

#### **Grazing-Incidence Focusing Mirrors for X-Ray Beams**

#### Abstract



Grazing-incidence reflective optics are widely used in X-ray beamlines, in particular Kirkpatrick-Baez (KB) elliptical mirror systems [A. Verhoeven, et al., Journal of Synchrotron Radiation 27.5 (2020): 1307-1319]. Focusing is accomplished by using two physically separated elliptical mirrors to focus the beam in two dimensions. The incoming Xrays can be focused by the system down to nanometer-scale spot size. Such system is modeled and simulated in VirtualLab Fusion and the focal field is calculated.

# **Modeling Task**

#### fundamental Gaussian



# **Analytical Design of the Elliptical Mirror (1)**



For the calculation of the elliptical surface height profile z(x) the following parameters are required:

- distance between source and the mirror center  $F_1$
- distance between image/focus and the mirror center  $F_2$
- grazing-incidence angle  $\theta$

1<sup>st</sup> elliptical mirror

- $F_1 = 50 \text{m}$
- $F_2 = 400 \text{ mm}$
- $\theta = 0.172^{\circ}$

2nd elliptical mirror

- $F_1 = 50.2 \text{m}$
- $F_2 = 200 \text{ mm}$
- $\theta = 0.172^{\circ}$

# **Analytical Design of the Elliptical Mirror (2)**



- To calculate the height function z(x), two equations need to be considered.
  - elliptical equation

$$\frac{x^{\prime 2}}{a^2} + \frac{z^{\prime 2}}{b^2} = 1$$

- coordinate transform

$$\begin{pmatrix} x'\\z' \end{pmatrix} = \begin{pmatrix} \cos\phi & -\sin\phi\\\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x\\z \end{pmatrix} + \begin{pmatrix} x_0\\z_0 \end{pmatrix}$$

• The final height function z(x) is

$$z(x) = (z' - z_0) \cos \phi - (x' - x_0) \sin \phi$$

with 
$$\phi = \arctan\left(-\frac{b^2}{a^2}\frac{x_0}{z_0}\right)$$
,  $x_0 = \frac{F_1^2 - F_2^2}{4c}$ , and  $z_0 = -b * \sqrt{1 - x_0^2/a^2}$ 

unknowns a, b, x' and z' in next slide

# **Analytical Design of the Elliptical Mirror (3)**

• *a* and *b* can be calculated from  $F_1$ ,  $F_2$  and  $\theta$ 

$$a = \frac{F_1 + F_2}{2}$$

$$c = \frac{1}{2}\sqrt{F_1^2 + F_2^2 - 2F_1F_2\cos(\pi - 2\theta)}$$

$$b = \sqrt{a^2 - c^2}$$

• parameter z' and x' are

$$z'(x') = -b\sqrt{1 - \frac{x'^2}{a^2}}, \qquad x'(x) = \frac{-n + \sqrt{n^2 - 4mt}}{2m}$$

with

$$m = \cos \phi^2 + \frac{b^2}{a^2} \sin \phi$$
$$n = -2\cos \phi \left(x + x_0 \cos \phi + z_0 \sin \phi\right)$$
$$t = (x + x_0 \cos \phi + z_0 \sin \phi)^2 - b^2 \sin^2 \phi$$



## **Energy Density & E-Field at Focal Plane**



## **Peek into VirtualLab Fusion**

Structure Height Discontinuities Scal	ing Coating	Periodization
Surface Specification		
Algorithms		
Snippet for Height Profile		Zedit Validity:
O Numerical Gradient Calculation		
User-Defined Gradient Calculatio	n	🥒 Edit Validity: 🕑
Parameters		
Angle		0.172°
F1		50 m
F2		400 mm
12		
	Source Coo	de Editor
	Source Code	e Global Parameters Snippet Help Advanced Settings
	50	********** INSERT YOUR CODE HERE ***********
lovible definition o	f 51	***************************************
	53	<pre>double a = (F1 + F2) / 2;// distance from center to</pre>
stomized interface	54	double c = Math.Sqrt(F1 * F1 + F2 * F2 - 2 * F1 * F double b = Math Sqrt(2 * 2 - 5 * c))// distance free
	56	double b = Math.sqrt(a - a - c - c);// distance fro
	57	<pre>double x0 = (F1 * F1 - F2 * F2) / (4 * c);// (x0, z</pre>
	58	double z0 = -b * Math.Sqrt(1 - x0 * x0 / (a * a));
	60	<pre>// now is to shift and rotate the coordinate the mi</pre>
	61	<pre>double phi = Math.Atan(-b * b * x0 / (a * a * z0)); double sin = Math Sin(abi);</pre>
	63	double sin = Math.Sin(phi); double cos = Math.Cos(phi);
	64	
	65	<pre>double m = b * b * sin * sin / (a * a) + cos * cos;</pre>
	66	double n = $-2 * \cos * (x + x0 * \cos + z0 * \sin);$
	68	double $x = (x + x0 - \cos + x0 - \sin)^{-1} (x + x0 - \cos - \cos - \sin)^{-1} (x + x0 - \cos - $
		<pre>double z1 = -b * Math.Sqrt(1 - x1 * x1 / (a * a));</pre>

lit Curved Surface	: Component	<
21	Basal Positioning Isolated Positioning Position Information (Absolute)	
Coordinate Systems	Position this Element's Input Axes with Respect to         Reference Element       4: Aperture         Reference Output Coordinate System       T         Relative Distance on Axis         Delta Z       0 mm	
<b>E</b> Structure	Lateral Shift         Delta X         0 mm         Delta Y         0 mm           Inclination / Rotation         0	
Solver	Orientation Definition Type Spherical Angles ✓ (iii) Z-Axis Direction Definition Angle / Axis Value Theta (Spherical) ✓ 89.828° Phi (Spherical) ✓ -90°	
Channel Configuration	Rotation About Z-Axis Z-Axis Rotation Angle 90°	

#### convinient definition of position and orientation

## **Workflow in VirtualLab Fusion**

- Set up input Gaussian field
  - Basic Source Models [Tutorial Video]
- Set the position and orientation of components
  - LPD II: Position and Orientation [Tutorial Video]
- Programmable the elliptical interface
  - How to Work with the Programmable Interface & Example (Spherical Surface) [Use Case]

mable Surface					×	
eight Discontinuities	Scaling	Coating	Periodization			
ecification						
IS						
for Height Profile			🥖 Edit	Validity: 🕑		
erical Gradient Calcula	tion					
Defined Gradient Calc	ulation		🥖 Edit	Validity: 🥑		
rs						
				0.172°	]	
				50 m		
				400 mm	]	
		Source Coo	le Editor			
		Source Code	Global Parameters	Snippet Help Advanced Settin	ngs	
		50 51 _	**********	INSERT YOUR CODE HE	RE ************************************	Aperture Diamete Aperture Diamete x (double)
		52	devilation of the			y [double]
		53	double a = (	(F1 + F2) / 2; // d1s'	tance from center to	Angle [double]
		55	double b = M	Math.Sqrt(a * a - c	<pre>* c);// distance from</pre>	F2 [double]
		57	double x0 =	(F1 * F1 - F2 * F2)	/ (4 * c):// (x0, 70	
		58 59	double z0 =	-b * Math.Sqrt(1 -	x0 * x0 / (a * a));	
		60	// now is to	shift and rotate t	he coordinate the mir	
		61	double phi =	Math.Atan(-b * b *	x0 / (a * a * z0));/	
		62	double sin =	Math.Sin(phi);		
		63	double cos =	<pre>Math.Cos(phi);</pre>		
		64				
		65	double m = b	* b * sin * sin /	(a * a) + cos * cos;	
		66	double n = -	2 * cos * (x + x0 *	cos + z0 * sin);	
		67	double t = (	x + x0 * cos + z0 *	sin) * (x + x0 * cos	
			double x1 =	(-n + Math.Sqrt(n *	n - 4 * m * t)) / (2	
			double z1 =	-b * Math.Sqrt(1 - :	x1 * x1 / (a * a));	

Edit Program

Surface Sp Algorithr Snippet

Num
 User
 Paramete
 Angle

F1 F2

### **VirtualLab Fusion Technologies**





title	Grazing-Incidence Focusing Mirrors for X-Ray Beams
document code	XRAY.0004
document version	2.0
software edition	VirtualLab Fusion Basic
software version	2021.1 (Build 1.180)
category	Application Use Case
further reading	- Single Grating Interferometer for X-Ray Imaging