A new AI based computation method of the Eddington factor in the M1-multigroup model

Gonzague Radureau, Claire Michaut

A new AI based computation method of the Eddington factor in the M1-multigroup model

Gonzague Radureau[a], Claire Michaut[a]

[a] *Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, Laboratoire Lagrange, Nice, 06300, France*

Abstract

The M1-multigroup model intricately captures the interaction between light and matter, incorporating the spectral behaviour of photons. However, a critical aspect of this model lies in determining the Eddington factor, which is used in the closure relation linking radiative pressure to radiative energy. Despite lacking an analytical expression, our investigation unveils that the Eddington factor depends solely on three parameters: the *radiative temperature*, the *reduced flux*, and the *frequency bounds' ratio* of a considered group. To address this challenge, we have devised a novel approach leveraging neural networks and polynomials, enabling rapid and accurate estimation of this quantity.

Our method showcases significant advantages over existing techniques. It demonstrates speeds up to 3,000 times faster than the most precise method utilising a line search algorithm and achieves precision levels up to 1,000 times higher than those relying on the M1-grey model's Eddington factor expression. Moreover, unlike interpolation-based methods, our approach eliminates the need for prior knowledge of radiative quantities. Consequently, our method emerges as one of the most efficient means to precisely compute the Eddington factor in the M1-multigroup model, offering a potent tool for advancing radiative hydrodynamics simulations.

Keywords: Radiation Hydrodynamics, Radiative Transfer, Artificial Intelligence, Closure Relation, Eddington Factor

1. Introduction

Radiative hydrodynamics models the coupling between a hypersonic hot plasma's dynamics and self-produced or external radiation. Almost all numerical codes use simplified models, such as the cooling functions or the diffusion approximation, that are, in most cases, limited in the general study of the interaction of photons with matter. To accurately model the photon transport, the HADES 2D code was specifically developed [1, 2, 3]. Such a code is indispensable for studying astrophysical objects, in which optically intermediate regions are still poorly modelled yet commonly encountered within such phenomena.

This code couples the hydrodynamics with the M1-grey and M1-multigroup models [4, 5, 6] for radiative transfer. The M1-multigroup model considers the spectral behaviour of light by partitioning the electromagnetic spectrum into groups. These models are based on the first two moments of the radiative transfer equations and require a closure relation between the second moment (radiative pressure) and the first (radiative energy) via the so-called Eddington tensor. This tensor depends on a scalar quantity called the Eddington factor [7, 8, 9], which has an analytical expression in the M1-grey model [10] but not in the M1-multigroup model. To compute the Eddington factor in the multigroup model, there exist three methods in the literature: the search algorithms [1], which are computationally expensive but precise; the interpolation of precalculated values [1, 11], which requires prior knowledge of the radiative quantities; and the expression of the M1-grey model [12], which the fastest method, but is not valid, as we will see in this paper.

Historically, the HADES code implemented two methods to compute the Eddington factor. The first method is based on search algorithms such as line search and Dichotomy-Newton's method [1], offers high precision but come with significant computational costs, thus limiting access to simulations using the M1-multigroup model. Additionally, this method occasionally fails to converge, resulting in instabilities within the simulations. The second method utilises interpolations [1, 11], which require prior knowledge of radiative quantities, a requirement that is not always met in radiative hydrodynamics simulations.

Neural networks have previously been employed in supernova numerical simulations to compute the Eddington tensor of the M1 model, which is used to model the interactions between neutrinos and matter [13]. Notably, the closure relation needs to have the required accuracy for this purpose. This study has shown the efficiency of neural networks in achieving reasonably precise results while keeping computational costs manageable. This finding underscores the potential of neural networks to significantly improve the accuracy and efficiency of supernova simulations compared to traditional methods.

In this paper, we introduce a new AI-based method developed within HADES, which computes precise values of the Eddington factor while effectively reducing computational expenses. First, we will analyse the dependencies of the Eddington factor in the M1-multigroup model (Section 2). Next, we will explore the intricacies of our devised approach (Section 3). Subsequently, we will conduct a comparative analysis, evaluating the performance of our method against other existing meth-

Email address: gonzague.radureau@oca.eu (Gonzague Radureau)

ods (Section 4). Finally, we will present our conclusions (Section 5). As the mathematical developments are complex, we detail the calculations in the seven appendices.

2. Analysis of the Eddington factor in the M1-multigroup model

We define all the variables used in Table 1.

2.1. Definition of the radiative quantities

The radiation is defined by its specific intensity $\mathcal{I}_\nu$, which depends on the frequency of the photons and their propagation direction. It reflects the quantity of energy transported by radiation per surface unit. One can derive respectively the radiative energy, the radiative flux and the radiative pressure as the two first moments of the specific intensity:

$$\mathrm{E}_\nu = \frac{1}{c}\int_{4\pi} \mathcal{I}_\nu(\mathbf{\Omega})\,\mathrm{d}\mathbf{\Omega} \quad ; \tag{1}$$

$$\mathbf{F}_\nu = \int_{4\pi} \mathbf{\Omega}\,\mathcal{I}_\nu(\mathbf{\Omega})\,\mathrm{d}\mathbf{\Omega} \quad ; \tag{2}$$

$$\mathbb{P}_\nu = \frac{1}{c}\int_{4\pi} \mathbf{\Omega}\otimes\mathbf{\Omega}\,\mathcal{I}_\nu(\mathbf{\Omega})\,\mathrm{d}\mathbf{\Omega} \quad . \tag{3}$$

Where c is the light velocity, ν is the photon frequency, $\mathbf{\Omega}$ is a given direction, and $\otimes$ is the dyadic product. The radiative pressure is related to the radiative energy via the Eddington tensor $\mathbb{D}_g$ according to the following expression:

$$\mathbb{P}_\nu = \mathbb{D}_\nu \mathrm{E}_\nu \quad . \tag{4}$$

According to the M1 model, which assumes that the radiation propagates symmetrically according to the direction of the radiative flux, the Eddington tensor is further related to a scalar quantity called the Eddington factor χ_ν. They are related according to the following relation:

$$\mathbb{D}_\nu = \frac{1-\chi_\nu}{2}\mathbb{I} + \frac{3\chi_\nu - 1}{2}\frac{\mathbf{F}_\nu\otimes\mathbf{F}_\nu}{\|\mathbf{F}_\nu\|^2} \quad . \tag{5}$$

Of course, it is impossible to consider the entirety of the electromagnetic spectrum, which is continuous. Thus there exist two models: one that averages the behaviour of photons in the electromagnetic spectrum (M1-grey model) and another that discretises the electromagnetic spectrum to achieve more precision (M1-multigroup model).

2.1.1. M1-grey model

The M1-grey model considers all the photons as a whole. Consequently, radiative quantities are integrated over all frequencies according to the following expressions:

$$\begin{cases} \mathrm{E_R} &= \int_0^\infty \mathrm{E}_\nu\,\mathrm{d}\nu \\ \mathbf{F_R} &= \int_0^\infty \mathbf{F}_\nu\,\mathrm{d}\nu \\ \mathbb{P}_\mathbb{R} &= \int_0^\infty \mathbb{P}_\nu\,\mathrm{d}\nu \end{cases} \quad . \tag{6}$$

In this case, the M1 closure relation (4) holds, and the Eddington tensor can still be expressed as a function of the Eddington factor following Eq. (5). In this case, the Eddington factor admits an analytical expression [10], depending solely on the reduced flux, which is defined by the relation:

$$\mathrm{f_R} = \frac{\|\mathbf{F_R}\|}{c\,\mathrm{E_R}} \quad . \tag{7}$$

Its values lie in [0, 1] and indicate the degree of anisotropy of the radiation ($\mathrm{f_R} = 0$: isotropic radiation, $\mathrm{f_R} = 1$: anisotropic radiation propagating in one direction). This expression is:

$$\chi_{grey} = \frac{3 + 4\mathrm{f}_g^2}{5 + 2\sqrt{4 - 3\mathrm{f}_g^2}} \quad . \tag{8}$$

We will refer to this as the *grey case Eddington factor*.

2.1.2. M1-multigroup model

By cutting it into groups, the M1-multigroup takes into account more precisely the spectral behaviour in the electromagnetic spectrum. We can thus define $\mathcal{G}$ groups, each having bound frequencies $[\nu_g, \nu_{g+1}]$. To each group are associated radiative quantities of each group defined as:

$$\forall g \in [1, \mathcal{G}], \quad \begin{cases} \mathrm{E}_g = \int_{\nu_g}^{\nu_{g+1}} \mathrm{E}_\nu\,\mathrm{d}\nu \\ \mathbf{F}_g = \int_{\nu_g}^{\nu_{g+1}} \mathbf{F}_\nu\,\mathrm{d}\nu \\ \mathbb{P}_g = \int_{\nu_g}^{\nu_{g+1}} \mathbb{P}_\nu\,\mathrm{d}\nu \end{cases} \quad . \tag{9}$$

Once again, Eq. (4) gives the closure relation of each group, and the Eddington tensor is related to the Eddington factor via Eq. (5). However, in this case, the Eddington factor lacks an analytical expression. To begin, we will determine the parameters on which it depends.

2.2. Dependencies of the Eddington factor in the M1-multigroup model

As seen in Eqs. (1), (2) and (3), we must first determine the expression of the specific intensity to compute the radiative quantities. By minimising the radiative entropy in the case of the M1-multigroup model [5], it can be expressed as:

$$\mathcal{I}_\nu(\mathbf{\Omega}) = \sum_{g=1}^{\mathcal{G}} \mathbb{1}_g(\nu)\,\mathcal{I}_{\nu,g}(\mathbf{\Omega}) \quad . \tag{10}$$

Where $\mathbb{1}_g(\nu)$ is the indicator function, worth 1 for $\nu \in [\nu_g, \nu_{g+1}]$ and 0 everywhere else. $\mathcal{I}_{\nu,g}$ is the specific intensity of the group g and can be expressed as:

$$\mathcal{I}_{\nu,g}(\mathbf{\Omega}) = \frac{2h\nu^3}{c^2}\left[\exp\left(\frac{h\nu}{k_B}(\alpha_{0,g} + \alpha_{1,g}\mathbf{n}_g\cdot\mathbf{\Omega})\right) - 1\right]^{-1} \quad . \tag{11}$$

Where h and k_B are, respectively, the Planck and Boltzmann constants, and $\mathbf{n}_g$ is the direction of the radiative flux of group g. If we introduce $\mathbf{n}_g\cdot\mathbf{\Omega} = \cos(\theta) = \mu$ and if we consider a frame in which $\theta = 0$ is the direction of the radiative flux, the radiative

Notation	Signification
$\mathcal{I}_\nu$	Spectral specific intensity (M1-multigroup model: Eq. 10)
$\mathcal{I}_{\nu,g}$	Spectral specific intensity of group g (Eq. 11)
ν_g, ν_{g+1}	Frequency bounds of group g
$\nu_{m,g}$	Geometric mean of group g's frequency bounds, $\nu_{m,g} = \sqrt{\nu_g\,\nu_{g+1}}$
δ_g	Group g's frequency bounds' ratio, $\delta_g = \nu_g/\nu_{g+1} \in [0,1]$
E_g	Radiative energy of group g (Eq. 9)
$\mathbf{F}_\mathrm{g}$	Radiative flux vector of group g (Eq. 9). Components: $\mathrm{F}_{g,x}$, $\mathrm{F}_{g,y}$
$\mathbb{P}_g$	Radiative pressure tensor of group g (Eq. 9). Components: $\mathbb{P}_{g,xx}$, $\mathbb{P}_{g,yy}$, $\mathbb{P}_{g,xy}$
$\mathcal{P}_g$	Component of the radiative pressure of group g in the direction of the radiative flux, $\mathcal{P}_g = \chi_g \mathrm{E}_g$
χ_{grey}, χ_g	Eddington factor of the M1-grey model (Eq. 8) and of group g in the M1-multigroup model
f_g	Reduced flux of group g, $\mathrm{f}_\mathrm{g} = \|\mathbf{F}_\mathrm{g}\|/(c\,\mathrm{E}_g) \in [0,1]$
T_g	Radiative temperature of group g, $\mathrm{T}_g = \left(\mathrm{E}_g/a_R\right)^{1/4}$
$\mathcal{T}_g$	Adimensionalised radiative temperatures using $\nu_{m,g}$ (Eq. 21)
$\mathcal{T}_g^{\nu_{g+1}}$	Adimensionalised radiative temperatures using ν_{g+1} (Eq. 23)
$\mathcal{T}_g^{\nu_g}$	Adimensionalised radiative temperatures using ν_g (Eq. 22)
$\mathcal{T}_g^s$	Shifted radiative temperature (see Appendix C)
$\alpha_{0,g}$, $\widetilde{\alpha_{0,g}}$	1st Lagrange multiplier and its adimensionalised form
$\alpha_{1,g}$, $\widetilde{\alpha_{1,g}}$	2nd Lagrange multiplier and its adimensionalised form
β_g	Anisotropy parameter, $\beta_g = \alpha_{1,g}/\alpha_{0,g} \in \,]-1,1[$

Table 1: Variables used in this article

quantities of group g are expressed as:

$$\mathrm{E}_g = \frac{2\pi}{c}\int_{\nu_g}^{\nu_{g+1}}\int_{-1}^{1}\frac{2h\nu^3/c^2}{e^{h\nu/k_B(\alpha_{0,g}+\alpha_{1,g}\mu)}-1}\,\mathrm{d}\nu\,\mathrm{d}\mu \;; \tag{12}$$

$$\|\mathbf{F}_\mathrm{g}\| = 2\pi\int_{\nu_g}^{\nu_{g+1}}\int_{-1}^{1}\frac{2h\nu^3\mu/c^2}{e^{h\nu/k_B(\alpha_{0,g}+\alpha_{1,g}\mu)}-1}\mathrm{d}\nu\,\mathrm{d}\mu \;; \tag{13}$$

$$\mathcal{P}_g = \frac{2\pi}{c}\int_{\nu_g}^{\nu_{g+1}}\int_{-1}^{1}\frac{2h\nu^3\mu^2/c^2}{e^{h\nu/k_B(\alpha_{0,g}+\alpha_{1,g}\mu)}-1}\mathrm{d}\nu\,\mathrm{d}\mu \;. \tag{14}$$

Where $\mathcal{P}_g$ is an eigenvector of the radiative pressure tensor and is related to the Eddington factor and the radiative energy as $\mathcal{P}_g = \chi_g \mathrm{E}_g$. $\|\mathbf{F}_\mathrm{g}\|$ is the norm of the radiative flux and $\alpha_{0,g}$ and $\alpha_{1,g}$ are called the *Lagrange multipliers* of the group g.
It has been demonstrated that if $(\mathrm{E}_g, \|\mathbf{F}_\mathrm{g}\|)$ is physically realistic, then there exist a unique couple $(\alpha_{0,g}, \alpha_{1,g})$ [5], which implies that the Eddington factor depends only on the radiative energy E_g, the norm of the radiative flux $\|\mathbf{F}_\mathrm{g}\|$ and the frequency bounds ν_g and ν_{g+1}. However, in the M1-grey model, we observed that the Eddington factor only depends on the reduced flux f_g, indicating that further analysis is required to identify the parameters significantly impacting the Eddington factor.

Supposing that $\nu_g > 0$ and $\nu_{g+1} < \infty$ if we do the change of variable $\tilde{\nu} = \nu/\nu_{m,g}$, where $\nu_{m,g}$ is the geometric mean of the group's frequency bounds. We obtain the following expressions:

$$\frac{k_B^4\,\mathrm{E}_g}{h^4\,\nu_{m,g}^4} = a_R\frac{15}{\pi^4}\int_{\delta_g^{1/2}}^{\delta_g^{-1/2}}\int_{-1}^{1}\frac{\tilde{\nu}^3}{e^{\tilde{\nu}(\widetilde{\alpha_{0,g}}+\widetilde{\alpha_{1,g}}\mu)}-1}\,\mathrm{d}\nu\,\mathrm{d}\mu \;; \tag{15}$$

$$\frac{k_B^4\,\|\mathbf{F}_\mathrm{g}\|}{h^4\,\nu_{m,g}^4} = a_R c\frac{15}{\pi^4}\int_{\delta_g^{1/2}}^{\delta_g^{-1/2}}\int_{-1}^{1}\frac{\tilde{\nu}^3\mu}{e^{\tilde{\nu}(\widetilde{\alpha_{0,g}}+\widetilde{\alpha_{1,g}}\mu)}-1}\,\mathrm{d}\tilde{\nu}\,\mathrm{d}\mu \;; \tag{16}$$

$$\frac{k_B^4\,\mathcal{P}_g}{h^4\,\nu_{m,g}^4} = a_R\frac{15}{\pi^4}\int_{\delta_g^{1/2}}^{\delta_g^{-1/2}}\int_{-1}^{1}\frac{\tilde{\nu}^3\mu^2}{e^{\tilde{\nu}(\widetilde{\alpha_{0,g}}+\widetilde{\alpha_{1,g}}\mu)}-1}\,\mathrm{d}\mu \;. \tag{17}$$

Where $\widetilde{\alpha_{i,g}} = \alpha_{i,g}\,h\nu_{m,g}/k_B$ are adimensionalised Lagrange multipliers, $\delta_g = \nu_g/\nu_{g+1}$ is the the group's frequency bounds' ratio, and a_R is the radiative constant. We then can express the radiative temperature, the reduced flux and the Eddington factor as:

$$\frac{k_B\,\mathrm{T}_g}{h\,\nu_{m,g}} = \left(\frac{15}{\pi^4}\int_{\delta_g^{1/2}}^{\delta_g^{-1/2}}\int_{-1}^{1}\frac{\tilde{\nu}^3}{e^{\tilde{\nu}(\widetilde{\alpha_{0,g}}+\widetilde{\alpha_{1,g}}\mu)}-1}\mathrm{d}\nu\,\mathrm{d}\mu\right)^{1/4} \;; \tag{18}$$

$$\mathrm{f}_g = \frac{\int_{\delta_g^{1/2}}^{\delta_g^{-1/2}}\int_{-1}^{1}\frac{\tilde{\nu}^3\mu}{e^{\tilde{\nu}(\widetilde{\alpha_{0,g}}+\widetilde{\alpha_{1,g}}\mu)}-1}\,\mathrm{d}\tilde{\nu}\,\mathrm{d}\mu}{\int_{\delta_g^{1/2}}^{\delta_g^{-1/2}}\int_{-1}^{1}\frac{\tilde{\nu}^3}{e^{\tilde{\nu}(\widetilde{\alpha_{0,g}}+\widetilde{\alpha_{1,g}}\mu)}-1}\,\mathrm{d}\tilde{\nu}\,\mathrm{d}\mu} \;; \tag{19}$$

$$\chi_g = \frac{\int_{\delta_g^{1/2}}^{\delta_g^{-1/2}}\int_{-1}^{1}\frac{\tilde{\nu}^3\mu^2}{e^{\tilde{\nu}(\widetilde{\alpha_{0,g}}+\widetilde{\alpha_{1,g}}\mu)}-1}\,\mathrm{d}\tilde{\nu}\,\mathrm{d}\mu}{\int_{\delta_g^{1/2}}^{\delta_g^{-1/2}}\int_{-1}^{1}\frac{\tilde{\nu}^3}{e^{\tilde{\nu}(\widetilde{\alpha_{0,g}}+\widetilde{\alpha_{1,g}}\mu)}-1}\,\mathrm{d}\tilde{\nu}\,\mathrm{d}\mu} \;. \tag{20}$$

From these expressions, it becomes evident that the Eddington factor in the M1-multigroup model is solely dependent on three quantities:

1. the *adimensionalised radiative temperature* $\mathcal{T}_g = k_B\, \mathrm{T}_g/(h\, \nu_{m,g})$: comparison of the group's radiation energy to the energy level of the photons,
2. the *reduced flux* f_g: degree of anisotropy of the radiation in group g. Its values are in [0, 1] ($\mathrm{f}_g = 0$: isotropic radiation, $\mathrm{f}_g = 1$: directional radiation),
3. the *frequency bounds' ratio* $\delta_g = \nu_g/\nu_{g+1}$: indication of the separation of the group's frequency bounds. Its values are in [0, 1] ($\delta_g \to 0$: frequency bounds far from each other, $\delta_g \to 1$: frequency bounds very close to each other).

One can show in the same way that we can choose the adimensionalised temperature between the three following possibilities[1]:

$$\mathcal{T}_g = k_B\, \mathrm{T}_g/(h\, \nu_{m,g}) \;; \tag{21}$$

$$\mathcal{T}_g^{\nu_g} = k_B\, \mathrm{T}_g/(h\, \nu_g) = \mathcal{T}_g/\sqrt{\delta_g} \;; \tag{22}$$

$$\mathcal{T}_g^{\nu_{g+1}} = k_B\, \mathrm{T}_g/(h\, \nu_{g+1}) = \mathcal{T}_g\, \sqrt{\delta_g} \;. \tag{23}$$

Depending on the context, it may be more appropriate to use one definition of the adimensionalised temperature over another. Therefore, we will always specify which adimensionalised radiative temperature we are using. For simplicity, we will refer to the adimensionalised radiative temperature simply as *radiative temperature* $\mathcal{T}$.

In Appendix H, we detail the cases $\nu_g = 0$ and $\nu_{g+1} \to \infty$. To efficiently compute the Eddington factor, we will analyse the significance of its dependence on all three parameters.

2.2.1. *Influence of the reduced flux*

First, we display curves of the Eddington factor χ_g as a function of the reduced flux f_g in Fig. 1, from which we draw two conclusions:

1. Whatever $\mathcal{T}$, χ_g always evolves from 1/3 at $\mathrm{f}_g = 0$ to 1 at $\mathrm{f}_g = 1$ and has a slope equal to 0 at $\mathrm{f}_g = 0$,
2. $\mathcal{T}$ only influences the evolution of χ_g at intermediate values of f_g.

This shows that χ_g always depends strongly on f_g. Let us further analyse the dependence on $\mathcal{T}$ and δ_g.

2.2.2. *Influence of the radiative temperature and the frequency bounds' ratio*

The ratio of the Eddington factor χ_g in the M1-multigroup model relative to its grey case expression is displayed as a function of the radiative temperature $\mathcal{T}_g$ using different frequency bounds' ratios δ_g (Fig. 2). Qualitatively when the group's frequencies are significantly different ($\delta_g << 1$), we observe the existence of a central "plateau" lying in the domain (Fig. 2a). As the frequencies get closer together, the size of this "plateau" diminishes until it disappears (Fig. 2b, 2c). When the frequencies are nearly identical, two distinct domains emerge: one at low and one at high temperatures. The transition between these domains occurs at lower radiative temperatures as the frequencies become closer ($\delta_g \to 1$, Fig. 2d). Overall, three main domains can be seen in Fig. 2:

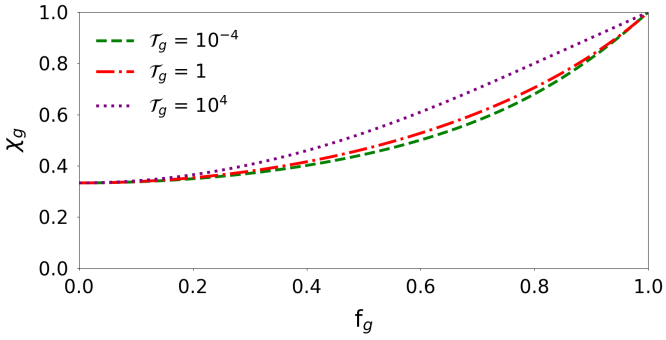


Figure 1: Evolution of the Eddington factor χ_g at different radiative temperatures $\mathcal{T}_g$ (Eq. 21), using a frequency bounds' ratio worth $\delta_g = 10^{-4}$.

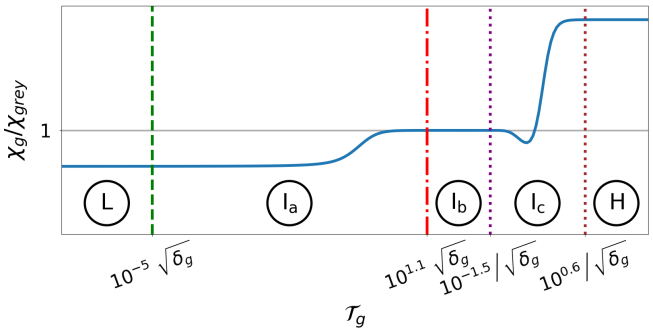
1. **Domain L:** Low radiative temperature. In this domain, $\mathcal{T}_g << \sqrt{\delta_g}$. χ_g depends solely on f_g. A method to compute it is detailed in Appendix A,
2. **Domain I:** Intermediate radiative temperatures. χ_g depends strongly on the $\mathcal{T}_g$, f_g and δ_g,
3. **Domain H:** High radiative temperature. In this domain, $\mathcal{T}_g >> 1/\sqrt{\delta_g}$. χ_g depends solely on f_g. A method to compute it is detailed in Appendix B.

The intermediate domain I, based on the separation of the group's frequencies, can be further categorised as follows:

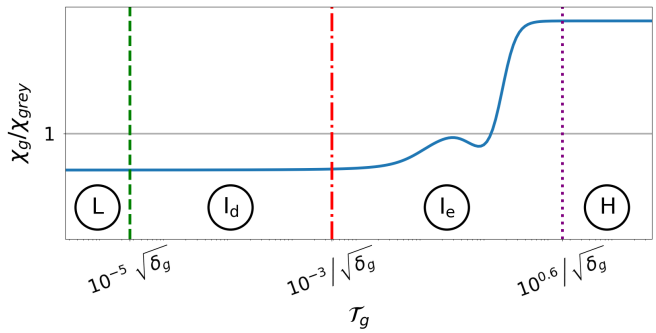
- **Case $\delta_g \to 0$:** Bound frequencies far from each other. There exist three intermediate domains (Fig. 2a):
  - $\mathrm{I_a}$: χ_g strongly depends on $\mathcal{T}$. Empirically, we have noticed χ_g only depends on f_g and $\mathcal{T}_g^{\nu_g}$,
  - $\mathrm{I_b}$: corresponds to the domain where the grey case Eddington factor is valid. The width of this domain is related to δ_g,
  - $\mathrm{I_c}$: χ_g strongly depends on $\mathcal{T}$. χ_g only depends on f_g and $\mathcal{T}_g^{\nu_{g+1}}$,
- **Intermediate δ_g:** In the domains $\mathrm{I_d}$ and $\mathrm{I_e}$, χ_g depends on $\mathcal{T}_g$, f_g and δ_g (Fig. 2b, 2c),
- **Case $\delta_g \to 1$:** Bound frequencies close to each other. In the domain $\mathrm{I_f}$, we observed that χ_g depends only on f_g and a *shifted radiative temperature* $\mathcal{T}_g^s$ (Fig. 2d). This shift accounts for the transition domains L to H moving towards lower radiative temperatures (see Appendix C).

We have chosen to use neural networks to efficiently compute the Eddington factor χ_g in domains where it depends on two or more parameters. In the next section, we detail our computation methodology.

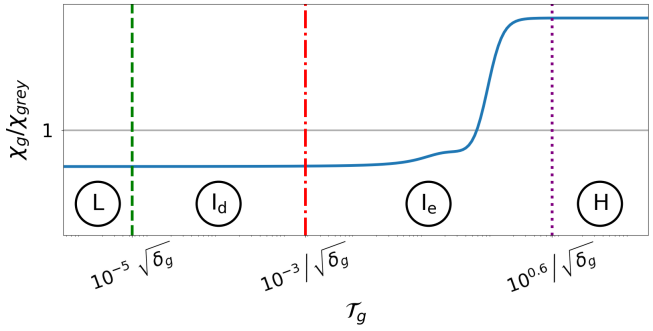
[1] The choice of $\mathcal{T}_g$ comes from the calculations we present in this paper. The choice of $\mathcal{T}_g^{\nu_g}$ and $\mathcal{T}_g^{\nu_{g+1}}$ comes from analogous calculations, taking $\tilde{\nu} = \nu/\nu_g$ and $\tilde{\nu} = \nu/\nu_{g+1}$, respectively.



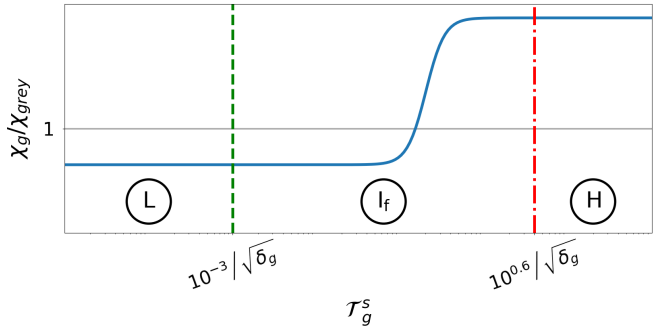
(a) $\delta_g = 1.10^{-4}$. L, H: Asymptotic domains, $\chi_g = \mathcal{F}(\mathrm{f}_g)$; $\mathrm{I_a}$: Low-intermediate radiative temperatures, $\chi_g = \mathcal{F}(\mathrm{f}_g, \mathcal{T}_g^{\gamma_g})$; $\mathrm{I_b}$: Domain in which χ_{grey} is valid; $\mathrm{I_c}$: High-intermediate radiative temperatures, $\chi_g = \mathcal{F}(\mathrm{f}_g, \mathcal{T}_g^{\gamma_{g+1}})$.



(b) $\delta_g = 7.10^{-2}$. L, H : Asymptotic domains, $\chi_g = \mathcal{F}(\mathrm{f}_g)$; $\mathrm{I_d}$: Low-intermediate radiative temperatures, $\chi_g = \mathcal{F}(\mathrm{f}_g, \mathcal{T}_g^{\gamma_{g+1}})$; $\mathrm{I_e}$: High-intermediate radiative temperatures, $\chi_g = \mathcal{F}(\mathrm{f}_g, \mathcal{T}_g^{\gamma_{g+1}}, \delta_g)$.



(c) $\delta_g = 7.10^{-2}$. L, H : Asymptotic domains, $\chi_g = \mathcal{F}(\mathrm{f}_g)$; $\mathrm{I_d}$: Low-intermediate radiative temperatures, $\chi_g = \mathcal{F}(\mathrm{f}_g, \mathcal{T}_g^{\gamma_{g+1}})$; $\mathrm{I_e}$: High-intermediate radiative temperatures, $\chi_g = \mathcal{F}(\mathrm{f}_g, \mathcal{T}_g^{\gamma_{g+1}}, \delta_g)$.



(d) $\delta_g = 0.99$. L, H: Asymptotic domains, $\chi_g = \mathcal{F}(\mathrm{f}_g)$; $\mathrm{I_f}$: Intermediate radiative temperatures, $\chi_g = \mathcal{F}(\mathrm{f}_g, \mathcal{T}_g^s)$. $\mathcal{T}_g^s$ is the shifted radiative temperature (see Appendix C).

Figure 2: Ratio of the grey case Eddington factor on the multigroup Eddington factor at $\mathrm{f}_g = 0.65$ and at different values of frequency bounds' ratio δ_g.

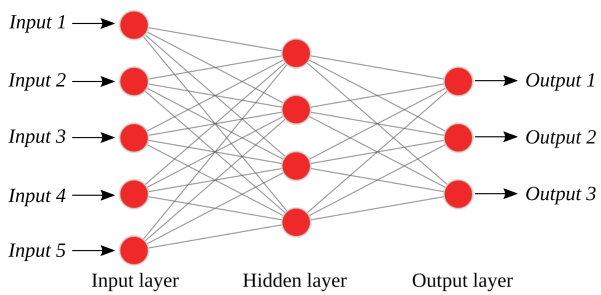


Figure 3: Multi-Layer Perceptron comprising 5 inputs and 3 outputs. It contains one *input layer*, one *hidden layer* and one *output layer*. The circles represent perceptrons.

3. Details on the AI method

3.1. A short introduction to supervised neural networks

In this work, we use the neural network architecture *Multi-Layer Perceptron* (MLP). It is composed of units called *perceptrons*. Each of these perceptrons takes a number n of inputs x_1, x_2, ..., x_n to which are associated *weights* w_1, w_2, ..., w_n and a *bias* b and returns an output $y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$, where f is called *activation function* and is a non-linear function.

Several perceptrons can then be arranged in *layers*. There are different types of layers: the *input layer* receives the input data, the *output layer* generates the output data, and the *hidden layers* are the layers between the input and the output layers (see Fig. 3).

The first thing one must do is to define the architecture of the neural network: the number of layers, the number of perceptrons, the activation function functions used at each layer, ... Once this is done, one must *train* the neural network for it to solve a given problem. This is done by finding the optimal values of the weights and biases of each perceptron in the neural network so that it outputs the correct answers to several known examples. This requires choosing a function that will evaluate the error of the neural network, which is called the *cost function*. Finally, one must select the *optimiser* that updates the weights and biases that minimise the cost function. The number of iteration steps during the optimisation phase is called *epoch*.

One splits the data into two sets: the *training set*, which is the set of known examples on which one trains the neural network and the *test set*, that enables to check if the neural network has correctly learnt.

3.2. Limits of the domains

As we have seen in Sec. 2, three main domains exist: the low radiative temperature domain L, several intermediate domains I and the high radiative temperature domain H. There are different sub-domains in the domain I, in which the Eddington factor depends more or less on the radiative temperature and the frequency bounds (domains $\mathrm{I_a}$, $\mathrm{I_b}$, $\mathrm{I_c}$, $\mathrm{I_d}$, $\mathrm{I_e}$ and $\mathrm{I_f}$).

In machine learning, there are two possible approaches: using a large neural network to predict the Eddington factor for all cases or using smaller, specialised neural networks to compute the Eddington factor in specific domains. We have chosen the latter approach to limit the size of the neural networks required, thereby reducing computation time. Table 2 summarises the methods used in the different domains, discussed in Sec. 1. Here are the names of the neural networks are used for specific domains:

- $\mathrm{MLP_a}$: For domain $\mathrm{I_a}$. It has two inputs,
- $\mathrm{MLP_c}$: For domain $\mathrm{I_c}$. It has two inputs,
- $\mathrm{MLP_{e,1}}$ and $\mathrm{MLP_{e,2}}$: For domain $\mathrm{I_e}$. We use two neural networks to reduced the size required and thus the prediction time. $\mathrm{MLP_{e,1}}$ is used for $\delta_g \in\ ]0.03, 0.1]$ and $\mathrm{MLP_{e,2}}$ for $\delta_g \in\ ]0.1, 0.9]$. These two neural networks have three inputs,
- $\mathrm{MLP_f}$: For domain $\mathrm{I_f}$. It has two inputs.

For domain $\mathrm{I_d}$, we use $\mathrm{MLP_a}$ that only takes two parameters and predicts accurate Eddington factors. For all these neural networks, we use normalised inputs and impose known constraints on the output (Appendix D).

	Range δ_g		Range $\mathcal{T}_g$		Method **
	Min	Max	Min	Max	
L	0	$10^{-2.6}$	$-\infty$	$10^{-5}\sqrt{\delta_g}$	P
	$10^{-2.6}$	0.9	$-\infty$	1	In: $\mathrm{f_g}$
	0.9	1	$-\infty$ *	$10^{-3}/\sqrt{\delta_g}$ *	
H	0	$10^{-2.6}$	$10^{0.6}/\sqrt{\delta_g}$	$+\infty$	P
	$10^{-2.6}$	0.9	1	$+\infty$	In: $\mathrm{f_g}$
	0.9	1	$10^{0.6}/\sqrt{\delta_g}$ *	$+\infty$ *	
$\mathrm{I_a}$	0	0.03	$10^{-5}\sqrt{\delta_g}$	$10^{1.1}\sqrt{\delta_g}$	MLP In: $\mathrm{f_g}$, $\mathcal{T}_g^{\nu_g}$
$\mathrm{I_b}$	0	$10^{-2.6}$	$10^{1.1}\sqrt{\delta_g}$	$10^{-1.5}/\sqrt{\delta_g}$	χ_{grey}
$\mathrm{I_c}$	0	0.03	$10^{-3}/\sqrt{\delta_g}$	$10^{0.6}/\sqrt{\delta_g}$	MLP In: $\mathrm{f_g}$, $\mathcal{T}_g^{\nu_{g+1}}$
$\mathrm{I_d}$	0.03	0.9	$10^{-5}/\sqrt{\delta_g}$	$10^{-3}/\sqrt{\delta_g}$	MLP In: $\mathrm{f_g}$, $\mathcal{T}_g^{\nu_{g+1}}$
$\mathrm{I_e}$	0.03	0.9	$10^{-3}/\sqrt{\delta_g}$	$10^{0.6}/\sqrt{\delta_g}$	MLP In: $\mathrm{f_g}$, δ_g, $\mathcal{T}_g^{\nu_{g+1}}$
$\mathrm{I_f}$	0.9	1	$10^{-3}/\sqrt{\delta_g}$ *	$10^{0.6}/\sqrt{\delta_g}$ *	MLP In: $\mathrm{f_g}$, $\mathcal{T}_g^{s}$

*Min/Max values for the shifted temperature $\mathcal{T}_g^s$ (see Appendix C).

**P: polynomial, MLP: neural network, χ_{grey}: grey case Eddington factor. The variables after "In:" are the inputs used for each method.

Table 2: Limits for each domain.

3.3. Training/validation data

We have generated training and test datasets using the line search algorithm we have developed (Appendix G). Table 3 summarises the parameters used to generate these datasets. Linear spacing has been used for all the quantities. We draw the reader's attention to the fact that the amount of data expected does not correspond to the amount of data obtained because the line search algorithm does not always converge.

		Training set	Test set	Range
MLP_a	$\log_{10}(\mathcal{T}_g^{\nu_g})$	500	1,000	[-5, 1.1]
	f_g	100	1,000	[0, 1]
	# data	49,695	993,372	-
MLP_c	$\log_{10}(\mathcal{T}_g^{\nu_{g+1}})$	300	1,000	[-1.5, 0.6]
	f_g	100	1,000	[0, 1]
	# data	28,621	951,821	-
$\mathrm{MLP}_{e,1}$	$\log_{10}(\mathcal{T}_g^{\nu_{g+1}})$	100	150	[-5, 0.6]
	f_g	100	100 *	[0, 1]
	δ_g	60	100	[0.03, 0.1]
	# data	590,160	1,467,561	-
$\mathrm{MLP}_{e,2}$	$\log_{10}(\mathcal{T}_g^{\nu_{g+1}})$	100	150	[-5, 1.1]
	f_g	100	100 *	[0, 1]
	δ_g	60	100	[0.1, 0.9]
	# data	589,910	1,465,462	-
MLP_f	$\log_{10}(\mathcal{T}_g^{s})$	300	1,000	[-3, 0.6]
	f_g	100	1,000	[0, 1]
	# data	28,712	955,143	-

*The range $\mathrm{f}_g \in [10^{-5}, 0.99999]$ has instead been used in the test sets of $\mathrm{MLP}_{e,1}$ and $\mathrm{MLP}_{e,2}$, to change the values of f_g.

Table 3: Description of the datasets.

3.4. Architectures of the neural networks

All neural networks were trained using Julia's Flux library[2], known for its versatility in neural network modelling. We selected architectures large enough to provide precise predictions yet small enough to ensure quick computations. The architecture of these MLPs is detailed in Table 4. Our selection methodology is explained in Appendix E.

3.5. Validation

Finally, we check the relative error of the MLPs' prediction on the test set (see Sec. 3.3). Table 5 summarises the average relative error on the Eddington factor and the components of the radiative pressure. These errors are low, which demonstrates the generalisability of these MLPs. The relative error can reach

[2] https://fluxml.ai/Flux.jl/stable/

Neural network	Total # of layers	Total # of perceptrons	# of parameters
MLP_a	3	15	77
MLP_c	3	15	77
$\text{MLP}_{e,1}$	4	24	146
$\text{MLP}_{e,2}$	3	19	116
MLP_f	3	15	77

Table 4: Architecture of the MLPs used.

100 % on the radiative pressure component $\mathbb{P}_{g,xy}$ for very low values of f_g, which is not an issue since this component does not have a significant impact on the physics in such domain, due to its low value compared to the other components of the radiative pressure.

	χ_g	$\mathbb{P}_{g,xx,yy}$	$\mathbb{P}_{g,xy}$
MLP_a	3.3×10^{-3} %	6.9×10^{-3} %	2.2×10^{-2} %
MLP_c	8.8×10^{-3} %	2.4×10^{-2} %	3.7×10^{-2} %
$\text{MLP}_{e,1}$	8.2×10^{-3} %	1.8×10^{-2} %	4.1×10^{-2} %
$\text{MLP}_{e,2}$	2.0×10^{-2} %	3.7×10^{-2} %	9.8×10^{-2} %
MLP_f	1.7×10^{-2} %	3.9×10^{-2} %	8.5×10^{-2} %

Table 5: Average relative errors of each MLP on the test set. The average errors on the components of the radiative pressure are calculated using Eqs. (F.7), (F.8) and (F.9) in Appendix F.

3.6. Utilisation in the code HADES

To exploit the trained neural networks, we have converted our Julia models to Keras models[3] and then translated these into C functions with the library keras2c[4]. The C functions are read by the code HADES, written in Fortran 90 (upgraded to version 2003).

4. Validation: 1D Marshak wave

To evaluate this new method's efficiency, we simulated a 1D Marshak wave and compared it to different methods for computing the Eddington factor.

4.1. Simulation description: Marshak wave

We consider a box of size $L = 0.1$ m for 1,000 cells, in which there is a fluid at rest, having a temperature of $T = 300$ K and a density of $\rho = 8.2\times10^{-9}$ kg/m^3. On its left boundary, the fluid experiences heating from an imposed temperature of $T_l = 5780$ K, accompanied by an influx of radiative flux governed by the following expression:

$$\forall g \in [\![1,\mathcal{G}]\!],\ \forall x \in \mathbb{R}_-^*,\ \mathrm{F}_{g,x}(x) = c\left(\mathrm{E}_g(x) - \mathrm{E}_g(0)\right)\ ; \quad (24)$$

where $\mathcal{G}$ is the total number of groups. Initially, the radiation is in equilibrium with the gas. The gas under consideration possesses an atomic mass of $m_a = 1$ u and an adiabatic index of $\gamma = 5/3$. The simulation spans a duration of 13.3 ns of physical time, during which various configurations of group numbers, as detailed in Table 6, are employed. To compute the Eddington factor, we considered four methods:

1. the line search algorithm described in Appendix G,
2. using the grey case Eddington factor (Eq. 8),
3. the AI method we have described in this paper,
4. the interpolation of the Eddington factor from precalculated values [1].

Number of groups		ν_g [eV]	ν_{g+1} [eV]	δ_g
2 groups	- group 1	0	1.0×10^{-1}	-
	- group 2	1.0×10^{-1}	∞	-
3 groups	- group 1	0	1.0×10^{-3}	-
	- group 2	1.0×10^{-3}	1.0×10^{-1}	1.0×10^{-2}
	- group 3	1.0×10^{-1}	∞	-
4 groups	- group 1	0	1.0×10^{-3}	-
	- group 2	1.0×10^{-3}	2.0×10^{-1}	5.0×10^{-2}
	- group 3	2.0×10^{-1}	4.0×10^{-1}	5.0×10^{-1}
	- group 4	4.0×10^{-1}	∞	-
5 groups	- group 1	0	1.0×10^{-3}	-
	- group 2	1.0×10^{-3}	1.0×10^{0}	1.0×10^{-3}
	- group 3	1.0×10^{0}	1.1×10^{0}	9.1×10^{-2}
	- group 4	1.1×10^{0}	1.0×10^{1}	1.1×10^{-2}
	- group 5	1.0×10^{1}	∞	-

Table 6: Definition of the groups

4.2. Computation time of different methods

Number of groups	Grey case expression	AI method	Interpolation	Line search
2 groups	3.11 min	4.18 min	3.70 min	3.58 days
3 groups	3.92 min	4.92 min	4.69 min	7.05 days
4 groups	4.55 min	10.37 min	5.65 min	22.6 days
5 groups	5.53 min	7.72 min	6.88 min	16.6 days

Table 7: Comparison of the simulation times (CPU time).

Table 7 presents a comprehensive comparison of simulation times across different multigroup configurations. Our developed method demonstrates remarkable efficiency, outperforming the line search algorithm by 1,000 to 3,000 times in computation time. However, it is slightly slower than the method using the grey case expression, *i.e.* the fastest method. by a factor of 1 to 3, its computation speed remains comparable to the fastest methods available.
It is worth noting that simulation times for the 4-group configuration are prolonged compared to other setups. This delay is attributed to utilising heavier neural networks, as seen in Sec. 3.4,

[3] https://keras.io
[4] https://f0uriest.github.io/keras2c/index.html

which necessitates more time for predictions. Consequently, our developed method exhibits accelerated performance, particularly evident with fewer groups avoiding the use of $\mathrm{MLP}_{e,1}$ and $\mathrm{MLP}_{e,2}$, which are heavier.

4.3. Accuracy

At time t=0.133 ns, we present the radiative pressure $\mathbb{P}_{R,xx}$ obtained using the four different methods for computing the Eddington factor. The simulation using the line search method is designated as the reference due to its superior accuracy. Analysis of the resulting curves reveals that the AI method exhibits errors of 10^{-3} %, whereas the grey case expression can yield errors up to 2.5 %. Additionally, the error in radiative pressure can reach 0.1 %, depending on the set of precalculated Eddington factors used. This disparity highlights the enhanced precision of our new method. Furthermore, we observe that the error associated with the grey case expression and the interpolation method increases as the number of groups increases.

5. Conclusion

To conclude, we have obtained a neural network approach that provides an average error on the prediction of the Eddington factor in the M1-multigroup case of 10^{-3} % to 10^{-2} % and is faster by a factor 3,000 compared to a line search algorithm and only slower by a factor 1 to 3 compared the using the grey case Eddington factor. Moreover, this approach does not need prior knowledge of the radiative energy and reduced flux, which is an advantage compared to the method using interpolations of precalculated Eddington factors, making it one of the most efficient existing.

Yet, the simulations we have done so far have not revealed significant differences in the hydrodynamics quantities between the different methods, even using many groups or 2D simulations. The impact of the grey case Eddington factor's lack of precision could yet prove decisive in the case where the radiation dominates, and the reduced flux is close to 1.

However, we propose this method to serve as a reference to future work using the M1-multigroup, as it renders precise estimation of the Eddington factor computationally accessible.

Acknowledgements

This work was supported by a French government grant managed by the Agence Nationale de la Recherche under the Investissement d'avenir program, reference ANR-19-P3IA-0002.

Declaration of generative AI and AI-assisted technologies in the writing process.

During the preparation of this work we used ChatGPT and Grammarly in order to improve the clarity of the article and correct the grammar. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Appendix A. Low-temperature approximation for the Eddington factor (Domain L)

We have fitted a polynomial of order 11 at $\mathcal{T}_g^{\nu_g} = 10^{-8}$, using Julia's least square regression library LsqFit[5]. The resulting expression is:

$$\begin{aligned}
\mathrm{P_L(f}_g) =& 1/3 \;+\; 2\mathrm{f}_g^2/3 + \\
& 0.3058945350580\,(\mathrm{f}_g^3 - \mathrm{f}_g^2) \;- \\
& 3.5960491961545\,(\mathrm{f}_g^4 - \mathrm{f}_g^2) \;+ \\
& 24.1130825450229\,(\mathrm{f}_g^5 - \mathrm{f}_g^2) \;- \\
& 92.2832025267787\,(\mathrm{f}_g^6 - \mathrm{f}_g^2) \;+ \\
& 220.1853600902954\,(\mathrm{f}_g^7 - \mathrm{f}_g^2) \;- \\
& 329.3727903123292\,(\mathrm{f}_g^8 - \mathrm{f}_g^2) \;+ \\
& 299.9957746394199\,(\mathrm{f}_g^9 - \mathrm{f}_g^2) \;- \\
& 151.3384873479037\,(\mathrm{f}_g^{10} - \mathrm{f}_g^2) + \\
& 32.2668624654447\,(\mathrm{f}_g^{11} - \mathrm{f}_g^2) \;\;.
\end{aligned} \tag{A.1}$$

The form of this polynomial ensures that the predicted Eddington factor is 1/3 at $\mathrm{f}_g = 0$, 1 at $\mathrm{f}_g = 1$ and that the slope is zero at $\mathrm{f}_g = 0$. The maximal relative error is 10^{-5} % on the Eddington factor and 10^{-3} % on the radiative pressure components.

Appendix B. High-temperature approximation for the Eddington factor (Domain H)

If we consider the Lagrange multiplier α_0 to be $\alpha_0 << k_B/(h\nu_{g+1})$ the specific intensity of group g can express $\mathcal{I}_{\nu,g}$ as:

$$\mathcal{I}_{\nu,g}(\mu) = \frac{2k_B}{c^2}\frac{\nu^2}{\alpha_{0,g}(1+\beta_g\mu)} \;\;. \tag{B.1}$$

Where $\beta_g = \alpha_{1,g}/\alpha_{0,g}$. The Eqs. (12), (13) and (14) can be rewritten:

$$\mathrm{E}_g = \frac{8\pi k_B(\nu_{g+1}^3 - \nu_g^3)}{3c^3\alpha_{0,g}\beta_g}\frac{\operatorname{arctanh}(\beta_g)}{\beta_g} \;\;; \tag{B.2}$$

$$\|\mathbf{F}_g\| = -\frac{8\pi k_B(\nu_{g+1}^3 - \nu_g^3)}{3c^3\alpha_{0,g}}\frac{\operatorname{arctanh}(\beta_g) - \beta_g}{\beta_g^2} \;\;; \tag{B.3}$$

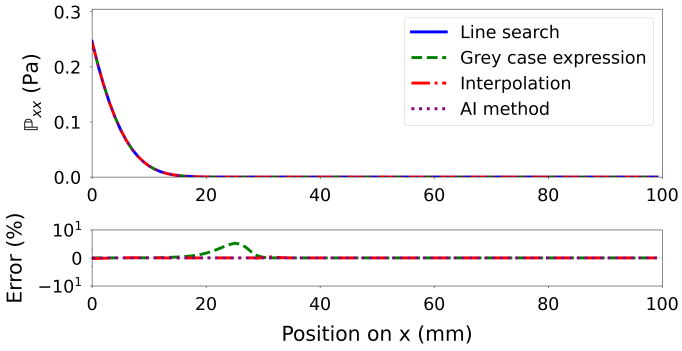
$$\mathcal{P}_g = \frac{8\pi k_B(\nu_{g+1}^3 - \nu_g^3)}{3c^3\alpha_{0,g}}\frac{\operatorname{arctanh}(\beta_g) - \beta_g}{\beta_g^3} \;\;. \tag{B.4}$$

One can verify using these expressions that these equations imply that the radiative temperature is such that $\mathrm{T}_g >> h\nu_{g+1}/k_B$, *i.e.* $\mathcal{T}_g^{\nu_{g+1}} >> 1$. The reduced flux and Eddington factor can then be expressed:

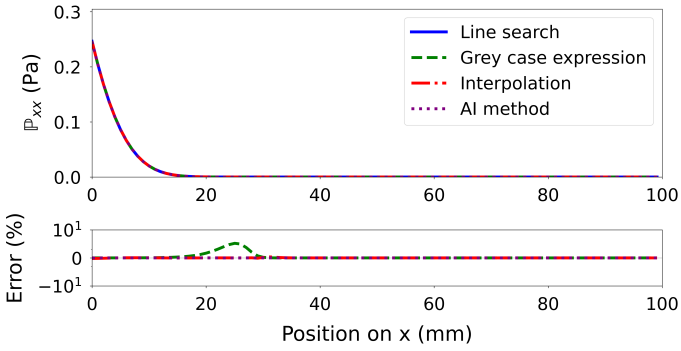
$$\mathrm{f}_g = -\frac{\operatorname{arctanh}(\beta_g) - \beta_g}{\beta_g \operatorname{arctanh}(\beta_g)} \;\;; \tag{B.5}$$

$$\chi_g = \frac{\operatorname{arctanh}(\beta_g) - \beta_g}{\beta_g^2 \operatorname{arctanh}(\beta_g)} \;\;. \tag{B.6}$$

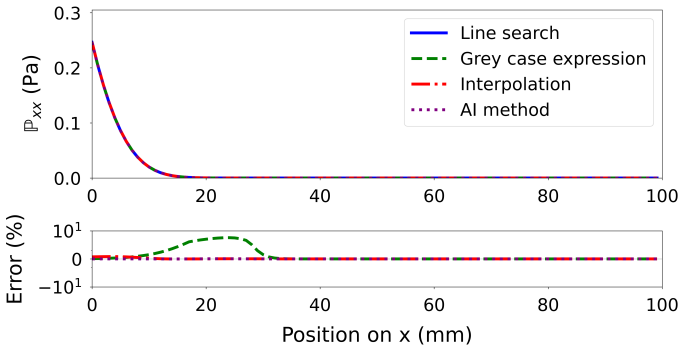
[5] https://julianlsolvers.github.io/LsqFit.jl/latest/



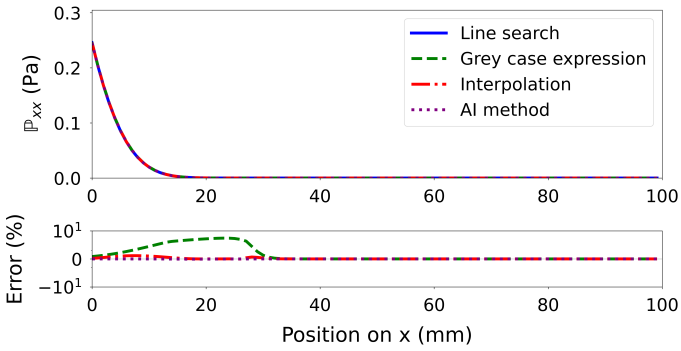
(a) Radiative pressure at time t=0.133 ns, using 2 groups



(b) Radiative pressure at time t=0.133 ns, using 3 groups



(c) Radiative pressure at time t=0.133 ns, using 4 groups



(d) Radiative pressure at time t=0.133 ns, using 5 groups

Figure 4: Component xx of the radiative pressure at time t=0.133 ns. The relative error is given by the formula: ($\mathbb{P}_{xx,method}$ - $\mathbb{P}_{xx,line\ search}$)/$\mathbb{P}_{xx,line\ search}$ × 100.

Thus, the Eddington factor only depends on the reduced flux. From these expressions, one can show that the Eddington factor is 1/3 at $f_g = 0$, 1 at $f_g = 1$ and that the slope of the Eddington factor is zero at $f_g = 0$, which is consistent with what we saw in Sec. 2.2.1. Additionally, at $f_g = 1$ the slope of the Eddington factor is worth $\partial_{f_g}\chi_g(1) = 1$.

Since these expressions do not provide an expression of χ_g as a function of f_g, We fitted a polynomial that incorporates the properties mentioned above. Using a dichotomy algorithm, we calculated 100 values of χ_g for a reduced flux varying between 0 and 1. Then, we fitted a polynomial P_H of degree 12 using Julia's library LsqFit[5]. The resulting polynomial expression is:

$$\begin{aligned} P_H(f_g) =& 1/3 + f_g^2 - f_g^3/3 \; + \\ & 6.5813886320063\,(f_g^4 - 2f_g^3 + f_g^2) \; - \\ & 44.5593808930324\,(f_g^5 - 3f_g^3 + 2f_g^2) \; + \\ & 179.9251066153469\,(f_g^6 - 4f_g^3 + 3f_g^2) \; - \\ & 463.2145547920471\,(f_g^7 - 5f_g^3 + 4f_g^2) \; + \\ & 776.0741150675088\,(f_g^8 - 6f_g^3 + 5f_g^2) \; - \\ & 841.0048410765069\,(f_g^9 - 7f_g^3 + 6f_g^2) \; + \\ & 566.6515584347650\,(f_g^{10} - 8f_g^3 + 7f_g^2) \; - \\ & 215.1853394583736\,(f_g^{11} - 9f_g^3 + 8f_g^2) \; + \\ & 35.1130440991836\,(f_g^{12} - 10f_g^3 + 9f_g^2) \; . \end{aligned} \tag{B.7}$$

The maximum relative error of this expression is worth 10^{-5}% on the Eddington factor and 10^{-3}% on the radiative pressure components.

Appendix C. Expression of the shifted radiative temperature (Case $\delta_g \to 1$)

For values of the bound frequencies that are very close ($\delta_g \geq 0.9$), we observed that the topology of the Eddington factor remains unchanged but shifts towards lower radiative temperatures (Fig. C.5). To determine the extent of this temperature shift, we examined how the evolution of the position of the most significant gradient, $\mathcal{T}_g^{\nabla,max}$ according to δ_g. We see that $\log_{10}(\mathcal{T}_g^{\nabla,max})$ varies linearly with $X = \log_{10}(1-\delta_g)$ as $\delta_g \to 1$ (Fig. C.6). Therefore, we fitted a linear function for $\delta_g > 0.99$ (*i.e.* $X < -2$) and obtained the following expression:

$$\log_{10}(\mathcal{T}_g^{\nabla,max}) = 0.250114\,X - 0.193613 \; . \tag{C.1}$$

For $\delta_g \in [0.9, 0.99]$ (*i.e.* $X \in [-2, -1]$), we fitted a second-degree polynomial as the relationship starts to deviate from linearity. We have obtained the following expression:

$$\log_{10}(\mathcal{T}_g^{\nabla,max}) = 0.007385\,X^2 + 0.277981\,X - 0.166850 \; . \tag{C.2}$$

We introduce a *shifted radiative temperature*, using the radiative temperature at $\delta_g = 0.99$ as a reference (*i.e.*, $\mathcal{T}_g^s$ equals $\mathcal{T}_g$ when $\delta_g = 0.99$). For $\delta_g > 0.99$, it is defined as:

$$\log_{10}(\mathcal{T}_g^s) = \log_{10}(\mathcal{T}_g) - 0.250114\,(X+2) \; . \tag{C.3}$$

For $\delta_g \in [0.9, 0.99]$, it is defined as:

$$\log_{10}(\mathcal{T}_g^s) = \log_{10}(\mathcal{T}_g) - 0.007385\,(X^2-4) - 0.277981\,(X+2) \; . \tag{C.4}$$

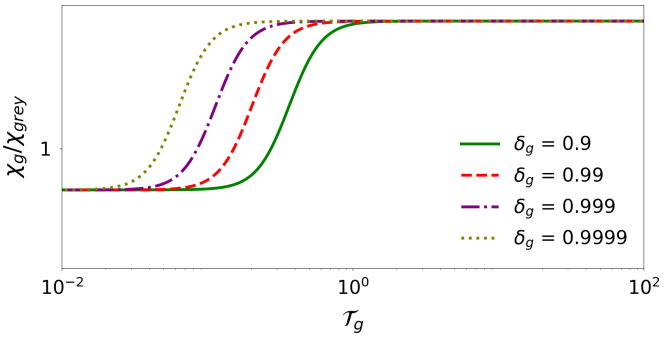


Figure C.5: Shift of the curve as δ_g gets closer to 1.

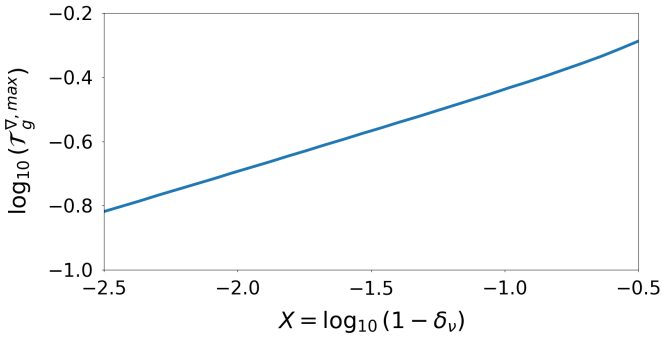


Figure C.6: Relation of the radiative temperature at the position of the most significant gradient as a function of $X = \log_{10}(1-\delta_g)$.

Appendix D. Input/Outputs of neural networks

As inputs for the neural networks, normalised values are used to enhance learning efficiency:

- Inputs for MLP$_a$:

$$x_1 = \frac{\log_{10}(\mathcal{T}_g^{\nu_g}) - \log_{10}(\mathcal{T}_{g,min}^{\nu_g})}{\log_{10}(\mathcal{T}_{g,max}^{\nu_g}) - \log_{10}(\mathcal{T}_{g,min}^{\nu_g})} \quad ;$$
$$x_2 = \mathrm{f}_g \quad .$$

- Inputs for MLP$_c$:

$$x_1 = \frac{\log_{10}(\mathcal{T}_g^{\nu_{g+1}}) - \log_{10}(\mathcal{T}_{g,min}^{\nu_{g+1}})}{\log_{10}(\mathcal{T}_{g,max}^{\nu_{g+1}}) - \log_{10}(\mathcal{T}_{g,min}^{\nu_{g+1}})} \quad ;$$
$$x_2 = \mathrm{f}_g \quad .$$

- Inputs for MLP$_{e,1}$ and MLP$_{e,2}$:

$$x_1 = \frac{\log_{10}(\mathcal{T}_g^{\nu_{g+1}}) - \log_{10}(\mathcal{T}_{g,min}^{\nu_{g+1}})}{\log_{10}(\mathcal{T}_{g,max}^{\nu_{g+1}}) - \log_{10}(\mathcal{T}_{g,min}^{\nu_{g+1}})} \quad ;$$
$$x_2 = \mathrm{f}_g \quad ;$$
$$x_3 = \frac{\delta_g - \delta_{g,min}}{\delta_{g,max} - \delta_{g,min}} \quad .$$

- Inputs for MLP$_f$:

$$x_1 = \frac{\log_{10}(\mathcal{T}_g^{s}) - \log_{10}(\mathcal{T}_{g,min}^{s})}{\log_{10}(\mathcal{T}_{g,max}^{s}) - \log_{10}(\mathcal{T}_{g,min}^{s})} \quad ;$$
$$x_2 = \mathrm{f}_g \quad .$$

Where the X_{min} and X_{max} quantities represent the minimum and maximum values within the considered domain, and x_i is an input of the neural networks.

Additionally, the MLPs do not directly predict the Eddington factor χ_g. Instead, they predict a related quantity, described by the expression:

$$\chi_g = 1/3 + 2/3\ \mathrm{f}_g^2\ \Big(1 + (1-\mathrm{f}_g)(3/2\ y - 1)\Big) \quad . \tag{D.1}$$

Where y is the output of the MLP, which has been empirically verified to fall within the range [0, 1]. This technique ensures that the predicted Eddington factor is 1/3 at $\mathrm{f}_g = 0$, 1 at $\mathrm{f}_g = 1$, and that the slope is zero at $\mathrm{f}_g = 0$.

Appendix E. Selection of the architecture of the neural network

Appendix E.1. Training strategy

To train the neural networks, we utilise Julia's library Flux[6]. We employ the Light Broyden-Fletcher-Goldfarb-Shanno (LBFGS) optimiser from the Julia library Optim[7], chosen for its rapid convergence and efficient minimisation of the cost function. Although LBFGS lacks stochastic properties, potentially affecting generalisation, our comparison with the vanilla Adam optimiser [14] shows that neural networks trained with LBFGS remain generalisable in our specific scenarios. For the training, we have defined two distinct cost functions:

$$\mathcal{L}_1(x_1, x_2, \chi_{g,data}, \chi_{g,pred}) = \left(\frac{\chi_{g,pred} - \chi_{g,data}}{\alpha_f\ \alpha_T}\right)^2 \quad ; \tag{E.1}$$

$$\mathcal{L}_2(x_1, x_2, x_3, \chi_{g,data}, \chi_{g,pred}) = \left(\frac{\chi_{g,pred} - \chi_{g,data}}{\alpha_f\ \alpha_T\ \alpha_\nu}\right)^2 \quad . \tag{E.2}$$

Where $\chi_{g,data}$ and $\chi_{g,pred}$ are, respectively, the label and predicted values of the Eddington factor, x_1, x_2, x_3 are the inputs of the neural network (see Appendix D), and α_f, α_T, α_ν are factors defined by the expressions:

$$\alpha_T = \min(x_1, 1-x_1) + \epsilon_T \quad ; \tag{E.3}$$

$$\alpha_f = \min(1-x_2, \chi_{g,data} - 1/3) + \epsilon_f \quad ; \tag{E.4}$$

$$\alpha_\nu = \min(x_3, 1-x_3) + \epsilon_\nu \quad . \tag{E.5}$$

Where ϵ_T, ϵ_f and ϵ_ν are constants that prevent these factors to be worth 0. We have respectively set these constants to 0.1, 10^{-3} and 0.1. These factors serve two purposes:

[6] https://fluxml.ai/Flux.jl/stable/

[7] https://julianlsolvers.github.io/Optim.jl/v0.9.3/algo/lbfgs/

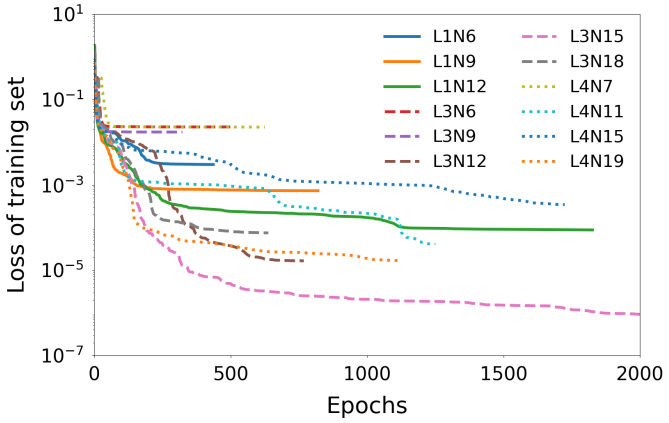


Figure E.7: Learning curves of the different architectures tested. L represents the number of hidden layers, and N is the total number of perceptrons. For example, architecture L1N6 has 1 hidden layer and 6 perceptrons.

- α_T and α_γ reduce the discontinuities at the interfaces of the domains defined in Sec. 3.2,
- α_f emphasises the importance of domains in which an error on the Eddington factor has a significant impact on the radiative pressure (see Appendix F).

We use $\mathcal{L}_1$ to train the neural networks MLP_a, MLP_c and MLP_f and $\mathcal{L}_2$ for $\mathrm{MLP}_{e,1}$ and $\mathrm{MLP}_{e,2}$.

The models are trained until the weights and biases of the MLP exhibit variations below 10^{-32}; the predicted values show variations below 10^{-32}, or the gradient falls below 10^{-8}.

Appendix E.2. Selection of the architecture

For all the neural networks, we use different activation functions:

- *input layer*: linear function,
- *hidden layers*: hyperbolic tangent function tanh,
- *output layer*: sigmoid function σ.

Let us consider the neural network $\mathrm{MLP_a}$ as an example. We trained various neural networks with different numbers of hidden layers and perceptrons that are simple enough to ensure fast prediction times. We have used all the training set data specified in Sec. 3.3 to train the neural networks. The architecture we have selected is the one whose cost function is the lowest at the end of the training, which is the architecture having 3 hidden layers and 15 neurons in the case of $\mathrm{MLP_a}$ (Fig. E.7). We have employed the same methodology for all the other neural networks.

Appendix F. Relative error of the radiative pressure

Using Eq. (5), one can express each component of the radiative pressure tensor as:

$$\mathbb{P}_{g,xx} = \left\{ \frac{1-\chi_g}{2} + \frac{3\chi_g - 1}{2} \frac{\mathrm{F}_{g,x}^2}{\|\mathbf{F}_g\|^2} \right\} \mathrm{E}_g \quad ; \tag{F.1}$$

$$\mathbb{P}_{g,yy} = \left\{ \frac{1-\chi_g}{2} + \frac{3\chi_g - 1}{2} \frac{\mathrm{F}_{g,y}^2}{\|\mathbf{F}_g\|^2} \right\} \mathrm{E}_g \quad ; \tag{F.2}$$

$$\mathbb{P}_{g,xy} = \frac{3\chi_g - 1}{2} \frac{\mathrm{F}_{g,x}\mathrm{F}_{g,y}}{\|\mathbf{F}_g\|^2} \mathrm{E}_g \quad . \tag{F.3}$$

Considering only an error $\delta\chi_g$ in the determination of the Eddington factor, The relative error on the radiative pressure components can be expressed:

$$\left| \frac{\delta\mathbb{P}_{g,xx}}{\mathbb{P}_{g,xx}} \right| = \left| \frac{3x^2 - 1}{(3x^2-1)\chi_g + 1 - x^2} \right| \delta\chi_g \quad ; \tag{F.4}$$

$$\left| \frac{\delta\mathbb{P}_{g,yy}}{\mathbb{P}_{g,yy}} \right| = \left| \frac{3y^2 - 1}{(3y^2-1)\chi_g + 1 - y^2} \right| \delta\chi_g \quad ; \tag{F.5}$$

$$\left| \frac{\delta\mathbb{P}_{g,xy}}{\mathbb{P}_{g,xy}} \right| = \frac{\delta\chi_g}{\chi_g - 1/3} \quad . \tag{F.6}$$

Where $x = \mathrm{F}_{g,x}/\|\mathbf{F}_g\|$ and $y = \mathrm{F}_{g,y}/\|\mathbf{F}_g\|$. $|\delta\mathbb{P}_{g,xx}/\mathbb{P}_{g,xx}|$ and $|\delta\mathbb{P}_{g,yy}/\mathbb{P}_{g,yy}|$ admit a maximum relative error when respectively $x = 0$ or 1 and $y = 0$ or 1. In this case, we can express the maximum relative errors as:

$$\left| \frac{\delta\mathbb{P}_{g,xx}}{\mathbb{P}_{g,xx}} \right|_{\max} = \frac{\delta\chi_g}{\min\left\{\chi_g, 1-\chi_g\right\}} \quad ; \tag{F.7}$$

$$\left| \frac{\delta\mathbb{P}_{g,yy}}{\mathbb{P}_{g,yy}} \right|_{\max} = \frac{\delta\chi_g}{\min\left\{\chi_g, 1-\chi_g\right\}} \quad ; \tag{F.8}$$

$$\left| \frac{\delta\mathbb{P}_{g,xy}}{\mathbb{P}_{g,xy}} \right|_{\max} = \frac{\delta\chi_g}{\chi_g - 1/3} \quad . \tag{F.9}$$

These expressions reveal the importance of accurate predictions of the Eddington factor at values of the reduced flux f_g close to 0 or 1 to reduce the relative errors on the components of the radiative pressure tensor.

Appendix G. A short description of the 2D line search algorithm developed

According to Turpault 2002 [5], it is established that a physically realistic pair (E_g, f_g) corresponds uniquely to a set of Lagrange multipliers ($\alpha_{0,g}$, $\alpha_{1,g}$). Therefore, using a search algorithm to derive these Lagrange multipliers from the radiative quantities is feasible. Instead of directly employing the pairs ($\alpha_{0,g}$, $\alpha_{1,g}$) and (E_g, f_g), we've opted for inputs ($x_1 = \log(\alpha_{0,g})$, $x_2 = 100\sigma^{-1}(\beta_g)$), where σ^{-1} denotes the inverse of the sigmoid function, and outputs ($y_1 = \log(\mathrm{E}_g)$, $y_2 = \log(\mathrm{f}_g)$). This choice facilitates the utilisation of a line search algorithm and ensures quicker convergence. The computation of $\log(\mathrm{E}_g)$ and $\log(\mathrm{f}_g)$ follows the methodology outlined in Chinh Hyunh Nguyen's

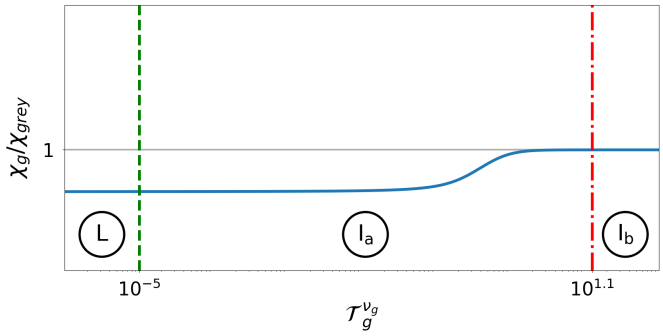


Figure H.8: Ratio χ_g/χ_{grey} with $f_g = 0.65$ $f_g = 0.65$ and $\nu_g = 0$. I_b: Domain in which χ_{grey} is valid; I_c: High-intermediate radiative, $\chi_g = \mathcal{F}(f_g, \mathcal{T}_g^{\nu_{g+1}})$; H: Asymptotically high radiative temperatures, $\chi_g = \mathcal{F}(f_g)$.

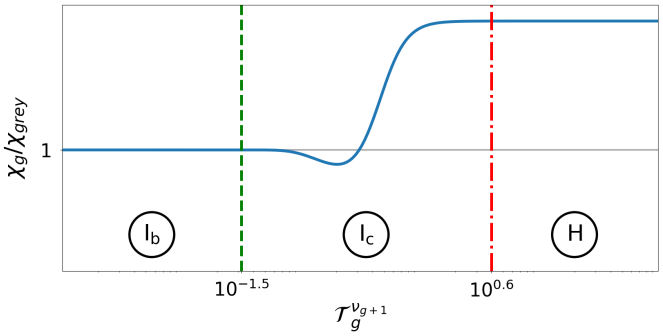


Figure H.9: Ratio χ_g/χ_{grey} with $f_g = 0.65$ and $\nu_{g+1} \to \infty$. L: Asymptotically low radiative temperatures, $\chi_g = \mathcal{F}(f_g)$; I_a: Low-intermediate radiative temperatures, $\chi_g = \mathcal{F}(f_g, \mathcal{T}_g^{\nu_g})$; I_b: Domain in which χ_{grey} is valid.

thesis [1]. As for the line search algorithm, we've implemented a Fortran version based on the code from Numerical Recipes for Fortran 90 [15]. The Lagrange multipliers are searched until the error on the radiative energy and the reduced flux is calculated with the Eqs. (12) and (19) are less than 10^{-6} %.

Appendix H. Case where $\nu_g = 0$ or $\nu_{g+1} \to \infty$

Appendix H.1. Case $\nu_g = 0$

In this scenario, the only well-defined adimensionalised radiative temperature is $\mathcal{T}_g^{\nu_{g+1}}$. Therefore, the Eddington factor is such that $\chi_g = \mathcal{F}(f_g, \mathcal{T}_g^{\nu_{g+1}})$. Fig. H.8 illustrates the Eddington factor's behaviour as a function of $\mathcal{T}_g^{\nu_{g+1}}$. In this case, we retrieve domains I_b, I_c, and H as discussed in the limit where $\delta_g \to 0$ (Sec. 2.2.2).

Appendix H.2. Case $\nu_{g+1} \to \infty$

Here, the only well-defined adimensionalised radiative temperature is $\mathcal{T}_g^{\nu_g}$. Consequently; the Eddington factor is such that $\chi_g = \mathcal{F}(f_g, \mathcal{T}_g^{\nu_g})$. Figure H.9 illustrates the Eddington factor's behaviour as a function of $\mathcal{T}_g^{\nu_g}$. In this case, we retrieve domains L, I_a, and I_b as discussed in the limit where $\delta_g \to 0$ (Sec. 2.2.2).

References

[1] H. C. Nguyen, Simulation de modèles hydrodynamiques et de transfert radiatif intervenant dans la description d'écoulements astrophysiques, Ph.D. thesis, Paris 11 (2011).

[2] C. Michaut, H. Nguyen, L. Di Menza, Computational radiation hydrodynamics, Astrophysics and Space Science 336 (1) (2011) 175–181.

[3] C. Michaut, L. Di Menza, H. Nguyen, S. Bouquet, M. Mancini, Hades code for numerical simulations of high-mach number astrophysical radiative flows, High Energy Density Physics 22 (2017) 77–89.

[4] B. Dubroca, J.-L. Feugeas, Etude théorique et numérique d'une hiérarchie de modèles aux moments pour le transfert radiatif, Comptes Rendus de l'Académie des Sciences - Series I - Mathematics 329 (10) (1999) 915–920.

[5] R. Turpault, Construction d'un modèle m1-multigroupe pour les équations du transfert radiatif, Comptes Rendus Mathematique 334 (4) (2002) 331–336.

[6] R. Turpault, A consistent multigroup model for radiative transfer and its underlying mean opacities, Journal of Quantitative Spectroscopy and Radiative Transfer 94 (3) (2005) 357–371.

[7] B. Freeman, L. Hauser, J. Palmer, S. Pickard, G. Simmons, D. Williston, J. Zerkle, Dasa report no. 2135, Vol. I (La Jolla: Systems, Science, and Software, Inc.) (1968).

[8] G. Spillman, Formulation of the eddington factor for use in one-dimensional nonequilibrium diffusion calculations, Los Alamos National Laboratory Office Memorandum, Los Alamos National Laboratory, Los Alamos, NM 87545 (1968) 25.

[9] D. Mihalas, B. Weibel-Mihalas, Foundations of radiation hydrodynamics, Oxford University Press, 1984.

[10] C. Levermore, Relating eddington factors to flux limiters, Journal of Quantitative Spectroscopy and Radiative Transfer 31 (2) (1984) 149–160.

[11] R. Turpault, Modelisation, approximation numerique et applications du transfert radiatif en desequilibre spectral couple avec l'hydrodynamique, Theses, Université Sciences et Technologies - Bordeaux I (Dec. 2003).

[12] N. Vaytet, E. Audit, B. Dubroca, F. Delahaye, A numerical model for multigroup radiation hydrodynamics, Journal of Quantitative Spectroscopy and Radiative Transfer 112 (8) (2011) 1323–1335.

[13] A. Harada, S. Nishikawa, S. Yamada, Deep learning of the eddington tensor in core-collapse supernova simulation, The Astrophysical Journal 925 (2) (2022) 117.

[14] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980 (2014).

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical recipes in Fortran 90 (2nd ed.): the art of parallel scientific computing, Cambridge University Press, USA, 1996.