Advanced Computation:
Computational Electromagnetics

FDFD Extras

Outline

• Guided-mode resonance filters
• FDFD for periodic structures
• FDFD for non-periodic and finite-size devices
• Validity of infinitely periodic approximation
• Animating steady-state fields from FDFD
• Parameter sweeps
• Dielectric averaging
• Gaussian beam source
• Cylindrical wave source
• Multiple sources
• Waveguide problems
• Calculating band diagrams using FDFD
• Iterative Solution
• Formulation with a stretched coordinate PML
• 3D anisotropic FDFD

Covered in Course
Bonus
What is a Guided-Mode Resonance Filter?

A guided-mode resonance (GMR) filter is both a diffraction grating and a slab waveguide.

A resonance occurs when a diffracted mode exactly matches a guided mode.

Away from resonance, the device behaves like an ordinary multilayer structure.

On resonance, the device reverses the background response (roughly speaking).
Qualitative Description of the GMR Response

Away From Resonance

Away from resonance, the GMR filter exhibits the “background” response of the multilayer structure.

At Resonance

On resonance, part of the applied wave is coupled into a guided mode. The guided mode slowly “leaks” out from the waveguide out of phase with the applied wave. The “leaked” wave interferes with the applied wave to produce the GMR filter response.

Role of GMRFs in CEM

Guided-mode resonance filters (GMRFs) are highly sensitive to just about everything.

- Mechanical dimensions
- Material properties (both internal and external)
- Angle of incidence
- Polarization
- Frequency

GMRFs are excellent devices to benchmark a new electromagnetic modeling code!

If anything is wrong with your code, this device will amplify it.

This is still no guarantee your code is correct, but close to it.
Rumpf’s Typical Benchmarking GMR

\[ T = 134 \text{ nm} \]
\[ \Lambda = 314 \text{ nm} \]
\[ n_1 = 1.0 \]
\[ n_2 = 1.52 \]
\[ n_L = 2.0 \]
\[ n_H = 2.1 \]
\[ f = 0.5 \]


FDFD for Periodic Structures
Recall 3D to 2D Representation

Typical Grid Scheme for Periodic Devices
Suppose we have the following field solution to our FDFD grating simulation.

Now suppose we wish to extrapolate the solution over multiple unit cells. If we just stack the solution, we get...

Notice the discontinuities? This is not correct!! 😞
Extrapolating Field Solution Across Multiple Unit Cells (3 of 3)

We forgot to include the phase discontinuity due to having an oblique angle of incidence.

\[ E_z(x, y) \]
\[ E_z(x, y)e^{ikz} \]
\[ E_z(x, y)e^{2ikz} \]

Field is continuous. This is correct!! 😊

Good Simulation

Here are the signs of a good simulation:

- All lines are smooth and continuous.
- There are no triangles or gaps.
- Conservation of energy is flat across the entire plot.
- No “rolling” behavior is seen.
Triangles in Parameter Sweeps are Bad

Parameter sweeps should be smooth and continuous. Triangles indicate that there are abrupt features that you are not resolving sufficiently. You may be missing something important!

Hint: you can use a non-uniform step size in a parameter sweep.

LAMBDA = [540:1:550, 550.01:0.01:552.99, 553:1:560];

Impact of Grid Resolution

Poor grid resolution can cause about every error there is.

1. Rolling behavior in the spectral response.
2. Violation in conservation of energy
3. Shifted spectral response
4. Inaccurate results

There is an overall trend from bad to better results with increasing wavelength (or decreasing frequency).
Impact of PML Size

When the PML is too small, reflections will be severe and a standing wave will be produced. This standing wave produces a “rolling” effect in the spectral response of a device.

Sign: Results do not get better for different wavelengths or frequencies.

Impact of Spacer Region

When the spacer regions are too small, evanescent fields penetrate the PML and couple energy into propagating waves. Since the field is evanescent in the record plane, the power is not accounted for.
FDFD for Nonperiodic and Finite-Size Devices

Typical Grid Scheme for Finite Devices
Validity of Infinitely Periodic Approximation

The field in a finite periodic device is very nearly periodic away from the edges. For this device, the infinitely periodic approximation does not predict the field in outermost four (or so) unit cells.
Finite Size GMRF Devices

Animating Steady-State Fields from FDFD
Animation of a Steady-State Field

Example Steady-State Field
Calculated with FDFD

Animated Steady-State Field

MATLAB Code

The trick is to keep plotting the field as usual, but add phase each time.

```matlab
% CALCULATE PHASE OF EACH FRAME
NPHI = 100;
phase = linspace(0,2*pi,NPHI);

% DRAW FRAMES
for nphi = 1 : NPHI

% Add Phase to Field Ez
f = Ez*exp(li*phase(nphi));

% Draw Field With Added Phase
pcolor(xa,ya,real(f')); shading interp;
axis equal tight;
drawnow;
end
```
What is a Parameter Sweep?

So far, we have learned to model a single device at a single frequency, or wavelength.

Suppose we calculate this data as we continuously change one or more parameters? This is called a parameter sweep.
It is almost always a good idea to display results during simulation. This lets you catch mistakes and abort early.

Make a Generic Function for FDFD

A great way to simplify programming your parameter sweeps is to first make a generic function out of your model.

For FDFD, you model will take as input arguments:

- **Source:** \( \lambda_0, \theta, \phi, \) polarization, etc.
- **Device:** UR2, ER2, grid parameters, etc.

Given these input arguments, your function will model the device and perhaps even calculate diffraction efficiencies, reflectance, transmittance, etc. It may return:

- **Field:** \( E_z \) or \( H_z \), other field components, etc.
- **Response:** RDE, TDE, REF, TRN, CON, etc.
Example Header for FDFD Function

```matlab
function DAT = fdfd2d(DEV,SRC)
    % FDFD2D Two-Dimensional Finite-Difference Frequency-Domain
    % DAT = fdfd2d(DEV,SRC)
    % INPUT ARGUMENTS
    % =================
    % DEV   Device Parameters
    %   .UR2    Relative permeability on 2X grid
    %   .ER2    Relative permittivity on 2X grid
    %   .NPML   Size of PML on 1X grid [xlo xhi ylo yhi]
    %   .RES    [dx2 dy2] grid resolution of 2X grid
    % SRC
    %   .lam0   free space wavelength
    %   .theta  Angle of incidence
    %   .MODE   Mode: 'E' or 'H'
    % OUTPUT ARGUMENTS
    % =================
    % DAT       Output Data
    %   .RDE    Array of diffraction efficiencies of reflected harmonics
    %   .REF    Overall Reflectance
    %   .TDE    Array of diffraction efficiencies of transmitted harmonics
    %   .TRN    Overall Transmittance
    %   .CON    Conservation of Energy
    %   .F      Field
```

Block Diagram of `fdfd2d()`

1. Dashboard: Source, device, learn, and grid
2. Calculate Grid: Nx, Ny, dx, dy, xa, ya
3. Build Device on Grid
4. Incorporate PML
5. Calculate Wave Vector Components: \( k_{x,\text{inc}}, k_{y,\text{src}}, k(m), k_{x,\text{ref}}(m), k_{y,\text{trans}}(m) \)
6. Build Wave Matrix \( A \)
7. Compute Source: \( b = (QA - AQ)\bar{E}_\text{inc} \)
8. Solve \( Ax = b \)
9. Post-Process: RDE, TDE, REF, TRN, CON

All hard-coded numbers. No work.

No hard-coded numbers. All work.
Wavelength or Frequency Parameter Sweep

By far, the most common parameter sweep is calculating the device behavior as a function of frequency or wavelength.

```
for nfreq = 1 : NFREQ
    SRC.lam0 = c0/FREQ(nfreq);
    DAT = fdfd2d(DEV, SRC);
    REF(nfreq) = DAT.REF;
end
```

```
for nlam = 1 : NLAM
    SRC.lam0 = LAMBDA(nlam);
    DAT = fdfd2d(DEV, SRC);
    REF(nlam) = DAT.REF;
end
```

Importance of Code Sequence (1 of 2)

**Fundamental Concept of Parameter Sweep**

Adjust only one parameter.
Keep all others fixed if at all possible.

The jagged lines are due to too many parameters changing at the same time.

1. lam0, dx, dy, Nx, Ny, BUFZ, placement of device, etc.
Importance of Code Sequence (2 of 2)

Fundamental Concept of Parameter Sweeps
Adjust only one parameter. Keep all others fixed if at all possible.

Grid resolution is determined according to the highest frequency, or shortest wavelength.

Spacer regions are determined according to the resonant frequency, lowest frequency, or longest wavelength.

Response Generated When Device is Built Before Frequency

Incorporating Material Dispersion

When simulating devices composed of materials that are dispersive, it is necessary to account for this in the parameter sweep since the material properties are changing as a function of frequency (or wavelength).

New step is added here to adjust the values of permittivity and permeability for the dispersive materials at wavelength $\lambda_0$.

Do NOT rebuild the device. Just overwrite the values of $\varepsilon$ and $\mu$ on the grid.
Dielectric Averaging

What is Dielectric Averaging?

Suppose we have a grid, but the size of the device is not an exact integer number of grid cells. What can we do?

Based on the above, we build our device on the grid as follows:

We look at a close-up of the cell in question and calculate a weighted average.

Based on the above, we build our device on the grid as follows:
Reason for Dielectric Averaging

• Dielectric averaging improves the rate of convergence.
• You can get away with coarser grid resolution using dielectric averaging. Your simulations will run faster and be more memory efficient.
  • Perhaps NRES=14 instead of NRES=30.


Typical Staircase Approximation
Representation with Dielectric Averaging

Physical Device

Representation on Grid with Dielectric Averaging

Device Modeled

DO NOT BLUR MORE THAN ONE PIXEL DISTANCE. This will artificially suppress reflections from your structures.

Procedure to Perform Averaging

Staircase representation on a higher resolution grid

Blurred version of the high-resolution device

Extract center value from each low resolution cell

We can use a convolution to calculate the average dielectric constant in each cell.
Suppose you have the product of two functions that you are approximating numerically.

\[ f(z) = a(z) \cdot b(z) \]

Slow convergence is encountered whenever one or both of these functions is discontinuous.

Suppose \( a(z) \) has a discontinuity. Numerical convergence can be significantly improved by “averaging” the discontinuous function at the discontinuity.

\[ f(z) = \langle a(z) \rangle \cdot b(z) \]
Double Discontinuity (1 of 2)

Suppose both \( a(z) \) and \( b(z) \) are discontinuous at the same point. It is very difficult to improve the convergence of this problem.

There exists, however, a special case where \( a(z) \) and \( b(z) \) are discontinuous at the same point, but their product is continuous. That is, \( f(z) \) is continuous.

We can rearrange the equation so that only a single discontinuity exists on each side.

\[
\frac{1}{a(z)} \cdot f(z) = b(z)
\]

Double Discontinuity (2 of 2)

We can now improve convergence by averaging \( 1/a(z) \).

\[
\frac{1}{a(z)} \cdot f(z) = b(z)
\]

Moving the \( a(z) \) term to the right-hand side leads to

\[
f(z) = \frac{1}{\left\langle \frac{1}{a(z)} \right\rangle} \cdot b(z)
\]

We conclude that in the double discontinuity case where the product is continuous, we average the reciprocal of \( a(z) \) and then reciprocate the averaged function.
**Summary of Smoothing**

When all functions are continuous, no averaging is needed.

\[ f(z) = a(z) \cdot b(z) \]

When one of the functions has a step discontinuity, convergence is improved by averaging the discontinuous function.

\[ f(z) = \left( a(z) \right) \cdot b(z) \]

When both \( a(z) \) and \( b(z) \) are discontinuous at the same point, but their product is continuous, convergence is improved by averaging the reciprocal of one of the functions.

\[ f(z) = \left( a^{-1}(z) \right)^{-1} \cdot b(z) \]

---

**Smoothing and Maxwell’s Equations**

In Maxwell’s equations, we have the product of two functions...

\[ \vec{D}(z) = \varepsilon_r(z) \cdot \vec{E}(z) \]

The dielectric function is discontinuous at the interface between two materials. Boundary conditions require that

\[ E_{1,||} = E_{2,||} \quad \text{Tangential component is continuous across the interface} \]

\[ \varepsilon_1 E_{1,\perp} = \varepsilon_2 E_{2,\perp} \quad \text{Normal component is discontinuous across the interface, but the product of } \varepsilon \text{ is continuous.} \]

We conclude that we must smooth the dielectric function differently for the tangential and normal components. This implies that the smoothed dielectric function will be anisotropic and described by a tensor.
In Maxwell’s equations, we have

\[ \vec{D} = [\varepsilon] \vec{E} \]

The dielectric tensor is

\[
[\varepsilon] = \begin{bmatrix}
\varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\
\varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\
\varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz}
\end{bmatrix}
\]

The electric field can be written as the sum of parallel and perpendicular polarizations.

\[ \vec{E} = \vec{E}_\parallel + \vec{E}_\perp \]

For an arbitrarily shaped device, these components can vary across the grid. Suppose we could calculate a vector function throughout the grid that is normal to all the interfaces. This called the “normal vector” field.

\[ \hat{n}(x, y, z) \]
The perpendicular component of $E$ can be computed from the normal vector field as follows.

$$\vec{E}_\perp = \hat{n} \left( \hat{n} \bullet \vec{E} \right)$$

It follows that the parallel component of $E$ is

$$\vec{E}_\parallel = \vec{E} - \vec{E}_\perp = \vec{E} - \hat{n} \left( \hat{n} \bullet \vec{E} \right)$$

In matrix notation, the perpendicular component can be written as

$$\vec{E}_\perp = \hat{n} \left( \hat{n} \bullet \vec{E} \right) \rightarrow \begin{bmatrix} E_x^\perp \\ E_y^\perp \\ E_z^\perp \end{bmatrix} = \begin{bmatrix} n_x \left( n_x E_x + n_y E_y + n_z E_z \right) \\ n_y \left( n_x E_x + n_y E_y + n_z E_z \right) \\ n_z \left( n_x E_x + n_y E_y + n_z E_z \right) \end{bmatrix}$$

$$= \begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{bmatrix} \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$
Let
\[
\hat{N} = \begin{bmatrix}
n_x^2 & n_x n_y & n_x n_z \\
n_y n_x & n_y^2 & n_y n_z \\
n_z n_x & n_z n_y & n_z^2
\end{bmatrix}
\]

The perpendicular and parallel components are then
\[
\begin{align*}
\vec{E}_\perp &= \hat{n} (\hat{n} \cdot \vec{E}) = \hat{N} \vec{E} \\
\vec{E}_\parallel &= \vec{E} - \hat{N} \vec{E} = ([I] - \hat{N}) \vec{E}
\end{align*}
\]

The constitutive relation can be written in terms of the parallel and perpendicular components of the \(\vec{E}\) field.
\[
\tilde{D} = [\varepsilon] (\vec{E}_\parallel + \vec{E}_\perp)
\]
\[
= [\varepsilon] \vec{E}_\parallel + [\varepsilon] \vec{E}_\perp
\]

We average the dielectric functions differently according to our rules for optimum convergence.
\[
\tilde{D} = \begin{bmatrix} \varepsilon_\parallel \end{bmatrix} \vec{E}_\parallel + \begin{bmatrix} \varepsilon_\perp \end{bmatrix} \vec{E}_\perp
\]
\[
\begin{bmatrix} \varepsilon_\parallel \end{bmatrix} = \langle \varepsilon \rangle
\]
\[
\begin{bmatrix} \varepsilon_\perp \end{bmatrix} = \langle \varepsilon^{-1} \rangle^{-1}
\]
Putting all of this together leads to

\[
\tilde{D} = \begin{bmatrix} \varepsilon_\parallel \end{bmatrix} \tilde{E}_{\parallel} + \begin{bmatrix} \varepsilon_\perp \end{bmatrix} \tilde{E}_{\perp}
\]

\[
= \begin{bmatrix} \varepsilon_\parallel \end{bmatrix} \left( [I] - [\hat{N}] \right) \tilde{E} + \begin{bmatrix} \varepsilon_\perp \end{bmatrix} [\hat{N}] \tilde{E}
\]

\[
= \left\{ \begin{bmatrix} \varepsilon_\parallel \end{bmatrix} \left( [I] - [\hat{N}] \right) + \begin{bmatrix} \varepsilon_\perp \end{bmatrix} [\hat{N}] \right\} \tilde{E}
\]

We can derive an effective dielectric tensor from this equation.

\[
[\varepsilon_{\text{smooth}}] = \begin{bmatrix} \varepsilon_\parallel \end{bmatrix} \left( [I] - [\hat{N}] \right) + \begin{bmatrix} \varepsilon_\perp \end{bmatrix} [\hat{N}]
\]

\[
= \begin{bmatrix} \varepsilon_\parallel \end{bmatrix} + \left( \begin{bmatrix} \varepsilon_\perp \end{bmatrix} - \begin{bmatrix} \varepsilon_\parallel \end{bmatrix} \right) [\hat{N}]
\]

Summary of Formulation

Given the simulation problem defined by

\[
\tilde{D} = [\varepsilon] \tilde{E}
\]

We can improve the convergence rate by smoothing the dielectric function according to

\[
[\varepsilon_{\text{smooth}}] = \begin{bmatrix} \varepsilon_\parallel \end{bmatrix} + \left( \begin{bmatrix} \varepsilon_\perp \end{bmatrix} - \begin{bmatrix} \varepsilon_\parallel \end{bmatrix} \right) [\hat{N}]
\]

\[
\begin{bmatrix} \varepsilon_\parallel \end{bmatrix} = \langle [\varepsilon] \rangle
\]

\[
\begin{bmatrix} \varepsilon_\perp \end{bmatrix} = \langle [\varepsilon]^{-1} \rangle^{-1}
\]
Dielectric Averaging of a Sphere (1 of 2)

Given a sphere with dielectric constant $\varepsilon_r=5.0$ in air and in a grid with $N_x=N_y=N_z=25$ cells, the dielectric tensor after averaging is:

- $\varepsilon_{xx}$
- $\varepsilon_{yy}$
- $\varepsilon_{zz}$
- $\varepsilon_{xy}$
- $\varepsilon_{xz}$
- $\varepsilon_{zx}$

Dielectric Averaging of a Sphere (2 of 2)

A 3D visualization is:
Comments on Dielectric Averaging

• Even if the original dielectric function is isotropic, the averaged dielectric function is anisotropic.
• Anisotropic averaging requires calculating the normal vector field. This can be difficult, especially for arbitrary structures.
• Convergence still tends to improve even when only isotropic averaging is used.

\[
\epsilon_{\text{smooth}} \approx \langle \epsilon \rangle
\]

• The \( E_z \) mode does not require anisotropic averaging of the dielectric function.
• The \( H_z \) mode does not require anisotropic averaging of the permeability.

Gaussian Beam Source
Anatomy of a Gaussian Beam (1 of 2)

Amplitude Profile

\[ A(x) = e^{-\left(\frac{x-x_0}{2}\right)^2} \]

Phase Accumulation (i.e. Oscillation)

\[ G(x, y) = A(x)e^{jk_0ny} \]

Anatomy of a Gaussian Beam (2 of 2)

%% BEAM SOURCE

\[
[Y, X] = \text{meshgrid}(ya, xa);
fsrc = \exp(-2*X/w).^2.*\exp(-1i*k0*n*Y);
\]

\[ e^{-\left(\frac{x-x_0}{2}\right)^2} e^{jk_0ny} = e^{-\left(\frac{x-x_0}{2}\right)^2} e^{jk_0ny} \]
Implementing a Gaussian beam source in FDFD only requires a different calculation of the source field $f_{src}$.

$$f_{src}(x,y)$$

Simulated Field

The simulated beam will actually diverge. In principle, we should make $f_{src}(x,y)$ diverge as well, but that is usually not needed.

Oblique Gaussian Beam Source

```matlab
[Y,X] = meshgrid(ya,xa);
[TH,R] = cart2pol(X,Y);
[X,Y] = pol2cart(TH+theta,R);
nxs = round(0.3*Nx);
%horizontal position of beam center
X = X - X(nxs,1);
fsrc = exp(-(2*X/w).^2).*exp(-1i*k*n*Y); %horizontal position of beam center
```

% BEAM SOURCE

The simulated beam will actually diverge. In principle, we should make $f_{src}(x,y)$ diverge as well, but that is usually not needed.

Oblique Gaussian Beam Source

```matlab
[Y,X] = meshgrid(ya,xa);
[TH,R] = cart2pol(X,Y);
[X,Y] = pol2cart(TH+theta,R);
nxs = round(0.3*Nx);
%horizontal position of beam center
X = X - X(nxs,1);
fsrc = exp(-(2*X/w).^2).*exp(-1i*k0*n*Y); %horizontal position of beam center
```
Oblique Gaussian Beam Source in FDFD

Controlling Beam Divergence

$w = 5A_0$  $w = 2A_0$

Wider beams exhibit better collimation.
The closed form expression for a cylindrical wave source is

\[ f_{\text{src}}(R) = \frac{A}{\sqrt{R}} \exp(jkR) \quad R = \sqrt{x^2 + y^2} \]

\[ \% \text{SOURCE FIELD} \\
[Y,X] = \text{meshgrid}(ya,xa); \\
R = \text{sqrt}(X.^2 + Y.^2); \\
fsrc = \exp(1i*k0*n*R)./\text{sqrt}(R) \]
Design of TF/SF Interface

The TF/SF interface must enclose the center of the source.

The easiest and most aesthetic is to tightly encircle the center of the source.

% CONSTRUCT Q
Q = R < (0.25 * lam0);
Q = diag(sparse(Q(:)));
Multiple Sources

Rules for Sources in FDFD

• The source field must be a rigorous solution to Maxwell’s equations.
• The TF/SF interface must completely enclose the source.
  • Extends all the way across the grid (implied out to $\infty$).
  • A closed contour around the source.
• When multiple sources are used, the source vector must be calculated independently for each source and then summed.

$$b_{\text{overall}} = b_1 + b_2 + \cdots + b_N$$

$N \equiv$ number of sources
Assembling Multiple Sources

$b_1$ and $b_2$ are calculated independently as if they were each the only source to be implemented.

The overall source vector $b$ is the sum of these.

Waveguide Problems
Extracting the Slab Waveguide(s)

A “window” just outside the PML is used for the source and to analyze reflected energy.

Another “window” just outside the PML is used for analyzing transmitted energy.

Calculate All the Eigen-Modes in the Cross Section(s)

Recall the formulation to analyze slab waveguides as an eigen-value problem.

\[
\begin{align*}
\text{E Mode:} & \quad -\left( D_x^s \mu_x D_x^s + \varepsilon_{yy} \right) a_y = \gamma^2 \mu_x a_y \\
\text{H Mode:} & \quad -\left( D_y^s \varepsilon_{zz} D_y^s + \mu_{yy} \right) b_y = \gamma^2 \varepsilon_{zz} b_y
\end{align*}
\]

A sampling from the complete set of eigen-modes is

External waves | Leaky modes | Guided modes

Slide 76
Field In Terms of Eigen-Modes

The field across the waveguide must be a linear sum of the eigen-modes.

\[
e_{\parallel} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = a_1 \mathbf{v}_1 + a_2 \mathbf{v}_2 + a_3 \mathbf{v}_3 + a_4 \mathbf{v}_4 + a_5 \mathbf{v}_5
\]

Identifying the Fundamental Mode

For most simulations, it is desired to excite waveguides with the fundamental mode. It is necessary to distinguish the fundamental mode from the other modes.

Here are some things that can be used to help identify the fundamental mode...

- The effective refractive index should have a very small imaginary part.
- The real part of the effective refractive index is typically just slightly lower than the refractive index of the core.
- The effective refractive index will be greater than the refractive index of the cladding materials.
- Usually the fundamental mode has the largest effective refractive index of all the modes.

\[
\text{Im}[n_{\text{eff}}] \to 0 \\
\text{Re}[n_{\text{eff}}] \approx n_{\text{core}} \\
\text{Re}[n_{\text{eff}}] > n_{\text{clad}}
\]
Calculating the Source Function

For plane wave sources, the source was calculated across a homogeneous grid. This is not what should be done for waveguide sources. Instead, the waveguide should be made to be continuous.

Field Solution Using FDFD

After the FDFD simulation, we know the field across the entire grid.

10/10/2019
Calculating the Energy in Each Mode

Using FDFD, we calculate the steady-state field $\vec{e}_z$ around the input and output(s) of the waveguide circuit. We can then calculate the amplitudes of all the modes.

$$\vec{e}^\text{ref}_z = V a^\text{ref} \rightarrow a^\text{ref} = V^{-1} \vec{e}^\text{ref}_z$$

$$\vec{e}^\text{tm}_z = V a^\text{tm} \rightarrow a^\text{tm} = V^{-1} \vec{e}^\text{tm}_z$$

Now we can calculate the fraction of power in all of the modes.

$$p^\text{ref} = \left| \frac{1}{a^\text{inc}} a^\text{ref} \right|^2$$

$$p^\text{tm} = \left| \frac{1}{a^\text{inc}} a^\text{tm} \right|^2$$

Most of the time we only care about the fraction of power in the fundamental mode.

$$p^\text{ref} = \left| \frac{a^\text{ref}}{a^\text{inc}} \right|^2$$

$$p^\text{tm} = \left| \frac{a^\text{tm}}{a^\text{inc}} \right|^2$$

Example Transmission Calculation

Assuming the waveguide was sourced with only the fundamental mode with unit amplitude...

$$a^\text{inc} = \begin{bmatrix} 1.0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$a^\text{tm} = \begin{bmatrix} 0.87 \\ 0.02 \\ 0.06 \\ 0.15 \end{bmatrix}$$

$$T_0 = \frac{0.87^2}{1} \approx 76\%$$
Calculating Band Diagrams Using FDFD

Don’t Normalize the Grid

We will derive an eigen-value problem from Maxwell’s equations where the frequency term \( k_0 \) will be the eigen-value.

Since \( k_0 \) this is the unknown quantity in this framework, we cannot use it to normalize the grid.

\[
\begin{align*}
\frac{\partial}{\partial y} E_y - \frac{\partial}{\partial z} E_z &= k_0 \mu_{\omega} H_y, \\
\frac{\partial}{\partial z} E_z - \frac{\partial}{\partial x} E_x &= k_0 \mu_{\omega} H_z, \\
\frac{\partial}{\partial x} E_x - \frac{\partial}{\partial y} E_y &= k_0 \mu_{\omega} H_x, \\
\frac{\partial}{\partial y} H_y - \frac{\partial}{\partial z} H_z &= k_0 \varepsilon_{\omega} E_y, \\
\frac{\partial}{\partial z} H_z - \frac{\partial}{\partial x} H_x &= k_0 \varepsilon_{\omega} E_z, \\
\frac{\partial}{\partial x} H_x - \frac{\partial}{\partial y} H_y &= k_0 \varepsilon_{\omega} E_x, \\
\Delta y \left( E_y^{n+1,k} - E_y^{n,k} \right) &= k_0 \mu_{\omega} \frac{H_y}{\Delta z}, \\
\Delta z \left( E_z^{n+1,k} - E_z^{n,k} \right) &= k_0 \mu_{\omega} \frac{H_z}{\Delta y}, \\
\Delta x \left( E_x^{n+1,k} - E_x^{n,k} \right) &= k_0 \mu_{\omega} \frac{H_x}{\Delta y}, \\
\Delta y \left( H_y^{n+1,k} - H_y^{n,k} \right) &= k_0 \varepsilon_{\omega} \frac{E_y^{n+1,k}}{\Delta z}, \\
\Delta z \left( H_z^{n+1,k} - H_z^{n,k} \right) &= k_0 \varepsilon_{\omega} \frac{E_z^{n+1,k}}{\Delta x}, \\
\Delta x \left( H_x^{n+1,k} - H_x^{n,k} \right) &= k_0 \varepsilon_{\omega} \frac{E_x^{n+1,k}}{\Delta y}.
\end{align*}
\]
Incorporate $\beta$ as a Periodic Boundary Condition

Our algorithm will be given a Bloch wave vector $\beta$.

This is incorporated into the periodic boundary conditions on ALL grid boundaries.

This of $\beta$ as $k_{\text{inc}}$ when incorporating the boundary conditions into the derivative matrices.

Formulation of 3D Eigen-Value Problem (1 of 2)

We can write our six equations in block matrix form.

$$
\begin{align*}
D_y^* e_y - D_z^* e_z &= k_\beta \mu_{\text{inc}} \hat{h}, \\
D_x^* e_x - D_y^* e_y &= k_\beta \mu_{\text{inc}} \hat{h}, \\
D_x^* e_x - D_z^* e_z &= k_\beta \mu_{\text{inc}} \hat{h}, \\
D_y^* \hat{h}_y - D_z^* \hat{h}_z &= k_\beta \mu_{\text{inc}} e_x, \\
D_z^* \hat{h}_z - D_x^* \hat{h}_x &= k_\beta \mu_{\text{inc}} e_y, \\
D_x^* \hat{h}_x - D_y^* \hat{h}_y &= k_\beta \mu_{\text{inc}} e_z.
\end{align*}
$$

$$
\begin{align*}
C \hat{e} &= k_\beta [\mu] \hat{h}, \\
C^\dagger \hat{e} &= k_\beta [\varepsilon] \hat{h}, \\
C^\dagger \hat{h} &= k_\beta [\varepsilon^\dagger] \hat{e}.
\end{align*}
$$

$C$ and $C^\dagger$ are block matrices defined as:

\begin{align*}
C &= \begin{bmatrix}
0 & -D_x^* & D_y^* \\
D_y^* & 0 & -D_z^* \\
-D_z^* & D_x^* & 0
\end{bmatrix}, \\
C^\dagger &= \begin{bmatrix}
0 & -D_x^* & D_y^* \\
-D_y^* & D_x^* & 0 \\
D_z^* & 0 & -D_x^*
\end{bmatrix}.
\end{align*}
Formulation of 3D Eigen-Value Problem (2 of 2)

Can derive an eigen-value problem in terms of the electric field or in terms of the magnetic field.

Electric Field Formulation

\[ \mathbf{C}^e \left[ \varepsilon \right]^{-1} \mathbf{C}^e \mathbf{\hat{e}} = \frac{1}{k_0^2} \left[ \varepsilon \right] \mathbf{\hat{e}} \]

Magnetic Field Formulation

\[ \mathbf{C}^m \left[ \mu \right]^{-1} \mathbf{C}^m \mathbf{\hat{h}} = k_0^2 \left[ \mu \right] \mathbf{\hat{h}} \]

These have the form of a generalized eigen-value problem and they are solved in that manner.

\[ \mathbf{Ax} = \lambda \mathbf{Bx} \]

\[ [\mathbf{V}, \mathbf{D}] = \text{eig}(\mathbf{A}, \mathbf{B}); \]

This calculates 10 modes.

For faster computation: (1) use sparse matrices, (2) make an initial guess at eigen-values, and (3) only calculate the minimum number of modes needed.

Formulation of 2D Eigen-Value Problem (1 of 2)

We will assume our device is uniform in the z-direction and that wave propagation is restricted to be in the x-y plane. When this the case, we have

\[ \mathbf{D}_x^e = \mathbf{D}_y^e = 0 \]

We see that Maxwell’s equations decouple into two distinct modes.

**E Mode**

\[ \mathbf{D}^e \mathbf{\hat{e}}_x = \mathbf{\hat{e}}_x, \quad \mathbf{D}^e \mathbf{\hat{e}}_y = \mathbf{\hat{e}}_y, \quad \mathbf{D}^e \mathbf{\hat{h}}_x = \mathbf{\hat{h}}_x, \quad \mathbf{D}^e \mathbf{\hat{h}}_y = \mathbf{\hat{h}}_y \]

**H Mode**

\[ \mathbf{D}^e \mathbf{\hat{e}}_x = \mathbf{\hat{e}}_x, \quad \mathbf{D}^e \mathbf{\hat{e}}_y = \mathbf{\hat{e}}_y, \quad \mathbf{D}^e \mathbf{\hat{h}}_x = \mathbf{\hat{h}}_x, \quad \mathbf{D}^e \mathbf{\hat{h}}_y = \mathbf{\hat{h}}_y \]
Formulation of 2D Eigen-Value Problem (2 of 2)

We can now derive an eigen-value problem for each mode.

\[ \begin{align*} 
E \text{ Mode} && H \text{ Mode} \\
\left( \mathbf{D}' \epsilon e + \mathbf{D}' \mu e \right) \mathbf{e}_z &= -k_0^2 \mathbf{e}_z \\
\mathbf{h}_x &= \frac{1}{k_0} \mu \mathbf{D}' \mathbf{e}_z \\
\mathbf{h}_y &= -\frac{1}{k_0} \mu \mathbf{D}' \mathbf{e}_z \\
\left( \mathbf{D}' \epsilon e + \mathbf{D}' \mu e \right) \mathbf{h}_z &= -k_0^2 \mathbf{h}_z \\
\mathbf{e}_x &= \frac{1}{k_0} \epsilon \mathbf{D}' \mathbf{h}_z \\
\mathbf{e}_y &= -\frac{1}{k_0} \epsilon \mathbf{D}' \mathbf{h}_z
\end{align*} \]

Formulation of 1D Eigen-Value Problem (1 of 2)

For 1D configuration, we will assume our device is uniform in the x and y directions and that propagation is restricted to just the z-direction. In this case, we have

\[ \mathbf{D}'_z = \mathbf{D}'_y = \mathbf{D}'_x = 0 \]

We see that Maxwell's equations again decouple into two distinct modes.

Ex Mode
\[ \begin{align*} 
\mathbf{D}' \mathbf{e}_x - \mathbf{D}' \mathbf{e}_z &= k \mu \mathbf{h}_y \\
\mathbf{D}' \mathbf{e}_y - \mathbf{D}' \mathbf{e}_z &= k \mu \mathbf{h}_y \\
\mathbf{D}' \mathbf{h}_x - \mathbf{D}' \mathbf{h}_z &= k \epsilon \mathbf{e}_y \\
\mathbf{D}' \mathbf{h}_y - \mathbf{D}' \mathbf{h}_z &= k \epsilon \mathbf{e}_y \\
\mathbf{D}' \mathbf{h}_z - \mathbf{D}' \mathbf{h}_z &= k \epsilon \mathbf{e}_y \\
\mathbf{D}' \mathbf{h}_z - \mathbf{D}' \mathbf{h}_z &= k \epsilon \mathbf{e}_y \\
\mathbf{D}' \mathbf{h}_z - \mathbf{D}' \mathbf{h}_z &= k \epsilon \mathbf{e}_y \\
\mathbf{D}' \mathbf{h}_z - \mathbf{D}' \mathbf{h}_z &= k \epsilon \mathbf{e}_y \\
\mathbf{D}' \mathbf{h}_z - \mathbf{D}' \mathbf{h}_z &= k \epsilon \mathbf{e}_y
\end{align*} \]

Ey Mode
\[ \begin{align*} 
\mathbf{D}' \mathbf{e}_x - \mathbf{D}' \mathbf{e}_z &= k \mu \mathbf{h}_y \\
\mathbf{D}' \mathbf{e}_y - \mathbf{D}' \mathbf{e}_z &= -k \mu \mathbf{h}_y \\
\mathbf{D}' \mathbf{h}_x - \mathbf{D}' \mathbf{h}_z &= 0 \\
\mathbf{D}' \mathbf{h}_y - \mathbf{D}' \mathbf{h}_z &= 0 \\
\mathbf{D}' \mathbf{h}_z - \mathbf{D}' \mathbf{h}_z &= 0 \\
\mathbf{D}' \mathbf{h}_z - \mathbf{D}' \mathbf{h}_z &= 0 \\
\mathbf{D}' \mathbf{h}_z - \mathbf{D}' \mathbf{h}_z &= 0 \\
\mathbf{D}' \mathbf{h}_z - \mathbf{D}' \mathbf{h}_z &= 0
\end{align*} \]
Formulation of 1D Eigen-Value Problem (2 of 2)

For each mode, we can derive two different eigen-value problems. These will give the same results, but one may be more numerically efficient than the other depending on your device and its material properties.

**Ex Mode**

\[
\begin{align*}
D_{xx}^1 \mu_{yy}^i D_x^i e_x &= -k_0^2 e_x e_x \\
\hat{h}_y &= \frac{1}{k_0} \mu_{yy} D_x^i e_x \\
D_x^i e_y D_y^i \hat{h}_x &= -k_0^2 \mu_{xx} \hat{h}_x \\
e_x &= \frac{1}{k_0} \varepsilon_{xx}^i D_x^i \hat{h}_y \\
\end{align*}
\]

**Ey Mode**

\[
\begin{align*}
D_{xx}^1 \mu_{yy}^i D_x^i e_y &= -k_0^2 e_y e_y \\
\hat{h}_x &= \frac{1}{k_0} \mu_{xx} D_x^i e_y \\
D_x^i e_x D_y^i \hat{h}_y &= -k_0^2 \mu_{yy} \hat{h}_y \\
e_y &= \frac{1}{k_0} \varepsilon_{yy}^i D_x^i \hat{h}_x \\
\end{align*}
\]

Periodic Boundary Conditions

For calculating electromagnetic bands, we use periodic boundary conditions at all boundaries.

Instead of \( k_{\text{inc}} \) defining the phase at the boundaries, we use the Bloch wave vector \( \vec{\beta} \).

For example,

\[
D_x^i = \frac{1}{\Delta x} \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & -1 & 1 \\
e^{i \vec{\beta} \cdot \vec{x}} & 0 & 0 & 0 & -1
\end{bmatrix}
\]
Iterative Solution to FDFD

What is an Iterative Solution?

Direct Solution

- Dashboard
- Compute Grid
- Build Device
- Perform FDFD
- Show Results

\[ \mathbf{x} = \mathbf{A} \backslash \mathbf{b}; \]

Iterative Solution

- Dashboard
- Compute Grid
- Build Device
- Perform FDFD
- Show Results

\begin{verbatim}
  tol = 1e-10;
  D = A.*I;
  f = zeros(Nx*Ny,1);
  err = inf;
  while err > tol
    df = D\(b - A*f);
    f = f + df;
    err = sum(abs(df));
  end
\end{verbatim}
Direct Vs. Iterative Solutions

**Direct Solution**
- **Benefits**
  - Most robust solution
  - Simple MATLAB code
  - Faster for smaller problems
- **Drawbacks**
  - Slow for large problems
  - Less accurate for large problems

**Iterative Solution**
- **Benefits**
  - Faster for larger problems
  - More accurate for larger problems
- **Drawbacks**
  - Matrix must be well conditioned

---

Conditioning of a Matrix

Suppose we solve $Ax = b$.

$$ x = A^{-1}b $$

Now suppose we make some small changes to $b$ and solve the problem again. This will give us some small changes in $x$.

$$ x + \Delta x = A^{-1} (b + \Delta b) \rightarrow \Delta x = A^{-1} \Delta b $$

If small changes in $b$ produce large changes in $x$, the matrix $A$ is said to be ill-conditioned. Think of $A$ as being “unstable” in this case.

The **condition number** $\kappa$ of a matrix is a measure of how ill-conditioned a matrix is. A low condition number means the matrix is “well conditioned” and can be solved in fewer iterations of an iterative solver.

$$ \kappa(A) = \frac{\text{max singular value}}{\text{min singular value}} = \frac{\sigma_{\text{max}}(A)}{\sigma_{\text{min}}(A)} \quad 1 \leq \kappa(A) < \infty $$
Conditioning of Our FDFD Wave Matrix $A$

As we have derived it, our wave matrix $A$ is not well conditioned.

There are two reasons:

1. We constructed $A$ using a uniaxial PML.
2. Large null space in $A$.

We must do two things to solve FDFD efficiently using an iterative solver.

1. Adopt a stretched-coordinate PML instead of uniaxial PML.
2. Add an additional term to $A$ to enforce continuity.

$$\begin{bmatrix} s_x \hat{s}_x s_z & 0 & 0 \\ 0 & s_x \hat{s}_x s_z & 0 \\ 0 & 0 & s_x \hat{s}_x s_z \end{bmatrix} \Rightarrow \nabla \times \frac{1}{\varepsilon} \nabla \times \vec{E} - \vec{E} = 0 \Rightarrow \nabla \times \mu^{-1} \nabla \times \vec{E} + \nabla \left[ \nabla \times (\vec{E}) \right] - \vec{E} = 0$$

More Information

- **GPU Accelerated FDFD**

- **Iterative Solvers**
Formulation with a Stretched-Coordinate PML

Maxwell’s Equations with SC-PML

Maxwell’s equations with a stretched-coordinate PML are

\[
\nabla_s \times \vec{E} = -j \omega \left[ \mu \right] \vec{H} \\
\nabla_s \times \vec{H} = j \omega \left[ \varepsilon \right] \vec{E}
\]

\[
\begin{bmatrix}
0 & -\frac{1}{s_z} \frac{\partial}{\partial z} & \frac{1}{s_s} \frac{\partial}{\partial y} \\
\frac{1}{s_z} \frac{\partial}{\partial z} & 0 & -\frac{1}{s_s} \frac{\partial}{\partial x} \\
-\frac{1}{s_y} \frac{\partial}{\partial y} & \frac{1}{s_s} \frac{\partial}{\partial x} & 0
\end{bmatrix}
\]

It is not possible to build the SC-PML into the derivative operators as we have derived them due to the staggered nature of the Yee grid. To do this, we would need separate \( s, y, \) and \( z \) derivatives for each of the six field components resulting in 24 separate derivative operators. Instead, the PML terms can be handled explicitly in the formulation and implementation.
Vector Expansion

Maxwell’s equations with a SC-PML expand to

Fully Anisotropic

\[
\frac{1}{s_y} \frac{\partial E_y}{\partial y} - \frac{1}{s_z} \frac{\partial E_z}{\partial z} = k_e (\varepsilon_x E_x + \varepsilon_y E_y + \varepsilon_z E_z)
\]

\[
\frac{1}{s_z} \frac{\partial E_z}{\partial z} - \frac{1}{s_x} \frac{\partial E_x}{\partial x} = k_e (\varepsilon_x E_x + \varepsilon_y E_y + \varepsilon_z E_z)
\]

\[
\frac{1}{s_x} \frac{\partial E_x}{\partial x} - \frac{1}{s_y} \frac{\partial E_y}{\partial y} = k_e (\varepsilon_x E_x + \varepsilon_y E_y + \varepsilon_z E_z)
\]

\[
\frac{1}{s_y} \frac{\partial H_y}{\partial y} - \frac{1}{s_z} \frac{\partial H_z}{\partial z} = k_s (\mu_x H_x + \mu_y H_y + \mu_z H_z)
\]

\[
\frac{1}{s_z} \frac{\partial H_z}{\partial z} - \frac{1}{s_x} \frac{\partial H_x}{\partial x} = k_s (\mu_x H_x + \mu_y H_y + \mu_z H_z)
\]

\[
\frac{1}{s_x} \frac{\partial H_x}{\partial x} - \frac{1}{s_y} \frac{\partial H_y}{\partial y} = k_s (\mu_x H_x + \mu_y H_y + \mu_z H_z)
\]

Diagonally Anisotropic

\[
\frac{1}{s_y} \frac{\partial E_y}{\partial y} - \frac{1}{s_z} \frac{\partial E_z}{\partial z} = k_e (\varepsilon_x E_x + \varepsilon_y E_y + \varepsilon_z E_z)
\]

\[
\frac{1}{s_z} \frac{\partial E_z}{\partial z} - \frac{1}{s_x} \frac{\partial E_x}{\partial x} = k_e (\varepsilon_x E_x + \varepsilon_y E_y + \varepsilon_z E_z)
\]

\[
\frac{1}{s_x} \frac{\partial E_x}{\partial x} - \frac{1}{s_y} \frac{\partial E_y}{\partial y} = k_e (\varepsilon_x E_x + \varepsilon_y E_y + \varepsilon_z E_z)
\]

\[
\frac{1}{s_y} \frac{\partial H_y}{\partial y} - \frac{1}{s_z} \frac{\partial H_z}{\partial z} = k_s (\mu_x H_x + \mu_y H_y + \mu_z H_z)
\]

\[
\frac{1}{s_z} \frac{\partial H_z}{\partial z} - \frac{1}{s_x} \frac{\partial H_x}{\partial x} = k_s (\mu_x H_x + \mu_y H_y + \mu_z H_z)
\]

\[
\frac{1}{s_x} \frac{\partial H_x}{\partial x} - \frac{1}{s_y} \frac{\partial H_y}{\partial y} = k_s (\mu_x H_x + \mu_y H_y + \mu_z H_z)
\]

Matrix Form of Maxwell’s Equations

Fully Anisotropic

\[
S^{(E)}_y D_x \hat{h}_y - S^{(E)}_z D_x \hat{h}_z = \varepsilon_{x} e_x + R_y^R R_y^R e_x + R_z^R R_z^R e_x
\]

\[
S^{(E)}_z D_x \hat{h}_z - S^{(E)}_x D_x \hat{h}_x = R_z^R R_y^R e_x + \varepsilon_{y} e_y + R_z^R R_z^R e_y
\]

\[
S^{(E)}_x D_x \hat{h}_x - S^{(E)}_y D_x \hat{h}_y = R_z^R R_y^R e_x + R_y^R R_z^R e_y + \varepsilon_{z} e_z
\]

\[
S^{(H)}_y D_x e_x - S^{(H)}_z D_x e_z = \mu_{x} \hat{h}_x + R_y^R R_y^R \mu_{x} \hat{h}_x + R_z^R R_z^R \mu_{x} \hat{h}_x
\]

\[
S^{(H)}_z D_x e_z - S^{(H)}_x D_x e_x = R_z^R R_y^R \mu_{x} \hat{h}_x + \mu_{y} \hat{h}_y + R_z^R R_z^R \mu_{y} \hat{h}_y
\]

\[
S^{(H)}_x D_x e_x - S^{(H)}_y D_x e_y = R_y^R R_y^R \mu_{x} \hat{h}_x + R_y^R R_z^R \mu_{y} \hat{h}_y + \mu_{z} \hat{h}_z
\]

\[
S^{(H)}_y e_y - S^{(H)}_z e_z = \mu_{x} \hat{h}_x + \mu_{y} \hat{h}_y + \mu_{z} \hat{h}_z
\]

\[
S^{(H)}_z e_z - S^{(H)}_x e_x = \mu_{x} \hat{h}_x + \mu_{y} \hat{h}_y + \mu_{z} \hat{h}_z
\]

Diagonally Anisotropic

\[
S^{(E)}_y D_x \hat{h}_y - S^{(E)}_z D_x \hat{h}_z = \varepsilon_{x} e_x
\]

\[
S^{(E)}_z D_x \hat{h}_z - S^{(E)}_x D_x \hat{h}_x = \varepsilon_{y} e_y
\]

\[
S^{(E)}_x D_x \hat{h}_x - S^{(E)}_y D_x \hat{h}_y = \varepsilon_{z} e_z
\]

\[
S^{(H)}_y D_x e_x - S^{(H)}_z D_x e_z = \mu_{x} \hat{h}_x
\]

\[
S^{(H)}_z D_x e_z - S^{(H)}_x D_x e_x = \mu_{y} \hat{h}_y
\]

\[
S^{(H)}_x D_x e_x - S^{(H)}_y D_x e_y = \mu_{z} \hat{h}_z
\]
Block Matrix Form

We can write our previous equations in block matrix form as

\[
\begin{pmatrix}
0 & -S^{(x)} D_x & S^{(x)} D_x \\
-S^{(x)} D_x & 0 & -S^{(x)} D_x \\
-S^{(x)} D_x & S^{(x)} D_x & 0
\end{pmatrix}
\begin{pmatrix}
h_x \\
h_y \\
h_z
\end{pmatrix}
= \begin{pmatrix}
\varepsilon_{xx} & R^x R^x \varepsilon_y & R^x R^x \varepsilon_z \\
R^y R^y \varepsilon_x & \varepsilon_{yy} & R^y R^y \varepsilon_z \\
R^z R^z \varepsilon_x & R^z R^z \varepsilon_y & \varepsilon_{zz}
\end{pmatrix}
\begin{pmatrix}
e_x \\
e_y \\
e_z
\end{pmatrix}
\]

For diagonally anisotropic problems we have

\[
\begin{pmatrix}
0 & -S^{(x)} D_x & S^{(x)} D_x \\
-S^{(x)} D_x & 0 & -S^{(x)} D_x \\
-S^{(x)} D_x & S^{(x)} D_x & 0
\end{pmatrix}
\begin{pmatrix}
h_x \\
h_y \\
h_z
\end{pmatrix}
= \begin{pmatrix}
\mu_{xx} & R^x R^x \mu_y & R^x R^x \mu_z \\
R^y R^y \mu_x & \mu_{yy} & R^y R^y \mu_z \\
R^z R^z \mu_x & R^z R^z \mu_y & \mu_{zz}
\end{pmatrix}
\begin{pmatrix}
e_x \\
e_y \\
e_z
\end{pmatrix}
\]

3D FDFD

Our wave matrix for 3D FDFD is

\[
\left(C^+ \left[ \mu^+ \right]^{-1} C - \left[ \varepsilon^+ \right] \right) \ddot{\varepsilon} = 0
\]

where

\[
\begin{pmatrix}
h_x \\
h_y \\
h_z
\end{pmatrix}
= \begin{pmatrix}
e_x \\
e_y \\
e_z
\end{pmatrix}
\]

\[
\begin{pmatrix}
\mu_{xx} & R^x R^x \mu_y & R^x R^x \mu_z \\
R^y R^y \mu_x & \mu_{yy} & R^y R^y \mu_z \\
R^z R^z \mu_x & R^z R^z \mu_y & \mu_{zz}
\end{pmatrix}
= \begin{pmatrix}
\varepsilon_{xx} & R^x R^x \varepsilon_y & R^x R^x \varepsilon_z \\
R^y R^y \varepsilon_x & \varepsilon_{yy} & R^y R^y \varepsilon_z \\
R^z R^z \varepsilon_x & R^z R^z \varepsilon_y & \varepsilon_{zz}
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & -S^{(x)} D_x & S^{(x)} D_x \\
-S^{(x)} D_x & 0 & -S^{(x)} D_x \\
-S^{(x)} D_x & S^{(x)} D_x & 0
\end{pmatrix}
\begin{pmatrix}
h_x \\
h_y \\
h_z
\end{pmatrix}
= \begin{pmatrix}
\mu_{xx} & R^x R^x \mu_y & R^x R^x \mu_z \\
R^y R^y \mu_x & \mu_{yy} & R^y R^y \mu_z \\
R^z R^z \mu_x & R^z R^z \mu_y & \mu_{zz}
\end{pmatrix}
\begin{pmatrix}
e_x \\
e_y \\
e_z
\end{pmatrix}
\]
For 2D problems that are uniform in the \( z \) direction and propagation is restricted to the \( xy \) plane,
\[
D_x^{(e)} = D_x^{(h)} = 0
\]
Dropping these terms as well as off-diagonal tensor elements, we get
\[
\begin{align*}
S_y^{(e)} D_y^x \tilde{h}_z &= \varepsilon_x e_x, \\
-S_y^{(e)} D_y^x \tilde{h}_z &= \varepsilon_y e_y, \\
S_x^{(e)} D_x^y \tilde{h}_z - S_y^{(e)} D_y^x \tilde{h}_z &= \varepsilon_z e_z,
\end{align*}
\]
\[
\begin{align*}
S_y^{(h)} D_y^x e_x &= \mu_x \tilde{h}_z, \\
-S_y^{(h)} D_y^x e_x &= \mu_y \tilde{h}_z, \\
S_x^{(h)} D_x^y e_y - S_y^{(h)} D_y^x e_y &= \mu_z \tilde{h}_z
\end{align*}
\]

We can derive wave matrices for each mode as follows.

**E-Mode**
\[
\begin{align*}
\left( S_y^{(e)} D_y^x \mu_x S_y^{(h)} D_y^x + S_y^{(e)} D_y^x \mu_y S_y^{(h)} D_y^x + \varepsilon_m \right) e_x &= 0 \\
\tilde{h}_x &= \mu_x^{-1} S_y^{(h)} D_y^x e_x, \\
\tilde{h}_y &= -\mu_y^{-1} S_y^{(h)} D_y^x e_y
\end{align*}
\]

**H-Mode**
\[
\begin{align*}
\left( S_y^{(h)} D_y^x \mu_x S_y^{(e)} D_y^x + S_y^{(h)} D_y^x \mu_y S_y^{(e)} D_y^x + \varepsilon_m \right) e_x &= 0 \\
e_x &= \varepsilon_m^{-1} S_y^{(e)} D_y^x \tilde{h}_z, \\
e_y &= -\varepsilon_m^{-1} S_y^{(e)} D_y^x \tilde{h}_z
\end{align*}
\]

We have arrived at our standard \( Ax = 0 \) form. The remainder of the algorithm is identical for both UPML and SC-PML.
Maxwell’s Equations for Doubly-Anisotropic Media

Maxwell’s equations in doubly-anisotropic media with a UPML can be written as

\[ \nabla \times \vec{E} = k_0 [\mu_r][S] \vec{H} \quad \nabla \times \vec{H} = k_0 [\varepsilon_r][S] \vec{E} \]

After absorbing the PML into the constitutive parameters, these equations expand to

\[ \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial z} = \mu_r' H_z + \mu_r' H_y + \mu_r' H_z \]
\[ \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial x} = \mu_r' H_x + \mu_r' H_y + \mu_r' H_z \]
\[ \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial y} = \mu_r' H_x + \mu_r' H_y + \mu_r' H_z \]

\[ \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial z} = \varepsilon_r' E_y + \varepsilon_r' E_y + \varepsilon_r' E_z \]
\[ \frac{\partial H_y}{\partial z} - \frac{\partial H_z}{\partial x} = \varepsilon_r' E_x + \varepsilon_r' E_x + \varepsilon_r' E_z \]
\[ \frac{\partial H_z}{\partial x} - \frac{\partial H_x}{\partial y} = \varepsilon_r' E_x + \varepsilon_r' E_x + \varepsilon_r' E_z \]

\[ \tilde{x} = k_0 x \quad \tilde{y} = k_0 y \quad \tilde{z} = k_0 z \]
The Main Problem with Anisotropy

All terms in a finite-difference equation must exist at the same point in time and space.

\[
\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial z} = \mu_x' \bar{H}_x + \mu_y' \bar{H}_y + \mu_z' \bar{H}_z
\]

These terms do not naturally fall onto the same points as \(H_x\).

This is "the equation for \(H_x\)."
All terms must exist where \(H_x\) exists.

The terms \(\mu_{xy} H_j\) and \(\mu_{xz} H_j\) must be interpolated at the same points as \(H_x\).

\[
\begin{align*}
\mu_x' \bar{H}_x & \approx \frac{\mu_x \tilde{H}_x^{i-1,j,k} + \mu_x \tilde{H}_x^{i+1,j,k} + \mu_x \tilde{H}_x^{i,j-1,k} + \mu_x \tilde{H}_x^{i,j+1,k}}{4} \\
\mu_y' \bar{H}_y & \approx \frac{\mu_y \tilde{H}_y^{i,j,k-1} + \mu_y \tilde{H}_y^{i,j,k+1} + \mu_y \tilde{H}_y^{i-1,j,k} + \mu_y \tilde{H}_y^{i+1,j,k}}{4}
\end{align*}
\]
Finite-Difference Approximation of Maxwell’s Equations (1 of 6)

\[ \frac{\partial E_y}{\partial y} - \frac{\partial E_z}{\partial z} = \mu'_n \vec{H}_y + \mu'_n \vec{H}_z + \mu'_n \vec{H}_y \]

\[ \frac{E_{y}^{j,k} - E_{z}^{j,k}}{\Delta y} = \mu'_n \vec{H}_y^{j,k} + \mu'_n \vec{H}_z^{j,k} + \mu'_n \vec{H}_y^{j,k+1} + \mu'_n \vec{H}_z^{j,k+1} \]

\[ \frac{E_{y}^{j,k+1} - E_{z}^{j,k}}{\Delta y} = \mu'_n \vec{H}_y^{j,k} + \mu'_n \vec{H}_z^{j,k} + \mu'_n \vec{H}_y^{j,k+1} + \mu'_n \vec{H}_z^{j,k+1} \]

Finite-Difference Approximation of Maxwell’s Equations (2 of 6)

\[ \frac{\partial E_z}{\partial z} = \mu'_n \vec{H}_x + \mu'_n \vec{H}_x + \mu'_n \vec{H}_x \]

\[ \frac{E_{x}^{j,k+1} - E_{x}^{j,k}}{\Delta z} = \mu'_n \vec{H}_x^{j,k} + \mu'_n \vec{H}_x^{j,k} + \mu'_n \vec{H}_x^{j,k+1} + \mu'_n \vec{H}_x^{j,k+1} \]

\[ \frac{E_{x}^{j,k} - E_{x}^{j,k-1}}{\Delta z} = \mu'_n \vec{H}_x^{j,k} + \mu'_n \vec{H}_x^{j,k} + \mu'_n \vec{H}_x^{j,k+1} + \mu'_n \vec{H}_x^{j,k+1} \]
Finite-Difference Approximation of Maxwell’s Equations (3 of 6)

\[ \frac{\partial E_x}{\partial x} - \frac{\partial E_y}{\partial y} = \mu'_x \hat{H}_y + \mu'_y \hat{H}_x + \mu'_z \hat{H}_z \]

\[ E_j^{x,i,j,k} - E_j^{x,i+1,j,k} - E_j^{x,i,j,k+1} - E_j^{x,i,j,k-1} = \frac{\mu'_x \hat{H}_y^{i,j,k} + \mu'_y \hat{H}_x^{i,j,k} + \mu'_z \hat{H}_z^{i,j,k} + \mu'_z \hat{H}_z^{i,j,k-1}}{4} \]

\[ + \frac{\mu'_x \hat{H}_y^{i,j,k} + \mu'_y \hat{H}_x^{i,j,k} + \mu'_z \hat{H}_z^{i,j,k-1}}{4} \]

\[ + \mu'_x \hat{H}_y^{i,j,k} + \mu'_y \hat{H}_x^{i,j,k} + \mu'_z \hat{H}_z^{i,j,k-1} \]

Finite-Difference Approximation of Maxwell’s Equations (4 of 6)

\[ \frac{\partial H_y}{\partial y} - \frac{\partial H_z}{\partial z} = \varepsilon'_x E_x + \varepsilon'_y E_y + \varepsilon'_z E_z \]

\[ H_j^{x,i,j,k} - H_j^{x,i,j,k+1} - H_j^{x,i,j,k+1} - H_j^{x,i,j,k-1} = \varepsilon'_x E_x^{i,j,k} \]

\[ + \frac{\varepsilon'_x E_x^{i,j,k} + \varepsilon'_y E_y^{i,j,k} + \varepsilon'_z E_z^{i,j,k}}{4} \]

\[ + \frac{\varepsilon'_x E_x^{i,j,k} + \varepsilon'_y E_y^{i,j,k} + \varepsilon'_z E_z^{i,j,k}}{4} \]

\[ + \varepsilon'_x E_x^{i,j,k} + \varepsilon'_y E_y^{i,j,k} + \varepsilon'_z E_z^{i,j,k} \]
Finite-Difference Approximation of Maxwell’s Equations (5 of 6)

\[
\frac{\partial \tilde{H}_z}{\partial z} - \frac{\partial \tilde{H}_y}{\partial x} = \varepsilon'_\varphi E_z + \varepsilon'_\varphi E_y + \varepsilon'_\varphi E_z
\]

\[
\frac{\tilde{H}_z^{j,k} - \tilde{H}_z^{j,k-1}}{\Delta z} = \frac{\tilde{H}_y^{j,k} - \tilde{H}_y^{j,k-1}}{\Delta \tilde{x}} + \frac{\varepsilon'_\varphi E_z^{j,k} + \varepsilon'_\varphi E_y^{j,k} + \varepsilon'_\varphi E_z^{j,k-1} + \varepsilon'_\varphi E_y^{j,k-1}}{4}
\]
Assigning Anisotropic Materials to Yee Grid

Parsing Onto 1x Grid for Fully Anisotropic Case

\[
\begin{bmatrix}
\varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\
\varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\
\varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz}
\end{bmatrix} = \begin{bmatrix}
\varepsilon_{x} \\
\varepsilon_{y} \\
\varepsilon_{z}
\end{bmatrix} = \begin{bmatrix}
\mu_{xx} & \mu_{xy} & \mu_{xz} \\
\mu_{yx} & \mu_{yy} & \mu_{yz} \\
\mu_{zx} & \mu_{zy} & \mu_{zz}
\end{bmatrix} = \begin{bmatrix}
\mu_{x} \\
\mu_{y} \\
\mu_{z}
\end{bmatrix}
\]

\[
ER_{xx} = ER_{2xx}(2:2:Nx2,1:2:Ny2,1:2:Nz2); \\
ER_{xy} = ER_{2xy}(1:2:Nx2,2:2:Ny2,1:2:Nz2); \\
ER_{xz} = ER_{2xz}(1:2:Nx2,1:2:Ny2,2:2:Nz2); \\
ER_{yx} = ER_{2yx}(2:2:Nx2,1:2:Ny2,1:2:Nz2); \\
ER_{yy} = ER_{2yy}(1:2:Nx2,2:2:Ny2,1:2:Nz2); \\
ER_{yz} = ER_{2yz}(1:2:Nx2,2:2:Ny2,1:2:Nz2); \\
\]

\[
UR_{xx} = UR_{2xx}(1:2:Nx2,2:2:Ny2,2:2:Nz2); \\
UR_{xy} = UR_{2xy}(2:2:Nx2,1:2:Ny2,2:2:Nz2); \\
UR_{xz} = UR_{2xz}(2:2:Nx2,2:2:Ny2,1:2:Nz2); \\
UR_{yx} = UR_{2yx}(1:2:Nx2,2:2:Ny2,2:2:Nz2); \\
UR_{yy} = UR_{2yy}(2:2:Nx2,1:2:Ny2,2:2:Nz2); \\
UR_{yz} = UR_{2yz}(2:2:Nx2,2:2:Ny2,1:2:Nz2); \\
\]
Implementing the Interpolations

\[
\frac{E^{j,k}_{i+\Delta k} - E^{j,k}_{i}}{\Delta y} = E^{j,k+\Delta k}_{i} - E^{j,k}_{i} = \mu_x^+ \tilde{H}^{i,j,k}_{i+1,j,k} + \mu_x^- \tilde{H}^{i,j,k}_{i-1,j,k} + \mu_y \tilde{H}^{i,j,k}_{i,j+1,k} + \frac{\mu_y \tilde{H}^{i,j,k}_{i,j-1,k}}{4} + \frac{\mu_x^+ \tilde{H}^{i,j,k}_{i+1,j,k+1} + \mu_x^- \tilde{H}^{i,j,k}_{i-1,j,k-1} + \mu_y \tilde{H}^{i,j,k}_{i,j+1,k+1} + \mu_y \tilde{H}^{i,j,k}_{i,j-1,k-1}}{4}
\]

\[
D_y^e e_z - D_z^e e_y = \mu'_{xx} \hat{h}_x + R_x R_y^+ \mu'_{xy} \hat{h}_y + R_x R_z^+ \mu'_{xz} \hat{h}_z
\]

Direction the interpolation reaches (- to the left, + to the right)
Axis across which the interpolation is performed.

Interpolation Matrices (1 of 2)

The derivative matrices were constructed from a simple finite-difference approximation of the form

\[
\frac{\partial f_{1,5}}{\partial x} \approx \frac{f_2 - f_1}{\Delta x}
\]

The interpolation matrices are constructed exactly the same way, but uses the following equation for interpolation:

\[
f_{1,5} \approx \frac{f_2 + f_1}{2}
\]

The interpolation matrices have the following interpretations:

- \( R_i^+ \) interpolates along the \( i \)-axis using a value from the next cell along \( i \)
- \( R_i^- \) interpolates along the \( i \)-axis using a value from the previous cell along \( i \)
Interpolation Matrices (2 of 2)

Think of forming the interpolation this way

\[
\begin{align*}
R_x^+ &\approx \frac{\Delta x'}{2} \cdot D'_{xy}^+ \\
R_y^+ &\approx \frac{\Delta y'}{2} \cdot D'_{yz}^+ \\
R_z^+ &\approx \frac{\Delta z'}{2} \cdot D'_{xz}^+
\end{align*}
\]

Strictly speaking, this will not work because the \(\|\|\) operation breaks some boundary conditions.

This calculation approach does work for Dirichlet boundary conditions and for periodic boundary conditions that do not include phase.

The interpolation matrices are related through:

\[
\begin{align*}
R_x^+ &= (R_x^-)^H \\
R_y^+ &= (R_y^-)^H \\
R_z^+ &= (R_z^-)^H
\end{align*}
\]

Matrix Form of Maxwell’s Equations

\[
\begin{align*}
\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} &= \mu_x \frac{\partial H_y}{\partial x} + \mu_y \frac{\partial H_x}{\partial y} + \mu_z \frac{\partial H_z}{\partial z} \\
\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} &= \mu_x \frac{\partial H_z}{\partial x} + \mu_y \frac{\partial H_x}{\partial z} + \mu_z \frac{\partial H_z}{\partial z} \\
\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} &= \mu_x \frac{\partial H_z}{\partial y} + \mu_y \frac{\partial H_y}{\partial z} + \mu_z \frac{\partial H_z}{\partial z} \\
\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} &= \mu_x \frac{\partial E_x}{\partial x} + \mu_y \frac{\partial E_y}{\partial y} + \mu_z \frac{\partial E_z}{\partial z} \\
\frac{\partial H_z}{\partial x} - \frac{\partial H_x}{\partial z} &= \mu_x \frac{\partial E_x}{\partial x} + \mu_y \frac{\partial E_y}{\partial y} + \mu_z \frac{\partial E_z}{\partial z} \\
\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} &= \mu_x \frac{\partial E_x}{\partial x} + \mu_y \frac{\partial E_y}{\partial y} + \mu_z \frac{\partial E_z}{\partial z}
\end{align*}
\]
Absorbing Interpolation Matrices into Constitutive Parameters

\[
\begin{bmatrix}
\mu_x^e & \mu_y^e & \mu_z^e \\
\mu_x^h & \mu_y^h & \mu_z^h \\
\mu_x^e & \mu_y^e & \mu_z^e \\
\end{bmatrix} = \begin{bmatrix}
\mu_x & R_x^e \mu_y & R_x^e \mu_z \\
R_y^h \mu_x & \mu_y & R_y^h \mu_z \\
R_z^h \mu_x & R_z^h \mu_y & \mu_z \\
\end{bmatrix}
\]

*Note:* the double-prime superscript indicates that both the UPML and interpolations matrices are now absorbed into the constitutive parameters.

\[
\begin{bmatrix}
\varepsilon_x^e & \varepsilon_y^e & \varepsilon_z^e \\
\varepsilon_x^h & \varepsilon_y^h & \varepsilon_z^h \\
\varepsilon_x^e & \varepsilon_y^e & \varepsilon_z^e \\
\end{bmatrix} = \begin{bmatrix}
\varepsilon_x' & R_x^e \varepsilon_y' & R_x^e \varepsilon_z' \\
R_y^h \varepsilon_x' & \varepsilon_y' & R_y^h \varepsilon_z' \\
R_z^h \varepsilon_x' & R_z^h \varepsilon_y' & \varepsilon_z' \\
\end{bmatrix}
\]

Compact Form of Matrix Equations

\[
\nabla \times \vec{E} = k_0 [\mu_r] [S] \vec{H}
\]

\[
C^e \vec{e} = [\mu_r^e] \vec{h}
\]

\[
\vec{e} = \begin{bmatrix}
e_x \\ e_y \\ e_z \\
\end{bmatrix}
\]

\[
\vec{h} = \begin{bmatrix}
h_x \\ h_y \\ h_z \\
\end{bmatrix}
\]

\[
C^e = \begin{bmatrix}
0 & -D_y^e & D_z^e \\
D_x^e & 0 & -D_z^e \\
-D_y^e & D_z^e & 0
\end{bmatrix}
\]

\[
\nabla \times \vec{H} = k_0 [\varepsilon_r] [S] \vec{E}
\]

\[
C^h \vec{h} = [\varepsilon_r^h] \vec{e}
\]

\[
\vec{h} = \begin{bmatrix}
h_x \\ h_y \\ h_z \\
\end{bmatrix}
\]

\[
C^h = \begin{bmatrix}
0 & -D_y^h & D_z^h \\
D_x^h & 0 & -D_z^h \\
-D_y^h & D_z^h & 0
\end{bmatrix}
\]
Matrix Wave Equations

\[ C^e \tilde{e} = \left[ \mu^e \right] \tilde{h} \quad \text{Eq. (1)} \]

\[ C^h \tilde{h} = \left[ \varepsilon^h \right] \tilde{e} \quad \text{Eq. (2)} \]

We derive a matrix wave equation in terms of the electric field by first solving Eq. (1) for the magnetic field.

\[ \tilde{h} = \left[ \mu^e \right]^{-1} C^e \tilde{e} \]

We not substitute this expression into Eq. (2).

\[ C^h \left( \left[ \mu^e \right]^{-1} C^e \tilde{e} \right) = \left[ \varepsilon^h \right] \tilde{e} \]

This expression is rearranged into the form of \( Ax = 0 \).

\[ \left( C^h \left[ \mu^e \right]^{-1} C^e - \left[ \varepsilon^h \right] \right) \tilde{e} = 0 \]

We could have derived a wave equation for the magnetic field.

\[ \left( C^h \left[ \mu^e \right]^{-1} C^e - \left[ \varepsilon^h \right] \right) \tilde{h} = 0 \]

Speed Trick for the Wave Matrix

\[ A = C^h \left[ \mu^e \right]^{-1} C^e - \left[ \varepsilon^h \right] \]

This matrix division is very slow to calculate when magnetic anisotropy is being simulated.

\[ [\psi] = \begin{bmatrix} \psi_{xx} & \psi_{xy} & \psi_{xz} \\ \psi_{yx} & \psi_{yy} & \psi_{yz} \\ \psi_{zx} & \psi_{zy} & \psi_{zz} \end{bmatrix} \]

Let \( [\psi] = \left[ \mu^e \right]^{-1} \) so that \( A = C^h [\psi] C^e - [\varepsilon^h] \).

To calculate \( [\psi] \) efficiently, calculate the elements of it from the permeability arrays. For brevity, the array indices \( i, j, \) and \( k \) were dropped.

\[ [\psi(i, j, k)] = \begin{bmatrix} \mu_{yy} \mu_{zz} - \mu_{yz} \mu_{zy} & \mu_{xz} \mu_{yy} - \mu_{xy} \mu_{yz} & \mu_{yx} \mu_{zz} - \mu_{zx} \mu_{zy} \\ \mu_{yz} \mu_{zy} - \mu_{yy} \mu_{zz} & \mu_{xx} \mu_{yy} - \mu_{xy} \mu_{xy} & \mu_{xy} \mu_{zz} - \mu_{yx} \mu_{zy} \\ \mu_{zy} \mu_{zy} - \mu_{zz} \mu_{yy} & \mu_{xz} \mu_{xz} - \mu_{xx} \mu_{xy} & \mu_{xx} \mu_{zz} - \mu_{yx} \mu_{zy} \end{bmatrix} \]

The individual elements of \( [\psi(i, j, k)] \) are diagonalized to get \( [\Psi] \).
Calculating the Source Vector

A scattered-field masking matrix $\mathbf{Q}$ must be constructed for each component of the source and then assemble them into a single $\mathbf{Q}$ matrix.

$$
\mathbf{e}_{\text{src}} = \begin{bmatrix}
  e_{x,\text{src}} \\
  e_{y,\text{src}} \\
  e_{z,\text{src}}
\end{bmatrix} \rightarrow Q_x \quad Q_y \rightarrow Q_z
$$

$$
\mathbf{Q} = \begin{bmatrix}
  Q_x & 0 & 0 \\
  0 & Q_y & 0 \\
  0 & 0 & Q_z
\end{bmatrix}
$$

The source vector $\mathbf{b}$ is then calculated as usual.

$$
\mathbf{b} = (\mathbf{QA} - \mathbf{AQ}) \mathbf{e}_{\text{src}}
$$