GhostSplitUDOP

A LightTools® User Defined Optical Property to help analyze ghost image reflections

Summary

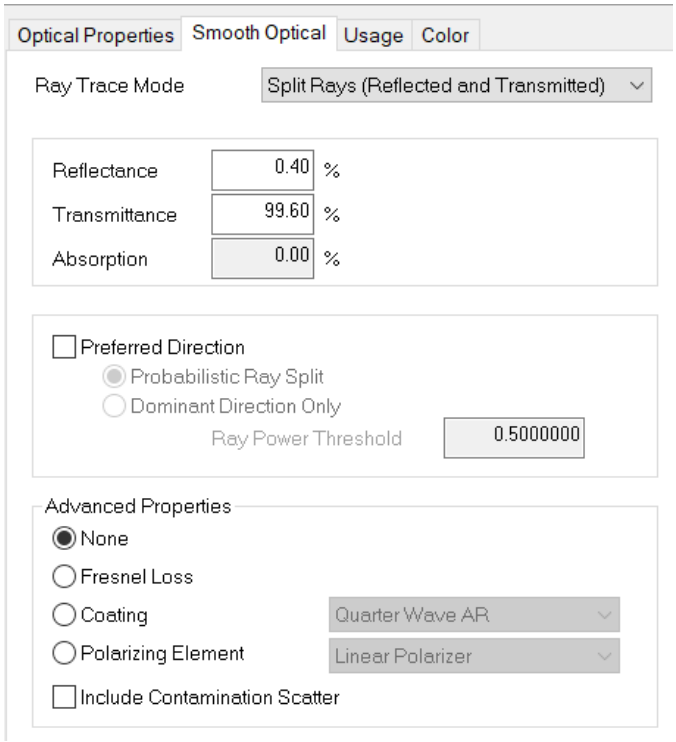
Analyzing ghost images and stray light in LightTools® software requires tracing rays through various possible stray light paths. This “plugin” allows to set the reflect/refract probability for a surface independently, keeping flux fractions physically correct. It is written in Visual Basic .NET, and placed in the public domain under CC0 1.0.

Problem description

Let’s assume we have a multi lens imaging (sub)system, and we would like to quantitatively analyze ghost images and other stray light issues. These are caused by partial reflection of light at the lens surfaces. Let’s assume a simple 0.4 % antireflective coating.

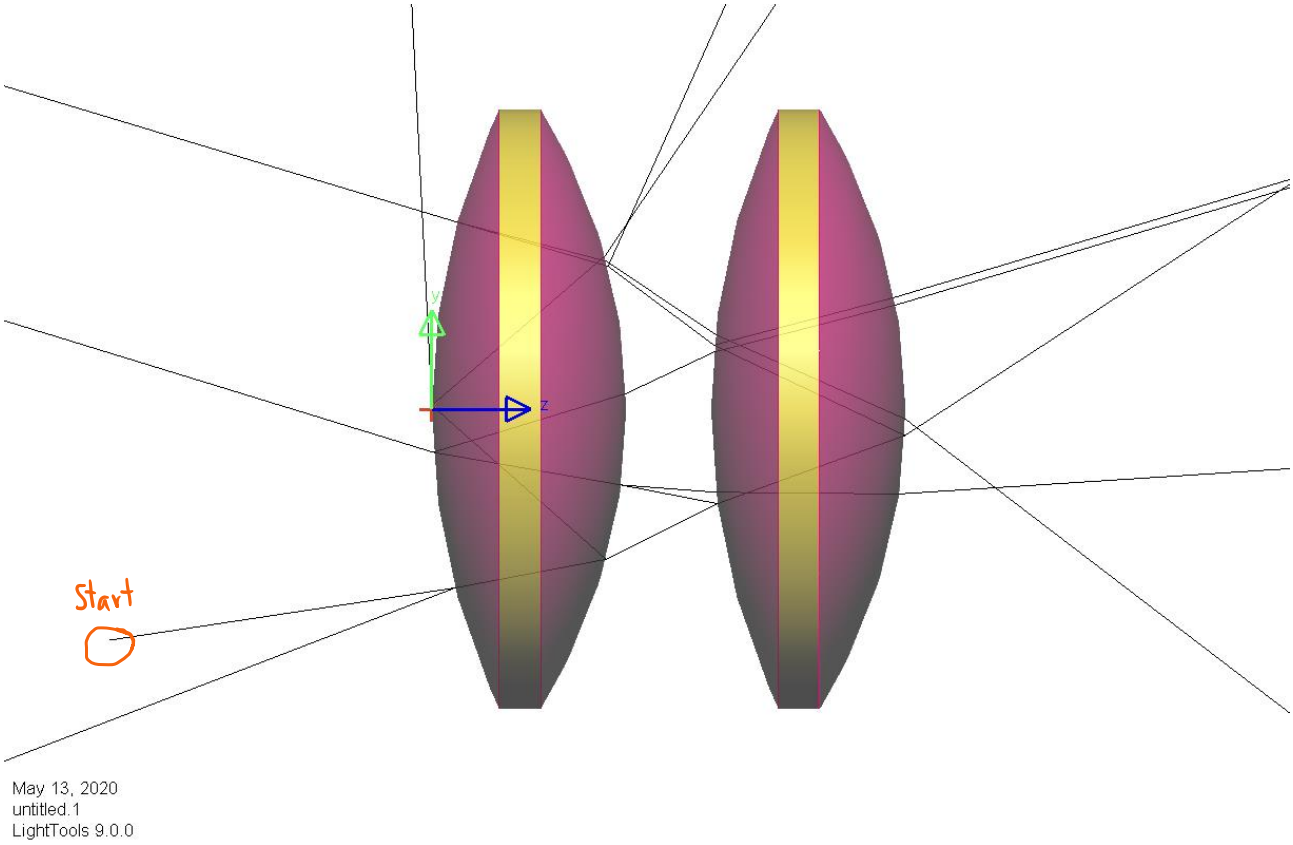
LightTools offers two ways to do this.

We can set the optical properties to “Split Rays” with the appropriate reflection/transmission:



Julius Muschaweck JMO GmbH Zugspitzstr. 66, 82131 Gauting, Germany ☎+49 89 89340203 julius@jmoptics.de

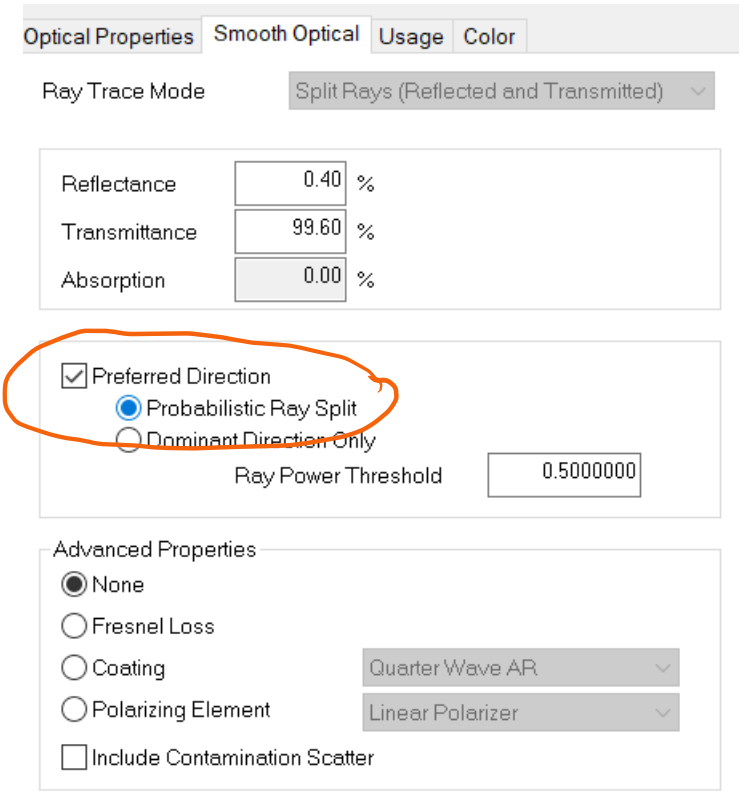
Then, each ray segment is being split at the surface into a strong transmitted and a weak reflected ray:



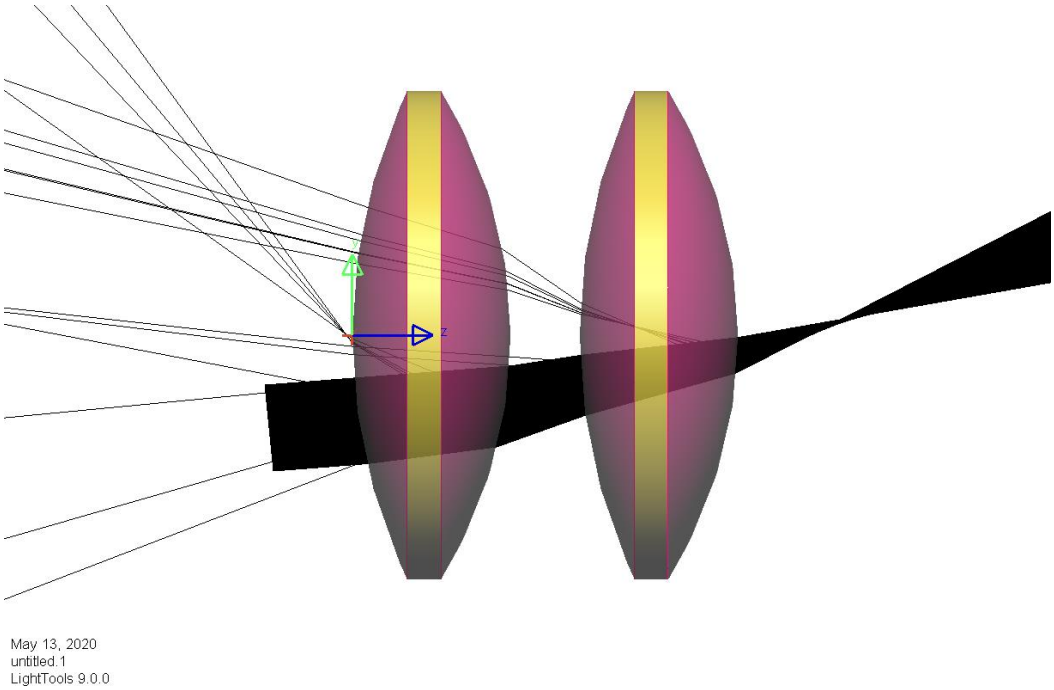
This image, using a single NS ray, shows how this correctly reproduces ray paths for each ray, until after two or more reflections, the ray segments become too weak and are no more traced. The “too weak” threshold can be set for each NS ray, or for the Monte Carlo simulation with a source. It is important to pay attention to this threshold, setting it low enough to avoid losing ray paths too early.

I’m often using values as low as 1e-12, but with ray splitting, this tends to cause overload: In a Monte Carlo simulation, there will be an exponential avalanche of very weak ray segments. Imagine a system with 10 lenses and 20 surfaces. Then each single ray will spawn up to 20 first generation and up to 19x19 = 361 second generation ray segments, and LightTools will spend a most of the computation time tracing those very weak rays. There may also be memory issues when receivers save ray data.

To avoid the avalanche, we can use “Probabilistic ray splitting”:



Now, rays keep their full power, while they are randomly refracted or reflected, with appropriate probabilities. To see the effect, we use a fan with 1,000 NS rays:



At each surface, about 4 of the 1,000 rays are reflected, as they should. LightTools traces only a single path per ray, and the avalanche is avoided .However, we now have only one out of

1 / 0.004^2 = 62,500 rays doing the ghost path we are interested in analyzing. In other words, we have to trace many, many rays to see actual ghost images emerge above the noise floor.

Both options offered by LightTools are less than ideal.

Separating refract/reflect fractions and probabilities

Inspired by the "Fixed Split" option which was recently added to the Harvey-Shack scatterer, I implemented a similar function in a User Defined Optical Property (UDOP).

As a reminder, a UDOP is a DLL, preferably a .NET Class Library, which implements an abstract interface defined by Synopsys. When a UDOP is attached to a property zone, LightTools calls a function in the DLL named `bendRay`, passing ray data to the DLL and basically asking "what shall I do with this ray?"

Here, I wanted to have a simple surface property with fixed reflectivity, transmissivity and absorption (these three always >= 0 and adding up to 100%), independent of wavelength and incidence angle for now, and only for unpolarized light.

There is a check box in the UDOP interface saying "Probabilistic Ray Split". When unchecked, the UDOP should behave just like the built-in "Split" option. When checked, however, the probabilities for refraction, reflection and termination should be separately adjustable, again all >= 0 and adding up to 100%.

The UDOP works much like Probabilistic Ray Splitting. For each incoming ray, at most one outgoing ray is generated. Ray powers are adjusted to keep expectation values correct. For example, if I had a refract power fraction of 1% and a refract probability of 10%, the refracted ray power is multiplied by 0.1, thus making sure that overall energies of the ray paths come out correctly.

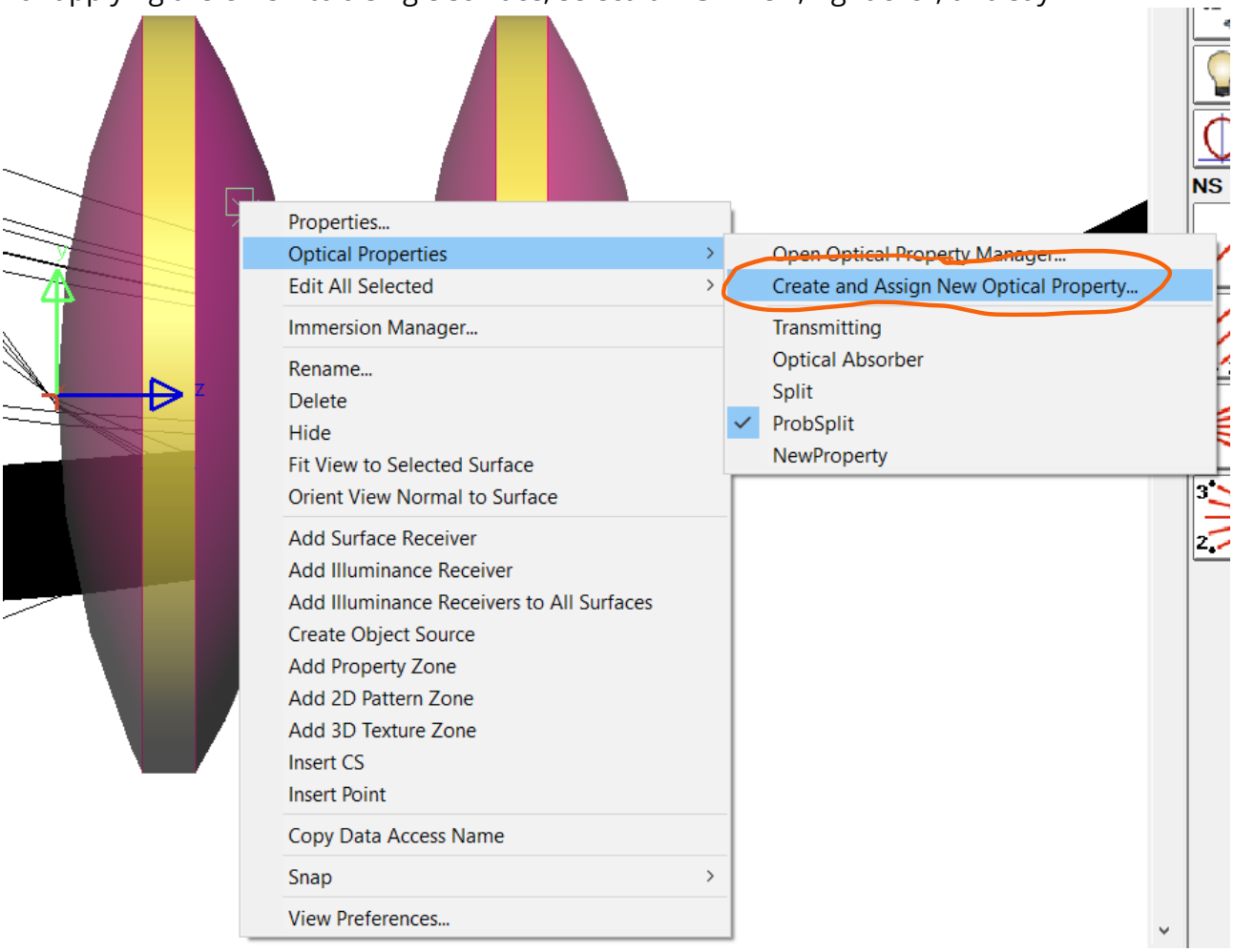
Why do I add "termination" to the probabilities? I can envision a situation where I would like to keep the flux coming from a certain surface, but would want LT to spend less time tracing rays from that surface. By setting the termination probability to something > 0, the UDOP randomly terminates rays, and assigns a compensating power increase to the non-terminated rays.

Using the UDOP

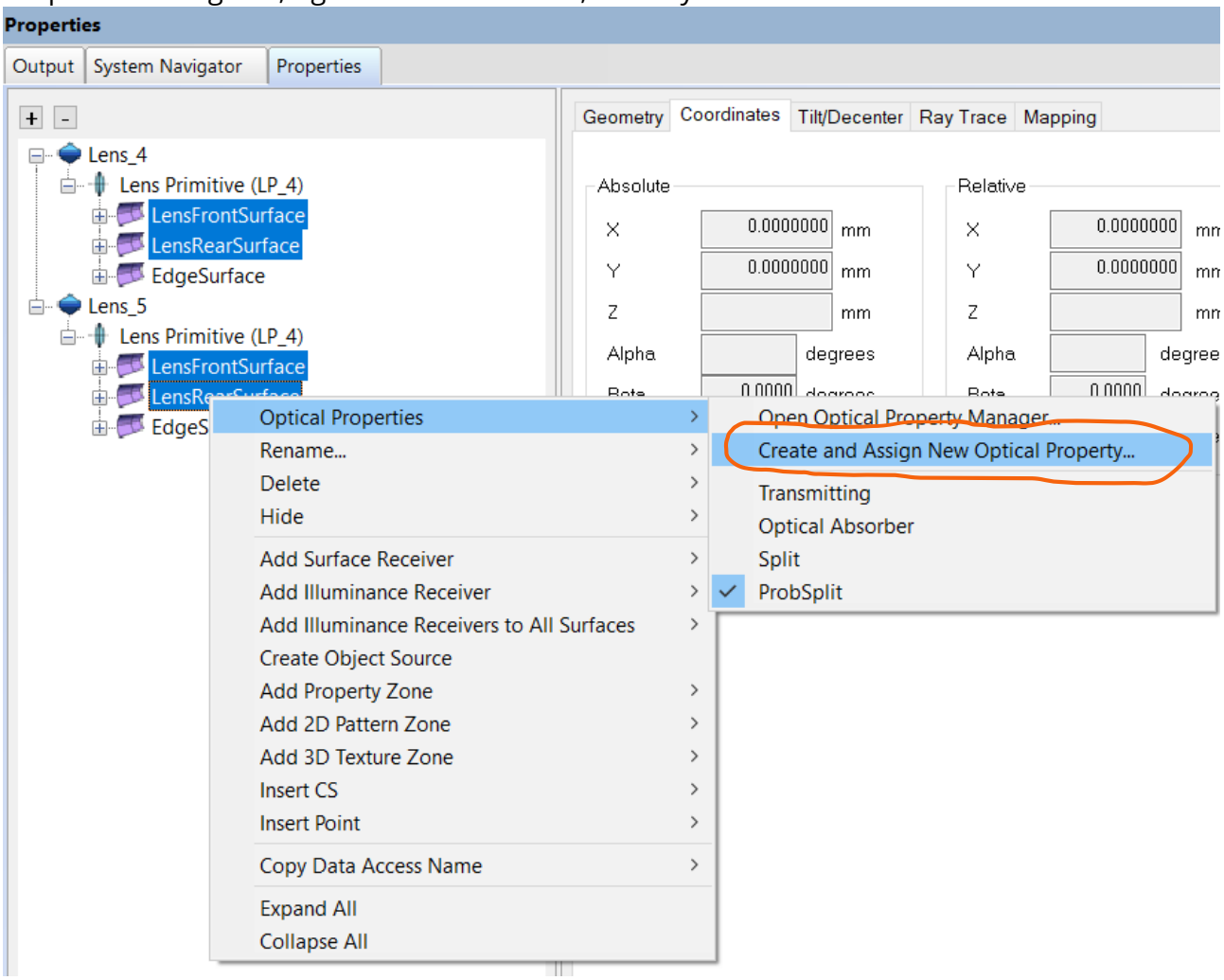
All you need is LightTools with an Advanced Physics Module license, and the file `GhostSplitUDOP.dll`. No installation required, just copy the DLL where you want it. (I tend to put it into the same directory where my `.lts` system files reside).

In LightTools, you need to create a new optical property. There are several ways to do that:

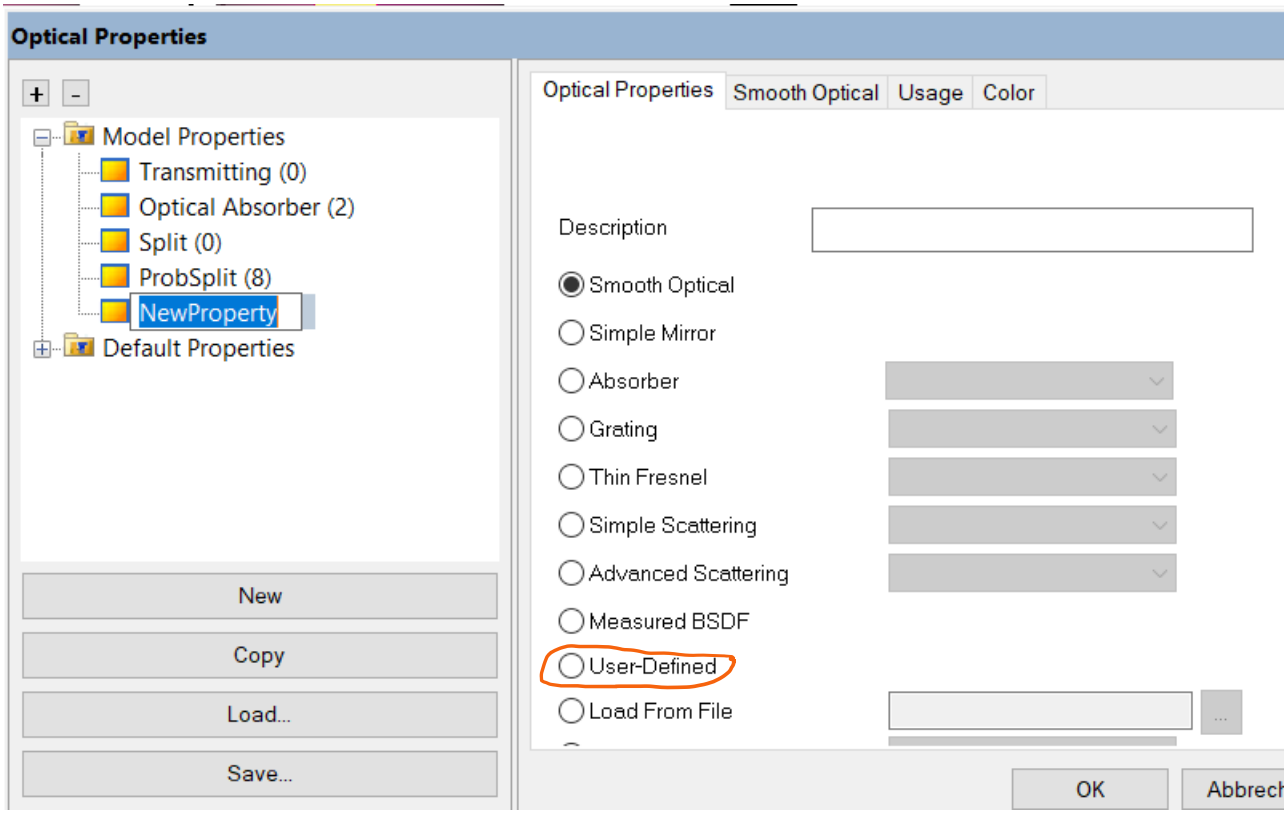
- For applying the UDOP to a single surface, select it in 3D view, right click, and say



- For applying the UDOP to all surfaces of a selection, select the surfaces in the Properties dialog box, right click one of them, and say

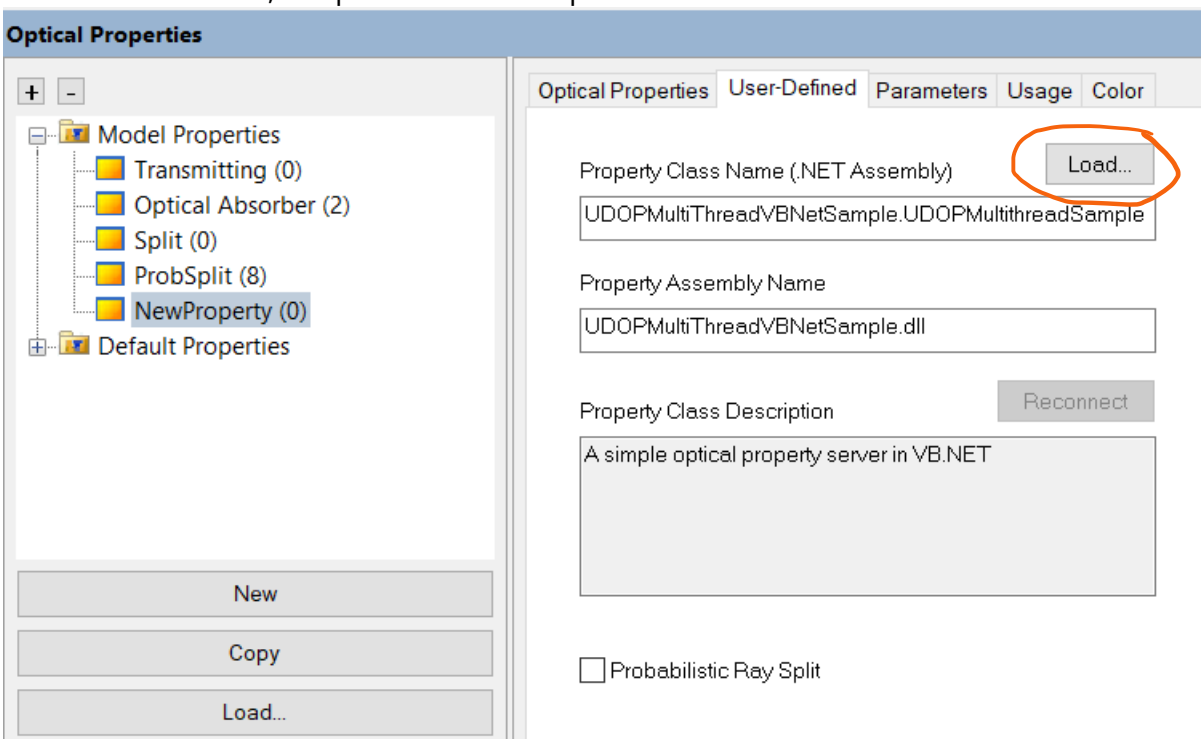


- For creating the UDOP optical property first, before you use it, you open the Optical Properties Manager (via Edit→Optical Properties, via the System Navigator, or like the two screenshots above, except that you click on “Open Optical Property Manager” instead of “Create and Assign New...”.
  In the Optical Properties Manager, you would click on “New”:

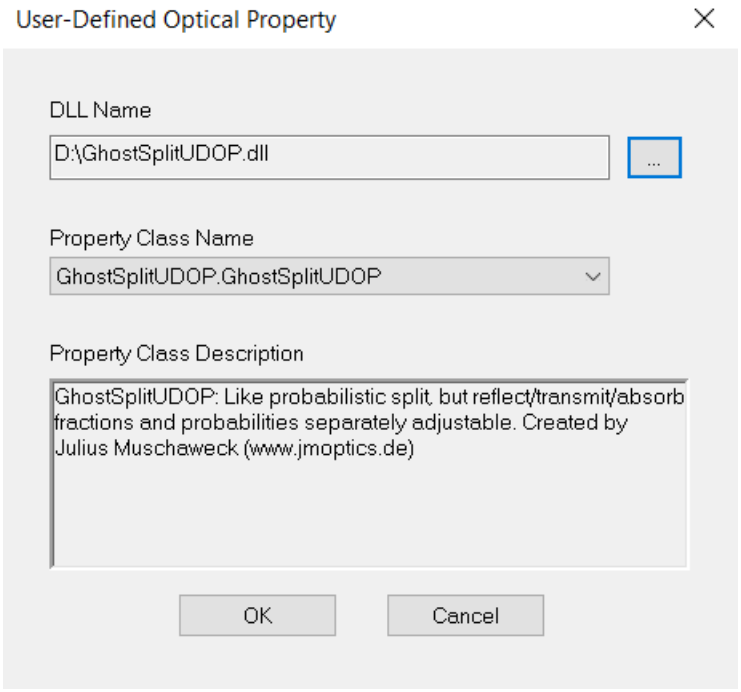


and give it a decent name like e.g. `GhostSplitUDOP_0_4`.

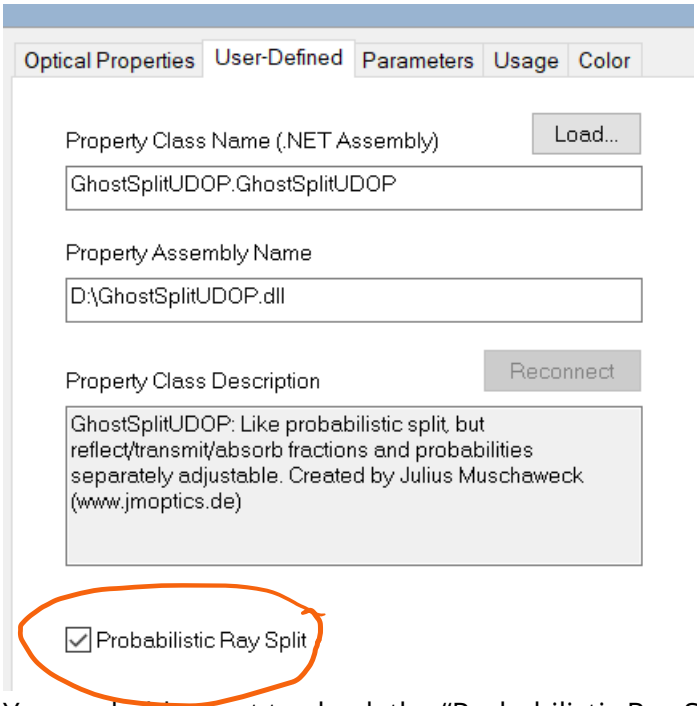
- You click on "User-Defined" and "Apply", and move to the newly appeared "User-Defined" tab. There, the predefined example UDOP is set:



- You click the “Load” button, navigate to the `GhostSplitUDOP.dll` file and load it.:

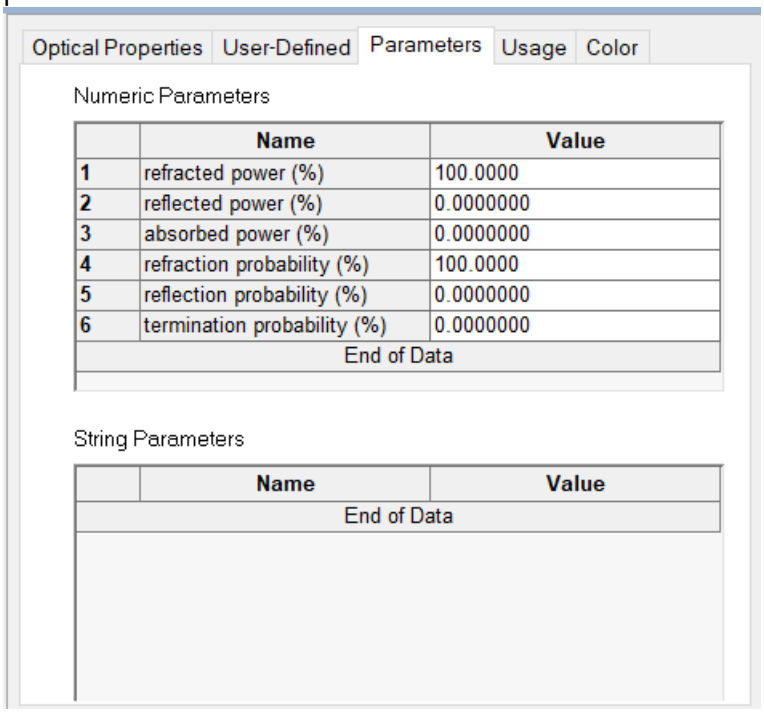


- Currently, there is only the simple `GhostSplitUDOP` class available.
  Then, the User-Defined tab looks like this:

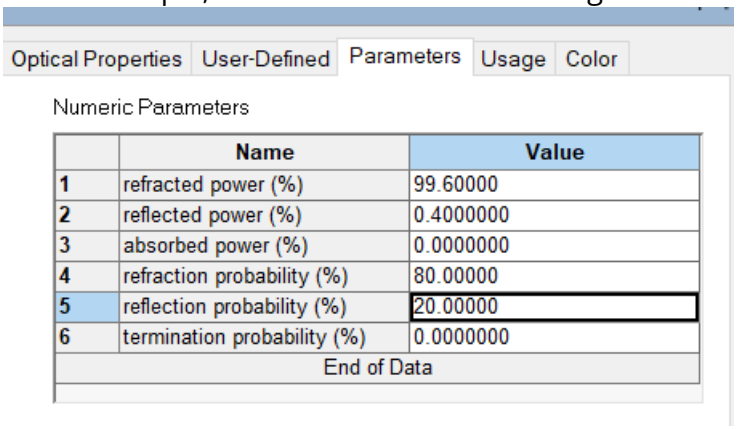


  You probably want to check the “Probabilistic Ray Split” box, because this is the point of the whole exercise.

- There is another new tab: The “Parameters” tab, where you can set and get certain parameter values the UDOP has defined:

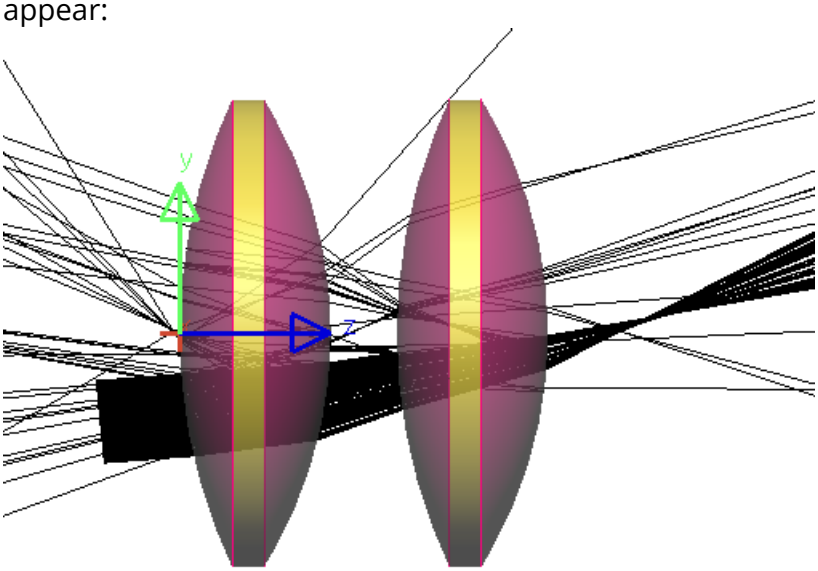


- By default, the surface is 100% refractive, that’s it. It behaves like the LightTools “Transmitted/TIR Rays” default. It refracts one ray with 100% power, except if there is total internal reflection, it reflects one ray with 100% power.
- You can enter any real number >= 0 and <= 100 (outside this range, LT says “Illegal value” and rejects it). The UDOP adapts the two other values, making sure the total of the power or probability triplet is always 100%.
- As an example, we take our 0.4% AR coating and set the reflection probability to 20%:



- If you selected all four lens surfaces and followed the instructions, these surfaces now are set to the UDOP. If you created the optical property from scratch, you need to assign the UDOP to these surfaces.

- Now, with a fan of just 100 rays, all ghost image ray paths involving two reflections appear:



Adjusting parameters

Refracted, reflected and absorbed power fractions should, of course, be adjusted according to the physical properties you want to model.

For the probabilities, you will probably have to experiment. In most cases, you would leave the termination probability at zero, and balance refraction vs. reflection probability. In this example, setting the reflection probability too low (like 1%, close to the reflectivity value of 0.4%) gives you not enough rays in the ghost paths. Setting it too high (like 80%) will result in not enough rays actually making it through the system by any path. I usually start with 50/50.

Where to get it

This UDOP is available at GitHub, https://github.com/JuliusMuschaweck/GhostSplitUDOP, including the source code and the Visual Studio 2019 project file.

I hope this helps you to analyze ghost images and stray light a little easier.

Please feel free to give me feedback at julius@jmoptics.de

Gauting, May 14, 2020

Julius Muschaweck