

Light Attenuation and Sensitivity Analysis Models

This repository contains Python scripts for simulating light attenuation in an *Edwardsiella* vaccine solution and performing sensitivity analysis on the model parameters. These scripts are standalone and can be run in any environment with the appropriate Python packages installed.

Model Introduction

- `Mie_scattering_and_refractive_index_fluctuations.py` (**Mie Scattering and Refractive Index Fluctuation Light Attenuation Model**)
 - **Functionality:** This model calculates the relationship between outgoing light intensity, incident intensity, and propagation distance, based on Mie scattering and turbulence effects.
 - **Output:** It generates a 2D heatmap showing the distribution of outgoing light intensity under different incident intensities and propagation distances, which is saved as `Edwardsiella_vaccine_solution.png`.
- `Optical_Attenuation_Extension_Model.py` (**Extended Light Attenuation Model**)
 - **Functionality:** This script simulates the light attenuation process in a dynamic Archimedes screw system, where the optical path length changes periodically over time.
 - **Output:** It generates a 2D heatmap showing the dynamic changes in outgoing intensity with respect to time and incident intensity, saved as `Edwardsiella_vaccine_dynamic_screw_system.png`. Additional analysis charts are also generated.
- `sobol_sensitive.py` (**Sobol Sensitivity Analysis**)
 - **Functionality:** This script performs a global sensitivity analysis on the light transmission model using the Sobol method. It quantifies the influence of optical parameters such as the absorption coefficient α , particle scattering coefficient β_p , and turbulence scattering coefficient β_t on the model's output.
 - **Output:** It generates a comparison plot of the first-order and total-effect Sobol indices (`sensitivity_comparison_optimized.png`) and a heatmap of the second-order interaction effects (`interaction_heatmap_optimized.png`).

Requirements

Running these models requires a Python 3 environment and the following third-party libraries:

- numpy
- matplotlib

- SALib
- scipy

You can install all dependencies with a single `pip` command:

```
pip install numpy matplotlib salib scipy
```

Example

You can navigate to the directory containing the scripts and execute any script using the Python command.

```
# Run the sensitivity analysis
python "sobol sensitive.py"
```