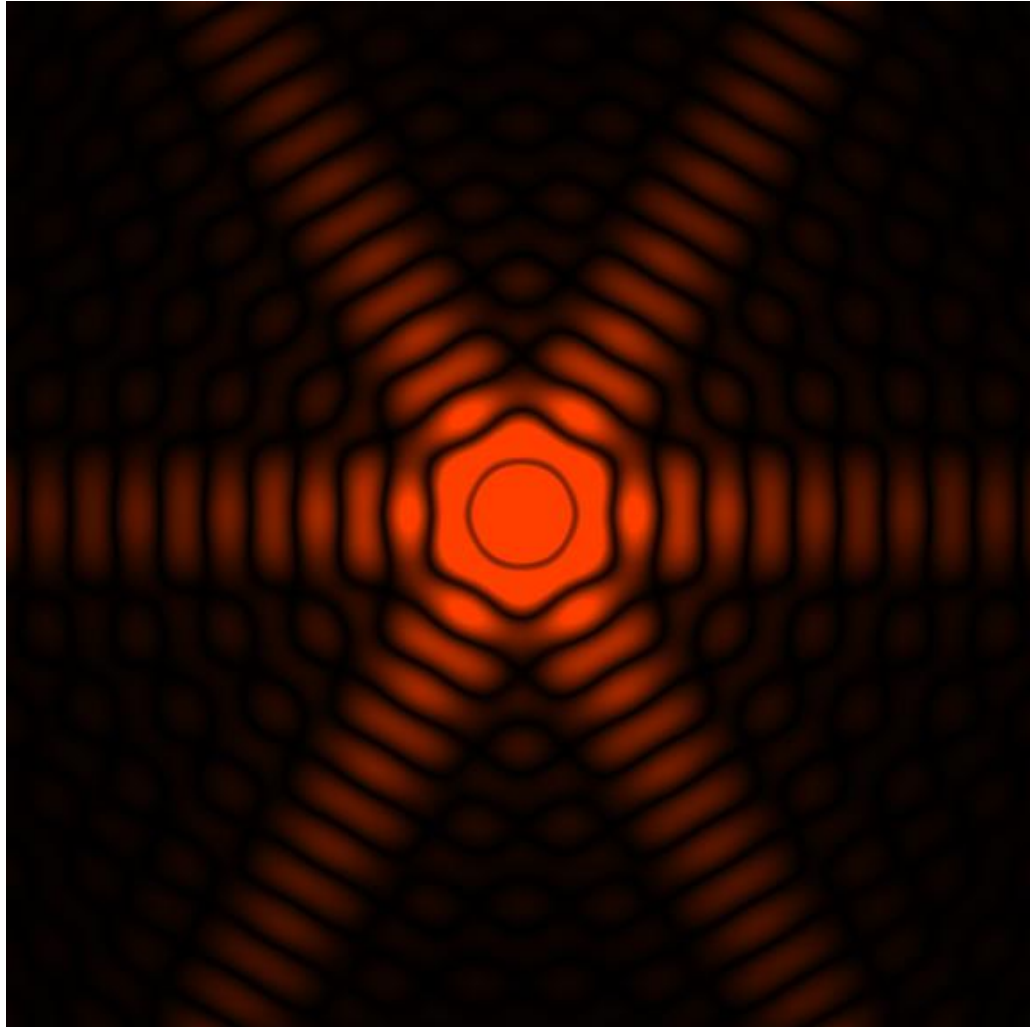


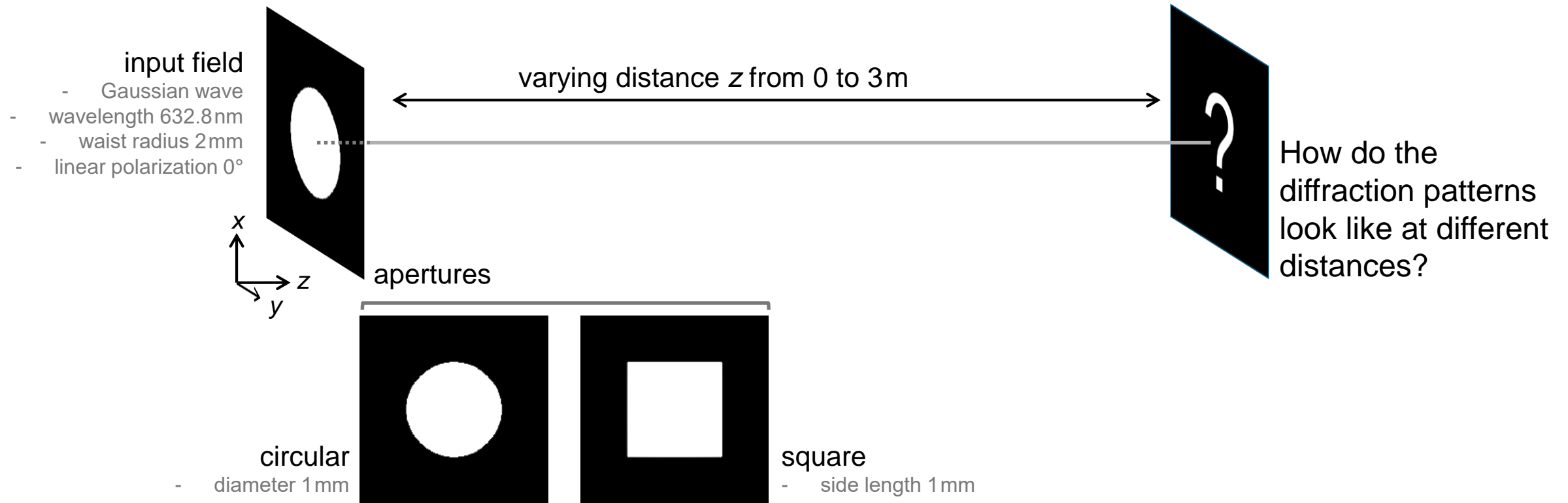
Diffraction Patterns behind Different Apertures

Abstract



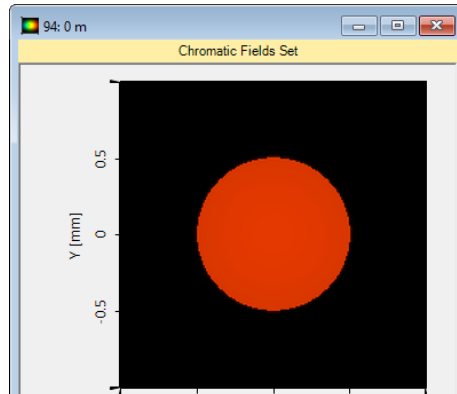
As one of the most well-known phenomena in physical optics, diffraction plays a role in various cases. VirtualLab Fusion, with its advanced propagation technologies, can handle diffraction effects in optical systems automatically. In this example, we selected some regular apertures, such as circular (or elliptical) and square (or rectangular), as well as apertures in other shapes, like pentagon or hexagon ones. The diffraction patterns from them are calculated and the property of diffraction is studied.

Modeling Task for Symmetric Apertures

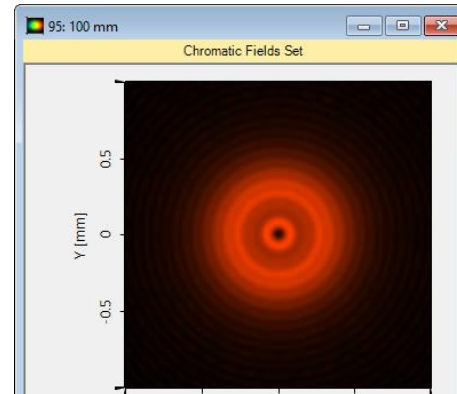


Fields after Symmetric Apertures

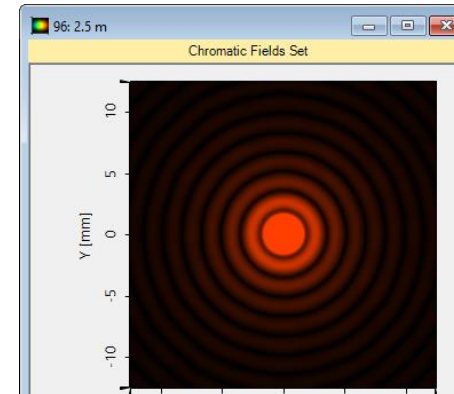
field behind aperture



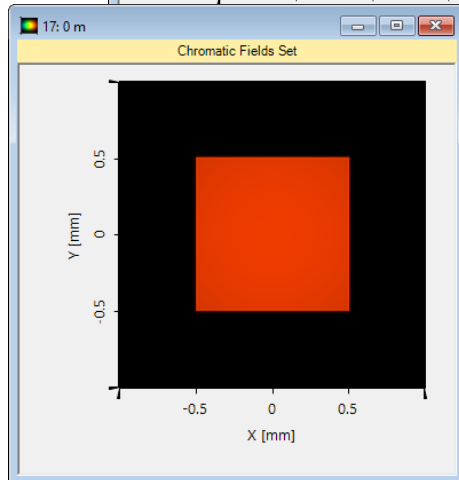
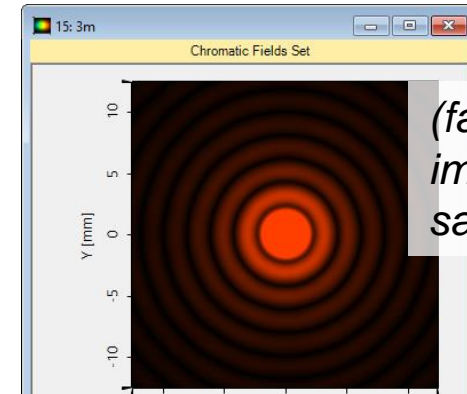
near-field



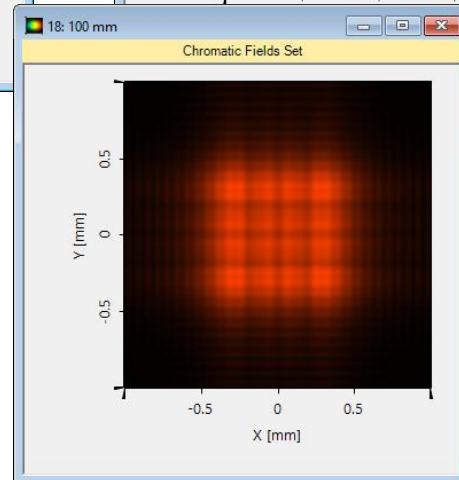
far-field



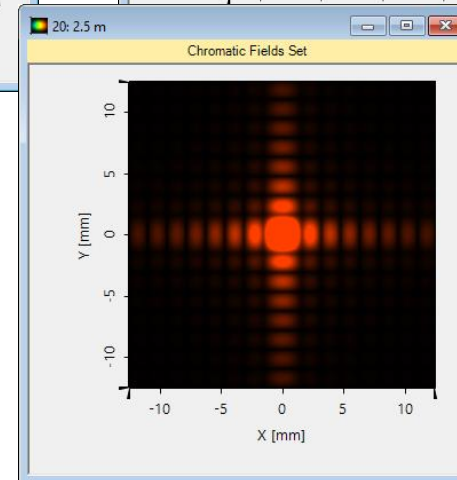
(far-field result
images are
saturated.)


$$z=0\text{ m}$$

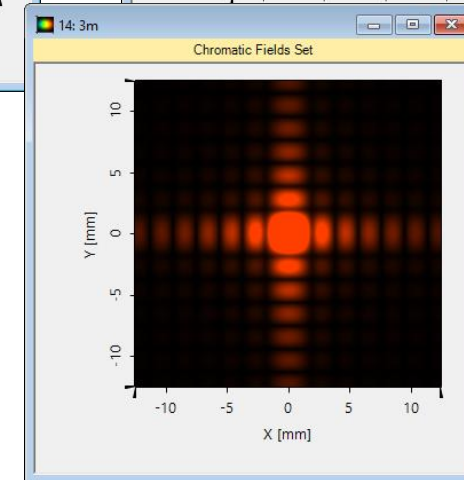
window size 2mm x 2mm

 $z=100\text{mm}$

window size 2mm x 2mm

 $z=2.5\text{m}$

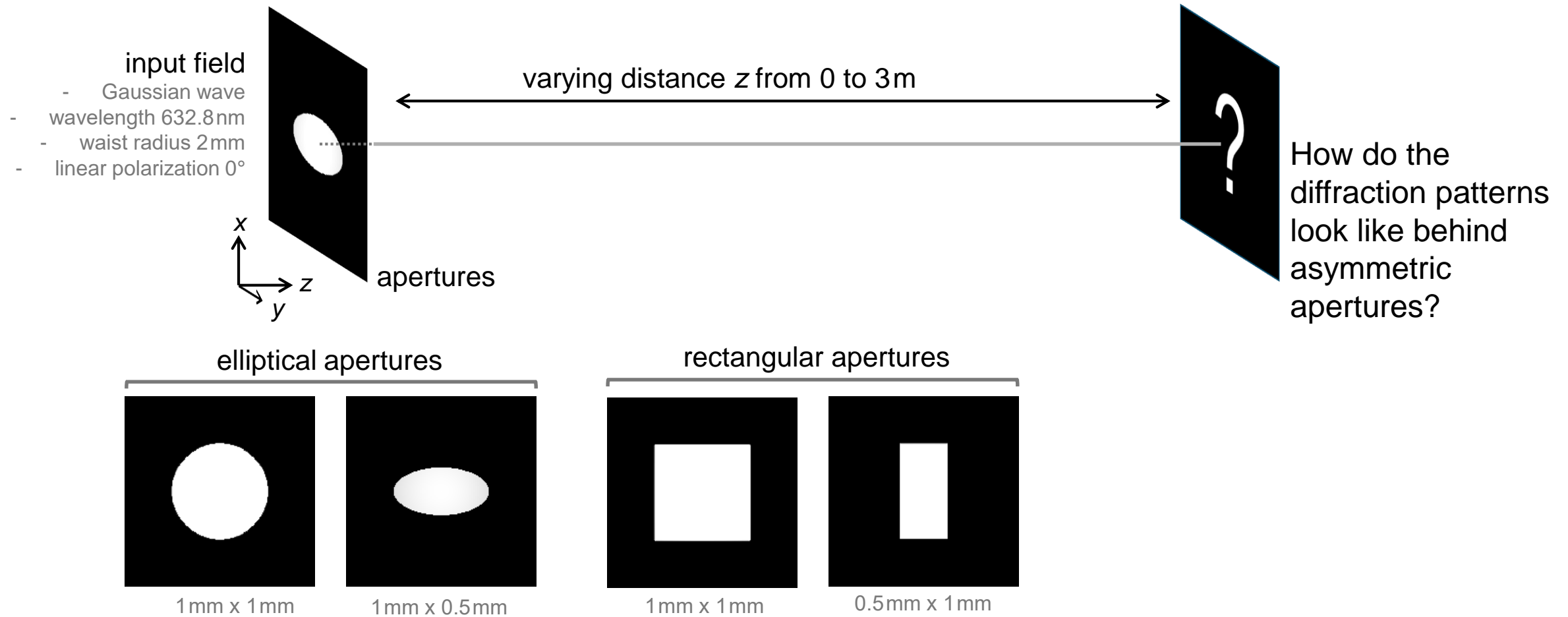
window size 25mm x 25mm

 $z=3\text{m}$

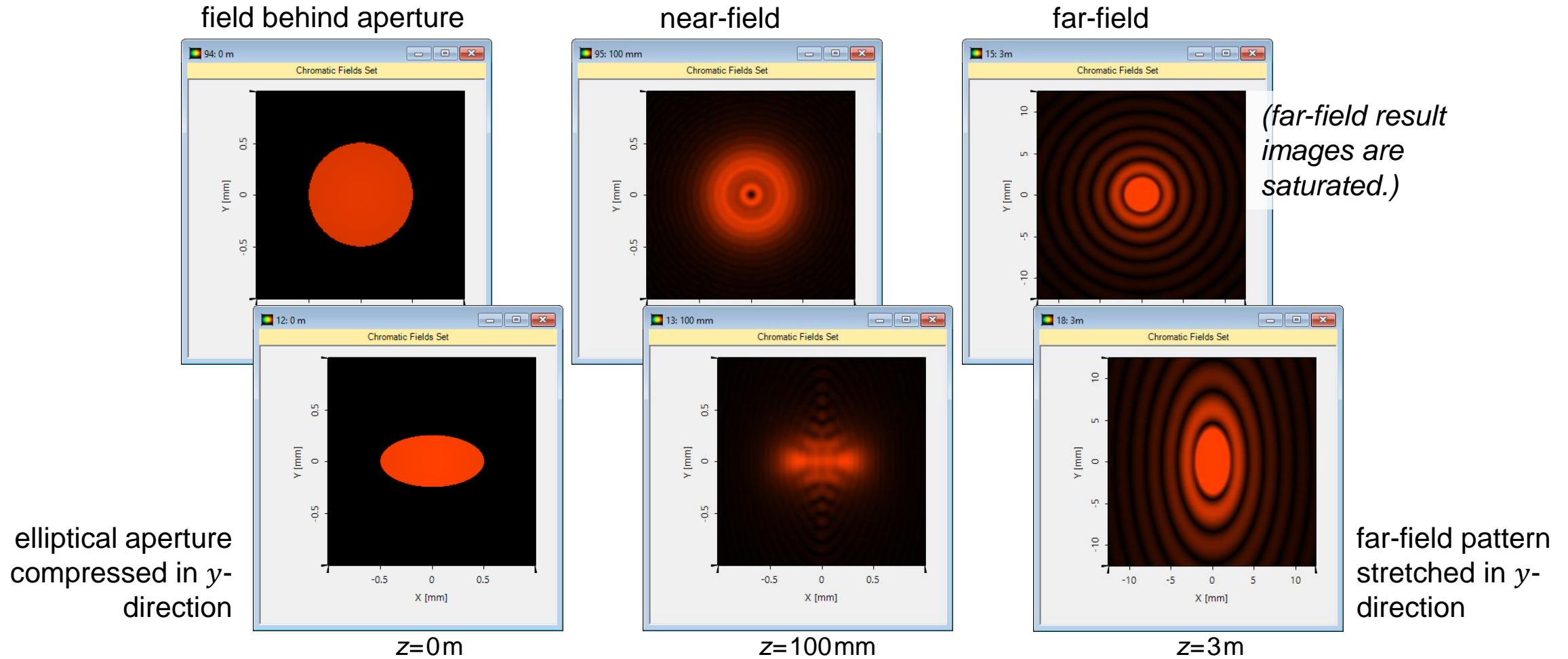
window size 25mm x 25mm

The far-field pattern changes only in size, but the profile keeps.

Modeling Task for Asymmetric Apertures

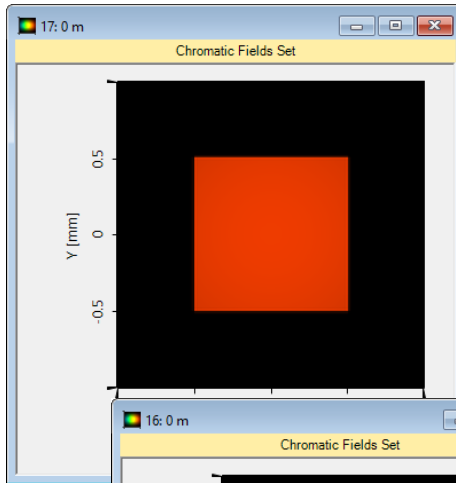


Fields after Elliptical Apertures

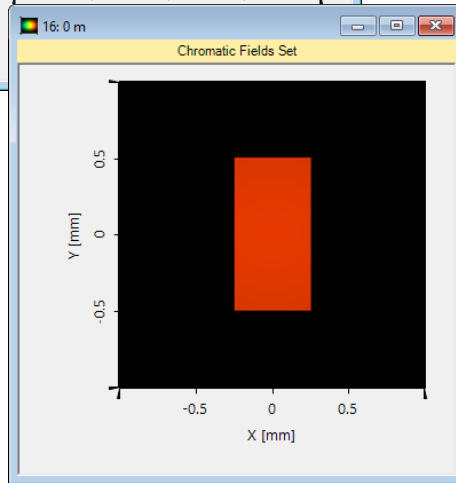


Fields after Rectangular Apertures

field behind aperture

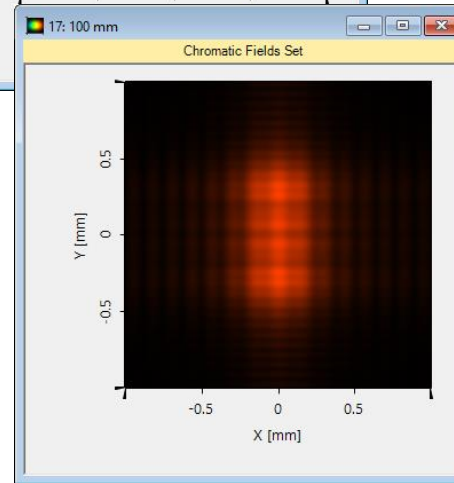
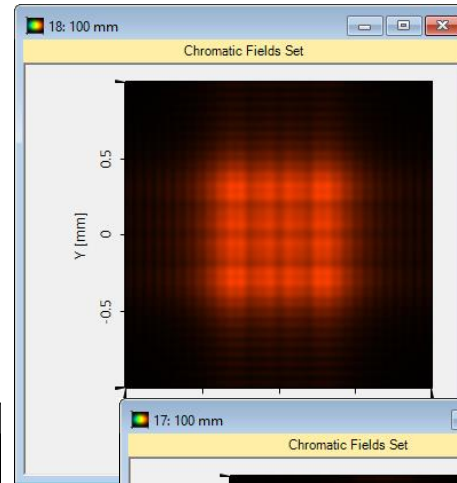


rectangular aperture
compressed in x -
direction



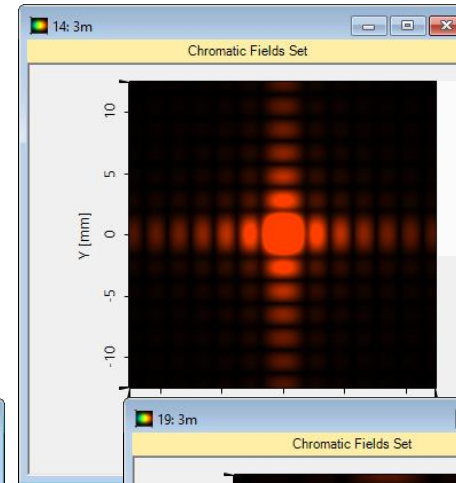
$z=0$ m

near-field

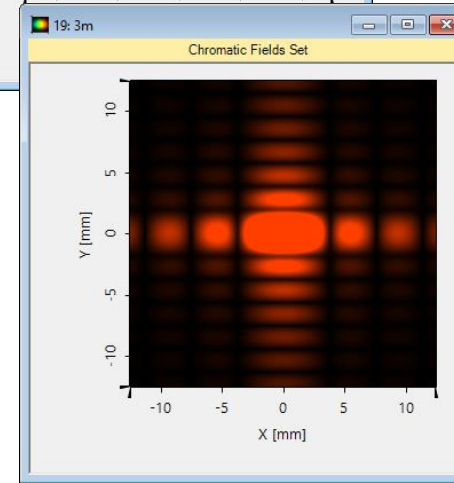


$z=100$ mm

far-field



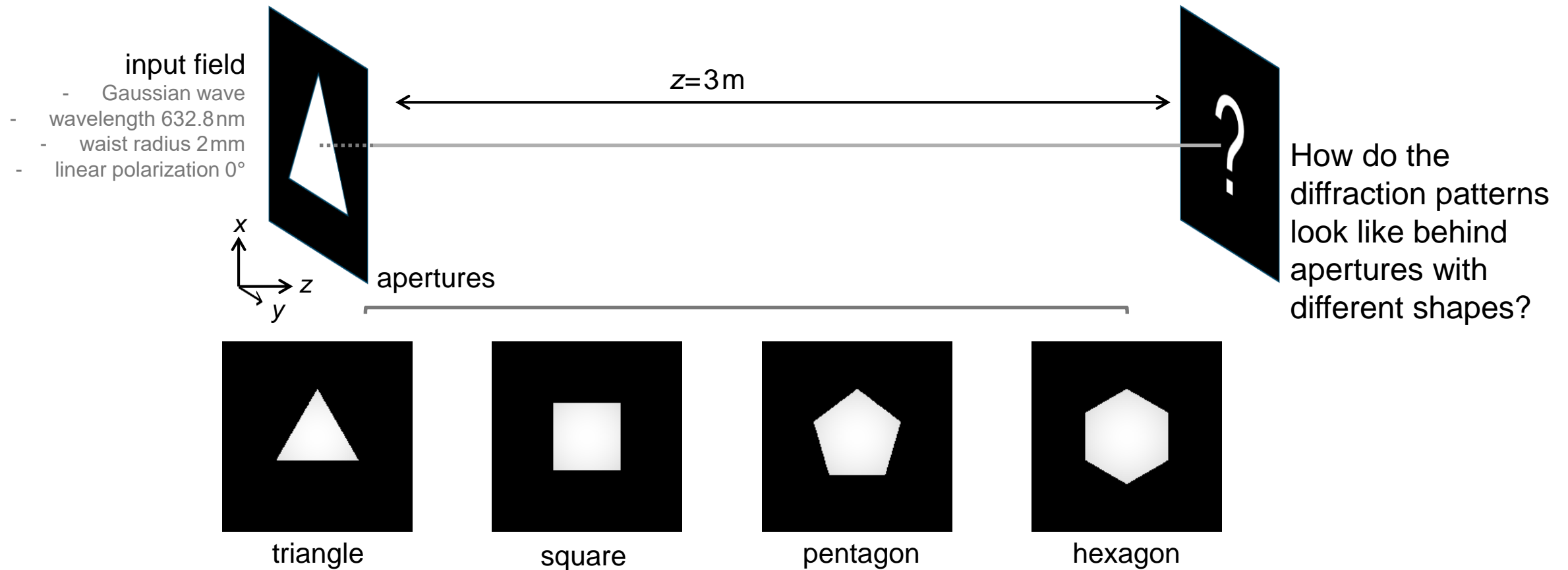
(far-field result
images are
saturated.)



far-field pattern
stretched in x -
direction

$z=3$ m

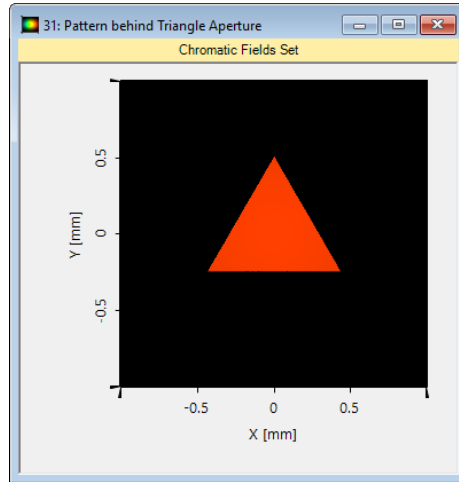
Modeling Task for Polygonal Apertures



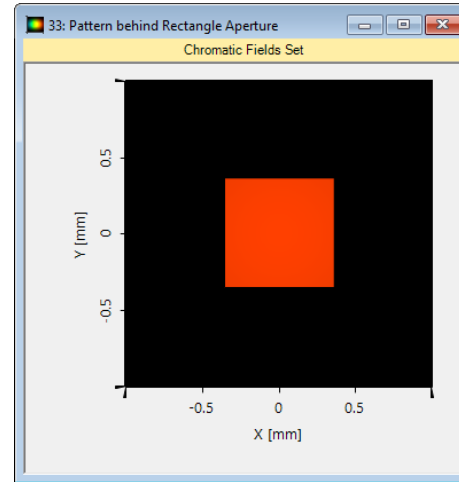
Fields after Polygonal Apertures

field behind aperture

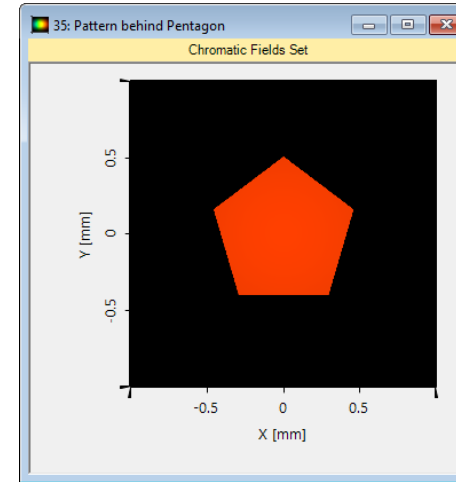
triangle



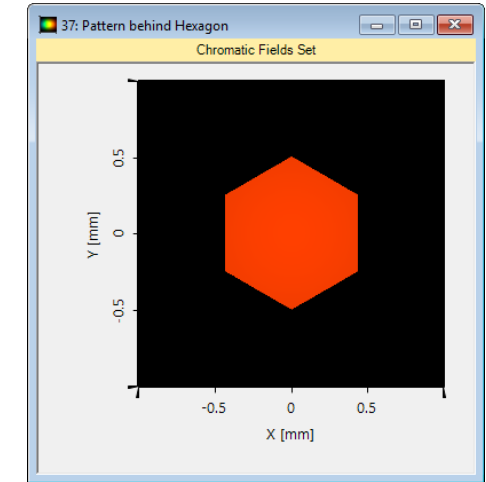
square



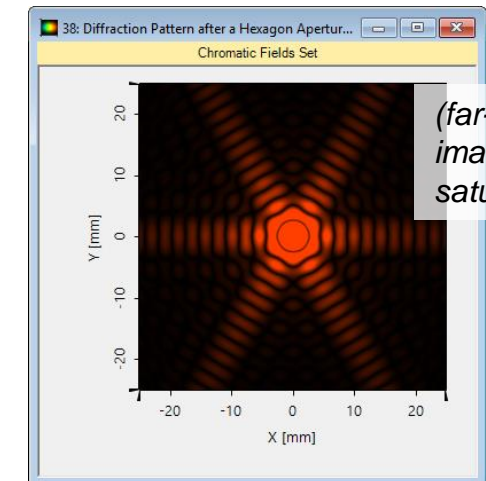
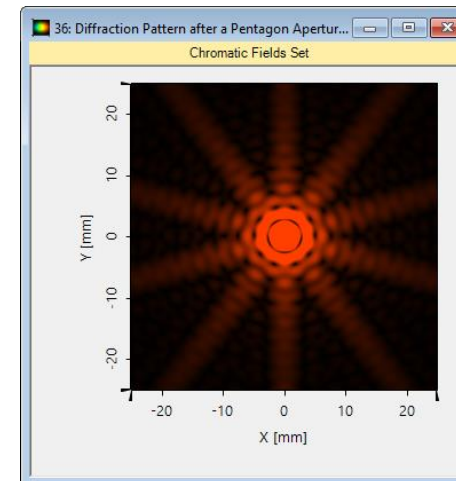
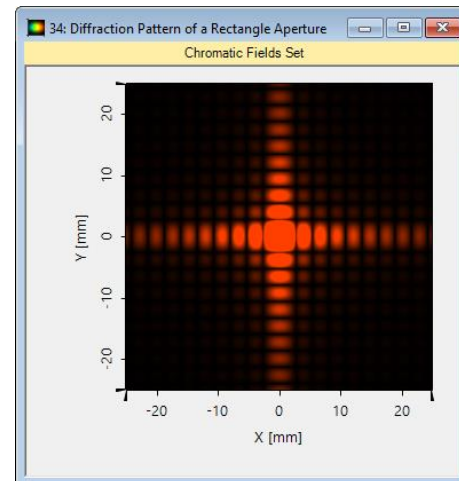
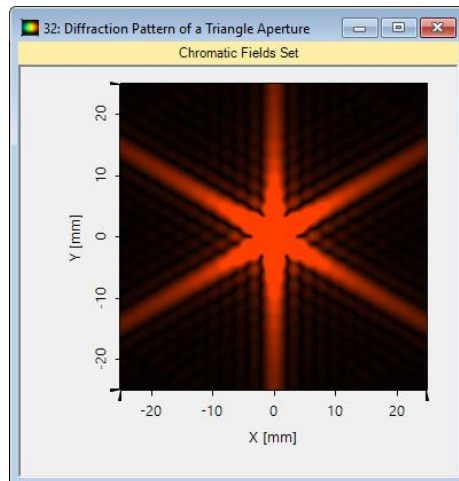
pentagon



hexagon

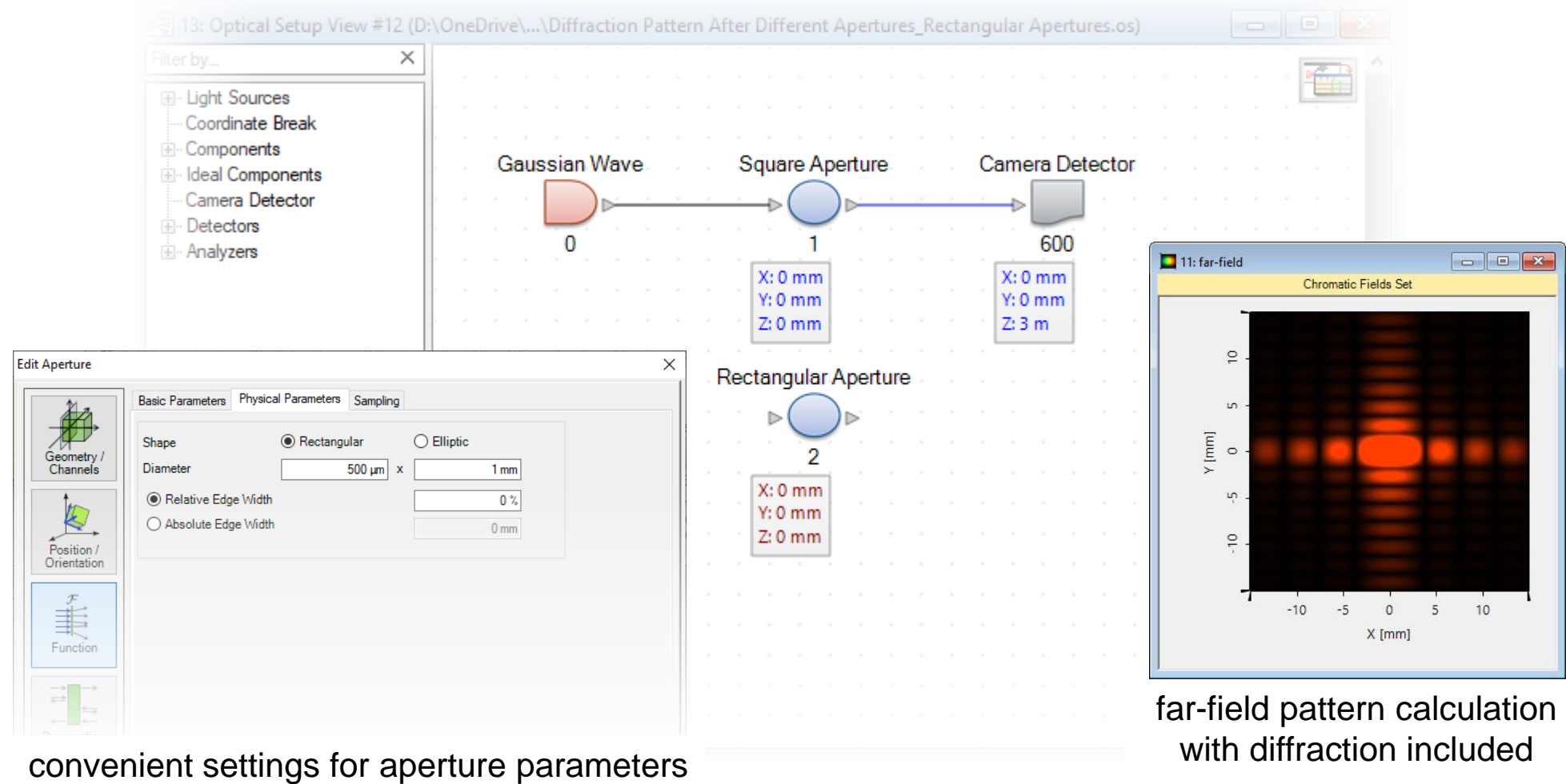


far-field



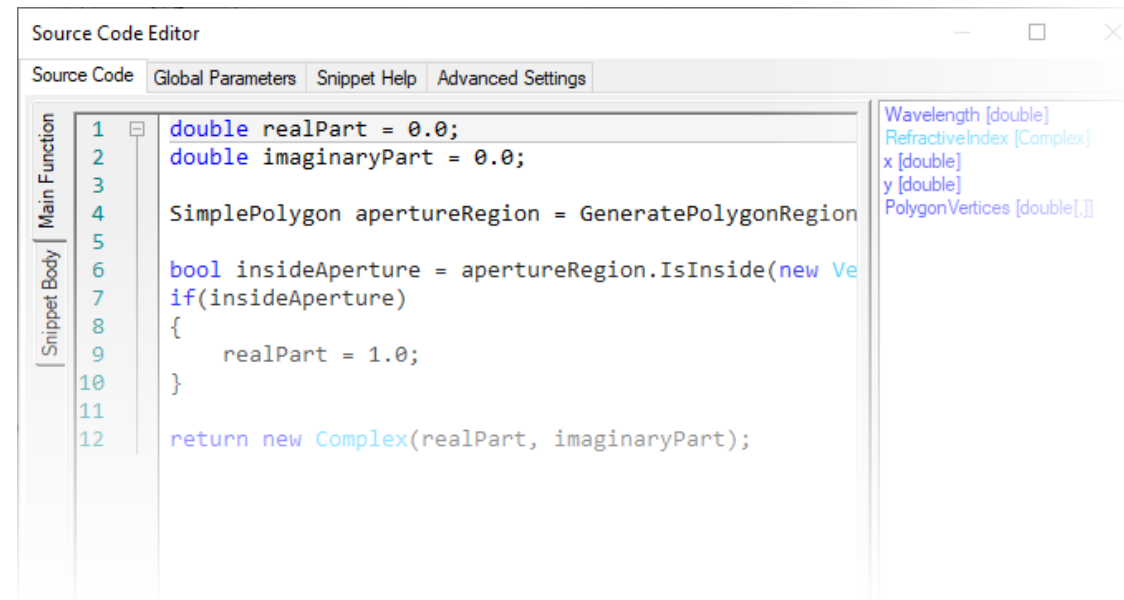
(far-field result images are saturated.)

Peek into VirtualLab Fusion



Workflow in VirtualLab Fusion

- Configure the Camera Detector
 - [Usage of Camera Detector](#) [Use Case]
- Specify or customize transmission functions
 - [How to Work with the Programmable Function & Example \(Cylindrical Lens\)](#) [Use Case]



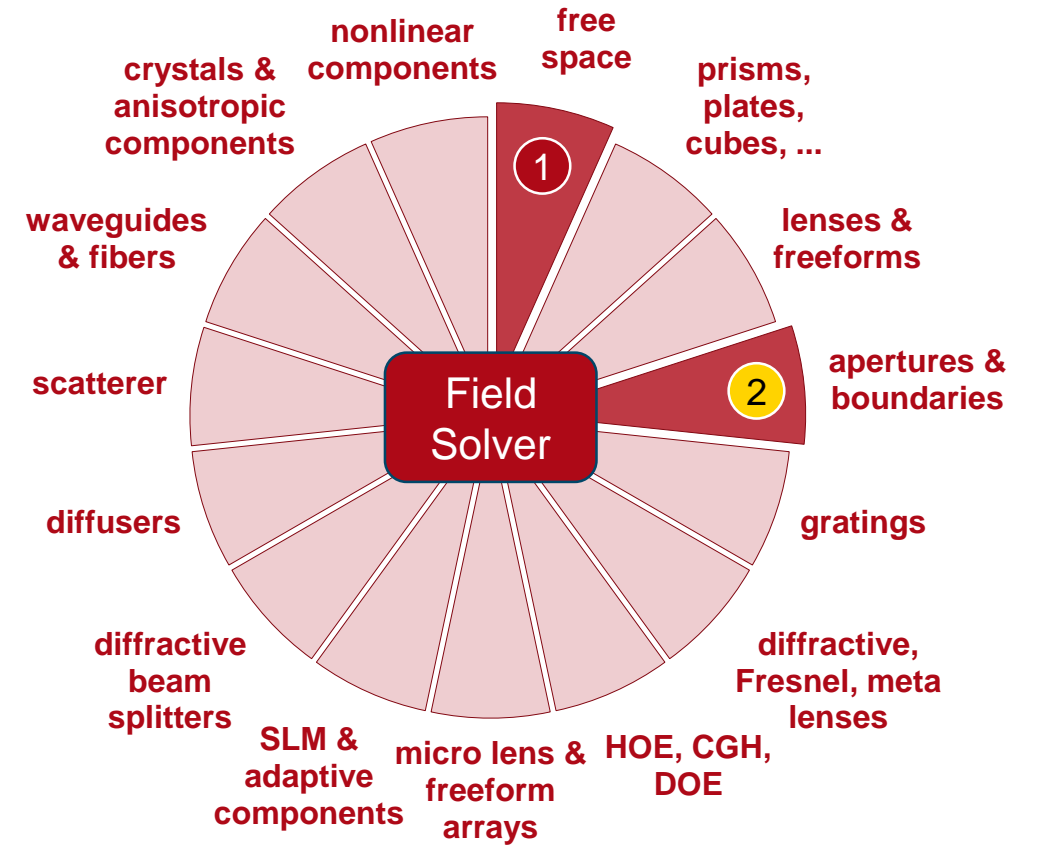
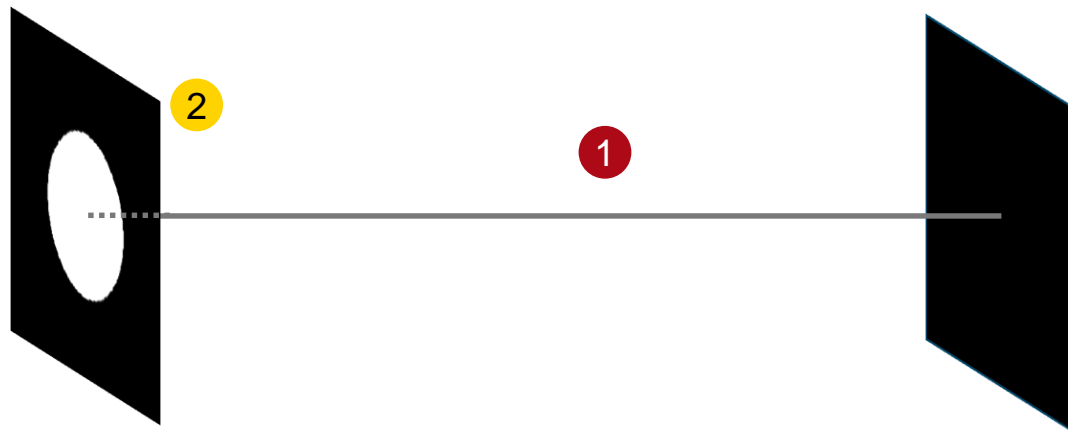
The screenshot shows the 'Source Code Editor' window in VirtualLab Fusion. The editor has tabs for 'Source Code', 'Global Parameters', 'Snippet Help', and 'Advanced Settings'. The 'Source Code' tab is active, displaying a C# script. The script defines a function that calculates the transmission function for a cylindrical lens. It starts by initializing 'realPart' and 'imaginaryPart' to 0.0. Then, it creates a 'SimplePolygon' object 'apertureRegion' using 'GeneratePolygonRegion'. Next, it checks if the current point (x, y) is inside the aperture region using 'apertureRegion.IsInside(new Vector2(x, y))'. If it is inside, 'realPart' is set to 1.0. Finally, it returns a new 'Complex' object with 'realPart' and 'imaginaryPart'.

```
1 double realPart = 0.0;
2 double imaginaryPart = 0.0;
3
4 SimplePolygon apertureRegion = GeneratePolygonRegion
5
6 bool insideAperture = apertureRegion.IsInside(new Vector2(x, y));
7 if(insideAperture)
8 {
9     realPart = 1.0;
10 }
11
12 return new Complex(realPart, imaginaryPart);
```

On the right side of the editor, there is a list of variables and their types:

- Wavelength [double]
- RefractiveIndex [Complex]
- x [double]
- y [double]
- PolygonVertices [double[,]]

VirtualLab Fusion Technologies



idealized component

Document Information

| | |
|---------------------------------|---|
| title | Diffraction Patterns behind Different Apertures |
| document code | MISC.0008 |
| version | 1.0 |
| toolbox(es) | Starter Toolbox |
| VL version used for simulations | VirtualLab Fusion Summer Release 2019 (7.6.1.18) |
| category | Application Use Case |
| further reading | <ul style="list-style-type: none">- Observation of Poisson Spot- Advanced PSF & MTF Calculation for System with Rectangular Aperture- Focal Spots for Different Aberrations |