

GaussianOptics tutorial

This is a tutorial of a *Mathematica* package `GaussianOptics``.

■ Preparation

Getting started

First, you have to import the package to your kernel. To do so just type in

```
In[3]:= << GaussianOptics`
```

Now you can see that the context "GaussianOptics`" is added to your `$ContextPath`.

```
In[4]:= $ContextPath
```

```
Out[4]= {GaussianOptics`, GOTutorial`, Global`, System`}
```

■ Construct a beam path

The basic object of `GaussianOptics` package is a beam path.

A beam path is just a list of optical components.

Here is an example of a beam path.

```
In[5]:= BP = {FreeSpace[1], Lens[0.1]}
```

```
Out[5]= {{FreeSpace, 1}, {Lens, 0.1}}
```

The beam path `BP` consists of a free space of length 1 and a lens of focal length 0.1.

There are constructor functions of optical components.

`FreeSpace[d]` will create a free space of length `d`.

`Lens[f]` will create a lens with focal length `f`.

`ABCDMatrix[M]` will create a general ABCD matrix object. `M` is the ABCD matrix.

In general, you have to put `FreeSpaces` between `Lens` or `ABCDMatrix` objects to construct a sensible beam path.

Once the beam path is constructed, you compile the beam path to a function.

You give the beam path object and the `q`-parameter of the beam at the origin of the beam path. Then `CompileBeamPath[]` function returns a function, which takes a distance from the origin as an argument and returns the `q`-parameter at the location.

■ Compile the beam path

```
In[6]:= q = CompileBeamPath[BP, q0];
```

```
In[7]:= q[z]
```

```
Out[7]= 
$$\begin{cases} q_0 + z & z \leq 1 \\ -1 + \frac{0.1(1+q_0)}{-0.9-q_0} + z & 1 < z \\ 0 & \text{True} \end{cases}$$

```

If you give a number to `q[]`, it will return a number.

```
In[8]:= q[0.5]
```

```
Out[8]= 0.5 + q0
```

In[9]:= `q[1.5]`

$$\text{Out[9]} = 0.5 + \frac{0.1(1 + q_0)}{-0.9 - q_0}$$

■ Propagate a beam

Let's propagate a beam with waist size 0.1mm through this beam path. First we prepare the q-parameter of the beam at the waist.

In[10]:= `q1 = i WaistSizeToRayleighRange[0.1 10-3, 1064 10-9]`

Out[10]= 0.0295262 i

WaistSizeToRayleighRange[w0, λ] converts a waist size w0 to a Rayleigh range (zr). λ is the wavelength of the beam. The q-parameter at the waist is just *i* zr.

Now we can propagate q1 through the beam path.

In[11]:= `q = CompileBeamPath[BP, q1];`

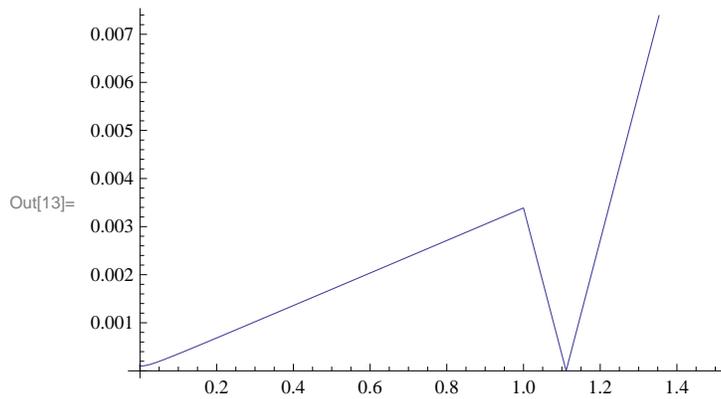
In[12]:= `q[0.3]`

Out[12]= 0.3 + 0.0295262 i

■ Plots

We can plot the beam size along the beam path like the following.

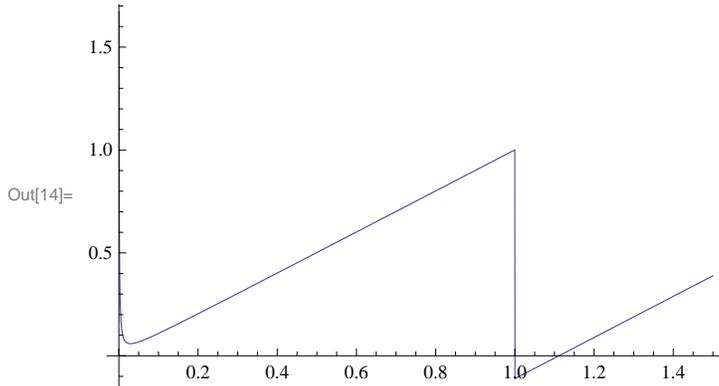
In[13]:= `Plot[QtoBeamSize[q[z], 1064 10-9], {z, 0, 1.5}]`



QtoBeamSize[q, λ] will convert a q-parameter into a $1/e^2$ beam radius.

We can also plot the radius of curvature of the beam using QtoROC function.

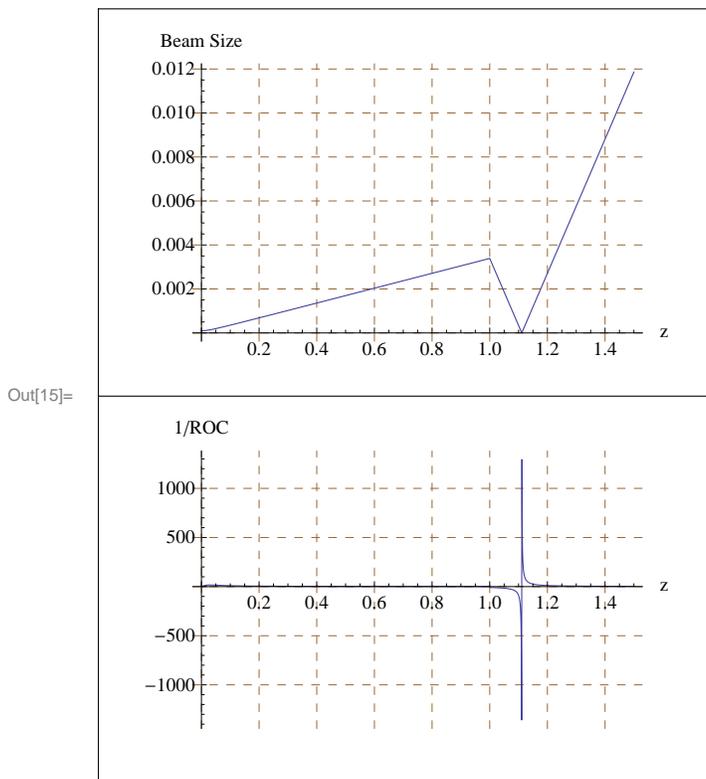
```
In[14]:= Plot[QtoROC[q[z], 1064 10-9], {z, 0, 1.5}, Exclusions -> None]
```



You may want to add "Exclusions -> None" option to Plot[] so that the discontinued points are connected. ROC generally shows jumps at lenses.

There is a utility function to plot the beam radius and ROC along the beam path.

```
In[15]:= PlotBeamPropagation[BP, q1, 1064 10-9, 1.5]
```



The syntax of the function is PlotBeamPropagation[BP, q0, λ zmax].

BP is a beam path object (not compiled), q0 is the initial beam parameter, λ is the wavelength. zmax specifies up to what distance you want to plot the beam propagation.

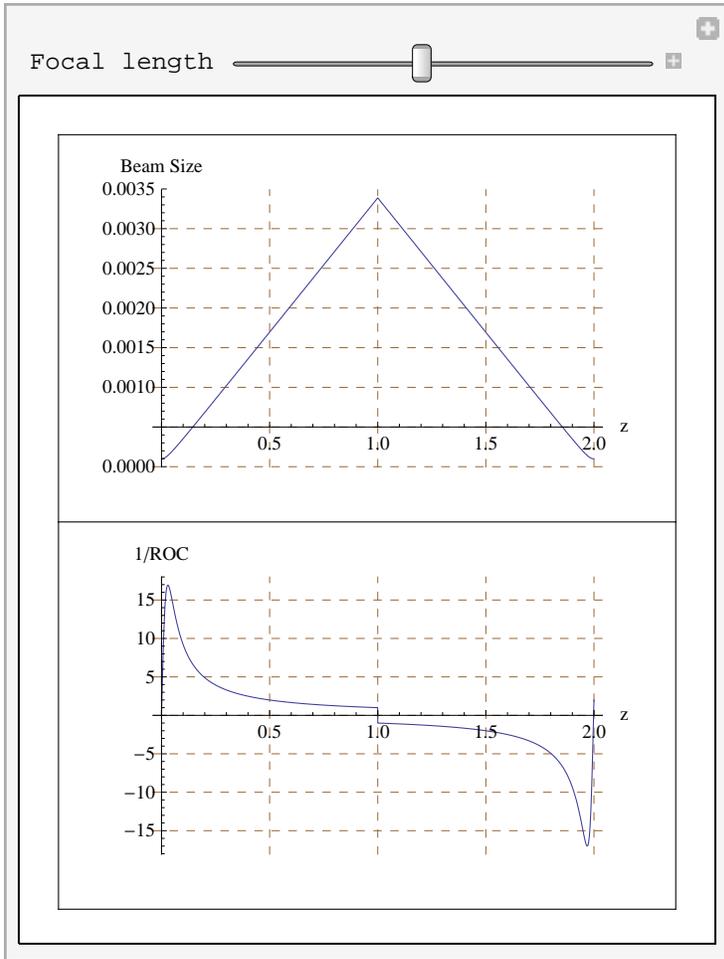
■ Interactive plot

You can make a manipulatable plot like this.

```
In[16]:= BP1 = {FreeSpace[1], Lens[f]};
```

```
In[17]:= Manipulate[PlotBeamPropagation[BP1 /. f -> x, q1, 1064 10-9, 2],
  {{x, 0.5, "Focal length"}, 0.1, 1}]
```

Out[17]=



Another example. In this case, we pre-compile the beam path. This is generally a bit faster.

```
In[18]:= q = CompileBeamPath[BP1, q1];
```

```
In[19]:= q[z]
```

$$\text{Out[19]= } \begin{cases} 0.0295262 i + z & z \leq 1 \\ -1 + \frac{(1+0.0295262 i) f}{(-1-0.0295262 i)+f} + z & 1 < z \\ 0 & \text{True} \end{cases}$$

```
In[20]:= Manipulate[Plot[QtoBeamSize[q[z] /. f -> x, 1064 10-9], {z, -1, 2}],  
  {{x, 0.5, "Focal length"}, 0.1, 1}]
```

Out[20]=

