Table of Contents

Designing Fresnel zone plates using formulas in **BEAMER** 2
Introduction 3
Process Flow 3
Import module 4
Loop module 4
Transform Module 5
P-XOR 6
Results 7
Using Variables 8
Additional Functions 8
Functions for specific modules: 9
Summary 11

Designing Fresnel zone plates using formulas in **BEAMER**

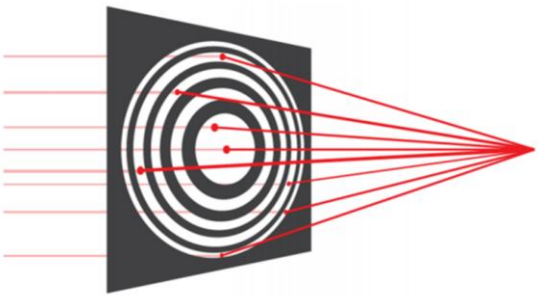
Manual design of layouts is valuable for designs with few elements and for special shapes, as irregular polygons; however, for regular shapes that need to cover large areas, the use of equations increases the precision of a pattern and speeds up the design process. Additionally, only a mathematical description translates into a specific optical performance of an optical element. This application note describes the flexibility of **BEAMER** to use equations within its modules, which allows designing and processing of layouts, for instance, Frensel zone plates.

Introduction

Fresnel zone plates are diffractive optical elements that consist of several radially symmetric rings called zones (http://zoneplate.lbl.gov/theory). The zones alternate between opaque and transparent regions as such as the light transmitted through the transparent zones constructively interferes at the desired focus. The thickness of the zones is not constant; they are narrowed towards the outer area of the plate and are described by:

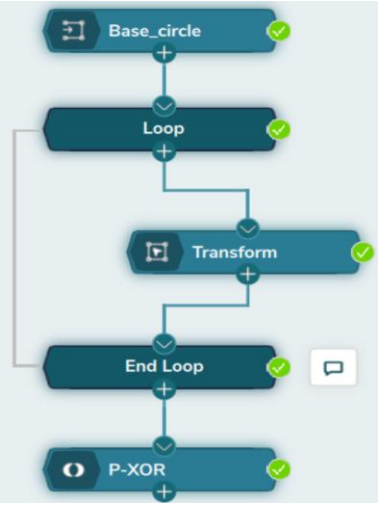
$$r_n = \sqrt{n\lambda f + \frac{n^2\lambda^2}{4}} \dots (1)$$

where *n* is the zone order, λ is the wavelength of light, and f is the focus of the plate.



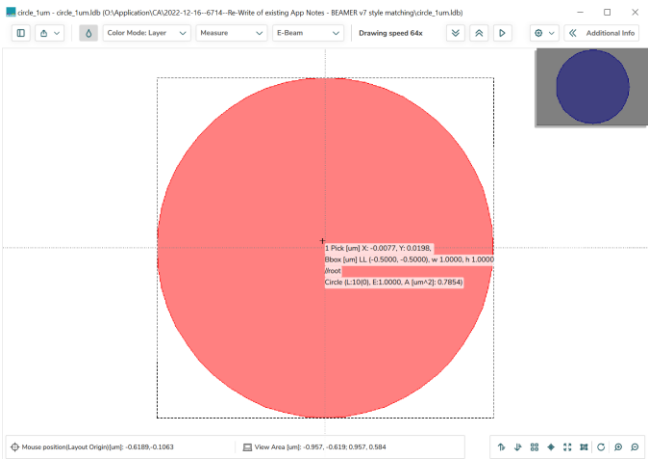
Process Flow

The next figure depicts the process flow in **BEAMER** to design a Fresnel zone plate (FZP) using formulas in modules. During this app note we want to design a FZP with a focal length of 300 µm and with a central wavelength at 139 nm. For this, in brief, a circle will be used as a base shape, its radius will increase following eq. 1, and finally, even-zones will be removed to get the final FZP.



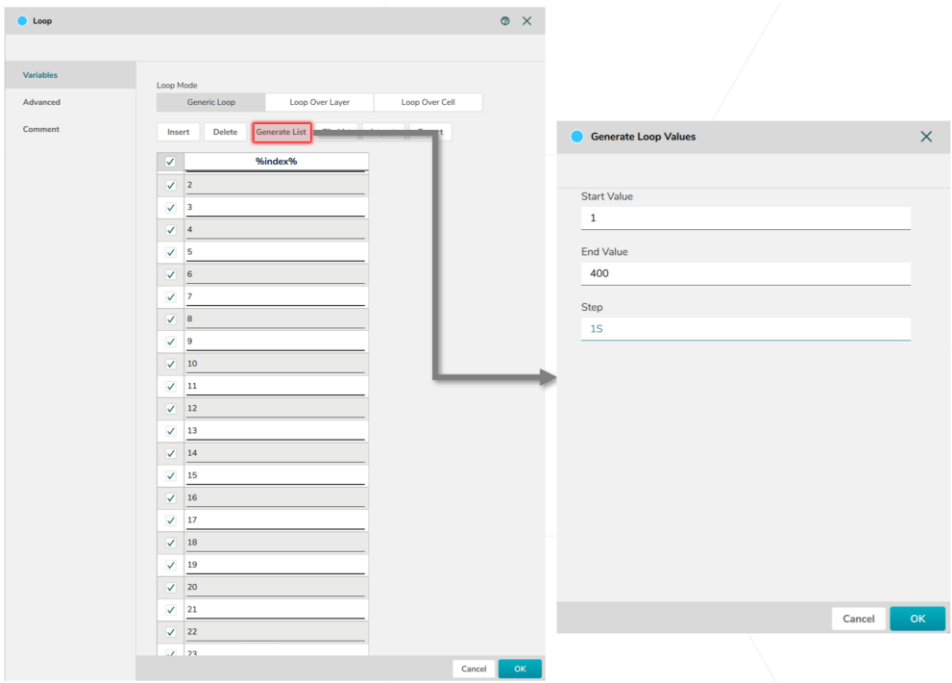
Import module

Here the file is imported. The input file is a circle of diameter 1µm.

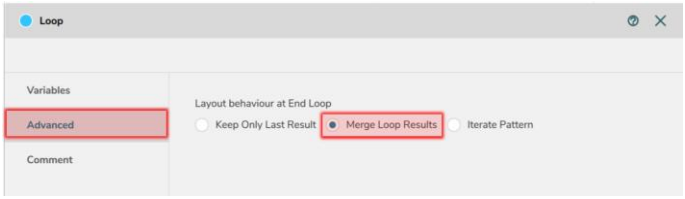


Loop module

For a given focus f and wavelength λ, eq. 1 only varies with n; therefore, the *Loop* module defines a cycle where n is a counter to set the order of the plate zones. In the module, we change the default %VarName% to %index% and use the *Generate List* to create the values that n will take during the operation. In this example, the starting point is 1 and increases in steps of 1 till 400.



In the *Advanced* tab activate the *Merge Loop Results* button to store all the results of the module.



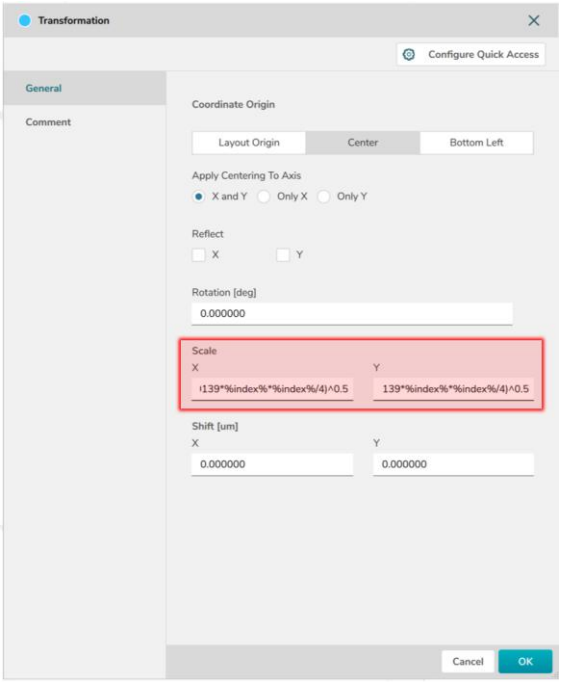
Transform Module

The next figure shows the *Transform* module in which a formula is used in the X- and Y-scaling. For zone plates, the formula for the radius of the n^{th} ring is as in eq. (1); therefore, the formula in the *Transform* module is:

2*(%index%*0.0139*300+0.0139*0.0139*%index%*%index%/4)^0.5

Notes:

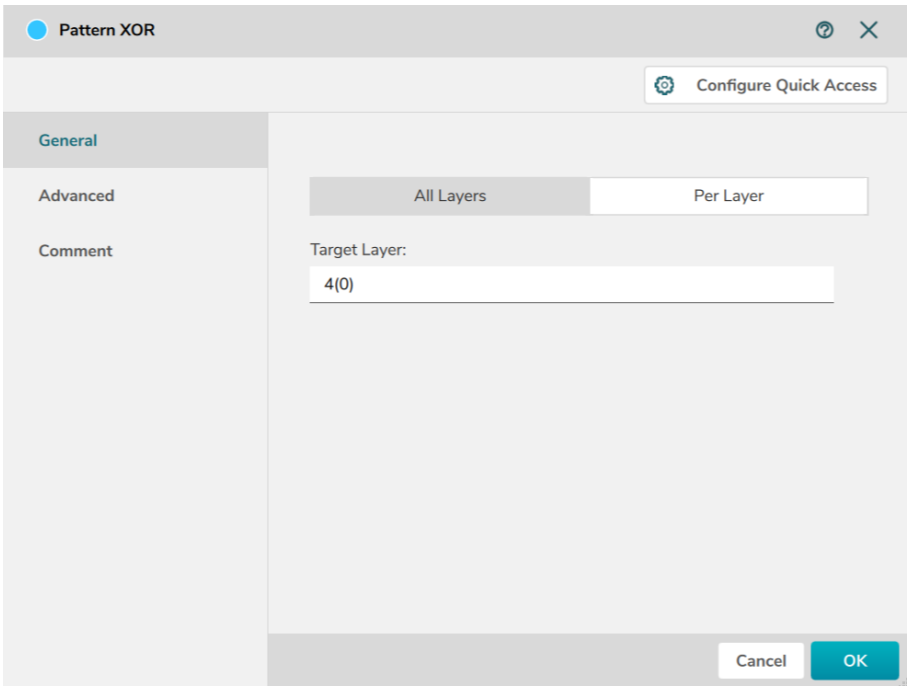
- The initial factor 2 is due to the P-XOR module which deletes every second ring.
- *n* is represented by %index% which was introduced in the *Loop* module.
- 2 * index gives the number of rings in the zone plate structure. The number of rings defines the resolution of the zone plate.
- 0.0139 is the wavelength in µm of the illuminating light.
- 300 is the focal length and is used to obtain the required diameter of the innermost ring and therefore all successive rings.



P-XOR

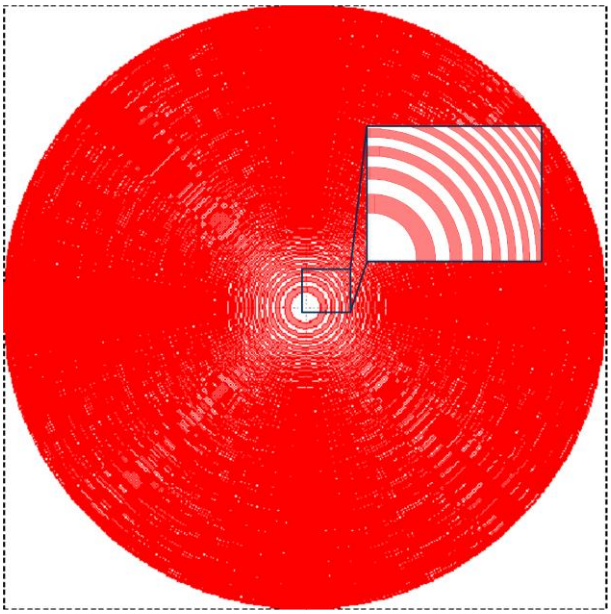
The P-XOR (Pattern-XOR) is a special Boolean operation in **BEAMER** that removes even overlaps and keeps odd overlaps. In this case, the inner ring and all alternate rings are removed resulting in the required zone plate structure.

The target layer and datatype in the result can be specified by opening the parameter window by double clicking the module.



Results

The next figure shows the final Fresnel Zone Plate for a wavelength of 139 nm, with focal length of 300 µm and 400 zone orders design using **BEAMER**.



The beauty and elegance of this example is that with just four basic **BEAMER** modules: *Import*, *Loop*, *Transform* and *P-XOR*, and using the appropriate equation in the *Transform* module, complex optical structures can be simply and quickly produced.

Using Variables

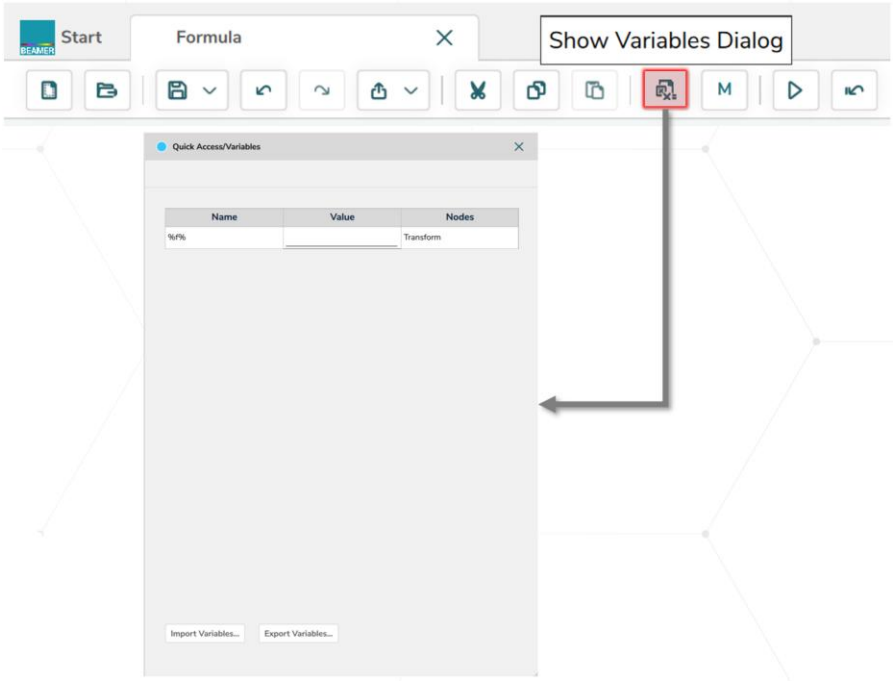
In the above example we used the formula:

2*(%index%*0.0139*300+0.0139*0.0139*%index%*%index%/4)^0.5

as the X - and Y - scaling factor in the TRANSFORM module. We could replace the fixed value of 300 for the focal length with the variable %f% as shown in red below:

2*(%index%*0.0139*%f%+0.0139*0.0139*%index%*%index%/4)^0.5

Using the *Variables* button on the tool bar opens a menu where the value for the focal length can be entered. Rerunning the flow then gives the desired results.



Additional Functions

Below is a list of additional functions that may be used in **BEAMER** modules:

- Basic mathematical operators : +, -, *, /, ^ (power function)
- Special functions:

1) SINH,COSH,TANH,COTH,E,PI,LN,LOG,SIN,
2) COS,TAN,COT,ASIN,ACOS,ATAN,ACOT,SQRT,INT
3) SINH(x) = (exp(x)-exp(-x))/2
4) COSH(x) = (exp(x)+exp(-x))/2
5) TANH(x) = SINH(x)/COSH(x)

6) COTH(x) = COSH(x)/SINH(x)
7) E: ~ 2.718
8) PI: π
9) LN: the natural logarithm
10) LOG: the common logarithm
11) SIN : sine function
12) COS : cosine function
13) TAN : tangent function TAN(x)= SIN(x)/COS(x)
14) COT: cotangent function COT(x)=1/TAN(x)
15) ASIN: arcsine function
16) ACOS: arccosine function
17) ATAN: arctangent function
18) ACOT: arccotangent function
19) SQRT: square root function
20) INT: floor function

Functions for specific modules:

1. Extract
    - Math formula is accepted for region definition.
    - ASIN, ACOS, ATAN, ACOT are not accepted due to the data type.
2. Filter
    - Math formula is accepted for “value” assignment.
    - ASIN, ACOS, ATAN, ACOT are not accepted due to the data type.
3. Transform
    - Math formula is accepted for “scale”, “shift” and “rotation”.
    - ASIN, ACOS, ATAN, ACOT are not accepted due to the data type.
4. Grid
    - Math formula is accepted for “database grid” and “layout smoothing tolerance”.
    - ASIN, ACOS, ATAN, ACOT are not accepted due to the data type.
5. Fracture
    - Math formula is accepted for “resolution” setting.
6. Heal, Not, P-Xor, And, Minus, Xor
    - Math formula is accepted for “Soft frame” setting.

21) For the present, the module “if” has a special parser which can be found by the link:

http://www.partow.net/programming/exprtk/

- to **BEAMER,** this can only be done with **BEAMER**.

Summary

BEAMER enables the generation of Fresnel zone plates through mathematical formulas implemented in its processing modules. The design workflow combines a Loop module, a Transform module, and a Boolean operation. The Loop module generates successive zone indices, the Transform module calculates the corresponding dimensions using the zone-radius equation, and the P-XOR operation removes alternating rings to produce the final zone plate pattern.

As an example, the application note demonstrates the design of a Fresnel zone plate with a focal length of 300 µm for an illumination wavelength of 139 nm. The wavelength and focal length are incorporated into the transformation formula, while the loop index determines the successive zone radii. The number of generated rings controls the overall pattern extent and influences the achievable feature resolution.

This formula-based approach provides a precise and efficient alternative to manual layout construction, particularly for complex patterns containing large numbers of regularly arranged features. By modifying the wavelength, focal length, and number of zones, the design can be adapted to different optical requirements while maintaining a consistent mathematical relationship between the individual zones. The resulting workflow demonstrates how parameterised geometry generation in **BEAMER** can simplify the design and processing of Fresnel zone plates for micro- and nanofabrication applications.