

# Les simulations optiques au LAPP

----

Romain Bonnand pour le groupe Virgo



# Les codes utilisés au LAPP

- **OptoCad (prop. gaussienne, 2D) :**
  - Gestion de la partie DET/SIN/ALS de l'OptoCad de Virgo (R. Bonnand).
  - Quelques autres utilisateurs occasionnels (R. Gouaty, L. Rolland).
- **Code ABCD avec astigmatisme (Matlab, prop. Gaussienne, 2D avec astigmatisme) :**
  - Développé pour les cavités stables (R. Bonnand).
  - Permet de calculer les paramètres gaussien des faisceaux dans une cavité couplé cavité de recyclage + un bras) et de faire du tolérancement.
- **OSCAR :**
  - Plusieurs utilisateurs (V. Hui, R. Bonnand).
  - Utilisé dans le cadre des cavités stables, pour des simulation sur l'OMC ou pour le commissioning (estimation focal des CP, impact PR/SR RoC sur puissance de la frange noire, simulation d'un diaphragme devant SR, comparaison mesure réel et simulation du gain des sidebands).



# Les codes utilisés au LAPP

- **Optickle :**
  - Simulations en ondes planes avec effet de pression de radiation.
  - Utilisé pour le noise budget (M. Was).
- **Finesse 3 :**
  - Simulation modal ou en ondes planes.
  - A partir de notebook déjà construit ou d'exemple de la doc (M. Was et R. Bonnand).
  - Dans le cadre des cavités stables.
- **Autre code ad-hoc sur Matlab :**
  - E. Tournefier pour le calcul des gains (porteuse et sideband) dans l'ITF.
  - Code Ray-Tracing 3D développé sur Matlab (à l'état v0), pour calculer la séparation des faisceaux B1 et B5 (R. Bonnand).
  - Code ABCD 4x4 (à l'état v0) pour l'analyse des effets de désalignements (R. Bonnand).
  - Simulations de la chaîne de démodulation et études de bruits (électronique) (S. Sayah).



# Quelques tentatives de développement.

- Code Ray-Tracing 3D développé sur Matlab :

- Simulation de B5 suite au problème du wedge horizontale de BS, simulation du télescope du banc SPRB (séparation B4/B4p), simulation pour le Pcal pour la phase II, simulation du faisceau B5/B1 dans les cavités stables.
- Code à l'état primitif, basé sur des changements de coordonnées pour suivre un rayon en 3D.
- Pas d'affichage graphique vraiment lisible.
- Bilbio :
  - [https://graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection\\_refraction.pdf](https://graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection_refraction.pdf)
  - General Ray Tracing Procedure, Spencer & Murty <https://doi.org/10.1364/JOSA.52.000672>
  - D'autres ressources glanés sur le web.

Based on Equation 2.4, the primed direction cosines satisfy  $\alpha'^2 + \beta'^2 + \gamma'^2 = 1$  (2.53)

### 2.3.3 Summary of equations and application

We have now finished the derivation of all the translation and refraction equations for exact ray tracing. One of the attractive features of these equations is that they work for all spherical surfaces: convex ( $r$  or  $c > 0$ ), concave ( $r$  or  $c < 0$ ), and plane ( $r = \infty$  or  $c = 0$ ). The flow chart in Figure 2.19

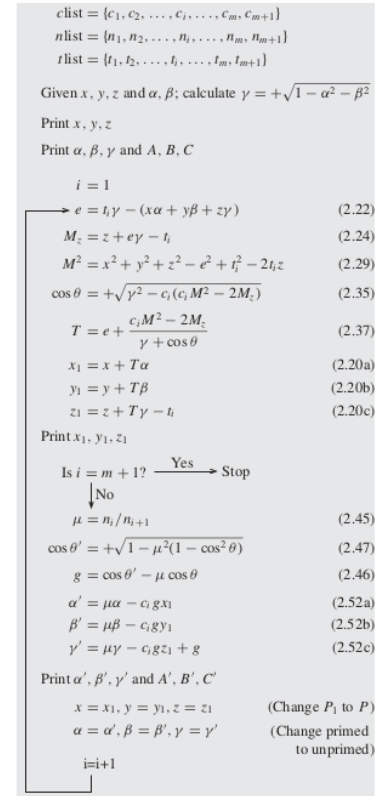


Figure 2.19 Basic flow chart for tracing skew rays.

outlines the way a computer program might be written in your favorite language to trace a ray through any number of refracting surfaces—the numbers in parentheses refer to the equation numbers where the equations were first derived in the preceding derivations. This flow chart also serves as a summary of all the important equations.

The generalized optical system that helps to describe the notation and the subscripts used in the flow chart is shown in Figure 2.20—it is drawn in two dimensions for simplicity. The notation is similar to that employed in Chapter 1, but has been simplified for use with a computer program. The last point is denoted by  $P'$ , and the last surface (here, imaginary) is plane because the last surface frequently is an image surface, which usually is a plane surface. Because of the  $-t_i$  in Equation 2.20c, the origin automatically moves to the vertex of the next refracting surface to keep the  $z$  coordinate the same order of magnitude as the  $x$  and  $y$  coordinates.

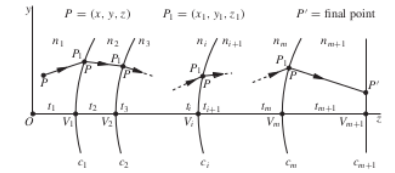


Figure 2.20 A generalized optical system to help understand the notation used in the flow chart for tracing skew rays.

**Example 2.3.1 Exact ray tracing of a skew ray.** In Figure 2.21 we show a biconvex lens, and in Figure 2.22 we give the numerical values for this system. We now want

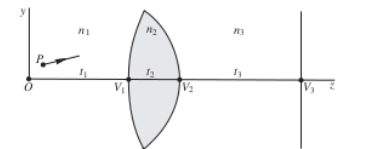


Figure 2.21

	$r$ (mm)	$n$	$t$ (mm)
$P: x = 2.00$ mm		1.000	10.00
$y = 1.00$	20.00		
$z = 2.00$	-10.00	1.500	5.00
$\alpha = 0.200$		1.000	12.00
$\beta = 0.200$	$\infty$		

Figure 2.22



# Quelques tentatives de développement.

- Code ABCD 4x4 pour suivre désalignement :
  - Développé récemment pour les cavités stables et l'étude des « requirements » des cavités stables.
  - Basé sur *Matrix methods in treating decentred optical systems*, Wang Shaomin  
<https://link.springer.com/article/10.1007/BF00619988>
  - Code à l'état primitif.

## Appendix

Augmented ray transfer matrices and their corresponding flow graphs for some misaligned elements and media.

Number	Description	Misaligned diagram	Augmented 4 × 4 ray transfer matrix	Corresponding flow graph
1	Misaligned uniform medium		$\begin{bmatrix} 1 & l & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
2	Misaligned spherical interface*		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ c_1 & n_1/n_2 & -ec_1 & e'(1-n_1/n_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
3	Misaligned plane interface		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & n_1/n_2 & 0 & e'(1-n_1/n_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
4	Misaligned planoparallel		$\begin{bmatrix} 1 & l/n & 0 & e'(1-1/n)l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	

Matrix methods in treating decentred optical systems

TABLE (contd.)

Number	Description	Misaligned diagram	Augmented 4 × 4 ray transfer matrix	Corresponding flow graph
5	Misaligned thick lens†		$\begin{bmatrix} a_1 & b_1 & a_2e & \beta_2e' \\ c_1 & d_1 & \gamma_2e & \delta_2e' \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
6	Misaligned thin lens		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/f & 1 & e/f & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
7	Misaligned wedge		$\begin{bmatrix} 1 & l/n & 0 & (1-1/n)e'_1 \\ 0 & 1 & 0 & (1-n)(e'_1-e'_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
8	Misaligned lens-like medium‡		$\begin{bmatrix} a_0 & b_0 & a_0e & \beta_0e' \\ c_0 & d_0 & \gamma_0e & \delta_0e' \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
9	Misaligned spherical mirror		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 2/R & 1 & -2e/R & -2e' \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
10	Misaligned flat mirror		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2e' \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
11	Misaligned corner-cube reflector		$\begin{bmatrix} -1 & 0 & 2e & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	
12	Misaligned PCM formed by DFWM		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	

\*  $c_1 = -(1 - n_1/n_2)/R$

†  $a_1 = 1 - (1 - 1/n)l/R_1$ ,  $b_1 = l/n$ ,  $c_1 = -[(n-1)(\frac{1}{R_1} + \frac{1}{R_2}) - (n-1)^2l/nR_1R_2]$ ,  $d_1 = 1 - (1 - 1/n)l/R_2$

‡  $\beta_0 > 0$ :  $a_0 = \cos[(2\beta_0)^{1/2}]$ ,  $b_0 = \frac{1}{2\beta_0} \sin[(2\beta_0)^{1/2}]$ ,  $c_0 = -2\beta_0 b_0$ ,  $d_0 = a_0$

$\beta_0 < 0$ :  $a_0 = \cosh[(-2\beta_0)^{1/2}]$ ,  $b_0 = \frac{1}{(-2\beta_0)^{1/2}} \sinh[(-2\beta_0)^{1/2}]$ ,  $c_0 = -2\beta_0 b_0$ ,  $d_0 = a_0$



# La suite ???

- Un optoCad 3D ?
  - Reprendre Theia qui semblait prometteur ? <https://github.com/bandang0/theia>
- Un code qui mélange ray-tracing en 3D et propagation gaussienne ?
- Un code temporel pour simuler le lock de l'interféromètre et les transients (commissioning)?
  - Il existait SIESTA dans le temps.
  - Simulation (temporel) de la TCS.
- Est ce qu'il y'a manque de code de simulations ou est ce qu'il y'a un manque d'expertise sur les différents codes ? Ou d'expert multi-code ?
- Est ce qu'on voudrait un code ultime qui simule « tout » ?
- Besoin de temps et d'expertises !

