**OPOSSUM**   
**An optics simulation framework for large-size laser systems**

The objective of OPOSSUM, which stands for **O**pen**-S**ource **O**ptics **S**imulation **S**ystem and **U**nified **M**odeler, is the development of a common software platform for simulating various aspects of optical systems in a holistic approach. It should be particularly useful for simulating and designing large-size, high-energy / intensity laser systems.    
This work is part of the task 3.4 (supporting calculations for system design) within the THRILL project and is led by GSI.    
  
TLDR down below. 

**Concept of high-intensity-laser-system modeling**

The design of complex high-energy / intensity laser systems requires a detailed simulation of optical (and sometimes mechanical) effects and aspects. Often, these aspects must be simultaneously taken into account while optimizing a system’s design. Distinct aspects might even stand against each other, such that optimizing (e.g., maximizing/ minimizing) one parameter degrades the performance of other system parameters. Therefore, a rather holistic approach is desirable.

In the past, many tools were developed, often addressing very particular optical effects at several companies and research institutes. These tools are often only used at the institutions which developed the software and even there only used by a few people (e.g., in the frame of a master or PhD thesis). This of course leads to rewriting of such simulation tools with the same functionality, as the knowledge and software or code was transferred or maintained. Hence, a common set of tools accompanied by proper knowledge exchange would significantly reduce this inefficiency.

Besides the solutions for modelling particular aspects of optical systems, there are many more general-purpose tools on the market, which are, unfortunately, mostly commercial, closed-source solutions, which each follow their own underlying design strategy. Furthermore, many of these tools (e.g., ZEMAX, OSLO, etc.) are designed for simulating more "traditional" optical systems such as camera objectives or illumination setups. In contrast, high-intensity-laser systems commonly demand different or additional features which are not always fully supported (or easy to model) by these software packages.

The usage of different tools during the design phase often requires repeatedly modelling the optical system in the particular software and providing a bunch of input parameters. A common platform would allow for modelling the desired system once and analyzing it with the above-mentioned tools and providing the input data in the needed format.

The first task of this work is the development of a concept, which allows for modeling the most common (or ideally all) optical systems. This effort leads to the development of the OPOSSUM framework. The entire software and its documentation is accessible at

<https://git.gsi.de/phelix/rust/opossum>

**Description of the simulation principle**

In general, optical systems consist of light sources which provide a light field and optical components, which modify this light field. These light fields may be described in numerous ways, e.g., time invariant or time dependent, rays, complex fields etc.

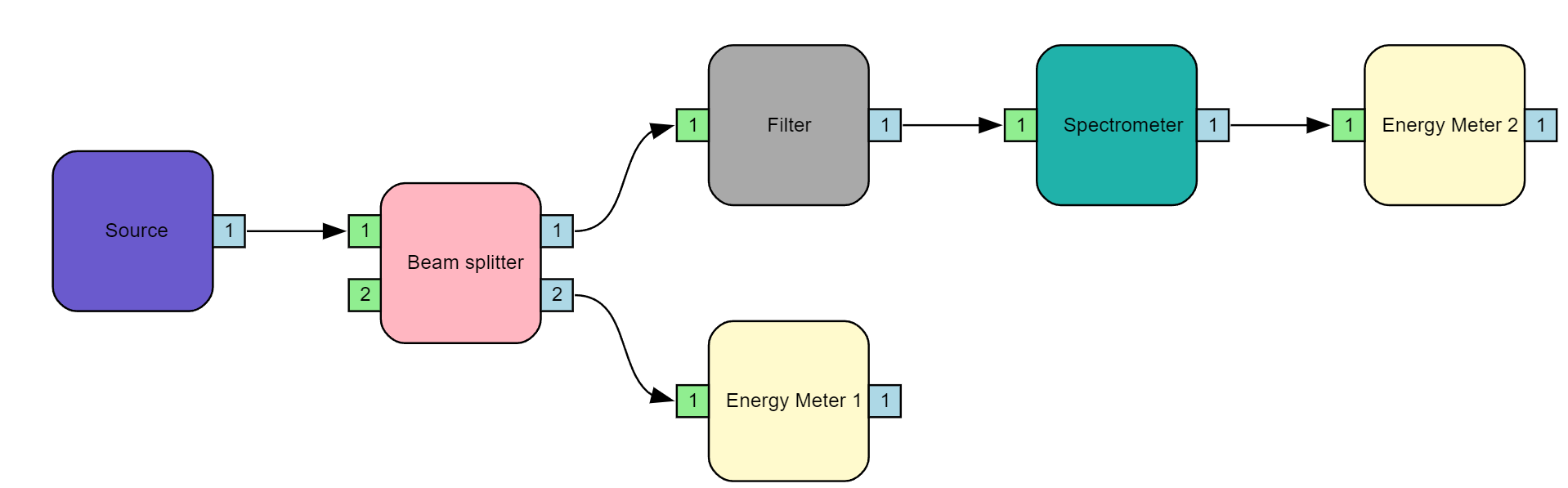
Furthermore, there are light sinks such as simple beam dumps, targets or detectors. These are the elements, which produce a "result" (e.g., measurable signal) and thus make a system "productive". The components - light sources (such as a laser) or optical elements (e.g., Faraday isolators) - might itself consist of subcomponents. In principle, these components again might consist of subcomponents with a nesting level as high as needed to fully describe the component.

Of course, for a full system description, it would be sufficient to place the mechanical model of the optical components in a 3D space. For certain tasks, such as illumination or stray light analysis this would be an appropriate approach (and thus will be supported by our model). However, typical systems mostly cast optical rays or light fields in a directed way from one component to the next one. Optical systems can thus be rather described in network- or most often in tree-like structures.

Following the above idea, well-established structures could be used which already exist for a long time: directed graphs (<https://en.wikipedia.org/wiki/Directed_graph>). A directed graph consists of so-called nodes and edges. For our purposes, nodes represent the optical components, while edges represent the information about the light (energy, wavelength, wave front, nearfield distribution, etc.) to be handed from one node to the next one.

A node has one or multiple ports to which edges can be connected to. We thereby strictly distinguish between incoming and outgoing ports. A node with no input ports represents a light source. A simple (ideal) propagation node would have one input and one output port. Furthermore, an ideal beam splitter has one input port and two or more output ports. More realistic components, such as a real lens could also have more than one input and output ports e.g., for simulating ghost reflections from lens surfaces. A graph example is shown in Fig. 1.

There will be different node types representing various optical components (ideal / real lenses, beam splitters, wave plates, etc.). Each node has, depending on its node type, various attributes, which describe component parameters such as length (e.g., for propagation nodes), focal length (ideal lenses), radii of curvature (real lenses) etc.



*Figure 1 Exemplary visualization of a directed graph for a simple setup.*

An optical model can now be investigated using different analyzer modules. Each module can concentrate on a specific aspect such as energy transmission, geometric raytracing, Fourier optics propagation or even finding all ghost foci in a system. The key aspect is the use of a model that only needs to be developed once, with no data conversion for different analyses.

Using the above-mentioned approach, nodes can include arbitrary analysis algorithms and simulate even complex effects. They might also use already existing simulation codes. As an example, a code for the simulation of second harmonic generation (SHG) using non-linear crystals has been developed as a separate project in our group. The OPOSSUM framework will allow for creating an SHG node incorporating this code. This way, arbitrary optical setups can make use of such a node.

**The OPOSSUM ecosystem**

For an efficient design workflow, knowledge of the optical / mechanical properties of the used materials is required. For this, the side project “Materialdb” provides a flexible database for storing material properties in a collaborative manner. Currently, it is in an early development stage, but a first tech preview is available under

<https://git.gsi.de/phelix/rust/materialdb_backend> (backend server)

<https://git.gsi.de/phelix/rust/materialdb_frontend> (web frontend)

In addition, a catalog of off-the-shelf optics and reusable components would also be necessary. For this, an additional database “OpticDB” is in planning.

As of now, OPOSSUM can be tested by defining setups within the software code itself (which is written in Rust: <https://www.rust-lang.org/>) or via a console interface. Future developments will also include a graphical user interface to facilitate the usage of OPOSSUM.

**OPOSSUM**

**O**pen**-S**ource **O**ptics **S**imulation **S**ystem and **U**nified **M**odeler - TLDR

With the development of OPOSSUM, we target a unified modeling software for large-size, high-energy / intensity laser systems in a holistic approach. By utilizing a directed-graph-based propagation algorithm with optical components or sub-systems, which are represented by nodes, a general description of the whole laser system will be given. Once defined, this system will be analyzable by a variety of modeling methods, such as raytracing, gain calculation, ghost-focus analysis etc., without the need to re-define the system in different simulation tools. To do so, we plan on not only implementing new modeling methods from scratch, but also allow the possibility to link external software to be used within the OPOSSSUM framework. With this, we target the knowledge exchange from different institutions, such that existing tools can be maintained and used without the need to “reinvent the wheel.”

This work is part of the task 3.4 (supporting calculations for system design) within the THRILL project and is led by GSI.