VirtualLab Fusion Technology – Solvers and Functions

Layer Matrix [S-Matrix]

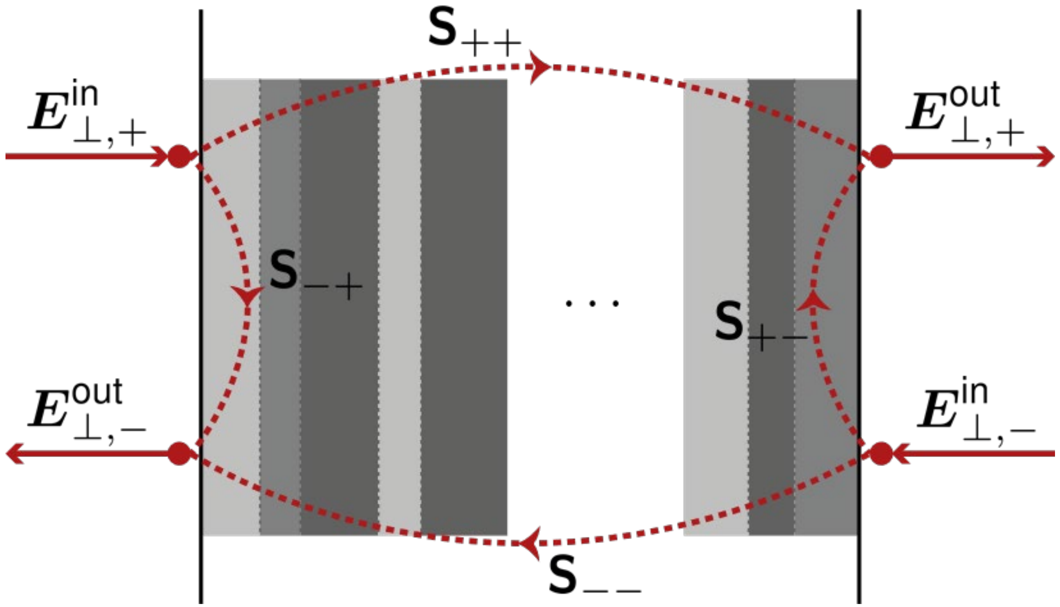
For the **Stratified Medium Component**

Abstract

The layer matrix solver works in the spatial frequency domain (**k domain**). It consists of

1. an eigenmode solver for each homogeneous layer and
2. an S-matrix for matching the boundary conditions at all the interfaces.

The eigenmode solver computes the field solution in the k domain for the homogeneous medium in each layer. The S-matrix algorithm calculates the response of the whole layer system by matching the boundary conditions in a recursive manner. It is well-known for its unconditional numerical stability since, unlike the traditional transfer matrix, it avoids the exponentially growing functions in the calculation steps.



Solver Algorithm – Eigenmode Solver

- In the Fresnel matrix calculation, we deal with Maxwell's equations for homogeneous isotropic media, as written below

$$\nabla \times \boldsymbol{E}(\boldsymbol{r}) = \mathrm{i}k_0 \boldsymbol{H}(\boldsymbol{r})\,,$$
$$\nabla \times \boldsymbol{H}(\boldsymbol{r}) = -\mathrm{i}k_0 \epsilon \boldsymbol{E}(\boldsymbol{r})$$

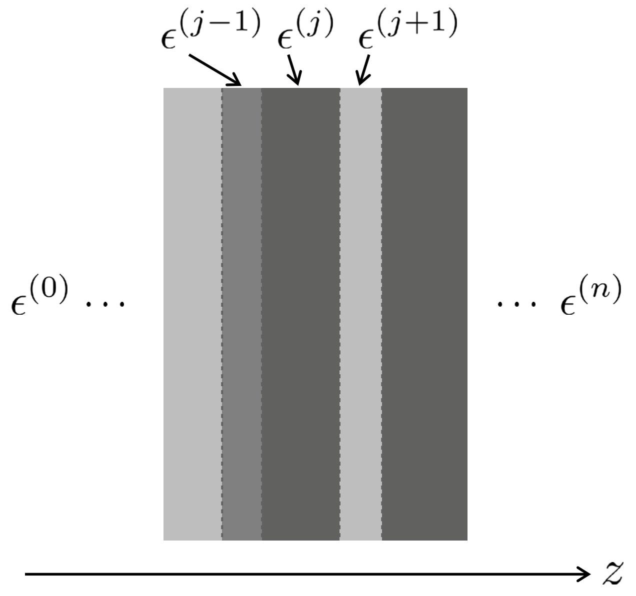
with constant **permittivity** $\epsilon = \epsilon^{(j)}$, for the layer with index j.

- The eigenmode solution in the k domain can be found via

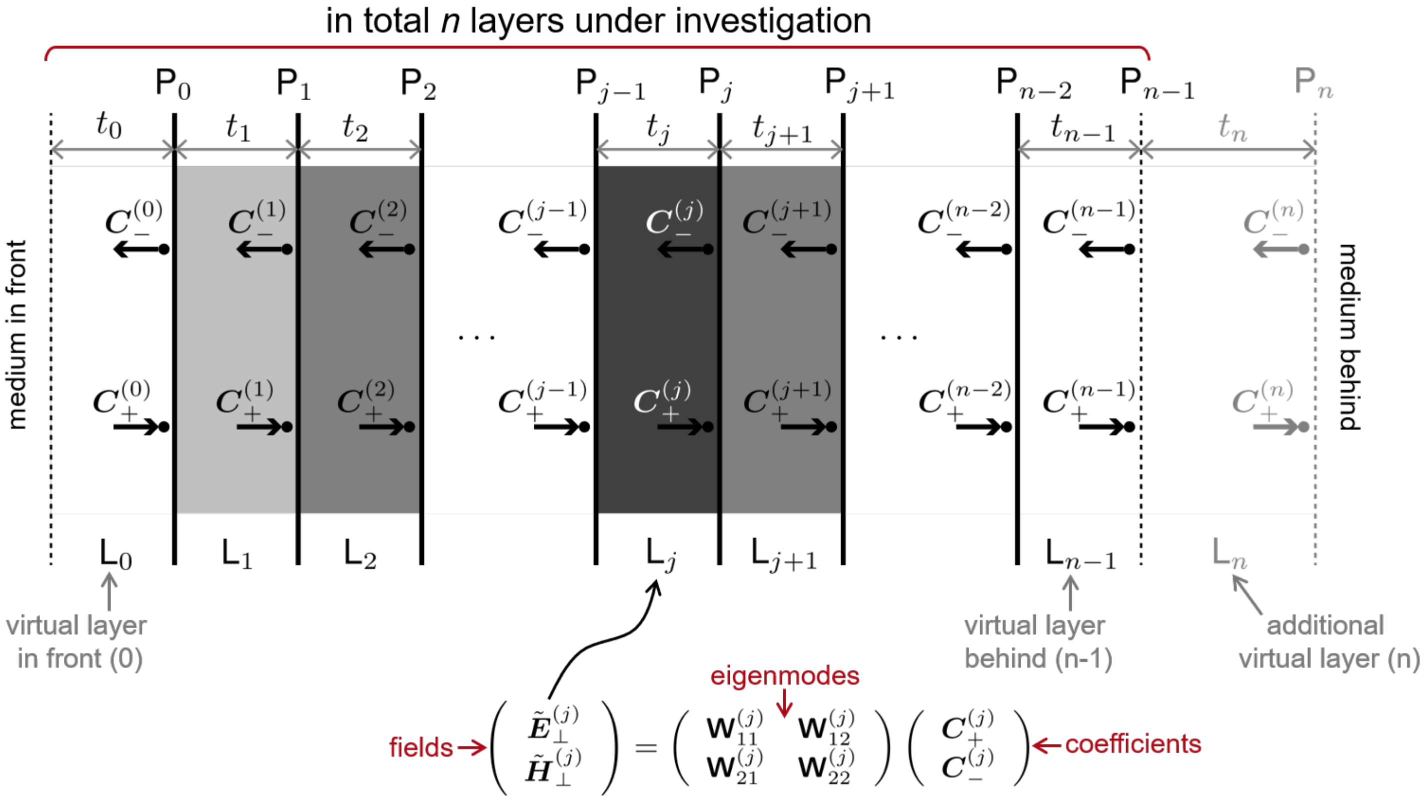
$$\begin{pmatrix} \tilde{E}_x(\boldsymbol{\kappa}, z) \\ \tilde{E}_y(\boldsymbol{\kappa}, z) \\ \tilde{H}_x(\boldsymbol{\kappa}, z) \\ \tilde{H}_y(\boldsymbol{\kappa}, z) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -\tilde{W}_B & -\tilde{W}_D & \tilde{W}_B & \tilde{W}_D \\ \tilde{W}_C & \tilde{W}_B & -\tilde{W}_C & -\tilde{W}_B \end{pmatrix} \begin{pmatrix} \tilde{C}_+^{\mathrm{I}} \exp(\gamma z) \\ \tilde{C}_+^{\mathrm{II}} \exp(\gamma z) \\ \tilde{C}_-^{\mathrm{I}} \exp(-\gamma z) \\ \tilde{C}_-^{\mathrm{II}} \exp(-\gamma z) \end{pmatrix},$$

with $\tilde{W}_B = \frac{n_x n_y}{n_z}$, $\tilde{W}_C = n_z + \frac{n_x^2}{n_z}$, $\tilde{W}_D = n_z + \frac{n_y^2}{n_z}$, $n_x = \frac{k_x}{k_0}$, $n_y = \frac{k_y}{k_0}$, and $n_z = (\epsilon\mu - n_x^2 - n_y^2)^{1/2}$, $\gamma = \mathrm{i}k_0 n_z$.

- Here we use $\boldsymbol{r} = (x, y, z)$ and $\boldsymbol{\rho} = (x, y)$ as the 3D position vector and its 2D projection onto the transversal plane respectively.



Solver Algorithm – S-Matrix



The task of the S-matrix is to compute the coefficients that connect the field in front of and behind the layered slab, as

$$\begin{pmatrix} C_+^{(n)} \\ C_-^{(0)} \end{pmatrix} = \begin{pmatrix} \mathbf{S}_{11}^{(0,n)} & \mathbf{S}_{12}^{(0,n)} \\ \mathbf{S}_{21}^{(0,n)} & \mathbf{S}_{22}^{(0,n)} \end{pmatrix} \begin{pmatrix} C_+^{(0)} \\ C_-^{(n)} \end{pmatrix} .$$

Solver Algorithm – S-Matrix

- At the surface with the index (j), based on the boundary conditions, it is not hard to write down the following relation

$$\begin{pmatrix} \mathbf{W}_{11}^{(j)} & \mathbf{W}_{12}^{(j)} \\ \mathbf{W}_{21}^{(j)} & \mathbf{W}_{22}^{(j)} \end{pmatrix} \begin{pmatrix} \boldsymbol{C}_{+}^{(j)} \\ \boldsymbol{C}_{-}^{(j)} \end{pmatrix} = \begin{pmatrix} \mathbf{W}_{11}^{(j+1)} & \mathbf{W}_{12}^{(j+1)} \\ \mathbf{W}_{21}^{(j+1)} & \mathbf{W}_{22}^{(j+1)} \end{pmatrix} \begin{pmatrix} [\boldsymbol{\Phi}_{+}^{(j+1)}]^{-1} & 0 \\ 0 & [\boldsymbol{\Phi}_{-}^{(j+1)}]^{-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{C}_{+}^{(j+1)} \\ \boldsymbol{C}_{-}^{(j+1)} \end{pmatrix}.$$

- By applying the boundary conditions at each surface, a recursive relation can be found to relate the field in front of and behind the layered slab in the form below

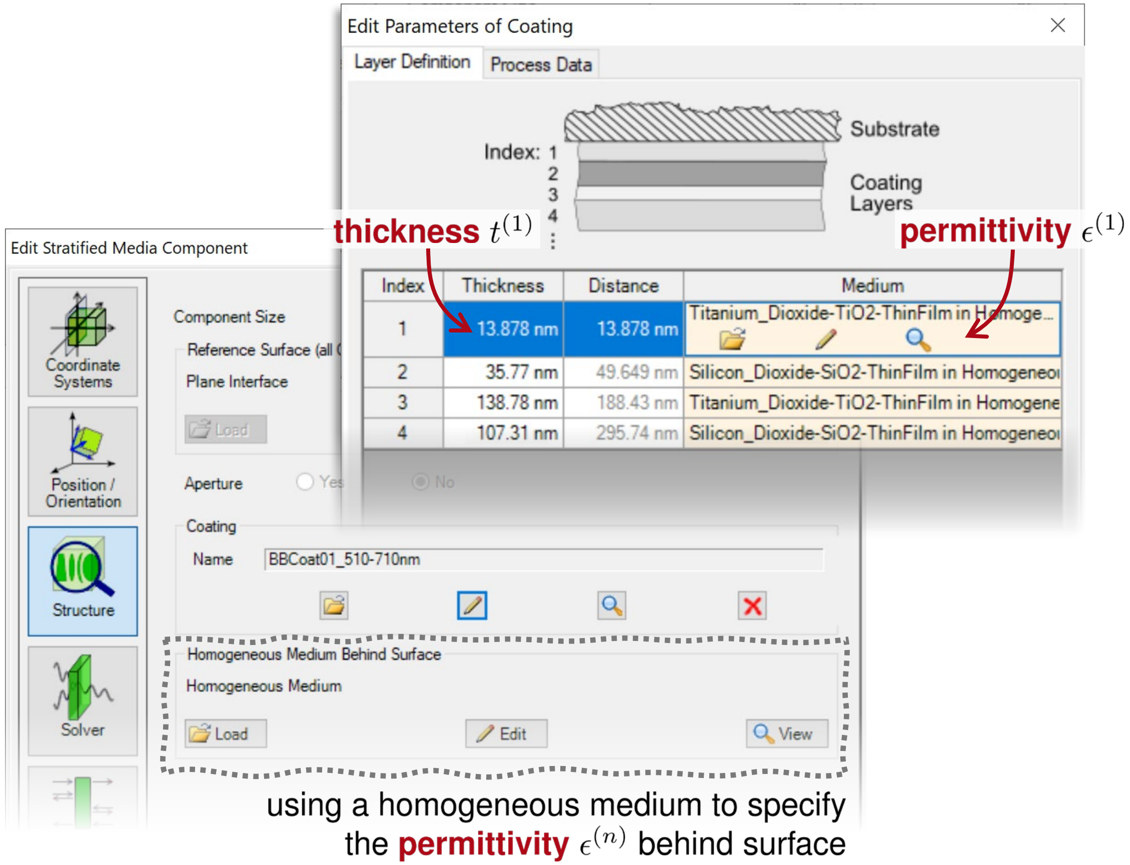
$$\begin{pmatrix} \boldsymbol{C}_{+}^{(n)} \\ \boldsymbol{C}_{-}^{(0)} \end{pmatrix} = \begin{pmatrix} \mathbf{S}_{11}^{(0,n)} & \mathbf{S}_{12}^{(0,n)} \\ \mathbf{S}_{21}^{(0,n)} & \mathbf{S}_{22}^{(0,n)} \end{pmatrix} \begin{pmatrix} \boldsymbol{C}_{+}^{(0)} \\ \boldsymbol{C}_{-}^{(n)} \end{pmatrix}.$$

- There are different variations to derive the recursive relation. In VirtualLab Fusion, we follow the **W→t→S variation**, according to [1, 2].

- Other recrusion variations will become avaliable in VirtualLab Fusion in future.

Usage in VirtualLab Fusion

- Take the Stratified Medium Component as an example:
  - the **permittivity** $\epsilon^{(0)}$ in front of the first surface is determined by the preceding optical setup;
  - the **permittivity** $\epsilon^{(j)}$ for each layer, and its **thickness**, $t^{(j)}$ are specified as a coating;
  - the **permittivity** $\epsilon^{(n)}$ behind the surface is specified by a homogeneous isotropic medium.
- The layer matrix is calculated for each spatial frequency $\boldsymbol{\kappa}$ contained in an arbitrary input field which reaches the plane surface.



List of References

[1] Lifeng Li, "Formulation and comparison of two recursive matrix algorithms for modeling layered diffraction gratings," J. Opt. Soc. Am. A 13, 1024-1035 (1996)

[2] Lifeng Li, "Note on the S-matrix propagation algorithm," J. Opt. Soc. Am. A 20, 655-660 (2003)

Document Information

title	VirtualLab Fusion Technology – Layer Matrix [S-Matrix]
document code	TEC.0002
version	1.0
category	Technology White Paper
further reading	- VirtualLab Fusion Technology – Fresnel Matrix - VirtualLab Fusion Technology – FMM / RCWA [S-Matrix]