



Computational Science:  
Computational Methods in Engineering

# Implementation of Transmission Line Analysis



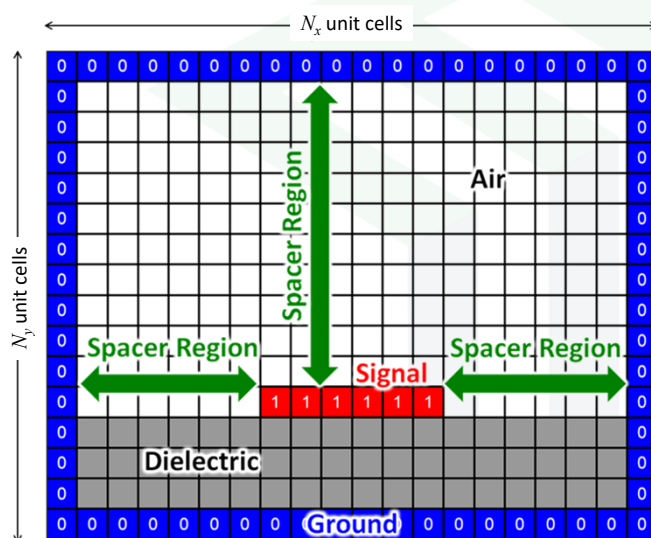
## Outline

- Building the device on the grid
- $2\times$  grid technique
- Incorporating voltage across the line
- Calculating the solution



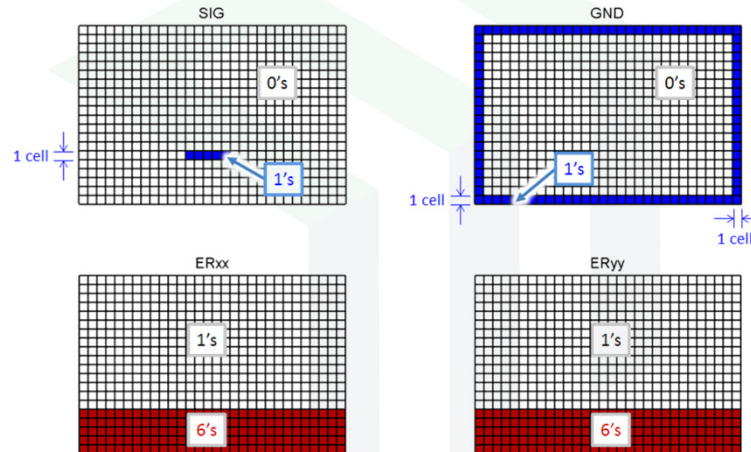
# Building the Device on the Grid

## Grid Representation of a Microstrip Transmission Line



The spacer regions are needed so that the transmission line is not affected by the edges of the grid.

## Four Arrays to Describe a Transmission Line ( $\epsilon_r = 6$ )



Note: If you wish to simulate a transmission line with more than two conductors, you will need one array for each conductor in addition to the two permittivity arrays.

## Building the Diagonal Matrices $\epsilon_{xx}$ and $\epsilon_{yy}$

1. Build ERxx and ERyy using the 2x grid technique.
2. Reshape these 2D arrays into 1D arrays (column vectors).  
`ERxx = ERxx(:);`  
`ERyy = ERyy(:);`
3. Declare the 1D arrays as sparse.  
`ERxx = sparse(ERxx);`  
`ERyy = sparse(ERyy);`
4. Convert the 1D sparse arrays into sparse diagonal matrices.  
`ERxx = diag(ERxx);`  
`ERyy = diag(ERyy);`

Actually, all of this can be accomplished in a single line of MATLAB code:

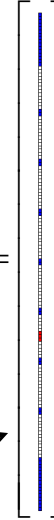
```
ERxx = diag(sparse(ERxx(:)));
ERyy = diag(sparse(ERyy(:)));
```

## The Forced Potentials Grid

$$v_f(x, y)$$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$[v_f] =$$



Reshaped to a  
column vector.  
`vf = vf(:);`

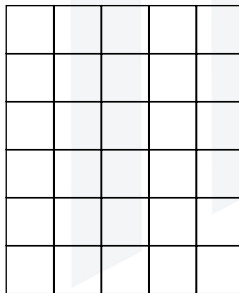
## 2× Grid Technique

## 2x Grid Technique (1 of 9)

Define the ordinary "1x" grid as usual.

The output of this step is the number of cells in the grid,  $N_x$  and  $N_y$ , and the size of the cells in the grid,  $dx$  and  $dy$ .

```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;
```

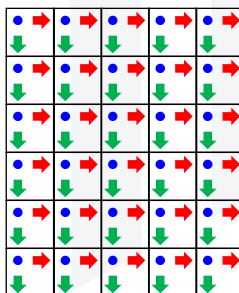


## 2x Grid Technique (2 of 9)

Recall how the various functions overlay onto the grid.

Functions assigned to the same grid cell are in physically different positions and may reside in different materials as a result.

```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;
```



## 2× Grid Technique (3 of 9)

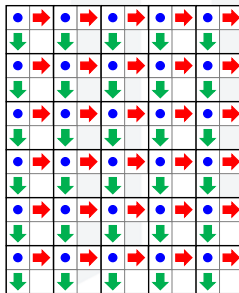
It is like the grid has twice the resolution due to the staggering of the functions.

In order to sort out what values go where, construct a “2×” grid at twice the resolution of the original grid.

The 2× grid occupies the same physical amount of space as the original grid.

```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;
```

```
% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;
```



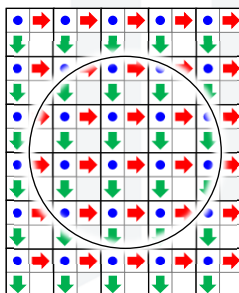
## 2× Grid Technique (4 of 9)

Suppose it is desired to construct a cylinder of radius 2 on the grid.

```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;
```

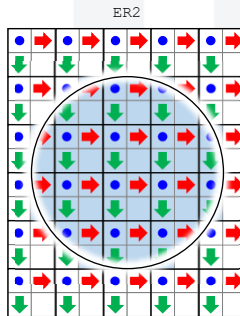
```
% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;
```

```
% CREATE CYLINDER
r = 2;
```



## 2x Grid Technique (5 of 9)

Start by building the cylinder on the 2x grid, ignoring anything about or original grid for now.



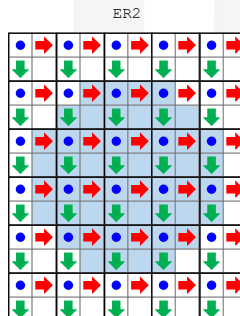
```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

% CREATE CIRCLE
r = 2;
xa2 = [0:Nx2-1]*dx2;
ya2 = [0:Ny2-1]*dy2;
xa2 = xa2 - mean(xa2);
ya2 = ya2 - mean(ya2);
[Y2,X2] = meshgrid(ya2,xa2);
ER2 = (X2.^2 + Y2.^2) <= r^2;
```

## 2x Grid Technique (6 of 9)

Given the object on the 2x grid, extract ERxx by grabbing values from ER2 that correspond to the locations of ERxx.



```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

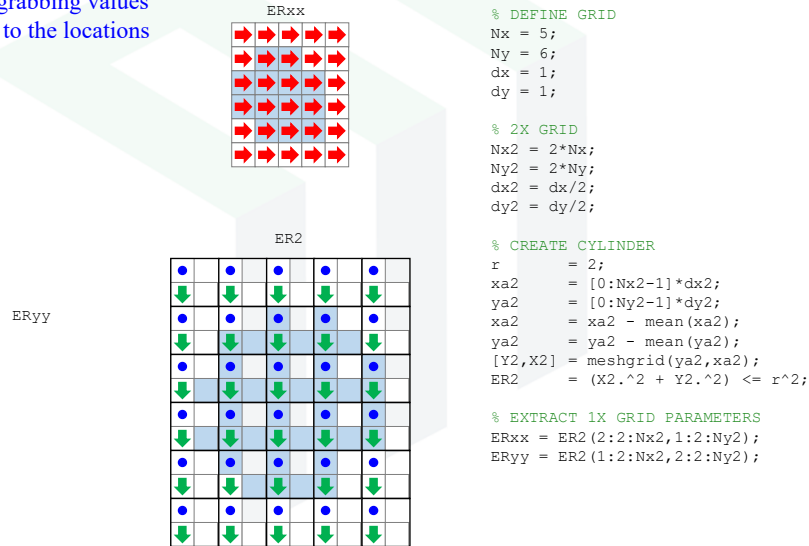
% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

% CREATE CYLINDER
r = 2;
xa2 = [0:Nx2-1]*dx2;
ya2 = [0:Ny2-1]*dy2;
xa2 = xa2 - mean(xa2);
ya2 = ya2 - mean(ya2);
[Y2,X2] = meshgrid(ya2,xa2);
ER2 = (X2.^2 + Y2.^2) <= r^2;

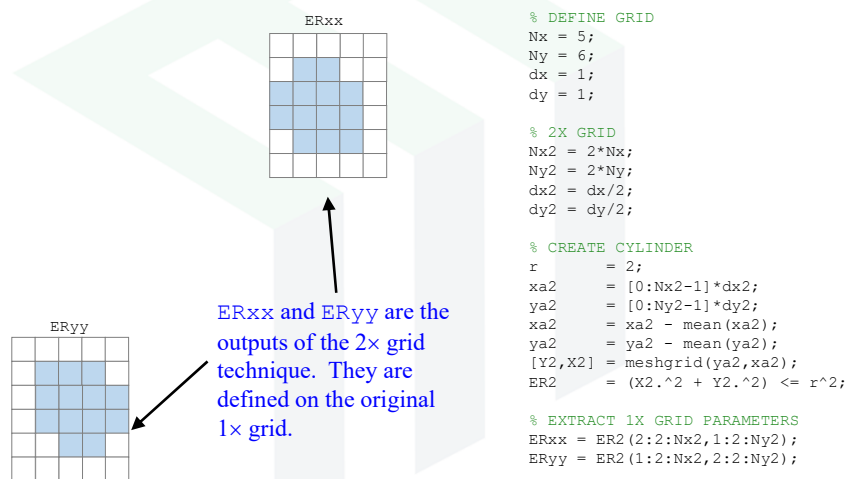
% EXTRACT 1X GRID PARAMETERS
ERxx = ER2(2:2:Nx2,1:2:Ny2);
```

## 2x Grid Technique (7 of 9)

We then extract  $ER_{yy}$  by grabbing values from  $ER_2$  that correspond to the locations of  $ER_{yy}$ .



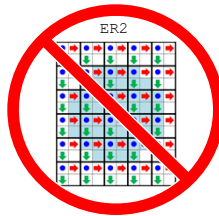
## 2x Grid Technique (8 of 9)



## 2x Grid Technique (9 of 9)

After building  $ER_{xx}$  and  $ER_{yy}$ , the 2x grid is no longer used anywhere.

All of the 2x grid parameters may be deleted at this point because they are no longer needed.



```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

% CREATE CYLINDER
r = 2;
xa2 = [0:Nx2-1]*dx2;
ya2 = [0:Ny2-1]*dy2;
xa2 = xa2 - mean(xa2);
ya2 = ya2 - mean(ya2);
[Y2,X2] = meshgrid(ya2,xa2);
ER2 = (X2.^2 + Y2.^2) <= r^2;

% EXTRACT 1X GRID PARAMETERS
ERxx = ER2(2:2:Nx2,1:2:Ny2);
ERyy = ER2(1:2:Nx2,2:2:Ny2);
```

**No more 2x grid!!!**

# Incorporating Voltage Across the Line

## Can the Matrix Equation be Solved?

So far the following steps have been performed:

- (1) Construction of the permittivity matrices,  $\boldsymbol{\epsilon}_{xx}$  and  $\boldsymbol{\epsilon}_{yy}$ .
- (2) Construction of the derivative operators,  $\mathbf{D}_x^e$  and  $\mathbf{D}_y^e$ .
- (3) Construction of the matrix problem.

This enables the final matrix equation to be constructed as:

$$\begin{bmatrix} \mathbf{D}_x^e & \mathbf{D}_y^e \end{bmatrix} \begin{bmatrix} \boldsymbol{\epsilon}_{xx} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\epsilon}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{D}_x^v \\ \mathbf{D}_y^v \end{bmatrix} \mathbf{v} = \mathbf{0}$$

Can this be solved for  $\mathbf{v}$ ?

$$\mathbf{L} = \begin{bmatrix} \mathbf{D}_x^e & \mathbf{D}_y^e \end{bmatrix} \begin{bmatrix} \boldsymbol{\epsilon}_{xx} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\epsilon}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{D}_x^v \\ \mathbf{D}_y^v \end{bmatrix}$$

$$\mathbf{L}\mathbf{v} = \mathbf{0}$$

$$\mathbf{v} = \mathbf{L}^{-1}\mathbf{0} = \mathbf{0}$$

→ This is just a trivial solution!

## What is Missing?

$$\mathbf{L}\mathbf{v} = \mathbf{0} \quad \rightarrow \quad \mathbf{v} = \mathbf{0}$$

This is missing the excitation  $\mathbf{b}$  → The forced potentials.  
The metals of the transmission line must be set to some known voltage.

### How is this done?

1. The matrix  $\mathbf{L}$  must be modified.
2. The column vector  $\mathbf{b}$  must be calculated.

$$\mathbf{L}\mathbf{v} = \mathbf{0} \quad \rightarrow \quad \boxed{\mathbf{L}'\mathbf{v} = \mathbf{b}}$$

## Force Known Potentials (1 of 2)

For each point where there is a forced potential, we do the following to the corresponding row in the matrix equation:

1. Replace entire row in  $\mathbf{L}$  with all zeros.
2. Place a 1 at the diagonal element.
3. Place the forced potential in  $\mathbf{b}$ .

$$\begin{array}{c}
 \begin{bmatrix}
 (\#) & (\#) & (\#) & \cdots & (\#) & (\#) & (\#) \\
 (\#) & (\#) & (\#) & \cdots & (\#) & (\#) & (\#) \\
 \vdots & & & & & & \\
 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\
 \vdots & & & & & & \\
 (\#) & (\#) & (\#) & \cdots & (\#) & (\#) & (\#) \\
 (\#) & (\#) & (\#) & \cdots & (\#) & (\#) & (\#)
 \end{bmatrix}
 \begin{bmatrix}
 V_1 \\
 V_2 \\
 \vdots \\
 V_m \\
 \vdots \\
 V_{N_x N_y - 1} \\
 V_{N_x N_y}
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 \vdots \\
 V_{\text{applied}} \\
 \vdots \\
 0 \\
 0
 \end{bmatrix}
 \end{array}
 \rightarrow V_m = V_{\text{applied}}$$

The applied voltages are usually just 0 and  $V_0$ , but you may use other values if you are doing something special.

## Force Known Potentials (2 of 2)

Define two functions:

$\mathbf{F} \equiv$  Force matrix

Diagonal matrix containing 1's in the diagonal positions corresponding to values with a known potential (i.e. metals). 0's otherwise.

$\mathbf{v}_f \equiv$  forced potentials

Column vector containing the forced potentials. Numbers in positions not being forced are ignored.

1. Replace rows in  $\mathbf{L}$  with all zeros that correspond to metals on the grid.

$$\mathbf{L}'' = (\mathbf{I} - \mathbf{F})\mathbf{L}$$

2. Then place a 1 in the diagonal element by adding  $\mathbf{F}$  to  $\mathbf{L}''$ .

$$\boxed{\mathbf{L}' = \mathbf{F} + (\mathbf{I} - \mathbf{F})\mathbf{L}} = \mathbf{F} + \mathbf{L}''$$

3. Place the forced potentials in  $\mathbf{b}$ . The remaining elements in  $\mathbf{b}$  must be 0 in order to be consistent with Laplace's equation..

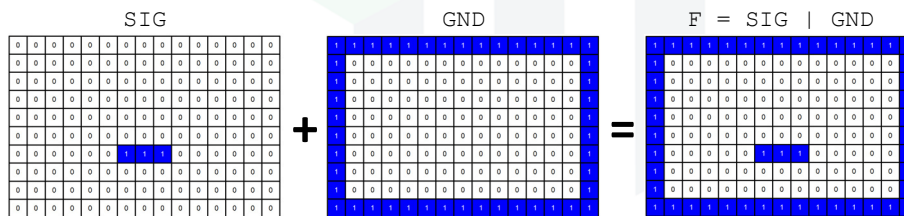
$$\boxed{\mathbf{b} = \mathbf{F}\mathbf{v}_f}$$

## The Force Matrix

The force matrix  $\mathbf{F}$  starts off as an array  $F(i, j)$  containing 1's in the positions that must be forced to a known potential and set to 0's everywhere else.

There are already two arrays defined this same way that describe the two conductors of the transmission line.

Simply combine all of the conductor arrays to construct the force array.



Last, construct the force matrix by diagonalizing the force array.

```
F = diag(sparse(F(:)));
```

## Calculating the Solution

## Complete Matrix Solution

The problem is ready to be solved. The scalar potential  $V$  is calculated as

$$\mathbf{v} = (\mathbf{L}')^{-1} \mathbf{b}$$

Next, calculate the  $\vec{E}$  field from  $V$ .

$$\begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_x^v \\ \mathbf{D}_y^v \end{bmatrix} \mathbf{v}$$

Next, calculate the  $\vec{D}$  field from the  $\vec{E}$  field and the permittivity tensor.

$$\begin{bmatrix} \mathbf{d}_x \\ \mathbf{d}_y \end{bmatrix} = \begin{bmatrix} \epsilon_{xx} & \mathbf{0} \\ \mathbf{0} & \epsilon_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{bmatrix}$$

There should be an  $\epsilon_0$  here. This was removed from the equation and incorporated into the final calculation of distributed capacitance  $C$  in order to keep the functions normalized.

## Extract the Vector Components

Extract the terms  $\mathbf{e}_x$  and  $\mathbf{e}_y$  from column vectors.

$$\begin{aligned} \mathbf{v} &= \mathbf{L} \backslash \mathbf{b}; \\ \mathbf{e} &= -[\mathbf{D}_x^v; \mathbf{D}_y^v] * \mathbf{v}; \end{aligned} \quad \longrightarrow \quad \mathbf{e} = - \begin{bmatrix} \mathbf{D}_x^v \\ \mathbf{D}_y^v \end{bmatrix} \mathbf{v}$$

$$\begin{aligned} \mathbf{M} &= \mathbf{N}_x * \mathbf{N}_y; \\ \mathbf{e}_x &= \mathbf{e}(1:\mathbf{M}); \\ \mathbf{e}_y &= \mathbf{e}(\mathbf{M}+1:2*\mathbf{M}); \end{aligned} \quad \longrightarrow \quad \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{bmatrix} = \mathbf{e}$$

## Reshape Back to 2D Arrays

The terms  $\mathbf{v}$ ,  $\mathbf{e}_x$ ,  $\mathbf{e}_y$ ,  $\mathbf{d}_x$ , and  $\mathbf{d}_y$  are column vectors, or 1D arrays.

These need to be reshaped back to 2D arrays before they can be visualized.

```
v = reshape( v, Nx, Ny );
ex = reshape( ex, Nx, Ny );
ey = reshape( ey, Nx, Ny );
```

## Distributed Capacitance, $C$

The distributed capacitance  $C$  is calculated by numerical integration of

$$C = \frac{\epsilon_0}{V_0^2} \iint_A (\vec{D} \cdot \vec{E}) dA$$

This is where constant  $\epsilon_0$  is reincorporated.

Assuming  $V_0 = 1$ , the numerical integration is easily evaluated as

$$C = \mathbf{d}^T \mathbf{e} (\epsilon_0 \Delta x \Delta y)$$

```
c = d.'*e*(e0*dx*dy);
```

## Distributed Inductance, $L$

First, the distributed capacitance  $C_h$  for the homogeneous case is calculated.

$$C_h = \mathbf{d}_h^T \mathbf{e}_h (\epsilon_0 \Delta x \Delta y)$$

$$C_h = d_h \cdot \epsilon_h \cdot (\epsilon_0 \cdot dx \cdot dy);$$

Second, the distributed inductance  $L$  is calculated from  $C_h$ .

$$L = \frac{\mu_{r,h}}{c_0^2 C_h}$$

$$L = \mu_{r,h} / (c_0^2 \cdot C_h);$$

## Characteristic Impedance, $Z_0$

Given the distributed inductance  $L$  and distributed capacitance  $C$ , the characteristic impedance  $Z_0$  of the transmission line is

$$Z_0 = \sqrt{\frac{L}{C}}$$

$$Z_0 = \text{sqrt}(L/C);$$

## Effective Dielectric Constant $\epsilon_{r,\text{eff}}$ & Phase Constant $\beta$

Given the distributed inductance  $L$  and distributed capacitance  $C$ , the effective dielectric constant  $\epsilon_{r,\text{eff}}$  and the phase constant  $\beta$  of the transmission line is

$$\epsilon_{r,\text{eff}} = c_0^2 LC$$

$$\epsilon_{\text{reff}} = c0^2 * L * C;$$

$$\beta = \omega \sqrt{LC}$$

$$\text{beta} = \text{omega} * \text{sqrt}(L * C);$$