VirtualLab Fusion Technology – Solvers and Functions

Local Plane Interface Approximation (LPIA)

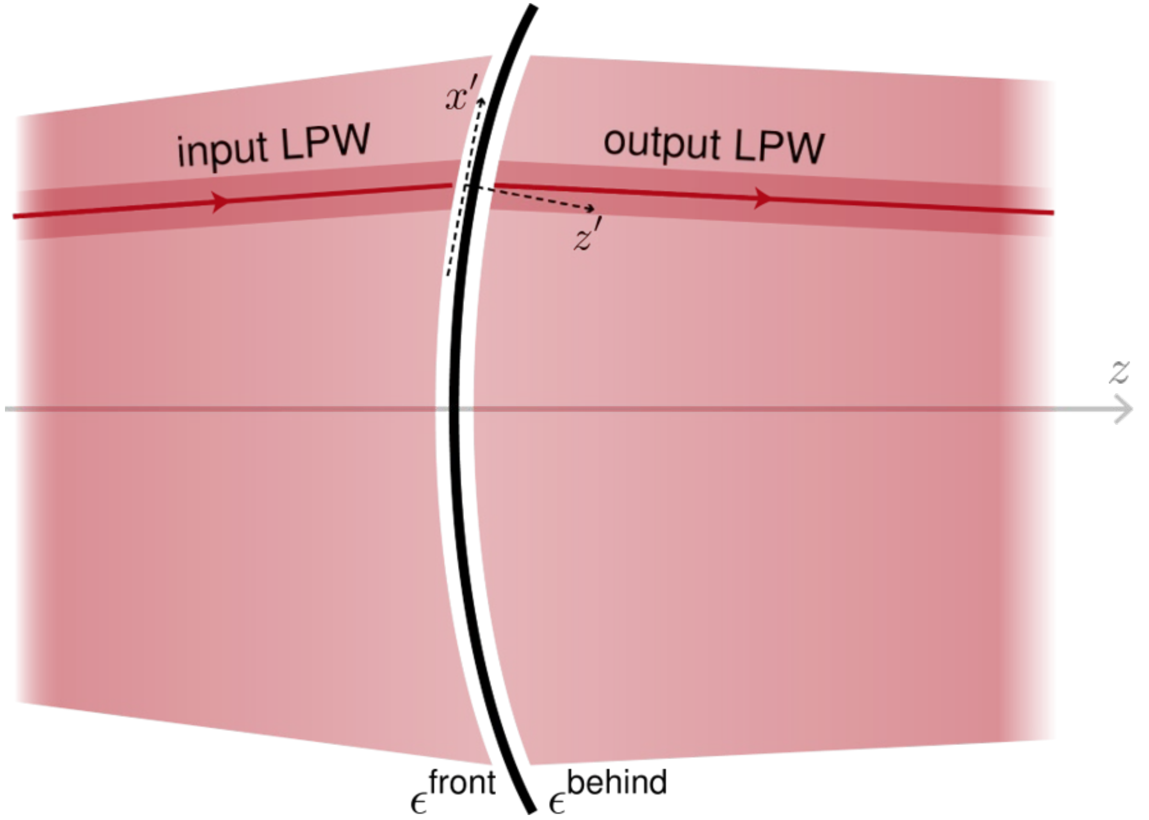
For the **Curved Surface Component, Lens System Component, Spherical Lens Component, Off-Axis Parabolic Mirror (Wedge Type) Component**

Abstract

The LPIA solver works in the spatial domain (**x domain**), locally, in a pointwise manner. The solver follows that

1. the input field on the surface is treated as a composition of local plane waves (LPWs),
2. the part of the surface seen by each LPW is considered a plane interface (locally), and,
3. the interaction of the LPW with the local plane interface can be modeled by the Fresnel (or the layer) matrix.

At an arbitrary location on the curved surface, an approximate local boundary condition is applied, which assumes the interaction of the LPW with the local plane interface. Thus, the Fresnel matrix (or layer matrix for coatings) can be used to connect input and output fields.



Solver Algorithm

- Both the input and output fields are defined on the **curved surface** $z = h(\boldsymbol{\rho})$, in the medium with **permittivity** ϵ^{front} **and** ϵ^{behind} respectively, in the following form

$$\boldsymbol{V}_{\perp}^{\text{in}}(\boldsymbol{\rho}, z) = \boldsymbol{U}_{\perp}^{\text{in}}(\boldsymbol{\rho}, z) \exp\left(\mathrm{i}\psi^{\text{in}}(\boldsymbol{\rho}, z)\right),$$
$$\boldsymbol{V}_{\perp}^{\text{out}}(\boldsymbol{\rho}, z) = \boldsymbol{U}_{\perp}^{\text{out}}(\boldsymbol{\rho}, z) \exp\left(\mathrm{i}\psi^{\text{out}}(\boldsymbol{\rho}, z)\right),$$

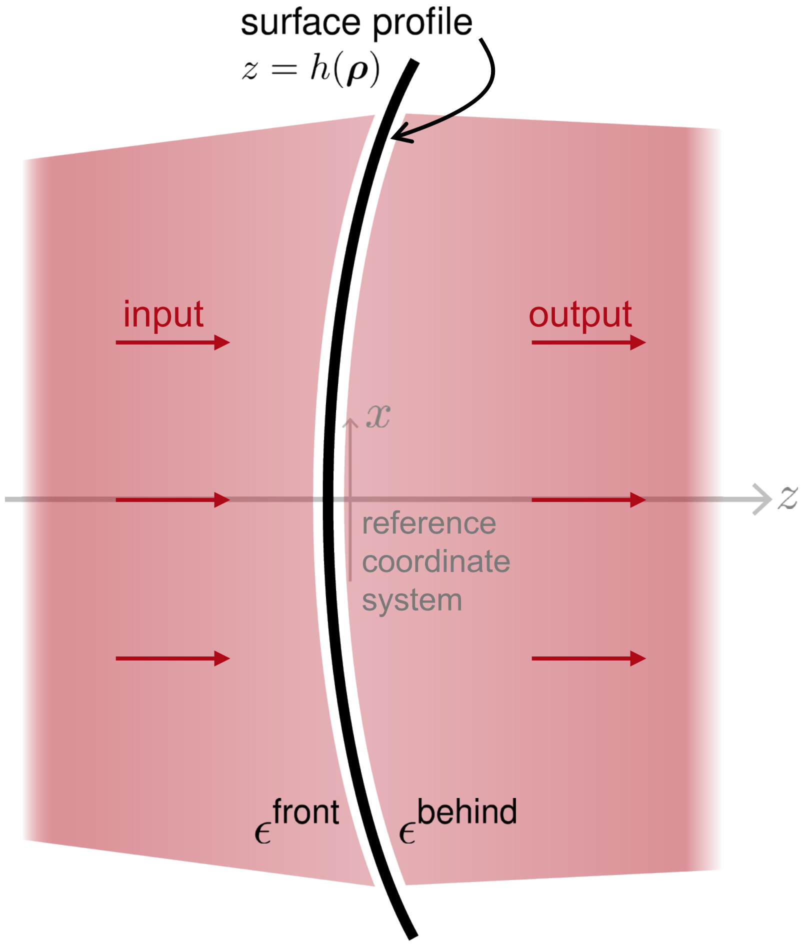
  where ψ is the wavefront phase part, $\boldsymbol{U}_{\perp}$ is the residual fields, and $\boldsymbol{\rho} = (x, y)$ as the transverse coordinates.

- The output field is to be calculated pointwisely

$$\boldsymbol{V}_{\perp}^{\text{out}}(\boldsymbol{\rho}, z) = \mathbf{B}(\boldsymbol{\rho}, z)\boldsymbol{V}_{\perp}^{\text{in}}(\boldsymbol{\rho}, z),$$

  or, explicitly, with

  - the redisual field: $\boldsymbol{U}_{\perp}^{\text{out}}(\boldsymbol{\rho}, z) = \mathbf{b}(\boldsymbol{\rho}, z)\boldsymbol{U}_{\perp}^{\text{in}}(\boldsymbol{\rho}, z)$, and
  - the wavefront phase part: $\psi^{\text{out}}(\boldsymbol{\rho}, z) = \psi^{\text{in}}(\boldsymbol{\rho}, z)$.



Solver Algorithm

- Both the input and output fields are defined on the **curved surface** $z = h(\boldsymbol{\rho})$, in the medium with **permittivity** ϵ^{front} **and** ϵ^{behind} respectively, in the following form

$$\boldsymbol{V}_{\perp}^{\text{in}}(\boldsymbol{\rho}, z) = \boldsymbol{U}_{\perp}^{\text{in}}(\boldsymbol{\rho}, z) \exp\left(\mathrm{i}\psi^{\text{in}}(\boldsymbol{\rho}, z)\right),$$

$$\boldsymbol{V}_{\perp}^{\text{out}}(\boldsymbol{\rho}, z) = \boldsymbol{U}_{\perp}^{\text{out}}(\boldsymbol{\rho}, z) \exp\left(\mathrm{i}\psi^{\text{out}}(\boldsymbol{\rho}, z)\right),$$

  where ψ is the wavefront phase part, $\boldsymbol{U}_{\perp}$ is the residual fields, and $\boldsymbol{\rho} = (x, y)$ as the transverse coordinates.

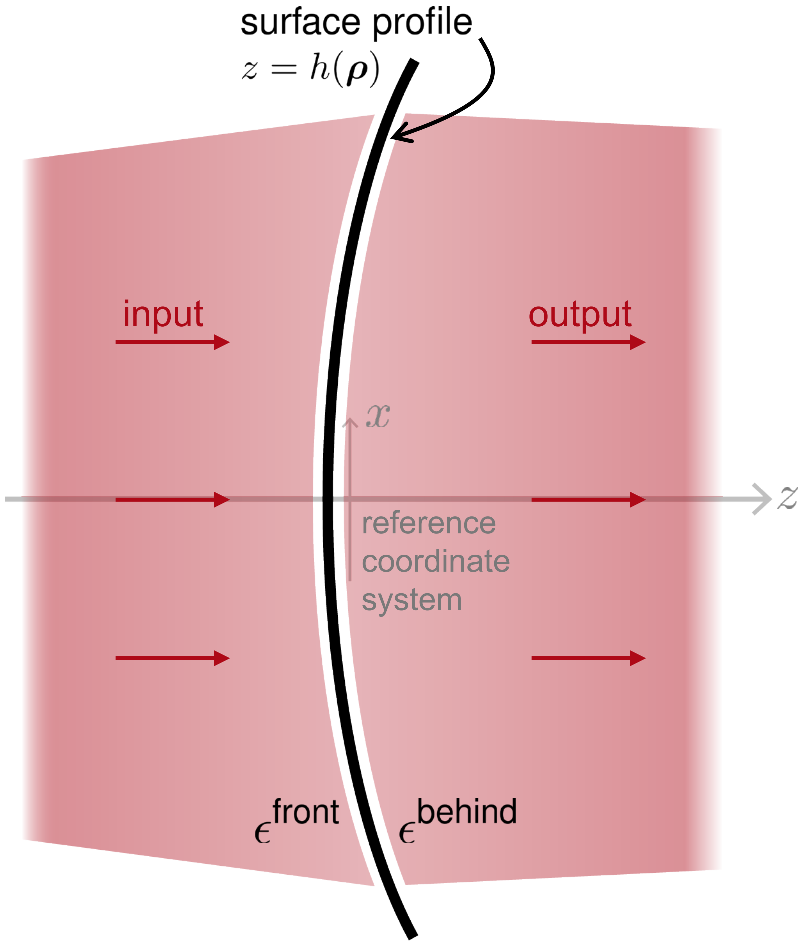
- The output field is to be calculated pointwisely

$$\boldsymbol{V}_{\perp}^{\text{out}}(\boldsymbol{\rho}, z) = \mathbf{B}(\boldsymbol{\rho}, z)\boldsymbol{V}_{\perp}^{\text{in}}(\boldsymbol{\rho}, z)$$

  or, explicitly, with
  - the redisual field: $\boldsymbol{U}_{\perp}^{\text{out}}(\boldsymbol{\rho}, z) = \mathbf{b}(\boldsymbol{\rho}, z)\boldsymbol{U}_{\perp}^{\text{in}}(\boldsymbol{\rho}, z)$, and
  - the wavefront phase part: $\psi^{\text{out}}(\boldsymbol{\rho}, z) = \psi^{\text{in}}(\boldsymbol{\rho}, z)$.

The output field can also be in either +z or -z direction. Here the +z is selected as an example.

The input field can be in either +z or -z direction. Here the +z is selected as an example.



Solver Algorithm

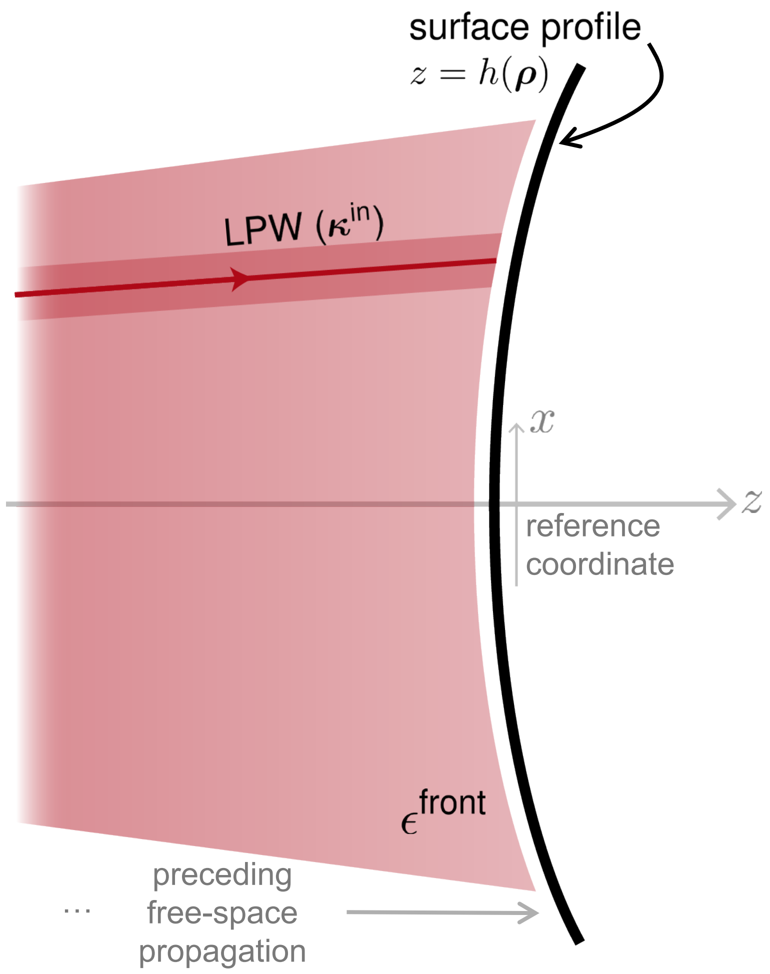
- To calculate **B**, which is equal to **b** in this case, we assume local plane wave and local plane interface interaction:
  - calculate the transverse wavevector components $\boldsymbol{\kappa}^{\text{in}} = (k_x^{\text{in}}, k_y^{\text{in}})$ in the reference coordinate system, on the surface, as

$$k_x^{\text{in}} = \frac{\partial \psi^{\text{in}}(\boldsymbol{\rho}, z)}{\partial x}, \quad k_y^{\text{in}} = \frac{\partial \psi^{\text{in}}(\boldsymbol{\rho}, z)}{\partial y};$$

  - transform to the local coordinate system and calculate $\boldsymbol{\kappa}' = (k'_x, k'_y)$ there (see appendix, [2, 3]);
  - calculate Fresnel matrix $\mathbf{b}'(k'_x, k'_y)$ with ϵ^{front} and ϵ^{behind} (or the layer matrix for the surface coating);
  - apply the Fresnel matrix together with coordinate transformation, and obtain

$$\mathbf{b} = \mathbf{Y}^{\text{ref}} \mathbf{b}' \mathbf{Y}^{\text{loc}},$$

with $\mathbf{Y}^{\text{in/out}}$ as the transformation matrices for the transverse field vectors (see appendix, [2, 3]).



Solver Algorithm

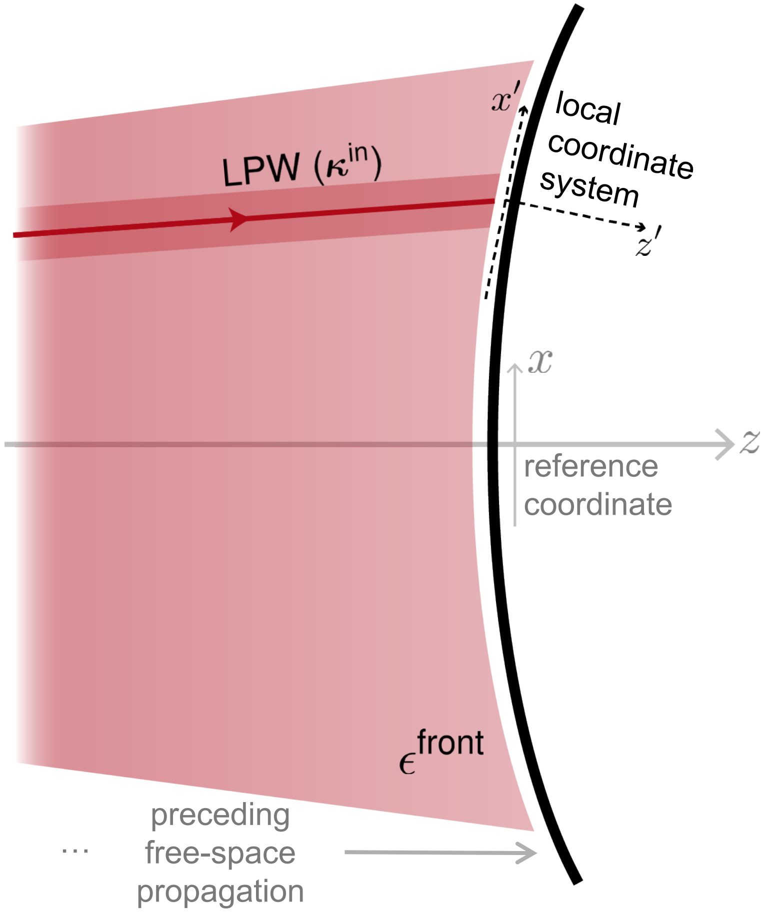
- To calculate **B**, which is equal to **b** in this case, we assume local plane wave and local plane interface interaction:
  - calculate the transverse wavevector components $\boldsymbol{\kappa}^{\text{in}} = (k_x^{\text{in}}, k_y^{\text{in}})$ in the reference coordinate system, on the surface, as

$$k_x^{\text{in}} = \frac{\partial \psi^{\text{in}}(\boldsymbol{\rho}, z)}{\partial x}, \quad k_y^{\text{in}} = \frac{\partial \psi^{\text{in}}(\boldsymbol{\rho}, z)}{\partial y};$$

  - transform to the local coordinate system and calculate $\boldsymbol{\kappa}' = (k_x', k_y')$ there (see appendix, [2, 3]);
  - calculate Fresnel matrix $\mathbf{b}'(k_x', k_y')$ with ϵ^{front} and ϵ^{behind} (or the layer matrix for the surface coating);
  - apply the Fresnel matrix together with coordinate transformation, and obtain

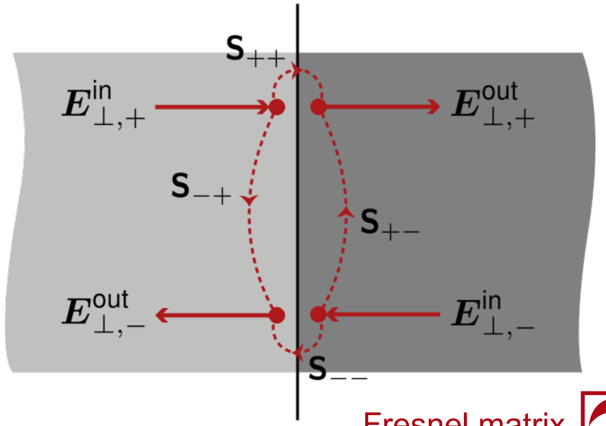
$$\mathbf{b} = \mathbf{Y}^{\text{ref}} \mathbf{b}' \mathbf{Y}^{\text{loc}},$$

    with $\mathbf{Y}^{\text{in/out}}$ as the transformation matrices for the transverse field vectors (see appendix, [2, 3]).

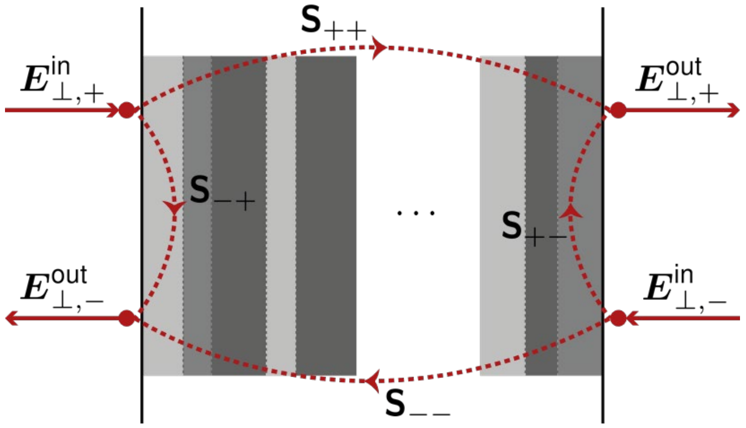


Solver Algorithm

- To calculate $\mathbf{B}$, which is equal to $\mathbf{b}$ in this case, we assume local plane wave and local plane interface interaction:
  - calculate the transverse wavevector components $\boldsymbol{\kappa}^{\text{in}} = (k_x^{\text{in}}, k_y^{\text{in}})$ in the reference coordinate system, on the surface, as
$$k_x^{\text{in}} = \frac{\partial \psi^{\text{in}}(\boldsymbol{\rho}, z)}{\partial x}, \quad k_y^{\text{in}} = \frac{\partial \psi^{\text{in}}(\boldsymbol{\rho}, z)}{\partial y};$$
  - transform to the local coordinate system and calculate $\boldsymbol{\kappa}' = (k_x', k_y')$ there (see appendix, [2, 3]);
  - calculate Fresnel matrix $\mathbf{b}'(k_x', k_y')$ with ϵ^{front} and ϵ^{behind} (or the layer matrix for the surface coating);
  - apply the Fresnel matrix together with coordinate transformation, and obtain
$$\mathbf{b} = \mathbf{Y}^{\text{ref}}\mathbf{b}'\mathbf{Y}^{\text{loc}},$$
with $\mathbf{Y}^{\text{in/out}}$ as the transformation matrices for the transverse field vectors (see appendix, [2, 3]).



Fresnel matrix



layer matrix

Solver Algorithm

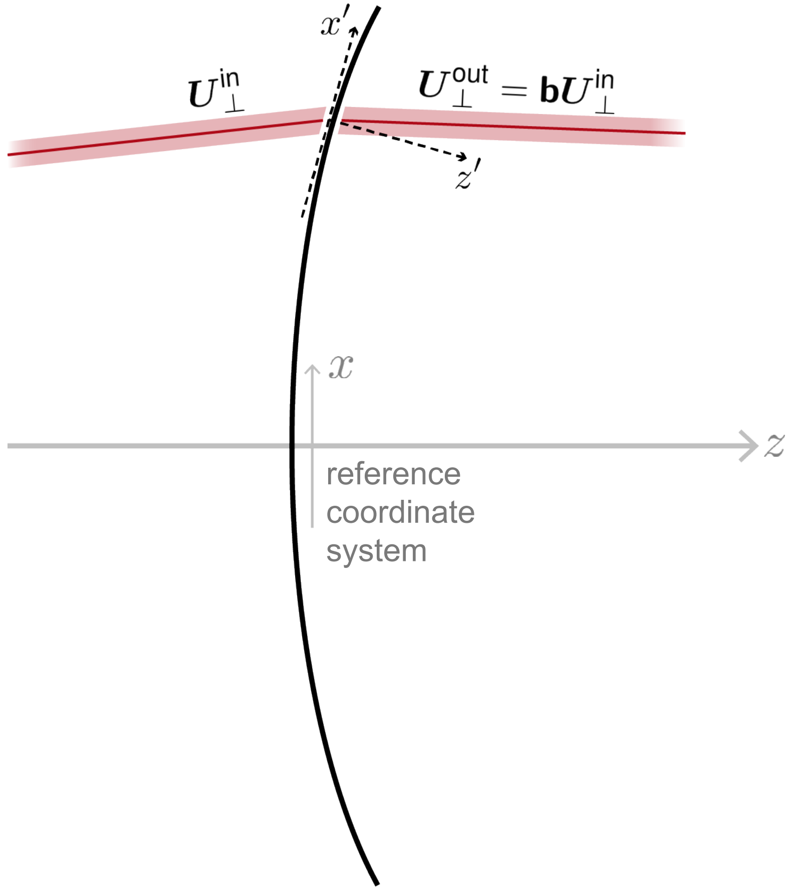
- To calculate **B**, which is equal to **b** in this case, we assume local plane wave and local plane interface interaction:
  - calculate the transverse wavevector components $\boldsymbol{\kappa}^{\text{in}} = (k_x^{\text{in}}, k_y^{\text{in}})$ in the reference coordinate system, on the surface, as

$$k_x^{\text{in}} = \frac{\partial \psi^{\text{in}}(\boldsymbol{\rho}, z)}{\partial x}, \quad k_y^{\text{in}} = \frac{\partial \psi^{\text{in}}(\boldsymbol{\rho}, z)}{\partial y};$$

  - transform to the local coordinate system and calculate $\boldsymbol{\kappa}' = (k'_x, k'_y)$ there (see appendix, [2, 3]);
  - calculate Fresnel matrix $\mathbf{b}'(k'_x, k'_y)$ with ϵ^{front} and ϵ^{behind} (or the layer matrix for the surface coating);
  - apply the Fresnel matrix together with coordinate transformation, and obtain

$$\mathbf{b} = \mathbf{Y}^{\text{ref}} \mathbf{b}' \mathbf{Y}^{\text{loc}},$$

with $\mathbf{Y}^{\text{in/out}}$ as the transformation matrices for the transverse field vectors (see appendix, [2, 3]).



Appendix – Coordinate System and Transformation

- At an arbitrary position on the curved surface, a local coordinate system x'-y'-z' can be defined, with x' and y' as the surface tangential directions.

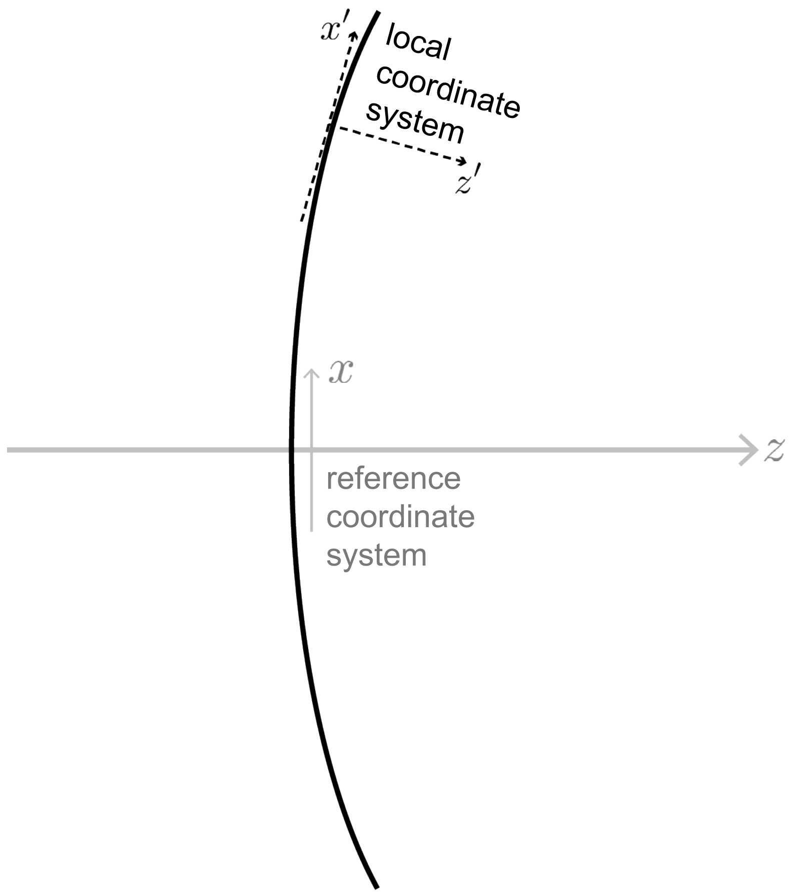
- If unit vectors of the local coordinate system can be written as

$$\hat{\boldsymbol{x}}' = a_{11}\hat{\boldsymbol{x}} + a_{12}\hat{\boldsymbol{y}} + a_{13}\hat{\boldsymbol{z}}\,,$$
$$\hat{\boldsymbol{y}}' = a_{21}\hat{\boldsymbol{x}} + a_{22}\hat{\boldsymbol{y}} + a_{23}\hat{\boldsymbol{z}}\,,$$
$$\hat{\boldsymbol{z}}' = a_{31}\hat{\boldsymbol{x}} + a_{32}\hat{\boldsymbol{y}} + a_{33}\hat{\boldsymbol{z}}\,.$$

then, we can calculate the transverse wavevector components in the local coordinate system as follows

$$\begin{pmatrix} k_x' \\ k_y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} k_x \\ k_y \end{pmatrix} + \begin{pmatrix} a_{13} \\ a_{23} \end{pmatrix} k_z\,,$$

with $k_z = \sqrt{\epsilon k^2 - ||\boldsymbol{\kappa}||^2}$, and ϵ is the permittivity of the corresponding medium.



Appendix – Coordinate System and Transformation

- For the the transverse field components, we can find the transformation rule as follows

$$\boldsymbol{U}'_{\perp} = \mathbf{Y}^{\text{loc}} \boldsymbol{U}_{\perp} \,.$$

with

$$\mathbf{Y}^{\text{loc}} = \begin{pmatrix} a_{11} - a_{13}\frac{k_x}{k_z} & a_{12} - a_{13}\frac{k_y}{k_z} \\ a_{21} - a_{23}\frac{k_x}{k_z} & a_{22} - a_{23}\frac{k_y}{k_z} \end{pmatrix} .$$

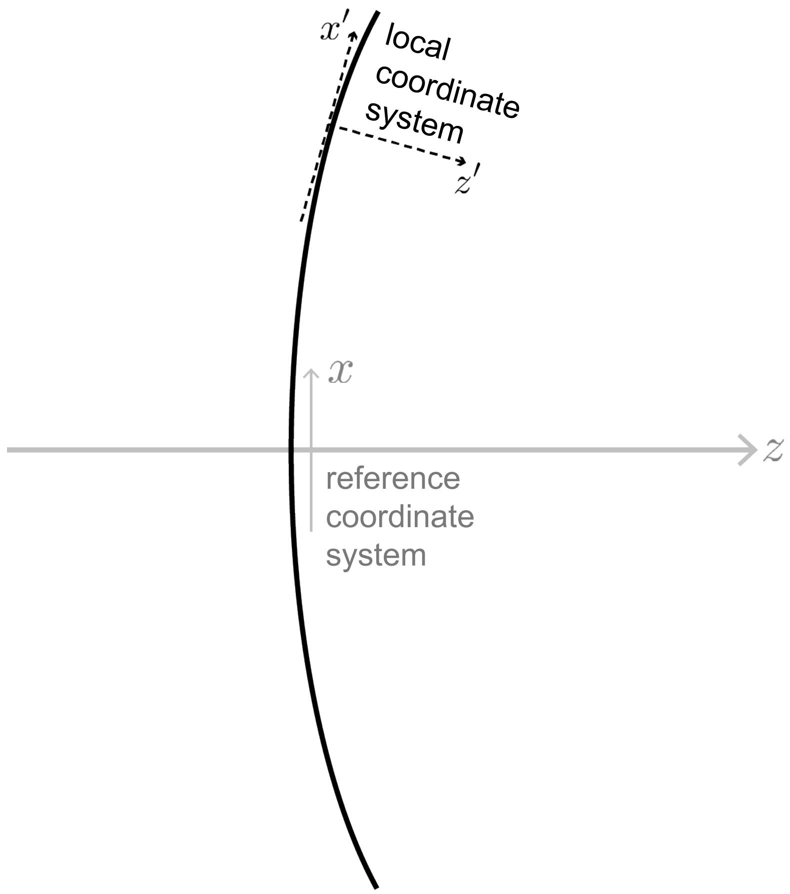
- And, the inverse transformation can be written as

$$\boldsymbol{U}_{\perp} = \mathbf{Y}^{\text{ref}} \boldsymbol{U}'_{\perp} \,.$$

with

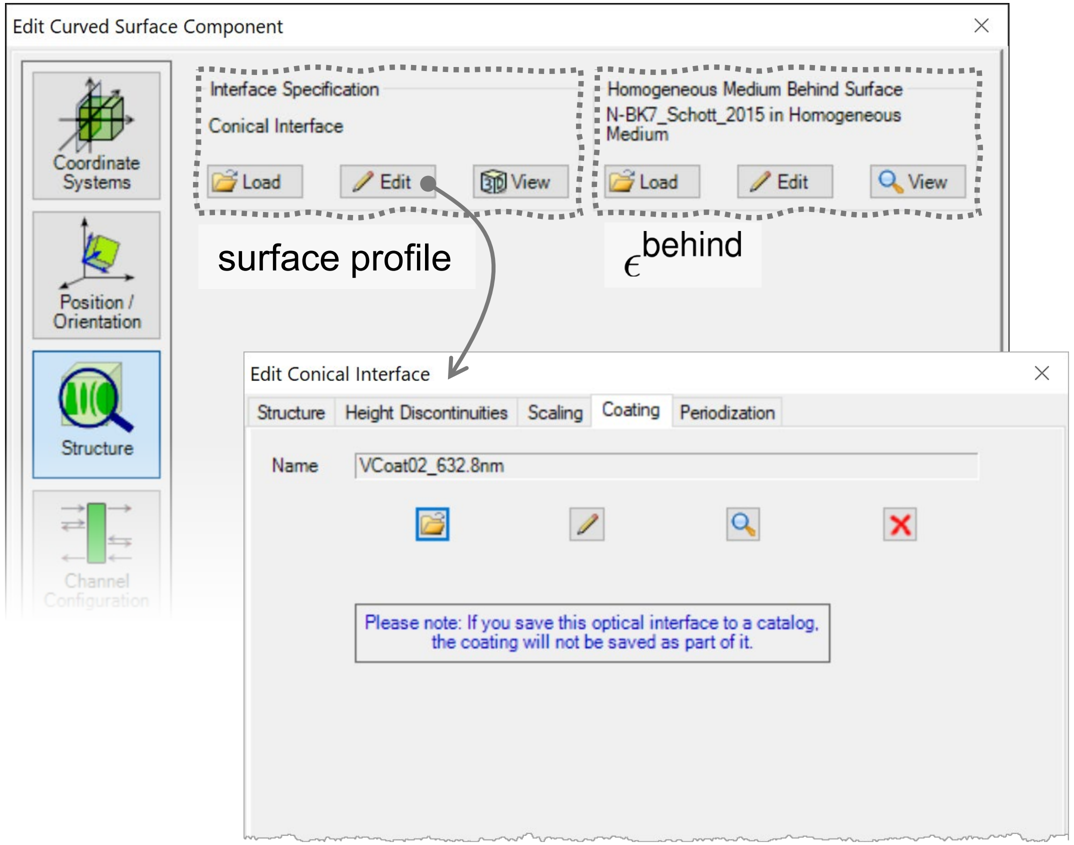
$$\mathbf{Y}^{\text{ref}} = \begin{pmatrix} a_{11} - a_{31}\frac{k'_x}{k'_z} & a_{21} - a_{31}\frac{k'_y}{k'_z} \\ a_{12} - a_{32}\frac{k'_x}{k'_z} & a_{22} - a_{32}\frac{k'_y}{k'_z} \end{pmatrix} .$$

- Details on the expressions above can be found in [2, 3].



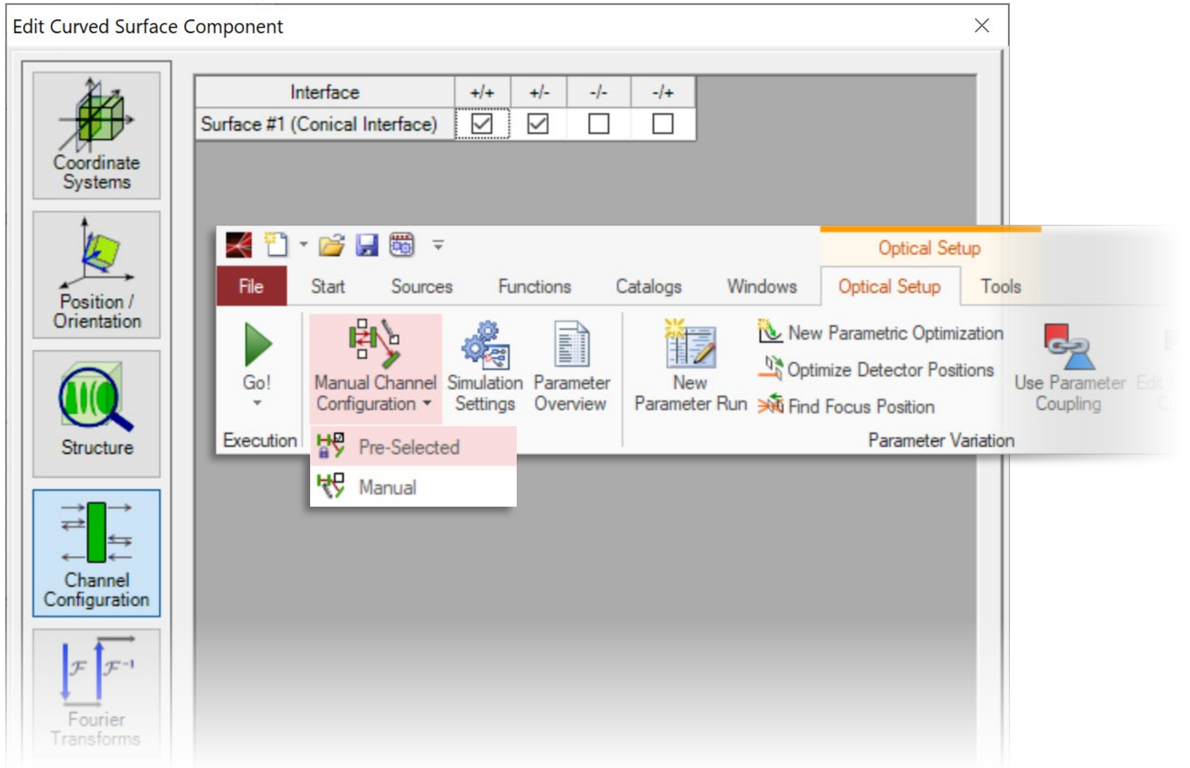
Usage in VirtualLab Fusion

- Take the Curved Surface Component as an example:
  - the **curved surface** profile is specified by an interface in the Structure tab;
  - one can define ϵ^{behind} with the homogeneous medium behind the surface, while ϵ^{front} is automatically determined by the preceding system;
  - when the **surface coating** is present, its structure can be edited under the Interface Specifications as well.



Usage in VirtualLab Fusion

- The channels of the solver can be configured for the curved surface component:
  - by default, the $+/+$ and/or $+/-$ channels are pre-selected, corresponding to the transmission and reflection when the input is along the forward direction;
  - in the manual mode, the $-/-$ and/or $-/+$ channels can be enabled, which corresponds to the transmission and reflection when the input is along the backward direction.



List of References

[1] Albrecht v. Pfeil, Frank Wyrowski, Andreas Drauschke, and Harald Aagedal, "Analysis of optical elements with the local plane-interface approximation," Appl. Opt. 39, 3304-3313 (2000)

[2] Rui Shi, Christian Hellmann, and Frank Wyrowski, "Physical-optics propagation through curved surfaces," J. Opt. Soc. Am. A 36, 1252-1260 (2019)

[3] Rui Shi and Frank Wyrowski, "Comparison of aplanatic and real lens focused spots in the framework of the local plane interface approximation," J. Opt. Soc. Am. A 36, 1801-1809 (2019)

Document Information

title	VirtualLab Fusion Technology – Local Plane Interface Approximation (LPIA)
document code	TEC.0006
version	1.0
category	Technology White Paper
further reading	- VirtualLab Fusion Technology – Fresnel Matrix - VirtualLab Fusion Technology – Layer Matrix [S-Matrix] - VirtualLab Fusion Technology – Local Linear Grating Approximation (LLGA)