The Journal of Open Source Software

ZOSPy: optical ray tracing in Python through OpticStudio

Luc van Vught [1,2*¶], **Corné Haasjes** [1,2,3*], **and Jan-Willem M. Beenakker** [1,2,3]

1 Department of Ophthalmology, Leiden University Medical Center, Leiden, the Netherlands **2** Department of Radiology, C.J. Gorter MRI Center, Leiden University Medical Center, Leiden, the Netherlands **3** Department of Radiation Oncology, Leiden University Medical Center, Leiden, the Netherlands ¶ Corresponding author * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Paul La Plante

Reviewers:

- @brandondube
- @alvesjnr

Submitted: 03 July 2023
Published: unpublished

Summary

Zemax OpticStudio (Ansys, Inc) is a commonly used software package for designing optical setups and performing ray tracing simulations. It offers an Application Programming Interface (API) but interacting with this API is complex. Consequently, current ray tracing simulations generally require substantial manual user interaction, which in turn hampers the sharing of methods between scientists. We have therefore developed `ZOSPy`, a Python package that provides an accessible interface as well as unit tests. As a result, `ZOSPy` enables scientists to focus more on optical modelling instead of coding and contributes to open science as optical setups and analyses can easily be shared amongst users.

Statement of need

Ray tracing simulations are widely used to design, optimize and analyze optical systems. Its applications are diverse, ranging from designing spectrometers (Naeem et al., 2022) or telescopes (Zhang et al., 2023), to understanding the optics of the human eye (Simpson, 2020; van Vught et al., 2022). Moreover, in ophthalmology, ray tracing is used to optimize the outcomes of cataract surgery (Artal et al., 2023; Canovas & Artal, 2011) and evaluate the accuracy of ocular radiotherapy (Jaarsma-Coes et al., 2023). These optical simulations are often performed in OpticStudio, which offers a powerful set of tools to design, optimize and evaluate optical systems.

Although OpticStudio offers an API, the `ZOS-API`, using this API in Python is complex and time-consuming. It involves, for example, establishing a connection with the API through the .NET framework, casting between .NET and Python datatypes, identifying which constants need to be set in specific cases, and working around non-uniform methods of parsing the output (Zemax LLC, 2021). This leads to studies which, in practice, largely rely on user interaction. Although OpticStudio can perform Monte Carlo analyses, where a large number of random perturbations of the system are generated and analysed in an automated way, this type of automation is not suitable when large sets of specific, non-random, combinations of parameters need to be analysed. In vision science, for example, ray tracing is used to design artificial lenses for the eye (Ellis, 2001; Holladay et al., 1999), but their evaluation in a large set of patients is hindered as the anatomical parameters of each subject's eye need to be entered manually. As a result, clinical studies typically describe vision-related complaints in cohorts of hundreds of eyes (Alfonso et al., 2007), but the ray tracing studies aiming to link these outcomes to the subject's ocular optics are limited to a small number of eyes (Simpson, 2020; van Vught et al., 2020).

With ZOSPy, we provide an easy-to-use and accessible interface to the OpticStudio API, enabling the user to focus on optical modelling instead of complex coding. As a result, those who are not familiar with the intricacies of the `ZOS-API` interface will be able to read and comprehend scripts that use `ZOSPy`. Thereby, `ZOSPy` provides greater accessibility to conducting analyses in OpticStudio through Python than directly using the `ZOS-API`.

Functionality

`ZOSPy` is, in its most basic form, a Python wrapper around the OpticStudio API. It facilitates the .NET connection required to connect to OpticStudio through its API, as well as all subsequent casting of variables between .NET and Python. Additionally, it provides object-oriented methods to define surfaces and their optical properties. Furthermore, it offers single-line, easy to understand, methods to perform analyses that return the analysis results in a uniform way. As a result, `ZOSPy` enables a straight-forward interaction with OpticStudio and improves code readability, which facilitates method sharing between scientists.

`ZOSPy` also offers autocompletion. Interacting with OpticStudio through its API requires the use of many constants, for example to define the shape of an optical surface or initiate an analysis. These constants do not autocomplete in IDEs such as PyCharm or VS Code as the API is built on the .NET framework. As a result, the user has to know the exact name of each constant, for example `ZOSAPI.Analysis.Settings.Mtf.MtfTypes.Modulation`. `ZOSPy`, however, includes stubs for all constants and functions, enabling full autocompletion.

Finally, `ZOSPy` offers a set of unit tests to assure that the software provides correct results. These tests provide means to compare results across `ZOSPy` and Python versions, as well as across versions of OpticStudio. The current version of `ZOSPy` provides basic tests for the most common optical surfaces and analyses.

Use cases

Multiple examples, from modelling the effect of a coated prism on the polarization of light to assessing the optical characteristics of the human eye have been contributed to `ZOSPy`. These examples provide new users with an easy start with `ZOSPy`. Part of a simple example of using `ZOSPy` to create and evaluate a thick lens is shown below, and the corresponding results are shown in Figure 1.

```python
#...

# Make a 10 mm thick lens with a radius of curvature of 30 mm
# and material type BK10
front_surface = oss.LDE.GetSurfaceAt(2)
front_surface.Radius = 30
front_surface.Thickness = 10
front_surface.SemiDiameter = 15
front_surface.Material = "BK10"

back_surface = oss.LDE.InsertNewSurfaceAt(3)
back_surface.Radius = -30
back_surface.Thickness = 29
back_surface.SemiDiameter = 15

#...

# Render the model
draw3d = zp.analyses.systemviewers.viewer_3d(oss)
```

```python
# Analyze the system by calculating the Modulation Transfer Function (MTF)
# and the Point Spread Function (PSF)
mtf = zp.analyses.mtf.fft_through_focus_mtf(
    oss, sampling="512x512", deltafocus=2.5, numberofsteps=51)

huygens_psf = zp.analyses.psf.huygens_psf(
    oss, pupil_sampling="512x512", image_sampling="512x512", normalize=True)
```

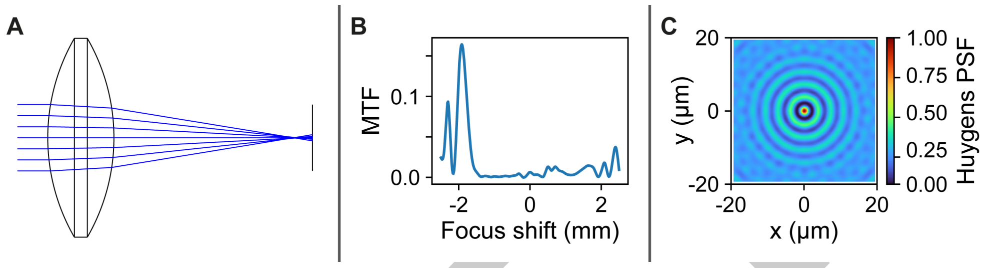

Figure 1: Results of the example code. A) The created optical system results in a slightly out of focus image. **B)** The calculated Modulation Transfer Function (MTF) shows that the image plane needs a 1.9 mm shift towards the lens to be in focus. **C)** The Huygens Point Spread Function (PSF) shows the aberrations of the system.

Furthermore, ZOSPy has been used in different ophthalmic studies. In one of these studies, ZOSPy was used to evaluate the relation of ocular anatomy to peripheral visual complaints (van Vught et al., 2022). In another study, ZOSPy showed that the extent of an intra-ocular tumor can be overestimated during surgery due to its shadow (Figure 2) (Jaarsma-Coes et al., 2023).

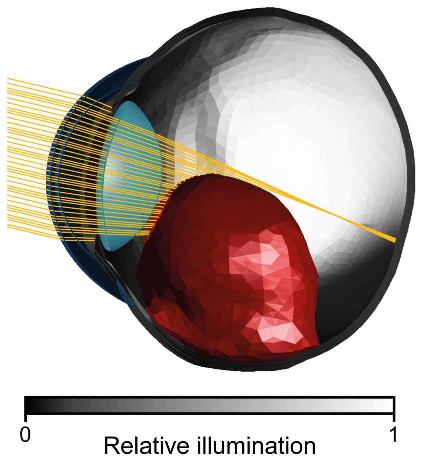

Figure 2: Simulation mimicking the clip surgery for radiotherapy of an intraocular tumor (Jaarsma-Coes et al., 2023). The ocular geometry including the dimension of the tumor were loaded into OpticStudio using ZOSPy and the *CAD Part: STL* object type, after which the retinal illumination was simulated. The results were rendered using the non-sequential Shaded Model analysis (`zospy.analyses.systemviewers.nsc_shaded_model`).

Acknowledgements

We would like to thank all users that have reported bugs or implemented new functionality.

References

Alfonso, J. F., Fernández-Vega, L., Baamonde, M. B., & Montés-Micó, R. (2007). Prospective visual evaluation of apodized diffractive intraocular lenses. *Journal of Cataract & Refractive Surgery*, *33*(7), 1235–1243. https://doi.org/10.1016/j.jcrs.2007.03.034

Artal, P., Ginis, H., Christaras, D., Villegas, E. A., Tabernero, J., & Prieto, P. M. (2023). Inverted meniscus intraocular lens as a better optical surrogate of the crystalline lens. *Biomedical Optics Express*, *14*(5), 2129–2137. https://doi.org/10.1364/BOE.490089

Canovas, C., & Artal, P. (2011). Customized eye models for determining optimized intraocular lenses power. *Biomedical Optics Express*, *2*(6), 1649–1662. https://doi.org/10.1364/BOE.2.001649

Ellis, M. F. (2001). Sharp-edged intraocular lens design as a cause of permanent glare. *Journal of Cataract & Refractive Surgery*, *27*(7), 1061–1064. https://doi.org/10.1016/S0886-3350(00)00856-7

Holladay, J. T., Lang, A., & Portney, V. (1999). Analysis of edge glare phenomena in intraocular lens edge designs. *Journal of Cataract & Refractive Surgery*, *25*(6), 748–752. https://doi.org/10.1016/S0886-3350(99)00038-3

Jaarsma-Coes, M. G., Ferreira, T. A., Marinkovic, M., Vu, T. K., Vught, L. van, Haren, G. R. van, Rodrigues, M. F., Klaver, Y. L., Verbist, B. M., Luyten, G. P., & others. (2023). Comparison of magnetic resonance imaging–based and conventional measurements for proton beam therapy of uveal melanoma. *Ophthalmology Retina*, *7*(2), 178–188. https://doi.org/10.1016/j.oret.2022.06.019

Naeem, M., Imran, T., Hussain, M., & Bhatti, A. S. (2022). Design simulation and data analysis of an optical spectrometer. *Optics*, *3*(3), 304–312. https://doi.org/10.3390/opt3030028

Simpson, M. J. (2020). Intraocular lens far peripheral vision: Image detail and negative dysphotopsia. *Journal of Cataract & Refractive Surgery*, *46*(3), 451–458. https://doi.org/10.1097/j.jcrs.0000000000000103

van Vught, L., Luyten, G. P., & Beenakker, J.-W. M. (2020). Distinct differences in anterior chamber configuration and peripheral aberrations in negative dysphotopsia. *Journal of Cataract and Refractive Surgery*, *46*(7), 1007. https://doi.org/10.1097/j.jcrs.0000000000000206

van Vught, L., Que, I., Luyten, G. P., & Beenakker, J.-W. M. (2022). Effect of anatomical differences and intraocular lens design on negative dysphotopsia. *Journal of Cataract & Refractive Surgery*, 10–1097. https://doi.org/10.1097/j.jcrs.0000000000001054

Zemax LLC. (2021). *Getting started with ZOS-API*. https://www.zemax.com/blogs/free-tutorials/getting-started-with-zos-api

Zhang, Y., Jiang, H., Shectman, S., Yang, D., Cai, Z., Shi, Y., Huang, S., Lu, L., Zheng, Y., Kang, S., & others. (2023). Conceptual design of the optical system of the 6.5 m wide field multiplexed survey telescope with excellent image quality. *PhotoniX*, *4*(1), 1–25. https://doi.org/10.1186/s43074-023-00094-4