



Chair of Fire Dynamics

Report

# Radiative Heat Transfer

— Project Report (2023/2024) —

Date: 12.03.2024 von

Rafay Nawaid Alvi

Matrikel-Nr. 2132500

Degree programm: Computer Simulation in Science (M.Sc.)

Supervisors: Univ.-Prof. Dr.-Ing. Fabian Brännström  
Johannes Sailer, M.Sc.

# Inhaltsverzeichnis

<b>1 Projects</b>	<b>3</b>
1.1 Preparation . . . . .	3
1.2 Cases . . . . .	4
1.2.1 Case 1: fireFOAM Benchmark in Quasi 1D Setup . . . . .	4
1.3 Implementation: Case 1 . . . . .	7
1.3.1 Introduction . . . . .	7
1.3.2 Project Files . . . . .	8
1.3.3 Simulation: Preprocessing . . . . .	9
1.3.4 Simulation: Postprocessing . . . . .	11
<b>Appendices</b>	<b>15</b>
<b>A Appendix</b>	<b>16</b>

# 1 Projects

As part of the lecture *Radiative Heat Transfer* you are working on projects. There are three different topics available. All projects are connected to radiation and include working with *fireFOAM* and *matplotlib*.

To successfully finish the project work, you work through the tasks and document the results using this L<sup>A</sup>T<sub>E</sub>X template. The purpose of the report is that you learn to work with *fireFOAM*, document, visualize and postprocess the results and teach your colleges how to use the tools they dont use in their project. In addition to that, your report will be given to students in the coming years such that they learn from your experience. We like to encourage you to study the soure code to see, in which sense the theory from the lecture is realized in the CFD code.

When working in the project groups try to distribute the effort uniformly among the team members. The report has to be handed in until the 29th of January 2024. Despite there is no size limit, dont write a novel. There is no grade for the report, but questions to your report will be asked during the oral exam.

The projects can be downloaded [here](#). The cases in the git are tested for OpenFOAM version OF2212.

Whenever something is unclear write into the [forum](#) and share your thoughts. All course members are encouraged to share knowledge and answer questions of their colleagues.

## 1.1 Preparation

Make sure that *OpenFOAM*, *paraview*, *python*, *matplotlib* and *pandas* is installed. To compile the L<sup>A</sup>T<sub>E</sub>X template you can use *Overleaf* or install L<sup>A</sup>T<sub>E</sub>X locally. For local installation you can use *LuaLaTex*. Other T<sub>E</sub>X systems are possible. A big but comprehensive installation of T<sub>E</sub>X is given by *texlive-full*. As L<sup>A</sup>T<sub>E</sub>X editor you can use *vs-code* or *texstudio*. To understand the source code, its helpful to install/compile *OpenFOAM* with debug switches and use a debugger (ex. *dgb* or a special *vscode* setup) to follow the code.

Compile the solver `fireRADFoam_MaCFP` by going into the folder `Solver/OpenFOAM/fireRADFoam_MaCFP` and typing `wmake`. Test if the process worked. By executing `fireRADFoam_MaCFP`. Note that you need to have a working *OpenFOAM* environment loaded to compile and test the solver. Compare the content of the `fireRADFoam_MaCFP` solver with the source code of `fireFoam`.

## 1.2 Cases

In this section the available projects are described.

### 1.2.1 Case 1: fireFOAM Benchmark in Quasi 1D Setup

For the case shown in Fig. 1.1 *fireFOAM* is benchmarked against an analytical solution. The aim of your project work is to create plots similar to the work done by [ge\\*comparison\\*2023](#). To understand the full context, read the cited paper (in the literature folder).

The domain is shown in Fig. 1.1 and consists of two infinite parallel plates, that are cold and black. Since *fireFOAM* neither knows what  $\infty$  is, nor what a true 2D domain is, a finite dimensional domain is chosen, with appropriate boundary conditions.

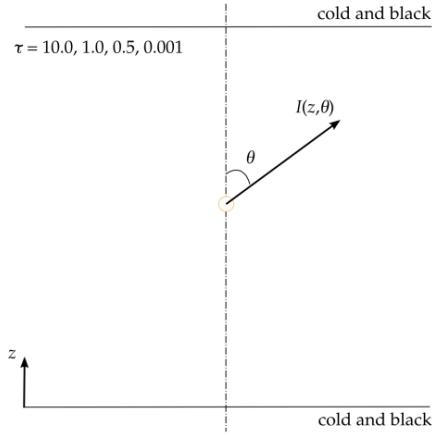


Abbildung 1.1: Quasi 1D setup.

The aim of the project is to document the simulation setup and recreate the two diagrams shown in Fig. 1.2 as close as possible and study the influence of different absorption coefficients, spatial and solid angular resolutions. The results may differ,

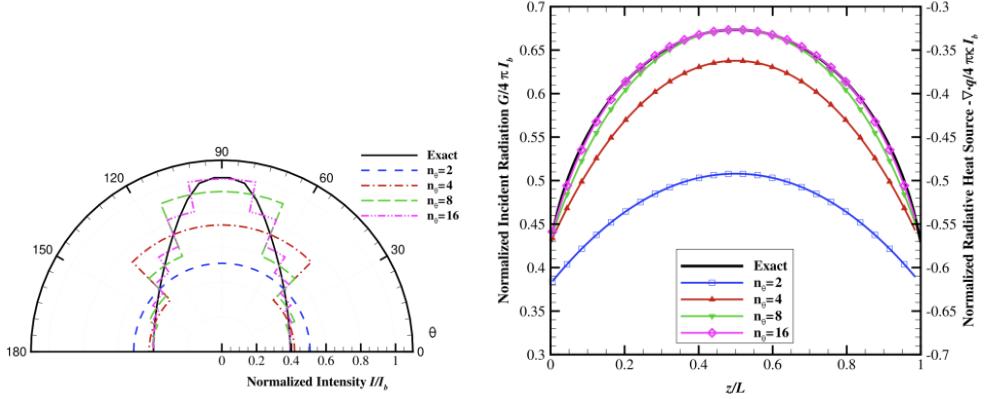


Abbildung 1.2: left: Directional distribution of normalized intensity. right: Normalized intensity along centerline.

since the temperature distribution in the paper is not given and we assume a constant temperature field.

The diagrams show simulations with respect to different solid angular resolutions. We compare the *fvDOM* solver with the analytical solution. The analytical solution is derived based on **modest•radiative•2013**[p.459] and looks like:

$$\begin{aligned} I^+(\tau, \theta) &= \frac{\sigma T_0^4}{\pi \cos(\theta)} \int_0^\tau e^{-(\tau-\tau')/\cos(\theta)} d\tau', \quad 0 < \theta < \frac{\pi}{2} \implies \\ \frac{I^+(\tau, \theta)}{I_b} &= 1 - e^{-\frac{\tau}{\cos(\theta)}}, \quad 0 < \theta < \frac{\pi}{2} \\ I^-(\tau, \theta) &= -\frac{\sigma T_0^4}{\pi \cos(\theta)} \int_\tau^{\tau_L} e^{(\tau'-\tau)/\cos(\theta)} d\tau', \quad \frac{\pi}{2} < \theta < \pi \implies \\ \frac{I^-(\tau, \theta)}{I_b} &= 1 - e^{\frac{-\tau+\tau_L}{\cos(\theta)}}, \quad \frac{\pi}{2} < \theta < \pi \end{aligned}$$

A constant temperature field was assumed to derive the equations above. To derive an equation for  $G$ , we assume symmetry with respect to the horizontal axis and receive:

$$\begin{aligned} \frac{G(\tau, \theta)}{I_b} &= \int_0^{2\pi} \left( \int_0^{\frac{\pi}{2}} \frac{I^+}{I_b} d\theta + \int_{\frac{\pi}{2}}^{\pi} \frac{I^-}{I_b} d\theta \right) d\phi = \left( \int_0^{\frac{\pi}{2}} \frac{I^+}{I_b} d\theta + \int_{\frac{\pi}{2}}^{\pi} \frac{I^-}{I_b} d\theta \right) 2\pi \\ &= \left( 2 \int_0^{\frac{\pi}{2}} \frac{I^+}{I_b} d\theta \right) 2\pi = 4\pi \int_0^{\frac{\pi}{2}} \frac{I^+}{I_b} d\theta = 4\pi \int_0^{\frac{\pi}{2}} 1 - e^{-\frac{\tau}{\cos(\theta)}} d\theta \\ \iff \frac{G(\tau, \theta)}{I_b 4\pi} &= \int_0^{\frac{\pi}{2}} 1 - e^{-\frac{\tau}{\cos(\theta)}} d\theta. \end{aligned}$$

For the last equation no closed analytical solution exists. It has to be solved numerically. Tasks:

1. After downloading the case ([here](#)) and compilation of the solver, apply `fireRADFoam_MaCFP` and check the log files and the results (*paraview*).
2. Use *paraview* to choose a meaningful perspective of your simulation domain, show the surface grid only and save a high resolution image (.png) with empty background to visualize the simulation domain for your report.
3. Find out which boundary conditions are used to setup the RTE and document the boundary conditions using the configuration files and the source code and the `lstlisting` environment to write code in `TeX`.
4. Use *matplotlib* to recreate the diagrams shown in Fig.[1.2](#) as close as possible. Note that *matplotlib* is fully customizable and allows you to control everything. Some *function objects* are already provided as a starting point to write data.
5. The influence of the absorption coefficient, given in terms of the optical thickness ( $\tau \in \{0.0001, 0.05, 0.1, 1, 10\}$ ), the resolution given in numbers of cells in  $x, y, z$  direction and the number of solid angles shell be examined. The number of polar ( $\theta$ ) and azimuthal ( $\phi$ ) angles is given in Tab. [1.1](#)

$\theta$	$\phi$	$N$
2	4	32
4	8	128
6	12	288
8	16	512

Tabelle 1.1: Number of azimuthal and polar angles and total number of solid angles.

For the grid sensitivity study adjust the resolution such that you have cube like cells (edge lengths equal). Then double the number of cells in each direction in four steps. Use the setup we built in the exercise for automatic case generation by first creating a base case in which the numbers you want to changed are replaced by unique tags. Use then copy and replacement operations to derive the cases. In total you should receive 80 different simulations.

6. Run the 80 simulations.

7. Create two proper dataframes. One that holds the information along the centerlines of all cases and one, that summarizes the information of the midpoint in all directions.
8. Plot the data into the diagrams. Choose meaningful selections of data (not all into one diagram).
9. Plot the analytical solution in both diagrams.
10. Put the plots and usefull information into the report and the code of your plots and your case generation into the Appendix A.

## 1.3 Implementation: Case 1

All tasks were completed more or less successfully. The following gives an overview of how the project files run, what changes are (and can be) made, the final results, and concluding remarks.

### 1.3.1 Introduction

The project is carried on from the Template provided, the changes are appropriately made. Some changes made to the template are due to the requirements demanded in the project, and others were made because of limitations of the system in which the simulation was done.

#### Domain Size and Resolution

1. Chosen Domain Size:  $9.6 \times 9.6 \times 1.2 \text{ m}^3$
2. First Resolution:  $24 \times 24 \times 3 \text{ m}^3$  & Final Resolution:  $192 \times 192 \times 24 \text{ m}^3$

The domain was reduced to a manageable size. Instead of the  $20 \times 20 \times 1.2 \text{ m}^3$ , which resulted in the coarsest grid to be:  $50 \times 50 \times 3$ . This was not desirable because due the requirement of the cell to have equal side lengths, as well as the grid should be made finer by doubling it 4 times, resulting in the finest grid to be:  $400 \times 400 \times 24$  resulting in more than 3.8 million cells.

Upon inspecting with various domain sizes and cell counts, it was understood that any more than a million cells proved one simulation case to run for at least a day. This didn't seem desirable since 20 cases would have to be run with such grid refinement. Finally, a domain length of  $9.6 \times 9.6 \times 1.2 \text{ m}^3$  was chosen with the

coarsest grid as 24x24x3 which would in its finest grid refinement would become 192 x 192 x 24 with 0.8 million cells

### 1.3.2 Project Files

The figure Fig. 1.3 shows how a fresh project looks like In a fresh project we only

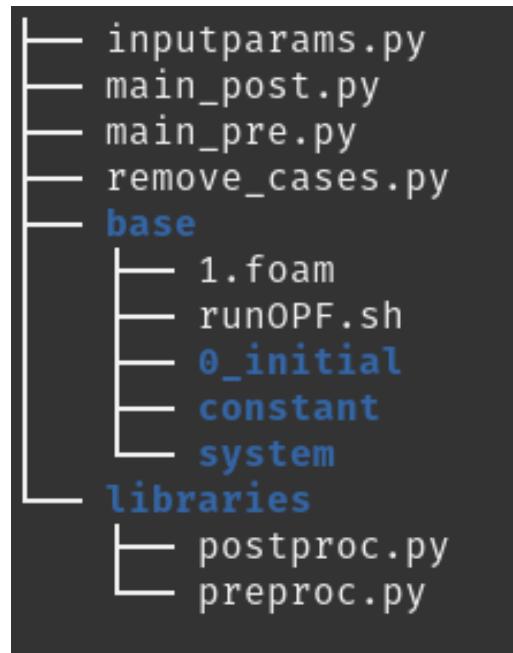


Abbildung 1.3

have a base folder, and some python files to automate all the simulations. The files `inputparams.py` includes all the initial parameters required to run the file `main_pre.py`, which is the preprocessing file responsible for creating copies of the base case folder into 80 different simulations based on the `theta`, `phi`, `tau`, and `grid resolution`.

Once `main_pre` has run, the following changes are made into our current directory, as shown in Fig. 1.4 We can see that the set of cases for theta and phi values are created. Within these directories, we can find the tau values and the grid refinement. Under each grid refinement case is the OpenFOAM base case with relevant values.

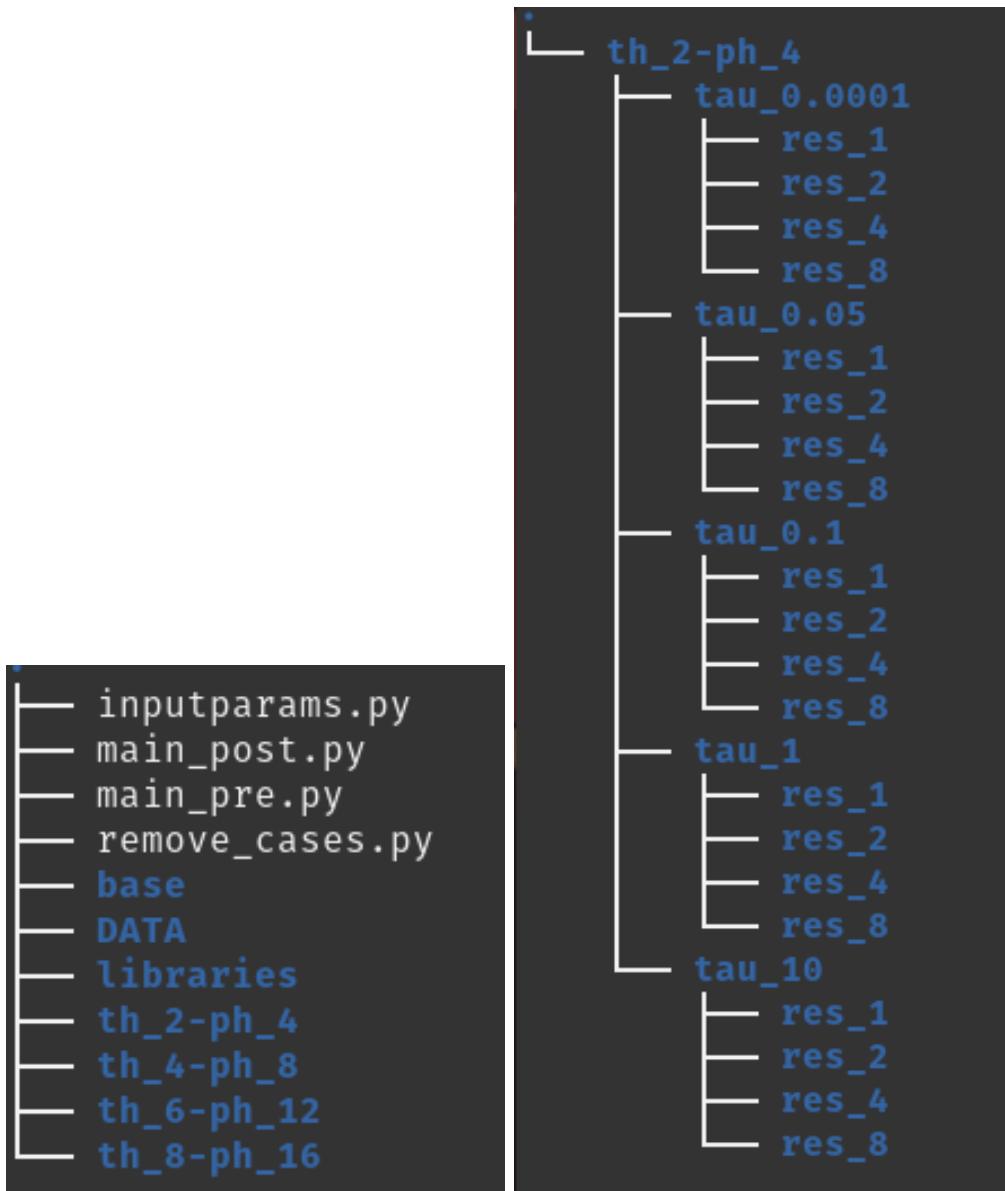


Abbildung 1.4

### 1.3.3 Simulation: Preprocessing

We will start by looking at the input parameters within the `inputparams.py`. This contains all of the required values for theta, phi, and tau based on the project. Furthermore, the first grid resolution is also defined, as well as the multiplier defined with the variable `lst_refining`.

All used functions can be found under the `libraries` folder in `pre_proc.py` and `post_proc.py` files.

Listing 1.1: `inputparams.py`

```
1 lst_theta = [2, 4, 6, 8]
2 lst_phi = [2*i for i in lst_theta] # [4, 8, 12, 16]
3 lst_tau = [0.0001, 0.05, 0.1, 1, 10] # optical thickness
4 lst_refining = [1,2,4,8] # doubled each time
5 total_cases = len(lst_theta)*len(lst_tau)*len(lst_refining)
6 def_cells = [24, 24, 3]
```

Once the user is satisfied with `inputparams.py`, they can run the simulations by executing `main_pre.py`. This file does the following:

1. creates subfolders for each case based on `theta`, `phi`, `tau`, and grid resolutions.
2. copies the content of the base case into each subfolder.
3. modifies the `theta`, `phi`, `tau`, and grid resolution for each case.
4. modifies the `controlDict` file for each for `ILambda.0_X` files to generate. This is needed to generate plots in the Postprocessing stage of the project.
5. run the solver for all cases.

Listing 1.2: `main_pre.py`

```
1 # creates subfolders for all cases
2 create_cases(lst_theta, lst_phi, lst_tau, lst_refining)
3 # modifies theta, phi, tau values for each case
4 mod_param(lst_theta, lst_phi, lst_tau, lst_refining)
5 # modifies the resolution for each case
6 mod_resolution(lst_theta, lst_phi, lst_tau, lst_refining, def_cells)
7 # modifies the number of ILambdas for probe
8 add_more_ILambdas(lst_theta, lst_phi, lst_tau, lst_refining)
9 # runs cases
10 run_cases(lst_theta, lst_phi, lst_tau, lst_refining)
```

## Paraview

While the cases are set up, we can view the domain using `paraFoam`.

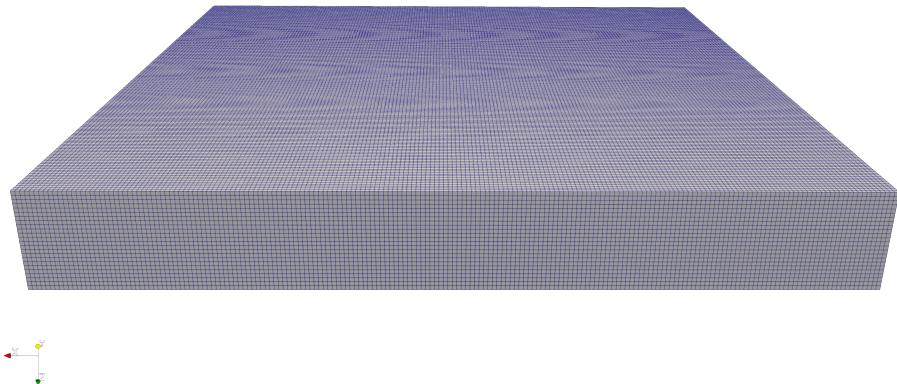


Abbildung 1.5: resolution: 192 x 192 x 24, all `res_8` cases

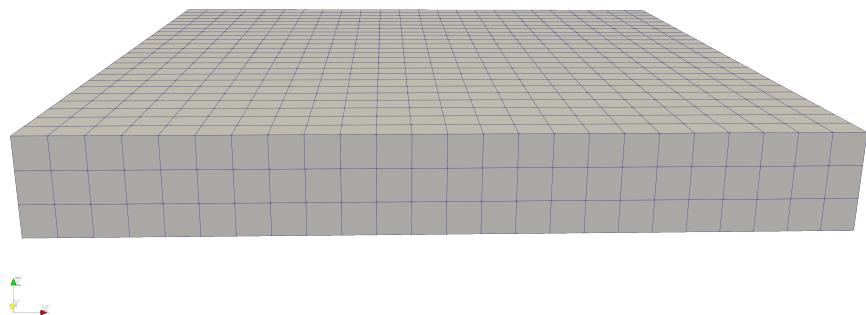


Abbildung 1.6: resolution: 24 x 24 x 3, all `res_1` cases

It can be seen that the cells have equal side lengths.

### 1.3.4 Simulation: Postprocessing

To gather the data and generate plots, we can execute `main_post.py`. Which does the following:

1. create a directory `DATA` and subfolders for relevant data from `postProcessing` folders of each case.

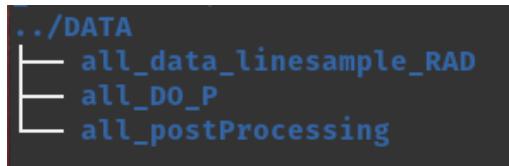


Abbildung 1.7

2. copies the contents of `D0_linesample_RAD.sets` and `D0_P.probes` to the subfolders in `DATA`
3. Use the data to generate plots

Listing 1.3: `main_post.py`

```
1 # MAIN PROGRAM
2 # creates subfolder for DATA and copies linesample_RAD files
3 gather_linesample_RAD(lst_theta, lst_phi, lst_tau, lst_refining)
4 # copies all Probe folders into DATA
5 gather_D0_P_dir(lst_theta, lst_phi, lst_tau, lst_refining)
6 # copies the entire postProcessing directory for each case into DATA
7 gather_postProcessing_dir(lst_theta, lst_phi, lst_tau, lst_refining)
8 # copies 08_log.rad files into DATA/all_D0_P
9 gather_directions(lst_theta, lst_phi, lst_tau, lst_refining)
```

### Plots

Once all the data has been gathered into the `DATA` directory, the plots are automatically generated upon the execution of `main_post.py`.

Listing 1.4: `main_post.py` plots

```
1 # PLOTS
2 lst_tau = [1]
3 lst_refining = [8]
4 plot_all_G(lst_theta, lst_phi, lst_tau, lst_refining)
5 plot_all_ILambda(lst_theta, lst_phi, lst_tau, lst_refining)
```

**Plot:**  $G/4\pi I_b$

This plot shows the normalized incident radiation  $G/4\pi I_b$  along the centreline of the domain:  $(0, 0, -0.6)$  to  $(0, 0, 0.6)$ . This can be extract from `postProcessing/D0_linesample_RAD.sets`. The generated plot for  $G/4\pi I_b$  is very similar (Fig. 1.8) to the plot provided in the literature (Fig. 1.2). However near the ends of the domain, one can observe 'spikes' (Fig. A.1) that have originated perhaps due to the shortening of the domain size resulting in errors.

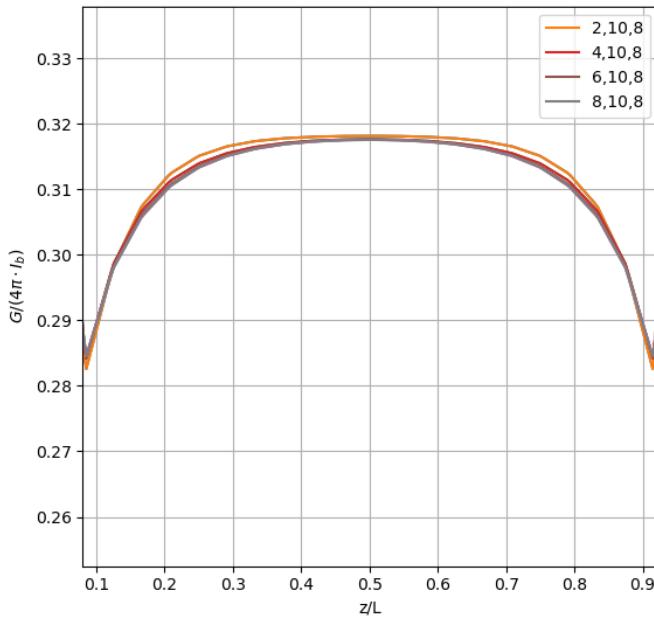


Abbildung 1.8: the labels are  $\theta$ ,  $\tau$ , and refinement respectively

This speculation is because of the initial requirement that the plates are **infinitely long**. This begs the question: how long must the domain be when modeling for this cases to get reasonable results with minimum error? It is clear now that a larger domain would've been a better choice, however much more computationally costly. The plots are constructed by extracting `D0_linesamples_RAD.sets`, the .csv files contains  $G$  along  $z$ . Which are used to construct the plots.

### Plot: $I/I_b$

This plot shows the normalized intensity  $I/I_b$  at a single centre point of the domain  $(0, 0, 0)$  from different  $\theta$  values. The first plot (Fig. 1.9) shows plot of the analytical solution provided in the literature. The next plot (Fig. A.2) shows the numerical

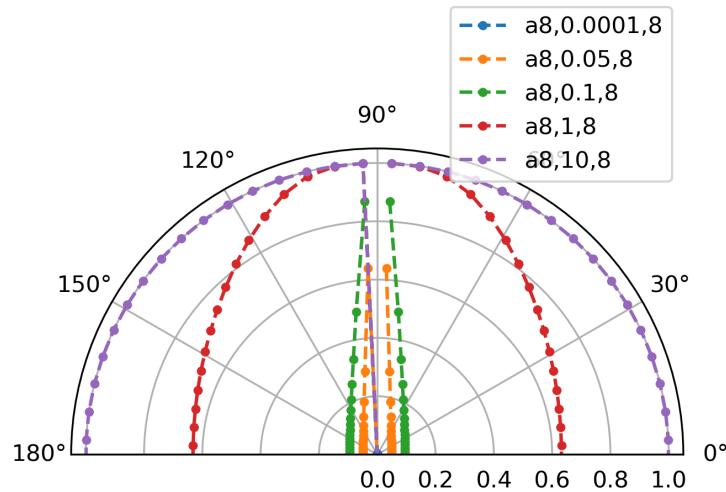


Abbildung 1.9: the labels indicate: a8: analytical solution for refinement 8,  $\theta$ ,  $\tau$ , and refinement respectively

results for a normalized intensity  $I/I_b$  obtained from openFOAM. However the plot is significantly different from the analytical results.

These plots were constructed by extracting the directions and magnitudes of each `ILambda_0_X` values. The directions are provided as 3D vector which needs to be converted to  $\theta$ ,  $\phi$  and  $\rho$ .

# Appendices

# A Appendix

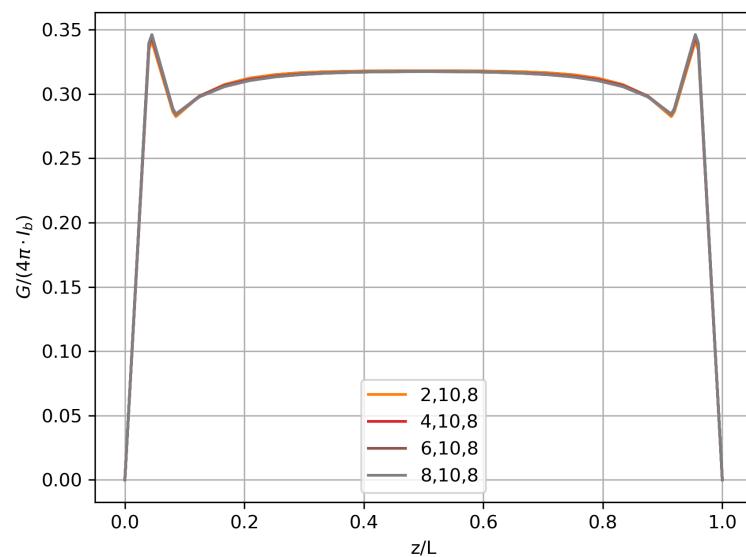


Abbildung A.1: Solution for  $G/4\pi I_b$  along the entire domain length  $z$ . The labels are  $\theta$ ,  $\tau$ , and refinement respectively

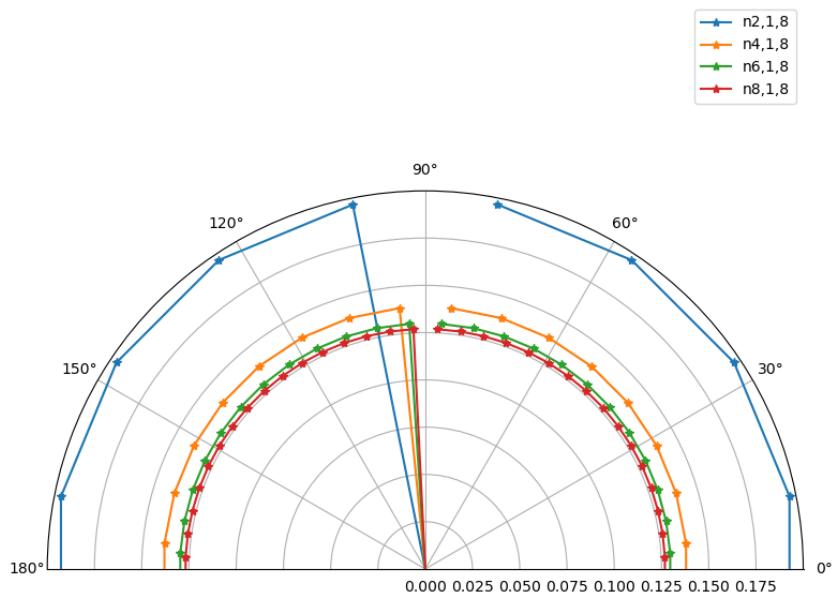


Abbildung A.2: Numerical Solution for  $I/I_b$ . The labels indicate: a8: analytical solution for refinement 8,  $\theta$ ,  $\tau$ , and refinement respectively

## Listings

1.1	inputparams.py	10
1.2	main_pre.py	10
1.3	main_post.py	12
1.4	main_post.py plots	12

# Abbildungsverzeichnis

1.1	Quasi 1D setup . . . . .	4
1.2	left: Directional distribution of normalized intensity. right: Normalized intensity along centerline. . . . .	5
1.3	. . . . .	8
1.4	. . . . .	9
1.5	resolution: 192 x 192 x 24, all <code>res_8</code> cases . . . . .	11
1.6	resolution: 24 x 24 x 3, all <code>res_1</code> cases . . . . .	11
1.7	. . . . .	12
1.8	the labels are $\theta$ , $\tau$ , and refinement respectively . . . . .	13
1.9	the labels indicate: a8: analytical solution for refinement 8, $\theta$ , $\tau$ , and refinement respectively . . . . .	14
A.1	Solution for $G/4\pi I_b$ along the entire domain length $z$ . The labels are $\theta$ , $\tau$ , and refinement respectively . . . . .	16
A.2	Numerical Solution for $I/I_b$ . The labels indicate: a8: analytical solution for refinement 8, $\theta$ , $\tau$ , and refinement respectively . . . . .	17