

B2M17CADA - Project I

2ND JUNE, 2023

MARTIN ŠIMÁK

Task *Analysis of a stripline using finite differences*

Determine the properties of a shielded coplanar stripline depicted in figure 1 designed on a suspended dielectric substrate ($h = 1$ mm, $\epsilon_r = 3.05$) using finite differences and the method of successive over-relaxation. Find the strip's width w and gap g between the strips so that the characteristic impedance is $Z_0 = 50 \Omega$. The distance of the shielding walls from the strips should be larger than $20h$ to ensure that the line's impedance is not affected by finite dimensions.

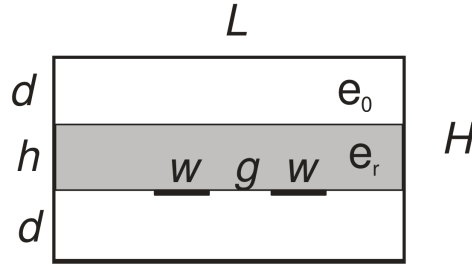


Figure 1: Task geometry

- (a) Illustrate equipotential countours and the distribution of electric field.
- (b) Evaluate the capacity per unit length and the characteristic impedance Z_0 .
- (c) Plot the value of Z_0 as a function of number of iterations.
- (d) Find the charge density on a strip.
- (e) Determine numerically optimal value of the relaxation coefficient α . Compare with the analytical solution

$$\alpha = \frac{8 - \sqrt{64 - 16r^2}}{r^2},$$

where $r = \cos(\pi/N_x) + \cos(\pi/N_y)$ and $N_x \times N_y$ are the dimensions of the discrete mesh.

1 Theoretical introduction

Generally speaking, finite-difference methods are a class of numerical analysis techniques used for solving differential equations. This is done via breaking down the computational domain (spatial or temporal in most cases) into a finite number of steps which allows for the replacement of differential operators, such as derivatives, with differences.

Methods based on finite differences are very useful for converting nonlinear differential equations into a system of linear equations which can be efficiently solved by matrix algebra techniques, e.g. Gauss-Seidel method and its variants. Since these linear algebra computations are efficiently implemented in modern computers, finite-difference methods are one of the most common approaches to the numerical solution of differential equations.

In our case, we use finite differences to approximate the differential operator from Gauss's law of electrostatics which can be expressed in the integral form as

$$\oint_S \epsilon \mathbf{E} \cdot d\mathbf{S} = Q, \quad (1)$$

where S is a closed surface encompassing the concerned structure. However, in a plane, equation 1 simplifies to

$$\oint_C \epsilon \mathbf{E} \cdot d\mathbf{c} = q_l, \quad (2)$$

where C is a closed curve with linear charge density $q_l = Q/l$. Furthermore, assuming the problem can be considered electrostatic, \mathbf{E} is an irrotational vector field, hence we can put $\mathbf{E} = -\text{grad } \varphi$, simplifying equation 2 even further into the form of

$$\begin{aligned} - \oint_C \epsilon \text{grad } \varphi \cdot d\mathbf{c} &= q_l, \\ \oint_C \epsilon \frac{\partial \varphi}{\partial n} dc &= -q_l. \end{aligned} \quad (3)$$

The integrand from equation 3 represents the flux of electric field

$$d\Psi = \epsilon \frac{\partial \varphi}{\partial n} dc$$

through C enclosing a node in the mesh. This situation is depicted in figure 2 and can be expressed as

$$\begin{aligned} \Psi_1 &= \epsilon_1 \frac{\varphi_1 - \varphi_0}{h} \frac{h}{2} + \epsilon_2 \frac{\varphi_1 - \varphi_0}{h} \frac{h}{2}, \\ \Psi_2 &= \epsilon_1 \frac{\varphi_2 - \varphi_0}{h} h, \\ \Psi_3 &= \epsilon_1 \frac{\varphi_3 - \varphi_0}{h} \frac{h}{2} + \epsilon_2 \frac{\varphi_3 - \varphi_0}{h} \frac{h}{2}, \\ \Psi_4 &= \epsilon_2 \frac{\varphi_4 - \varphi_0}{h} h. \end{aligned}$$

However, from Gauss's law, we know that $\Psi_1 + \Psi_2 + \Psi_3 + \Psi_4 = 0$ because there is no charge present. This yields

$$\frac{\epsilon_1 + \epsilon_2}{2} \varphi_0 = \frac{1}{4} \left(\frac{\epsilon_1 + \epsilon_2}{2} \varphi_1 + \epsilon_1 \varphi_2 + \frac{\epsilon_1 + \epsilon_2}{2} \varphi_3 + \epsilon_2 \varphi_4 \right). \quad (4)$$

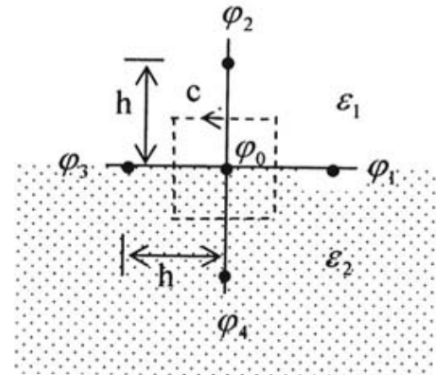


Figure 2: Modified Gauss's law around a node of potential φ_0

This final equation serves as the finite-difference approach to solving the electrostatic potential distribution in the *free nodes* of the configuration, i.e., nodes not in the electrodes or planar dielectric interfaces. In our environment, the permittivities are $\epsilon_1 = \epsilon_0$, $\epsilon_2 = \epsilon_r \epsilon_0$ and a constant voltage is connected to the strips. The rest of the procedure is successive over-relaxation.

Second part of the calculation yields charge density from which we can extract information about the capacity per unit length and the characteristic impedance.

Again, we utilize the equation 3 which now has a non-zero right-hand side because the curve encompasses the charged strips. The situation is similar to a one-electrode problem in figure 3.

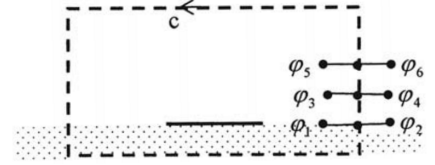


Figure 3: Modified Gauss's law around a charged electrode

Adapting this knowledge, we can write

$$\Psi = \epsilon_1 \frac{\varphi_2 - \varphi_1}{2h} \frac{h}{2} + \epsilon_2 \frac{\varphi_2 - \varphi_1}{2h} \frac{h}{2} + \epsilon_2 \frac{\varphi_4 - \varphi_3}{2h} h + \epsilon_2 \frac{\varphi_6 - \varphi_5}{2h} h + \dots = q_l \quad (5)$$

which serves as the main computational formula for further results.

2 Implementation

The code below represents the core computational loop. The first step is to update the values of electric potential over the whole mesh using equation 4. Then, for a given iteration, the script computes charge density using equation 5. Results are saved for every iteration in order to track the evolution of impedance values. The loop breaks once the previous relaxation step hasn't improved the solution more than the threshold value 10^{-3} .

```

while Rmax > 1e-3
    % Step 1: SOR method
    for j = 2:Nx-1 for i = 2:Ny-1
        if i == IF1
            R(i,j) = (eps_r*V(i+1,j)+1*V(i-1,j)+er_avg*V(i,j-1)+ ...
                er_avg*V(i,j+1))/4/er_avg-V(i,j);
        elseif i == IF2
            R(i,j) = (1*V(i+1,j)+eps_r*V(i-1,j)+er_avg*V(i,j-1)+ ...
                er_avg*V(i,j+1))/4/er_avg-V(i,j);
        else
            R(i,j) = (V(i+1,j)+V(i-1,j)+V(i,j-1)+V(i,j+1))/4-V(i,j);
        elseif i == IF1-1 && ( ...
            (j >= Nx/2-g/2-w/2 && j <= Nx/2-g/2+w/2) || ...
            (j >= Nx/2+g/2-w/2 && j <= Nx/2+g/2+w/2))
            R(i,j)= R(i,j);
            V(i,j)= V(i,j);
        end
        V(i,j) = V(i,j)+alpha*R(i,j);
    end end
    Rmax = max(max(R));
    % Step 2: numerical evaluation of the intergral
    qA = 0; qB = 0; qC = 0; qD = 0;
    for i=top:1:bottom
        if i < IF1
            qA = qA + eps_r*(V(i,left+1) - V(i,left-1))/2;
            qC = qC + eps_r*(V(i,right-1) - V(i,right+1))/2;
        elseif i == IF1
            qA = qA + er_avg*(V(i,left+1) - V(i,left-1))/2;
            qC = qC + er_avg*(V(i,right-1) - V(i,right+1))/2;
        else
            qA = qA + (V(i,left+1) - V(i,left-1))/2;
            qC = qC + (V(i,right-1) - V(i,right+1))/2;
        end
    end
    for j=left:1:right
        qB = qB + (V(bottom+1,j) - V(bottom-1,j))/2;
        qD = qD + eps_r*(V(top-1,j) - V(top+1,j))/2;
    end
    Q = eps_0*(qA + qB + qC + qD);
end
C = Q/(V0-Vgnd);
Z = 1./(c*C);

```

3 Results

- (a) Figure 4 shows the resultant electric field distribution.
- (b) The final value of characteristic impedance is close to $50\ \Omega$ can be seen in the title of figure 4 along with the capacity per unit length.
- (c) In figure 5, we can inspect the evolution of characteristic impedance throughout the run.
- (d) Charge density can be inspected in figure 6.
- (e) Numerical solution for the optimal value of α is 1.94 which is a clear global minimum in figure 7 and which is compliant with the analytical value of $\alpha_0 = 1.96$.

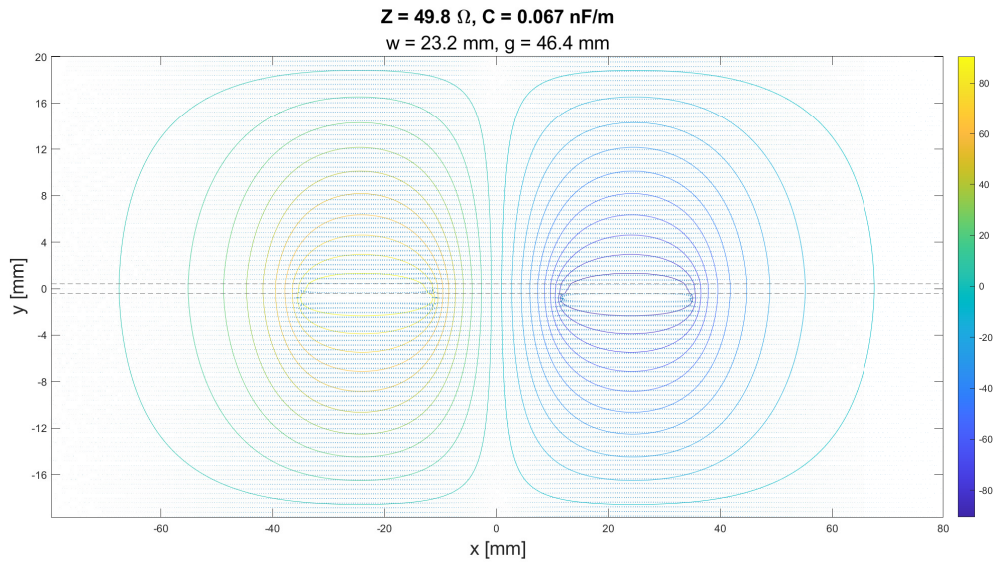


Figure 4: Output field

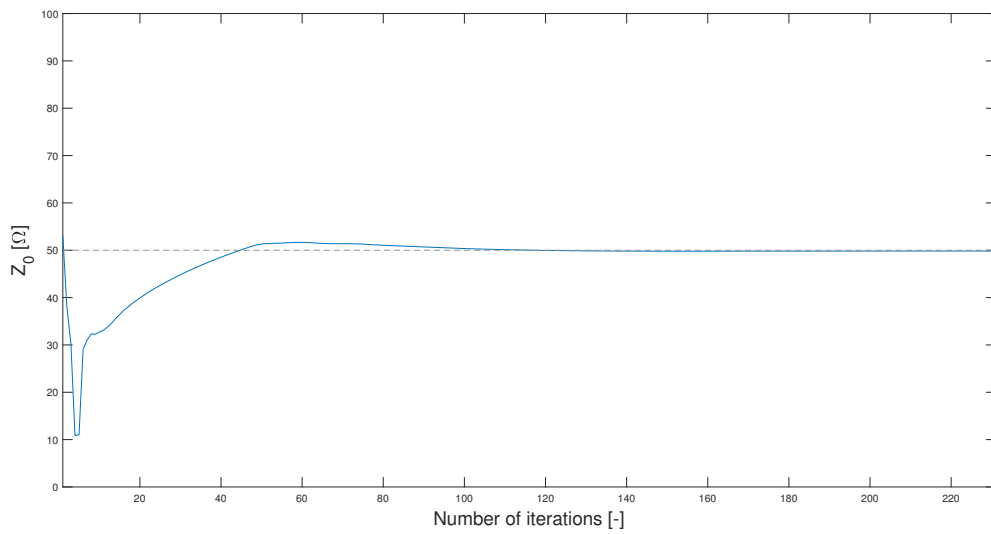


Figure 5: Evolution of the characteristic impedance over iterations

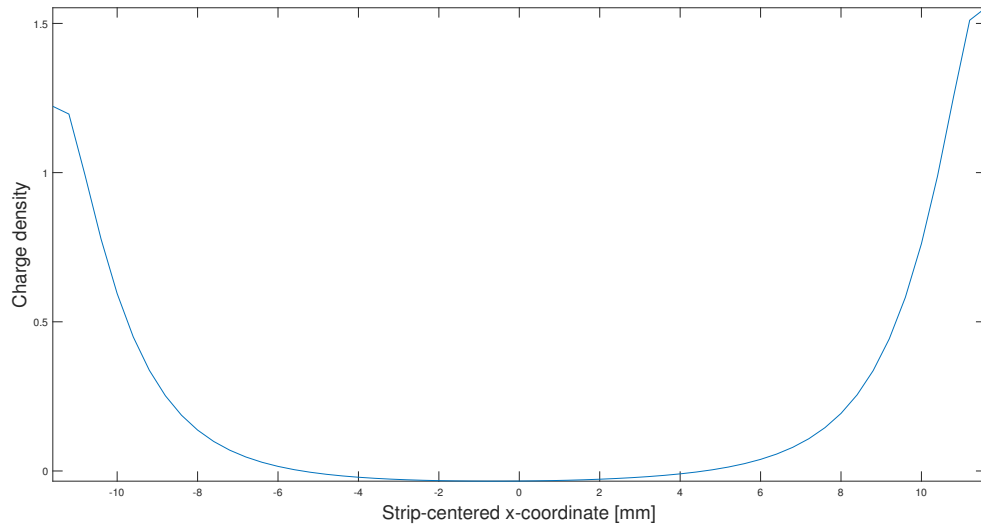


Figure 6: Charge density of the left (positively charged) strip

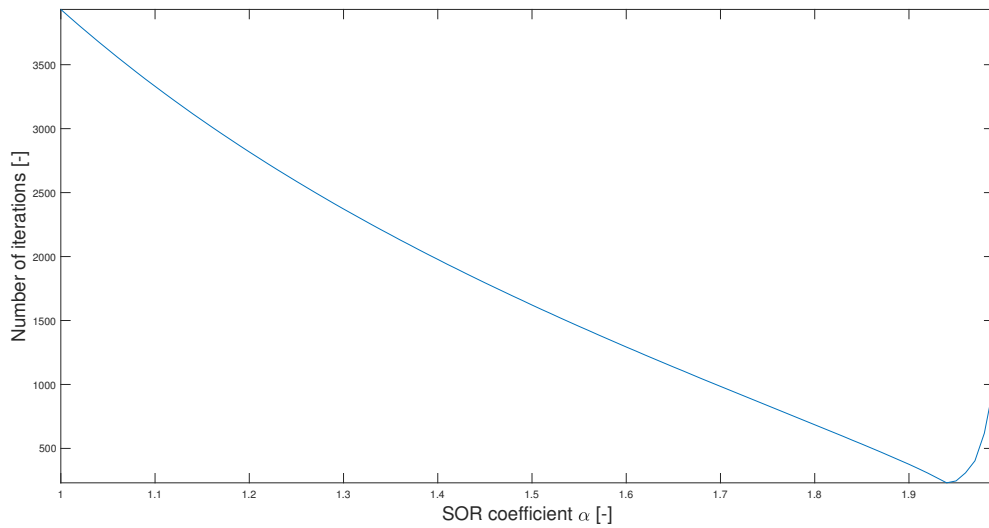


Figure 7: Number of iterations as a function of the SOR coefficient α

4 Conclusion

The main goal of this project was to get a hands-on experience with solving an electromagnetic problem. To achieve that, we have utilized the method of finite differences, which we have implemented in Matlab. Although quite time-consuming, I consider this project a valuable challenge and a lesson for my future work. During implementation, I encountered a handful of issues which I was capable of resolving with a tad bit of guidance from doc. Polívka.