Finite elements for inverse optical design

"How to efficiently design mirrors and lenses so that you get a desired light distribution from a given source distribution" this is the question at the heart of this project, and at the heart of much of the industrial development of Eindhoven in the past century. It comes up whenever a target illumination has to be achieved (like in lighting applications: street lights, head lights, or Glow!!), or in the manufacturing of precision equipment, from camera lenses to the lithography machines. See Figure 1 for an example, and click [this link]() for a Youtube video on "futuristic" applications. Mathematically this question inevitably leads to equations of the Monge-Ampère type. For prescribed source and target distributions $f: X \to \mathbb{R}^+$, $g: Y \to \mathbb{R}^+$ (X, Y are domains in $\mathbb{R}^d$) the problem is to find a mapping $\nabla u: X \to Y$ with u parametrizing the optical surface which solves the equation (energy conservation + minimum transport cost)

$$g(\nabla u)\ \det(D^2 u) = f \quad \text{in} \quad X \qquad \textit{(Eq 1)}$$

with the boundary condition $\nabla u(\partial X) = \partial Y$ (boundary maps to boundary). This leads to a highly nonlinear problem that is quite challenging to solve. So far the focus has been to develop least square solvers for obtaining numerical approximations. Finite difference and volume schemes have also been suggested for the problem.

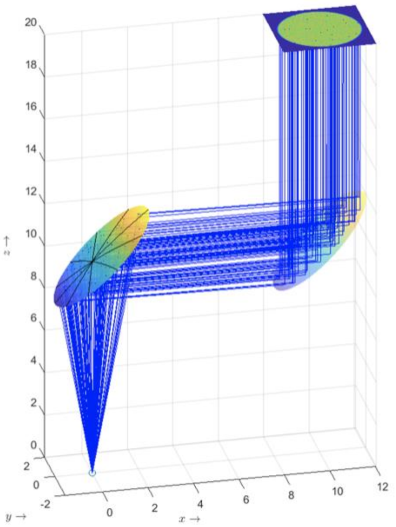

Figure 1 A collimated beam from a point source using two mirrors, computed using algorithms that will be considered during this project.

This project aims to develop a finite element solver for the problem in Python (FEniCS) or other free-source platforms. The advantages of this approach are manyfold including handling more general meshes and geometries, going higher order in discretization, improving error control, and adaptive discretization and iterations.

The project is targeted to give a wide overview to students on the development process of numerical algorithms (in terms of theory and implementation) for challenging application-based problems.