# Readme

Elena Demattè*, Juan J.L. Velázquez†

July 31, 2025

The Python code “code.py” is a numerical simulation of a one-dimensional radiative transfer equation. It has been tested with the Python’s version 3.10.12 using Jupiter Notebook and IDLE. The libraries required are NumPy and Matplotlib.

The aim of this simulation is to show numerically the behavior of the radiation intensity, solution to a one-dimensional radiative transfer equation, as the mean free-path of the photons tends to zero. For more information we refer to [1], for which this simulation was developed. To be more precise, we considered the following stationary problem for the integrated radiation intensity $J(x,n)$.

$$\begin{cases} n\partial_x J(x,n) = \frac{1}{\varepsilon}\left(\frac{1}{2}\int_{-1}^{1} J(x,n')\,dn' - J(x,n)\right) & x\in(0,1),\ n\in[-1,1],\\ J(0,n)=n & n>0,\\ J(1,n)=0 & n<0. \end{cases} \tag{1}$$

This problem is obtained integrating over all frequency the following radiative transfer equation for the radiation intensity $I_\nu(x,n)$

$$\begin{cases} n\cdot\nabla_x I_\nu(x_1,n_1) = \frac{1}{2\varepsilon}\left[B_\nu(T(x_1)) + \frac{1}{4\pi}\int_{\mathbb{S}^2} I_\nu(x_1,n_1')\,dn' - 2I_\nu(x_1,n_1)\right] & x\in(0,1)\times\mathbb{R}^2,\ n\in\mathbb{S}^2,\\ \operatorname{div}\int_0^\infty\int_{\mathbb{S}^2} nI_\nu(x_1,n_1)\,dn\,d\nu = 0 & x\in(0,1)\times\mathbb{R}^2,\ n\in\mathbb{S}^2,\\ I_\nu(0,n_1) = n_1e^{-\nu} & x\in\{0\}\times\mathbb{R}^2,\ n_1>0,\\ I_\nu(1,n_1)=0 & x\in\{1\}\times\mathbb{R}^2,\ n_1<0, \end{cases}$$

under the further assumption that $I_\nu(x,n) = I_\nu(x_1,n_1)$ and $T(x) = T(x_1)$. We remark that $T$ is the temperature and that $B_\nu(T)$ is the Planck distribution of the temperature, cf. [1]. Assuming in addition that $\varepsilon \ll 1$, we study the diffusion approximation regime. Moreover, we consider at $x\in\{0\}\times\mathbb{R}^2$ an anisotropic incoming boundary condition, i.e. $I_\nu(0,n_1) = n_1e^{-\nu}$ for $n_1>0$, and at $x\in\{1\}\times\mathbb{R}^2$ an isotropic incoming boundary value, namely $I_\nu(1,n_1)=0$ for $n_1<0$.

The aim of the numerical simulation of the problem (1) is to show that $J(x,n)$ becomes isotropic and linear at distances of order $\varepsilon$ form $x=0$ as $\varepsilon$ goes to 0. This corresponds to the formation of a boundary layer near $x=0$ and to the convergence of $J$ to the solution of the Laplace equation. Such behavior is exactly the one expected studying the problem (1) via the method of matched asymptotic expansions as obtained in [1].

The numerical approximate solution to (1) obtained in “code.py” is achieved discretizing the domain $[0,1]\times[-1,1]$ and solving the corresponding system of linear equations. Specifically, for $N>0$ we divide $[0,1]$ and $[-1,1]$ in $2N$ intervals and we define the grid

$$x_i = \frac{1+2i}{4N}\in(0,1) \quad\text{and}\quad n_j = \frac{1+2j}{2N} - 1 \in(-1,1)$$

for $i,j\in\{0,...,2N-1\}$. Applying the up-wind scheme for the transport term, using that $\delta x = x_{i+1}-x_i = \frac{1}{2N}$ and $\delta n = n_{j+1}-n_j = \frac{1}{N}$, we obtain the following system of linear equations

$$\begin{cases} 2Nn_j\left[J(x_{i+1},n_j) - J(x_i,n_j)\right] - \frac{1}{\varepsilon}\left[\frac{1}{2N}\sum_{k=0}^{2N-1} J(x_i,n_k) - J(x_i,n_j)\right] & j<N,\ i\neq 2N-1,\\ 2Nn_j\left[J(x_i,n_j) - J(x_{i-1},n_j)\right] - \frac{1}{\varepsilon}\left[\frac{1}{2N}\sum_{k=0}^{2N-1} J(x_i,n_k) - J(x_i,n_j)\right] & j\geq N,\ i\neq 0,\\ -4Nn_jJ(x_{2N-1},n_j) - \frac{1}{\varepsilon}\left[\frac{1}{2N}\sum_{k=0}^{2N-1} J(x_{2N-1},n_k) - J(x_{2N-1},n_j)\right] & j<N,\ i\neq 2N-1,\\ 4Nn_j\left[J(x_0,n_j) - n_j\right] - \frac{1}{\varepsilon}\left[\frac{1}{2N}\sum_{k=0}^{2N-1} J(x_0,n_k) - J(x_0,n_j)\right] & j\geq N,\ i=0, \end{cases}$$

*Institute for Applied Mathematics, University of Bonn, 53115 Bonn, Germany. *E-mail address:* `dematte@iam.uni-bonn.de`

†Institute for Applied Mathematics, University of Bonn, 53115 Bonn, Germany. *E-mail address:* `velazquez@iam.uni-bonn.de`

In “code.py” we computed the solution for $N = 70$ and $\varepsilon = 0.01$. Notice that $\frac{1}{140} = \delta x < \varepsilon = \frac{1}{100}$. Moreover, we plotted the resulting $J(x, n)$ for four different values of the direction $n_j$, namely for $j \in \{20, 55, 80, 130\}$.

# References


[1] E. Demattè and J. J. L. Velázquez, *Equilibrium and Non-Equilibrium diffusion approximation for the radiative transfer equation.* `https://arxiv.org/abs/2407.11797`, 2024.