_{E,m}(x,y)\cdot\nabla\phi_{E,n}(x,y)dxdy &= \nabla\phi_{E,m}\cdot\nabla\phi_{E,n}(\det \mathbf{A}^{-1}) \iint_{\text{proto}} dx'dy' \\ &= (\mathbf{A}^T\nabla\phi_m)\cdot(\mathbf{A}^T\nabla\phi_n)(\det \mathbf{A}^{-1}) \iint_{\text{proto}} dx'dy'\end{aligned}$$

# Practical implementation of 2D FEM

$N$ points, $N_e$ elements
Mesh is defined by:

- `points` array ($N$x2): each row has the $x$ and $y$ coordinates of a point
- simplices array ($N_e$x3): each row has 3 indices to the `points` array, defining the 3 point of the triangle element

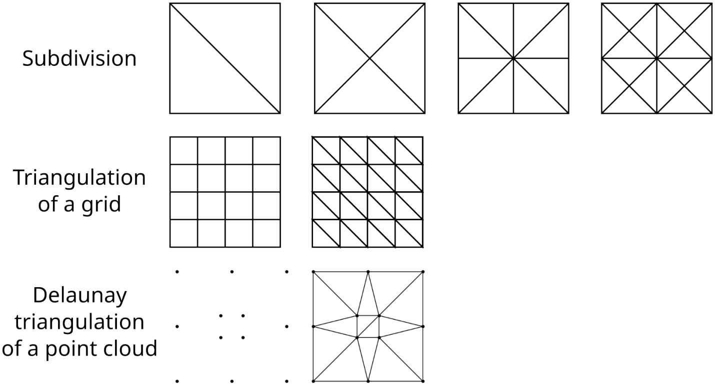
The transformations $\mathbf{A}^{-1}$ and $A$ can be easily calculated from these.

# Practical implementation of 2D FEM

Build the system and mass matrices element by element:

- Get this element's indices of the three points ($i_1$, $i_2$ and $i_3$) and the transform $\mathbf{A}$.
- Calculate gradients of the element's basis functions: $\nabla\phi_E(x, y) = \mathbf{A}^T \nabla\phi(x, y)$
- For the pair of points ($i_1$, $i_2$) in the triangle (and associated basis functions):
  - Add to Laplace operator $L_{i_1,i_2} -= 1/2(\det \mathbf{A}^{-1})(\nabla\phi_{E,1} \cdot \nabla\phi_{E,2})$
  - Add to wave number operator $W_{i_1,i_2} += k_0^2 \epsilon_E (\det \mathbf{A}^{-1})(\int dx' dy' \phi_1 \phi_2)$
  - Add to mass matrix $M_{i_1,i_2} += (\det \mathbf{A}^{-1})(\int dx' dy' \phi_1 \phi_2)$
- Do the same for every other pair of basis functions (9 pairs in total).

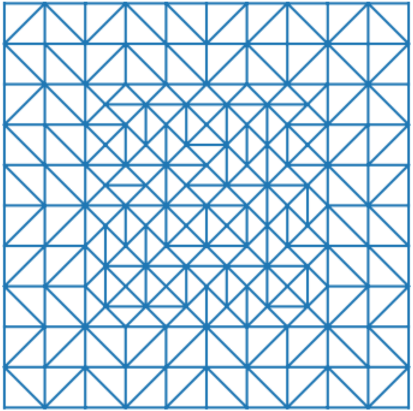
# Mesh construction methods



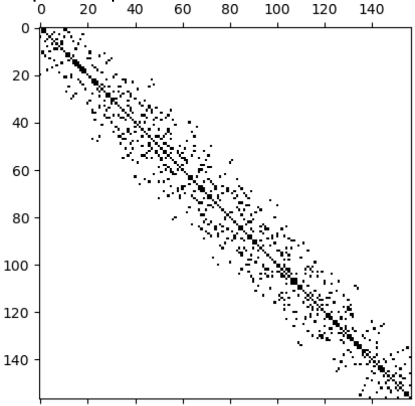

Subdivision can be applied on any triangular mesh: automatic mesh refinement.

# Example

Mesh



Laplace operator matrix



# Ingredients of FEM for vectorial Maxwell's equations

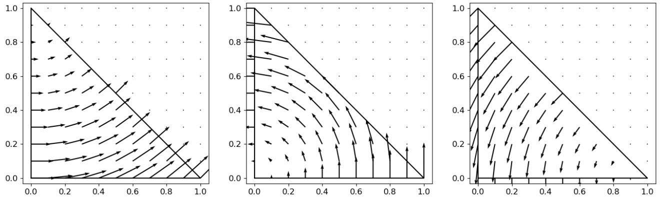
$$\nabla \times \nabla \times \mathbf{E} = k_0^2 \epsilon \mathbf{E}$$

Deriving the eigenvalue form with $\beta^2$ is a bit more complicated

- Vector fields: vector basis functions $\mathbf{E}_n$, weak form

$$-\int_{\partial V} \mathbf{E}_m \times (\nabla \times \mathbf{E}_n) \cdot d\mathbf{S} + \int_V (\nabla \times \mathbf{E}_m) \cdot (\nabla \times \mathbf{E}_n) dV = k_0^2 \int_V \epsilon \mathbf{E}_m \cdot \mathbf{E}_n dV$$

- Allowing discontinuous solutions (most of the time $\nabla \cdot \mathbf{E}$ is not zero): edge-type elements, curl elements



# Finite element method

Computational Photonics 2023