Assignment VII - Modal Analysis

- Computer Projects:

1. Write a code to complete the shielded microstrip problem to obtain the first TM_{01} mode. Assume $h = w/2 = 2$ mm, $\epsilon_r = 4$, $a = 40h$ and $b = 2.5h$. M_h should be even and no less that 10.
2. Start with $f = 2$ GHz and demonstrate the following two solution observations:
   (a) Within the substrate space under the trace, E_z will indeed articulate TM_{01} behavior.
   (b) Throughout the cross-section, however, substrate region will have limited effect on where the peak E_z location(s) are, whereas the trace will cause the field distribution to mimic a TM_{21} mode as far as the $a \times b$ metal enclosure is concerned.
3. Repeat the analysis at 40 GHz and demonstrate the increased role of the substrate on field distribution.
4. Calculate, at 40 GHz, the energy percentage contained within the substrate, compared to the total energy in the microstrip's cross-section.

Theory

Gauss-Seidel Iterative Method with Over-Relaxation

The Gauss-Seidel (G-S) method is a classical iterative technique for solving systems of linear equations, particularly those arising from finite-difference (FD) approximations of partial differential equations (PDEs). It is especially effective for sparse systems and large-scale numerical simulations such as electromagnetic field problems in waveguides or microstrip lines.

1. Finite Difference Formulation

For a second-order differential equation such as:

$$f''(x) - 2f(x) = 0, \quad f(0) = 1, \quad f'(1) = 0$$

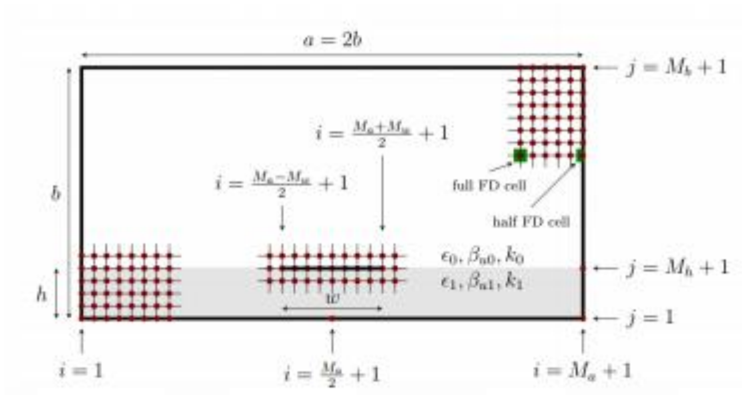
we discretize the domain into a grid and apply finite differences. The central difference approximation gives:

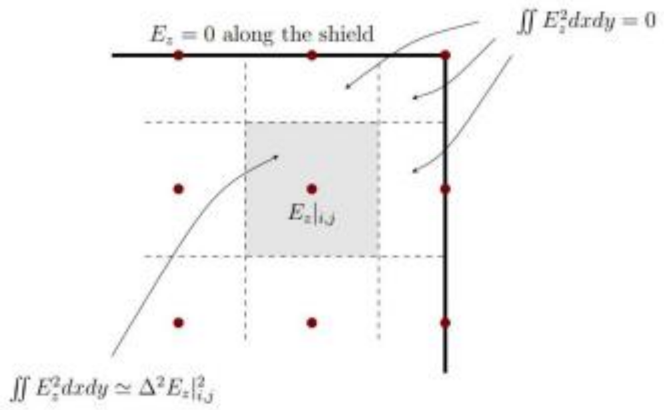
$$\frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta^2} - 2f_i = 0$$

which rearranges to the Gauss-Seidel update formula:

$$f_i^{(n)} = \frac{f_{i+1}^{(n-1)} + f_{i-1}^{(n)}}{2(1 + \Delta^2)}$$

Here, $f_i^{(n)}$ is the updated value at iteration n, using the most recent values of neighboring points.





- Let us consider the eigenvalue problem of the open microstrip line in Fig. 7.3, where $w = 2h = 4$ mm.
- We are interested in isolating the first TM mode underneath the trace (TM_{01}) which signifies a half cycle change vertically and a variation across the trace's width which is governed by its expected U-shaped current density distribution.
- The corresponding wave equation is

$$(\nabla^2 + \beta_u^2)E_z(x, y)\ e^{-\gamma z} = 0 \quad \rightarrow \quad \overbrace{\left(\underbrace{\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}}_{\nabla_T^2} + \underbrace{\beta_u^2 + \gamma_z^2}_{= k^2}\right) E_z}^{\mathcal{L}E_z} = 0$$

  where $\beta_u^2 = \omega^2 \mu_0 \epsilon_0$ in air and $\omega^2 \mu_0 \epsilon_0 \epsilon_r$ within the substrate, with the corresponding boundary condition $E_z = 0$ along the two conductors.
- To truncate the discrete model, we will enclose the open microstrip structure with an $a \times b$ PEC box where $a = 10w$ and $b = a/2$ and the ground plane forming the base of this box.
- We will then divide the cross-section into square FD cells such that we have M_h cells between trace and ground plane. The other cell counts of import are $M_w = 2M_h$, $M_a = 20M_h$ and $M_b = 10M_h$, as defined in Fig. 7.4. (Figure is not fully to scale, $a = 5w$ was used.)
- Thus divided, we can approximate the wave equation with the following FD equation for every node not lying on a PEC occupied space within the shielded enclosure

$$\frac{\overbrace{E_z|_{i+1,j} + E_z|_{i-1,j} + E_z|_{i,j+1} + E_z|_{i,j-1}}^{S_{i,j}} - 4E_z|_{i,j}}{\Delta^2} + k^2 E_z|_{i,j} = 0$$

- Re-arranging as an update equation renders our first of two leap-frog equations

$$
\begin{aligned}
E_z|_{i,j} &= \frac{S_{i,j}}{\alpha_j} \\
\alpha_j &= 4 - (k_j\Delta)^2 \\
k_j^2 &= \begin{cases} k_1^2 & ,\ 2 \le j \le M_h \\ (k_1^1 + k_0^2)/2 & ,\ j = M_h + 1 \\ k_0^2 & ,\ M_h + 2 \le j \le M_b \end{cases}
\end{aligned} \tag{7.1}
$$

- The exception is E_z must be zero at the trace;

$$
E_z|_{i,M_h+1} = 0 \quad \text{when} \quad \frac{M_a - M_w}{2} + 1 \le i \le \frac{M_a + M_w}{2} + 1
$$

- Furthermore, k_0 and k_1 are related to the unchanging γ_z throughout the *waveguide's* cross-section;

$$
\gamma_z^2 = k^2 - \beta_u^2 \quad \rightarrow \quad k_1^2 - \beta_{u1}^2 = k_0^2 - \beta_{u0}^2,
$$

leading to

$$
\begin{aligned}
k_1^2 &= k_0^2 + \beta_{u1}^2 - \beta_{u0}^2 = k_0^2 + \beta_{u0}^2(\epsilon_r - 1) \\
\frac{k_0^2 + k_1^2}{2} &= k_0^2 + \beta_{u0}^2\frac{\epsilon_r - 1}{2}
\end{aligned}
$$

- The second leap-frog equation is obtained from the following chosen inner product

$$
\begin{aligned}
\langle E_z, \mathcal{L}E_z\rangle &= \int_{\Omega_x}\int_{\Omega_y} E_z(\nabla_T^2 + k^2)E_z dxdy = 0 \\
\underbrace{\int_{\Omega_x}\int_{\Omega_y} E_z\nabla_T^2 E_z\ dxdy}_{=I_1} &+ \underbrace{\int_{\Omega_x}\int_{\Omega_y} k^2 E_z^2\ dxdy}_{=I_2} = 0
\end{aligned}
$$

$$
I_1 \simeq \sum_{i=2}^{M_a}\sum_{j=2}^{M_b} E_z|_{i,j}[S_{i,j} - 4E_z|_{i,j}] \tag{7.2}
$$

$$
I_2 \simeq \Delta^2 \sum_{i=2}^{M_a}\left\{\sum_{j=2}^{M_h} k_1^2 E_z^2|_{i,j} + \sum_{j=M_h+2}^{M_b} k_0^2 E_z^2|_{i,j} + \frac{k_0^2 + k_1^2}{2}E_z^2|_{i,M_h+1}\right\} \tag{7.3}
$$

$$
\simeq k_0^2\ \underbrace{\Delta^2\sum_{i=2}^{M_a}\sum_{j=2}^{M_b} E_z^2|_{i,j}}_{=I_{2a}} + \underbrace{\beta_{u0}^2(\epsilon_r - 1)\Delta^2\sum_{i=2}^{M_a}\left[\frac{1}{2}E_z^2|_{i,M_h+1} + \sum_{j=2}^{M_h} E_z^2|_{i,j}\right]}_{=I_{2b}} \tag{7.4}
$$

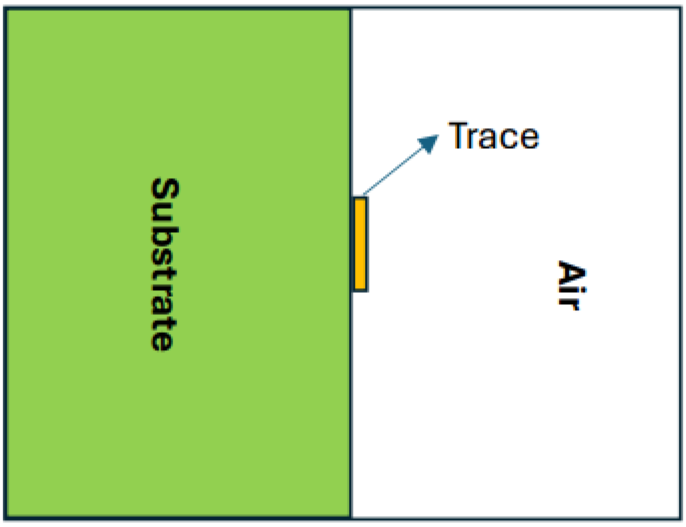
- The following is resulting second iterative update equation

$$
k_0^2 = -\frac{I_1 + I_{2b}}{I_{2a}} \tag{7.5}
$$

- To capture the TM_{01} mode under the trace we can start leap-frogging equations (7.1–7.5) with the following initial guess

$$
E_z|_{i,j} = \sin\frac{\pi(j-1)}{M_h}\ ,\quad j = 2, 3, \ldots, M_h \tag{7.6}
$$

with zero values everywhere else.



---

2. Gauss-Seidel Algorithm

The algorithm proceeds by sweeping through all interior points of the domain and updating each value based on the latest available neighboring values. The boundary conditions are applied explicitly and remain fixed throughout the iterations.

3. Over-Relaxation

To accelerate convergence, a **relaxation factor** R is introduced:

$$f_i^{(n)} = R \cdot \tilde{f}_i^{(n)} + (1 - R) \cdot f_i^{(n-1)}$$

where $\tilde{f}_i^{(n)}$ is the updated value from the standard Gauss-Seidel step. This technique is known as **Successive Over-Relaxation (SOR)**. For over-relaxation, $R > 1$; the optimal value of R is problem-specific but typically lies in the range $1.2 \leq R \leq 1.5$.

4. Convergence Criteria

The iteration continues until the solution converges, typically defined by:

$$\max_i |f_i^{(n)} - f_i^{(n-1)}| < \epsilon$$

for some small threshold ϵ (e.g., 10^{-6}).

5. Application in Waveguide/Microstrip Problems

In solving Helmholtz or Poisson equations for waveguide cross-sections or electrostatic problems, the Gauss-Seidel method iteratively updates the scalar field (e.g., E_z or potential ϕ) across the 2D mesh. Over-relaxation significantly improves convergence speed, especially for high-resolution meshes.

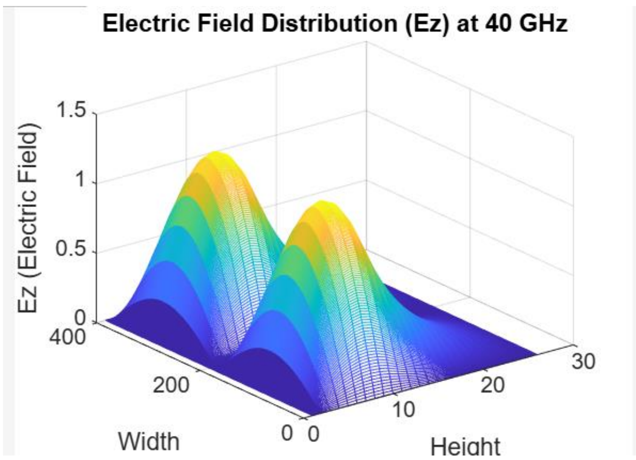
CODE:

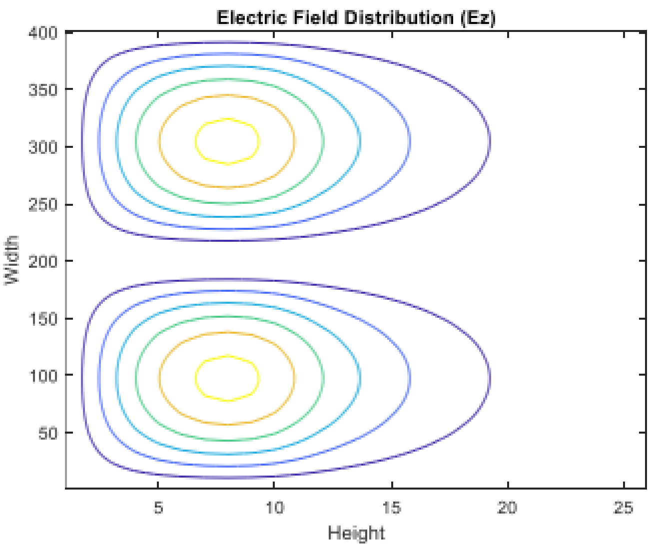
```
% format and initialization
format long
clearvars
close all

% Constants
frequencies = [2e9, 40e9]; % Frequencies in Hz
er = 4;
u0 = 4 * pi * 1e-7;
e0 = 8.8542e-12;

% Geometry
w = 4e-3;
h = w / 2;
a = 40 * h; % horizontal length
b = 2.5 * h; % vertical length

% Mesh parameters
Mh = 10; % cells in height
Mw = 2 * Mh; % width
Ma = 40 * Mh; % horizontal resolution
Mb = 2.5 * Mh; % vertical resolution

del_a = a / Ma;
del_b = b / Mb;

% Loop over frequencies
for f = frequencies
    omega = 2 * pi * f;
    Bu0 = omega * sqrt(u0 * e0);

    % Initialization
    s = zeros(Ma + 1, Mb + 1);
    Ez = zeros(Ma + 1, Mb + 1);
    Ez_previous = Ez;
    alpha = zeros(Mb + 1, 1);
    k_sq = zeros(Mb + 1, 1);

    % Initial guess for Ez
    for i = 2:Ma
        for j = 2:Mh
            Ez(i, j) = sin((pi * (j - 1)) / Mh);
        end
    end

    % Iterative FDFD solver
    iteration = 0;
```

```
convergence = 0;
tol = 1e-4;

while ~convergence
    Ez_previous = Ez;
    iteration = iteration + 1;

    % Calculate s
    for i = 2:Ma
        for j = 2:Mb
            s(i, j) = Ez(i, j + 1) + Ez(i, j - 1) + Ez(i + 1, j) + Ez(i - 1, j);
        end
    end

    % I1 term
    I1 = 0;
    for i = 2:Ma
        for j = 2:Mb
            I1 = I1 + (Ez(i, j) * s(i, j) - 4 * Ez(i, j)^2);
        end
    end

    % I2a term
    I2a = sum(sum(Ez(2:Ma, 2:Mb).^2)) * del_a * del_b;

    % I2b term (substrate region only)
    I2b = sum(sum(Ez(2:Ma, 2:Mh).^2));
    for i = 2:Ma
        I2b = I2b + 0.5 * Ez(i, Mh + 1)^2;
    end
    I2b = I2b * (Bu0^2 * (er - 1) * del_a * del_b);

    % Compute k^2
    k0_sq = - (I1 + I2b) / I2a;
    k1_sq = k0_sq + Bu0^2 * (er - 1);

    for j = 2:Mb
        if j <= Mh
            k_sq(j) = k1_sq;
        elseif j == Mh + 1
            k_sq(j) = (k1_sq + k0_sq) / 2;
        else
            k_sq(j) = k0_sq;
        end
        alpha(j) = 4 - k_sq(j) * del_a * del_b;
    end

    % Over-relaxation update
    R = 1.3;
    for relaxation = 1:5
        for i = 2:Ma
            for j = 2:Mb
                Ez(i, j) = R * (Ez(i, j + 1) + Ez(i, j - 1) + Ez(i + 1, j) + Ez(i - 1, j)) / alpha(j) ...
                         + (1 - R) * Ez(i, j);
            end
            % Enforce trace (center line) Ez = 0
            if i >= ((Ma - Mw)/2 + 1) && i <= ((Ma + Mw)/2 + 1)
                Ez(i, Mh + 1) = 0;
            end
        end
    end
```

```
            % Check convergence after 100 iterations
            if iteration >= 100
                if max(abs(Ez(:) - Ez_previous(:))) <= tol
                    convergence = 1;
                end
            end
        end

        % Energy computation
        energy_substrate = sum(sum(Ez(2:Ma, 2:Mh).^2)) * del_a * del_b;
        energy_substrate = 0.5 * e0 * er * energy_substrate;

        Air_energy = sum(sum(Ez(2:Ma, Mh+1:Mb).^2)) * del_a * del_b;
        Air_energy = 0.5 * e0 * Air_energy;

        total_energy = energy_substrate + Air_energy;
        energy_percent = (energy_substrate / total_energy) * 100;

        % Display result
        fprintf('\nEnergy within the substrate when frequency = %.1f GHz is %.2f%%\n', f / 1e9, energy_percent);

        % Plot for each frequency
        figure;
        mesh(Ez)
        xlabel('Height')
        ylabel('Width')
        zlabel('Ez (Electric Field)')
        title(['Electric Field Distribution (Ez) at ', num2str(f/1e9), ' GHz'])
    end
```

```
Energy within the substrate when frequency = 2.0 GHz is 59.90%

Energy within the substrate when frequency = 40.0 GHz is 92.46%
>>
```





When Frequency = 2GHz, energy contained within the substrate = 59.9%