

# RALEF-2D: A 2D hydrodynamic code with heat conduction and radiation transport.

M.M. Basko

*GSI, Darmstadt; on leave from KIAM, Moscow\**

J. Maruhn and Anna Tauschwitz

*Frankfurt University, Frankfurt*

(Dated: December 5, 2017)

## Contents

<b>1. Introduction</b>	4
<b>2. Basic equations</b>	5
1. Hydrodynamics and radiation transfer	5
2. The diffusion limit of radiation transport	7
3. Units of measurement and dimensional physical constants	8
1. User-defined units in the RALEF code	8
2. The DEIRA units.	10
3. Ideal-gas equation of state	11
4. The inverse bremsstrahlung absorption coefficient	11
4. Thermal conduction	13
1. Spitzer thermal conductivity	13
2. A semi-empirical model for thermal conductivity of mono-atomic metals and plasmas in the mean ion approximation	14
<b>3. Discretization of the fluid energy equation</b>	17
1. Time discretization with the SSI method	17
2. Time step control	19
3. Discretization of radiation transport: general approach	21
<b>4. Radiation transport</b>	23
1. Discretization of frequencies	23
1. General formulae	23
2. Frequency sets	25
2. Discretization of angular variables with the $S_n$ method	25
1. Discrete $S_n$ ordinates in the Cartesian $(x, y)$ and axial $(R, Z)$ 2D coordinate systems	25
2. Angular integrals and particular sets of the $S_n$ -nodes	30
3. Extended numbering system for the $S_n$ nodes	33
4. Sequential order of processing the $S_n$ beamlets	36
3. Computation of the radiation field	38

---

\*Electronic address: [basko@itep.ru](mailto:basko@itep.ru)

1. The octagon stencil for integration along a short characteristic	38
2. The principal finite-difference integration formula	40
3. Initial values for characteristic integration	42
4. Positive piece-wise parabolic interpolation of $B$ along the short characteristic and calculation of the $\tau$ -derivative $B'$	46
5. Piece-wise parabolic interpolation of the source function $B$ along the cell edges	48
6. Vertex emissivities	50
7. Temperature derivatives	52
4. Radiant energy deposition	54
1. Volumetric versus flux-based algorithms	54
2. Flux-based formulae	55
3. Incompatibility of the flux-based formulae with nodal collocation for $I$	57
4. Column-flux algorithm	58
5. Column heating rate	60
6. The diffusion limit	62
7. Reduction to the 2D case	63
5. Boundary conditions for radiation transfer	66
1. Allowed types of global reflective boundaries	66
2. Processing order for angular directions in the Cartesian case $\text{iradial} = 0$	71
3. Physical boundary conditions	72
4. Boundary conditions for radiation intensity	73
5. Boundary conditions for the Planckian source function $B_\nu(T)$	73
6. Details of the numerical algorithm for radiation transport	76
1. Allowed mesh deformations	76
2. Mesh notation: physically associated vertices (PAVs) and their $j$ -ordering	78
3. Causal dependencies among PAVs for a given beamlet	81
4. R-donor data representation	84
5. Principal cases of rd-linkage	84
6. Algorithm for calculation of the r-donor cells and vertices	84
7. Solution of the ray-octagon problem	84
8. Remote r-donor vertices across a void region	86
9. Local r-donor vertices across an r-donor cell: algorithm I	86
10. Local r-donor vertices across an r-donor cell: algorithm II	88
11. General scheme of the algorithm	89
12. Sorting PAVs to solve the Cauchy problem	91
<b>5. Laser energy deposition</b>	<b>93</b>
1. Classification of implemented models	93
2. Different options for laser beam geometry	94
3. Non-refractive XS model #1	100
1. Transfer equation for the laser light	100
2. Treatment of reflective boundaries	102
3. Description of the LASDEPR subroutine	106
4. The hybrid refractive/reflective model of laser light transport	107
1. General equations for monochromatic electromagnetic fields in dispersive media	107
2. Propagation of a monochromatic planar wave across a non-uniform planar layer	110
5. Test problems for laser deposition: exact solutions	116

1. Absorption on a step profile (metallic surface) between two half-spaces	116
2. S-wave transmission through a 3-layer slab	118
6. Test problems for laser deposition: the RALEF results	124
1. Test 1: reflection/absorption on a step profile	124
2. Test 2: reflection/absorption on a linearly ramped profile	126
7. Laser absorption coefficient	136
1. Complex dielectric permittivity within the Drude model	136
2. The inverse-bremsstrahlung collisional absorption	140
<b>6. EOS and opacities</b>	140
1. “F17” file format for data exchange	140
2. Tabular GLT equation of state	143
1. Units of measurement	144
2. Principal arrays with the EOS data	146
3. Density and temperature grids	146
4. Structure of the principal arrays <code>petab17(:)</code> and <code>pexstab17(:)</code>	147
5. Interpolation scheme and the GLT-EOS in the caloric form $p = p(\rho, e)$ , $T = T(\rho, e)$	149
6. The Maxwell construction for the equilibrium (EQ) version of the GLT-EOS	151
7. Extrapolation beyond the main “warm” table <code>petab17(:)</code>	153
3. Tabular GLT conduction coefficient and opacities	155
1. Interpolation method	156
2. Extrapolation beyond table limits for thermal conduction	156
3. Extrapolation beyond table limits for radiation absorption coefficients	157
4. Ionization degree at low energy densities	160
<b>7. Mesh rezoning and remapping of principal variables</b>	162
1. General scheme	162
2. General features of the rezoning algorithm	162
3. Defining purely Lagrangian parts of the mesh	164
4. Algorithm for smoothing corrugated interfaces	164
5. Algorithm for pocket filling where an interface tends to fold up	167
6. Second-order remapping	170
1. Gradient limiting	171
2. Limits on the 2nd-order advection terms	173
<b>8. Parallelization scheme</b>	178
<b>9. Numerical diagnostics</b>	178
1. Angular distribution of emission in selected spectral band	178
<b>10. Test problems</b>	180
1. Searchlight beam in vacuum ( $xy$ geometry)	181
2. Thermal emission from a slab ( $xy$ geometry)	184
1. Isothermal slab	186
2. Slab with a sine $B$ -profile: optically thin mesh	189
3. Slab with a sine $B$ -profile: optically thick mesh and the diffusion limit	191
3. Radiation transport in a cylindrical cavity ( $xy$ geometry)	193

1. Total flux balance	196
2. The ray effect	197
4. Thermal emission from a sphere ( $rz$ geometry)	200
1. Analytical solution	200
2. Radiative cooling of an isothermal sphere	202
3. Sphere with a cosine $B$ -profile: optically thin mesh	205
4. Sphere with a cosine $B$ -profile: optically thick mesh and the diffusion limit	207
5. Thermal emission from a narrow opaque core ( $xy$ and $rz$ geometries)	207
1. Analytical solution for a sphere in the $rz$ -geometry	209
2. Analytical solution for a cylinder in the $xy$ -geometry	210
3. Numerical results	211
<b>A. The <math>ES_n</math> angular ordinates</b>	<b>216</b>
<b>B. Intersection of a ray with a conical segment in the axisymmetric <math>rz</math>-geometry</b>	<b>218</b>
1. Basic formulae	218
2. Classification of solutions and selection rules	220
3. Rounding errors and safety margins	224
<b>C. Second-order interpolation schemes</b>	<b>227</b>
1. Positive piece-wise parabolic oscillation-suppressed interpolation on a 3-point stencil (PPOS3P)	227
2. Symmetric 4-point piece-wise parabolic interpolation	230
3. Positive piece-wise parabolic interpolation	232
4. Monotonic piece-wise parabolic interpolation	232
5. Symmetric monotonic piece-wise parabolic interpolation	234
<b>References</b>	<b>238</b>

## 1. INTRODUCTION

The two-dimensional numerical code RALEF, described in this report, has been developed with the intention to adequately simulate plasma flows where non-local spectral energy transport by thermal radiation plays a significant role: hence the acronym **RALEF-2D** from **R**adiative **A**rbitrary **L**agrange-**E**ulerian **F**luid dynamics in two dimensions. The RALEF code has been built around the CAVEAT package [1] for solution of the equations of ideal hydrodynamics in two dimensions. Hence, it inherited from CAVEAT the topology, the structure and all the notations for the computational mesh, as well as the 2nd-order Godunov-type scheme for solving the hydrodynamic equations and the arbitrary Lagrangian-Eulerian (ALE) technique for mesh adaptation. Concerning all these hard-core features of the RALEF code, the reader is referred to the above cited report [1]. Here it should probably be only reminded that we use a structured quadrilateral mesh, which generally consists of a number of touching blocks, each of which has a topology of a rectangle with two logical mesh directions 1 and 2 ordered in a right-handed manner. All the principal dynamic variables in the hydrodynamics package — like density, velocity, specific internal energy, etc. — are assigned to the geometrical centers of the quadrilateral mesh cells.

As a first step to RALEF — thermal conduction [2, 3]. It was important as a preparation for radiation transport because of the SSI.

## 2. BASIC EQUATIONS

### 1. Hydrodynamics and radiation transfer

The RALEF-2D code solves the following system of fluid dynamics equations

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \vec{u}) = 0, \quad (2.1)$$

$$\frac{\partial}{\partial t}(\rho \vec{u}) + \text{div}(\rho \vec{u} \otimes \vec{u}) + \nabla p = 0, \quad (2.2)$$

$$\frac{\partial(\rho E)}{\partial t} + \text{div}[(\rho E + p)\vec{u}] = Q_T + Q_r + Q_{dep}, \quad (2.3)$$

in two spatial dimensions. Equations (2.1)–(2.3) are written in a general three-dimensional (3D) form, where all dependent variables are functions of time  $t$  and position  $\vec{x}$  (a 3D Cartesian vector). More specifically,  $\rho = \rho(t, \vec{x})$  is the fluid mass density,  $\vec{u} = \vec{u}(t, \vec{x})$  is the fluid velocity,  $p = p(\rho, e)$  is the pressure,  $e = e(t, \vec{x})$  is the mass specific internal energy,  $E = e + u^2/2$  is the mass-specific total (internal + kinetic) energy,  $Q_{dep} = Q_{dep}(t, \vec{x})$  is the volume-specific heating power by external (other than from thermal conduction and coupling to the radiation field) energy sources. Symbol  $\otimes$  denotes the tensor product of two vectors. Reduction to the pertinent 2D coordinate systems (either  $xy$  Cartesian or  $rz$  axi-symmetric), used in the RALEF code, is described in Refs [1] and [2].

The energy equation (2.3) includes the source term due to thermal conduction,

$$Q_T = -\text{div}(\vec{h}_T), \quad \vec{h}_T = -\kappa \nabla T, \quad (2.4)$$

and the source term due to energy transport by thermal radiation,

$$Q_r = -\text{div}(\vec{h}_r), \quad \vec{h}_r = \int_0^\infty d\nu \int_{4\pi} I_\nu \vec{\Omega} d\vec{\Omega}, \quad (2.5)$$

expressed in terms of the corresponding energy flux densities  $\vec{h}_T$  and  $\vec{h}_r$  [erg cm<sup>-2</sup> s<sup>-1</sup>]. In Eq. (2.4)  $T = T(\rho, e)$  is the matter temperature, and  $\kappa = \kappa(\rho, T)$  is the heat conduction coefficient.

The radiation energy flux  $\vec{h}_r$  is obtained by integrating the *spectral intensity* (or, simply, *intensity*) of radiation  $I_\nu = I_\nu(t, \vec{x}, \vec{\Omega}, \nu)$  over the full solid angle  $\int d\vec{\Omega} = 4\pi$ , and all the photon frequencies  $\nu > 0$ . We assume that the spectral intensity  $I_\nu$  obeys the simplest version of the transfer equation in the quasi-static approximation [4]

$$\vec{\Omega} \cdot \nabla I_\nu = k_\nu (B_\nu - I_\nu), \quad (2.6)$$

where the time-derivative term  $c^{-1}\partial I_\nu/\partial t$  is neglected. Here  $\vec{\Omega}$  is the propagation direction of a photon with an energy  $h\nu$ ,  $k_\nu = k_\nu(\nu, \rho, T)$  [cm<sup>-1</sup>] is the spectral absorption coefficient corrected for induced emission,  $c$  is the speed of light, and  $B_\nu = B_\nu(\nu, \rho, T)$  is the radiation

source function, which is supposed to be a known function of its arguments when Eq. (2.6) is solved. Possible role of scattering is ignored. The quasi-static treatment of radiation transfer is justified so long as the characteristic timescale of variation of the physical fluid variables greatly exceeds  $L/c$ , where  $L$  is the characteristic size of the simulated region.

Thus, the primary dependent variable, which describes the radiation field in our case, is the spectral intensity  $I_\nu$ . In addition, we will also use the  $4\pi$ -integrated intensity

$$U_\nu = U_\nu(t, \vec{x}, \nu) = \int_{4\pi} I_\nu d\vec{\Omega}, \quad (2.7)$$

the *frequency-integrated intensity*

$$I = I(t, \vec{x}, \vec{\Omega}) = \int_0^\infty I_\nu d\nu, \quad (2.8)$$

and the *full integral intensity*

$$U_r = U_r(t, \vec{x}) = \int_0^\infty U_\nu d\nu = \int_{4\pi} I d\vec{\Omega} = \int_0^\infty d\nu \int_{4\pi} I_\nu d\vec{\Omega} \quad (2.9)$$

to characterize the local intensity of thermal radiation. Respectively, the full energy density of the thermal radiation is given by

$$\mathcal{E}_r = \frac{1}{c} U_r. \quad (2.10)$$

In this report we describe the LTE (local thermal equilibrium) version of the RALEF-2D code, where

$$B_\nu = B_\nu(\nu, T) = \frac{2h\nu^3}{c^2} \frac{1}{\exp(h\nu/T) - 1} \quad (2.11)$$

is the black-body *spectral Planckian intensity*. Here and below the “non-barred” Planckian constant  $h = 2\pi\hbar$  should not be mixed up with the energy fluxes (like  $\vec{h}_T, \vec{h}_r$ ), for which the same letter  $h$  is used. Since  $B_\nu$  does not depend on the photon propagation direction  $\vec{\Omega}$ , its integration over the full solid angle is reduced to multiplication by  $4\pi$ .

The relevant integral quantity for the equilibrium radiation field is the frequency-integrated Planckian intensity

$$B = B(T) = \int_0^\infty B_\nu d\nu = \frac{\sigma_{SB} T^4}{\pi}, \quad (2.12)$$

where

$$\sigma_{SB} = \frac{2\pi^5}{15h^3c^2} = \frac{\pi^2}{60\hbar^3c^2} \quad (2.13)$$

is the Stefan-Boltzmann constant. This constant is usually associated with the black-body energy flux emitted by a surface with temperature  $T$

$$h_{r,Pl} = \pi B = \sigma_{SB} T^4. \quad (2.14)$$

The energy density of the black-body radiation is given by

$$\mathcal{E}_{r,Pl} = \frac{4\pi B}{c} = a_{St}T^4, \quad (2.15)$$

where

$$a_{St} = \frac{4\sigma_{SB}}{c} = \frac{8\pi^5}{15h^3c^3} = \frac{\pi^2}{15\hbar^3c^3} \quad (2.16)$$

is the Stefan constant. Very often, even when the local radiation field is highly non-equilibrium, its energy density is characterized not by the value of  $\mathcal{E}_r$  (or  $U_r$ ) but by the equivalent radiation temperature  $T_r$ , defined as

$$T_r = \left( \frac{U_r}{4\sigma_{SB}} \right)^{1/4}. \quad (2.17)$$

The quasi-static form of the transfer equation (2.6) implies that at any given time  $t$  the energy content of the radiation field (hence, its heat capacity) is neglected in the overall energy inventory: the energy radiated by one fluid element is either instantaneously reabsorbed by other fluid elements or lost from the system. In this respect, from the point of view of fluid dynamics, this approximation to radiation transport is equivalent to a specific kind of non-local heat transport model. Note also that the transfer equation Eq. (2.6) is, strictly speaking, written for a motionless medium and does not contain fluid velocity terms; for this reason it becomes a poor approximation to the energy transport in narrow lines across fluid configurations with fast motions; for more details see Ref. [4].

Note that by virtue of Eq. (2.6), the radiative source term in the energy equation (2.3) can be cast in the form

$$Q_r = \int_0^\infty k_\nu \left( \int_{4\pi} I_\nu d\vec{\Omega} - 4\pi B_\nu \right) d\nu. \quad (2.18)$$

## 2. The diffusion limit of radiation transport

The diffusion limit of radiation transport is obtained by asymptotic expansion with respect to the small parameter  $k_\nu^{-1} \ll L$ , where  $L$  is the characteristic length scale of  $T$  and  $k_\nu$  variation. In this limit one can apply the method of successive approximations to the transfer equation (2.6), with the zero approximation given by  $I_\nu = B_\nu$ . Then, in the next (first) approximation Eq. (2.6) yields

$$I_\nu = B_\nu - \frac{1}{k_\nu} \vec{\Omega} \cdot \nabla B_\nu + O\left(\frac{1}{k_\nu^2}\right). \quad (2.19)$$

Integrating Eq. (2.19) over the frequencies, we obtain

$$\int_0^\infty I_\nu d\nu = B - \frac{1}{k_R} \vec{\Omega} \cdot \nabla B + O\left(\frac{1}{k_R^2}\right), \quad (2.20)$$

where  $k_R$  is the Rosseland mean absorption coefficient defined as

$$k_R^{-1} = \frac{\int_0^{\infty} k_\nu^{-1} (\partial B_\nu / \partial T) d\nu}{\int_0^{\infty} (\partial B_\nu / \partial T) d\nu}. \quad (2.21)$$

Multiplying Eq. (2.20) by the unit vector  $\vec{\Omega}$  and integrating over the directions, we obtain

$$\vec{h}_r = -\frac{4\pi}{3k_R} \nabla B + O\left(\frac{1}{k_R^2}\right), \quad (2.22)$$

which, upon substitution into Eq. (2.5), yields

$$Q_r = \text{div} \left( \frac{4\pi}{3k_R} \nabla B \right). \quad (2.23)$$

Having recalled the expression (2.15) for the energy density of the equilibrium black-body radiation, we can rewrite Eq. (2.23) in a more familiar form

$$Q_r = \text{div} \left( \frac{c}{3k_R} \nabla \mathcal{E}_{r,Pl} \right) = \text{div} \left( \frac{16\sigma_{SB}}{3k_R} T^3 \nabla T \right), \quad (2.24)$$

where  $c/3k_R = l_{RC}/3$  is the canonical expression for the radiation diffusion coefficient. Equation (2.24) tells us that the diffusion limit of the quasi-static transfer equation (2.6) is in fact equivalent to the approximation of *radiative heat conduction*, where the energy transport by thermal radiation is described by simply adding the term

$$\kappa_r = \frac{16\sigma_{SB}}{3k_R} T^3 \quad (2.25)$$

to the full coefficient of thermal heat conduction  $\kappa$ .

### 3. Units of measurement and dimensional physical constants

#### 1. User-defined units in the RALEF code

Equations of hydrodynamics (2.1)–(2.3) have no physical constants and preserve their form for any self-consistent system of units based on three fundamental units of measurement

$$\begin{aligned} \text{length unit} &= [l] \text{ centimeters,} \\ \text{time unit} &= [t] \text{ seconds,} \\ \text{mass unit} &= [m] \text{ grams,} \end{aligned} \quad (2.26)$$

which can be chosen arbitrarily. In Eq. (2.26) and everywhere below we assume that numerically the freely chosen fundamental units  $[m]$ ,  $[l]$  and  $[t]$  are always expressed in the CGS (centimeter-gram-second) units. The system of units is called self-consistent when the units

TABLE 2.1: Derived units of measurement in hydrodynamic equations.

quantity	notation	unit	dimensions in CGS
velocity	$\vec{u}$	$[l][t]^{-1}$	cm s <sup>-1</sup>
density	$\rho$	$[m][l]^{-3}$	g cm <sup>-3</sup>
pressure	$p$	$[m][l]^{-1}[t]^{-2}$	erg cm <sup>-3</sup>
mass-specific energy	$E, e$	$[l]^2[t]^{-2}$	erg g <sup>-1</sup>
energy flux	$\vec{h}, \vec{h}_r$	$[m][t]^{-3}$	erg cm <sup>-2</sup> s <sup>-1</sup>
volume-specific power	$Q_T, Q_r, Q_{dep}$	$[m][l]^{-1}[t]^{-3}$	erg cm <sup>-3</sup> s <sup>-1</sup>

for all the other physical quantities in Eqs. (2.1)–(2.3) are derived from the base units (2.26) as listed in Table 2.1.

Once everybody agrees to use only self-consistent systems of units, the units of measurement do not have to be specified in a pure hydrodynamic code. However, the situation changes after the heat conduction term and the radiation transfer equation are added to the basic equations. First of all, the temperature  $T$  emerges as one of the principal variables, and dimensional physical constants appear in expressions for the heat conduction coefficient  $\kappa = \kappa(\rho, T)$ , the Planckian intensity  $B_\nu = B_\nu(\nu, T)$ , and the radiation absorption coefficient  $k_\nu = k_\nu(\nu, \rho, T)$ . Evidently, the numerical values of dimensional constants will be different in different systems of user-defined units. The speed of light, for example, will be equal to

$$c \stackrel{\text{R}}{=} 2.99792458 \times 10^{10} \frac{[t]}{[l]}. \quad (2.27)$$

For practical (and/or historical) reasons, it is convenient to introduce the fourth independent base unit of measurement for temperature

$$\text{temperature unit} = [T] \text{ ergs}. \quad (2.28)$$

In the RALEF-2D code, the four base units (2.26) and (2.28) are treated as user-defined free parameters, and all dimensional formulae are cast in a form, which yields correct values for any choice of the four base units. With regard to the radiation frequency  $\nu$ , we assume that the photon energy  $h\nu$  is measured in the same energy units  $[T]$  as the temperature  $T$ ; in other words, if, for example, temperature is measured in eV (i.e.  $[T] = 1.60217733 \times 10^{-12}$ ), then the radiation frequency  $\nu$  (i.e. the photon energy  $h\nu$ ) is also measured in eV.

Below the system of units, built upon the four base units (2.26) and (2.28) with a freely chosen set of the  $[m]$ ,  $[l]$ ,  $[t]$ , and  $[T]$  values, is called the *RALEF-code units*. Table 2.2 lists several secondary (derived) units expressed in terms of the four base ones for selected physical quantities, related to thermal properties, heat and radiation transport. Generally, certain equations and formulae — like the basic radiation hydrodynamics equations (2.1)–(2.6) — remain invariant when the base units  $[m]$ ,  $[l]$ ,  $[t]$ , and  $[T]$  are changed, but others do not. Below all the unit-dependent formulae that are cast in the form suitable for incorporation into the RALEF-code have the equality sign marked as  $\stackrel{\text{R}}{=}$  [like in Eq. (2.27)].

TABLE 2.2: Derived units of measurement in heat and radiation transport.

quantity	notation	unit	dimensions in CGS
radiation frequency	$\nu$	$h^{-1}[T]$	Hz
radiation intensity	$I_\nu, B_\nu$	$h[m][t]^{-3}[T]^{-1}$	$\text{erg cm}^{-2} \text{ s}^{-1} \text{ ster}^{-1} \text{ Hz}^{-1}$
group intensity	$B, \int I_\nu d\nu$	$[m][t]^{-3}$	$\text{erg cm}^{-2} \text{ s}^{-1} \text{ ster}^{-1}$
thermal conduction coefficient	$\kappa$	$[m][l][t]^{-3}[T]^{-1}$	$\text{cm}^{-1} \text{ s}^{-1}$
mass-specific heat capacity	$c_V$	$[l]^2[t]^{-2}[T]^{-1}$	$\text{g}^{-1}$

The Planckian spectral and frequency-integrated radiation intensities in the RALEF-code units are given by

$$B_\nu \stackrel{\text{R}}{=} K_{Pl} \frac{\nu^3}{e^{\nu/T} - 1}, \quad B \stackrel{\text{R}}{=} K_{Pl} \int_0^\infty \frac{\nu^3 d\nu}{e^{\nu/T} - 1}, \quad (2.29)$$

where

$$K_{Pl} \stackrel{\text{R}}{=} \frac{2}{h^3 c^2} \frac{[t]^3 [T]^4}{[m]} \stackrel{\text{R}}{=} 7.64926047 \times 10^{57} \frac{[t]^3 [T]^4}{[m]}. \quad (2.30)$$

Expressions (2.12) for  $B(T)$ , (2.14) for the black-body flux  $h_{r,Pl}$ , and (2.15) for the black-body radiation energy density  $\mathcal{E}_{r,Pl}$  preserve their form, provided that the Stefan-Boltzmann and the Stefan constants are given by

$$\sigma_{SB} \stackrel{\text{R}}{=} \frac{\pi^5}{15} K_{Pl} \stackrel{\text{R}}{=} 1.56054952 \times 10^{59} \frac{[t]^3 [T]^4}{[m]}, \quad (2.31)$$

$$a_{St} \stackrel{\text{R}}{=} 2.08217315 \times 10^{49} \frac{[l][t]^2 [T]^4}{[m]}. \quad (2.32)$$

## 2. The DEIRA units.

In our notation, the parent CGS system of units corresponds to  $[m] = [l] = [t] = 1$ . These units are, however, rather inconvenient for most applications in the physics of high energy density states, where, for example, typical plasma velocities are often 6–8 orders of magnitude higher than 1 cm/s. A more suitable unit system would, say, be the one with  $[m] = 10^{-3}$ ,  $[l] = 0.1$ ,  $[t] = 10^{-8}$ , i.e. a system of units based on mass, length, and time units of 1 mg, 1 mm, and 10 ns. With this choice the mass density is measured in  $\text{g/cm}^3$ , the velocity in  $10^7 \text{ cm/s}$ , the pressure in  $10^{14} \text{ erg/cm}^3 = 100 \text{ Mbar}$ , the mass-specific energy in  $10^{14} \text{ erg/g} = 10 \text{ MJ/g}$ ; the external heating rate  $Q_{dep}$  should be given in  $10^{22} \text{ erg cm}^{-3} \text{ s}^{-1} = 1 \text{ PW/cm}^3$ . Having added a temperature unit of  $[T] = 1.60217733 \times 10^{-9} = 1 \text{ keV}$ , below we refer to these system as the DEIRA units — after the 1D DEIRA code, developed for simulation of heavy-ion driven ICF targets [5]. The unit-dependent formulae in the DEIRA units are marked by the equality sign  $\stackrel{\text{D}}{=}$ . In the DEIRA units the values of the above mentioned constants become

$$K_{Pl} \stackrel{\text{D}}{=} 50.403626, \quad a_{St} \stackrel{\text{D}}{=} 1.372016, \quad \sigma_{SB} \stackrel{\text{D}}{=} 1028.3001. \quad (2.33)$$

The DEIRA unit for the coefficient of thermal conduction  $\kappa$  is  $10^{20}$  erg cm<sup>-1</sup> s<sup>-1</sup> keV<sup>-1</sup>. The DEIRA units is the default option for the units of measurements in the present version of the RALEF-2D code.

### 3. Ideal-gas equation of state

The ideal-gas equation of state of a plasma, composed of single-species ions with a constant mean ionization degree  $z_i$ , has the form

$$p = (n_e + n_i)T = \frac{\rho}{m_u} \frac{1 + z_i}{A} T \stackrel{\text{R}}{=} 6.022137 \times 10^{23} \frac{[T][t]^2}{[l]^2} \frac{1 + z_i}{A} \rho T, \quad (2.34)$$

$$e = \frac{3}{2} \frac{n_e + n_i}{\rho} T = \frac{3}{2} \frac{1 + z_i}{A} \frac{T}{m_u} \stackrel{\text{R}}{=} 6.022137 \times 10^{23} \frac{[T][t]^2}{[l]^2} \frac{3}{2} \frac{1 + z_i}{A} T, \quad (2.35)$$

where  $m_u = 1.6605402 \times 10^{-24}$  g is the atomic mass unit, and  $A$  is the atomic mass of plasma ions. The mass-specific heat capacity

$$c_V = \left( \frac{\partial e}{\partial T} \right)_V \quad (2.36)$$

of such a plasma is given by

$$c_V = \frac{3}{2} \frac{1 + z_i}{A m_u} \stackrel{\text{R}}{=} 6.022137 \times 10^{23} \frac{[T][t]^2}{[l]^2} \frac{3}{2} \frac{1 + z_i}{A}. \quad (2.37)$$

### 4. The inverse bremsstrahlung absorption coefficient

As an example of opacity calculation, we consider the absorption coefficient for inverse bremsstrahlung [6]

$$k_{\nu,ff}(\nu, T_e) = \frac{32\pi^3}{3\sqrt{6}\pi} \alpha a_0^5 \left( \frac{e^2/a_0}{T_e} \right)^{1/2} \left( \frac{e^2/a_0}{h\nu} \right)^3 (1 - e^{-h\nu/T}) n_e n_i z_i^2 \bar{g}, \quad (2.38)$$

in a plasma of hydrogen-like ions with an electric charge  $+ez_i$ ; here  $e$  is the elementary charge,  $\alpha = e^2/\hbar c = 1/137.036$  is the fine structure constant,  $a_0 = \hbar^2/m_e e^2 = 0.52918 \times 10^{-8}$  cm is the Bohr radius,  $n_e = z_i n_i$  is the number of free electrons per unit volume,  $n_i = \rho/A m_u$  is the number of ions per unit volume,  $\bar{g}$  is the mean value of the Gaunt factor. This absorption coefficient is already corrected for stimulated emission.

From Eq. (2.38) one readily calculates the absorption coefficient, averaged over a spectral interval  $\nu \in [\nu_k, \nu_{k+1}]$  with the Planckian weight function  $B_\nu \propto \nu^3 (e^{h\nu/T} - 1)^{-1}$  (the Planckian mean)

$$\begin{aligned} k_{ff,[k]} &\stackrel{\text{R}}{=} K_{ff} \left( \frac{\rho}{A} \right)^2 \frac{z_i^3 \langle g \rangle_{[k]}}{T^{1/2}} \frac{\int_{\nu_k}^{\nu_{k+1}} e^{-\nu/T} d\nu}{\int_{\nu_k}^{\nu_{k+1}} \nu^3 (e^{\nu/T} - 1)^{-1} d\nu} \\ &\stackrel{\text{R}}{=} K_{ff} \left( \frac{\rho}{A} \right)^2 \frac{z_i^3 \langle g \rangle_{[k]}}{T^{7/2}} \frac{e^{-x_k} - e^{-x_{k+1}}}{\Phi(x_k) - \Phi(x_{k+1})}, \end{aligned} \quad (2.39)$$

where

$$K_{ff} \stackrel{\text{R}}{=} 4.577350 \times 10^{-31} \frac{[m]^2}{[\mu]^5 [T]^{7/2}} \stackrel{\text{D}}{=} 0.2780532, \quad x_k = \frac{\nu_k}{T}, \quad (2.40)$$

$\langle g \rangle_{[k]}$  is the group-mean value of the Gaunt factor, and

$$\Phi(x) = \int_x^\infty \frac{t^3}{e^t - 1} dt. \quad (2.41)$$

For fast numerical evaluation one can use an approximation

$$\frac{t}{1 - e^{-t}} \approx t + e^{-t/2}, \quad (2.42)$$

which yields

$$\Phi(x) \approx (x^3 + 3x^2 + 6x + 6)e^{-x} + \frac{2}{3} \left( x^2 + \frac{4}{3}x + \frac{8}{9} \right) e^{-3x/2}. \quad (2.43)$$

The relative error of the approximation (2.42) does not exceed 2.4%.

The full (i.e. over the entire spectrum  $0 \leq \nu \leq \infty$ ) Planckian and Rosseland mean absorption coefficients are given by

$$k_{ff,P} \stackrel{\text{R}}{=} \frac{15}{\pi^4} K_{ff} \left( \frac{\rho}{A} \right)^2 \frac{z_i^3 \langle g \rangle}{T^{7/2}} \stackrel{\text{R}}{=} 0.15398973382 K_{ff} \left( \frac{\rho}{A} \right)^2 \frac{z_i^3 \langle g \rangle}{T^{7/2}}, \quad (2.44)$$

$$\begin{aligned} k_{ff,R} &\stackrel{\text{R}}{=} \frac{4\pi^4}{15} \left( \int_0^\infty \frac{x^7 e^{-x}}{(1 - e^{-x})^3} dx \right)^{-1} K_{ff} \left( \frac{\rho}{A} \right)^2 \frac{z_i^3 \bar{g}}{T^{7/2}} \\ &\stackrel{\text{R}}{=} 0.00508855177 K_{ff} \left( \frac{\rho}{A} \right)^2 \frac{z_i^3 \bar{g}}{T^{7/2}}; \end{aligned} \quad (2.45)$$

here  $\langle g \rangle$  and  $\bar{g}$  are the corresponding mean values of the Gaunt factor.

To describe the absorption of laser light with  $h\nu \lesssim 3$  eV in hot plasmas, usually the monochromatic formula (2.38) in the limit of  $h\nu \ll T$  is used. In our context it can be written as

$$k_{\nu,ff} \stackrel{\text{R}}{=} K_{ff} \left( \frac{\rho}{A} \right)^2 \frac{z_i^3 \bar{g}}{T^{3/2} \nu^2}. \quad (2.46)$$

The Maxwellian-mean Gaunt factor  $\bar{g}$  is most readily calculated in the Born approximation

$$\bar{g} = \frac{\sqrt{3}}{\pi} \exp\left(\frac{h\nu}{2T}\right) K_0\left(\frac{h\nu}{2T}\right), \quad (2.47)$$

where  $K_0(x)$  is the Macdonald function of order zero. In the limit of  $h\nu \ll T$  we can use the corresponding asymptotic expansion of  $K_0(x)$ , which leads us to a simple approximation

$$\bar{g} \stackrel{\text{R}}{=} \max \left\{ 1; \frac{\sqrt{3}}{\pi} \ln \left( 2.24584 \frac{T}{\nu} \right) \right\}. \quad (2.48)$$

#### 4. Thermal conduction

##### 1. Spitzer thermal conductivity

The Spitzer coefficient of electron thermal conductivity in a plasma of hydrogen-like ions  $+ez_i$ , as originally presented by L. Spitzer [7], is written as

$$\kappa_e = 20 \left( \frac{2}{\pi} \right)^{3/2} \frac{T_e^{5/2}}{m_e^{1/2} e^4 z_i \ln \Lambda} \epsilon \delta_T. \quad (2.49)$$

An analytic expression for the factor  $\epsilon \delta_T$  was first calculated by V.S. Imshennik [8] using the Chapman-Enskog method in the two-polynomial approximation. Later, it was independently obtained by M. Lampe [9, 10], and still later recalculated by N. Bobrova and P. Sasorov [11]. The final result, as first given by Brysk *et al.* [12], reads

$$\epsilon \delta_T = \frac{15\pi}{256} \frac{45\zeta + 433\zeta^2}{9 + 151\zeta + 217\zeta^2}, \quad \zeta = \frac{1}{4\sqrt{2}} \frac{z_i \ln \Lambda_{ei}}{\ln \Lambda_{ee}}. \quad (2.50)$$

If we assume  $\ln \Lambda_{ei} = \ln \Lambda_{ee} = \ln \Lambda$ , we get

$$\begin{aligned} \kappa_e &= \frac{3 \cdot 5^3}{2^7 \sqrt{\pi}} \frac{1 + \frac{433}{180\sqrt{2}} z_i}{1 + \frac{151}{36\sqrt{2}} z_i + \frac{217}{288} z_i^2} \frac{T_e^{5/2}}{m_e^{1/2} e^4 \ln \Lambda} \\ &\stackrel{\text{R}}{=} 1.028908 \times 10^{51} \frac{[t]^3 [T]^{7/2}}{[m][l]} \frac{1 + \frac{433}{180\sqrt{2}} z_i}{1 + \frac{151}{36\sqrt{2}} z_i + \frac{217}{288} z_i^2} \frac{T_e^{5/2}}{\ln \Lambda}. \end{aligned} \quad (2.51)$$

The result of Imshennik differs slightly from Eq. (2.51) in that the factor by  $z_i^2$  is 212/288 instead of the correct value of 217/288.

To evaluate the Coulomb logarithm  $\ln \Lambda$ , we follow the original work of Spitzer [7]. We start with the formula

$$\Lambda = \frac{v/\omega_{pe}}{[(z_i e^2/1.123 m_e v^2)^2 + (\hbar/2m_e v)^2]^{1/2}}, \quad (2.52)$$

which combines the Bohr-Kramers classical formula with the Bethe-Lindhard-Larkin quantum formula for a fast ion  $z_i e$  moving with velocity  $v$  past motionless plasma electrons. Next, we change to the ion rest frame, replace the adiabatic impact parameter  $v/\omega_{pe}$  ( $\omega_{pe}$  is the plasma frequency) in the numerator with the Debye length  $\lambda_D$  given by

$$\lambda_D^{-2} = \frac{4\pi n_e e^2}{T_e} + \frac{4\pi n_i z_i^2 e^2}{T_i} = 4\pi n_e e^2 \left( \frac{1}{T_e} + \frac{z_i}{T_i} \right), \quad (2.53)$$

and replace  $m_e v^2$  in the denominator with  $3T_e$ . In result we obtain

$$\begin{aligned} \Lambda &= 3.369 \frac{T_e}{e^2} [\lambda_D^{-2} (z_i^2 + 0.945847 \hbar^2 T_e / m_e e^4)]^{-1/2} \\ &\stackrel{\text{R}}{=} 1.105164 \times 10^{16} \frac{([l][T])^{3/2}}{[m]^{1/2}} T_e \left[ \left( \frac{\rho z_i}{A} \right) \left( \frac{1}{T_e} + \frac{z_i}{T_i} \right) (z_i^2 + 2.1695 \times 10^{10} [T] T_e) \right]^{-1/2} \end{aligned} \quad (2.54)$$

When a flux limit is applied to the electron thermal conduction, the limiting heat flux  $h_l$  [erg cm<sup>-2</sup> s<sup>-1</sup>] is written in the form

$$h_l = f_{inh} n_e T_e \left( \frac{T_e}{m_e} \right)^{1/2} \stackrel{R}{=} 1.99529 \times 10^{37} \frac{[t]^3 [T]^{3/2}}{[l]^3} f_{inh} \left( \frac{\rho z_i}{A} \right) T_e^{3/2}, \quad (2.55)$$

where  $f_{inh} \simeq 0.03$ – $1$  is a dimensionless inhibition factor.

In the RALEF package the Spitzer conductivity is implemented as model # 3 in subroutine TCCOEF, file ‘f06\_eos.f’.

2. *A semi-empirical model for thermal conductivity of mono-atomic metals and plasmas in the mean ion approximation*

Starting from the Spitzer formula, a semi-empirical formula can be proposed for thermal conductivity of simple (mono-atomic) metals and plasmas, which is based on the mean ion approximation and matches the high-temperature Spitzer plasma limit with the measured conductivity at normal conditions by choosing an appropriate value of a single fitting parameter. It is implemented as model # 5 in subroutine TCCOEF (file ‘f06\_eos.f’) of the RALEF package.

The mono-atomic plasma of an element ( $A, Z$ ) with

$$n = \frac{\rho}{m_u A} \quad (2.56)$$

nuclei per unit volume is characterized by a mean ionization degree

$$z_{ion} = \frac{n_e}{n}, \quad 0 \leq z_{ion} \leq Z, \quad (2.57)$$

(provided by the EOS model), and is assumed to consist of identical point-like ions with a generally fractional charge

$$\tilde{z}_i = \max(1, z_{ion}) = \begin{cases} z_{ion}, & z_{ion} \geq 1, \\ 1, & z_{ion} < 1, \end{cases} \quad (2.58)$$

and with the number density

$$n_i = \begin{cases} n, & z_{ion} \geq 1, \\ n z_{ion} = n_e, & z_{ion} < 1. \end{cases} \quad (2.59)$$

Then, the coefficient of thermal conductivity due to free electrons in such a plasma can be evaluated by generalizing the Spitzer formula (2.51) to include (approximately) the effects of electron degeneracy and of the electron scattering by neutral atoms.

For the Coulomb logarithm  $\ln \Lambda$ , the degeneracy effects are accounted for by simply replacing  $T_e$  in Eq. (2.54) with

$$T_F = \sqrt{T_e^2 + \left( \frac{2}{3} E_F \right)^2}, \quad (2.60)$$

where

$$E_F = \frac{\hbar^2}{2m_e} (3\pi^2 n_e)^{2/3} \stackrel{\text{R}}{=} 4.16628 \times 10^{-11} \frac{[m]^{2/3}}{[l]^2 [T]} \left( \frac{\rho z_{ion}}{A} \right)^{2/3} \quad (2.61)$$

is the Fermi energy, because in the degenerate case it is the electrons at the Fermi surface (in the narrow energy range  $\Delta(\frac{1}{2}m_e v_e^2) \simeq T_e \ll E_F$ ) which determine the plasma transport properties, i.e. the electrons with the mean energy equal to  $E_F$  instead of  $\frac{3}{2}T_e$  [12]. At the same time, the factor  $T_e^{5/2}$  in the numerator of Eq. (2.51) must be split into  $T_e \tilde{T}_F^{3/2}$ , where

$$\tilde{T}_F = \sqrt{T_e^2 + (\beta_{ec} E_F)^2}, \quad \beta_{ec} = 0.34. \quad (2.62)$$

The numerical coefficient  $\beta_{ec}$  in Eq. (2.62) is adjusted such as to yield the correct value of  $\kappa_e$  in the limit of  $T_e \ll E_F$  for the degenerate Lorentzian plasma [9, 12].

Inclusion of the electron-atom scattering is done by recalling that generally

$$\kappa_e = \text{const} \cdot \frac{n_e T_e}{m_e \nu_e} = \text{const} \cdot \frac{n z_{ion} T_e}{m_e \nu_e}, \quad (2.63)$$

where the electron collision frequency  $\nu_e$  is the sum of the  $ei$ -collision and  $ea$ -collision terms

$$\nu_e = \nu_{ei} + \nu_{ea} = \text{const} \cdot \frac{e^4}{m_e^{1/2} \tilde{T}_F^{3/2}} n z_{ion} \tilde{z}_i L_{ei} + n \sigma_{ea} \left( \frac{T_F}{m_e} \right)^{1/2} \max(0, 1 - z_{ion}); \quad (2.64)$$

when evaluating the coefficient  $K_{ea}$  in Eq. (2.67), a fixed value of the electron-atom scattering cross-section  $\sigma_{ea} = 10^{-15} \text{ cm}^{-2}$  is used.

Finally, we obtain the following expression for the electron conduction coefficient

$$\begin{aligned} \kappa_e &= \frac{3 \cdot 5^3}{2^7 \sqrt{\pi}} \frac{\tilde{z}_i + \frac{433}{180\sqrt{2}} \tilde{z}_i^2}{1 + \frac{151}{36\sqrt{2}} \tilde{z}_i + \frac{217}{288} \tilde{z}_i^2} \frac{z_{ion} T_e \tilde{T}_F^{3/2}}{m_e^{1/2} e^4 \left[ z_{ion} \tilde{z}_i L_{ei} + K_{ea} \tilde{T}_F^{3/2} T_F^{1/2} \max(0, 1 - z_{ion}) \right]} \\ &\stackrel{\text{R}}{=} K_{ec} \frac{\tilde{z}_i + \frac{433}{180\sqrt{2}} \tilde{z}_i^2}{1 + \frac{151}{36\sqrt{2}} \tilde{z}_i + \frac{217}{288} \tilde{z}_i^2} \frac{z_{ion} T_e \tilde{T}_F^{3/2}}{z_{ion} \tilde{z}_i L_{ei} + K_{ea} \tilde{T}_F^{3/2} T_F^{1/2} \max(0, 1 - z_{ion})}, \end{aligned} \quad (2.65)$$

where

$$K_{ec} \stackrel{\text{R}}{=} 1.028908 \times 10^{51} \frac{[t]^3 [T]^{7/2}}{[m][l]} \stackrel{\text{D}}{=} 1.693806, \quad (2.66)$$

$$K_{ea} \stackrel{\text{R}}{=} 5.621421 \times 10^{21} [T]^2 \stackrel{\text{D}}{=} 1.4430 \times 10^4. \quad (2.67)$$

For the Coulomb logarithm  $\ln \Lambda_{ei}$  of  $ei$  collisions we use an interpolation expression

$$L_{ei} = \ln \left[ 1 + \frac{\Lambda_{ei}(g_{ec} + \Lambda_{ei})}{1 + \Lambda_{ei} + (6.5g_{ec}\Lambda_{ei})^{-1}} \right], \quad (2.68)$$

which has one free parameter  $g_{ec}$  for the limit of  $\Lambda_{ei} \ll 1$  [5, 13], and becomes  $\ln \Lambda_{ei}$  in the limit of  $\Lambda_{ei} \gg 1$ . The formula for  $\Lambda_{ei}$  is obtained from Eq. (2.54) by replacing  $T_e$  with  $T_F$ , and  $z_i$  with either  $z_{ion}$  (where appropriate) or  $\tilde{z}_i$ :

$$\begin{aligned} \Lambda_{ei} &= 3.369 \frac{T_F}{e^2} \left[ \lambda_D^{-2} (\tilde{z}_i^2 + 0.945847 \hbar^2 T_F / m_e e^4) \right]^{-1/2} \\ &\stackrel{\text{R}}{=} 1.105164 \times 10^{16} \frac{([l][T])^{3/2}}{[m]^{1/2}} T_F \left[ \left( \frac{\rho z_{ion}}{A} \right) \left( \frac{1}{T_F} + \frac{\tilde{z}_i}{T_i} \right) (\tilde{z}_i^2 + 2.1695 \times 10^{10} [T] T_F) \right]^{-1/2} \end{aligned} \quad (2.69)$$

When a flux limit is applied to the electron thermal conduction, the limiting heat flux  $h_l$  [erg cm<sup>-2</sup> s<sup>-1</sup>] is written in the form

$$h_l = f_{inh} n_e T_e \left( \frac{T_F}{m_e} \right)^{1/2} \stackrel{R}{=} 1.99529 \times 10^{37} \frac{[t]^3 [T]^{3/2}}{[l]^3} f_{inh} \left( \frac{\rho z_{ion}}{A} \right) T_e \sqrt{T_F}, \quad (2.70)$$

where  $f_{inh} \simeq 0.03$ – $1$  is a dimensionless inhibition factor.

The value of the only fitting parameter  $g_{ec}$  in Eq. (2.68) should normally be adjusted such as to reproduce the empirical value of thermal conductivity  $\kappa_0$  in a liquid metal near normal conditions at  $\rho = \rho_0$ ,  $T = T_0 \simeq 300$ – $1000$  K. It is assumed that the “ionization degree” (equal to the number of conduction electrons per atom and taken from the employed equation of state)  $z_0$  under this conditions takes on a value  $z_0 \gtrsim 1$ . Since near normal conditions one typically has  $T_e = T_i = T \ll T_F \ll z_0^2 (m_e c^4 / \hbar^2)$ , the above formulae yield (near  $z_0 \approx 3.5$ )

$$\kappa_{0[W/K\ m]} \approx 200 \frac{z_0^2}{g_{ec}^2} \left( \frac{\rho_0 z_0}{A} \right)^{2/3}, \quad (2.71)$$

where  $\kappa_{0[W/K\ m]}$  is the tabular value of the conduction coefficient measured in W/K·m, and  $\rho_0$  is measured in g/cm<sup>3</sup>. The conversion formula from the conventional unit [W m<sup>-1</sup> K<sup>-1</sup>] for  $\kappa$  to the DEIRA system of units is

$$\kappa_{[DEIRA]} = 1.1604 \times 10^{-8} \kappa_{[W/K\ m]}. \quad (2.72)$$

Typically, the values of parameter  $g_{ec}$  fall in the range  $g_{ec} \simeq 1$ – $10$ . For example, for liquid tin ( $Z = 50$ ,  $A = 118.7$ ,  $\rho_0 \approx 6.85$  g/cm<sup>3</sup> at  $T = 700$  K) the experimental value  $\kappa_0 \approx 41$  W/K·m is obtained with  $g_{ec} = 4.5$ , when the value  $z_0 = 3.5$  from the FEOS model is used.

---

*Correspondence with the code variables:*

$t =$	<b>t</b>	time;
$\rho =$	<b>rho</b>	matter density;
$\vec{u} =$	<b>uc</b>	matter velocity;
$p =$	<b>pr</b>	matter pressure;
$E =$	<b>te</b>	total mass-specific energy;
$e =$	<b>sie</b>	mass-specific internal energy;
$T =$	<b>temp</b>	matter temperature;
$I_\nu =$	<b>Ir radnu</b>	spectral radiation intensity;
$U_r =$	<b>U raden</b>	full integral radiation intensity;
$z_i =$	<b>zion</b>	ionization degree;
$[l] =$	<b>unilngth</b>	RALEF-code length unit [in centimeters];
$[t] =$	<b>unitime</b>	RALEF-code time unit [in seconds];
$[m] =$	<b>unimass</b>	RALEF-code mass unit [in grams];
$[T] =$	<b>unitemp</b>	RALEF-code temperature unit [in ergs];
$\sigma_{SB} =$	<b>ccStBol</b>	the Stefan-Boltzmann constant;
$K_{Pl} =$	<b>ccBPlck</b>	numerical coefficient for the Planckian intensity;
$K_{ff} =$	<b>cckffab</b>	numerical coefficient for the inverse bremsstrahlung absorption coefficient;

---

### 3. DISCRETIZATION OF THE FLUID ENERGY EQUATION

#### 1. Time discretization with the SSI method

Because the focus of this report is on the numerical algorithm for radiation energy transport, which is only coupled to the last of the three hydrodynamic equations (2.1)–(2.3), we begin with the discretization of the energy equation (2.3). Building upon the fully explicit CAVEAT scheme for ideal hydrodynamics [1], we start with a fully explicit finite-difference representation of Eq. (2.3) in the Lagrangian phase of the hydro algorithm in the form

$$M_i(\bar{E}_i - E_i) = \left( W_i^T + W_i^r + W_i^{dep} \right) \Delta t + W_i^p \Delta t. \quad (3.1)$$

Here and below usually a single global index  $i$  is used to identify mesh cells and nodes of the structured quadrilateral CAVEAT mesh. The relation between the single mesh index  $i$  and the more familiar block-local double index  $(i_1, i_2) \equiv (i, j)$  is given by

$$i \equiv \mathbf{I} = i_2 \cdot [\mathbf{n1}(\mathbf{iblk}) + 3] + i_1 + 1 + \mathbf{mob}(\mathbf{iblk}); \quad (3.2)$$

for more details see Ref. [1].

In Eq. (3.1)

$$M_i = \rho_i V_i \quad (3.3)$$

is the mass of matter in cell  $i$  (which is strictly conserved at the Lagrangian step),

$$V_i = \int_{A_i} R dx_1 dx_2 \quad (3.4)$$

is the volume of cell  $i$ ,

$$R = \begin{cases} 1, & \mathbf{iradial} = 0, \\ x_1, & \mathbf{iradial} = 1, \\ x_2, & \mathbf{iradial} = 2, \end{cases} \quad (3.5)$$

is the cylindrical radius,  $x_1$  and  $x_2$  are the two global mutually orthogonal coordinates (right-handed) in the 2D simulation plane, and  $A_i$  is the area of quadrilateral  $i$  in the  $(x_1, x_2)$  plane; below very often the two coordinates  $(x_1, x_2)$  are denoted as  $(x, y)$ . Note that in the case of Cartesian metric (i.e. for  $\mathbf{iradial} = 0$ ) all the extensive quantities (like mass  $M_i$ , volume  $V_i$ , etc.) are the corresponding 3D quantities per unit length along the normal to the computational  $(x_1, x_2)$  plane; in the  $rz$ -geometry (i.e. for  $\mathbf{iradial} = 1$  or  $2$ ) all the extensive quantities are the corresponding 3D quantities per radian of the global azimuth angle  $\phi$  around the rotation axis. Further on,  $E_i$  and  $\bar{E}_i$  in Eq. (3.1) are, respectively, the old (at time  $t$ ) and the new (at time  $t + \Delta t$ ) values of the mass-specific total energy  $E = e + u^2/2$  [erg g<sup>-1</sup>] in cell  $i$ , and

$$W_i^T = \int_{A_i} Q_T R dx_1 dx_2, \quad W_i^r = \int_{A_i} Q_r R dx_1 dx_2, \quad (3.6)$$

$$W_i^{dep} = \int_{A_i} Q_{dep} R dx_1 dx_2, \quad W_i^p = - \int_{A_i} \text{div}(p\vec{u}) R dx_1 dx_2 \quad (3.7)$$

are, respectively, the cell heating powers [erg/s] due to thermal conduction, radiation transport, external heat sources, and the pressure ( $p dV$ ) work.

Following the CAVEAT algorithm, we assume that the  $p dV$  term  $\tilde{W}_i^p$  is always calculated explicitly, i.e. by using the old values of all the principal variables — which implies that the numerical stability imposes a corresponding constraint [the Courant-Friedrichs-Lewy (CFL) condition] on the value of  $\Delta t$ . However, if we do the same for  $\tilde{W}_i^T$  and  $\tilde{W}_i^r$  (i.e. construct a fully explicit scheme), we will have to impose additional stability constraints on  $\Delta t$ , which would make such a scheme impractical when intense conductive and radiative energy fluxes are generated. These additional stability constraints can be avoided if one chooses a partially implicit approach, where  $\tilde{W}_i^p$  is still calculated explicitly, while  $\tilde{W}_i^T$  and  $\tilde{W}_i^r$  are calculated partially implicitly by using intermediate new temperature values  $\tilde{T}_i$ , obtained from the numerical solution (at the Lagrangian step only!) of a static heat transfer equation

$$\rho c_V \frac{\partial T}{\partial t} = Q_T + Q_r + Q_{dep}, \quad (3.8)$$

where  $c_V$  is the mass-specific heat capacity at constant volume. When solving Eq. (3.8), the mesh configuration and the density distribution are fixed, and only the matter temperature  $T$  is allowed to change. As a result, no additional stability constraints on  $\Delta t$  arise when the new values  $\tilde{W}_i^T$  and  $\tilde{W}_i^r$ , calculated from the discretized form

$$c_{V,i} M_i (\tilde{T}_i - T_i) = (\tilde{W}_i^T + \tilde{W}_i^r + W_i^{dep}) \Delta t \quad (3.9)$$

of equation (3.8) are used in Eq. (3.1) instead of  $W_i^T$  and  $W_i^r$ . In Eq. (3.9) the heating powers  $\tilde{W}_i^T$  and  $\tilde{W}_i^r$  are calculated with the new temperatures  $\tilde{T}_i$  but the old values of densities  $\rho_i$ , heat capacities  $c_{V,i}$ , cell volumes  $V_i$ , etc.

Unfortunately, for radiation transport the literal implementation of such partially implicit algorithm is out of question because, for any given  $i$ , the value of  $\tilde{W}_i^r$  formally depends on the  $\tilde{T}_i$  values in all the mesh cells [via the transfer equation (2.6)]; hence, in order to find  $\tilde{T}_i$ , one has to solve a huge system of non-linear equations with a non-sparse matrix. A possible way out of this predicament [14] is to employ the symmetric semi-implicit (SSI) method proposed in Ref. [15]. In this method the  $\tilde{W}_i^r$  value in cell  $i$  is calculated by using the new temperature  $\tilde{T}_i$  in only the same cell  $i$ , and the old temperatures  $T_j$  in all the remaining cells  $j \neq i$ . Finally, introducing a notation

$$\tau_i \equiv \tilde{T}_i - T_i, \quad (3.10)$$

and linearizing the dependence of  $\tilde{W}_i^T$  and  $\tilde{W}_i^r$  on the central temperatures  $\tilde{T}_i$  as

$$\tilde{W}_i^T = -D_i^T \tau_i + W_i^T, \quad \tilde{W}_i^r = -D_i^r \tau_i + W_i^r, \quad (3.11)$$

where

$$D_i^T = -\frac{\partial W_i^T}{\partial T_i}, \quad D_i^r = -\frac{\partial W_i^r}{\partial T_i}, \quad (3.12)$$

we obtain the following discretized version of the full energy equation (2.3)

$$\bar{E}_i - E_i = c_{V,i} \tau_i + \frac{\Delta t}{M_i} W_i^p, \quad (3.13)$$

where

$$\tau_i = \frac{(W_i^T + W_i^r + W_i^{dep})\Delta t + \delta_i^T + \delta_i^r}{c_{V,i}M_i + \Delta t(D_i^T + D_i^r)}. \quad (3.14)$$

In addition to the above described terms, Eq. (3.14) also contains energy corrections  $\delta_i^T$  and  $\delta_i^r$  that are taken from the previous time step, and that are needed to restore the conservativeness of the SSI algorithm. One readily verifies that, if the original fully explicit scheme (3.1) conserves energy (when, for example, based on fluxes across the cell faces), then the SSI method inevitably violates energy conservation. The global energy balance can be restored by redepositing the numerically lost energy during the next time step (hydrocycle) as correction terms  $\delta_i^T$  and  $\delta_i^r$ ; for more details see Ref. [15].

Implementation of the SSI algorithm for thermal conduction — i.e. the algorithm for calculating the  $W_i^T$ ,  $D_i^T$  and  $\delta_i^T$  terms in Eq. (3.14) — has been described in the previous report [2] and in Ref. [3]. Below we describe the algorithm for calculating the radiative terms  $W_i^r$ ,  $D_i^r$  and  $\delta_i^r$ .

## 2. Time step control

Below we apply the term *thermal processes* to physical processes associated with the heating-cooling terms on the right-hand of Eq. (3.1) other than the  $p dV$  work. In our case we have three such terms due, respectively, to thermal conduction  $W_i^T$ , radiation transport  $W_i^r$ , and possible external energy deposition  $W_i^{dep}$ . Because time discretization of these thermal terms is done by using the SSI method rather than fully implicit one, we have to impose additional (with respect to the usual CFL condition) “thermal” restrictions on the value of the time step  $\Delta t$ . These restrictions are based on the requirement that the temperature increment  $|\tilde{T}_i - T_i|$  in cell  $i$  at the SSI phase of the Lagrangian step must not be too large, namely, on the condition

$$\left| \tilde{T}_i - T_i \right| = \left| \frac{W_i \Delta t + \delta_i}{c_{V,i}M_i + D_i \Delta t} \right| \leq \varepsilon_0 (T_i + T_s), \quad (3.15)$$

where  $W_i = W_i^{dep} + W_i^T + W_i^r$  is the total thermal heating power of cell  $i$ ,  $c_{V,i}M_i$  its heat capacity,  $D_i = -\partial W_i / \partial t$ ,  $\delta_i = \delta_{T,i} + \delta_{r,i}$  is the SSI energy correction, taken from the previous hydro cycle, and  $\varepsilon_0$  and  $T_s$  are two user-defined parameters.

Presently there are two versions of the thermal limit on  $\Delta t$  implemented in the RALEF-2D code: a “hard” one and a “soft” one. They are distinguished by the value of the user-defined parameter  $c_{1,tst}$ : for  $c_{1,tst} = 0$  the “hard” version is active, for  $c_{1,tst} > 0$  the “soft” one applies. The “hard” version of the time-step limit corresponds to the most strict implementation of condition (3.15). Here, however, the main obstacle is the fact that, when  $\delta_i$  is calculated, the value of  $\Delta t$  for the next hydro cycle (where  $\delta_i$  must be redeposited) is not known. As a consequence, the “hard” time-step limit is realized by splitting the criterion (3.15) into the following two conditions

$$\left| \frac{W_i \Delta t}{c_{V,ij}M_{ij} + \Delta t D_i} \right| \leq (\varepsilon_0 - \varepsilon_1) (T_i + T_s), \quad (3.16)$$

$$\left| \frac{\delta_i}{c_{V,i}M_i} \right| \leq \varepsilon_1 (T_i + T_s), \quad (3.17)$$

where  $\varepsilon_1$  is an additional user-defined accuracy parameter in our criterion. Clearly, one must always have  $\varepsilon_1 < \varepsilon_0$ . The bar over  $\delta_i$  in Eq. (3.17) means that this is a “postponed” quantity, to be used for calculating  $\tilde{T}_i - T_i$  only in the next hydrocycle. Because  $\bar{\delta}_i$  is, in its turn, proportional to the current value of  $\Delta t$  (see Eq. (??) below and Eq. (17) in Ref. [3]), one can for any arbitrarily small  $\varepsilon_0 > 0$  always satisfy conditions (3.16) and (3.17) by choosing a sufficiently small  $\Delta t > 0$  — which, in its turn, ensures fulfillment of the basic condition (3.15).

The principal drawback of the “hard” limit (3.16) is that in many practical situations, where one has  $D_i \Delta t \gg 1$ , this constraint turns out to be too restrictive and either unnecessarily significantly slows down the simulation or completely blocks it. Typically it occurs when strong heat-conduction fluxes are present in a tenuous medium with relatively small values of cell heat capacities  $c_{V,i} M_i$ . To speed up simulation of such cases, an alternative “soft” version of the criterion (3.15) has been implemented.

The “soft” time-step control is based on the following strategy: we keep track of the relative temperature change

$$\frac{\delta T}{T} = \frac{\delta T(\Delta t)}{T} \stackrel{\text{def}}{=} \max_i \left\{ (T_i + T_s)^{-1} \left| \frac{W_i \Delta t + \delta_i}{c_{V,i} M_i + D_i \Delta t} \right| \right\} \quad (3.18)$$

and make small corrections to  $\Delta t$  in order to keep  $\delta T/T$  within a range  $0.5\varepsilon_0 < \delta T/T < 0.6\varepsilon_0$ . More precisely, the SSI phase of every hydrocycle starts with a trial value of the time step

$$\Delta t_* = \min \{ c_{dtgr} \Delta t_{prev}; \Delta t_{CFL}; \Delta t_{ev} \}, \quad (3.19)$$

where  $c_{dtgr} > 1$  is a user-defined growth factor (typically,  $c_{dtgr} = 1.05\text{--}1.1$ ),  $\Delta t_{prev}$  is the value of  $\Delta t$  in the previous hydrocycle,  $\Delta t_{CFL}$  is the value of  $\Delta t$  obtained from the purely hydrodynamic CFL criterion, and  $\Delta t_{ev}$  is some other eventual time step limit. Next, the maximum relative temperature change  $\delta T(\Delta t_*)/T$  is calculated for the starting value  $\Delta t_*$ , which is subsequently modified by the following “fine-tuning” correction)

$$\Delta t = \begin{cases} \Delta t_*, & \delta T(\Delta t_*)/T \leq 0.5\varepsilon_0, \\ \min\{\Delta t_{prev}, \Delta t_*\}, & 0.5\varepsilon_0 < \delta T(\Delta t_*)/T \leq 0.6\varepsilon_0, \\ \frac{\min\{\Delta t_{prev}, \Delta t_*\}}{1 + c_{1,tst}(c_{dtgr} - 1)}, & 0.6\varepsilon_0 < \delta T(\Delta t_*)/T \leq 1.2\varepsilon_0. \end{cases} \quad (3.20)$$

This correction is made only once, i.e. the above “fine-tuning” procedure is non-iterative. In Eq. (3.20) one can use the values  $c_{1,tst} > 1$  to speed up the reduction of  $\Delta t$  relative to its growth rate prescribed by the parameter  $c_{dtgr} > 1$ .

Occasionally the “fine-tuning” procedure (3.20) becomes unstable and one ends up with a large temperature variation  $\delta T(\Delta t_*)/T > 1.2\varepsilon_0$ . Such a situation is considered as a failure (or a “crash”) of the “soft” time-step control procedure. Here we have to impose a more dramatic reduction of  $\Delta t$  to bring down the temperature variation  $\delta T/T$ , and it has to be done in an iterative loop. The problem however is that  $\delta_i$ , calculated in the previous hydrocycle, remains unchange as  $\Delta t \rightarrow 0$  — which means that generally we cannot satisfy (3.15) by making  $\Delta t$  sufficiently small. As a practical solution, we choose to sacrifice energy

conservation and scale down the values of  $\delta_i$  (in proportion to  $\Delta t$ ) in those cells  $i$  where  $(\delta T/T)_i > 0.5\varepsilon_0$ . In other words, when the “fine-tuning” procedure (3.20) crashes, we violate strict energy conservation and bring down the temperature change to a level  $\delta T/T < 0.5\varepsilon_0$  by strong reduction of  $\Delta t$  in an iterative loop similar to that in the “hard” version of the thermal time-step control algorithm. If  $D_i\Delta t \gg 1$ , the “crash”-induced reduction of  $\Delta t$  may be by several orders of magnitude.

For optimal performance, the following values of the user-defined control parameters can be recommended:

- for the “hard” option of the thermal time-step control:

$$c_{1,tst} = 0, \quad \varepsilon_0 = 0.1\text{--}0.2, \quad \varepsilon_1 = 0.5\varepsilon_0; \quad (3.21)$$

- for the “soft” option of the thermal time-step control:

$$c_{1,tst} = 1\text{--}2, \quad \varepsilon_0 = 0.06. \quad (3.22)$$

In the “soft” option, the value of  $\varepsilon_1$  is irrelevant. The values of  $c_{dtgr}$  are allowed to be in the range  $1 < c_{dtgr} < 2$ , recommended is  $c_{dtgr} = 1.05\text{--}1.1$ . Practical experience is that in many cases the “soft” option of the thermal time-step control allows to reduce the computing time by a significant factor 2–3 or even more at tolerable levels of energy disbalance.

### 3. Discretization of radiation transport: general approach

From the point of view of hydrodynamics, the numerical algorithm for radiation transport must provide the cell-centered quantities  $W_i^r$ ,  $D_i^r = -\partial W_i^r / \partial T_i$ , and  $\delta_i^r$  that are to be calculated for a given distribution of the cell-centered thermodynamic variables  $\{\rho_i, T_i\}$  on a given quadrilateral mesh  $\{x_i, y_i\}$ . The radiation-matter coupling quantities  $W_i^r$ ,  $D_i^r$ ,  $\delta_i^r$  can be computed only after one solves the transfer equation (2.6). Since the latter contains no time derivatives, the numerical scheme for radiation transport involves only spatial discretization, and logically naturally splits into two parts:

- *part 1, integration of the transfer equation:* here, given the values of  $\{\rho_i, T_i\}$  on the spatial mesh  $\{x_i, y_i\}$ , one computes the radiation field, i.e. the discrete radiation intensities  $\{I_{[k],L,i}\}$  for prescribed discrete sets of frequency groups  $[k]$ , angular directions  $L$ , and mesh points  $i$ .
- *part 2, calculation of the deposition rate:* here, given the radiation field  $\{I_{[k],L,i}\}$  on a discrete space-direction-frequency grid, one computes the values of  $W_i^r$ ,  $D_i^r$  and  $\delta_i^r$ .

Since our basic transfer equation (2.6) coincides with a particular case of the stationary neutron transport equation without scattering [16], where the source function  $B_\nu$  does not depend on the intensity  $I_\nu$ , we can benefit from numerous research on numerical methods in the neutron transport theory. In particular, we want to solve it on a discrete grid  $\{\vec{\Omega}_L\}$  of angular variable  $\vec{\Omega}$  by applying an appropriate method of discrete ordinates that would satisfy the following important conditions

- (1) it should be conservative,

- (2) it should be strictly positive, i.e. should guarantee  $I_{[k],L,i} > 0$  for  $B_\nu > 0$ , and
- (3) it should reproduce (or obey) the diffusion limit in the sense that in the limit of *mean free path*  $\ll$  *characteristic length scale* we must get  $W_i^r \rightarrow Q_{r,i}V_i$ , where  $Q_r$  is given by Eq. (2.23).

Here and below the term *conservative* is understood in the sense that the cell heating power  $W_i^r$  is obtained from a balance equation, which is a discrete version of the divergence theorem for the transfer equation (2.6) applied to the quadrilateral  $i$ , and which contains well defined fluxes across the cell edges (faces) that are the same for the two cells on both sides of the corresponding edge (face).

From the point of view of practical applications of the radiation hydrodynamics, the most important (in our opinion, indispensable) are the conditions (3) and (2). In radiation-dominated plasmas violation of either (2) or (3) may easily lead to either wrong physical results or full inability to obtain a solution of the problem. Because  $Q_r$  in the diffusion limit is proportional to the second derivative of the source function  $B$ , and because the optical thickness of the mesh cells in this limit is often either comparable to or much greater than 1, the radiation transport scheme must be of high accuracy (at least in some of its parts). From the studies on neutron transport it is known [17] that it is difficult to combine (1) and (2) for high-accuracy schemes even on simple rectangular grids. For example the numerical scheme, advocated by Larsen *et al.* [18] as fully satisfying (3), is not strictly positive. The algorithm implemented in the MULTI-2D code [14] satisfies (1) and (2), but the authors' arguments that it correctly reproduces the diffusion limit do not sound convincing, especially for arbitrary optical thickness of the mesh cells and non-uniform spatial grids; no explicit demonstration of convergence to the exact diffusion solution has been presented.

In the present work we sacrifice conservativeness and construct an algorithm which satisfies conditions (2) and (3). Part 1 is realized by an as exact as possible integration of Eq. (2.6) along the short characteristics. Generally, the characteristic methods have an advantage of being open to straightforward means of increasing the computational accuracy. The method of short characteristics [19] has another important advantage of uniform coverage of the logical mesh space in cases of spatially highly non-uniform meshes, where, for the same amount of computing effort, it generates considerably less numerical noise in the energy coupling terms  $W_i^r$  than, for example, the method of long characteristics. Because the radiation intensity  $I_\nu$  by its physical meaning is a flux-associated quantity, we make a natural choice of mesh vertices  $\{x_i, y_i\}$  to be the collocation points for the discrete intensities  $\{I_{[k],L,i}\}$ , with linear interpolation along the edges of mesh quadrilaterals.

Part 2 of the algorithm is based on an elaborate scheme of integration of the residual intensity  $F_\nu \equiv I_\nu - B_\nu$  over the area of mesh quadrilaterals, which leads to a finite-difference expression for  $W_i^r$  that uses only the cell-entry values of the intensity  $I_{[k],L}$ . Hence, our numerical method for radiation transport may be called as a characteristic entry-flux-only (CFEO) scheme.

---

*Correspondence with the code variables:*

$c_{1,tst}$  = `c1dtssi` – user-defined parameter which controls application of a “hard” or a “soft” thermal time-step limit; default = 0;  
 $c_{dtgr}$  = `dtgrow` – user-defined growth factor for increasing the time step  $\Delta t$ ; default = 1.05;

$\varepsilon_0$	= <code>eps0ssi</code> – principal time-step control parameter at the SSI stage; default = 0.1;
$\varepsilon_1$	= <code>eps1ssi</code> – secondary time-step control parameter at the SSI stage; default = 0.05;
$T_s$	= <code>tempsns</code> – sensitivity threshold for $T$ variation;
$T_{flr}$	= <code>tempflr</code> – absolute minimum for $T$ values; default = <code>floor</code> ;
$\delta T/T$	= <code>dtmpssi</code> – relative temperature change at the SSI phase of the hydro cycle;
$\Delta t_{prev}$	= <code>dtprev</code> – time step in the previous hydro cycle;

---

## 4. RADIATION TRANSPORT

### 1. Discretization of frequencies

#### 1. General formulae

Because the transfer equation (2.6) for thermal radiation contains no derivatives with respect to the frequency variable  $\nu$ , discretization along the  $\nu$  axis is straightforward: we simply introduce  $N_\nu + 1$  discrete frequency values

$$\{\nu_k\} = \nu_1, \nu_2, \dots, \nu_{N_\nu+1}, \quad (4.1)$$

which delineate the boundaries of  $N_\nu \geq 1$  frequency intervals (groups)

$$[\nu_k, \nu_{k+1}], \quad k = 1, 2, \dots, N_\nu, \quad (4.2)$$

covering the total simulated frequency range  $\nu_1 \leq \nu \leq \nu_{N_\nu+1}$ . If  $N_\nu = 1$ , we normally assume that the covered range extends from  $\nu_1 = 0$  to  $\nu_2 = \infty$ . For every frequency group we define the *integrated group intensity* (or, simply, the *group intensity*) as

$$I_{[k]} = I_{[k]}(t, \vec{x}, \vec{\Omega}) = \int_{\nu_k}^{\nu_{k+1}} I_\nu(t, \vec{x}, \vec{\Omega}, \nu) d\nu, \quad (4.3)$$

and the mean *Planckian group intensity*

$$B_{[k]} = B_{[k]}(T) = \int_{\nu_k}^{\nu_{k+1}} B_\nu(\nu, T) d\nu. \quad (4.4)$$

Here and below index  $[k]$  marks the frequency group  $[\nu_k, \nu_{k+1}]$ . With only one frequency group  $N_\nu = 1$ , we have  $B_{[1]} = B$ , where  $B = B(T)$  is defined in Eq. (2.12). In the RALEF-code units the Planckian group intensities are given by

$$B_{[k]} \stackrel{\text{R}}{=} K_{Pl} \int_{\nu_k}^{\nu_{k+1}} \frac{\nu^3 d\nu}{e^{\nu/T} - 1}. \quad (4.5)$$

Evidently, if the the spectral groups  $\{\nu_k\}$  cover the entire range  $0 < \nu < \infty$ , then the Planckian group intensities obey the sum rule

$$\sum_k B_{[k]} = B(T) = \frac{\sigma_{SB} T^4}{\pi}. \quad (4.6)$$

We rewrite the transfer equation for the group intensities in the form

$$\vec{\Omega} \cdot \nabla I_{[k]} = k_{[k]} (B_{[k]} - I_{[k]}), \quad (4.7)$$

where

$$k_{[k]} = k_{[k]}(\rho, T) = \frac{\int_{\nu_k}^{\nu_{k+1}} k(\nu, \rho, T) B_\nu(\nu, T) d\nu}{\int_{\nu_k}^{\nu_{k+1}} B_\nu(\nu, T) d\nu} \quad (4.8)$$

is the Planckian-mean group absorption coefficient (or, simply, the *group absorption coefficient*).

Here it must be noted that Eq. (4.7) does not directly follow from the original transfer equation (2.6) because  $k_{[k]} I_{[k]} \neq \int_{\nu_k}^{\nu_{k+1}} k_\nu I_\nu d\nu$  when  $I_\nu$  differs from  $B_\nu$ . As a consequence, Eq. (4.7) with the Planckian-mean values of  $k_{[k]}$  may lead to large errors in the values of  $I_{[k]}$  in optically thin situations. However, if we have an optically thin plasma volume with  $I_{[k]} \ll B_{[k]}$ , a large percentage error in the value of  $I_{[k]}$  has only a weak effect on the plasma cooling rate  $Q_r$ , i.e. is irrelevant from the point of view of the fluid motion. For such situations, the advantage of using the Planckian-mean group opacities is that one ensures correct description of the plasma radiative cooling even on coarse frequency grids (or just with a single frequency group). In the opposite limit of a large optical thickness, the values of  $I_{[k]}$  approach  $B_{[k]}$ , and one automatically recovers the correct form of the group transfer equation (4.7). Nonetheless, the accuracy of the calculated fluid heating rate  $Q_r$  may be significantly compromised when  $k_\nu$  strongly fluctuates (as a function of frequency) between the optically thin and thick limits within a single spectral group. In such difficult cases the accuracy of the radiation-hydrodynamics simulation can only be improved by increasing the number of frequency groups.

Below, when constructing the numerical scheme for radiation transport, we use the quantity

$$F_\nu = I_\nu - B_\nu \quad (4.9)$$

as the primary variable to describe the radiation field, and we call it the *residual spectral intensity* (or, simply the *residual intensity*); the corresponding group-averaged quantity is given by

$$F_{[k]} = I_{[k]} - B_{[k]}. \quad (4.10)$$

To characterize the local spectrum of the radiation field, we use the *4 $\pi$ -integrated group intensity*

$$U_{[k]} = \int_{4\pi} I_{[k]} d\vec{\Omega}. \quad (4.11)$$

Then the total integral intensity, defined in Eq. (2.9), is given by

$$U_r = \sum_k U_{[k]}. \quad (4.12)$$

## 2. Frequency sets

Every simulation with the RALEF code can be performed with either one or two sets of discrete frequencies (4.1). The user makes the corresponding choice by specifying the value of the parameter `nfreqsets` (= 1 or 2). All the radiative energy exchange between fluid elements is computed with the first main frequency set `kfreqset = 1`: it is for this frequency set that the transfer equation (2.6) is solved at every time step and for every angular direction from the  $S_n$  quadrature.

The second frequency set `kfreqset = 2` (when loaded) is used in the postprocessor mode for diagnostic purposes: for this set, the transfer equation is only solved at preselected time moments and only for a preselected photon propagation direction along a preselected set of lineout rays; in contrast to the main radiation transport algorithm coupled to hydrodynamics, it is solved by the method of long characteristics. Typically, it is used to construct detailed output spectra with hundreds of spectral groups, whereas the number of spectral groups in the first set usually lies in the range  $N_\nu = 10$ –100. Solution of the transfer equation with the second frequency set is not parallelized.

The two sets of discrete frequencies are fully defined by the values of the integer parameters `nfreqsets`, `nfreqs(1:2)`, and the frequency partition array `freqns(1:nfrmax,1:2)` [of type `real(8)`]. Parameters `nfreqsets` and `nfreqs(1:2)` are assigned via the `namelist/input/` in file ‘`in2d`’. The array `freqns(1:nfrmax,1:2)`, where `nfrmax = max{nfreqs(1),nfreqs(2)}`, is loaded in the subroutine `RADINPT` (file ‘`f10_taskinpt.f`’), called immediately after having read out the `namelist/input/`. If tabular opacities are used and the opacity-table file ‘`TABLE_GLT-TCRAD`’ has been prepared with a different frequency partition, then preference is given to the partition in the table, i.e. the values of `freqns(1:nfrmax,1:2)`, loaded in the subroutine `RADINPT`, are overwritten by those from the table; `nfreqsets` is overwritten only if the opacity table has been prepared with `nfreqsets = 1`, whereas `nfreqsets = 2` has been assigned in the `namelist/input/`; any of the values `nfreqs(1:2)` from the `namelist/input/` is overwritten only when it is greater than 1 and not equal to the corresponding value in the opacity table.

---

*Correspondence with the code variables:*

$N_\nu =$	<code>nfreqs(m)</code>	total number of frequency groups in the frequency set <code>m</code> ;
$\nu_k =$	<code>freqns(k,m)</code>	lower end of the $k$ -th frequency interval in the frequency set <code>m</code> ;
$B_{[k],i} =$	<code>BRADNU(i)</code>	Planckian group intensity for the $k$ -th frequency group at vertex $i$ ;
$F_{[k],i} =$	<code>FRADNU(i)</code>	residual group intensity $I_{[k]} - B_{[k]}$ for the $k$ -th frequency group at vertex $i$ ;

---

## 2. Discretization of angular variables with the $S_n$ method

### 1. Discrete $S_n$ ordinates in the Cartesian $(x, y)$ and axial $(R, Z)$ 2D coordinate systems

In the  $S_n$  method of discrete ordinates the radiation intensity  $I_\nu = I_\nu(\vec{x}, \vec{\Omega}, \nu)$  at each point  $\vec{x}$  in space is calculated along a fixed discrete set of angular directions (beamlets)  $\vec{\Omega}_L$ ,

uniformly distributed over a unit sphere [16]; there are

$$N_\Omega = \frac{1}{2}k_\Omega(k_\Omega + 1) \quad (4.13)$$

directions per octant (see Fig. 4.1), where  $k_\Omega = n/2$ . Index  $k_\Omega = 1, 2, 3, \dots$  is the principal user-assigned parameter, which defines the order of the angular  $S_n$  quadrature used in a particular simulation. Beside the discrete ordinates  $\vec{\Omega}_L$ ,  $L = 1, 2, \dots, 8N_\Omega$ , the  $S_n$  quadrature provides the values of weights  $w_L$  such that any integral over the  $4\pi$  of the full solid angle is discretized as

$$\int_{4\pi} f(\vec{\Omega}) d\vec{\Omega} = \sum_{L=1}^{8N_\Omega} w_L f(\vec{\Omega}_L). \quad (4.14)$$

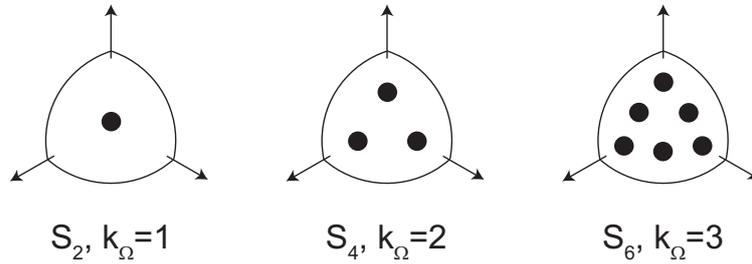


FIG. 4.1:  $S_n$  ordinates in the first octant for the first three  $S_n$  configurations.

The RALEF-2D code solves the equations of radiation hydrodynamics in two types of two-dimensional coordinate systems, having two different metrics: (i) the *Cartesian* coordinates  $(x_1, x_2) = (x, y)$  (*iradial=0*), which have the Cartesian metric  $dV = dx_1 dx_2 dx_3$  ( $dV$  is the 3D volume element, and nothing depends on  $x_3$ ), and (ii) the *cylindrical (axial)* coordinates  $(r, z)$  with either  $(x_1, x_2) = (r, z)$  (*iradial=1*) or  $(x_1, x_2) = (z, r)$  (*iradial=2*); in this case either  $dV = x_1 dx_1 dx_2 d\phi$ , or  $dV = dx_1 x_2 dx_2 d\phi$ , where nothing depends on the azimuthal angle  $0 \leq \phi \leq 2\pi$ . The above three options for the global orthogonal 2D coordinates  $(x_1, x_2)$  on the computational plane are distinguished by the user-defined parameter *iradial*, whose allowed values are 0, 1, and 2.

To describe radiation transport, we need a correspondence with the general 3D case, for which we introduce global Cartesian 3D coordinates  $(X, Y, Z)$ . The *2D Cartesian* coordinates describe translationally invariant flows, where fluid variables depend only on  $X$  and  $Y$  but not on  $Z$ . The *2D axial* coordinates describe axially-symmetric flows, where fluid variables depend on  $Z$  and  $R = (X^2 + Y^2)^{1/2}$ , but not on the azimuthal angle  $\phi = \arctan(X/Y)$ . To preserve this convention, ***here and everywhere below we assume that the rotational axis of the 2D  $rz$  geometry is always the  $Z$ -axis of the global 3D coordinate system.***

In both types of the 2D coordinates a mirror symmetry is observed for the function  $I_\nu = I_\nu(\vec{\Omega})$  in the 3D space. As a result, it is sufficient to consider discrete directions  $\vec{\Omega}_L$  in only 4 octants of a corresponding hemisphere, with the index  $L = 1, 2, \dots, 4N_\Omega$  running through  $4N_\Omega$  different values.

For coordinate representation of the unit vector  $\vec{\Omega}$  along the photon propagation direction, we introduce two types of its decomposition into three orthogonal components, namely,

- a *global* decomposition

$$\vec{\Omega} = \Omega_X \vec{n}_X + \Omega_Y \vec{n}_Y + \Omega_Z \vec{n}_Z, \quad (4.15)$$

where  $\vec{n}_X, \vec{n}_Y, \vec{n}_Z$  are, respectively, the fixed unit vectors along the  $X, Y,$  and  $Z$  global 3D coordinate axes, and

- a *local* decomposition

$$\vec{\Omega} = \Omega_{x_1} \vec{n}_{x_1} + \Omega_{x_2} \vec{n}_{x_2} + \Omega_{x_3} \vec{n}_{x_3}, \quad (4.16)$$

where  $\vec{n}_{x_1}, \vec{n}_{x_2}, \vec{n}_{x_3}$  are certain local mutually orthogonal unit vectors, which may vary from point to point in the 3D space.

The principal difference between the global and the local decompositions is that for any fixed direction  $\vec{\Omega}$  the global components  $\Omega_X, \Omega_Y, \Omega_Z$  remain constant in space, while the local components  $\Omega_{x_1}, \Omega_{x_2}, \Omega_{x_3}$  generally vary along the photon path, i.e. are functions of coordinates  $(X, Y, Z)$ . Of course, both types of  $\vec{\Omega}$  components obey the normalization

$$\Omega_X^2 + \Omega_Y^2 + \Omega_Z^2 = \Omega_{x_1}^2 + \Omega_{x_2}^2 + \Omega_{x_3}^2 = 1. \quad (4.17)$$

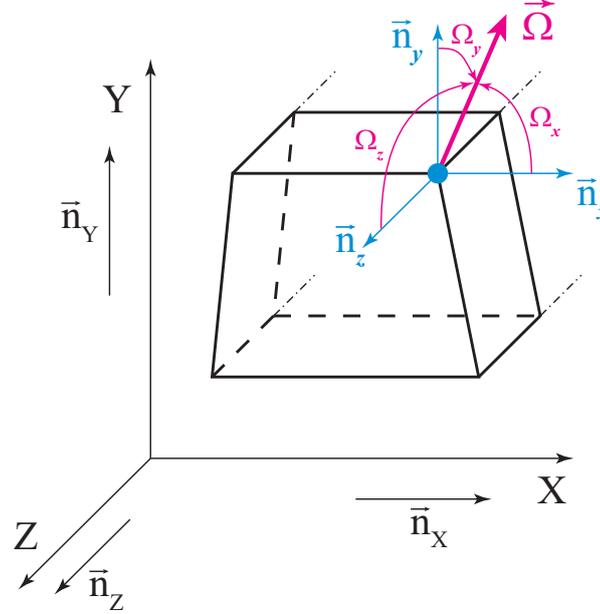


FIG. 4.2: Local decomposition of the photon propagation direction  $\vec{\Omega}$  in the Cartesian  $xy$  geometry with  $(x_1, x_2) = (x, y) = (X, Y)$ .

Then, the system of notation adopted for the three types of 2D coordinates used in the RALEF-2D code can be summarized as follows (for illustration see Figs. 4.2 and 4.3):

2D Cartesian coordinates  $(x_1, x_2) = (x, y)$  for `iradial` = 0:

$$\begin{aligned} x_1 &\equiv x = X, & \vec{n}_{x_1} &\equiv \vec{n}_x = \vec{n}_X, & \Omega_{x_1} &\equiv \Omega_x = \Omega_X = \vec{\Omega} \cdot \vec{n}_X, \\ x_2 &\equiv y = Y, & \vec{n}_{x_2} &\equiv \vec{n}_y = \vec{n}_Y, & \Omega_{x_2} &\equiv \Omega_y = \Omega_Y = \vec{\Omega} \cdot \vec{n}_Y, \\ x_3 &\equiv z = Z, & \vec{n}_{x_3} &\equiv \vec{n}_z = \vec{n}_Z, & \Omega_{x_3} &\equiv \Omega_z = \Omega_Z = \vec{\Omega} \cdot \vec{n}_Z. \end{aligned} \quad (4.18)$$

2D axial coordinates  $(x_1, x_2) = (R, Z)$  for `iradial = 1`:

$$\begin{aligned}
 x_1 \equiv x \equiv R &= (X^2 + Y^2)^{1/2}, & \vec{n}_{x_1} \equiv \vec{n}_x \equiv \vec{n}_R &= \vec{n}_X \cos \phi + \vec{n}_Y \sin \phi, \\
 x_2 \equiv y &= Z, & \vec{n}_{x_2} \equiv \vec{n}_y &= \vec{n}_Z, \\
 x_3 \equiv \phi &= \arctan(Y/X); & \vec{n}_{x_3} \equiv \vec{n}_z &= -\vec{n}_\phi = \vec{n}_X \sin \phi - \vec{n}_Y \cos \phi;
 \end{aligned} \tag{4.19}$$

$$\begin{aligned}
 \Omega_{x_1} \equiv \Omega_x \equiv \Omega_R &= \vec{\Omega} \cdot \vec{n}_R = \Omega_X \cos \phi + \Omega_Y \sin \phi, \\
 \Omega_{x_2} \equiv \Omega_y &= \Omega_Z = \vec{\Omega} \cdot \vec{n}_Z, \\
 \Omega_{x_3} \equiv \Omega_z &= -\vec{\Omega} \cdot \vec{n}_\phi = \Omega_X \sin \phi - \Omega_Y \cos \phi.
 \end{aligned} \tag{4.20}$$

2D axial coordinates  $(x_1, x_2) = (Z, R)$  for `iradial = 2`:

$$\begin{aligned}
 x_1 \equiv x &= Z, & \vec{n}_{x_1} \equiv \vec{n}_x &= \vec{n}_Z, \\
 x_2 \equiv y \equiv R &= (X^2 + Y^2)^{1/2}, & \vec{n}_{x_2} \equiv \vec{n}_y \equiv \vec{n}_R &= \vec{n}_X \cos \phi + \vec{n}_Y \sin \phi, \\
 x_3 \equiv \phi &= \arctan(Y/X); & \vec{n}_{x_3} \equiv \vec{n}_z &= \vec{n}_\phi = -\vec{n}_X \sin \phi + \vec{n}_Y \cos \phi;
 \end{aligned} \tag{4.21}$$

$$\begin{aligned}
 \Omega_{x_1} \equiv \Omega_x &= \Omega_Z = \vec{\Omega} \cdot \vec{n}_Z, \\
 \Omega_{x_2} \equiv \Omega_y \equiv \Omega_R &= \vec{\Omega} \cdot \vec{n}_R = \Omega_X \cos \phi + \Omega_Y \sin \phi, \\
 \Omega_{x_3} \equiv \Omega_z &= \vec{\Omega} \cdot \vec{n}_\phi = -\Omega_X \sin \phi + \Omega_Y \cos \phi.
 \end{aligned} \tag{4.22}$$

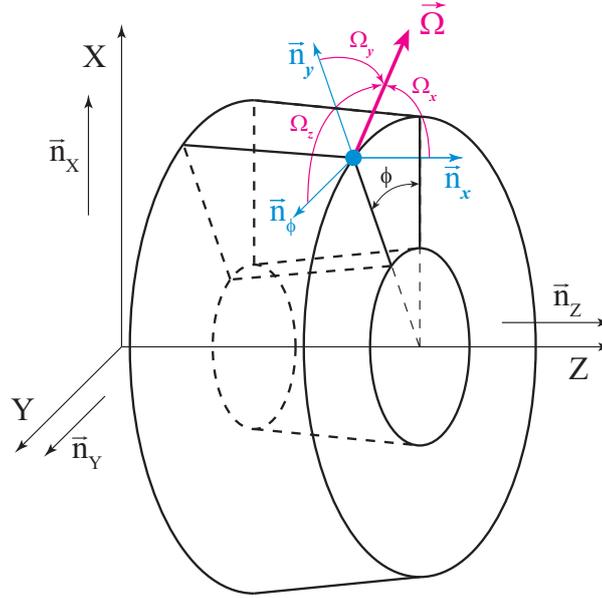


FIG. 4.3: Local decomposition of the photon propagation direction  $\vec{\Omega}$  in the axial  $rz$  geometry with the rotation axis along the mesh direction 1 (`iradial=2`), i.e. with  $x = Z$ .

The notation (4.18) for the 2D Cartesian case, where the global and the local decompositions of  $\vec{\Omega}$  coincide, is straightforward and clear. In any of the 2D axial cases we can choose either the global components  $\Omega_X, \Omega_Y, \Omega_Z$ , or the local components  $\Omega_x, \Omega_y, \Omega_z$  of vector  $\vec{\Omega}$  as primary angular variables. The global components have an advantage for visualization

because they remain constant along the 3D photon path [the characteristic of the transfer equation (2.6)]. However, calculation of all angular integrals for axially symmetric flows is dramatically simplified when we define the vector  $\vec{\Omega}$  in terms of its local components  $\Omega_x, \Omega_y, \Omega_z$ , and below we stick to this latter choice.

One readily verifies that in terms of the local components  $\Omega_x, \Omega_y, \Omega_z$  the radiation field  $I_\nu(\vec{\Omega})$  is symmetric with respect to  $\Omega_z$ ,

$$I_\nu(\Omega_x, \Omega_y, \Omega_z) = I_\nu(\Omega_x, \Omega_y, -\Omega_z), \quad (4.23)$$

for all the three metric cases — provided of course that the boundary conditions for  $I_\nu(\vec{\Omega})$  obey the same symmetry. Hence, we can always choose our computational hemisphere to be at positive values of  $\Omega_z > 0$ . The latter means that we can adopt  $\Omega_x$  and  $\Omega_y$  as independent angular variables, which vary within the intervals

$$-1 \leq \Omega_x \leq +1, \quad -1 \leq \Omega_y \leq +1 \quad (4.24)$$

and cover the 4 octants in the  $\Omega_z > 0$  hemisphere, as shown in Fig. 4.4. The value of  $\Omega_z$  is then given by

$$\Omega_z = +\sqrt{1 - \Omega_x^2 - \Omega_y^2}. \quad (4.25)$$

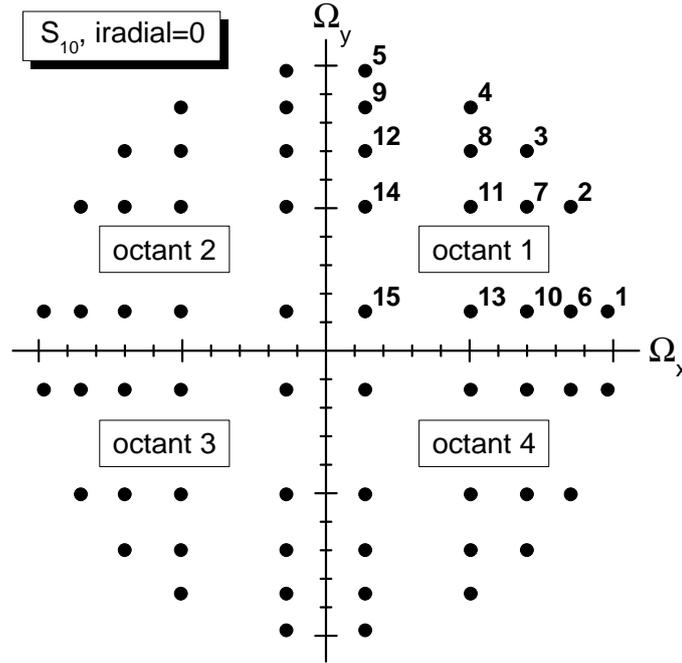


FIG. 4.4: Discrete angular directions (angular quadrature nodes)  $\vec{\Omega}_L = (\Omega_{xL}, \Omega_{yL})$  in the  $S_{10}$  configuration from Ref. [20]. Numbered are only the nodes  $\vec{\Omega}_l = (\Omega_{xl}, \Omega_{yl})$  in the first octant, where the labels indicate the values of index  $l = 1, 2, \dots, 15$  in the Cartesian  $xy$  case with  $\text{iradial} = 0$ .

Because the  $S_n$  nodes are always assumed to obey the mirror symmetry between the octants, we can adopt the following numbering convention for the  $4N_\Omega$  relevant beamlets in the 4 octants of the  $\Omega_z > 0$  hemisphere:

- every beamlet is identified by the values of the following three indices

$$i_{ox} = \pm 1, \quad i_{oy} = \pm 1, \quad l = 1, 2, \dots, N_\Omega, \quad (4.26)$$

where  $i_{ox}$  and  $i_{oy}$  are, respectively, the signs of  $\Omega_{xL}$  and  $\Omega_{yL}$ ; here we distinguish between the global 4-octant index  $L = (i_{ox}, i_{oy}, l)$ , which covers all the 4 relevant octants and runs through  $L = 1, 2, \dots, 4N_\Omega$ , and between the single-octant index  $l$ , which runs through  $l = 1, 2, \dots, N_\Omega$ ;

- by definition,  $\Omega_{xl} > 0$ ,  $\Omega_{yl} > 0$ ; for any fixed value of  $l$ , the four beamlets with  $(i_{ox}, i_{oy}) = (\pm 1, \pm 1)$ , i.e. with

$$(\Omega_{xL}, \Omega_{yL}) = (\pm \Omega_{xl}, \pm \Omega_{yl}) \quad (4.27)$$

comprise the  $l$ -th *quad* of mutually *conjugate* beamlets; the ordering convention for the  $N_\Omega$  beamlets  $\vec{\Omega}_l$  in the first octant is explained in section 4.2.3 below.

As already mentioned, in the simplest 2D Cartesian case both the local and the global components of  $\vec{\Omega}$  remain constant along the photon path, whose projection onto the computational  $(x_1, x_2) = (x, y)$  plane is a straight-line. By contrast, only one local component of  $\vec{\Omega}$  — namely,  $\Omega_x = \Omega_z$  for `iradial` = 2, and  $\Omega_y = \Omega_z$  for `iradial` = 1 — remains constant along the photon path in any of the 2D axial cases because the azimuthal angle  $\phi$  changes along the light trajectory. As a result, the photon trajectories (characteristics of the transfer equation) in the  $(x_1, x_2) = (x, y)$  computational plane are no longer straight lines but hyperbolae.

## 2. Angular integrals and particular sets of the $S_n$ -nodes

Let  $f(\vec{\Omega}) = f(\Omega_x, \Omega_y, \Omega_z)$  be an arbitrary scalar function of a unit vector  $\vec{\Omega}$ , which is even with respect to  $\Omega_z$ , i.e.  $f(\Omega_x, \Omega_y, \Omega_z) = f(\Omega_x, \Omega_y, -\Omega_z) = f(\Omega_x, \Omega_y)$ . Then the integral of  $f(\vec{\Omega})$  over the entire  $4\pi$  solid angle in the  $S_n$  formalism is given by a sum

$$\int_{4\pi} f(\vec{\Omega}) d\vec{\Omega} = 2 \sum_{L=1}^{4N_\Omega} w_L f(\Omega_{xL}, \Omega_{yL}) = 2 \sum_{l=1}^{N_\Omega} w_l \sum_{i_{ox}=\pm 1} \sum_{i_{oy}=\pm 1} f(i_{ox}\Omega_{xl}, i_{oy}\Omega_{yl}), \quad (4.28)$$

where  $w_l$  are the node weights normalized to yield  $4\pi$  for  $f(\vec{\Omega}) = 1$ .

In the RALEF code we use two different sets of the  $S_n$  nodes: (i) the  $S_n$  nodes from Ref. [20], tabulated with 16 digits for  $k_\Omega = 1, 2, \dots, 6$ , and (ii) a simpler and less symmetric  $ES_n$  set of ordinates proposed by Carlson [21] and generated by the subroutine `SNOME_IPM` from the module `SN_NODES` (file ‘`f08_util.f`’) as described in Appendix A. Because the  $S_n$  nodes from Ref. [20] are invariant with respect to interchange of the  $x, y, z$  axes, it does not matter which of the three axes in Ref. [20] we choose to be our global  $Z$ -axis. However, this equivalence between the coordinate axes is destroyed in the  $ES_n$  case: here we have only one axis (the polar axis), with respect to which the nodes within any given tier all have equal  $Z$ -coordinates (see Fig. 4.5 below). Naturally, we identify this polar axis with our global  $Z$ -axis, i.e. with the local  $z$ -axis for `iradial` = 0, with the local  $y$ -axis for `iradial` = 1, and with the local  $x$ -axis for `iradial` = 2. The latter means that, for any possible value of

iradial, the projection  $\Omega_Z$  always remains constant along the characteristics of the transfer equation. Numerical tests indicate that for  $\text{iradial} = 0$  this choice reduces by about a factor 1.5 the errors due to the ray effect (see section 10.3 below). For  $\text{iradial} = 1, 2$  this choice allows construction of a more economical numerical algorithm.

By default, the  $S_n$  nodes from Ref. [20] are used for  $k_\Omega \leq 3$ , and the  $ES_n$  nodes for  $k_\Omega > 3$ . For  $k_\Omega = 4, 5, 6$ , for which both options are available, the preference is given to a simpler  $ES_n$  quadrature because it tends to produce smaller spurious intensity fluctuations due to the ray effect (see section 10.3 below). Note that in the  $ES_n$  case all the weights  $w_l$  are equal to  $\pi/2N_\Omega$ .

As an accuracy test for the  $S_n$  and  $ES_n$  quadratures, we made use of the following three analytical integrals

$$\mathcal{I}_0 = \frac{1}{4\pi} \int_{4\pi} (\vec{a} \cdot \vec{\Omega}) d\vec{\Omega} = 0, \quad (4.29)$$

$$\mathcal{I}_1 = \frac{1}{4\pi} \int_{4\pi} (\vec{a} \cdot \vec{\Omega})(\vec{b} \cdot \vec{\Omega}) d\vec{\Omega} = \frac{1}{3} (\vec{a} \cdot \vec{b}), \quad (4.30)$$

$$\mathcal{I}_2 = \frac{1}{4\pi} \int_{4\pi} (\vec{a} \cdot \vec{\Omega})^2 (\vec{b} \cdot \vec{\Omega})^2 d\vec{\Omega} = \frac{1}{5} (\vec{a} \cdot \vec{b})^2 + \frac{1}{15} (\vec{a} \times \vec{b})^2, \quad (4.31)$$

where  $\vec{a}$  and  $\vec{b}$  are arbitrary fixed vectors. Table 4.2 gives the values of  $\mathcal{I}_0$ , and relative errors for  $\mathcal{I}_1$ ,  $\mathcal{I}_{12}$ , and  $\mathcal{I}_2$  as calculated with the ordinates and weights from Ref. [20] for  $k_\Omega = 1, 2, \dots, 6$  and the vectors

$$\vec{a} = \{ \sqrt{2}; \sqrt{3}; -1 \}, \quad (4.32)$$

$$\vec{b} = \{ 3\sqrt{2}; -2/\sqrt{3}; 1 \}, \quad (4.33)$$

for which

$$\mathcal{I}_1 = 1, \quad \mathcal{I}_2 = 9\frac{1}{3}. \quad (4.34)$$

Table 4.3 presents similar errors for the  $ES_n$  ordinates and weights from Ref. [21]. Note that the  $ES_n$  ordinates, which have lower rotational symmetry, typically provide lower accuracy for the same values of  $k_\Omega$ .

TABLE 4.2: Numerical errors of quadrature (4.28) in the  $S_n$  method with ordinates  $\Omega_{xl}, \Omega_{yl}$  and weights  $w_l$  from Ref. [20].

	$S_2$	$S_4$	$S_6$	$S_8$	$S_{10}$	$S_{12}$
$\mathcal{I}_0$	$-7.1 \times 10^{-17}$	$1.8 \times 10^{-17}$	$-4.4 \times 10^{-18}$	0.0	$-1.8 \times 10^{-17}$	0.0
$\delta\mathcal{I}_1/\mathcal{I}_1$	0.0	$6.7 \times 10^{-16}$	$2.2 \times 10^{-16}$	$4.4 \times 10^{-16}$	$6.7 \times 10^{-16}$	$-3.3 \times 10^{-16}$
$\delta\mathcal{I}_2/\mathcal{I}_2$	-0.31	$6.7 \times 10^{-16}$	$2.2 \times 10^{-16}$	$2.6 \times 10^{-3}$	$6.9 \times 10^{-4}$	$5.4 \times 10^{-4}$

For adequate description of the radiation transport across interfaces between strongly opaque and transparent regions, it is important to have high-accuracy quadrature formulae

TABLE 4.3: Numerical errors of quadrature (4.28) in the  $ES_n$  method with the ordinates  $\Omega_{xl}, \Omega_{yl}$  and weights  $w_l$  from Ref. [21].

	$S_4$	$S_6$	$S_{12}$	$S_{24}$	$S_{48}$
$\mathcal{I}_0$	$-8.8 \times 10^{-18}$	$-1.3 \times 10^{-17}$	$-3.3 \times 10^{-18}$	$-4.7 \times 10^{-18}$	$-2.8 \times 10^{-19}$
$\delta\mathcal{I}_1/\mathcal{I}_1$	$6.7 \times 10^{-16}$	$8.9 \times 10^{-16}$	$1.3 \times 10^{-15}$	$4.4 \times 10^{-16}$	$-2.6 \times 10^{-15}$
$\delta\mathcal{I}_2/\mathcal{I}_2$	$3.5 \times 10^{-2}$	$1.5 \times 10^{-2}$	$3.6 \times 10^{-3}$	$8.8 \times 10^{-4}$	$2.2 \times 10^{-4}$

for the first-order angular half-moments, like

$$\mathcal{I}_x = \frac{1}{\pi} \int_{2\pi} \Omega_x d\vec{\Omega} = 1 = \frac{4}{\pi} \sum_{l=1}^{N_\Omega} w_l \Omega_{xl}, \tag{4.35}$$

$$\mathcal{I}_y = \frac{1}{\pi} \int_{2\pi} \Omega_y d\vec{\Omega} = 1 = \frac{4}{\pi} \sum_{l=1}^{N_\Omega} w_l \Omega_{yl}. \tag{4.36}$$

A more general example of a half-moment would be given by an integral

$$\mathcal{I}_{1/2} = \frac{1}{4\pi} \int_{4\pi} (\vec{a} \cdot \vec{\Omega} + |\vec{a} \cdot \vec{\Omega}|) d\vec{\Omega} = \frac{1}{2} \sqrt{\vec{a} \cdot \vec{a}}, \tag{4.37}$$

where  $\vec{a}$  is an arbitrary vector. Relative errors in the values of these three integrals, calculated with the ordinates and weights from Ref. [20], and for  $\vec{a}$  defined in Eq. (4.32), are given in Table 4.4; Table 4.5 presents these errors for the  $ES_n$  ordinates from Ref. [21].

TABLE 4.4: Numerical errors of quadrature (4.28) in the  $S_n$  method with the ordinates and weights from Ref. [20] for the first-order angular half-moments.

$S_n$	$S_2$	$S_4$	$S_6$	$S_8$	$S_{10}$	$S_{12}$
$\delta\mathcal{I}_x/\mathcal{I}_x$	0.155	$4.6 \times 10^{-2}$	$-1.1 \times 10^{-16}$	0.0	0.0	$-5.6 \times 10^{-16}$
$\delta\mathcal{I}_y/\mathcal{I}_y$	0.155	$4.6 \times 10^{-2}$	$-3.3 \times 10^{-16}$	0.0	$-1.1 \times 10^{-16}$	$-4.4 \times 10^{-16}$
$\delta\mathcal{I}_{1/2}/\mathcal{I}_{1/2}$	$-2.3 \times 10^{-2}$	$1.2 \times 10^{-2}$	$-1.85 \times 10^{-2}$	$-3.1 \times 10^{-3}$	$-2.6 \times 10^{-4}$	$-2.9 \times 10^{-4}$

TABLE 4.5: Numerical errors of quadrature (4.28) in the  $ES_n$  method with the ordinates and weights from Ref. [21] for the first-order angular half-moments.

$S_n$	$S_4$	$S_6$	$S_{12}$	$S_{24}$	$S_{48}$
$\delta\mathcal{I}_x/\mathcal{I}_x$	$3.2 \times 10^{-2}$	$1.8 \times 10^{-2}$	$5.5 \times 10^{-3}$	$1.5 \times 10^{-3}$	$4.1 \times 10^{-4}$
$\delta\mathcal{I}_y/\mathcal{I}_y$	$3.3 \times 10^{-2}$	$1.8 \times 10^{-2}$	$5.5 \times 10^{-3}$	$1.5 \times 10^{-3}$	$4.1 \times 10^{-4}$
$\delta\mathcal{I}_{1/2}/\mathcal{I}_{1/2}$	$1.8 \times 10^{-2}$	$-4.9 \times 10^{-4}$	$-5.3 \times 10^{-5}$	$1.3 \times 10^{-4}$	$-1.1 \times 10^{-5}$

3. *Extended numbering system for the  $S_n$  nodes*

a.  *$S_n$  node indexing in the  $rz$  geometry.* Because in the axial  $rz$  geometry the mesh directions 1 and 2 are not equivalent, we need a more detailed numbering system for the  $S_n$  nodes  $\vec{\Omega}_L$  than a simple octant-oriented decomposition

$$L = \{i_{ox}, i_{oy}, l\} \quad (4.38)$$

of the global angular index  $L$ . Keeping in mind that the  $S_n$  ordinates used in the RALEF code are arranged in tiers around the global  $Z$ -axis, we introduce an alternative decomposition

$$L = \{i_{oZ}, i_{oR}, m_Z, m_R\}, \quad (4.39)$$

where

$$(i_{oZ}, i_{oR}) = \begin{cases} (i_{oy}, i_{ox}), & \text{iradial} = 1, \\ (i_{ox}, i_{oy}), & \text{iradial} = 2. \end{cases} \quad (4.40)$$

In this decomposition the single intra-octant index  $l$  is split into two components  $l = \{m_Z, m_R\}$  (see Fig. 4.5), where

$$m_Z = 1, 2, \dots, k_\Omega \quad (4.41)$$

is the sequential tier number along the  $Z$ -axis (in the first octant), and

$$m_R = 1, 2, \dots, N_{Z\Omega} \quad (4.42)$$

is the sequential number of projection onto the  $R$ -axis within a given  $Z$ -tier  $m_Z$ ; note that the upper bound

$$N_{Z\Omega} = k_\Omega + 1 - m_Z \quad (4.43)$$

of  $m_R$  variation depends on  $m_Z$ . We assume that the projection  $\Omega_Z$  increases with the increasing  $m_Z$ , and that  $\Omega_R$  increases with the increasing  $m_R$ ; the latter means that the lowest tier  $m_Z = 1$  is the closest to the equator. The single index  $l$  can then be expressed in terms of  $m_Z$  and  $m_R$  as

$$l = m_R + \frac{1}{2}(m_Z - 1)(2k_\Omega + 2 - m_Z), \quad (4.44)$$

which means that the sequential order of the intra-octant nodes  $\vec{\Omega}_l$  is as follows

$$\begin{aligned} l = 1 & \Leftrightarrow (m_Z, m_R) = (1, 1), \\ l = 2 & \Leftrightarrow (m_Z, m_R) = (1, 2), \\ & \dots \\ l = N_\Omega & \Leftrightarrow (m_Z, m_R) = (k_\Omega, 1). \end{aligned} \quad (4.45)$$

In the Cartesian  $xy$  case the numbering of  $\vec{\Omega}_l$  follows the same principle but with the  $Z$ -axis perpendicular to the computational  $xy$  plane. Here every  $Z$ -tier is represented by a circle in the  $xy$ -plane with a fixed value of  $\Omega_x^2 + \Omega_y^2 = 1 - \Omega_Z^2$ . The nodes  $\vec{\Omega}_l$  are numbered tier after tier, starting from the closest to the equator with the largest value of  $\Omega_x^2 + \Omega_y^2$ , as shown in Fig. 4.4. Within the same tier,  $l$  increases with the increasing value of the  $y$ -projection  $\Omega_{yl}$  — i.e. we use the same ordering (4.45) under the assumption that the  $R$ -axis is identified with the  $y$ -axis.

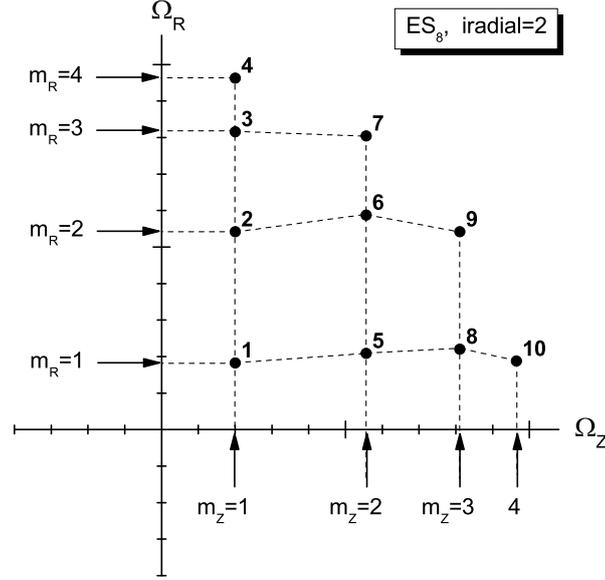


FIG. 4.5: Numbering convention with the  $m_Z, m_R$  indices on the example of the  $ES_8$  quadrature from [21] for the axial  $rz$  case with  $\text{iradial} = 2$ . The numerical labels near the  $ES_8$  nodes indicate the values of index  $l = 1, 2, \dots, 10$ .

*b. Independent blocks of  $S_n$  nodes.* For parallel programming it is important to know whether the radiation intensity  $I_L(\vec{x}) = I(\vec{x}, \vec{\Omega}_L)$  for a given direction  $\vec{\Omega}_L$  can be computed independently of other directions, i.e. without knowing  $I_{L'}(\vec{x})$  at  $L' \neq L$ . In the  $rz$  case and/or when reflective boundaries are present, only certain groups — called *independent  $\Omega$ -blocks* — of directions  $\vec{\Omega}_L$  can be computed independently from one another.

In the Cartesian geometry all the different  $l$ -values are mutually independent, and the independent  $\Omega$ -blocks can be associated with the  $l$ -quads, each containing the 4 beamlets

$$(i_{ox}, i_{oy}) = (\pm 1, \pm 1) \quad (4.46)$$

with the same value of index  $l$ . Such a decomposition into mutually independent  $\Omega$ -blocks remains valid for all allowed combinations of reflective boundaries (for details see section 4.5.1) and automatically satisfies a special requirement of our algorithm that, for any angular direction  $\vec{\Omega}_L$ , the  $\Omega$ -block must also contain the opposite direction  $-\vec{\Omega}_L$ . When there are no reflective boundaries, the number of beamlets in every independent  $\Omega$ -block can be further reduced by half, but this hardly brings any computational advantage when the hydrodynamics part of the algorithm is not parallelized. Hence, in the Cartesian  $xy$  geometry representation (4.38) is the most natural and convenient for parallel programming.

In the axial  $rz$  geometry, where the  $Z$ -component  $\Omega_Z$  of  $\vec{\Omega}$  remains constant but the  $R$ -component  $\Omega_R$  monotonically increases along the photon path, mutually independent can only be the groups of directions with different values of index  $m_Z$ . After we add their opposite counterparts, we find that in the  $rz$  case an independent  $\Omega$ -block consists of beamlets with a fixed value of  $m_Z$  combined with the four values

$$(i_{oZ}, i_{oR}) = (\pm 1, \pm 1) \quad (4.47)$$

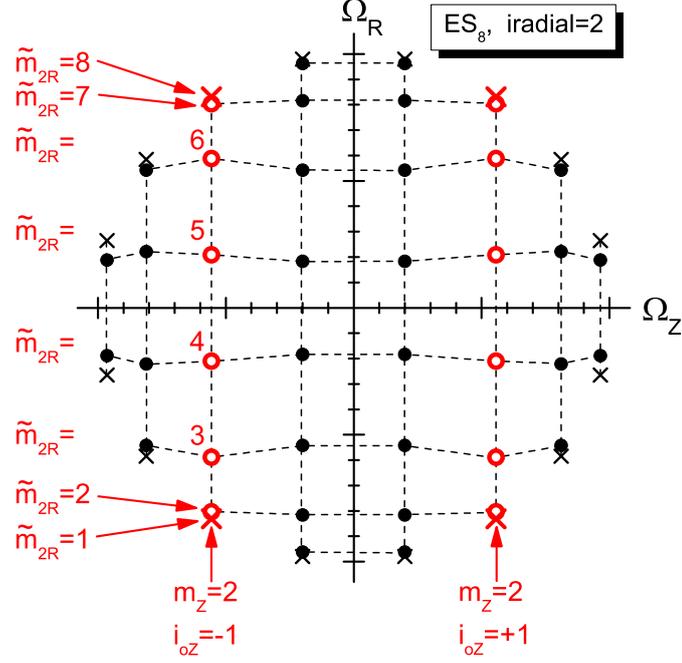


FIG. 4.6: The ordinates of the  $m_Z = 2$  independent  $\Omega$ -block are shown with red open circles and crosses for the case of the  $ES_8$  quadrature with  $\text{iradial} = 2$ . Within this  $\Omega$ -block, numbering of nodes with the index  $\tilde{m}_{2R} = 1, 2, \dots, 8$  is indicated.

and the full set of values  $m_R = 1, 2, \dots, N_{Z\Omega}$ . In mathematical terms, an independent  $\Omega$ -block is a Cartesian product  $i_{oZ} \times i_{oR} \times m_R$  in the logical space of the 4 indices  $(i_{oZ}, i_{oR}, m_Z, m_R)$  in representation (4.39). As an illustrative example, in Fig. 4.6 the  $m_Z = 2$  independent  $\Omega$ -block is highlighted in red.

To facilitate high-accuracy interpolation along the  $\Omega_R$  variable over the entire range  $-\sqrt{1 - \Omega_Z^2} \leq \Omega_R \leq \sqrt{1 - \Omega_Z^2}$  of its variation along a photon path with a given  $\Omega_Z$ , it is convenient to augment every subset  $(i_{oZ}, m_Z)$  of the  $S_n$  nodes with two additional “fringe” (axial) nodes  $(\Omega_Z, \Omega_R) = (\Omega_Z, \Omega_{\pm})$ , where

$$\Omega_{\pm} = \pm\sqrt{1 - \Omega_Z^2}. \quad (4.48)$$

These additional nodes may be called “axial” because they correspond to light rays which pass through the axis of symmetry  $R = 0$ ; in Fig. 4.6 they are marked with crosses.

To include the additional “fringe” nodes into our numbering system, we extend the variation range of index  $m_R$  to

$$\tilde{m}_R = 1, 2, \dots, \tilde{N}_{Z\Omega}, \quad (4.49)$$

where

$$\tilde{N}_{Z\Omega} = N_{Z\Omega} + 1 = k_{\Omega} + 2 - m_Z. \quad (4.50)$$

Also, instead of the pair  $(i_{oR}, \tilde{m}_R)$ , it is convenient to introduce a single index

$$\tilde{m}_{2R} = 1, 2, \dots, 2\tilde{N}_{Z\Omega} = \begin{cases} \tilde{N}_{Z\Omega} + 1 - \tilde{m}_R, & i_{oR} = -1, \\ \tilde{N}_{Z\Omega} + \tilde{m}_R, & i_{oR} = +1. \end{cases} \quad (4.51)$$

covering the same range as the pair  $(i_{oR}, \tilde{m}_R)$ . Now, as  $\tilde{m}_{2R}$  monotonously increases from 1 to  $2\tilde{N}_{Z\Omega}$ , the  $R$ -projection  $\Omega_R$  monotonously grows from  $\Omega_- = -\sqrt{1 - \Omega_Z^2}$  to  $\Omega_+ = +\sqrt{1 - \Omega_Z^2}$ ; see Fig. 4.6.

#### 4. Sequential order of processing the $S_n$ beamlets

*a. Processing order within a single  $\Omega$ -block.* Within a single  $\Omega$ -block, where individual  $S_n$  beamlets are coupled to each other, they should be processed in a certain definite order. In the Cartesian  $xy$  case (i.e. for `iradial` = 1) this order is dictated by the presence of global reflective boundaries, i.e. is equivalent to the order in which the four combinations  $(i_{ox}, i_{oy}) = (\pm 1, \pm 1)$  are ordered. In the RALEF code this order is set by the values of the two arrays `iouseqx(1:4)` and `iouseqy(1:4)`, loaded in the subroutine `RADSET`, file ‘`f09_rad.f`’; in other words, the beamlet processing order is defined by

$$(i_{ox}, i_{oy}) = (\text{iouseqx}(k), \text{iouseqy}(k)), \quad k = 1, 2, 3, 4. \quad (4.52)$$

Detailed explanations about how the values `iouseqx(1:4)`, `iouseqy(1:4)` are set for different configurations of the reflective boundaries are given in section 4.5.2 below.

In the  $rz$  geometry the processing order is in the first place stipulated by the fact that the projection  $\Omega_R$  monotonically increases along the photon path. As a result, the processing order is prescribed by the index  $\tilde{m}_{2R}$ . Taking into account a possible reflective boundary at  $Z = 0$ , we always begin with the  $i_{oZ} = -1$  sequence  $\tilde{m}_{2R} = 1, 2, 3, \dots, 2\tilde{N}_{Z\Omega}$ , and then do the analogous sequence for  $i_{oZ} = +1$  (see Fig. 4.6).

*b. Global processing order.* A priori, the order of processing the mutually independent  $\Omega$ -blocks is arbitrary. Hence, one could simply adopt the loop over individual  $\Omega$ -blocks as the principal external loop whose iterations could all be done in parallel. However, in the axial  $rz$  case, where independent  $\Omega$ -blocks contain different numbers of nodes, it is advantages to process in parallel not individual  $\Omega$ -blocks but larger units that we name  $\Omega$ -bundles: one  $\Omega$ -bundle can contain either one or two independent  $\Omega$ -blocks. In the Cartesian case every  $\Omega$ -bundle contains only one  $\Omega$ -block. Combining  $\Omega$ -blocks into pairs allows a more uniform distribution of work among different OpenMP threads and MPI tasks (for more details see section 8). In this case the scalar parameters  $\tilde{m}_Z$  and  $\tilde{N}_{Z\Omega}$  become two-element arrays  $\tilde{m}_Z(k_{ot})$  and  $\tilde{N}_{Z\Omega}(k_{ot})$ , where  $k_{ot} = 1, 2$  is the index for the current  $\Omega$ -block inside a given  $\Omega$ -bundle  $i_{ob}$ . The distribution of individual  $\Omega$ -blocks (identified uniquely by the corresponding  $m_Z$  value) is prescribed by the array `mzObndl(1 : 2, 1 : N $\Omega$ )` loaded in the subroutine `RADSET`, file ‘`f09_rad.f`’:

$$\text{mzObndl}(k_{ot}, i_{ob}) = m_Z \quad (4.53)$$

is the  $m_Z$  value of the  $\Omega$ -block number  $k_{ot}$  in the  $\Omega$ -bundle number  $i_{ob}$ , where

$$1 \leq i_{ob} \leq N_{ob} = \begin{cases} N_{\Omega} = \frac{1}{2}k_{\Omega}(k_{\Omega} + 1), & \text{iradial} = 0, \\ \frac{1}{2}(k_{\Omega} + 1) + 1, & \text{iradial} = 1, 2. \end{cases} \quad (4.54)$$

The  $N_{\Omega}$   $\Omega$ -blocks are distributed among  $N_{ob}$   $\Omega$ -bundles so that the the first  $\Omega$ -bundle contains one  $\Omega$ -block with  $m_Z = k_{\Omega}$ , the second contains one  $\Omega$ -block with  $m_Z = k_{\Omega} - 1$ , the third contains two  $\Omega$ -blocks with  $m_Z = 1$  and  $m_Z = k_{\Omega} - 2$  (or one  $\Omega$ -block with  $m_Z = k_{\Omega} - 2$ ), the fourth contains two  $\Omega$ -blocks with  $m_Z = 2$  (or  $m_Z = 1$ ) and  $m_Z = k_{\Omega} - 3$ , etc.

For the end-to-end numbering of individual directions  $\Omega_L$  in a given  $\Omega$ -bundle, a single beamlet index

$$\begin{aligned} i_o &= 4(k_{ot} - 1)\tilde{N}_{Z\Omega}(1) + (i_{oZ} + 1)\tilde{N}_{Z\Omega}(k_{ot}) + \tilde{m}_{2R} = \\ &= 1, 2, 3, \dots, 4[\tilde{N}_{Z\Omega}(1) + \tilde{N}_{Z\Omega}(2)] \end{aligned} \quad (4.55)$$

is introduced. For a given  $\Omega$ -bundle number  $i_{ob}$  this index is easily decomposed into its constituent subindices  $k_{ot}$ ,  $i_{oZ}$  and  $\tilde{m}_{2R}$ , once the values of `mz0bndl(kot, iob)` are available (as programmed in the subroutine `DCPOSIO`, file 'f09\_rad.f'). If the beamlet index (4.55) corresponds to the direction  $\vec{\Omega}_L$ , then the opposite direction  $-\vec{\Omega}_L$  has the beamlet index

$$i_{mo} = 4(k_{ot} - 1)\tilde{N}_{Z\Omega}(1) + (3 - i_{oZ})\tilde{N}_{Z\Omega}(k_{ot}) + 1 - \tilde{m}_{2R}. \quad (4.56)$$

Some additional useful formulae are

$$\tilde{m}_R = \begin{cases} \tilde{N}_{Z\Omega} + 1 - \tilde{m}_{2R}, & \tilde{m}_{2R} \leq \tilde{N}_{Z\Omega}, \\ \tilde{m}_{2R} - \tilde{N}_{Z\Omega}, & \tilde{m}_{2R} > \tilde{N}_{Z\Omega}, \end{cases} \quad (4.57)$$

$$i_{oR} = \begin{cases} -1, & \tilde{m}_{2R} \leq \tilde{N}_{Z\Omega}, \\ +1, & \tilde{m}_{2R} > \tilde{N}_{Z\Omega}, \end{cases} \quad (4.58)$$

$$l = \frac{1}{2} \left( k_{\Omega} + \tilde{N}_{Z\Omega} \right) (m_Z - 1) + \min \left( \tilde{m}_R, \tilde{N}_{Z\Omega} - 1 \right). \quad (4.59)$$

The above described scheme of processing order for the  $S_n$  beamlets can be illustrated with the following FORTRAN do-loops.

In the Cartesian  $xy$  geometry (`iradial = 0`) the loops over  $\Omega$ -blocks and over individual directions in a given  $\Omega$ -block can be programmed as

```
do l=1,nomegs      ! loop over independent Omega-blocks
  do io=1,4
    ioctx=iouseqx(io)
    iocty=iouseqy(io)
    ...
  enddo
enddo
```

For the axial  $rz$  geometry (`iradial = 1, 2`) such loops can be written as

```
do mZ=1,kradSn    ! loop over independent Omega-blocks
  do ioZ=-1,1,2
    do m2R=1,2*(kradSn+2-mZ)
      ...
    enddo
  enddo
enddo
```

Note that the actual do-loops over the  $S_n$  beamlets in the subroutine `RADDRV` (file 'f09\_rad.f'), based on the  $\Omega$ -bundle structure, have the form

```
do i0bndl=1,n0bndls  ! loop over Omega-bundles
  do io=1,ni0bndl
    call DCPOSIO(io,.true.)
    ...
  enddo
```

where  $n0bndls = N_{ob}$ , and

$$ni0bndl = 4[\tilde{N}_{Z\Omega}(1) + \tilde{N}_{Z\Omega}(2)]. \quad (4.60)$$

---

*Correspondence with the code variables:*

$k_{\Omega} =$	<code>kradSn</code>	principal user-defined parameter ( $= 1, 2, 3, \dots$ ), which defines the order of the $S_n = S_{2k_{\Omega}}$ angular quadrature;
$N_{\Omega} =$	<code>nomegs</code>	number of beamlets per octant in the $S_n$ quadrature; $N_{\Omega} = k_{\Omega}(k_{\Omega} + 1)/2$ ;
$l =$	<code>l=lomega</code>	index for numbering the $S_n$ nodes in the 1-st octant; $l = 1, 2, \dots, N_{\Omega}$ ;
$i_{ox} =$	<code>ioctx</code>	octant index along the $x_1$ -axis;
$i_{oy} =$	<code>iocty</code>	octant index along the $x_2$ -axis;
$i_{oZ} =$	<code>ioctZ</code>	octant index along the global $Z$ -axis;
$i_{oR} =$	<code>ioctR</code>	octant index along the global cylindrical radius $R$ ;
$m_Z =$	<code>mZ=mZomega</code>	index for numbering the $Z$ -projections of the $S_n$ nodes in the 1-st octant; $m_Z = 1, 2, \dots, k_{\Omega}$ ;
$k_{ot} =$	<code>kot=kotierZ</code>	index for numbering $\Omega$ -blocks in a given $\Omega$ -bundle; $k_{ot} = 1, 2$ ;
$\tilde{N}_{Z\Omega} =$	<code>nzomegs(1:2)</code>	total number of $S_n$ nodes in a given $Z$ -tier (i.e. at a given $m_Z$ value) per octant;
$\tilde{m}_R =$	<code>mR=mRomega</code>	index for numbering the $R$ -projections of the $S_n$ nodes in the 1-st octant; $\tilde{m}_R = 1, 2, \dots, \tilde{N}_{Z\Omega}$ ;
$\tilde{m}_{2R} =$	<code>m2R=m2Rcount</code>	index for numbering all the $R$ -projections of the $S_n$ nodes for given $m_Z$ and $i_{oZ}$ ; $\tilde{m}_{2R} = 1, 2, \dots, 2\tilde{N}_{Z\Omega}$ ;
$i_o =$	<code>io=iobmlt</code>	end-to-end index for numbering individual directions $\Omega_L$ in a given $\Omega$ -bundle; $i_o = 1, 2, \dots, 4[\tilde{N}_{Z\Omega}(1) + \tilde{N}_{Z\Omega}(2)]$ ;
$i_{mo} =$	<code>imo=imobmlt</code>	end-to-end index for the opposite to $i_o$ beamlet;
$i_{ob} =$	<code>i0bndl</code>	index for the current $\Omega$ -bundle;
$N_{ob} =$	<code>n0bndls</code>	total number of $\Omega$ -bundles in a given $S_n$ quadrature.

---

### 3. Computation of the radiation field

#### 1. The octagon stencil for integration along a short characteristic

On a structured quadrilateral grid every mesh vertex  $i$  is a corner of four adjoining cells, which make up a generally non-regular octagon whose perimeter surrounds the central vertex  $i$ ; see Fig. 4.7. This octagon represents the principal stencil (the *octagon stencil*) for numerical integration of the transfer equation (4.7) along the short characteristic  $\vec{s} = s\vec{\Omega}_L$  passing through vertex  $i$ : as a result, we obtain the value  $I_{[k],L,i}$  of the radiation intensity at this vertex. The corresponding integration formula needs the discrete values  $B_i$  of the source function  $B$ , and  $B'_i$  of its  $\tau$ -derivative  $B'$  (see Eq. (4.65)) to be known at all physical vertices  $i$  together with the down-stream nodal values of the radiation intensity  $I_{[k],L,i}$ . In our algorithm the characteristic integration is always done in 3D space, where all the characteristics of Eq. (4.7) are straight lines. It might be helpful to remember that in three dimensions the 2D octagon becomes an octagonal prism in the  $xy$  case, and an octagonal toroid in the  $rz$  case (similar to the trapezoidal toroid shown in Fig. 4.3).

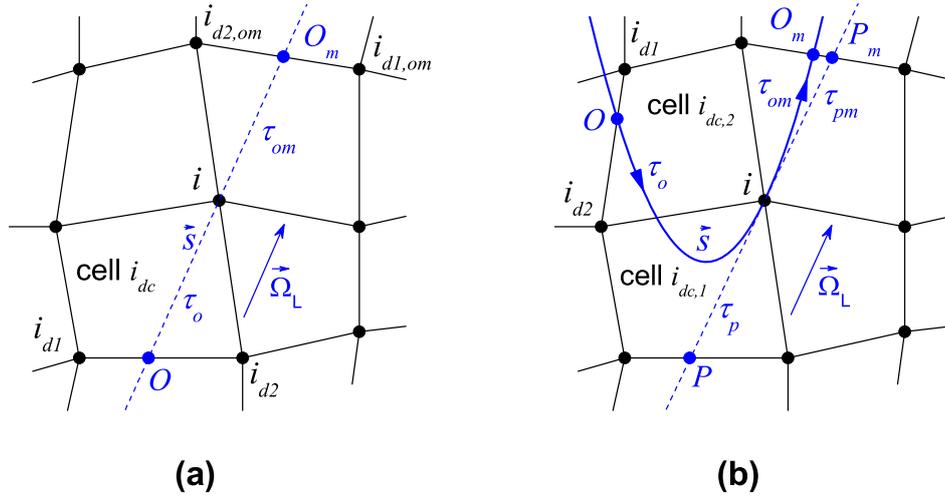


FIG. 4.7: Short characteristics passing through the r-donor cells of vertex  $i$  within its octagon stencil (a) in the Cartesian  $xy$  case, and (b) in the axi-symmetric  $rz$  case.  $\vec{\Omega}_L$  is the photon propagation direction at the central node  $i$ . In case (b) the 3D characteristic  $O - i - O_m$  enters the octagon stencil at the 3D donor point  $O$ , passes through two r-donor cells  $i_{dc,1}$  and  $i_{dc,2}$ , and exits the octagon stencil at point  $O_m$  which, in its turn, is the 3D donor point for the opposite direction  $-\vec{\Omega}_L$ . Cell  $i_{dc,1}$  is the primary r-donor cell, which contains the 2D primary donor point  $P$ . By definition the segment  $P - i - P_m$  is straight in the 2D computational plane and tangent to  $O - i - O_m$  at the central node  $i$ .

Below we assume that every physical mesh vertex  $i$  lies inside its octagon stencil: a mesh where this is not the case is considered to be too strongly distorted to allow numerical solution of the transfer equation. A corresponding check is performed in the **subroutine** MESHCH (file ‘f03\_bound.f’) at every hydrocycle. Under this condition there is always a point  $O$ , where a given characteristic  $\vec{s}$  enters the octagon stencil, and the point  $O_m$  where it exits the octagon stencil. Evidently, the entry and the exit points  $O$  and  $O_m$  interchange their meaning when an opposite beamlet  $-\vec{\Omega}_L$  is considered.

When traced back from the central vertex  $i$  down to the entrance point  $O$ , the characteristic passes through either one or several *r-donor cells* belonging to the octagon stencil. In the  $xy$  case, where the projection of the 3D characteristic onto the 2D computational plane is also straight, there can only be one r-donor cell (see Fig. 4.7a). In the  $rz$  case, where this projection is a hyperbola, the characteristic segment  $O - i$  may cross up to five r-donor cells within the octagon stencil of vertex  $i$  (see Fig. 4.7b and Fig. 4.8).

By definition, the entrance point  $O$  lies on the perimeter of the octagon stencil, on one of the eight *peripheral edges* of the 4 stencil cells. The end vertices  $i_{d1}$  and  $i_{d2}$  of the cell edge, containing the entrance point  $O$ , are the two principal r-donor vertices of the central node  $i$ . They are always ordered such that when moving from  $i_{d1}$  to  $i_{d2}$  the octagon stencil stays on the left. The complete data set (the *r-donor data set*, or, simply, the *rd-set*) pertaining to the short characteristic  $O - i$  is calculated and stored in memory as a result of solution of the *ray-octagon problem* (see section 4.6.7 below) in the **subroutine** RADON12, file ‘f09\_rad.f’.

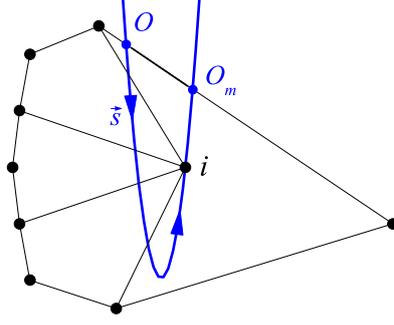


FIG. 4.8: Example of a short characteristic in the  $rz$  geometry passing through 5 consecutive r-donor cells in a deformed octagon stencil.

## 2. The principal finite-difference integration formula

The radiation intensity  $I_i$  (here and below we omit the spectral,  $[k]$ , and the angular,  $L$ , indices unless they are absolutely needed) at a physical vertex  $i$  is calculated by integrating Eq. (4.7) along its straight 3D characteristic

$$X = X_i + s\Omega_{XL}, \quad Y = Y_i + s\Omega_{YL}, \quad Z = Z_i + s\Omega_{ZL} \quad (4.61)$$

passing through this vertex; here  $(X_i, Y_i, Z_i)$  are the global 3D coordinates of vertex  $i$ ,  $s$  is the distance along the characteristic  $\vec{s}$ , and  $\vec{\Omega}_L = (\Omega_{XL}, \Omega_{YL}, \Omega_{ZL})$  is the 3D photon propagation direction. Once we introduce the optical distance  $\tau$  along the characteristic, defined by

$$d\tau = k_{i_d} ds, \quad \tau(0) = 0, \quad (4.62)$$

where  $k_{i_d}$  is the group-averaged absorption coefficient in the donor cell  $i_d$ , we can write the transfer equation (4.7) in the form

$$\frac{dI}{d\tau} = B - I, \quad (4.63)$$

or

$$\frac{dF}{d\tau} = -F - B', \quad (4.64)$$

where

$$F = I - B, \quad B' = \frac{dB}{d\tau}. \quad (4.65)$$

Note that, while the source function  $B = B(\vec{x})$  is a function of position  $\vec{x}$  only, its  $\tau$ -derivative  $B' = B'(\vec{x}, \vec{\Omega}_L)$  is a function of position  $\vec{x}$  and the photon propagation direction  $\vec{\Omega}_L$ .

Since  $B(\tau)$  is a known function, Eqs. (4.63) and (4.64) are readily integrated to yield

$$I(\tau) = \left[ I_i + \int_0^\tau e^t B(t) dt \right] e^{-\tau}, \quad (4.66)$$

$$F(\tau) = \left[ F_i - \int_0^\tau e^t B'(t) dt \right] e^{-\tau}. \quad (4.67)$$

Applying the above solution to the downstream segment  $[-\tau_o \leq \tau \leq 0]$  of the characteristic  $\vec{s}$  across the octagon stencil, we obtain the following solution of the Cauchy problem along the short characteristic

$$I_i = I_O e^{-\tau_o} + \int_{-\tau_o}^0 e^t B(t) dt, \quad (4.68)$$

$$F_i = F_O e^{-\tau_o} - \int_{-\tau_o}^0 e^t B'(t) dt, \quad (4.69)$$

where  $I_O$  and  $F_O = I_O - B_O$  are the initial values at the entrance point  $O$ . Mathematically, the two expressions (4.68) and (4.69) are fully equivalent. Equation (4.68) tells us that our numerical scheme will be positive (i.e. will produce  $I_i > 0$ ) whenever the corresponding interpolation schemes guarantee the positiveness of  $I_O$  and  $B(\tau)$  at  $-\tau_o \leq \tau \leq 0$ . Equation (4.69) is more suitable for practical use because it ensures higher numerical accuracy in optically thick environments, where  $|I - B| \ll B$ , and simplifies mathematics because we only need a linear interpolation for  $B'$ .

To transform Eq. (4.69) into a finite-difference equation, we have to adopt an interpolation scheme for the source function  $B(\tau)$  along the characteristic  $\vec{s}$ . Correct transition to the diffusion limit requires that this interpolation be at least of a second-order (i.e. quadratic) in  $\tau$ . We construct such positive piece-wise parabolic interpolation (see section 4.3.4 below) by using the three values  $B_O \geq 0$ ,  $B_i \geq 0$ , and  $B_{O_m} \geq 0$  of the source function on  $\vec{s}$ ; recall that  $O_m$  is the entrance point to the octagon stencil for the opposite beamlet  $-\vec{\Omega}_L$ , which is always present in the  $S_n$  angular quadrature. Between points  $O$  and  $i$ , where  $-\tau_o \leq \tau \leq 0$ , we have

$$B(\tau) = B_i + B'_i \tau + \frac{1}{2} B''_i \tau^2. \quad (4.70)$$

Making use of the boundary value  $B(-\tau_o) = B_O$ , we get

$$B''_i = \frac{2}{\tau_o^2} (B_O - B_i + B'_i \tau_o). \quad (4.71)$$

Note that the piece-wise parabolic interpolation to  $B(\tau)$  along the segment  $O - i - O_m$  is constructed in such a way that the  $\tau$ -derivative  $B'$  is continuous at  $\tau = 0$ , i.e.  $B'(-0) = B'(0)$ , while generally  $B''(-0) \neq B''(0)$ . In other words, in our scheme only  $B'_i$  is guaranteed to be a single-valued quantity, defined at every physical mesh vertex, but not  $B''$ . Substituting Eq. (4.71) into (4.70), we obtain a linear interpolation for  $B'(\tau)$

$$B'(\tau) = B'_i \left( 1 + 2 \frac{\tau}{\tau_o} \right) + 2 \frac{\tau}{\tau_o} \frac{B_O - B_i}{\tau_o}, \quad (4.72)$$

which can, in its turn, be substituted into Eq. (4.69) to yield

$$F_i = e^{-\tau_o} F_O + \beta_{BO} (B_O - B_i) - \beta_{B1} B'_i, \quad (4.73)$$

where

$$\begin{aligned}\beta_{BO} &= 2 \frac{1 - (1 + \tau_o)e^{-\tau_o}}{\tau_o^2} = \\ &= 1 - \frac{2}{3}\tau_o + \frac{\tau_o^2}{4} - \frac{\tau_o^3}{15} + \frac{\tau_o^4}{72} - \dots, \quad |\tau_o| \ll 1,\end{aligned}\tag{4.74}$$

$$\begin{aligned}\beta_{B1} &= \frac{\tau_o - 2 + (2 + \tau_o)e^{-\tau_o}}{\tau_o} = \\ &= \tau_o^2 \left( \frac{1}{6} - \frac{\tau_o}{12} + \frac{\tau_o^2}{40} - \frac{\tau_o^3}{180} + \frac{\tau_o^4}{1008} - \dots \right), \quad |\tau_o| \ll 1.\end{aligned}\tag{4.75}$$

Equations (4.73), (4.74) and (4.75) represent the principal finite-difference formula for numerical integration of the transfer equation (4.7), implemented in the subroutine RADFNU (file ‘f09\_rad.f’).

The optical length  $\tau_o$  in the above formulae is calculated by summing up the 3D optical distances traversed by the ray  $\vec{s}$  across all the donor cells in a given octagon stencil,

$$\tau_o = \sum_m k_{i_{dc,m}} \Delta s_{i_{dc,m}}.\tag{4.76}$$

Here  $\Delta s_{i_{dc,m}}$  is the 3D geometrical distance traversed by the ray  $\vec{s}$  across the  $m$ -th donor cell, known from the solution of the ray-octagon problem (see section 4.6.7 below).

Although our principal finite-difference quadrature (4.73) does not readily reveal that it is positive with respect to the radiation intensity  $I_i = F_i + B_i$ , its very derivation provides a rigorous proof that this is true whenever

- $I_O = F_O + B_O \geq 0$ , and
- the values of  $B'_i$  and  $B_O \geq 0$ ,  $B_i \geq 0$  are *positively consistent* in the sense that they ensure positivity of the quadratic interpolation (4.70), (4.71) on the interval  $[-\tau_o \leq \tau \leq 0]$ .

In addition to the residual intensities  $F_i$  at mesh nodes, we also need for the SSI part of the algorithm (see section 3.1) their partial derivatives  $\partial F_i / \partial T_{i\alpha}$  with respect to the cell temperatures  $T_{i\alpha}$  in the adjoining 4 cells ( $\alpha = 1, 2, 3, 4$  is the quadrant index). Because these derivatives are only needed to ensure numerical stability, they can generally be calculated less accurately, e.g. by assuming constant transport coefficients — i.e. by assuming  $\tau_o$  to be independent of  $T_{i\alpha}$ . Then, from Eq. (4.73) we simply have

$$\frac{\partial F_i}{\partial T_{i\alpha}} = e^{-\tau_o} \frac{\partial F_O}{\partial T_{i\alpha}} + \beta_{BO} \left( \frac{\partial B_O}{\partial T_{i\alpha}} - \frac{\partial B_i}{\partial T_{i\alpha}} \right) + \beta_{B1} \frac{\partial B'_i}{\partial T_{i\alpha}}.\tag{4.77}$$

### 3. Initial values for characteristic integration

*a. Linear interpolation in space between the donor vertices.* To apply the finite-difference quadrature (4.73), we need the initial values  $F_O$ ,  $B_O$  at point  $O$  which lies on the edge between the two donor vertices  $i_{d1}$  and  $i_{d2}$ . The preparatory CP (Cauchy-problem) sorting procedure (see section 4.6.12 below) ensures that, by the time  $F_i$  is to be calculated,

the node values  $F_{i_{d1}}$  and  $F_{i_{d2}}$  have already been computed. This suffices for calculating  $F_O$  as a linear interpolation between the two end nodes  $i_{d1}$  and  $i_{d2}$

$$F_O = F_O^{(1)} \equiv F_{i_{d1}} + q_{Os}(F_{i_{d2}} - F_{i_{d1}}), \quad (4.78)$$

where

$$0 \leq q_{Os} = \frac{|\vec{x}_O - \vec{x}_{i_{d1}}|}{|\vec{x}_{i_{d2}} - \vec{x}_{i_{d1}}|} \leq 1 \quad (4.79)$$

is known from the solution of the ray-octagon problem and stored in the rd data set. If, in addition, the source function  $B_O$  is also calculated as the linear interpolation

$$B_O = B_O^{(1)} \equiv B_{i_{d1}} + q_{Os}(B_{i_{d2}} - B_{i_{d1}}), \quad (4.80)$$

then the positivity of our algorithm with respect to the radiation intensity  $I = F + B$  is a straightforward consequence of the above formulae.

*b. Higher-order interpolation in space between the donor vertices.* The above linear interpolation has an important drawback of being not fully consistent with the necessity to have a second-order (quadratic) interpolation for the source function  $B$  in order to reproduce the diffusion limit. Therefore, it leads to a degradation of accuracy and loss of convergence on optically thick meshes — as is demonstrated by the test problems in section 10.

At the same time, there is no principal obstacle to raising the interpolation order for the values  $B_O$  of the known source function  $B(\vec{x})$ . To do the same for the unknown field variable  $F_i$  (or  $I_i$ ) would require an iterative loop (or introduction of additional collocation points) — an option that we would prefer to avoid. Instead, we note that in the diffusion limit, where the optical thickness of the simulated region is much greater than 1 and  $|F| \ll B$ , the linear interpolation for  $F$  should suffice. But if we use the linear value (4.78) for  $F_O$  together with a higher-order interpolation for  $B_O = B_O^{(2)}$ , we loose positivity of the intensity  $I$ . Its positivity can be restored by applying the linear interpolation to  $I_O$  rather than to  $F_O$ , but then the convergence to the diffusion limit is compromised. Here, as a practical solution to this dilemma, the following modification of Eq. (4.73) is proposed which preserves positivity of  $I_i$ .

With  $F_O = F_O^{(1)}$  and  $B_O = B_O^{(2)}$  we have

$$I_O = F_O^{(1)} + B_O^{(2)} = I_O^{(1)} - \Delta_{BO}, \quad \Delta_{BO} = B_O^{(1)} - B_O^{(2)}. \quad (4.81)$$

Since  $I_O^{(1)} \geq 0$ , the value of  $I_i$  calculated from Eq. (4.68) will be non-negative whenever

$$\Delta_{BO} \leq e^{\tau_o} \int_{-\tau_o}^0 e^t B(t) dt. \quad (4.82)$$

But because, by virtue of Eq. (4.114), we have

$$e^{\tau_o} \int_{-\tau_o}^0 e^t B(t) dt \geq (e^{\tau_o} - 1) B_{min} > (\tau_o + \tau_o^2/2) B_{min}, \quad (4.83)$$

where

$$B_{min} = (1 - \varepsilon_{du}) \min[B_O^{(2)}, B_i], \quad (4.84)$$

it is sufficient to satisfy the condition

$$\Delta_{BO} \leq (\tau_o + \tau_o^2/2)B_{min} \quad (4.85)$$

to ensure positivity of  $I_i$ .

However, in addition to (4.85), the following consideration must be taken into account. Because  $\Delta_{BO}$  is in fact an unphysical correction to the radiation intensity  $I_O$ , arising from mathematical inconsistency in the internode interpolation schemes for  $F_O$  and  $B_O$ , its absolute value must also be limited to avoid strong unphysical distortions of the radiation field in the optically thin cases  $\tau_o \ll 1$ . Hence, we have to impose an additional constraint

$$|\Delta_{BO}| \leq \varepsilon_{2BO} \left[ F_O^{(1)} + B_O^{(1)} + (\tau_o + \tau_o^2/2)B_{min} \right], \quad (4.86)$$

where  $\varepsilon_{2BO}$  is a small parameter; by default the value  $\varepsilon_{2BO} = 0.03$  is used.

Finally, we arrive at the following modified version of Eq. (4.73) which always produces non-negative intensities  $I_i$  with the linearly interpolated values of  $F_O = F_O^{(1)}$ :

$$F_i = e^{-\tau_o} \left( F_O^{(1)} + \tilde{\Delta}_{BO} \right) + \beta_{BO} \left( B_O^{(2)} - B_i \right) - \beta_{B1} B_i' \quad (4.87)$$

where  $\tilde{\Delta}_{BO} = \Delta_{BO}$  (a purely linear interpolation for  $I_O$ ) if either (4.85) or (4.86) are not satisfied, and

$$\tilde{\Delta}_{BO} = \begin{cases} 0, & \Delta_{BO} \leq \tau_o B_{min}, \\ \Delta_{BO} \frac{\Delta_{BO} - \tau_o B_{min}}{\tau_o^2 B_{min}/2}, & \tau_o B_{min} < \Delta_{BO} < (\tau_o + \tau_o^2/2)B_{min}, \end{cases} \quad (4.88)$$

otherwise. Note that at this stage the node values  $B_i$  are considered to be fixed and not affected by the choice of the internode interpolation scheme. The second-order positive interpolation (4.70) along the short characteristic should, of course, be also constructed by using the higher-order values  $B_O^{(2)}$ ,  $B_{Om}^{(2)}$ ,  $B_P^{(2)}$ ,  $B_{Pm}^{(2)}$  — as it is described in subsection 4.3.4 below.

*c. Linear and quadratic interpolation along the angular coordinate  $\Omega_R$ .* In the  $rz$  geometry, where the angular variable  $\vec{\Omega}$  is discretized with respect to its local components  $\vec{\Omega} = (\Omega_R, \Omega_Z)$ , only the  $\Omega_Z$  component remains constant along the characteristic, while  $\Omega_R$  monotonically increases along the photon trajectory. Hence, if we apply Eq. (4.73) to calculate  $F_i = F_{L,i}$  at node  $i$  for the  $S_n$  direction  $(\Omega_{R,L}, \Omega_{Z,L})$ , the initial value  $F_O$  must be known for a different value  $\Omega_{RO} \leq \Omega_{R,L}$  of the radial angular component  $\Omega_R$  — i.e.  $F_O$  must be calculated by interpolating not only between the two donor vertices  $i_{d1}$  and  $i_{d2}$  in space, but also by interpolating between the two donor nodes  $\tilde{m}_{2R,od1}$  and  $\tilde{m}_{2R,od2} = \tilde{m}_{2R,od1} + 1$  along the  $\Omega_R$  axis (see section 4.2.3 and Fig. 4.6). To simplify the notation, we designate  $\tilde{m}_{2R,od1} = m_{2R}$  below in this subsection. Then we can write

$$\Omega_{RO} = \Omega_{R,m_{2R}} + q_{O\Omega_R} (\Omega_{R,m_{2R}+1} - \Omega_{R,m_{2R}}), \quad (4.89)$$

where  $m_{2R}$  and the parameter  $0 \leq q_{O\Omega_R} \leq 1$  [similar to the parameter  $q_{Os}$  in Eqs. (4.78)–(4.80)] are known from the solution of the ray-octagon problem and stored in the rd data set.

Note that the values  $B_O$  and  $B_i$  of the source function in Eq. (4.73) do not depend on the direction  $\vec{\Omega}$ , whereas  $B'_i$  is automatically calculated for the same local ordinate  $(\Omega_{R,L}, \Omega_{Z,L})$ , for which  $F_i = F_{L,i}$  is to be found.

When interpolating along the  $\Omega_R$  axis, we can assume that known are (whenever  $m_{2R} > 1$ ) the three values  $F_{O,m_{2R}-1}^{(1)}$ ,  $F_{O,m_{2R}}^{(1)}$ , and  $F_{O,m_{2R}+1}^{(1)}$  from the positive linear interpolation in space between the donor nodes  $i_{d1}$  and  $i_{d2}$ ; for  $m_{2R} = 1$  the lowest value  $F_{O,m_{2R}-1}^{(1)}$  is not available, and we are forced to use the linear interpolation between the  $F_{O,m_{2R}}^{(1)}$  and  $F_{O,m_{2R}+1}^{(1)}$ . Because the source function  $B_O^{(1)}$  does not depend on  $\Omega_R$ , a *monotonic* positive interpolation for  $F_O^{(1)}$  along  $\Omega_R$  will guarantee positivity of the intensity  $I = F + B$ . A monotonic interpolation, based on the three angular donor nodes  $\Omega_{R,m_{2R}-1} < \Omega_{R,m_{2R}} < \Omega_{R,m_{2R}+1}$ , is readily obtained as a particular case of the PPOS3P interpolation scheme, described in appendix C 1, with  $\varepsilon_d = \varepsilon_u = 0$ .

We write down the sought-for angular interpolation for  $F_O^{(1)}$  in the form

$$F_O^{(1)} = F_{O,m_{2R}}^{(1)} + a_- f_- + a_+ f_+, \quad (4.90)$$

where

$$f_- = F_{O,m_{2R}-1}^{(1)} - F_{O,m_{2R}}^{(1)}, \quad f_+ = F_{O,m_{2R}+1}^{(1)} - F_{O,m_{2R}}^{(1)}, \quad (4.91)$$

and  $a_-$  and  $a_+$  are unknown interpolation coefficients. Having denoted

$$0 \leq \xi \equiv q_{O\Omega_R} = \frac{\Omega_{RO} - \Omega_{R,m_{2R}}}{\Omega_{R,m_{2R}+1} - \Omega_{R,m_{2R}}} \leq 1, \quad \eta = \frac{\Omega_{R,m_{2R}+1} - \Omega_{R,m_{2R}}}{\Omega_{R,m_{2R}-1} - \Omega_{R,m_{2R}}} < 0, \quad (4.92)$$

we can write

$$F_O^{(1)} = F_{O,m_{2R}}^{(1)} + g(\xi - \xi^2) + \xi^2 f_+, \quad (4.93)$$

where  $g$  is the slope at  $\xi = 0$ . In the regular case, where no slope correction is needed, we obtain

$$g = g_r \equiv \frac{f_+ - \eta^2 f_-}{1 - \eta}. \quad (4.94)$$

The linear interpolation is recovered as a particular case of

$$a_- = 0, \quad a_+ = \xi. \quad (4.95)$$

Finally, the *monotonic piece-wise parabolic* interpolation is obtained by applying the following slope corrections in Eq. (4.93).

*Case 1:*  $\bar{f}_- < 0 < \bar{f}_+$ :

$$g = \min(g_r, 2\eta f_-, 2f_+) \Rightarrow \quad (4.96)$$

$$a_- = \begin{cases} \frac{\eta^2(\xi^2 - \xi)}{1 - \eta}, & g_r \leq \min(2\eta f_-, 2f_+), \\ 0, & g_r > \min(2\eta f_-, 2f_+) \wedge f_+ \leq \eta f_-, \\ 2\eta(\xi - \xi^2), & g_r > \min(2\eta f_-, 2f_+) \wedge f_+ > \eta f_-, \end{cases} \quad (4.97)$$

$$a_+ = \begin{cases} \frac{\xi - \xi^2 \eta}{1 - \eta}, & g_r \leq \min(2\eta f_-, 2f_+), \\ 2\xi - \xi^2, & g_r > \min(2\eta f_-, 2f_+) \wedge f_+ \leq \eta f_-, \\ \xi^2, & g_r > \min(2\eta f_-, 2f_+) \wedge f_+ > \eta f_-. \end{cases} \quad (4.98)$$

Case 2:  $\bar{f}_- > 0 > \bar{f}_+$ :

$$g = \max(g_r, 2\eta f_-, 2f_+) \Rightarrow \quad (4.99)$$

$$a_- = \begin{cases} \frac{\eta^2(\xi^2 - \xi)}{1 - \eta}, & g_r \geq \max(2\eta f_-, 2f_+), \\ 0, & g_r < \max(2\eta f_-, 2f_+) \wedge f_+ \geq \eta f_-, \\ 2\eta(\xi - \xi^2), & g_r < \max(2\eta f_-, 2f_+) \wedge f_+ < \eta f_-, \end{cases} \quad (4.100)$$

$$a_+ = \begin{cases} \frac{\xi - \xi^2\eta}{1 - \eta}, & g_r \geq \max(2\eta f_-, 2f_+), \\ 2\xi - \xi^2, & g_r < \max(2\eta f_-, 2f_+) \wedge f_+ \geq \eta f_-, \\ \xi^2, & g_r < \max(2\eta f_-, 2f_+) \wedge f_+ < \eta f_-. \end{cases} \quad (4.101)$$

Cases 3 and 4:  $\bar{f}_- \cdot \bar{f}_+ \geq 0$ :

$$g = 0 \Rightarrow a_- = 0, \quad a_+ = \xi^2. \quad (4.102)$$

In the subroutine `RADFNU`, where this interpolation is implemented, the choice between the linear and quadratic interpolation options along the  $\Omega_R$  axis is made by setting the logical flag `if2ord_omR` equal to either `.false.` (linear  $\Omega_R$ -interpolation) or `.true.` (quadratic  $\Omega_R$ -interpolation).

#### 4. Positive piece-wise parabolic interpolation of $B$ along the short characteristic and calculation of the $\tau$ -derivative $B'$

To calculate the value  $B'_i$  of the  $\tau$ -derivative (4.65) at node  $i$ , we use the values of the source function  $B$  at two r-donor points, found by the solution of the ray-octagon problems for the direct beamlet  $\vec{\Omega}_L$  and the opposite (reverse) beamlet  $-\vec{\Omega}_L$  (plus the value  $B_i$  at the node  $i$  itself). In the  $rz$  case we generally have two pairs of such donor points, namely, (i) the 3D octagon-entry points  $O$  and  $O_m$ , and (ii) the 2D *primary* donor points  $P$  and  $P_m$ , calculated along a straight line in the computational plane that is tangent to the 3D vector  $\pm\vec{\Omega}_L$  at vertex  $i$ ; see Fig. 4.7. The 2D primary pair  $P, P_m$  is used to calculate the *regular* (i.e. without slope correction) value

$$B'_{i,reg} = a_-(B_P - B_i) + a_+(B_{P_m} - B_i), \quad (4.103)$$

where  $B_P$  and  $B_{P_m}$  are, respectively, the values of  $B$  at points  $P$  and  $P_m$ ,

$$a_- = -\frac{\tau_{pm}}{\tau_p} \frac{1}{\tau_p + \tau_{pm}}, \quad a_+ = \frac{\tau_p}{\tau_{pm}} \frac{1}{\tau_p + \tau_{pm}}, \quad (4.104)$$

$$\tau_p = \frac{k_{i_{dc,1}} \sigma_p}{\sqrt{\Omega_{xL}^2 + \Omega_{yL}^2}}, \quad \tau_{pm} = \frac{k_{i_{dcm,1}} \sigma_{pm}}{\sqrt{\Omega_{xL}^2 + \Omega_{yL}^2}}, \quad (4.105)$$

$\sigma_p$  (or  $\sigma_{pm}$ ) is the 2D distance between points  $i$  and  $P$  (or  $i$  and  $P_m$ ) in the computational plane, and  $k_{i_{dc,1}}$  (or  $k_{i_{dcm,1}}$ ) is the radiation absorption coefficient in the primary donor cell  $i_{dc,1}$  (or  $i_{dcm,1}$ ) for the direct beamlet  $\vec{\Omega}_L$  (or the reverse beamlet  $-\vec{\Omega}_L$ ).

Once the unconstrained value  $B'_{i,reg}$  has been evaluated, the conditions are checked for an eventual slope correction to ensure positivity of the piece-wise parabolic interpolation along the 3D short-characteristic segment  $O - i - O_m$ . These conditions must, of course, be based on the  $B_O$  and  $B_{O_m}$  values of the source function at the 3D donor points  $O$  and  $O_m$ , and if the slope correction is enacted, then the  $\tau$ -derivative  $B'_i$  is calculated from the formula

$$B'_i = b_- B_O + b_i B_i + b_+ B_{O_m}, \quad (4.106)$$

where the coefficients  $b_-$ ,  $b_i$ , and  $b_+$  are determined in the process of checking the following conditions

Case A:  $B'_{i,reg} \geq 0$ :

$$B'_i = \min(B'_{i,reg}, B'_{i,ll}, B'_{i,lr}); \quad (4.107)$$

$$B'_{i,ll} = \begin{cases} \frac{4\varepsilon_{du} B_O - 2f_-}{\tau_o} \Rightarrow b_- = \frac{4\varepsilon_{du} - 2}{\tau_o}, b_i = \frac{2}{\tau_o}, b_+ = 0; & f_- < 0; \\ \frac{4\varepsilon_{du} B_i}{\tau_o} \Rightarrow b_- = b_+ = 0, b_i = \frac{4\varepsilon_{du}}{\tau_o}; & f_- \geq 0; \end{cases} \quad (4.108)$$

$$B'_{i,lr} = \begin{cases} \frac{4\varepsilon_{du} B_i}{\tau_{om}} \Rightarrow b_- = b_+ = 0, b_i = \frac{4\varepsilon_{du}}{\tau_{om}}; & f_+ < 0; \\ \frac{4\varepsilon_{du} B_{O_m} + 2f_+}{\tau_{om}} \Rightarrow b_- = 0, b_i = -\frac{2}{\tau_{om}}, b_+ = \frac{4\varepsilon_{du} + 2}{\tau_{om}}; & f_+ \geq 0. \end{cases} \quad (4.109)$$

Case B:  $B'_{i,reg} < 0$ :

$$B'_i = \max(B'_{i,reg}, B'_{i,ll}, B'_{i,lr}); \quad (4.110)$$

$$B'_{i,ll} = \begin{cases} -\frac{4\varepsilon_{du} B_i}{\tau_o} \Rightarrow b_- = b_+ = 0, b_i = -\frac{4\varepsilon_{du}}{\tau_o}; & f_- < 0; \\ -\frac{4\varepsilon_{du} B_O + 2f_-}{\tau_o} \Rightarrow b_- = -\frac{4\varepsilon_{du} + 2}{\tau_o}, b_i = \frac{2}{\tau_o}, b_+ = 0; & f_- \geq 0; \end{cases} \quad (4.111)$$

$$B'_{i,lr} = \begin{cases} \frac{2f_+ - 4\varepsilon_{du} B_{O_m}}{\tau_{om}} \Rightarrow b_- = 0, b_i = -\frac{2}{\tau_{om}}, b_+ = \frac{2 - 4\varepsilon_{du}}{\tau_{om}}; & f_+ < 0; \\ -\frac{4\varepsilon_{du} B_i}{\tau_{om}} \Rightarrow b_- = b_+ = 0, b_i = -\frac{4\varepsilon_{du}}{\tau_{om}}; & f_+ \geq 0. \end{cases} \quad (4.112)$$

where

$$f_- = B_O - B_i, \quad f_+ = B_{O_m} - B_i, \quad (4.113)$$

and  $B'_{i,ll}$  and  $B'_{i,lr}$  are, respectively, the slope limits stemming from the left,  $[-\tau_o, 0]$ , and the right,  $[0, \tau_{om}]$ , interpolation half-intervals. Conditions (4.107)–(4.112) represent a simplified version of the PPOS3P interpolation scheme, described in appendix C 1, with  $\varepsilon_{du} = \varepsilon_d = \varepsilon_u$ ,  $0 \leq \varepsilon_{du} < 1$  being the maximum allowed relative amplitude of non-monotonic variation. In other words, the above slope correction guarantees

$$\begin{aligned} B(\tau) &\geq (1 - \varepsilon_{du}) \min(B_O, B_i), \quad \tau \in [-\tau_o, 0], \\ B(\tau) &\geq (1 - \varepsilon_{du}) \min(B_i, B_{O_m}), \quad \tau \in [0, \tau_{om}]. \end{aligned} \quad (4.114)$$

For  $\varepsilon_{du} = 0$  we obtain a monotonic positive interpolation of  $B(\tau)$  for  $\tau \in [-\tau_o, \tau_{om}]$ . By default the value  $\varepsilon_{du} = 1/32 \approx 3\%$  is used. Note that the corrected slope  $B'_i$  always has the same sign as the regular value  $B'_{i,reg}$ .

If none of the above conditions for slope correction is fulfilled, the regular value  $B'_i = B'_{i,reg}$ , given by Eq. (4.103), is adopted. The reason for using the 2D primary donor points  $P, P_m$  instead of  $O, O_m$  for evaluating  $B'_{i,reg}$  is simply the higher final accuracy achieved in this way. Of course, in the Cartesian  $xy$  case the donor points  $O$  and  $O_m$  coincide with  $P$  and  $P_m$ .

To calculate the inter-nodal values  $B_P, B_{P_m}, B_O, B_{O_m}$ , either a straightforward linear interpolation (4.80) (i.e. the values  $B_P^{(1)}, B_{P_m}^{(1)}, B_O^{(1)}, B_{O_m}^{(1)}$ ), or a more sophisticated second-order interpolation scheme, described in the next subsection, (i.e. the values  $B_P^{(2)}, B_{P_m}^{(2)}, B_O^{(2)}, B_{O_m}^{(2)}$ ) can be used. As is demonstrated with numerical tests (see subsections 10 2 3 and 10 4 4), although the linear inter-nodal interpolation (4.80) already reproduces the diffusion limit to a fair accuracy, it still (in the 2013 versions of the RALEF code) lacks numerical convergence to the exact diffusive solution as the spatial mesh is refined.

### 5. Piece-wise parabolic interpolation of the source function $B$ along the cell edges

Consider an arbitrary function  $f(x)$  of a single variable  $x$ , whose discrete values  $f_1 = f(x_1)$  and  $f_2 = f(x_2)$  are known at two different points  $x_1 < x_2$ . Generally we adopt the following standard form

$$f(x) = f_1 + (f_2 - f_1)\xi + d^2 f_{12} \xi(\xi - 1) \quad (4.115)$$

for the piece-wise parabolic (2nd-order) interpolation of  $f(x)$  on the normalized interval

$$0 \leq \xi \equiv \frac{x - x_1}{x_2 - x_1} \leq 1. \quad (4.116)$$

The quantity  $d^2 f_{12}$  may be called a *2nd-order increment* of  $f(x)$  over the original interval  $[x_1, x_2]$ . It can be evaluated from one the following formulae

$$d^2 f_{12} = \frac{1}{2}(x_2 - x_1)^2 f'', \quad (4.117)$$

$$d^2 f_{12} = \frac{1}{2}(x_2 - x_1)(f'_2 - f'_1), \quad (4.118)$$

$$d^2 f_{12} = 4 \left[ f_1 + \frac{1}{2}(f_2 - f_1) - f_{3/2} \right], \quad (4.119)$$

that are equivalent from the point of view of finite-difference approximation; here  $f'$  and  $f''$  are, respectively, the 1st and the 2nd derivatives of  $f(x)$ ,  $f_{3/2}$  is the value of  $f(x)$  at the midpoint  $(x_1 + x_2)/2$ .

According to the above definitions, the 2nd-order interpolated value  $B_O^{(2)}$  of the source function  $B$  at a point  $O$  on a cell edge  $\vec{\lambda}_{12}$  between two donor vertices  $i_{d1} = 1$  and  $i_{d2} = 2$  is calculated as

$$B_O^{(2)} = B_1 + (B_2 - B_1) q_{O_s} + d^2 B_{12} q_{O_s}(q_{O_s} - 1) \quad (4.120)$$

— provided that the 2nd-order increment  $d^2 B_{12}$  is known for every physical cell edge;  $q_{O_s}$  has been defined in Eq. (4.79).

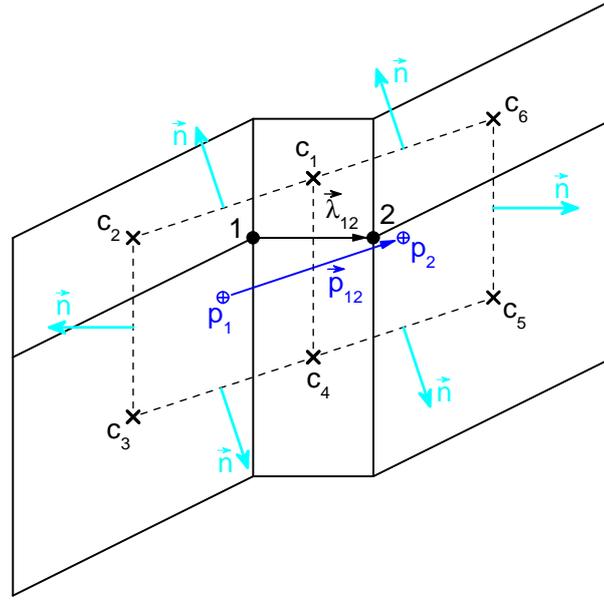


FIG. 4.9: The 6-point stencil for evaluating the 2nd-order increment  $d^2 B_{12}$  along a cell edge  $\vec{\lambda}_{12}$  between two mesh nodes 1 and 2;  $c_1, c_2, \dots, c_6$  are the centers of the 6 mesh cells surrounding the edge  $\vec{\lambda}_{12}$ ;  $p_1$  and  $p_2$  are the secondary cell centers, i.e. the centers of quadrilaterals  $c_{1-4}$  and  $c_{1-6}$ .

We evaluate the 2nd-order increment  $d^2 B_{12}$  along a given cell edge  $\vec{\lambda}_{12}$  by using the cell-centered values  $B_{c_1}, B_{c_2}, \dots, B_{c_6}$  of the source function on a 6-cell stencil shown in Fig. 4.9. As a first step, the gradients  $\nabla B_1$  and  $\nabla B_2$  at the secondary central points  $p_1$  and  $p_2$  are evaluated by integrating  $B$  along two adjacent closed contours  $c_1-c_2-c_3-c_4-c_1$  and  $c_1-c_4-c_5-c_6-c_1$

$$\nabla B_1 = \frac{1}{A_{c_{1-4}}} \oint_{c_{1-4}} B \vec{n} d\lambda, \quad (4.121)$$

$$\nabla B_2 = \frac{1}{A_{c_{1-6}}} \oint_{c_{1-6}} B \vec{n} d\lambda, \quad (4.122)$$

where  $d\lambda$  is the length element along the corresponding polygon contour,  $\vec{n}$  is the outward unit normal to that contour, and  $A_{c_{1-4}}$  [ $A_{c_{1-6}}$ ] is the area of the corresponding quadrilateral. The highest (2nd-order) accuracy is achieved when the values of  $\nabla B_1$  and  $\nabla B_2$  are assigned to the centroids  $p_1$  and  $p_2$  of the quadrilaterals  $c_{1-4}$  and  $c_{1-6}$ . If, however, the contour integrals in (4.121) and (4.122) are calculated by using the linear interpolation of  $B$  along the edges of  $c_{1-4}$  and  $c_{1-6}$  (as is the case in the 2013 versions of the RALEF code), we do not of course fully recover the 2nd-order approximation for the gradients.

Now we can take into account that vectors  $\vec{\lambda}_{12}$  between the original nodes 1 and 2, and  $\vec{p}_{12}$  between the secondary centers  $p_1$  and  $p_2$ , differ in magnitude and direction and evaluate the second derivative along  $\vec{\lambda}_{12}$  as

$$\frac{d^2 B}{d\lambda_{12}^2} = \frac{(\nabla B_2 - \nabla B_1) \cdot \vec{\lambda}_{12}}{\vec{p}_{12} \cdot \vec{\lambda}_{12}}. \quad (4.123)$$

Having substituted it into the general formula (4.118), we obtain the final expression for the 2nd-order  $B$ -increment along the primary cell edge  $\vec{\lambda}_{12}$

$$d^2 B_{12} = \frac{1}{2} \lambda_{12}^2 \frac{(\nabla B_2 - \nabla B_1) \cdot \vec{\lambda}_{12}}{\vec{p}_{12} \cdot \vec{\lambda}_{12}}. \quad (4.124)$$

To ensure positivity and (possibly) monotonicity of the general piece-wise parabolic interpolation (4.115), it is sufficient (but not necessary) to impose the following constraints on the values of the 2nd-order increments  $d^2 f_{12}$ :

$$\begin{aligned} -(f_2 - f_1) - 4\varepsilon_{du} f_2 &\leq d^2 f_{12} \leq (f_2 - f_1) + 4\varepsilon_{du} f_1, \text{ for } f_1 \leq f_2, \\ -(f_1 - f_2) - 4\varepsilon_{du} f_1 &\leq d^2 f_{12} \leq (f_1 - f_2) + 4\varepsilon_{du} f_2, \text{ for } f_1 > f_2, \end{aligned} \quad (4.125)$$

where a small parameter  $\varepsilon_{du} < 1$  defines the maximum relative amount of parabolic overshoot or undershoot with respect to strictly monotonic variation; for  $\varepsilon_{du} = 0$  we obtain a monotonic interpolation. Conditions (4.125) are easily derived by considering corresponding limiting parabolas (shown as dashed curves in Figs. C.1–C.4 in appendix C1) that are tangent to the basic interpolation parabola (4.115). The two conditions (4.125) are readily combined into one

$$-|f_2 - f_1| - 4\varepsilon_{du} \max(f_1, f_2) \leq d^2 f_{12} \leq |f_2 - f_1| + 4\varepsilon_{du} \min(f_1, f_2), \quad (4.126)$$

which is implemented in the subroutine DD2BNU where the values of  $d^2 B_{12}$  are calculated. The default value of the non-monotonicity parameter  $\varepsilon_{du}$  is  $\varepsilon_{du} = 1/32$ . Besides (4.126), an additional absolute restriction

$$|d^2 B_{12}| \leq 2\varepsilon_{2mx} (B_1 + B_2) \quad (4.127)$$

is imposed in the subroutine DD2BNU on possible values of  $d^2 B_{12}$ , where the default value of parameter  $\varepsilon_{2mx}$  is set to be  $\varepsilon_{2mx} = 0.1$ .

## 6. Vertex emissivities

In the transport equation (4.7) the source term  $B_{[k]} = B_{[k]}(T)$  is a function of temperature  $T$  only. Hence, it naturally becomes a cell-centered quantity in our numerical scheme for the hydrodynamic equations. The latter means that the integration procedure described in subsection 4.3.2 can be applied only after we specify the algorithm for calculating the nodal values  $B_{v[k],i}$  of the source function.

We proceed from the assumption that the nodal values  $B_{v[k],i}$  at a vertex  $i$  in different spectral groups  $[k]$  should be calculated from a single value of the nodal  $B$ -temperature  $T_{B,i}$ ,

$$B_{v[k],i} = B_{[k]}(T_{B,i}). \quad (4.128)$$

There is, of course, no ground to assume that the  $B$ -temperatures  $T_{B,i}$  should be equal to the vertex temperatures  $T_{v,i}$ , used in the thermal conduction scheme. A useful hint with regard to the most appropriate algorithm for evaluating  $T_{B,i}$  can be inferred by considering the limit of an optically thin plasma cloud, where  $I_\nu \ll B_\nu$  and the transfer equation (2.6) becomes

$$\vec{\Omega} \cdot \nabla I_\nu = k_\nu B_\nu. \quad (4.129)$$

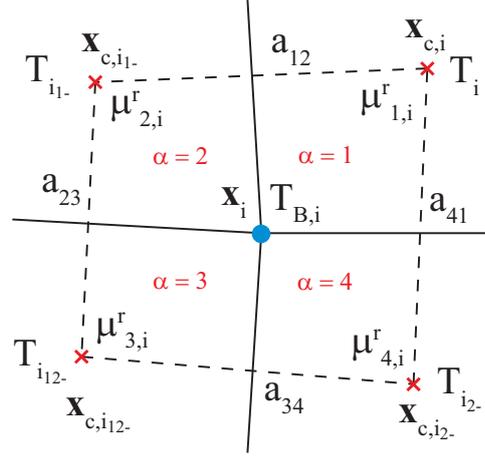


FIG. 4.10: Numbering of cell-centered quantities and quadrants used to evaluate the vertex  $B$ -temperatures  $T_{B,i}$ . The dashed contour delineates the  $c$ -quadrilateral.

Evidently, one would expect that in this case numerical integration of the transfer equation (4.129) should produce the most accurate results for the  $I_\nu$  field (hence, for the radiative cooling rate  $Q_r$ ) when the integral emissivity

$$J_r = 4\pi \int_0^\infty k_\nu B_\nu d\nu = 4\sigma_{SB} k_{Pl} T^4 \quad (4.130)$$

is interpolated over the spatial mesh in order to calculate the nodal  $B$ -temperatures  $T_{B,i}$ . Numerical tests confirm that this recipe is adequate for optically thick plasmas as well — including the extreme situations where the optical thickness of individual mesh cells is much greater than unity — though its mathematical justification for such situations is less obvious.

By analogy with the heat conduction, we use the bilinear interpolation from the four surrounding cell centers  $\vec{x}_{c,i_\alpha}$

$$\vec{x} = \frac{1}{4} [\vec{x}_{c,i}(1+\xi)(1+\eta) + \vec{x}_{c,i1-}(1-\xi)(1+\eta) + \vec{x}_{c,i2-}(1+\xi)(1-\eta) + \vec{x}_{c,i12-}(1-\xi)(1-\eta)], \quad (4.131)$$

$$J_r(\vec{x}) = \frac{1}{4} [J_{r,i}(1+\xi)(1+\eta) + J_{r,i1-}(1-\xi)(1+\eta) + J_{r,i2-}(1+\xi)(1-\eta) + J_{r,i12-}(1-\xi)(1-\eta)] \quad (4.132)$$

in the natural coordinates  $(\xi, \eta)$ , obtained from the mapping of a corresponding  $c$ -quadrilateral (see Fig. 4.10) onto the square  $(\xi, \eta) \in [-1, +1] \times [-1, +1]$ . Given the coordinates  $\vec{x} = \vec{x}_i$  of vertex  $i$ , we insert them into the left-hand side of Eq. (4.131) and calculate the natural coordinates  $(\xi_{v,i}, \eta_{v,i})$  of this vertex; then we substitute  $(\xi, \eta) = (\xi_{v,i}, \eta_{v,i})$  into the right-hand side of Eq. (4.132) to calculate the integral emissivity at node  $i$ . The values

of index  $i_\alpha$  are given by

$$i_\alpha = i - \text{icv}(\alpha, \text{iblk}) = \begin{cases} i = (i, j), & \alpha = 1, \\ i_{1-} = (i - 1, j), & \alpha = 2, \\ i_{12-} = (i - 1, j - 1), & \alpha = 3, \\ i_{2-} = (i, j - 1), & \alpha = 4, \end{cases} \quad (4.133)$$

where  $\alpha$  is the quadrant index of the corresponding cell relative to vertex  $i$ .

The final formulae for calculating the nodal  $B$ -temperatures  $T_{B,i}$  take the following form

$$T_{B,i}^4 = \mu_{1,i}^r T_i^4 + \mu_{2,i}^r T_{i_{1-}}^4 + \mu_{3,i}^r T_{i_{12-}}^4 + \mu_{4,i}^r T_{i_{2-}}^4 = \sum_{\alpha=1}^4 \mu_{\alpha,i}^r T_{i_\alpha}^4, \quad (4.134)$$

where

$$\mu_{\alpha,i}^r = \beta_{\alpha,i}^r \left( \sum_{\alpha=1}^4 \beta_{\alpha,i}^r \right)^{-1}, \quad \alpha = 1, 2, 3, 4, \quad (4.135)$$

$$\begin{aligned} \beta_{1,i}^r &= \max \{0; (1 + \xi_{v,i}) (1 + \eta_{v,i}) k_{Pl,i}\}, \\ \beta_{2,i}^r &= \max \{0; (1 - \xi_{v,i}) (1 + \eta_{v,i}) k_{Pl,i_{1-}}\}, \\ \beta_{3,i}^r &= \max \{0; (1 - \xi_{v,i}) (1 - \eta_{v,i}) k_{Pl,i_{12-}}\}, \\ \beta_{4,i}^r &= \max \{0; (1 + \xi_{v,i}) (1 - \eta_{v,i}) k_{Pl,i_{2-}}\}, \end{aligned} \quad (4.136)$$

and  $k_{Pl,i}^{-1}$  is the cell-centered value of the Planckian mean free path in cell  $i$ .

### 7. Temperature derivatives

To calculate the derivatives

$$F_{[k],L,i}^\alpha = \frac{\partial F_{[k],L,i}}{\partial T_{i_\alpha}} \quad (4.137)$$

with respect to the quadrant cell-centered temperatures  $T_{i_\alpha}$  (see Fig. 4.10), we make a simplifying assumption that variation of the cell temperature  $T_i$  does not affect the absorption coefficient  $k_{[k],i}$  inside this cell. Then, differentiating Eq. (??) with  $\tau_o$ ,  $\beta_{BO}$ ,  $\beta_{BM}$  kept constant, we obtain

$$\begin{aligned} F_{[k],L,i}^\alpha &= e^{-\tau_o} \left[ (1 - \chi_O) F_{[k],L,i_{d1}}^{\alpha_{d1}} + \chi_O F_{[k],L,i_{d2}}^{\alpha_{d2}} \right] + \\ &+ \left\{ \beta_{BO} \left[ (1 - \chi_O) \dot{B}_{v[k],i_{d1}} \mu_{\alpha_{d1},i_{d1}}^r + \chi_O \dot{B}_{v[k],i_{d2}} \mu_{\alpha_{d2},i_{d2}}^r \right] + \right. \\ &+ \beta_{BM} \left[ (1 - \chi_M) \dot{B}_{v[k],i_{d1},m_o} \mu_{\alpha_{d1},m_o,i_{d1},m_o}^r + \chi_M \dot{B}_{v[k],i_{d2},m_o} \mu_{\alpha_{d2},m_o,i_{d2},m_o}^r \right] - \\ &\left. - (\beta_{BO} + \beta_{BM}) \dot{B}_{v[k],i} \mu_{\alpha,i}^r \right\} (T_{i_\alpha})^3, \end{aligned} \quad (4.138)$$

where we have denoted

$$\dot{B}_{v[k],i} = \frac{1}{T^3} \frac{dB_{[k]}}{dT} \Big|_{T=T_{B,i}}, \quad (4.139)$$

and  $\alpha_{d1}$  is the quadrant index with respect to vertex  $i_{d1}$  for a cell that has the quadrant index  $\alpha$  with respect to vertex  $i$ ; similarly,  $\alpha_{d2}$  is the quadrant index with respect to vertex

$i_{d2}$  for a cell that has the quadrant index  $\alpha$  with respect to vertex  $i$ , and  $i_\alpha$  is defined in Eq. (??).

For given  $i, i_{d1}, i_{d2}$  the values of  $\alpha_{d1}$  and  $\alpha_{d2}$  can be calculated by using a universal local quadrant-to-quadrant mapping

$$i_{qq}(l, \alpha) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 \\ 0 & 1 & 4 & 0 \\ 1 & 2 & 3 & 4 \\ 2 & 0 & 0 & 3 \\ 0 & 4 & 0 & 0 \\ 4 & 3 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{pmatrix} \quad (4.140)$$

Here  $l$  is the local index on the 9-vertex stencil defined in Fig. 4.11, and  $i_{qq}(l, \alpha)$  is the quadrant index relative to vertex  $l$  for a cell that has the quadrant index  $\alpha$  relative to vertex  $i$  (vertex  $l = 5$  in local notation). Then, knowing global indices  $i$  and  $i_{d1}$ , we calculate the local index  $l_{d1}$  for vertex  $i_{d1}$  on the 9-point stencil around  $i$ , and then get  $\alpha_{d1} = i_{qq}(l_{d1}, \alpha)$ ; similarly,  $\alpha_{d2} = i_{qq}(l_{d2}, \alpha)$ . The indices  $\alpha_{d1,mo}$  and  $\alpha_{d2,mo}$  are calculated by using the same procedure with the same local stencil.

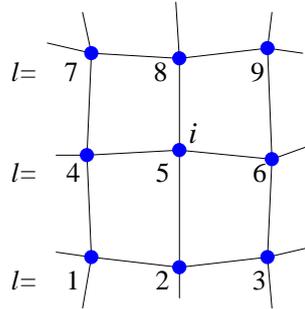


FIG. 4.11: Local numbering of vertices on the 9-point stencil around vertex  $i$ , used in the quadrant-to-quadrant mapping.

If the quadrant index  $\alpha_{d1}$  happens to be zero, all the terms in Eq. (4.138) marked by this index are set equal to zero; same for  $\alpha_{d2}, \alpha_{d1,mo}, \alpha_{d2,mo}$ . Here we actually make a second simplifying assumption that no dependence of  $F_{[k],L,i}$  on  $T_{i_\alpha}$  arises via distant loops, i.e. beyond the considered local vicinity of vertex  $i$  in the logical space. This is true only when the mesh is not too strongly distorted. The vertex-centered quantities  $\dot{B}_{v[k],i}$  are calculated simultaneously with  $B_{v[k],i}$ . As a result, the derivatives  $F_{[k],L,i}^\alpha$  are calculated in the same succession as  $F_{[k],L,i}$  by using the same causal r-donor-acceptor dependencies among the mesh vertices.

#### 4. Radiant energy deposition

##### 1. Volumetric versus flux-based algorithms

Here we assume that the discrete values of the radiation intensity,  $I_{[k],L,i}$ , and of the source function,  $B_{[k],i}$ , are known for each frequency group  $[k]$  and angular direction  $\vec{\Omega}_L$  at each physical mesh vertex  $i$ . Then, our task is to calculate the net radiative heating power

$$W_i^r = \int_{V_i} Q_r dV = - \int_{V_i} \text{div}(\vec{h}_r) dV \quad (4.141)$$

[erg s<sup>-1</sup>] of cell  $i$  whose volume is  $V_i$ . This part of the algorithm can be based on either the volumetric expression (2.18) for  $Q_r$ , which yields

$$W_i^r = \int_{V_i} \left[ \int_0^\infty k_\nu \left( \int_{4\pi} I_\nu d\vec{\Omega} - 4\pi B_\nu \right) d\nu \right] dV, \quad (4.142)$$

or on the flux-based expression

$$W_i^r = - \int_{S_i} (\vec{h}_r \cdot \vec{n}) dS = - \int_{S_i} \left[ \int_{4\pi} (\vec{\Omega} \cdot \vec{n}) \left( \int_0^\infty I_\nu d\nu \right) d\vec{\Omega} \right] dS, \quad (4.143)$$

which is obtained by applying the divergence theorem to the second part of Eq. (4.141); here  $S_i$  is the closed outer surface of cell  $i$ , and  $\vec{n}$  is the outward unit normal vector to this surface. Recall that in the 2D geometries considered here all the extensive quantities like volume, mass, energy, etc. are measured either per unit length along the  $z$ -axis (for `iradial` = 0) or per radian of the azimuth angle (for `iradial` = 1, 2); correspondingly, the infinitesimal volume and surface elements are given by

$$dV = \begin{cases} dx_1 dx_2, & \text{iradial} = 0, \\ R dx_1 dx_2, & \text{iradial} = 1, 2, \end{cases} \quad dS = \begin{cases} d\lambda, & \text{iradial} = 0, \\ R d\lambda, & \text{iradial} = 1, 2, \end{cases} \quad (4.144)$$

where  $d\lambda$  is the length element along the perimeter of the mesh cell in the computational  $(x_1, x_2)$  plane.

After we apply to Eq. (4.142) our discretization schemes for the frequency  $\nu$  and photon propagation direction  $\vec{\Omega}$ , we obtain [cf. Eq. (4.28)]

$$W_i^r = 2 \sum_{L=1}^{4N_\Omega} w_L \sum_{[k]} \int_{V_i} k_{[k]} (I_{[k],L} - B_{[k]}) dV. \quad (4.145)$$

The order of integration in Eqs. (4.142) and (4.143), and of summation in Eq. (4.145) is, of course, fully interchangeable. Because the summation over the spectral groups is trivial, we often omit it for brevity together with the corresponding index  $[k]$ .

## 2. Flux-based formulae

For a quadrilateral mesh cell  $i$  the discretized version of the flux-based formula (4.143) can be written as

$$W_i^{rh} = H_{i1}^r + H_{i2}^r - H_{i1+2}^r - H_{i2+1}^r, \quad (4.146)$$

where  $H_{im}^r$  is the total (in  $[\text{erg s}^{-1}]$ ) radiation energy flux across face  $im$  of the quadrilateral mesh — as it is shown in Fig. 4.12. Here we introduced a separate symbol  $W_i^{rh}$  for the cell heating power, obtained from a flux-based formula, because its discretized values will generally differ from the discretized volumetric heating power  $W_i^r$  defined by Eq. (4.145). It might be worth reminding also that in our notation a single index  $i$  combines the two mesh indices  $ij$  along the two mesh directions  $m = 1$  and  $m = 2$  that are usually used for structured quadrilateral grids: node (vertex)  $i$  is the lower-left corner of the quadrilateral  $i$ ; face  $im$  is the edge of the quadrilateral  $i$  starting from vertex  $i$  and extending along the mesh direction  $m$ , i.e. connecting the vertex  $i$  with the neighbor vertex  $i_{m+}$ .

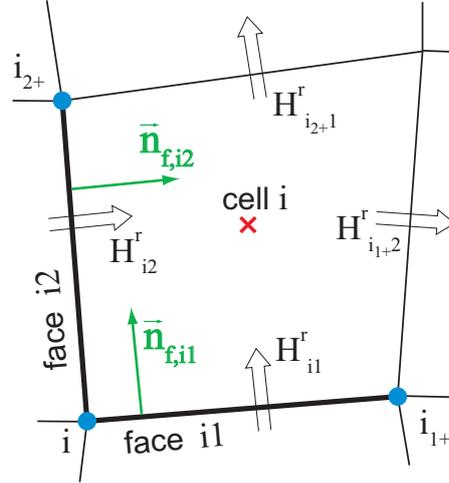


FIG. 4.12: Single-index convention for cell, vertex and face numbering on a structured quadrilateral mesh. Also shown are full radiation energy fluxes  $H_{im}^r$  across the corresponding cell edges.

Generally we have

$$H_{im}^r = \int_{\text{face } im} (\vec{h}_r \cdot \vec{n}_{f,im}) R d\lambda, \quad (4.147)$$

where

$$\vec{h}_r = \int_{4\pi} \vec{\Omega} I(\vec{\Omega}) d\vec{\Omega} \quad (4.148)$$

is the flux density of the radiant energy, and

$$I(\vec{\Omega}) = \int_0^\infty I_\nu(\vec{\Omega}, \nu) d\nu = \sum_{[k]} I_{[k]}(\vec{\Omega}) \quad (4.149)$$

is the *frequency-integrated intensity*,  $\vec{n}_{f,im}$  is the unit normal vector to face  $im$ , pointing to the interior of cell  $i$  (see Fig. 4.12),  $R$  is the cylindrical radius ( $R = 1$  for  $\text{iradial} = 0$ ),  $d\lambda$  is the length element along the face  $im$ . Besides  $\vec{\Omega}$ , all the quantities in Eqs. (4.147) and (4.148) depend also on the spatial location  $\vec{x}$ , which is omitted for brevity.

Note that Eq. (4.147) already assumes that the scalar product  $\vec{\Omega} \cdot \vec{n}_{f,im}$  does not depend on the azimuth angle  $\phi$  of the cylindrical coordinates  $(r, z, \phi)$  in the axisymmetric geometry — which is only possible when the local components  $(\Omega_x, \Omega_y, \Omega_z)$  of vector  $\vec{\Omega}$  with respect to the triad  $\{\vec{n}_z, \vec{n}_r, \vec{n}_\phi\}$  are used as the primary angular variables. In this case the scalar product  $\vec{\Omega} \cdot \vec{n}_{f,im}$  remains constant along any straight segment in the  $(x_1, x_2)$  plane, and one can write

$$H_{im}^r = 2 \sum_{L=1}^{4N_\Omega} w_l \left( \vec{\Omega}_L \cdot \vec{n}_{f,im} \right) \int_{\vec{\lambda}_{im}} I_L R d\lambda, \quad (4.150)$$

where  $\vec{\lambda}_{im}$  is the straight segment connecting the nodes  $i$  and  $i_{m+}$ ; see Fig. 4.13.

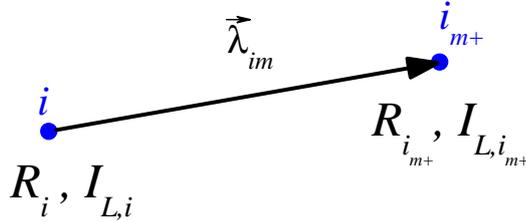


FIG. 4.13: Notation to the integrals along a straight cell edge  $\vec{\lambda}_{im}$  in Eqs. (4.150) and (4.151).

To integrate  $I_L$  along a straight edge  $\vec{\lambda}_{im}$  (see Fig. 4.13) of mesh cell  $i$ , we assume a linear interpolation for the spatial dependence of  $I_L$  between its values  $I_{L,i}$  and  $I_{L,i_{m+}}$  at the corresponding end vertices of the straight segment  $\vec{\lambda}_{im}$ . Because the cylindrical radius  $R$  is always an exact linear function of the edge length  $\lambda$ , varying between  $R_i$  and  $R_{i_{m+}}$ , the integral over  $\lambda$  in Eq. (4.150) can be represented as

$$\int_{\vec{\lambda}_{im}} I_L R d\lambda = \frac{1}{4} \lambda_{im} \left[ (I_{L,i} + I_{L,i_{m+}})(R_i + R_{i_{m+}}) + \frac{1}{3} (I_{L,i} - I_{L,i_{m+}})(R_i - R_{i_{m+}}) \right], \quad (4.151)$$

where  $\lambda_{im} = |\vec{\lambda}_{im}|$  is the length of face  $im$ .

An important advantage of the flux-based formula (4.146) is that in the process of summing up the cell heating powers  $W_i^{rh}$  over all cells  $i$  belonging to a given mesh block  $\text{iblk}$  the fluxes  $H_{im}^r$  across all internal cell faces cancel out, and for the total block heating power we obtain

$$W_{\text{iblk}}^{rh} = \sum_{i \in \text{iblk}} W_i^{rh} = \mathcal{H}_1^r + \mathcal{H}_3^r - \mathcal{H}_2^r - \mathcal{H}_4^r, \quad (4.152)$$

where  $\mathcal{H}_{ib}^r$  is the total radiant energy flux across edge  $\text{ib} = 1, 2, 3, 4$  of block  $\text{iblk}$  given by

$$\mathcal{H}_1^r = \sum_{i \in \text{ib}=1} H_{i1}^r, \quad \mathcal{H}_2^r = \sum_{i \in \text{ib}=2} H_{i1}^r, \quad \mathcal{H}_3^r = \sum_{i \in \text{ib}=3} H_{i2}^r, \quad \mathcal{H}_4^r = \sum_{i \in \text{ib}=4} H_{i2}^r. \quad (4.153)$$

In the RALEF code thus defined flux quantities are used for the global accuracy control of our numerical scheme for energy transport by radiation.

3. Incompatibility of the flux-based formulae with nodal collocation for  $I$ 

Once we choose to assign the discrete intensity values  $I_{[k],L,i}$  to the mesh nodes, we immediately discover that we loose radiation flux conservation in vacuum. Indeed, with  $k_{[k]} = 0$ , the transport equation (4.7) implies  $\vec{\Omega} \cdot \nabla I_{[k]} = \text{div} \left( \vec{\Omega} I_{[k]} \right) = 0$ , and, by virtue of the divergence theorem, for any beam direction  $\vec{\Omega}$  we obtain the law of flux conservation in the form of

$$\int_S \left( \vec{\Omega} \cdot \vec{n} \right) I_{[k]} dS = 0, \quad (4.154)$$

where  $S$  is the outer surface of an arbitrary volume bounded by a certain closed curve in the 2D computational domain, and  $\vec{n}$  is the outward unit normal to this surface. If the discretized version of the transport operator is to obey the conservation law (4.154), then Eq. (4.154) must be satisfied for each individual mesh cell  $i$ , or any group of such cells. The following simple example demonstrates that the latter cannot be ensured in the general case with an algorithm, based on the nodal group intensities  $I_{[k]}$ .

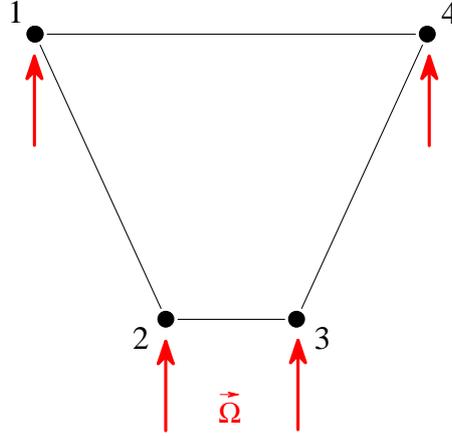


FIG. 4.14: Violation of flux conservation in vacuum.

Consider a single mesh cell shown in Fig. 4.14, where the three edges 1-2, 2-3, and 3-4 all have the same projected horizontal length  $l$  and the radiation propagates along the vertical direction  $\vec{\Omega} = \{0; 1\}$ . Then the group intensity  $I_i$  (here, for simplicity, we omit the spectral and the angular indices  $[k]$  and  $L$ ) at all the four nodes  $i = 1, 2, 3, 4$  is prescribed by the boundary condition. In this case the flux entering through faces 1-2, 2-3, and 3-4 is generally not equal to the flux which exits through the face 1-4. With the linear interpolation along the edges, we calculate the *in* and *out* total fluxes to be

$$H_{in} = \left[ \frac{1}{2}(I_1 + I_2) + \frac{1}{2}(I_2 + I_3) + \frac{1}{2}(I_3 + I_4) \right] l = \left[ \frac{1}{2}(I_1 + I_4) + I_2 + I_3 \right] l,$$

$$H_{out} = \frac{3}{2}(I_1 + I_4)l.$$

From this example we conclude that, for sufficiently small values of the absorption coefficient  $k_\nu$ , the flux-based algorithm (4.146) for evaluating  $W_i^r$  may lead to arbitrarily large

errors (both positive and negative) and, for this reason, must be discarded as unacceptable. Not surprisingly, the alternative volumetric formula (4.145) — when applied in a straightforward manner — usually leads to infinitely large errors in the opposite limit of  $k_\nu \rightarrow \infty$  (see also a discussion of this issue in Ref. [22]). Below we construct a numerical scheme, based on a combined *column-flux* approach, which automatically yields correct values of  $W_i^r$  in both limits of  $k_\nu \rightarrow 0$  and  $k_\nu \rightarrow \infty$ .

#### 4. Column-flux algorithm

Consider a finite 3D volume  $V_i$ . The heating power  $W_i^r$  [erg/s] of this volume due to interaction with the radiation field is given by

$$W_i^r = \int_0^\infty W_i^\nu d\nu, \quad (4.155)$$

where

$$W_i^\nu = \int_{V_i} \left[ k_\nu \int_{4\pi} (I_\nu - B_\nu) d\vec{\Omega} \right] dV = - \int_{S_i} \left[ \int_{4\pi} (\vec{\Omega} \cdot \vec{n}) I_\nu d\vec{\Omega} \right] dS; \quad (4.156)$$

the second integral in Eq. (4.156) is taken over the external surface  $S_i$  of volume  $V_i$ , with  $\vec{n}$  being the unit outward normal to this surface.

Because individual frequencies  $\nu$  in our treatment are independent of one another, and the frequency integral in Eq. (4.155) trivially reduces to a summation over the frequency groups  $[k]$ , we consider below just the spectral heating power  $W_i^\nu$ . Note that, once the surface element  $d\vec{S} = \vec{n} dS$  is fixed, the inner integration in the second part of Eq. (4.156) over the solid angle  $4\pi$  can be performed by using either the local or the global decomposition of the propagation vector  $\vec{\Omega}$ .

The angular integral in Eq. (4.156) can be reduced by half if we consider a shaft of light, entering the volume  $V_i$  across the surface element  $d\vec{S}_E$ , and combine the entrance and the exit intensities. Let a shaft of light with a fixed propagation direction  $\vec{\Omega}$  enter the volume  $V_i$  at a point  $E$  (entrée) across a surface element  $d\vec{S}_E$ , as is shown in Fig. 4.15, and exit this volume at a point  $S$  (sortie) across a surface element  $d\vec{S}_S$ . Because

$$- \left( \vec{\Omega} \cdot \vec{n}_E \right) dS_E = dS_\perp = + \left( \vec{\Omega} \cdot \vec{n}_S \right) dS_S, \quad (4.157)$$

we can combine the incident (entrance) intensity  $I_E = I_\nu(E, \vec{\Omega})$  with the emerging (exit) intensity  $I_S = I_\nu(S, \vec{\Omega})$  and reduce the angular integral in Eq. (4.156) to only the incident

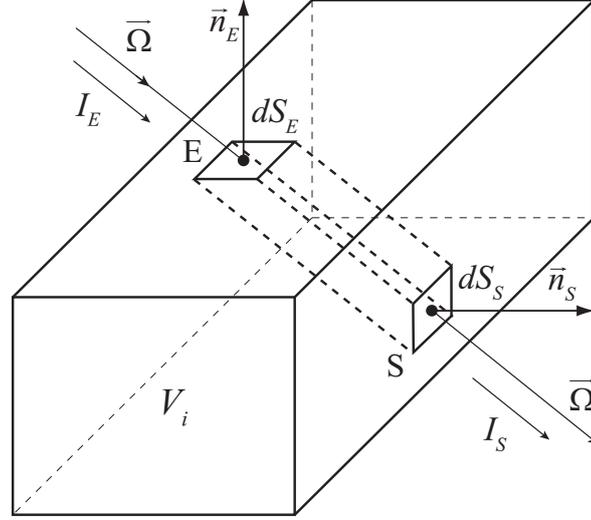


FIG. 4.15: A shaft of light propagating in the direction  $\vec{\Omega}$  enters a finite 3D volume  $V_i$  at point  $E$  and exits it at point  $S$ .

light directions, which have  $\vec{\Omega} \cdot \vec{n}_E < 0$ ,

$$W_i^\nu = \int_{S_i} \left[ \int_{4\pi: \vec{\Omega} \cdot \vec{n}_E < 0} \left( -\vec{\Omega} \cdot \vec{n}_E \right) (I_E - I_S) d\vec{\Omega} \right] dS_E. \quad (4.158)$$

It should be stressed here that in the axisymmetric  $(r, z)$  geometry — as contrasted to the Cartesian  $(x, y)$  geometry — the local entry components  $(\Omega_{E,R}, \Omega_{E,Z})$  of the propagation vector  $\vec{\Omega}$  generally differ from its local exit components  $(\Omega_{S,R}, \Omega_{S,Z})$ .

An additional reduction of the angular integral in Eq. (4.158) can be achieved after we combine two mutually opposite propagation directions  $\vec{\Omega} = \vec{\Omega}^+$  and  $\vec{\Omega}^- = -\vec{\Omega}^+$  for the same light shaft  $\vec{\Omega}$ , as it is shown in Fig. 4.16. If  $E$  and  $S$  are, respectively, the entry and the exit points for the direct beam  $\vec{\Omega}^+$ , then  $S$  is the entry point (and  $E$  is the exit point) for the reverse beam  $\vec{\Omega}^-$ . Having denoted the spectral intensities  $I_\nu$  of the direct beam at points  $E$  and  $S$  as  $I_E^+$  and  $I_S^+$ , and the corresponding intensities of the reverse beam as  $I_E^-$  and  $I_S^-$ , we can rewrite Eq. (4.158) as

$$W_i^\nu = \int_{S_i} \left[ \int_{2\pi: \vec{\Omega}^+ \cdot \vec{n}_E < 0} \left( -\vec{\Omega}^+ \cdot \vec{n}_E \right) \mathcal{I}_\Omega^\nu d\vec{\Omega}^+ \right] dS_E, \quad (4.159)$$

where the quantity

$$\mathcal{I}_\Omega^\nu = I_E^+ - I_S^+ + I_S^- - I_E^- \quad (4.160)$$

may be called the *column heating rate* by a shaft of light  $\vec{\Omega}$  with an infinitely small transverse area  $dS_\perp$ . In Eq. (4.159) the directions  $\vec{\Omega}^+ \in 2\pi$  are picked from only one hemisphere (say, with  $\Omega_Z^+ < 0$  in the  $(r, z)$  geometry) and only such that  $\vec{\Omega}^+ \cdot \vec{n}_E < 0$  for a given  $d\vec{S}_E = \vec{n}_E dS_E$ .

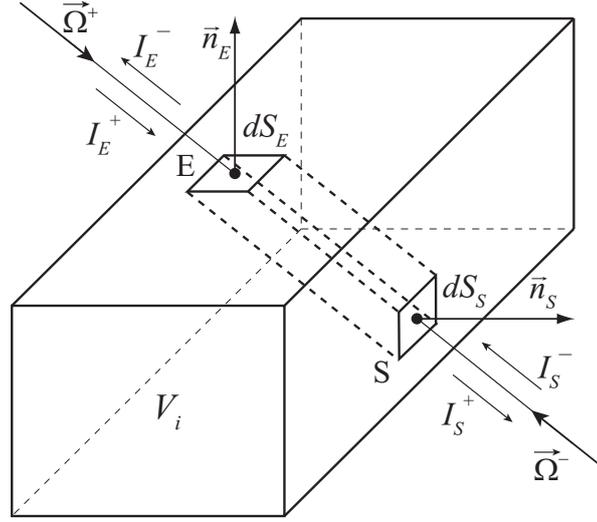


FIG. 4.16: A shaft of light with two opposite propagation directions:  $\vec{\Omega}^+$  for the direct beamlet, and  $\vec{\Omega}^-$  for the reverse beamlet.

In the next subsection we derive expression (4.168) for the column heating rate  $\mathcal{I}_\Omega^\nu$  which, together with Eq. (4.159), provides the basis for our numerical algorithm for calculating the radiative heating power  $W_i^r$  of a cell  $i$ .

### 5. Column heating rate

Consider a radiation beam, which propagates in a direction  $\vec{\Omega} = \vec{\Omega}^+$  along an infinitely narrow column of matter with an entry point  $E$  and an exit point  $S$ , as is shown in Fig. 4.17. Let the spectral intensity of this beam be  $I_\nu^+ = I_\nu^+(\tau)$ , where the spectral optical depth

$$\tau = \int_{s_E}^s k_\nu ds \quad (4.161)$$

is measured from the entry point  $E$  in the positive direction  $\vec{\Omega}^+$ ;  $ds$  is the length element along the light ray  $\vec{\Omega}^+$ . The corresponding reverse beam propagates in the opposite direction  $\vec{\Omega}^- = -\vec{\Omega}^+$  along the same column and has an intensity  $I_\nu^- = I_\nu^-(\tau)$ . Then the transfer equation (2.6) takes the form

$$\frac{dI_\nu^+}{d\tau} = B_\nu - I_\nu^+, \quad \frac{dI_\nu^-}{d\tau} = I_\nu^- - B_\nu. \quad (4.162)$$

Trivial integration of the two equations (4.162) yields

$$\begin{aligned} I_\nu^+(\tau) &= I_E^+ e^{-\tau} + e^{-\tau} \int_0^\tau e^t B_\nu(t) dt, \\ I_\nu^-(\tau) &= I_S^- e^{\tau-\tau_\Omega} + e^\tau \int_\tau^{\tau_\Omega} e^{-t} B_\nu(t) dt, \end{aligned} \quad (4.163)$$

where

$$\tau_{\Omega} = \int_{s_E}^{s_S} k_{\nu} ds \quad (4.164)$$

is the optical distance between the points  $E$  and  $S$ . In Eq. (4.163) we used the initial conditions at the entry point  $E$  for the direct beam  $\vec{\Omega}^+$ , and at the exit point  $S$  for the reverse beam  $\vec{\Omega}^-$ . For the residual intensities  $F_{\nu}^{\pm}(\tau) = I_{\nu}^{\pm}(\tau) - B_{\nu}$  we get

$$\begin{aligned} F_{\nu}^+(\tau) &= F_E^+ e^{-\tau} - e^{-\tau} \int_0^{\tau} e^t B'_{\nu}(t) dt, \\ F_{\nu}^-(\tau) &= F_S^- e^{\tau-\tau_{\Omega}} + e^{\tau} \int_{\tau}^{\tau_{\Omega}} e^{-t} B'_{\nu}(t) dt, \end{aligned} \quad (4.165)$$

where  $B'_{\nu}(\tau) = dB_{\nu}/d\tau$ .

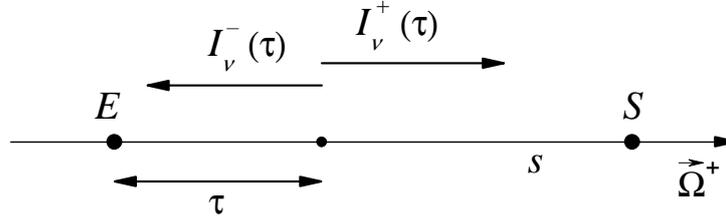


FIG. 4.17: Intensity variation along the positive propagation direction  $\vec{\Omega}^+$  for two mutually opposite beamlets.

Now, having taken the values of  $I_S^+ = I_{\nu}^+(\tau_{\Omega})$  and  $I_E^- = I_{\nu}^-(0)$  from Eq. (4.163), we can substitute them into Eq. (4.160) to obtain

$$\begin{aligned} \mathcal{I}_{\Omega}^{\nu} &= (I_E^+ + I_S^-) (1 - e^{-\tau_{\Omega}}) - \int_0^{\tau_{\Omega}} (e^{t-\tau_{\Omega}} + e^{-t}) B(t) dt \\ &= (F_E^+ + F_S^-) (1 - e^{-\tau_{\Omega}}) + \int_0^{\tau_{\Omega}} (e^{t-\tau_{\Omega}} - e^{-t}) B'(t) dt. \end{aligned} \quad (4.166)$$

Formula (4.166) contains only the boundary values  $I_E^+$  and  $I_S^-$  of the radiation intensity  $I_{\nu}^{\pm}$  but requires the functional dependence of the source function  $B_{\nu}$  on the position inside cell  $i$ . The lowest-order interpolation to the dependence  $B_{\nu}(\tau)$  that ensures correct transition to the diffusion limit is a parabolic one. If we use the second part of Eq. (4.166), the parabolic interpolation of  $B_{\nu}(\tau)$  becomes a linear interpolation for  $B'_{\nu}(\tau)$ , which can be written in the form

$$B'(\tau) = B'_E + (B'_S - B'_E) \left( \frac{\tau}{\tau_{\Omega}} \right). \quad (4.167)$$

Having substituted Eq. (4.167) into the second part of Eq. (4.166), we arrive at

$$\mathcal{I}_{\Omega}^{\nu} = (F_E^+ + F_S^-) (1 - e^{-\tau_{\Omega}}) + \frac{B'_S - B'_E}{\tau_{\Omega}} \phi_3(\tau_{\Omega}), \quad (4.168)$$

where

$$\phi_3(x) = x - 2 + (x + 2)e^{-x} = \begin{cases} \frac{x^3}{6} \left( 1 - \frac{x}{2} + \frac{3x^2}{20} - \frac{x^3}{30} + \dots \right), & x \ll 1, \\ x - 2, & x \gg 1. \end{cases} \quad (4.169)$$

Formula (4.168) plays a central role in our numerical scheme for calculating the radiative heating power  $W_i^r$ : it contains only the boundary values of the residual intensity  $F_\nu$  and of the directional  $\tau$ -derivative  $B'_\nu$  of the source function  $B_\nu$ . As is shown below, it automatically ensures correct transition to the diffusion limit provided that a parabolic interpolation is used for spatial variation of  $B_\nu$  when calculating the values of  $F_E^+$  and  $F_S^-$ .

### 6. The diffusion limit

The diffusion limit is obtained when the source function  $B_\nu$  varies on a spatial scale much larger than the mean free path  $k_\nu^{-1}$ . The latter means that the condition

$$|B_\nu''''| \ll |B_\nu''| \ll |B_\nu'| \ll B_\nu \quad (4.170)$$

will be satisfied along the column considered in the previous subsection; here each additional prime denotes the next derivative with respect to the spectral optical depth  $\tau$ . The rigorous expression for the column heating rate in the diffusion approximation, which is the exact equivalent of the more familiar formula (2.23), is given by

$$\mathcal{I}'_{\Omega, diff} = 2(B'_S - B'_E). \quad (4.171)$$

Note that in the diffusion limit this expression is considered to be exact independent of whether the optical distance  $\tau_\Omega$  between the entry and exit points  $E$  and  $S$  is small or large compared to unity. The latter means that, even if  $\tau_\Omega \gg 1$ , an appropriate finite difference scheme must ensure that the next-order terms (proportional to  $B_E''''$  and higher derivatives) in the Taylor expansion

$$2(B'_S - B'_E) = 2B_E''\tau_\Omega + B_E'''\tau_\Omega^2 + B_E^{IV}\tau_\Omega^2 + \dots \quad (4.172)$$

are small compared to the leading term  $2B_E''\tau_\Omega$ , i.e.

$$|B_E'''\tau_\Omega| \ll |B_E''|, \quad |B_E^{IV}\tau_\Omega^2| \ll |B_E''| \quad (4.173)$$

(otherwise there will be no convergence to the exact solution of the corresponding diffusion equation).

In the diffusion limit the exact solution (4.163) of the transfer equation can be expanded as

$$\begin{aligned} I_\nu^+(\tau) &= B_\nu(\tau) - B'_\nu(\tau) + B''_\nu(\tau) + \dots, \\ I_\nu^-(\tau) &= B_\nu(\tau) + B'_\nu(\tau) + B''_\nu(\tau) + \dots, \end{aligned} \quad (4.174)$$

or

$$\begin{aligned} F_E^+ &= -B'_E + B''_E + \dots, \\ F_S^- &= +B'_S + B''_S + \dots \end{aligned} \quad (4.175)$$

Having substituted this expansion into our basic formula (4.168), we obtain

$$\begin{aligned} \mathcal{I}'_{\Omega} &= 2(B'_S - B'_E) + (1 - e^{-\tau_{\Omega}}) \left[ B''_E + B''_S - 2 \frac{B'_S - B'_E}{\tau_{\Omega}} \right] \\ &= 2(B'_S - B'_E) + (1 - e^{-\tau_{\Omega}}) \left[ \frac{1}{3} B''_E \tau_{\Omega}^2 + \dots \right]. \end{aligned} \quad (4.176)$$

By virtue of condition (4.173), this result is equivalent to the original expression (4.171) for the diffusion-limit column heating  $\mathcal{I}'_{\Omega, diff}$  at any value of  $\tau_{\Omega}$ . As a consequence, it may be concluded that once the numerical algorithm for solving the transfer equation ensures the correct diffusion-limit expansion (4.174) for radiation intensities down to the  $B''_{\nu}$  terms, Eq. (4.168) automatically guarantees correct energy coupling between the fluid and the radiation for any value of the column optical thickness  $\tau_{\Omega}$ . Of course, the finite-difference formulae for  $B'_E$  and  $B'_S$  in Eq. (4.168) must capture the correct variation of  $B_{\nu}(\tau)$  down to the  $B''_{\nu}$  terms as well. To fulfill the last two requirements, one has (i) to use a parabolic (accurate down to the  $B''_{\nu}$  terms) interpolation for  $B_{\nu}(\tau)$  along every treated  $S_n$  propagation direction  $\vec{\Omega}$  — which, in addition, must be the same for the two opposite directions  $\vec{\Omega}^+$  and  $\vec{\Omega}^-$ , and (ii) to use the exact integrals (4.165) to calculate the residual intensities  $F_{\nu}$  for a given  $B_{\nu}(\tau)$ .

### 7. Reduction to the 2D case

Here we adapt our basic 3D formula (4.159) to the 2D Cartesian and axisymmetric geometries. First of all, we replace the angular integral at each surface element  $d\vec{S}_E$  by a discrete sum

$$\int_{4\pi} f(\vec{\Omega}) d\vec{\Omega} = \sum_{L=1}^{4N_{\Omega}} 2w_L f(\vec{\Omega}_L) \quad (4.177)$$

over the  $S_n$  beamlets according to the general rule (4.28). Here we assume that in the axisymmetric case for a fixed  $L$  every beamlet  $\vec{\Omega}_L$ , selected from a given  $S_n$  set, is specified by the same pair of local components  $(\Omega_{RL}, \Omega_{ZL})$  at each entry point  $E$ ; its local exit components at point  $S$ , however, vary from point to point and are generally different from  $(\Omega_{RL}, \Omega_{ZL})$ . In contrast, in the Cartesian case every beamlet  $\vec{\Omega}_L$  has the same local components  $(\Omega_{xL}, \Omega_{yL})$  at both the entry and the exit points  $E$  and  $S$ .

The surface element  $d\vec{S}_E$  in two dimensions becomes

$$\vec{n}_E dS_E = \vec{n}_{\lambda} R d\lambda, \quad (4.178)$$

where  $R$  is the cylindrical radius ( $R = 1$  in the Cartesian case),  $d\lambda$  is the length element along the boundary contour  $S_i$  of cell  $i$ , and  $\vec{n}_{\lambda}$  is the local 2D outward normal to  $S_i$ ; see Fig. 4.18. Interchanging the order of integration along  $S_i$  and of summation over  $L$ , we transform Eq. (4.159) to

$$W_i^{\nu} = \sum_{L: \Omega_{xmL} < 0} 2w_L \int_{S_{i, \Omega_L}^+} \left( -\vec{\Omega}_L \cdot \vec{n}_{\lambda} \right) \mathcal{I}'_{\Omega_L} R d\lambda, \quad (4.179)$$

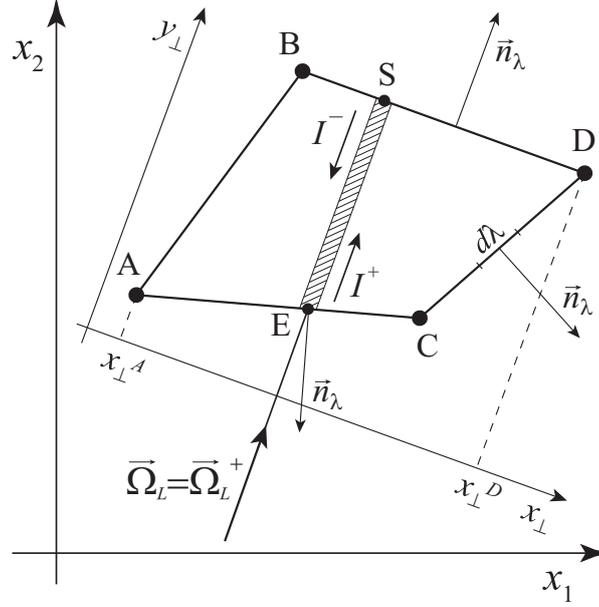


FIG. 4.18: General illustration to calculation of the cell heating power  $W_i^\nu$  in two dimensions.

where the summation is performed over one hemisphere either with  $\Omega_{x_1L} < 0$  or with  $\Omega_{x_2L} < 0$ , and the integration is carried out only along the “illuminated” part  $S_{i,\Omega_L}^+$  of the boundary contour where  $\vec{\Omega}_L \cdot \vec{n}_\lambda < 0$ .

Next, we can rewrite the product  $(-\vec{\Omega}_L \cdot \vec{n}_\lambda) d\lambda$  as

$$(-\vec{\Omega}_L \cdot \vec{n}_\lambda) d\lambda = \Omega_{pL} dx_\perp, \quad (4.180)$$

where

$$\Omega_{pL} = (\Omega_{x_1L}^2 + \Omega_{x_2L}^2)^{1/2}, \quad (4.181)$$

and new orthogonal coordinates  $(x_\perp, y_\perp)$  are obtained from  $(x_1, x_2)$  by a rotation

$$\begin{aligned} x_\perp &= (x_1\Omega_{x_2L} - x_2\Omega_{x_1L}) / \Omega_{pL}, \\ y_\perp &= (x_1\Omega_{x_1L} + x_2\Omega_{x_2L}) / \Omega_{pL}, \end{aligned} \quad (4.182)$$

such that the  $y_\perp$ -axis is along the 2D vector  $(\Omega_{x_1L}, \Omega_{x_2L})$ ; see Fig. 4.18. As a result, Eq. (4.179) can be rewritten as

$$W_i^\nu = \sum_{L: \Omega_{x_mL} < 0} 2w_l \Omega_{pL} \int_{x_\perp^A}^{x_\perp^D} \mathcal{I}_{\Omega_L}^\nu R dx_\perp. \quad (4.183)$$

Here integration is performed over the interval  $[x_\perp^A, x_\perp^D]$  made up by the projection of cell  $i$  onto the  $x_\perp$ -axis, the entry point  $E$  slides from the left-most vertex  $A$  to the right-most vertex  $D$  along the lower branch of the boundary contour with respect to the rotated coordinates  $(x_\perp, y_\perp)$ ; in the axisymmetric geometry the radius  $R$  is measured at the entry point  $E$ , and

the propagation vector  $\vec{\Omega}_L$  has the same fixed local components  $(\Omega_{x_1L}, \Omega_{x_2L})$  also at point  $E$ , i.e. along the lower branch of the boundary contour in the  $(x_\perp, y_\perp)$  coordinate system.

...

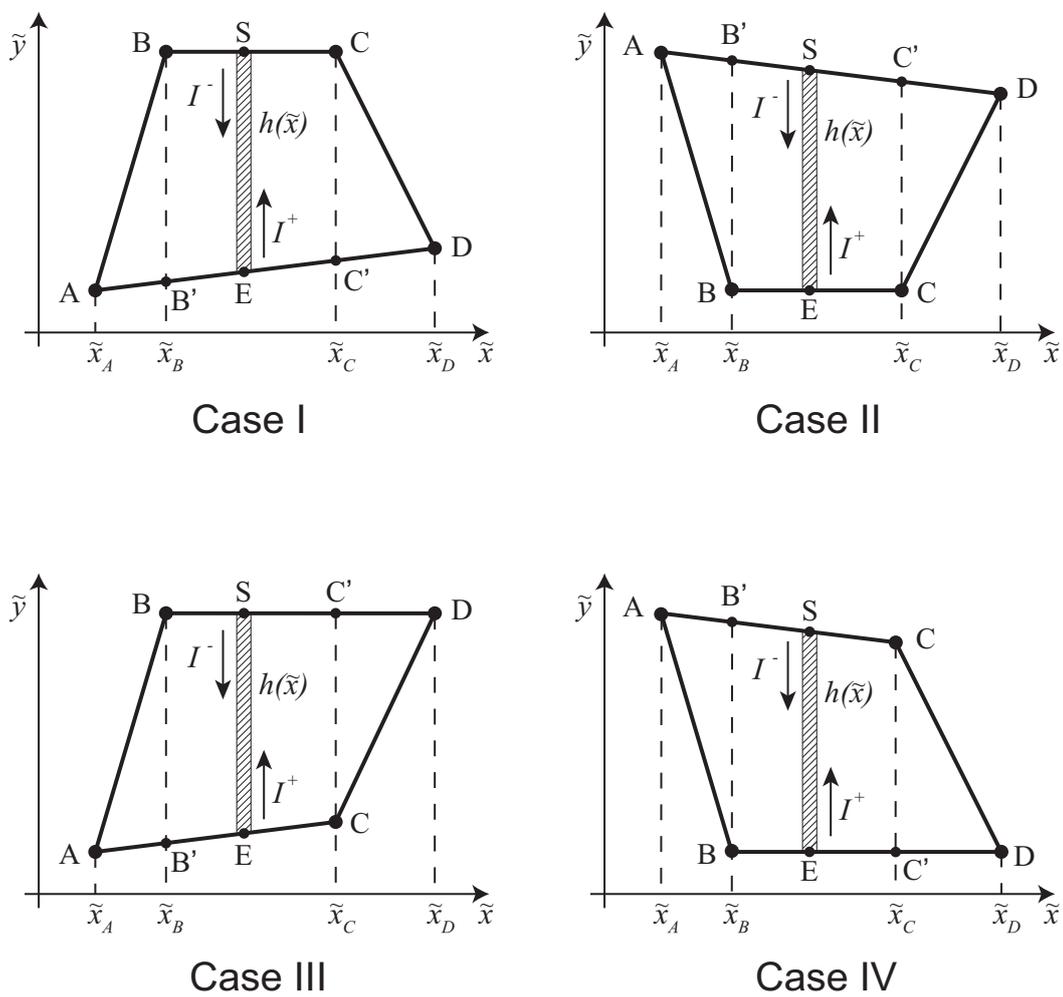


FIG. 4.19: Four different orientation cases for a convex cell  $i$  with respect to the rotated coordinates  $(\tilde{x}, \tilde{y})$ .

---

*Correspondence with the code variables:*

$\mathcal{H}_k^r =$	HRADBCIB(k,iblk)	total radiation-energy flux across edge $\mathbf{ib} = k$ (= 1 or 3) into block $\mathbf{iblk}$ , block-edge-associated;
$-\mathcal{H}_k^r =$	HRADBCIB(k,iblk)	total radiation-energy flux across edge $\mathbf{ib} = k$ (= 2 or 4) into block $\mathbf{iblk}$ , block-edge-associated;

---

## 5. Boundary conditions for radiation transfer

### 1. Allowed types of global reflective boundaries

A possibility to introduce reflective boundaries can be an important advantage for efficient simulation of many problems. Here and below a *reflective boundary* is understood as a rigid planar wall, where the normal components of the fluid velocity and thermal flux are zero, and which reflects the thermal (and laser) radiation as a perfect mirror. Because solution of the transfer equation (2.6) for thermal radiation in the quasi-steady approximation is an inherently global boundary-value problem, we have to introduce some additional (with regard to pure hydro) constraints on possible types of such boundaries.

When radiation transport is included into simulation, we restrict our treatment to only three possible cases

- (i) there are no reflective boundaries;
- (ii) there is only one straight global reflective boundary either along the global  $x_1$ -axis (an **rb1** boundary along the  $x_2 = 0$  coordinate line), or along the global  $x_2$ -axis (an **rb2** boundary along the  $x_1 = 0$  coordinate line);
- (iii) there are two mutually perpendicular straight global reflective boundaries **rb1** and **rb2** which coincide, respectively, with the global  $x_1$  and  $x_2$  axes; below, such reflective boundaries, coinciding with global coordinate axes  $x_1$  or/and  $x_2$ , will be called *coaxial reflective boundaries*.

The latter means, for example, that two different reflective boundaries, which run parallel to one another, are not allowed for radiation transport because a light ray could, in principle, undergo an arbitrarily large number of reflections before escaping from the computational domain. Nevertheless, together with coaxial reflective boundaries, the RALEF code does accept non-coaxial reflective boundaries as well: in such a case only the coaxial boundaries are treated as truly reflective for thermal radiation and laser beams; the non-coaxial ones, which lie off the two coordinate axes, are considered to be reflective for hydrodynamics and thermal conduction but transparent for thermal radiation and laser light (here, of course, caution is advised when interpreting the results!).

Different situations with zero, one, or two coaxial reflective boundaries are distinguished by the value of parameter **irflty1**:

**irflty1** = 0  $\Rightarrow$  there are no reflective boundaries (for radiation);

**irflty1** = 1  $\Rightarrow$  there is one reflective boundary along the  $x_1$ -axis;

**irflty1** = 2  $\Rightarrow$  there is one reflective boundary along the  $x_2$ -axis;

`irflty1 = 3`  $\Rightarrow$  there are two mutually perpendicular reflective boundaries along both the  $x_1$ - and  $x_2$ -axes.

Special consideration must be given to the rotation axis in the  $rz$ -geometry. In what concerns the numerical scheme for hydrodynamics and thermal conduction, the rotation axis is fully equivalent to a reflective boundary. This, however, is not the case for the equation of radiation transfer: in the  $rz$ -geometry the photon path is calculated in the 3D space, where the rotation axis is not a boundary at all. As a consequence, when constructing the *extended mesh* to account for the shadowing effects, the original mesh must not be reflected across the rotation axis. Thus, depending on a particular segment of the code, the rotation axis must, or must not be treated as a reflective boundary.

To retain the possibility to either count or not count (depending on the context) the rotation axis as a reflective boundary, we introduce a second parameter `irflty0` to characterize the relevant type of reflective symmetry, whose value — as distinguished from `irflty1` — is calculated under the assumption that the rotation axis is just another global reflective boundary (either `rb1` or `rb2`); in other words,

`irflty0 = 0`  $\Rightarrow$  there are no reflective boundaries and `iradial=0` (no rotation axes);

`irflty0 = 1`  $\Rightarrow$  there is either one reflective boundary along the  $x_1$ -axis and `iradial=0`, or no reflective boundaries but a rotation axis along  $x_1$  for `iradial=2`;

`irflty0 = 2`  $\Rightarrow$  there is either one reflective boundary along the  $x_2$ -axis and `iradial=0`, or no reflective boundaries but a rotation axis along  $x_2$  for `iradial=1`;

`irflty0 = 3`  $\Rightarrow$  there are either two mutually perpendicular reflective boundaries along both the  $x_1$ - and  $x_2$ -axes and `iradial=0`, or one reflective boundary perpendicular to the rotation axis for either `iradial=1` or `iradial=2`.

Thus, parameter `irflty0` may take any integer value between 0 and 3 for any allowed value of `iradial = 0, 1, 2`. In contrast, possible values of `irflty1` are restricted to

for `iradial=0`:

$$\text{irflty1} = \begin{cases} 0, & \text{no reflective boundaries,} \\ 1, & \text{one reflective boundary along } x_1, \\ 2, & \text{one reflective boundary along } x_2, \\ 3, & \text{two reflective boundaries along } x_1 \text{ and } x_2; \end{cases} \quad (4.184)$$

for `iradial=1`:

$$\text{irflty1} = \begin{cases} 0, & \text{no reflective boundaries,} \\ 1, & \text{one reflective boundary along } x_1; \end{cases} \quad (4.185)$$

for `iradial=2`:

$$\text{irflty1} = \begin{cases} 0, & \text{no reflective boundaries,} \\ 2, & \text{one reflective boundary along } x_2. \end{cases} \quad (4.186)$$

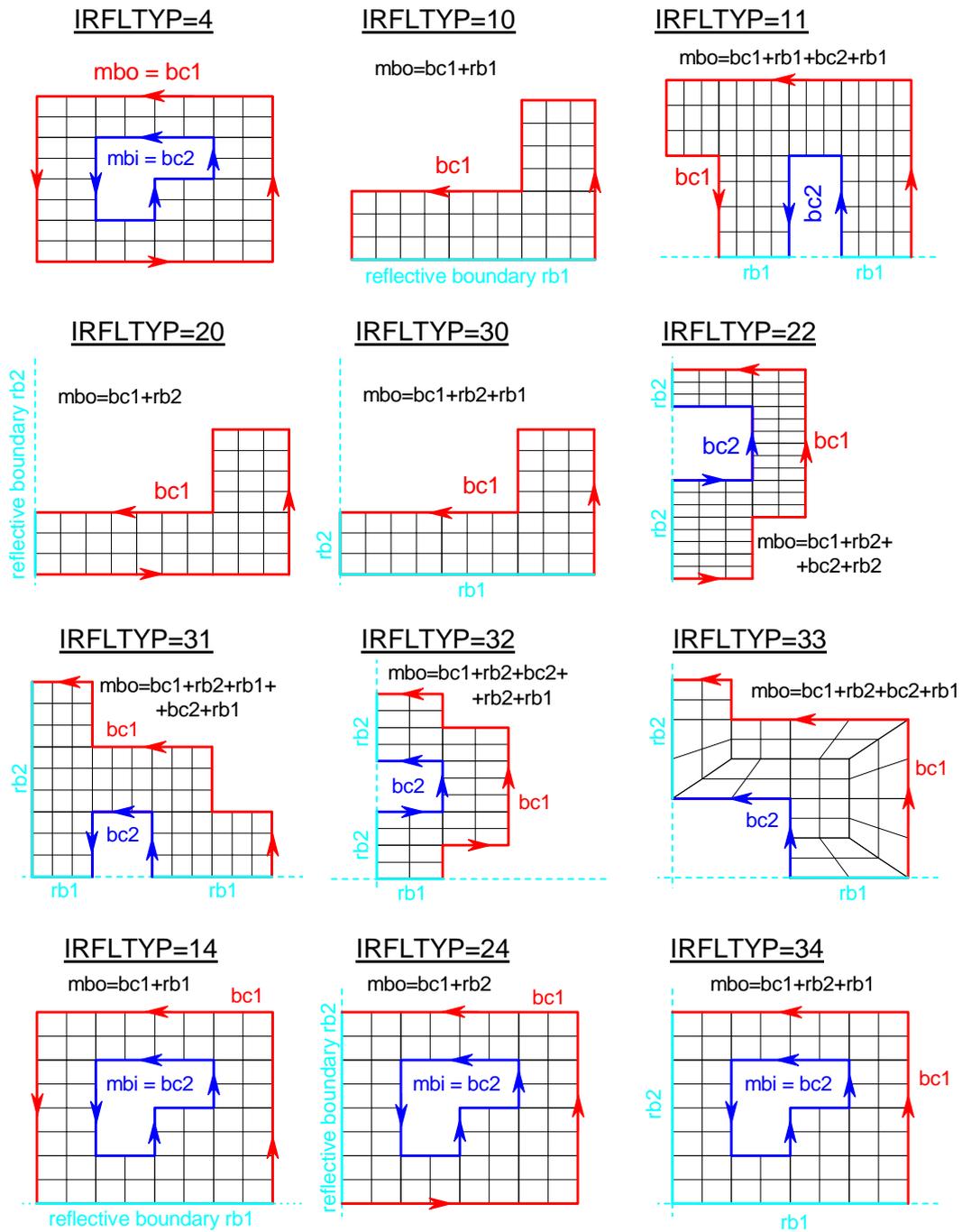


FIG. 4.20: Different types of mesh boundary contours for  $iradial=0$ .

Clearly, for  $iradial = 0$  we always have  $irflty1 = irflty0$ .

With reflective boundaries, not all the segments of the closed contour  $mbo$  of the outer boundary of the original mesh correspond to the physical boundary of the simulated configuration. Also, we want to have a possibility to introduce at least one inner vacuum cavity. Under such assumptions we can distinguish between four different types of segments of the original outer mesh boundary  $mbo$  (see Fig. 4.20):

- rb1** → a reflective segment lying on the  $x_1$ -axis, which is either a global reflective boundary or a rotation axis;
- rb2** → a reflective segment lying on the  $x_2$ -axis, which is either a global reflective boundary or a rotation axis;
- bc1** → the true outer physical boundary of the simulated configuration;
- bc2** → the true inner physical boundary of a vacuum cavity inside the simulated configuration.

Evidently, the **bc1** (**bc2**) segment generally represents only part of the full outer (inner) physical boundary. To obtain the full physical boundary, we have to construct the *extended mesh* composed of the original mesh plus its corresponding mirror images across the reflective boundaries (but not across the rotation axis). Since we only allow a single vacuum cavity, in cases with global reflective boundaries a closed outer boundary contour **mbo** generally consists of reflective segments **rb1** and **rb2** plus a segment **bc1** (part of the *outer extended physical boundary*) plus a segment **bc2** (part of the *inner extended physical boundary*). A closed inner mesh-boundary contour **mbi** can contain neither the **rb1** nor the **rb2** segments, and always coincides with the inner physical boundary **bc2**. Different combinations of reflective and non-reflective segments, making up the outer mesh-boundary contour **mbo**, plus a possible inner mesh-boundary contour **mbi** are illustrated in Fig. 4.20.

Our system of notation for different types of boundary contours can be summarized as follows:

- the entire *mesh boundary* may consist of at most two contours — a closed *outer mesh contour* **mbo** and, possibly, a closed *inner mesh contour* **mbi**;
- the physical part of the mesh boundary may consist of at most two contours — an *outer physical contour* **bc1** (which may be either open or closed) and, possibly, an *inner physical contour* **bc2** (which also may be either open or closed);
- when there are no reflective boundaries and no rotation axes ( $\text{irflty0} = 0$ ), the contour **bc1** is closed and coincides with **mbo**; otherwise, the contour **bc1** is open and constitutes only part of the closed contour **mbo**: it begins and ends at either a **rb1** or a **rb2** segment;
- when there are no reflective boundaries and no rotation axes ( $\text{irflty0} = 0$ ), the contour **bc2** is closed and coincides with **mbi**; otherwise, the contour **bc2** may be either open (then it begins and ends at either a **rb1** or a **rb2** segment; in such a case contour **mbi** is not allowed) or closed (then it does not touch any of the two possible symmetry axes and coincides with the contour **mbi**);

To distinguish between different possibilities for location of the inner physical boundary contour **bc2**, a single-digit parameter **irflty2** is introduced:

- irflty2=0** ⇒ there is no inner vacuum cavity, and the inner physical contour **bc2** is absent;
- irflty2=1** ⇒ the inner physical contour **bc2** is attached with both its ends to the **rb1** part of the outer mesh boundary **mbo**;

- `irflty2=2`  $\Rightarrow$  the inner physical contour `bc2` is attached with both its ends to the `rb2` part of the outer mesh boundary `mbo`;
- `irflty2=3`  $\Rightarrow$  the inner physical contour `bc2` is attached with one its end to the `rb1` part, and with the other to the `rb2` part of the outer mesh boundary `mbo`;
- `irflty2=4`  $\Rightarrow$  the inner physical contour `bc2` is detached from both the  $x_1$ - and the  $x_2$ -axis and coincides with the closed inner boundary contour `mbi`.

The two single-digit parameters `irflty1` and `irflty2` are combined and stored as a single variable `irfltyp` with two decimal digits defined as

$$\text{irfltyp} = 10 \times \text{irflty1} + \text{irflty2}. \quad (4.187)$$

For each problem the values of parameters `irflty0`, `irflty1`, `irflty2` and `irfltyp` are automatically calculated in the radiation-initialization subroutine `RADSET`.

In the RALEF code (as inherited from CAVEAT) different types of boundary conditions are prescribed by loading the corresponding values of the user-defined array `ibc(ib,iblk)` — for each edge `ib` of every block `iblk`. The reflective boundary condition is set by assigning `ibc(ib,iblk) = 1` for the corresponding block edges. In the  $rz$ -geometry one must also set `ibc(ib,iblk) = 1` for every block edge that lies on the rotation axis. When the initial mesh configuration is analyzed in the subroutine `RADSET` and at least one block edge, lying on the coordinate axis  $x_1$  (or  $x_2$ ), is found to have `ibc(ib,iblk) = 1`, this axis as a whole is identified as a global reflective boundary — unless it has already been defined as a rotation axis.

The ensemble of all block edges having `ibc(ib,iblk) = 1` and lying on the  $x_1$ -axis (irrespective of whether it is a true reflective boundary or a rotation axis) makes up the `rb1` segment of the boundary contour `mbo`. On the total, there are  $k = 1, 2, \dots, \text{nibrfl1}$  such edges, with `iblkrfl1(k)` and `ibrfl1(k)` being, respectively, the edge number `ib` and the block number `iblk` for the  $k$ -th such edge; all the `nibrfl1` `rb1`-edges are ordered by the increasing values of the  $x_1$ -coordinates of their ends. Similarly, the ensemble of all block edges having `ibc(ib,iblk) = 1` and lying on the  $x_2$ -axis (irrespective of whether it is a true reflective boundary or a rotation axis) makes up the `rb2` segment of the boundary contour `mbo`. On the total, there are  $k = 1, 2, \dots, \text{nibrfl2}$  such edges, with `iblkrfl2(k)` and `ibrfl2(k)` being, respectively, the edge number `ib` and the block number `iblk` for the  $k$ -th such edge; all the `nibrfl2` `rb2`-edges are ordered by the increasing values of the  $x_2$ -coordinates of their ends.

Logical arrays `loredmbo(ib,iblk)`, `loredrb1(ib,iblk)`, `loredrb2(ib,iblk)`, `loredbc1(ib,iblk)`, `loredbc2(ib,iblk)`, set equal to `.true.` (in subroutine `RADSET`) for edges belonging to the corresponding boundary segments, enable quick access to the vertices and cells along these segments. The logical array `loredaxi(ib,iblk)`, set equal to `.true.` (in subroutine `RADSET`) for edges lying on the rotation axis, enables quick access to the part of the boundary along the rotation axis.

All the `nbndc1` physical vertices along the `bc1` boundary segment can be readily accessed via their  $j$ -number, which runs through  $j = 1, 2, \dots, \text{nbndc1}$ . The `nbndc2` physical vertices along the `bc2` boundary can be accessed via their global  $I$ -numbers, listed in the array `ibndc2(1:mszmb0)`. In addition, for every `bc2` vertex  $I$  the logical flag `lorbnd2(I)` is set to be `.true.`

2. Processing order for angular directions in the Cartesian case  $iradial = 0$

When there are no coaxial reflective boundaries, all angular directions  $L = (i_{ox}, i_{oy}, l)$  in the Cartesian geometry are independent of one another and can be processed in an arbitrary order. However, this is not the case when one or two coaxial reflective boundaries are present because the boundary conditions for a reflected beamlet at the **rb1** and **rb2** pieces of the outer mesh boundary **mbo** become available only after the corresponding incident beamlet has been calculated. Hence the order, in which the conjugate beamlets from different octants should be treated, is specific to a particular case of the reflective symmetry.

For the allowed types of the reflective symmetry defined above, there always exist one or two octants, whose beamlets do not require boundary conditions along the **rb1** and **rb2** pieces of the **mbo** contour and, for this reason, may be called the *primary* (or *incident*) beamlets. The primary beamlets correspond to the values of

$$(i_{ox}, i_{oy}) = \begin{cases} (-1, -1), (+1, -1), & \text{for } irflty0 = irflty1 = 1, \\ (-1, -1), (-1, +1), & \text{for } irflty0 = irflty1 = 2, \\ (-1, -1), & \text{for } irflty0 = irflty1 = 3, \end{cases} \quad (4.188)$$

for any  $l = 1, 2, \dots, N_\Omega$ . Correspondingly, the beamlets defined by

$$(i_{ox}, i_{oy}) = \begin{cases} (-1, +1), (+1, +1), & \text{for } irflty0 = 1, \\ (+1, -1), (+1, +1), & \text{for } irflty0 = 2, \\ (-1, +1), (+1, -1), (+1, +1), & \text{for } irflty0 = 3, \end{cases} \quad (4.189)$$

may be called the *secondary* (or *reflected*). Thus, when  $irflty0 = 1$  or  $2$  (only one symmetry axis), for every primary beamlet there is one dependent (conjugate) secondary beamlet, which has the same value of  $l$  but the opposite value of either  $i_{oy}$  (for  $irflty0 = 1$ ) or  $i_{ox}$  (for  $irflty0 = 2$ ). Analogously, for  $irflty0 = 3$  (two symmetry axes) every primary beamlet has three dependents — three conjugate secondary beamlets with the same value of  $l$  but opposite values of either  $i_{ox}$ , or  $i_{oy}$ , or both. As a consequence, angular directions must be processed in the following order: first we choose (arbitrarily) and calculate a primary beamlet, store the values of radiation intensities along the boundary contour **mbo**, and then calculate the conjugate secondary beamlet by using the stored boundary intensities from the primary beamlet — under the condition that we have only one symmetry axis, i.e. for  $irflty0 = 1$  or  $2$ .

With two symmetry axes ( $irflty0 = 3$ ) the calculation is somewhat more involved. First, we calculate a primary beamlet  $(i_{ox}, i_{oy}) = (-1, -1)$  and store the values of radiation intensities along the boundary contour **mbo** (an **rbnd0** store). Then we calculate the first reflected beamlet  $(i_{ox}, i_{oy}) = (-1, +1)$  by using the **rbnd0** store where needed in the boundary condition, and again store the values of radiation intensities along the boundary contour **mbo** (an **rbnd1** store). Then we calculate the second reflected beamlet  $(i_{ox}, i_{oy}) = (+1, -1)$  by using the **rbnd0** store where needed in the boundary condition, and once more store the values of radiation intensities along the boundary contour **mbo** (an **rbnd2** store). Finally, we calculate the third reflected beamlet  $(i_{ox}, i_{oy}) = (+1, +1)$  by using either the **rbnd1** store (above the ray  $\vec{x} = s\vec{\Omega}_L$ ) or the **rbnd2** store (below the ray  $\vec{x} = s\vec{\Omega}_L$ ) where needed in the boundary condition.

In conclusion, the order in which the four relevant octants are processed for any given value of index  $l$ , is summarized in Table 4.8.

TABLE 4.8: Basic order of octant processing in Cartesian geometry.

io	beam	irflty0 = 0		irflty0 = 1 or 3		irflty0 = 2	
		$(i_{ox}, i_{oy})$	$\vec{\Omega}_l$	$(i_{ox}, i_{oy})$	$\vec{\Omega}_l$	$(i_{ox}, i_{oy})$	$\vec{\Omega}_l$
1	incident	(-1, -1)	↙	(-1, -1)	↙	(-1, -1)	↙
2	reflected	(-1, +1)	↖	(-1, +1)	↖	(+1, -1)	↘
3	opposite-reflected	(+1, -1)	↘	(+1, -1)	↘	(-1, +1)	↖
4	opposite-incident	(+1, +1)	↗	(+1, +1)	↗	(+1, +1)	↗

### 3. Physical boundary conditions

The equation (2.6) of radiation transfer requires a separate boundary condition for the radiation intensity  $I_\nu = I_\nu(t, \vec{x}, \nu, \vec{\Omega})$  — in addition to the usual boundary conditions for the hydrodynamic equations (2.1)–(2.3) with thermal conduction. Since we use a quasi-stationary transport equation, no initial condition is needed for  $I_\nu(t, \vec{x}, \nu, \vec{\Omega})$ . To distinguish among several typical options for the radiative boundary condition, a separate flag IRADFBC is introduced in the RALEF code (see the next subsection).

Once the source function  $B_\nu = B_\nu(\nu, T)$  is assumed to be Planckian, i.e. is uniquely determined by matter temperature  $T$ , the above defined set of boundary conditions is fully sufficient from the viewpoint of differential equations (2.1)–(2.3), (2.6). However, from the viewpoint of discretized equations, an additional separate boundary condition for the source function  $B_\nu(T)$  might be justified. This becomes the case, for example, when mesh cells near the boundary with vacuum have large optical thicknesses. In such a case radiative cooling of the fluid near the boundary is quite sensitive to the temperature gradient across the boundary cells: radiative cooling occurs essentially from the surface, i.e. is given by  $\sigma_{SB}T_b^4$ , where  $T_b$  is the effective surface temperature. And if the transverse (perpendicular to the surface) optical thickness of the boundary mesh cell is sufficiently large, the effective surface temperature  $T_b$  may be much lower than the bulk cell temperature  $T_i$  (associated with the cell center), which characterizes the bulk heat content and enters the discretized hydrodynamics and heat conduction equations.

Because in our numerical scheme the radiation intensity  $I_\nu$  and the source function  $B_\nu$  are allocated to mesh nodes, we may significantly improve numerical accuracy of our algorithm by adopting an additional and formally independent boundary condition for the source function  $B_\nu$ . More precisely, we assume that the Planckian source function is a function of a separate  $B$ -temperature,

$$B_\nu = B_\nu(\nu, T_B), \tag{4.190}$$

which may, in principle, differ from the matter temperature  $T$ . Then we assume that the cell-centered values of the  $B$ -temperature,  $T_{Bc,i}$ , are equal to the cell-centered values  $T_i$  of matter temperature in all physical mesh cells. Because the discretized transport equation contains vertex-centered (nodal) values  $T_{Bv,i}$  of the  $B$ -temperature, we use a special algorithm to calculate  $T_{Bv,i}$  from the cell-centered values  $T_{Bc,i} = T_i$  (in physical cells) with a special boundary condition for the  $B$ -temperature  $T_B$ . Typical options for this boundary condition are controlled by another separate flag IRADBEC. Hence, we allow two independent boundary conditions for the radiation transport in the RALEF code — one for the radiation intensity  $I_\nu$  and the other for the  $B$ -temperature  $T_B$  — that are controlled by two independent flags

IRADFBC and IRADBBC. A certain caution should be exercised when assigning the values of these two flags to avoid possible unphysical combinations of the two boundary conditions.

#### 4. Boundary conditions for radiation intensity

Because group intensities  $I_{[k],L,i}$  are defined at mesh nodes, implementation of the boundary conditions for this quantity along *physical* (i.e. non-interblock and non-reflective) boundaries is straightforward: at each node  $i$  along the entrance boundary for angular direction  $L$  we assign the corresponding boundary value of the incident intensity,

$$I_{[k],L,i} = I_{[k],L,bc}, \quad (4.191)$$

which, in principle, may be an arbitrary function of time, space, frequency group  $[k]$ , and angular direction  $L$ . We single out the following four options, distinguished by the value of the user-defined flag IRADFBC:

IRADFBC = 0	zero incident radiative flux, $I_{[k],L,bc} = 0$ (default),	
IRADFBC = 1	monochromatic monodirectional incident intensity,	(4.192)
IRADFBC = 2	isotropic Planckian incident intensity,	
IRADFBC = 3	arbitrary external radiation field.	

Note that we assume for simplicity that one and the same value of IRADFBC applies to all edges of all blocks (except for the reflective and interblock edges). Complex variation of  $I_{[k],L,bc}$  along different edges of different blocks can be implemented by editing the DOUBLE PRECISION FUNCTION FNUGENBC(i) under the option IRADFBC = 3.

For IRADFBC = 1 the boundary group intensity  $I_{[k],L,bc}$  is non-zero only for the initially prescribed values  $k = k_{bc}$  and  $L = L_{bc}$ ; the dependence of  $I_{[k_{bc}],L_{bc},bc}$  on time and space should be programmed in the DOUBLE PRECISION FUNCTION FNUMONBC(i) for each particular problem.

For IRADFBC = 2 we have

$$I_{[k],L,bc} = B_{[k]}(T_{F,bc}), \quad (4.193)$$

where  $T_{F,bc}$  is the temperature of the incident external radiation (the boundary F-temperature, not necessarily equal to the boundary matter  $B$ -temperature); here one has to program the dependence of the  $T_{F,bc}$  on time and space in the SUBROUTINE BPDFTRAD.

With IRADFBC = 3 the dependence of the incident group intensity  $I_{[k],L,bc}$  on all relevant variables must be preprogrammed in DOUBLE PRECISION FUNCTION FNUGENBC(i).

Note that, in what concerns radiation transport, *reflective* boundaries require special treatment and are discussed below in section ??.

#### 5. Boundary conditions for the Planckian source function $B_\nu(T)$

The boundary condition for the  $B$ -temperature is implemented by assigning the corresponding boundary values  $T_{B,bc}$  to the centers of the ghost cells along a corresponding boundary, followed by application of the general procedure for calculation of the required nodal values  $T_{Bv,i}$  from the known cell-centered values  $T_{Bc,i}$ . We will distinguish among four

typical options for this boundary condition, depending on the value of the user-defined flag IRADBBC:

$$\begin{aligned}
 \text{IRADBBC} = 0 & \quad \text{diffusion-corrected matter temperature, corresponding} \\
 & \quad \text{to a boundary with vacuum under zero external flux,} \\
 \text{IRADBBC} = 1 & \quad \text{reflective condition, } T_{B,bc} \text{ is set equal to } T_i \text{ from the} \\
 & \quad \text{neighboring physical cell (default),} \\
 \text{IRADBBC} = 2 & \quad \text{specified boundary } B\text{-temperature } T_{B,bc}, \\
 \text{IRADBBC} = 3 & \quad \text{diffusion-corrected matter temperature, corresponding} \\
 & \quad \text{to a boundary under thermal external flux at tempera-} \\
 & \quad \text{ture } T_{F,bc}.
 \end{aligned}
 \tag{4.194}$$

The cases of IRADBBC = 1 and IRADBBC = 2 are, in a sense, analogous to the corresponding cases of ITCONBC = 1, 2. For IRADBBC = 1 the ghost-cell value of the  $B$ -temperature is set equal to the cell-centered matter temperature  $T_i$  from the neighboring physical cell. For IRADBBC = 2 the ghost-cell value of the  $B$ -temperature is set equal to the user-defined boundary temperature TEMPBC(iprt,ib,iblk).

**The case of IRADBBC = 0** is based on the steady state solution of the Milne problem [23], employed to evaluate the boundary value of the  $B$ -temperature  $T_{B,bc}$ .

Consider a semi-infinite steady-state planar atmosphere with a constant radiation flux  $\vec{h}_r = H\vec{n}$ , where  $\vec{n}$  is an outward unit normal vector, and  $H > 0$ . If  $\tau$  is the Rosseland optical depth measured inward from the surface, we have

$$\vec{\Omega} \frac{dI}{d\tau} = I(\vec{\Omega}, \tau) - B(\tau),
 \tag{4.195}$$

$$\vec{h}_r = \int_{4\pi} \vec{\Omega} I d\vec{\Omega} = \text{constant} = H\vec{n}.
 \tag{4.196}$$

Having defined

$$J(\tau) = \frac{1}{4\pi} \int_{4\pi} I d\vec{\Omega},
 \tag{4.197}$$

we can integrate Eq. (4.195) over  $\vec{\Omega}$  to obtain

$$B(\tau) = J(\tau).
 \tag{4.198}$$

The solution of the radiative transfer problem (4.195), (4.196) is usually given as [23]

$$J(\tau) = \frac{3H}{4\pi} [\tau + q(\tau)],
 \tag{4.199}$$

where  $q(\tau)$  is a slowly varying monotonic function, called the Hopf function, with  $q(0) = 1/\sqrt{3}$ ,  $q(\infty) = 0.71\dots$

Now consider a physical boundary cell with the value of  $B = B_1$  at its center. If  $\tau_\perp$  is the optical distance between the cell center and the boundary surface, then in the diffusion limit  $\tau_\perp \gg 1$  we have

$$H = \frac{4\pi}{3} \frac{dB}{d\tau} \approx \frac{4\pi}{3} \frac{B_1 - B_0}{\tau_\perp},
 \tag{4.200}$$

where  $B_0$  is the value of  $B(\tau)$  at the boundary surface. On the other hand, according to the Milne solution (4.199), we have

$$B_0 = J(0) = \frac{\sqrt{3}}{4\pi}H. \quad (4.201)$$

Eliminating  $H$  between (4.200) and (4.201), we get

$$B_0 = \frac{B_1}{1 + \sqrt{3}\tau_\perp}. \quad (4.202)$$

Equation (4.202) can be rewritten as

$$T_{B,bc}^4 = \frac{T_i^4}{1 + \sqrt{3}\tau_\perp}, \quad (4.203)$$

where  $T_i$  is matter temperature at the center of a physical boundary cell  $i$ , and  $T_{B,bc}$  is the  $B$ -temperature at the boundary face of this cell. Because the ghost cells are usually relatively very narrow along physical boundaries, we can assign the value of  $T_{B,bc}$  from Eq. (4.203) to the center of the adjacent ghost cell. Note that, although derived in the limit of  $\tau_\perp \gg 1$ , Eq. (4.203) gives a physically correct result for  $\tau_\perp = 0$  as well.

**The case of IRADBBC = 3** corresponds to the Planckian incident radiation intensity  $I_{[k],L,bc} = B_{[k]}(T_{F,bc})$  with temperature  $T_{F,bc}$ , which is a task-specific free parameter. Here one might assume that can simply put  $T_{B,bc} = T_{F,bc}$ ; however, as will be seen in a moment, this is generally not the case. Evidently, in the limit of  $T_{F,bc} = 0$  we recover the case of IRADBBC = 0 with Eq. (4.203) for the boundary  $B$ -temperature  $T_{B,bc}$ . Now we have to generalize Eq. (4.203) to the case of  $B_{bc} \equiv B(T_{F,bc}) > 0$ .

Let us assume that the intensity of the outgoing radiation at the boundary edge of a considered cell is equal to  $I_0$  over the entire outward-looking hemisphere, whereas the intensity of the incoming radiation is equal to  $B_{bc}$  over the entire inward-looking hemisphere. Then in the steady state we obtain

$$B_0 = J_0 = \frac{1}{2}(B_{bc} + I_0). \quad (4.204)$$

Now, if  $B_1$  is the source value at the center of the considered cell, which is at an optical distance  $\tau_\perp$  from the edge, then in the diffusion limit we can write

$$H = \pi(I_0 - B_{bc}) = \frac{4\pi}{3} \frac{B_1 - B_0}{\tau_\perp}. \quad (4.205)$$

From Eqs. (4.204) and (4.205) we obtain

$$B_0 = \frac{B_1 + \frac{3}{2}\tau_\perp B_{bc}}{1 + \frac{3}{2}\tau_\perp}. \quad (4.206)$$

Clearly, the factor  $\frac{3}{2}$  in this equation should be replaced by  $\sqrt{3}$  to bring it into concord with the more rigorous formula (4.202). The limit of  $\tau_\perp = 0$  is also not quite satisfactory: for  $B_1 = 0$  and  $\tau_\perp = 0$  we must get  $B_0 = \frac{1}{2}B_{bc}$  (the outgoing radiation is absent, but the incoming external radiation is there and heats up the matter). Finally, replacing  $B_1$

with  $\sigma_{SB}T_i^4/\pi$ , and  $B_0$  with  $\sigma_{SB}T_{B,bc}^4/\pi$ , we adopt the following simple generalization of Eq. (4.206)

$$T_{B,bc}^4 = \frac{\max\{T_i^4; \frac{1}{2}T_{F,bc}^4\} + \sqrt{3}\tau_{\perp}T_{F,bc}^4}{1 + \sqrt{3}\tau_{\perp}}, \quad (4.207)$$

which adequately reproduces all the limiting cases.

The described boundary conditions for the ghost-cell  $B$ -temperature  $T_{B,bc}$  are applied in the SUBROUTINE BNDROSSB.

---

*Correspondence with the code variables:*

	IRADFBC	user defined flag for the type of boundary condition for incident radiation intensity;
	IRADBBC	user defined flag for the type of boundary condition for the source function $B_{[k]}$ ;
$k_{bc} =$	kfreqbc	frequency group of the monochromatic monodirectional incident intensity;
$l_{bc} =$	iomegbc	beamlet number $l$ of the monochromatic monodirectional incident intensity; $L_{bc} = (i_{ox,bc}, i_{oy,bc}, l_{bc})$ ;
$i_{ox,bc} =$	ioctxbc	sign of the $x$ -component of the propagation direction of the monochromatic monodirectional incident intensity;
$i_{oy,bc} =$	ioctybc	sign of the $y$ -component of the propagation direction of the monochromatic monodirectional incident intensity;
$T_{F,bc} =$	FTEMPVBC(i)	nodal values of the boundary $F$ -temperature; vertex-associated, I-ordered;
$T_{F,bc} =$	FTEMPCBC(i)	ghost-cell values of the boundary $F$ -temperature; vertex-associated, I-ordered;
$I_{[k],L,bc} =$	FNUPLABC(j)	nodal values of the boundary incident group intensity; vertex-associated, j-ordered;

---

## 6. Details of the numerical algorithm for radiation transport

### 1. Allowed mesh deformations

The logical algorithm for solution of the radiation transfer equation has certain tolerance limits for possible mesh deformations in the process of hydrodynamic motion and mesh rezoning. First of all, merging of any two physical neighbor vertices within a single block is not allowed. When two vertices coalesce, the direction of the corresponding mesh face becomes undefined: this situation cannot be handled by the logic of determination of the r-donor cells and vertices. Hence, a quadrilateral cell cannot be allowed to degenerate into a triangle.

Another, more restrictive constraint concerns possible shapes of the quadrilateral cells. An arbitrary quadrangle can only have one of the three possible shapes shown in Fig. 4.21. In a regular convex quadrilateral all the four internal angles measure less than  $180^\circ$ . In a quadrilateral with a boomerang-like shape one internal angle by some corner (corner 2 in Fig. 4.21) exceeds  $180^\circ$  (only one such corner is possible), but the diagonal coming out of this corner (vector  $\vec{a}_d$  in Fig. 4.21) lies inside the quadrilateral. If, however, the diagonal coming out of a corner with an internal angle in excess of  $180^\circ$  lies outside the quadrilateral,

we deal with an *inverted cell*, around one of whose vertices the two principal mesh directions 1 and 2 have a wrong order (i.e. become inverted).

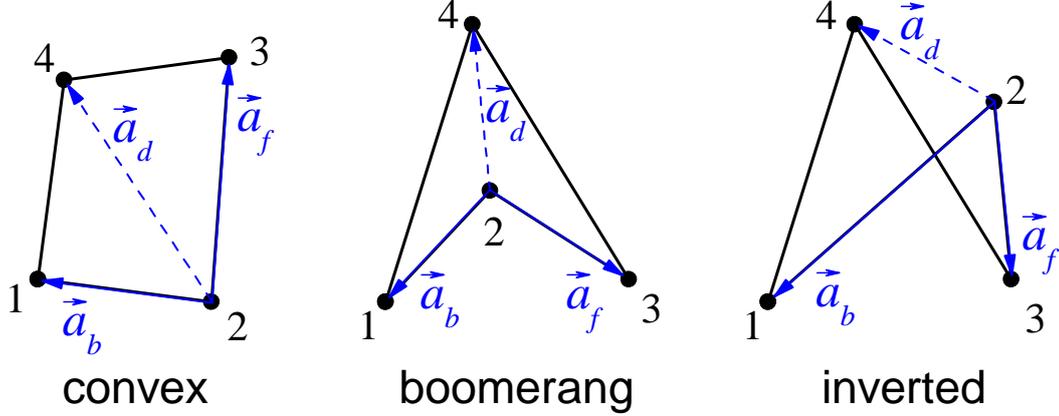


FIG. 4.21: Three possible shapes of a quadrilateral. In this example mesh directions around vertex 3 of the “inverted” cell have the wrong order.

If at least one physical cell of the deformed mesh has an “inverted” shape, we diagnose an *inverted mesh*. The logic of our algorithm for solution of the transfer equation cannot handle inverted physical cells. With an inverted mesh one can construct examples where vertex  $A$  has r-donor vertices  $B$  and  $C$ , while vertex  $B$  has r-donor vertices  $A$  and  $D$ . In such a case, the radiative intensity at  $A$  cannot be calculated before  $B$ , and the intensity at  $B$  cannot be calculated before  $A$ ; as a consequence, our sorting procedure gets into a logical trap. Hence, once the mesh becomes inverted, the simulation is stopped. The *boomerang* physical cells are not fatal but one should keep in mind that the error in the calculated radiative heating (cooling) rates of such cells may be of the order of 100%, and may be even of the wrong sign.

Mathematically, the boomerang and the inverted cells are easily found by evaluating the corresponding vector products. Let the four vertices of a quadrilateral be successively numbered by a cyclic index  $k = 1, 2, 3, 4$  in the counter-clockwise sense as shown in Fig. 4.21. Then, for each vertex  $k$  one can define three vectors

$$\vec{a}_b = \vec{x}_{k-1} - \vec{x}_k, \quad \vec{a}_f = \vec{x}_{k+1} - \vec{x}_k, \quad \vec{a}_d = \vec{x}_{k+2} - \vec{x}_k, \quad (4.208)$$

originating from this vertex:  $\vec{a}_b$  and  $\vec{a}_f$  are, respectively, the backward and the forward “looking” vectors along the cell faces, and  $\vec{a}_d$  is the diagonal vector. The type of the cell is identified by using the following conditions:

- the cell is identified as a convex one if

$$\vec{a}_f \times \vec{a}_b \geq 0 \quad \text{for all } k = 1, 2, 3, 4; \quad (4.209)$$

- the cell is identified as of a boomerang type if there exists a vertex  $k$ , for which

$$\vec{a}_f \times \vec{a}_b < 0 \quad \text{and} \quad (\vec{a}_f \times \vec{a}_d \geq 0 \quad \text{and} \quad \vec{a}_d \times \vec{a}_b \geq 0); \quad (4.210)$$

- the cell is identified as an inverted one if there exists a vertex  $k$ , for which

$$\vec{a}_f \times \vec{a}_b < 0 \quad \text{and} \quad (\vec{a}_f \times \vec{a}_d < 0 \quad \text{or} \quad \vec{a}_d \times \vec{a}_b < 0). \quad (4.211)$$

The mesh quality is checked in subroutine `MESHCH(ich)`, called at the beginning of every hydrocycle. For `ich = 1` only the coalescence of neighbor vertices is checked; for `ich = 2` only the boomerang and the inverted cells are identified; for `ich = 3` both checks are performed. If either a boomerang physical cell or an inverted ghost cell are found, a warning is issued. If an inverted physical cell is found, the simulation is stopped with a message “inverted mesh”.

## 2. Mesh notation: physically associated vertices (PAVs) and their $j$ -ordering

In our method of solving the radiation transfer equation, we have to calculate the group-averaged intensities  $F_\nu(\vec{\Omega})$  at each physical vertex of the multi-block CAVEAT mesh for each beam direction  $\vec{\Omega}$ , and for each frequency group  $\nu$ . In the CAVEAT ordering convention every mesh vertex is identified by a global CAVEAT index

$$I = i_2 \cdot [\mathbf{n1}(\mathbf{iblk}) + 3] + i_1 + 1 + \mathbf{mob}(\mathbf{iblk}), \quad (4.212)$$

where  $i_1 = 0, 1, \dots, \mathbf{n1}(\mathbf{iblk}) + 2$  and  $i_2 = 0, 1, \dots, \mathbf{n2}(\mathbf{iblk}) + 2$  are local (inside a topologically rectangular block `iblk`) indices along, respectively, mesh directions 1 and 2. When index  $I$  runs from  $I = 1$  to  $I = \mathbf{msoccp} = \mathbf{mob}(\mathbf{nblks}) + \mathbf{msz}(\mathbf{nblks})$ , it covers both the physical and the ghost vertices. *Physical vertices* correspond to the values of  $i_1 = 1, 2, \dots, \mathbf{n1}(\mathbf{iblk}) + 1$ ,  $i_2 = 1, 2, \dots, \mathbf{n2}(\mathbf{iblk}) + 1$ .

In a multi-block mesh, when two rectangular blocks touch one another along a common boundary, two different physical vertices (i.e. having different values of index  $I$ ) from two neighboring blocks may represent one and the same physical point, i.e. be *physically identical* (or *equivalent*). A single physical point may be represented by up to 4 logically different physical vertices in the logical space of  $I$ -values. To avoid confusion, we introduce a term “physically associated vertex”, or PAV. *Physically associated vertices* are all the vertices covered by the value ranges

$$\begin{aligned} i_1 &= 1, 2, \dots, \mathbf{n1}(\mathbf{iblk}) + 1, \\ i_2 &= 1, 2, \dots, \mathbf{n2}(\mathbf{iblk}) + 1, \\ \mathbf{iblk} &= 1, 2, \dots, \mathbf{nblks}, \end{aligned} \quad (4.213)$$

in the expression (4.212) for the global CAVEAT index  $I$ . Our task is to calculate  $F_\nu(\vec{\Omega})$  at all physically associated vertices of the computational domain. Because index  $I$  combines PAVs with the ghost vertices, it will be convenient — specially for radiation transport — to introduce a new global sequential numbering of all PAVs, which will be called r-ordering of the mesh.

In the new r-order for the physical mesh vertices, each PAV is identified by an index  $j$ , running sequentially over the range

$$1 \leq j \leq \mathbf{NPAVTOT} = \sum_{\mathbf{iblk}=1}^{\mathbf{nblks}} [\mathbf{n1}(\mathbf{iblk}) + 1] [\mathbf{n2}(\mathbf{iblk}) + 1]. \quad (4.214)$$

The correspondence between the CAVEAT-order and the r-order is established by setting two integer arrays:

$$\text{IPAV}(j) = \text{the value of the CAVEAT } I\text{-index for vertex } j \text{ in the r-ordering;} \quad (4.215)$$

$$\text{JPAV}(I) = \text{the value of } j\text{-index for vertex } I \text{ in the CAVEAT-ordering.} \quad (4.216)$$

This definition does not, of course, provide the values of  $\text{JPAV}(I)$  for all possible values of  $I$ , but only for those which represent PAVs. For the remaining values of the  $I$ -index, which mark ghost vertices, we set  $\text{JPAV}(I) = 0$ .

In addition, to simplify programming, an array

$$\text{MBLKIV}(I) = \text{block number } \text{iblk}, \text{ which contains vertex } I, \quad (4.217)$$

is set in the subroutine FLAGSET. This array is defined for all possible values of  $1 \leq I \leq \text{MSOCCP}$ .

In principle, the order in which PAVs appear in the array  $\text{IPAV}(j)$  can be arbitrary. We adopt the following ordering principle, based on the notion of a *tier*: all PAVs are bunched into a sequence of  $\text{NTIERS}$  tiers. The lowest (or outermost) tier 1 is simply the outer boundary contour (loop)  $\text{mbo}$  of the computational mesh. This boundary loop is comprised of vertices  $\text{IPAV}(j)$ , with  $j$  running from  $j = \text{J1TIER}(1) = 1$  to  $j = \text{J1TIER}(2) - 1$ . Here

$$\text{J1TIER}(\text{kt}) = j\text{-index of the first vertex in tier } \text{kt}, \quad (4.218)$$

where  $1 \leq \text{kt} \leq \text{NTIERS} + 1$ , and  $\text{J1TIER}(\text{NTIERS}+1) = \text{NPAVTOT}+1$ . The convention is that as  $j$  increases from  $j = 1$  to  $j = \text{J1TIER}(2) - 1$ , we encircle the mesh along its outer boundary so that the computational domain stays on the left (for the right-handed coordinate system; see Fig. 4.22). Note that left-handed coordinate systems are not allowed when treating the radiation transport.

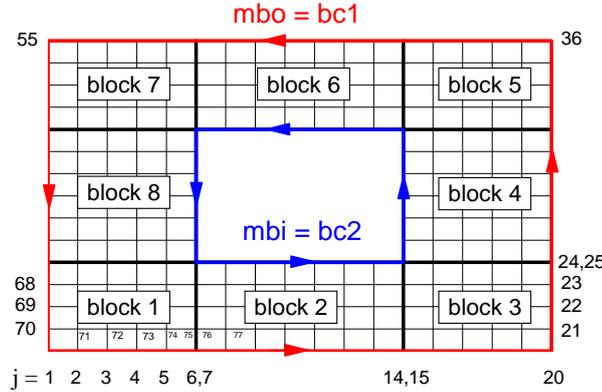


FIG. 4.22: Example of global r-numbering for an 8-block mesh without reflective boundaries ( $\text{IRFLYP}=0$ ) with an inner vacuum cavity. The outer,  $\text{mbo}=\text{bc1}$ , and the inner,  $\text{mbi}=\text{bc2}$ , boundary contours together with  $j$ -indices along  $\text{mbo}$  are indicated. Tier 1 is comprised of vertices  $j = 1, 2, \dots, 70$ ; tier 2 is comprised of vertices  $j = 71, 72, \dots$

Tier  $\text{kt} = 2$  is comprised of the first-order neighbors of vertices in tier  $\text{kt} = 1$ , which do not belong to tier  $\text{kt} = 1$ , and so on. A *first-order* neighbor vertex of vertex  $I$  is defined

such that either its  $i_1$ -index differs by  $\pm 1$  from the  $i_1$ -index of  $I$ , or its  $i_2$ -index differs by  $\pm 1$  from the  $i_2$ -index of  $I$  (but not both!). Clearly, each PAV has exactly 4 first-order neighbors (some of them may be ghost vertices). The uppermost tier  $kt = NTIERS$  is comprised of the innermost vertices that are the farthest from the outer boundary loop  $mbo$ .

Because for  $IRFLTYP \geq 10$  the physical outer boundary  $bc1$  is only part of the outer mesh boundary  $mbo$  (see Fig. 4.20), we introduce the following additional convention:

- the first vertex  $j = 1$  of tier 1 is always the first vertex of the  $bc1$  contour;
- the sequential number  $j = NBND1$  of the last vertex of contour  $bc1$  specifies the total number of vertices  $NBND1$  along the  $bc1$  contour.

Clearly,  $NBND1 = J1TIER(2) - 1$  for  $IRFLTYP = 0$ , and  $NBND1 < J1TIER(2) - 1$  for  $IRFLTYP \geq 10$ . Figure 4.23 gives an example of mesh numbering for the case of  $IRFLTYP = 33$ .

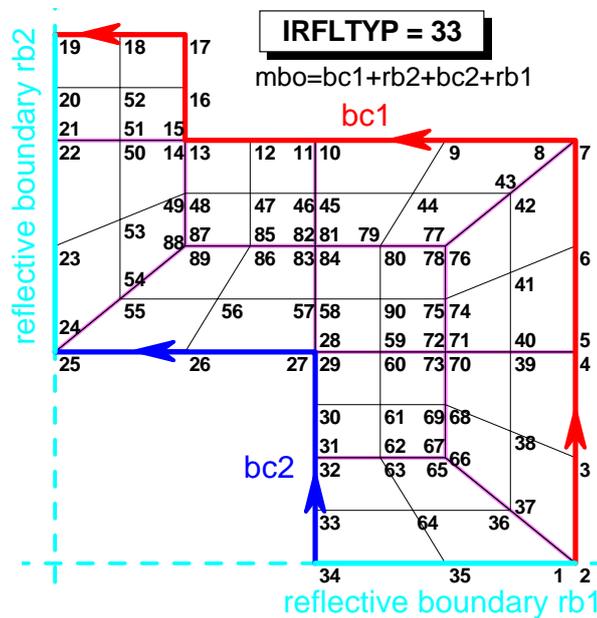


FIG. 4.23: Example of global r-numbering for a 10-block mesh with two reflective boundaries  $rb1$  and  $rb2$ , and an inner vacuum cavity ( $IRFLTYP=33$ ). The outer,  $bc1$ , and the inner,  $bc2$ , boundary contours together with  $j$ -indices for the entire mesh are indicated. Contour  $bc1$  consists of  $NBND1 = 19$  PAVs  $j = 1, 2, \dots, 19$ . Contour  $bc2$  consists of  $NBND2 = 11$  PAVs  $j = 34, 33, \dots, 25, 24$ .

If an inner vacuum cavity is present, its boundary  $bc2$  should not necessarily be represented by the innermost tier  $kt = NTIERS$ . For quick access, the inner boundary contour  $bc2$  is memorized in a separate array  $IBND2(jj)$ ,  $1 \leq jj \leq NBND2$ . As the index  $jj$  increases, we are encircling the inner vacuum cavity so that it always stays on the left (for the right-handed coordinate system; see Fig. 4.22). This is an important convention for the logic of sorting out the PAVs.

To facilitate the sort procedure, described below, we need quick access to the 4 first-order neighbors of each physically associated vertex  $j$ . This is realized by setting an array

JNEIBR(4,msize)

$$\text{JNEIBR}(\alpha, j) = j\text{-index of the neighbor } \alpha \text{ of PAV } j. \quad (4.219)$$

The convention for numbering the 4 neighbors is as follows: neighbor  $\alpha = 1$  is along positive direction of mesh direction 1, neighbor  $\alpha = 2$  is along positive direction of mesh direction 2, neighbor  $\alpha = 3$  is along negative direction of mesh direction 1, neighbor  $\alpha = 4$  is along negative direction of mesh direction 2. If a certain neighbor turns out to be a ghost vertex, there are two possibilities:

- (i) the corresponding neighbor vertex lies across an interblock boundary; in this case

$$\begin{aligned} \text{JNEIBR}(\alpha, j) &= j\text{-index of the PAV from the neighboring block,} \\ &\text{which is physically equivalent to PAV } j; \end{aligned} \quad (4.220)$$

- (ii) the corresponding neighbor vertex lies across a physical boundary; in this case we set

$$\text{JNEIBR}(\alpha, j) = 0. \quad (4.221)$$

All the above mentioned integer arrays and variables, defining global r-ordering, are calculated once and for all at the start of the run in the subroutine RADSET.

### 3. Causal dependencies among PAVs for a given beamlet

Once the ray direction  $\vec{\Omega}$  is fixed, the group-averaged intensity  $F_{[k]}(\vec{\Omega})$  at vertex  $j$  is calculated by integrating the transfer equation (4.7) along the short characteristic of length  $s$  across an *r-donor cell*  $D$ , which is one of the 4 mesh cells adjacent to vertex  $j$ : see Fig. 4.24; the corresponding characteristic originates from point  $O$  at one of the edges of cell  $D$ . Clearly, each PAV  $j$  has a single r-donor cell (unless the mesh is inverted; see below), which may be either a physical or a ghost one. To identify the r-donor cells, we introduce an integer array

$$\text{IPAVD}(j) = \begin{cases} \text{the value of } I\text{-index for the r-donor cell } D \text{ of vertex } j, \\ \quad \text{when cell } D \text{ is a } \textit{physical} \text{ one;} \\ 0, \text{ when cell } D \text{ is a } \textit{ghost} \text{ one across either a reflective boundary,} \\ \quad \text{or a boundary with vacuum region;} \\ -1, \text{ when cell } D \text{ is a } \textit{ghost} \text{ one across an } \textit{interblock boundary} \text{ and is} \\ \quad \text{a copy of a physical cell in the neighbor block.} \end{cases} \quad (4.222)$$

Respectively, the ends  $D_1$  and  $D_2$  of the edge, containing the origin  $O$  of the short characteristic  $s$ , are called the *r-donor vertices* for vertex  $j$ . These are identified by two additional arrays

$$\begin{aligned} \text{JPAVD1}(j) &= \text{the value of } j\text{-index for the r-donor vertex } D_1 \\ &\quad \text{of the acceptor vertex } j, \end{aligned} \quad (4.223)$$

$$\begin{aligned} \text{JPAVD2}(j) &= \text{the value of } j\text{-index for the r-donor vertex } D_2 \\ &\quad \text{of the acceptor vertex } j. \end{aligned} \quad (4.224)$$

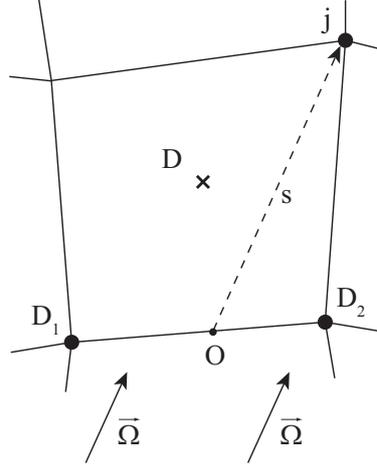


FIG. 4.24: Donor-acceptor causal dependency for transport of a beamlet  $\vec{\Omega}$ : to calculate the intensity at vertex  $j$ , one has to integrate the transfer equation across the r-donor cell  $D$  along the ray (short characteristic of length  $s$  shown as a dashed segment) originating at point  $O$  on edge  $D_1 - D_2$ . The vertices  $D_1$  and  $D_2$  are the two r-donor vertices for the acceptor vertex  $j$ .

Radiation intensity at the characteristic origin  $O$ , needed for integration across cell  $D$ , is evaluated as a linear interpolation between the two intensities at the r-donors  $D_1$  and  $D_2$ , i.e. intensity at  $j$  can be calculated only after the intensities at  $\text{JPAVD1}(j)$  and  $\text{JPAVD2}(j)$  are known; the latter means that vertex  $j$  receives information from its donor PAVs  $\text{JPAVD1}(j)$  and  $\text{JPAVD2}(j)$ . In this way, a chain of causal relations among all PAVs is established.

When a vertex  $j$  belongs to the outer mesh boundary  $\text{mbo}$ , it may receive information not from other PAVs but from the boundary condition for the incoming external intensity. In such a case we set

$$\text{JPAVD1}(j) = \text{JPAVD2}(j) = 0. \quad (4.225)$$

If the mesh has reflective boundaries and a secondary (reflected) beamlet is processed, certain vertices at  $\text{mbo}$  receive their information via a reflected photon path from the previous solution for the conjugate primary beamlet. To distinguish this case, we set either  $\text{JPAVD1}(j) = -j_{mr1}$  negative, or  $\text{JPAVD2}(j) = -j_{mr2}$  negative, or both (see case **IV** below); here  $j_{mr1}$  and  $j_{mr2}$  are the  $j$ -indices of the mirror-image r-donor vertices for vertex  $j$ .

Note that in thus introduced nomenclature  $\text{IPAVD}(j)$  provides the CAVEAT global  $I$ -index for the r-donor cell  $D$ , while  $\text{JPAVD1}(j)$  and  $\text{JPAVD2}(j)$  provide the r-order  $j$ -indices for the r-donor vertices  $D_1$  and  $D_2$ .

Finally, we can single out 5 different logical cases of how the intensity should be calculated at a given PAV  $j$ :

**I (physical donor):** the r-donor cell is a *physical* one: in this case vertex  $j$  receives information from two PAVs in the same block, i.e. one has to integrate along a short characteristic across a physical r-donor cell, and one can do this within local CAVEAT mesh nomenclature; this case is distinguished by the values

$$\text{IPAVD}(j) \geq \text{n1}(1) + 5, \quad \text{JPAVD1}(j) \geq 1, \quad \text{JPAVD2}(j) \geq 1;$$

**II (interblock donor):** the r-donor cell is a *ghost* one across an *interblock boundary*, i.e. coincides with a physical cell in a neighbor block: in this case vertex  $j$  receives

information from a single PAV  $j_{db}$  (which is physically equivalent to vertex  $j$ ) in the corresponding neighbor block (where the r-donor cell of  $j_{db}$  is a physical one), and one has simply to copy the intensity from PAV  $j_{db}$  to PAV  $j$ ; this case is distinguished by the values

$$\text{IPAVD}(j) = -1, \quad \text{JPAVD1}(j) = \text{JPAVD2}(j) = j_{db} \geq 1;$$

**III (vacuum bay):** the r-donor cell is a *ghost* one across a boundary with vacuum but the incoming ray  $\vec{\Omega}$  is shadowed by another region within the original computational mesh; this may only occur when vertex  $j$  belongs either to the outer boundary contour **bc1** or to the inner boundary contour **bc2**; in such a case vertex  $j$  receives information from two remote PAVs  $\text{JPAVD1}(j)$  and  $\text{JPAVD2}(j)$ , which also belong either to **bc1** or **bc2**; this case is distinguished by the values

$$\text{IPAVD}(j) = 0, \quad \text{JPAVD1}(j) \geq 1, \quad \text{JPAVD2}(j) \geq 1;$$

**IV (reflected path):** for reflected (secondary) beamlets there is a possibility that the r-donor cell is a *ghost* one across the boundary **mbo** of the original mesh but vertex  $j$  receives information via a reflected primary beamlet from r-donor vertices  $j_{mr1}$  and  $j_{mr2}$  belonging to one of the mirror images of the original mesh; this may occur when  $j$  belongs either to contour **bc1**, or to contour **bc2**, or to the reflective boundaries **rb1** and **rb2**; shadowing of both the primary and the reflected beamlets is also possible; this case is distinguished by the values

$$\text{IPAVD}(j) = 0, \quad \begin{cases} \text{JPAVD1}(j) = -j_{mr1} \leq -1, \text{JPAVD2}(j) = -j_{mr2} \leq -1, & \text{rbnd0 case,} \\ \text{JPAVD1}(j) = j_{mr1} \geq +1, \text{JPAVD2}(j) = -j_{mr2} \leq -1, & \text{rbnd1 case,} \\ \text{JPAVD1}(j) = -j_{mr1} \leq -1, \text{JPAVD2}(j) = j_{mr2} \geq +1, & \text{rbnd2 case,} \end{cases}$$

if vertex  $j$  belongs to a segment **rb1** or **rb2**, then  $j_{mr1} = j_{mr2} = j$ ;

**V (external flux):** the r-donor cell is a *ghost* one across an outer boundary with vacuum, and the incoming ray  $\vec{\Omega}$  is *not shadowed* by other regions of the *extended* mesh; vertex  $j$  may only belong to the outer boundary loop **bc1**; it receives information from the boundary condition for the incoming external intensity; this case is distinguished by the values

$$\text{IPAVD}(j) = \text{JPAVD1}(j) = \text{JPAVD2}(j) = 0.$$

Arrays  $\text{IPAVD}(j)$ ,  $\text{JPAVD1}(j)$ ,  $\text{JPAVD2}(j)$  should be recalculated for each ray direction  $\vec{\Omega}_L$  and at each time step. This is done in a single sweep over all PAVs in the subroutine **RADON12**. In addition, this calculation provides the values of arrays

$$\text{SPATHR}(j) = s = |j - O|, \tag{4.226}$$

$$\text{FFRACR}(j) = q = \frac{|O - D_1|}{|D_2 - D_1|}, \tag{4.227}$$

that are needed to calculate  $F_\nu(\vec{\Omega})$  at vertex  $j$ .

4. *R-donor data representation*

5. *Principal cases of rd-linkage*

6. *Algorithm for calculation of the r-donor cells and vertices*

We have two different cases for calculating  $\text{IPAVD}(j)$ ,  $\text{JPAVD1}(j)$ ,  $\text{JPAVD2}(j)$  values:

- 1) local calculation, when an intersection of the light ray  $\vec{\Omega}$  with an 8-edge closed polygon line is to be found for an 8-point stencil around the acceptor vertex  $j$  (see Fig. 4.26 below);
- 2) global calculation of possible shadowing effects, when all possible intersections of the light ray  $\vec{\Omega}$  with a polygon line, representing either the *extended* (across the reflective symmetry axes) boundary contour  $\text{bc1}$  or the *extended* boundary contour  $\text{bc2}$ , are to be found.

Both cases can be reduced to the solution of the following ray-polygon problem. Assume that we have a unit vector  $\vec{\omega}$ , emerging from a reference point  $j$ , and that a polygon curve  $\mathcal{C}$  with a finite number of vertices  $C_k$  is given (see Fig. 4.25). Of all possible radial triangles  $C_k\text{-}j\text{-}C_{k+1}$ , emerging from the origin  $j$ , we have to identify those which contain the ray  $s\vec{\omega}$  with  $0 \leq s < \infty$ , and to calculate the coordinates  $\vec{x}_O$  of the intersection points  $O$  of this ray with the polygon segments  $C_k\text{-}C_{k+1}$ .

7. *Solution of the ray-octagon problem*

Let  $\vec{x}_j$  and  $\vec{x}_k$  be, respectively, the coordinates of the reference point  $j$  and of a polygon vertex  $C_k$ . Then

$$\vec{x} = \vec{x}_j + s \cdot \vec{\omega}, \quad 0 \leq s < +\infty \quad (4.228)$$

is the equation of the ray  $s\vec{\omega}$ , and

$$\vec{x} = \vec{x}_k + q(\vec{x}_{k+1} - \vec{x}_k), \quad -\infty < q < +\infty \quad (4.229)$$

is the equation of the straight line connecting points  $C_k$  and  $C_{k+1}$ . The intersection point  $O$  is given by the solution

$$q = -\frac{\vec{\omega} \times \vec{a}_k}{\vec{\omega} \times \vec{a}_{k+1} - \vec{\omega} \times \vec{a}_k}, \quad (4.230)$$

$$s = \frac{\vec{a}_k \times \vec{a}_{k+1}}{\vec{\omega} \times \vec{a}_{k+1} - \vec{\omega} \times \vec{a}_k}, \quad (4.231)$$

where (see Fig. 4.25)

$$\vec{a}_k = \vec{x}_k - \vec{x}_j, \quad (4.232)$$

and  $\vec{a} \times \vec{b} = -\vec{b} \times \vec{a} = a_x b_y - a_y b_x$  is the vector product of two 2-vectors  $\vec{a}$  and  $\vec{b}$ . Evidently, acceptable solutions correspond to the values

$$0 \leq q \leq 1, \quad s \geq 0. \quad (4.233)$$

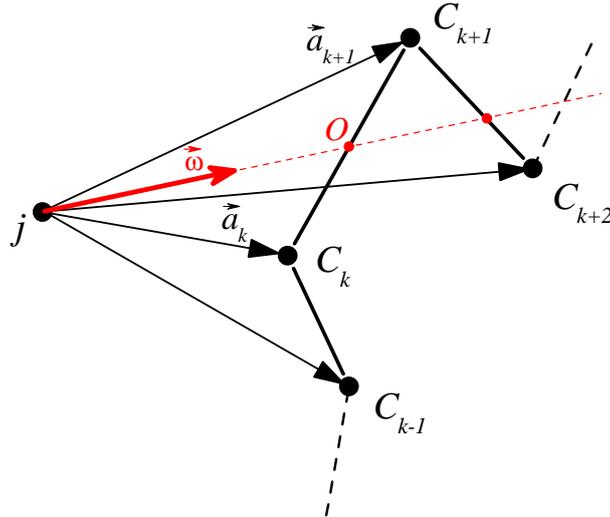


FIG. 4.25: Ray-polygon problem: one has to find the intersection point  $O$  of a ray  $s\vec{\omega}$  with a polygon curve  $C_k$ . Here  $\vec{\omega}$  is a unit vector emerging from a reference point  $j$ ,  $s \geq 0$  is a parameter along the ray.

Also, if an acceptable solution must, for example, cross the polygon curve  $\mathcal{C}$  from left to right (when looking in the direction of increasing  $k$ ), we must discard solutions with  $\vec{\omega} \times (\vec{a}_{k+1} - \vec{a}_k) < 0$ .

The degenerate case of a zero denominator in Eqs. (4.230) and (4.231) requires special treatment. This may happen when either (i)  $\vec{\omega}$  is colinear with  $\vec{x}_{k+1} - \vec{x}_k$ , or (ii)  $\vec{x}_{k+1} = \vec{x}_k$ . In both cases the ray-polygon problem has a solution when, and only when the ray  $s\vec{\omega}$  passes through both vertices  $C_k$  and  $C_{k+1}$ . Thus, if  $\vec{\omega} \times (\vec{a}_{k+1} - \vec{a}_k) = 0$ , we have only to check if the vertex  $C_k$  (or  $C_{k+1}$ , or both) lies on the ray  $s\vec{\omega}$  with  $s > 0$ .

The algorithm for solving the ray-polygon problem is as follows:

- 0:** set  $k = 1$  and calculate  $\vec{\omega} \times \vec{a}_1$ ;
- 1:** calculate  $\vec{\omega} \times \vec{a}_{k+1}$ , then the value of  $q$  from Eq. (4.230), then check whether  $0 \leq q \leq 1$ :
  - if not, go to step **2**,
  - else, calculate  $s$  from Eq. (4.231) and check whether  $s \geq 0$ : if yes, accept the solution as one of several possible and go to step **2**, else goto step **2**;
- 2:** check whether  $k + 1$  was the last vertex along  $\mathcal{C}$ : if not, increment  $k$  by 1 and return to step **1**, else exit the loop.

For a polygon curve with  $N$  vertices and  $n$  possible solutions, this algorithm will require  $2(N + n)$  floating-point multiplications and  $N + n$  floating-point divisions.

8. Remote r-donor vertices across a void region

To calculate remote r-donor vertices for a beamlet  $\vec{\Omega}_L$  across a void region in cases **III** and **IV** of the previous section, we set

$$\omega_x = -\frac{\Omega_{lx}}{\sqrt{\Omega_{lx}^2 + \Omega_{ly}^2}}, \quad \omega_y = -\frac{\Omega_{ly}}{\sqrt{\Omega_{lx}^2 + \Omega_{ly}^2}}, \quad (4.234)$$

and apply the above ray-polygon algorithm to every not-yet-filled vertex  $j$  of the extended **bc1** contour, and to every not-yet-filled  $j$  vertex of the extended **bc2** contour. Of all the possible solutions found for a given vertex  $j$ , we choose the one which corresponds to a “from-outside” crossing of the mesh boundary and has a minimum value of parameter  $s > 0$  (solution of category A): the corresponding polygon edge will be the true one, which irradiates vertex  $j$  across the void interval of length  $s$ . If no solution of category A is found, we choose a solution which corresponds to a “from-outside” crossing of the mesh boundary and has a maximum value of parameter  $s < 0$  (a solution of category B): this, in fact, is a fictitious solution that enables us to treat (within certain limits) “multi-sheet” configurations of overlapping regions of fluid. Solution of category B is applied under the condition that there exists at least one crossing (either “from-outside” or “from-inside”) of the extended **bc1** contour with  $s > 0$ .

This algorithm for calculating remote r-donors is generally robust but not efficient. If  $N_b$  is the total number of boundary vertices to be checked for shadowing, we have to perform  $\sim N_b^2$  operations to calculate all the remote r-donors. Development of a more efficient algorithm for dealing with shadowed parts of the outer boundary **bc1** is left as a task for future work.

9. Local r-donor vertices across an r-donor cell: algorithm I

Consider a reference acceptor vertex  $I$ , for which we use the CAVEAT index because for any vertex  $j$  the calculation is performed locally, inside the block `iblk = MBLKIV(IPAV(j))`. Each physical vertex  $I$  is surrounded by 8 neighbor vertices. The resulting 8-point stencil consists of 4 surrounding quadrilateral mesh cells, as is shown in Fig. 4.26. After we draw the principal diagonal in each of these cells, which originates from the central vertex  $I$ , we obtain 8 “radial” triangles of the 8-point stencil. The numbering of the four principal mesh directions, represented by the four unit vectors  $\vec{e}_\alpha$ ,  $\alpha = 1, 2, 3, 4$ , emerging from  $I$ , has already been described above.

In algorithm I we begin our calculation by identifying the mesh edge  $\alpha_m$ , which is the most distant from vector  $\vec{\omega}$  [as defined in Eq. (4.234)] in terms of the rotation angle  $\theta$  around point  $I$  in the counter-clockwise direction. This is accomplished by calculating the four values of a specially introduced indicator function,

$$f_\alpha = -\frac{1 + \vec{\omega} \cdot \vec{e}_\alpha}{\vec{\omega} \times \vec{e}_\alpha} = -\cot\left(\frac{1}{2}\theta_\alpha\right), \quad (4.235)$$

and finding the maximum of the four:  $f_{\alpha_m} = \max_\alpha \{f_\alpha\}$ . In the case shown in Fig. 4.26 we have  $\alpha_m = 2$ . Clearly, the indicator function  $-\cot(\frac{1}{2}\theta)$  grows monotonically with  $\theta$  in the

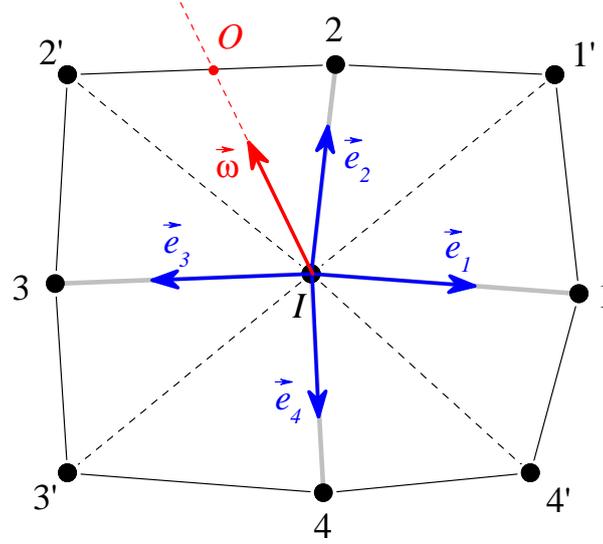


FIG. 4.26: Local 8-point stencil for a reference vertex  $I$ : it includes the 4 surrounding mesh cells, subdivided into 8 “radial” triangles. The unit vector  $\vec{\omega}$  is opposite in direction to the projection of the light propagation vector  $\vec{\Omega}_L$  onto the  $(x_1, x_2)$  plane. In this example the r-donor cell is made up by vertices  $I, 2, 2',$  and  $3$ . Vertices  $2$  and  $2'$  are the two r-donor vertices of the acceptor vertex  $I$ .

interval  $0 < \theta < 2\pi$ . For practical calculations, the following expression is used

$$f_\alpha = \begin{cases} -\frac{1 + \vec{\omega} \cdot \vec{e}_\alpha}{\vec{\omega} \times \vec{e}_\alpha + \text{sign}(\text{floor}, \vec{\omega} \times \vec{e}_\alpha)}, & |1 + \vec{\omega} \cdot \vec{e}_\alpha| > 10^{-6}, \\ -\left(\frac{1}{2}\vec{\omega} \times \vec{e}_\alpha\right) \left[1 + \left(\frac{1}{2}\vec{\omega} \times \vec{e}_\alpha\right)^2\right], & |1 + \vec{\omega} \cdot \vec{e}_\alpha| \leq 10^{-6}, \end{cases} \quad (4.236)$$

where `floor` is a very small positive number (like  $10^{-100}$ ).

Once the mesh edge  $\alpha_m$  is identified, the global CAVEAT index `IPAVD` of the r-donor cell is given by

$$\text{IPAVD}(j) = \text{IPAV}(j) - \text{ICV}(\alpha_m, \text{iblk}), \quad (4.237)$$

where the cell-vertex neighbor array

$$\text{ICV}(\alpha, \text{iblk}) = \begin{cases} 0, & \alpha = 1, \\ 1, & \alpha = 2, \\ \text{n1}(\text{iblk}) + 4, & \alpha = 3, \\ \text{n1}(\text{iblk}) + 3, & \alpha = 4, \end{cases} \quad (4.238)$$

is predefined in the original CAVEAT version.

This procedure works under the assumption that the original order of mesh directions  $\vec{e}_\alpha$  is preserved for all possible mesh deformations, i.e. that vector  $\vec{e}_\alpha$  is always followed by  $\vec{e}_{[\alpha+1]}$  in the counter-clockwise rotation direction; here we have denoted  $[\alpha + 1] = \text{mod}(\alpha, 4) + 1$ . Thus, the situation with mesh “inversion” shown below in Fig. 4.28 is forbidden. A corresponding check that the direction  $[\alpha_m + 1]$  is reached by minimal counter-clockwise rotation of  $\vec{\omega}$  is implemented as part of the algorithm.

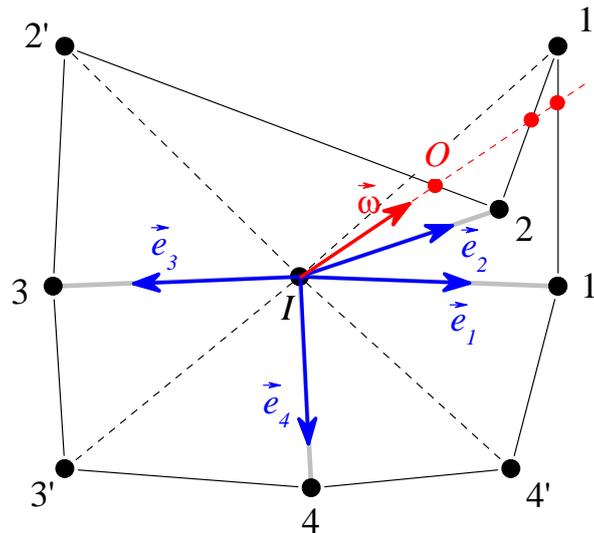


FIG. 4.27: Local 8-point stencil for a reference vertex  $I$  with a negative “radial” triangle  $1'-I-2$ . The rotational order of the principal mesh edges  $\vec{e}_\alpha$  is preserved (no mesh “inversion” occurs).

Once the r-donor cell IPAVD has been identified, the two r-donor vertices IPAVD1 and IPAVD2 are calculated by applying the ray-polygon solution to cell IPAVD; in this case the polygon curve is made up by the 3 (other than  $I$ ) vertices of the r-donor cell (curve 2-2'-3 in Fig. 4.26). Concerning its efficiency, the present algorithm requires on average about 23 multiplications and 6.5 divisions per PAV. This algorithm cannot handle situations with mesh “inversion”, but provides a physically reasonable solution when simply a “negative” triangle appears among 8 radial stencil triangles around the reference vertex  $I$ , as is illustrated in Fig. 4.27. A radial stencil triangle is called “negative” when its area, calculated as  $\vec{a}_k \times \vec{a}_{k+1}$ , becomes negative; here  $\vec{a}_1 = \vec{x}(1) - \vec{x}(I)$ ,  $\vec{a}_2 = \vec{x}(1') - \vec{x}(I)$ ,  $\vec{a}_3 = \vec{x}(2) - \vec{x}(I)$ ,  $\vec{a}_4 = \vec{x}(2') - \vec{x}(I)$ , ... — and so on in the counter-clockwise rotation direction.

### 10. Local r-donor vertices across an r-donor cell: algorithm II

In this algorithm we solve the ray-polygon problem for the closed polygon curve 1-1'-2-2'-3-3'-4-4'-1, which surrounds the reference vertex  $I$  and delimits its 8-point stencil; see Fig. 4.26. The 8 radial stencil triangles are treated in pairs — each pair comprising the corresponding quadrilateral mesh cell (the potential r-donor cell) — and checked successively for the presence of an intersection point  $O$  (a solution). If a single solution is found for a given pair of the stencil triangles, and the corresponding r-donor cell is *physical*, this solution is immediately accepted as the true one and the search is terminated. If a single solution is found for a given pair of the stencil triangles, and the corresponding r-donor cell turns out to be a *ghost* one, this solution is stored (as potentially the true one) and the search is continued. Once the loop over all 8 stencil triangles is finished and no single solution with a physical donor cell is found, the last stored (if any) single solution for a ghost donor cell is accepted as the true one. In this way a preference is always given to solutions in physical r-donor cells.

When one of the constituent triangles turns out to be “negative”, a possibility for two

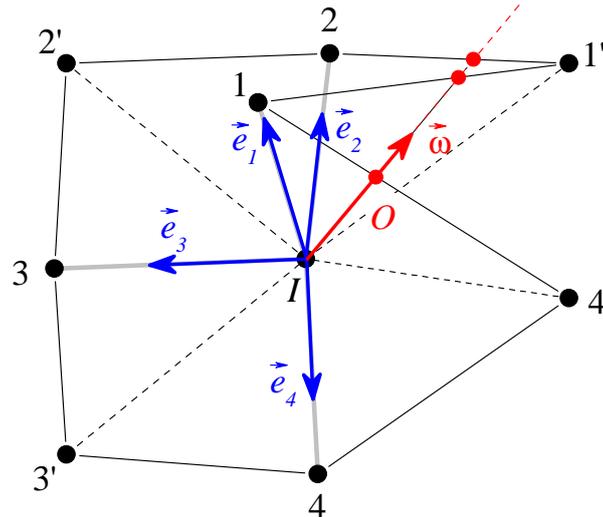


FIG. 4.28: Local 8-point stencil for a reference vertex  $I$  with a “negative” radial triangle  $1-I-1'$ . The rotational order of the principal mesh edges  $\vec{e}_\alpha$  is violated (an “inverted” mesh).

solutions within one cell arises (see Figs. 4.27 and 4.28): in such a case both solutions are ignored and the search is continued until a single solution is found in one of the next neighboring cells. Evidently, the total number of possible ray intersections from inside of a closed polygon curve  $1-1'-2-2'-3-3'-4-4'-1$  is always uneven. To speed up the search procedure, the search is started not always from triangle # 1, but from triangle # 5 if  $\vec{\omega} \times \vec{e}_1 > 0$ , i.e. whenever the solution is expected (for not too strongly deformed meshes) to lie in the lower half-plane  $x_2 < 0$ .

This algorithm appears to be generally more robust than algorithm I because it produces a certain solution even on an “inverted” mesh. It is only later, during the causality sort, that this solution may turn out to be unacceptable. Also, algorithm II is approximately twice as efficient as algorithm I, requiring on average about 11 multiplications and 4 divisions per PAV.

### 11. General scheme of the algorithm

Calculation of arrays  $\text{IPAVD}(j)$ ,  $\text{JPAVD1}(j)$ ,  $\text{JPAVD2}(j)$ ,  $\text{FFRACR}(j)$ , and  $\text{SPATHR}(j)$ , as defined in section 4.6.3, is performed in the subroutine `RADON12`. This is done in five consecutive loops over PAVs: the first two loops over all PAVs  $j = 1, 2, \dots, \text{NPAVTOT}$ , the third loop over the `rb1` and `rb2` segments of the radiation-reflective boundaries, the fourth loop over the PAVs along the outer boundary contour `bc1`, and the fifth loop over the PAVs along the inner boundary contour `bc2`. Initially, the values of the flag  $\text{LORYES}(j)$  are set to `.false.` for all  $j = 1, 2, \dots, \text{NPAVTOT}$ , and vector  $\vec{\omega}$  is calculated.

**Loop 1:** Here we calculate the “literal” values of  $\text{IPAVD}(j)$ ,  $\text{FFRACR}(j)$ ,  $\text{SPATHR}(j)$ , and the  $I$ -indices  $\text{IPAVD1}$ ,  $\text{IPAVD2}$  of the two r-donor vertices for all PAVs  $j = 1, 2, \dots, \text{NPAVTOT}$  with a bias towards physical r-donor cells, i.e. whenever the ray  $s\vec{\omega}$  passes along the interface between a physical and a ghost cell, the physical one is chosen as the r-donor; also, if more than one solution for the r-donor cell  $\text{IPAVD}(j)$  is found, a physical donor

cell is always preferred to the ghost one. “Literal” means that  $\text{IPAVD}(j)$  may be the  $I$ -index of a ghost cell, and  $\text{IPAVD1}$ ,  $\text{IPAVD2}$  may be the  $I$ -indices of ghost vertices. By default, algorithm II is applied to the 8-point stencil of each PAV  $j$ .

If no solution for  $\text{IPAVD}(j)$  is found for a vertex  $j$  along any of the two boundary contours  $\text{mbo}$  or  $\text{mbi}$ , the value  $\text{IPAVD}(j)$  is set equal to  $\text{IPAVD}(j) = -2$  and calculation is continued. In this way we allow boundary PAVs to have strongly distorted (“turned inside out”) neighboring ghost cells.

After the preferred “literal” value of  $\text{IPAVD}(j)$  for a given  $j$  has been determined, it is checked for representing a physical r-donor cell; if yes, then the “literal” values of  $\text{IPAVD}(j)$ ,  $\text{FFRACR}(j)$ ,  $\text{SPATHR}(j)$  are accepted as the true ones, the values of  $\text{JPAVD1}(j) = \text{JPAV}(\text{IPAVD1})$  and  $\text{JPAVD2}(j) = \text{JPAV}(\text{IPAVD2})$  are assigned, the flag  $\text{LORYES}(j)$  is set to `.true.` to mark vertex  $j$  as “filled”. In this way all causal relationships of type **I** (i.e. with a physical donor cell) get properly accounted for.

**Loop 2:** In this second loop over  $j = 1, 2, \dots, \text{NPAVTOT}$ , only vertices with  $\text{LORYES}(j) = \text{.false.}$  are considered, which all have either ghost r-donor cells, or belong to a boundary and have no r-donor cell (when  $\text{IPAVD}(j) = -2$ ). Among all such vertices, a subset is identified, whose ghost r-donor cells are identical copies of physical cells from a corresponding neighbor block across an interblock boundary (case **II** in classification of subsection 4.6.3). For this subset the values  $\text{IPAVD}(j) = -1$ ,  $\text{JPAVD1}(j) = \text{JPAVD1}(j) = j_{db}$ ,  $\text{FFRACR}(j) = \text{SPATHR}(j) = 0$ ,  $\text{LORYES}(j) = \text{.true.}$  are ascribed; here  $j_{db}$  is the  $j$ -index of the physical counterpart of vertex  $j$  across an interblock boundary.

After **Loop 2** only the vertices belonging to the boundary contours  $\text{mbo}$  and  $\text{mbi}$  may remain not filled.

**Loop 3:** This loop is performed over all vertices along the  $\text{rb1}$  and  $\text{rb2}$  segments of the radiation-reflective boundaries (i.e. along reflective boundaries that coincide with either the  $x_1$  or  $x_2$  axis) — if such are present. If the considered beamlet crosses such a segment through vertex  $j$  from outside, then  $\text{IPAVD}(j) = 0$  is assigned, and either  $\text{JPAVD1}(j)$ , or  $\text{JPAVD2}(j)$ , or both (depending on whether we need the  $\text{rbnd0}$ , the  $\text{rbnd1}$ , or the  $\text{rbnd2}$  boundary condition) are set to be negative.

**Loop 4:** This loop is performed over all vertices  $j = 1, 2, \dots, \text{NBND1}$  of the outer boundary contour  $\text{bc1}$ , which have  $\text{LORYES}(j) = \text{.false.}$  to this point. These are sorted between two categories: shadowed and non-shadowed ones. The non-shadowed vertices receive their radiative intensity values from the external boundary condition (case **V**); for them the values  $\text{IPAVD}(j) = \text{JPAVD1}(j) = \text{JPAVD2}(j) = 0$ ,  $\text{FFRACR}(j) = \text{SPATHR}(j) = 0$  are set.

Each shadowed vertex receives its radiation intensity from a certain donor segment of the  $\text{bc1}$  contour across a vacuum “bay”: this donor segment connects vertex  $\text{IPAVD1}$  with a neighbor vertex  $\text{IPAVD2}$  (both newly calculated); for such shadowed boundary vertices  $j$  the values  $\text{IPAVD}(j) = 0$ ,  $\text{JPAVD1}(j) = \text{JPAV}(\text{IPAVD1})$ ,  $\text{JPAVD2}(j) = \text{JPAV}(\text{IPAVD2})$  are assigned; also, the corresponding new (“true”) values of  $\text{FFRACR}(j)$  and  $\text{SPATHR}(j)$  are calculated instead of those obtained in **Loop 1**. When a shadowed vertex receives its intensity along a reflected vacuum-bay path, then, depending on whether the  $\text{rbnd0}$ , the  $\text{rbnd1}$ , or the  $\text{rbnd2}$  boundary condition is to be applied, either  $\text{JPAVD1}(j)$ , or  $\text{JPAVD2}(j)$ , or both are set to be negative.

In this loop, for every processed vertex  $j$  the value  $LORYES(j) = .true.$  is set.

**Loop 5:** This loop does the same job for the contour  $bc2$  as the previous Loop 4 for the contour  $bc1$ . It is performed over all vertices  $jj = 1, 2, \dots, NBND2$  of the inner boundary contour  $bc2$ , which have  $LORYES(j) = .false.$  to this point; here  $j = JPAV(IBND2(jj))$ . For each vertex  $jj$  considered, the donor vertices  $IPAVD1$ ,  $IPAVD2$  across the inner vacuum cavity are calculated; possible shadowing by a warped inner boundary  $bc2$  is accounted for; for all vertices thus processed, the values  $IPAVD(j) = 0$ ,  $JPAVD1(j) = JPAV(IPAVD1)$ ,  $JPAVD2(j) = JPAV(IPAVD2)$ ,  $LORYES(j) = .true.$  are assigned; also, new “true” values of  $FFRACR(j)$  and  $SPATHR(j)$  are calculated.

An example of causal dependencies calculated with the above algorithm for an 8-block mesh of Fig. 4.22 is presented in Fig. ?? below.

### 12. Sorting PAVs to solve the Cauchy problem

Once the causal relationships among PAVs are established, one has to sort them out to determine the sequential order in which the vertex intensities can be calculated. To simplify terminology, we will use expression “to fill vertex  $j$ ” instead of “to calculate  $F_\nu(\vec{\Omega})$  at vertex  $j$ ”. Clearly, vertex  $j$  can be filled only after both its r-donors  $JPAVD1(j)$  and  $JPAVD2(j)$  has been filled.

To distinguish filled vertices from the non-filled (empty) ones, we introduce a logical flag array  $LORYES(0:msize)$ :

$$LORYES(j) = \begin{cases} .false., & \text{vertex } j \text{ is empty,} \\ .true., & \text{vertex } j \text{ is filled.} \end{cases} \quad (4.239)$$

Before starting the sort, we set  $LORYES(0) = .true.$ , and  $LORYES(j) = .false.$  for all  $j \geq 1$ . The sorted list of  $j$ -indices is stored in the array  $JSORTED(j)$ .

*Algorithm based on local connectivity.* The sort starts with a sweep over the outer boundary contour  $bc1$ , where  $1 \leq j \leq NBND1$ . In this sweep all vertices that have  $JPAVD1(j) = 0$  (i.e. receive unshadowed external incoming radiation) are marked as filled. Then, the process of filling the mesh proceeds as sort of a wave (similar to an ionization wave) starting from the filled portion of the outer boundary. The wave propagates in the process of checking the acceptors of the filled vertices: if such an acceptor has both donors filled, it can in its turn also be marked as filled. This wave has its front, which is comprised of filled vertices that have at least one empty acceptor. Clearly, when advancing the sort, there is no point in checking the vertices that lie deep behind the front or far ahead of it. Most efficient will be a sort procedure, where at any moment only the closest neighbors of the front vertices are checked as candidates to be filled. In the local-connectivity algorithm we check for the unfilled acceptors *only among the 4 first-order neighbors* of every filled front vertex.

The sort procedure is realized by placing all the filled front vertices into a special working array  $JRSTACK(msize)$ , organized as a circular stack (an r-stack). “Circular” means that its index  $j$  is incremented not with a  $j=j+1$  operator but with a  $j=\text{mod}(j,msize)+1$  operator. The sort is accomplished by incrementing independently two pointers delimiting the active part of the r-stack: the lower pointer  $jp1$  indicates the first filled location, and the upper pointer  $jp2$  indicates the last filled location. Since the r-stack locations between

(and including)  $jp1$  and  $jp2$  contain all filled PAVs that have at least one empty first-order neighbor, the sort is ended when  $jp1 = jp2$ .

The logical scheme of the algorithm can be presented as follows:

- 0:** place into the r-stack  $JRSTACK$  and into the sorted list  $JSORTED$  all vertices of the outer boundary  $bc1$ , marked as filled during the initial sweep of  $bc1$  (which receive unshadowed external incoming radiation); initialize  $jp1$  to 1, and  $jp2$  — to the last used location in  $JRSTACK$ ;
- 1:** check if any of the 4 first-order neighbors of vertex  $j = JRSTACK(jp1)$  is empty:
  - if not, go to step **2**;
  - if yes, check whether the two r-donors of the empty neighbor are both filled: if yes, mark this neighbor as filled, put it into the sorted list  $JSORTED$  and on top of the r-stack, having simultaneously incremented  $jp2$  by 1; otherwise leave this neighbor empty and do nothing;
  - if in the process of such check any of the 4 first-order neighbors of vertex  $JRSTACK(jp1)$  is left empty, reinsert vertex  $JRSTACK(jp1)$  on top of the r-stack, having simultaneously incremented  $jp2$  by 1;
- 2:** if  $jp1 < jp2$ , then increment  $jp1$  by 1 and return to step **1**; otherwise exit the sort procedure.

In this procedure each newly filled vertex is passed through the r-stack. A filled vertex drops out of the r-stack only after all 4 of its first-order neighbors get filled.

*Algorithm for non-local connectivity.* Although on a quadrangular mesh the 4 first-order neighbors always provide connection from any vertex to any other vertex, the above described algorithm fails in certain situations where the boundary with vacuum becomes strongly distorted, and multiply shadowed areas appear. An example of such a situation is shown in Fig. 4.29, where the algorithm based on local connectivity fails to “fill” the unshaded area.

To fix this problem, the above algorithm has to be modified to include non-local checks for remote acceptor vertices across vacuum regions. First of all, two lists of such remote acceptors are prepared before starting the sort:  $jbc1vacc(j)$ ,  $j = 1, 2, \dots, nbc1vacc$  for the contour  $bc1$ , and  $jbc2vacc(j)$ ,  $j = 1, 2, \dots, nbc2vacc$  for the contour  $bc2$ . Second, two more steps **1a** and **1b** are added after step **1** in the above algorithm, where the remote acceptors from the prepared lists are checked along, respectively, the contours  $bc1$  and  $bc2$ :

- 1a:** if the treated vertex  $j = JRSTACK(jp1)$  from the r-stack belongs to contour  $bc1$  (i.e. if  $1 \leq j \leq NBND1$ ) and the  $jbc1vacc$  list is not empty, do the following:
  - check every remote acceptor from the  $jbc1vacc$  list for being empty and having its two donors filled: if yes, mark it as filled, put it into the sorted list  $JSORTED$  and on top of the r-stack, having simultaneously incremented  $jp2$  by 1;
  - if at least one acceptor from the  $jbc1vacc$  list is not filled and has  $j$  as a donor, reinsert  $j$  into the r-stack;

In step **1b** the same checks are done for contour  $bc2$ .

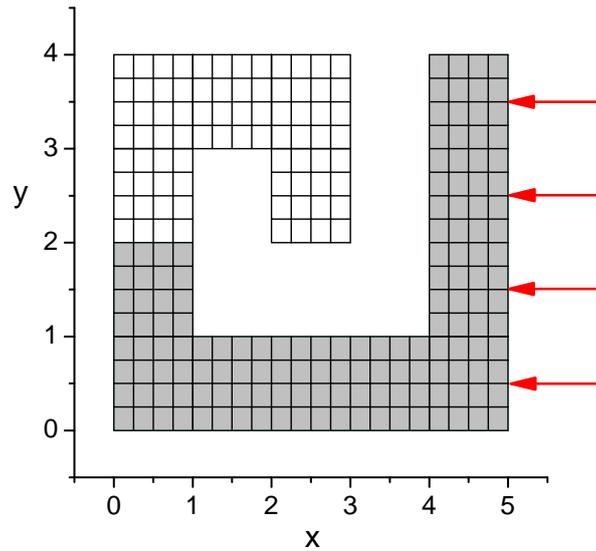


FIG. 4.29: Mesh example where the sort algorithm, based on local connectivity, fails to “fill” the unshaded area.

*Efficiency of the sort procedure.* In typical situations, the number of operations in the above sort procedure should scale as  $\propto N$ , where  $N = \text{NPAVTOT}$ . However, in some special cases (a narrow-band-like mesh) it may scale as  $\propto N^2$ , which is terribly inefficient. As a remedy for such loss of efficiency in certain cases, one can propose an occasional turnover (from top to bottom) of the r-stack. The r-stack is turned over whenever the coefficient

$$f_{s,eff0} = \frac{\text{number of jp1 incremets since the last stack turnover}}{\text{number of newly filled locations in JSORTED since the last stack turnover}} \quad (4.240)$$

exceeds a certain threshold value (typically 2-3), and the numerator in (4.240) exceeds  $\text{NBND C1}$ . The overall efficiency of the sort procedure is measured by a coefficient

$$f_{s,eff} = \frac{\text{NBND C1} + \text{total number of incremets of the pointer jp1}}{\text{NPAVTOT}}. \quad (4.241)$$

Typically, the values of  $f_{s,eff}$  lie in the range  $1.5 < f_{s,eff} < 2.0$ . Figure ?? gives a specific example of the sorted order for PAV  $j$ -indices obtained with the above described algorithm.

## 5. LASER ENERGY DEPOSITION

### 1. Classification of implemented models

Presently, there are three different laser deposition models implemented in the RALEF code. They are distinguished by the value of the user-defined integer flag `iflasdep`:

- `iflasdep`  $\leq 0$ : no laser deposition;
- `iflasdep` = 1: a non-refractive model #1 (an XS model) of the laser light transport in the X mode along straight *short* characteristics; laser intensity is assigned to mesh nodes (a non-conservative numerical scheme similar to that for radiation transport);

- `iflasdep = 2`: a refractive-reflective model #2, which splits into a reflective H-model along straight *long* characteristics, based on a straight-through solution of the 1D Helmholtz equation (when `ala2gop = 0.0`), and a refractive/reflective RH-model along curved short characteristics in the underdense plasma combined with the 1D Helmholtz equation along straight long characteristics in the overdense plasma (when `ala2gop > dfloor`); laser intensity is assigned to cell edges (a conservative numerical scheme).
- `iflasdep = 3`: a universal model #3 of the laser transport along *long* characteristics, which combines all three possible *transport modes*, namely the X, H, and RH modes (a conservative numerical scheme).

All the incident laser light is assumed to consist of several (`nblas`  $\geq$  1) mutually incoherent independent *beams*, each of which is characterized by individual *beam geometry*, individual values of the laser frequency, propagation direction, focal distance, temporal power profile, etc. However, in one and the same job all the laser beams are treated within one and the same model, fixed by the value of the user-defined parameter `iflasdep`. In principle, it is not forbidden to change the value of `iflasdep` at subsequent restarts of the job, but it is not recommended to do so because the output on laser related quantities is model-dependent and may become confusing.

As already mentioned, each laser deposition model may have (or not have) different options concerning the *transport mode* of the laser light. The transport mode refers to the way how propagation and dissipation of the laser energy along an *elementary ray* is described mathematically. Model #1 has only one X transport mode. In model #2 the user may choose between the H and RH transport modes, but once chosen, the transport mode remains the same for all the laser beams; it may, however, be changed at subsequent restarts of the job or in the process of its execution. Model #3 allows a free choice between the three transport modes (X, H, and RH) for any individual laser beam.

Having been the first implemented, model #1 is physically the most primitive and has a disadvantage (though not a big one) of being based on a non-conservative scheme. It is completely unsuitable for long-wavelength lasers like CO<sub>2</sub> and may be considered as obsolete. Model #2 was an experiment on implementing the method of short characteristics for describing the laser energy dissipation in refractive media — not a particularly successful experiment: the method of short characteristics in this case turned out to be inferior in terms of the accuracy and computational efficiency to the method of long characteristics. The last implemented model #3 is the most universal and flexible (in terms of available options), the most accurate and computationally least demanding when the job is run on a multiprocessor machine. It is recommended to be always used for practical applications. This explains why the focus of the description below is primarily on model #3.

## 2. Different options for laser beam geometry

The geometry of individual laser beams (the *beam geometry*) is defined regarding the propagation of laser light in vacuum, i.e. under the assumption that the computational domain is filled with an infinitely tenuous gas (plasma). Every individual beam is assumed to have a single propagation axis (the *beam axis*), and to be composed of a large number of *elementary rays* (the geometric optics rays). The propagation direction along a given elementary ray is defined by a unit 3D vector  $\vec{\Omega}_{ray}$  in the global 3D space, and by a unit

2D vector  $\vec{\omega}_{ray}$  in the computational 2D plane. Both  $\vec{\Omega}_{ray}$  and  $\vec{\omega}_{ray}$  may vary in space. The beam axis is always defined as a straight line in the 2D computational plane by specifying (for beam #`iblas`) the 2D coordinates

$$(x_{foc}, y_{foc}) = (\text{x foc blas}(\text{iblas}), \text{y foc blas}(\text{iblas})) \quad (5.1)$$

of the *focal point*, lying on this axis, and the two fixed projections

$$(\Omega_{las,x}, \Omega_{las,y}) = (\text{Ome blas x}(\text{iblas}), \text{Ome blas y}(\text{iblas})) \quad (5.2)$$

onto the computational plane of a unit 3D propagation vector  $\vec{\Omega}_{las}$ . The 2D vector  $\vec{\omega}_{las}$ , defined by Eq. (5.5), is colinear with the laser axis in the 2D computational plane.

In the Cartesian *xy*-geometry the beam axis is allowed to have a non-zero *z*-component  $-1 < \Omega_{las,Z} < 1$ , perpendicular to the computational *xy*-plane, which can be calculated as

$$\Omega_{las,Z} = \pm \sqrt{1 - \Omega_{las,x}^2 - \Omega_{las,y}^2}. \quad (5.3)$$

In other words, in the *xy*-geometry we generally have

$$\Omega_{las,x}^2 + \Omega_{las,y}^2 \leq 1, \quad (5.4)$$

and the unit 2D vector along the beam axis is given by

$$\vec{\omega}_{las} = \left( \frac{\Omega_{las,x}}{\sqrt{\Omega_{las,x}^2 + \Omega_{las,y}^2}}, \frac{\Omega_{las,y}}{\sqrt{\Omega_{las,x}^2 + \Omega_{las,y}^2}} \right). \quad (5.5)$$

In the axi-symmetric *rz*-geometry our treatment of laser deposition is restricted to the cases where the beam axis lies fully in the computational *rz*-plane, i.e. to the cases with

$$\Omega_{las,x}^2 + \Omega_{las,y}^2 = \Omega_{las,R}^2 + \Omega_{las,Z}^2 = 1, \quad (5.6)$$

and the 2D unit vector along the beam axis given by

$$\vec{\omega}_{las} = (\Omega_{las,x}, \Omega_{las,y}). \quad (5.7)$$

Depending on the relative orientation of elementary rays with respect to the beam axis, model #3 admits the following types (options) of the beam geometry, distinguished by the value of the beam-individual integer flag `iblasgeom(iblas)`:

- `iblasgeom(iblas) = 0`: a *cylindrical* laser beam, whose elementary rays all have the same fixed propagation direction  $\vec{\Omega}_{ray} = \vec{\Omega}_{las}$  in both the 2D and 3D space;  $d_{Rl} = 0$ ,  $d_{vp} = \text{ceiling}$ ;
- `iblasgeom(iblas) = 1`: a *conical* laser beam in the *rz*-geometry with  $\Omega_{las,R} \neq 0$ , whose elementary rays all have the same fixed 2D propagation direction  $(\Omega_{ray,R}, \Omega_{ray,Z}) = (\Omega_{las,R}, \Omega_{las,Z})$  in the *rz*-plane (but not in the 3D space !); every elementary ray of a conical beam is straight in both the 2D and 3D spaces; different rays have one and the same direction in the 2D computational plane, but different directions in the 3D space;  $d_{Rl} = 0$ ,  $d_{vp} = \text{ceiling}$ ;

- `iblasgeom(iblas) = 2`: a *geometrically focused* (or, simply, *focused*) laser beam, whose elementary rays are straight lines (both in 2D and 3D) passing through a single focal point  $(x_{foc}, y_{foc})$  on the beam axis, and all having  $\vec{\omega}_{ray} \cdot \vec{\omega}_{las} > 0$ ; different rays have different 2D (as well as 3D) propagation directions; the target body can be positioned either before (in front of) the focus (*converging incidence*), or after (behind) the focus (*diverging incidence*);  $d_{RI} = 0$ ,  $-\text{ceiling} < d_{vp} < \text{ceiling}$ ;
- `iblasgeom(iblas) = 3`: a *convergent* laser beam, whose elementary rays are straight lines (both in 2D and 3D) converging upon a single central (focal) point  $(x_{foc}, y_{foc})$  on the beam axis; the 2D propagation directions of elementary rays  $\vec{\omega}_{ray}$  cover the entire  $2\pi$  of the azimuth/polar angle; the direction of the beam axis  $\vec{\omega}_{las}$  has no meaning; the target body is always before the focus;  $d_{RI} = d_{vp} = 0$ ;
- `iblasgeom(iblas) = 4`: a *quasi-Gaussian* laser beam, whose elementary rays in the 2D computational plane are hyperbolas, defined by the equation

$$r^2 = r_w^2 + \frac{r_w^2}{d_{RI}^2}(z - z_{foc})^2, \quad (5.8)$$

where  $z$  is the coordinate along the beam axis,  $r$  is the coordinate in the perpendicular direction,  $r_w$  is the radius of the crossing point with the beam-waist plane, and  $d_{RI}$  is the Rayleigh length;  $\text{floor} < d_{RI} < \text{ceiling}$ .

The flag `iblasgeom(iblas)` is not a user-defined parameter: it is calculated from the values of two other user-defined parameters

- $d_{vp} = \text{dvpblas}(\text{iblas})$  — the distance from the focal point  $(x_{foc}, y_{foc})$  to the *view plane* along the beam axis (can be both positive and negative), and
- $d_{RI} = \text{dRIblas}(\text{iblas})$  — the Rayleigh length of a quasi-Gaussian beam (assumed to be a non-negative quantity).

The *view plane* is defined as a perpendicular to  $\vec{\Omega}_{las}$  plane where the spatial profile of laser intensity for the considered beam is specified. The beam `iblas` is qualified as quasi-Gaussian when  $\text{floor} < d_{RI} < 0.5\text{ceiling}$ ; in this case the value of  $d_{vp}$  is not used and is irrelevant. For  $d_{RI} \leq \text{floor}$  (or  $d_{RI} \geq 0.5\text{ceiling}$ ) the beam is qualified as either cylindrical or conical for  $|d_{RI}| > 0.5\text{ceiling}$ , as convergent for  $|d_{RI}| < \text{floor}$ , and as focused otherwise.

In a cylindrical beam (see Fig. 5.1), all elementary light rays are parallel to one other in the 3D space and have a common fixed 3D propagation vector  $\vec{\Omega}_{las}$ . Hence, such a beam does not change its cross-section along the propagation direction, and the energy flux  $F_{las}$  [erg cm<sup>-2</sup> s<sup>-1</sup>] remains constant along the elementary ray whenever the absorption coefficient  $k_{las}$  is zero. In the Cartesian case (`iradial` = 0), cylindrical beams are possible for any  $\vec{\Omega}_{las}$ , whereas in the axisymmetric cases (`iradial` = 1, 2) they are only possible if  $\vec{\Omega}_{las}$  is colinear with the global rotational  $Z$ -axis. Respectively, the transverse cross-section of a cylindrical beam will be either an infinite (along the  $Z$ -axis) band — for `iradial` = 0, or a circle (circular annulus) with the radius constant along the propagation direction — for `iradial` = 1, 2.

A conical beam is allowed only in the axisymmetric geometry: in such a beam light propagates along the rays that converge upon the axis of rotational symmetry; see Fig. 5.2.

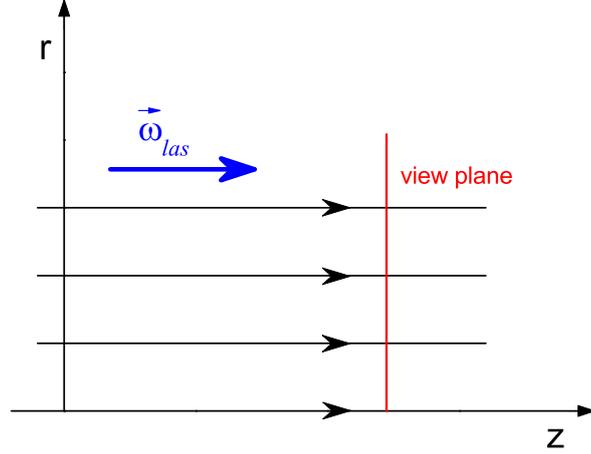


FIG. 5.1: Cylindrical laser beam, propagating along the  $z$ -axis in the  $rz$ -geometry.

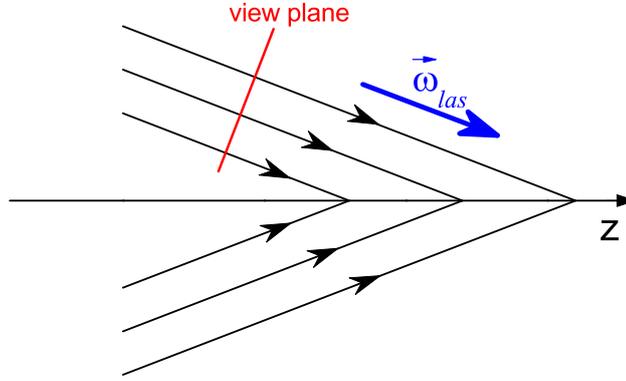


FIG. 5.2: Conical laser beam, propagating along the  $z$ -axis in the  $rz$ -geometry.

When the rotation axis is along the  $Z$  coordinate, the 3D propagation direction

$$\vec{\Omega}_{las} = \vec{\Omega}_{las}(X, Y, Z) = \left\{ \frac{X}{R} \Omega_{las,R}; \frac{Y}{R} \Omega_{las,R}; \Omega_{las,Z} \right\} \quad (5.9)$$

of different light rays in a given conical beam is a function of position in the 3D space; here  $R = (X^2 + Y^2)^{1/2}$  is the cylindrical radius, and  $\Omega_{las,R}$  and  $\Omega_{las,Z}$  are the fixed (for a given beam) components of the 2D propagation vector, obeying the normalization

$$\Omega_{las,R}^2 + \Omega_{las,Z}^2 = 1. \quad (5.10)$$

The transverse cross-section of a conical beam is a circular annulus, coaxial with the rotation axis  $Z$ , whose radius  $R$  varies along  $Z$ ; the laser energy flux  $F_{las}$  along such a beam varies in inverse proportion to  $R$ . Clearly, in the particular case of  $\Omega_{las,R} = 0$  and  $\Omega_{las,Z} = 1$  the conical beam becomes a cylindrical one. The conical beams are marked by setting a special integer flag `iconblas(iblas) = 1`, which otherwise has a zero value.

Because the laser beams — in contrast to the beamlets of thermal radiation — are allowed to propagate exactly along the 2D coordinate axes (either  $x$  or  $y$ ), and because in the  $rz$ -geometry such *coaxial* beams can be “topologically” different from the non-coaxial beams (like the difference between coaxial cylindrical and non-coaxial conical beams), a special integer flag `iaxlblas(iblas)` is introduced in the RALEF code to mark this distinction for a given beam `iblas`:

- `iaxlblas(iblas) = 0`: a *non-coaxial* beam,  $|\Omega_{las,x}| > 0$ ,  $|\Omega_{las,y}| > 0$ ,
- `iaxlblas(iblas) = 1`: an *x-coaxial* beam,  $|\Omega_{las,x}| > 0$ ,  $\Omega_{las,y} = 0$ ,
- `iaxlblas(iblas) = 2`: an *y-coaxial* beam,  $\Omega_{las,x} = 0$ ,  $|\Omega_{las,y}| > 0$ .

Numerically the condition  $\Omega_{las,x} = 0$  ( $\Omega_{las,y} = 0$ ) is implemented as the inequality  $|\Omega_{las,x}| < \text{dfloor}$  ( $|\Omega_{las,y}| < \text{dfloor}$ ), where `dfloor` is the “decimal floor”, i.e. a small number on the order of  $10^{-15}$ – $10^{-13}$ . The flags `iconblas(iblas)`, `iaxlblas(iblas)` are loaded in subroutine `LASCHECK`, file ‘`f02_init.f`’.

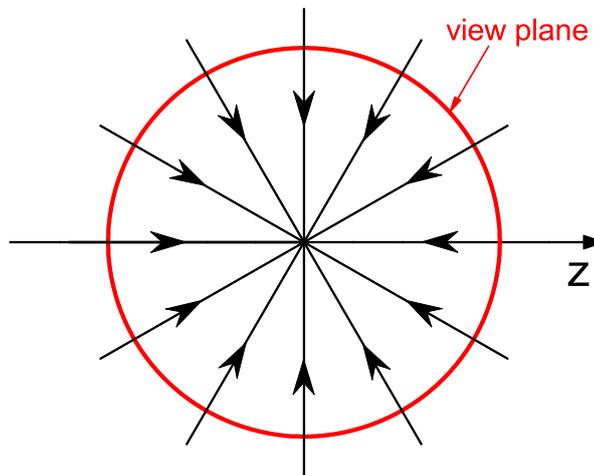


FIG. 5.3: Convergent laser beam in the  $rz$ -geometry.

At a first glance, the converging beam geometry, illustrated in Fig. 5.3, might appear as exotic and not quite relevant to realistic 2D (or 3D) configurations; it can, however, be useful when, for example, establishing a correspondence with 1D simulations of laser-irradiated cylindrical or spherical targets. In this case the “view plane” appears as a unit circle in the 2D computational plane; when setting up the spatial laser profile, the coordinate along this “view circle” is the azimuthal (`iradial` = 0) or polar (`iradial` = 1,2) angle in radians, measured either from the  $x_1$ -axis (`iradial` = 0), or from the axis of rotation (`iradial` = 1,2); the reference laser intensity `Fblas00` is either the laser beam power per unit length per radian [ $\text{erg cm}^{-1} \text{s}^{-1} \text{radian}^{-1}$ ] (`iradial` = 0), or the power per unit solid angle [ $\text{erg s}^{-1} \text{ster}^{-1}$ ] (`iradial` = 1,2).

Quasi-Gaussian beams represent a geometric-optics version of Gaussian beams from wave optics: here all the elementary rays in vacuum are hyperbolas; see Fig. 5.4. In the  $rz$ -geometry, the beam axis must coincide with the rotation axis, and the beam has a circular

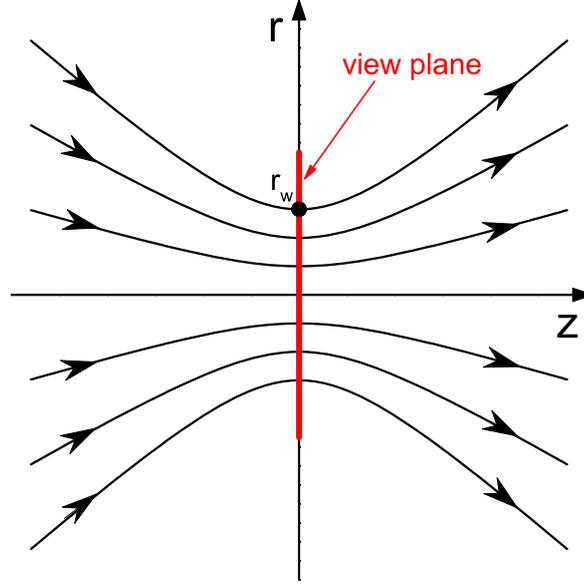


FIG. 5.4: Quasi-Gaussian laser beam in the  $rz$ -geometry.

cross-section. In the  $xy$ -geometry, the direction  $\Omega_{blas}$  of the beam axis can be arbitrary, and the beam cross-section has a shape of an infinite (along the global  $Z$ -axis) band.

In the  $rz$ -geometry, where the beam axis coincides with the rotational  $z$ -axis, a quasi-Gaussian beam is fully defined by the values of two parameters and by one profile function, namely,

- $z_{foc}$  — the coordinate of the beam waist (identified with the focal point) along the beam axis,
- $d_{RL}$  — the Rayleigh length, i.e. the distance from the focal point where the area of the beam cross-section doubles relative to the beam waist, and
- $p_s(r_w)$  — the normalized radial profile of the laser intensity in the beam-waist cross-section.

In the geometric-optics approximation, we assume that (i) the value of the Rayleigh length  $d_{RL}$  is decoupled from the waist size  $r_{w0}$  and the laser wavelength  $\lambda$ , i.e. that the relationship  $\pi r_{w0}^2 = \lambda d_{RL}$  must not necessarily be satisfied, and (ii) the spatial beam profile  $p_s(r_w)$  must not necessarily be Gaussian — hence the term a “quasi-Gaussian” beam.

Once  $z_{foc}$ ,  $d_{RL}$  and  $p_s(r_w)$  are specified, every elementary ray of a quasi-Gaussian beam is identified by its radius  $r_w$  in the beam-waist plane, and the equation  $r = r(z)$  of the ray is given by Eq. (5.8).

### 3. Non-refractive XS model #1

#### 1. Transfer equation for the laser light

*a. Cartesian geometry, cylindrical beams.* In the Cartesian case (i.e. for `iradial` = 0), where  $x = X, y = Y, z = Z$ , and only cylindrical laser beams are allowed, the propagation vector  $\vec{\Omega}_{las}$  is constant in space and time for all the light rays within a given beam. The three components of vector  $\vec{\Omega}_{las}$  obey the normalization condition

$$\Omega_{las,x}^2 + \Omega_{las,y}^2 + \Omega_{las,z}^2 = 1. \quad (5.11)$$

In case of  $\Omega_{las,z} \neq 0$  we have  $\Omega_{las,x}^2 + \Omega_{las,y}^2 < 1$ . Note that, unlike in the case of thermal radiation, if an incident laser beam with  $\Omega_{las,z} > 0$  is specified, it does not mean that its mirror-image beam with  $-\Omega_{las,z}$  is automatically present.

To characterize the intensity of the laser irradiation, we use the energy flux  $F_{las}$  [erg cm<sup>-2</sup> s<sup>-1</sup>] per unit surface area perpendicular to the propagation vector  $\vec{\Omega}_{las}$ , which is a function of the two relevant spatial coordinates  $x = X, y = Y$ . Then the vector of the laser energy flux density within a given beam can be written as

$$\vec{h}_{las} = F_{las} \vec{\Omega}_{las}. \quad (5.12)$$

Assuming that there are no sources of laser light in the simulated domain, one can write the transfer equation in the form

$$\text{div } \vec{h}_{las} \equiv \Omega_{las,x} \frac{\partial F_{las}}{\partial x} + \Omega_{las,y} \frac{\partial F_{las}}{\partial y} = -k_{las} F_{las}, \quad (5.13)$$

where  $k_{las}$  [cm<sup>-1</sup>] is the absorption coefficient for the monochromatic light in the given beam.

Numerical solution of the transfer equation (5.13) along the short characteristics is a trivial particular case of the solution for thermal radiation constructed in section 4.3.2. In particular, the laser flux  $F_{las,i}$  at a node  $i$  (see Fig. 5.5) is calculated as

$$F_{las,i} = F_{las,O} e^{-\tau_o}, \quad (5.14)$$

where

$$\tau_o = \frac{k_{las,i_d} |\vec{x}_i - \vec{x}_O|}{\sqrt{\Omega_{las,x}^2 + \Omega_{las,y}^2}}, \quad (5.15)$$

$k_{las,i_d}$  is the laser absorption coefficient in the donor cell  $i_d$ , and the initial flux  $F_{las,O}$  is obtained as a linear interpolation between the fluxes at the donor vertices  $i_{d1}$  and  $i_{d2}$ .

Similar to thermal radiation, our numerical scheme for calculating the total power of laser heating of a given cell  $i$  is based on the volumetric formula

$$W_{las,i} = k_{las,i} \int_{\text{cell } i} F_{las,i} dx dy. \quad (5.16)$$

To evaluate the surface integral in Eq. (5.16), we employ the same algorithm from section 4.4.4 that was developed for thermal radiation. In this algorithm the four beamlets, obtained by corresponding reflections across the  $x$ - and  $y$ -coordinate axes of a single primary beam, are treated as two pairs of two mutually opposite beamlets. After we rotate the

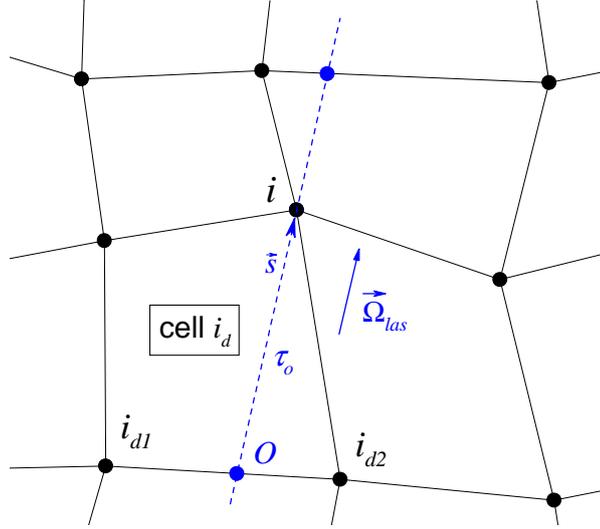


FIG. 5.5: R-donor cell for vertex  $i$  and the laser propagation direction  $\vec{\Omega}_{las}$ . The short characteristic  $\vec{s} = \vec{x}_i - \vec{x}_O$  is shown as a dashed line  $O-i$ .

axes as described in section 4.4.4 and accept the notation in Fig. 4.19, we can write down the finite-difference form of Eq. (5.16) as

$$W_{las,i} = \sqrt{\Omega_{las,x}^2 + \Omega_{las,y}^2} \sum_{2 \text{ pairs}} \int_{\tilde{x}_A}^{\tilde{x}_D} (F_{las,E}^+ + F_{las,S}^-) (1 - e^{-\tau_h}) d\tilde{x}, \quad (5.17)$$

where

$$\tau_h = \tau_h(\tilde{x}) = \frac{k_{las,i} h(\tilde{x})}{\sqrt{\Omega_{las,x}^2 + \Omega_{las,y}^2}}, \quad (5.18)$$

and  $F_{las,E}^+$  and  $F_{las,S}^-$  are the laser energy fluxes of the direct (+) and reverse (-) laser beamlets at points  $E$  and  $S$ , respectively, on the cell boundary as obtained by linear interpolation from the neighbor vertices; for details see Fig. 4.19.

Note that without reflective boundaries along the  $x$ - and  $y$ -axes, there is only one direct incident beamlet in Eq. (5.17) for every individual laser beam; the energy fluxes in the other 3 conjugate beamlets are identically zero. If, however, reflective boundaries are present, it may be needed to treat up to 4 mutually conjugate beamlets for every individual laser beam (see section 5.3.2 below).

*b. Axial ( $r, z$ ) geometry, cylindrical or conical beams.* In the case of axial geometry with  $\text{iradial} = 1, 2$ , where the laser propagation direction  $\vec{\Omega}_{las}$  varies in space according to Eq. (5.9), the transfer equation for the laser energy flux  $F_{las}$  becomes

$$\Omega_{las,R} \frac{\partial}{\partial R} (RF_{las}) + \Omega_{las,Z} \frac{\partial}{\partial Z} (RF_{las}) = -k_{las} (RF_{las}), \quad (5.19)$$

where the rotation axis is assumed to be along  $Z$ , and  $R = \sqrt{X^2 + Y^2}$  is the cylindrical radius. Once our treatment has been restricted to the conical and cylindrical beams, where

$\Omega_{las,R}$  and  $\Omega_{las,Z}$  are constant, equation (5.19) becomes mathematically identical with the Cartesian equation (5.13) after we introduce a modified laser flux

$$\tilde{F}_{las} = \begin{cases} F_{las}, & \text{iradial} = 0, \\ RF_{las}, & \text{iradial} = 1, 2, \end{cases} \quad (5.20)$$

and use the notation  $x = R, y = Z$  for `iradial` = 1, and  $x = Z, y = R$  for `iradial` = 2. This applies also to the expression (5.16) for the heating power  $W_{las,i}$  of a mesh cell  $i$ , where in the axial case we have to replace  $F_{las}$  by the product  $RF_{las}$ . Note that, because of the normalization condition (5.6), the term  $\sqrt{\Omega_{las,x}^2 + \Omega_{las,y}^2}$  becomes identically unity.

As a result, the difference between the Cartesian and the axial geometries in treatment of the laser energy deposition is reduced to the following: where in the Cartesian case (`iradial` = 0) one has to provide the boundary condition for the incident energy flux density  $F_{las}$  [erg cm<sup>-2</sup> s<sup>-1</sup>], in the axial case (`iradial` = 1, 2) the corresponding boundary condition must be specified for the quantity  $\tilde{F}_{las} = RF_{las}$  [erg cm<sup>-1</sup> s<sup>-1</sup>]. After that, the same algorithm is applied in both cases to calculate the cell heating powers  $W_{las,i}$ .

## 2. Treatment of reflective boundaries

Because the transfer equation for the modified laser flux  $\tilde{F}_{las}$  is in all geometries `iradial` = 0, 1, 2 equivalent to the Cartesian case, treatment of the reflective boundaries also follows the logic of the Cartesian case. The latter, in particular, means that, for every individual laser beam, one generally has to process 4 mutually conjugate laser beamlets (a quad of beamlets; see section 4.5.2), and these beamlets must be processed in the following order: the incident beamlet  $\vec{\Omega}_{las}$ , the reflected (off the corresponding symmetry axis) beamlet  $\vec{\Omega}_{las}^*$ , the opposite-reflected  $-\vec{\Omega}_{las}^*$ , and the opposite-incident one  $-\vec{\Omega}_{las}$ .

However, there are two lesser aspects in which the laser beamlet quads differ from the beamlet quads of thermal radiation. Firstly, the laser beams are allowed to be coaxial, i.e. allowed to propagate exactly along the  $x$ - or  $y$ -axis (in practice, this is usually the most typical case). But if a laser beam is coaxial with the global reflective boundary, the amount of required donor-acceptor information (the rd-set data) is reduced by half because the quad beamlet, reflected off this boundary, coincides with the incident beam. Hence, the amount of computing work can be substantially reduced by skipping the superfluous beamlets in a given quad. This circumstance is taken into account by computing the rd-set not for all the 4 beamlets in a given laser beam `iblas` but for only `nocFblas(iblas)` beamlets, where  $1 \leq \text{nocFblas}(\text{iblas}) \leq 4$ . The value of `nocFblas(iblas)` is computed in the subroutine `LASCHECK`, file 'f02\_init.f'.

Secondly, the standard sequential processing order of mutually dependent beamlets within a given  $\Omega$ -block (see section 4.2.4) in the axial geometry is generally different from that in the Cartesian case. We bring the two ordering algorithms into accord by adopting the following scheme. Let `io` = 1, 2, 3, 4 be the basic sequential number of a particular beamlet in the quad such that

- `io` = 1 corresponds to the incident beamlet  $\vec{\Omega}_{las}$ ,
- `io` = 2 corresponds to the reflected beamlet  $\vec{\Omega}_{las}^*$ ,
- `io` = 3 corresponds to the opposite-reflected beamlet  $-\vec{\Omega}_{las}^*$ , and

`io = 4` corresponds to the opposite-incident beamlet  $-\vec{\Omega}_{las}$ .

Different types of reflective symmetry are distinguished by the value of flag `irflty0` computed in the subroutine `JORDSET`, file ‘`f09_rad.f`’ (see section 4.5.1). The convention in the RALEF code is such that in the Cartesian case the reflected beam  $\vec{\Omega}_{las}^*$  is obtained from the incident one  $\vec{\Omega}_{las}$

- by changing the sign of  $\Omega_{las,y}$ , when `irflty0 = 1` and `3`,
- by changing the sign of  $\Omega_{las,x}$ , when `irflty0 = 2`.

In both axial cases the reflected beam is obtained from the incident one by changing the sign of its  $\Omega_{las,R}$  component. Then the regular processing order will be

$$\text{io} = 1, \quad \text{or} \quad \text{io} = 1, 2, \quad \text{or} \quad \text{io} = 1, 2, 3, 4,$$

depending on the value of `nocFblas(iblas)`. In addition, there are three exceptions to this rule, all for `irflty0 = 3`, where either `iradial = 0` and  $\Omega_{las,y} = 0$ , or `iradial = 1` and  $\Omega_{las,x} = 0$ , or `iradial = 2` and  $\Omega_{las,y} = 0$ : in all these cases the processing order will be

$$\text{io} = 1, 3.$$

Summarily, all the possible variants of the laser beamlet processing order can be sorted out as follows.

**Case 0:** `irflty0 = 0`, possible only when `iradial = 0`. In this case one always deals with a cylindrical laser beam in the Cartesian geometry, for which

$$\Omega_{las,x}^2 + \Omega_{las,y}^2 \leq 1, \quad \text{nocFblas(iblas)} = 1. \quad (5.21)$$

The F-block (computation of the  $F_{las}$  field) in subroutine `LASDEPR` is executed for the single incident beamlet `io = 1`.

**Case 1:** `irflty0 = 1`, possible for `iradial = 0, 2` with a reflective boundary (or the rotation axis) along the  $x$ -axis; see Fig. 5.6.

This case splits into the following two subcases:

1a: a cylindrical laser beam with

$$\Omega_{las,y} = 0, \quad -1 \leq \Omega_{las,x} \leq +1 \quad (5.22)$$

propagates along the symmetry (rotation)  $x$ -axis; `nocFblas(iblas) = 1`, the F-block is executed for only the incident beamlet `io = 1`;

1b: either a cylindrical (for `iradial = 0`) or a conical (for `iradial = 2`) laser beam propagates along a vector with components

$$\Omega_{las,y} < 0, \quad -1 < \Omega_{las,x} < +1; \quad (5.23)$$

`nocFblas(iblas) = 2`, the F-block is executed for `io = 1, 2` (the incident and the reflected beamlets).

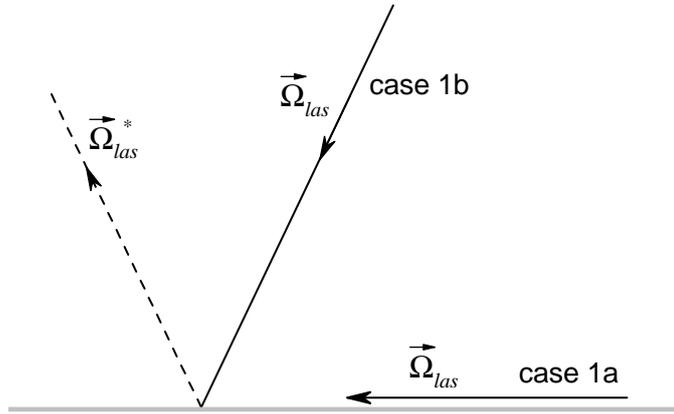


FIG. 5.6: Geometry of laser beamlets in the case of a single reflective boundary (or the rotation axis) along the  $x$ -axis.

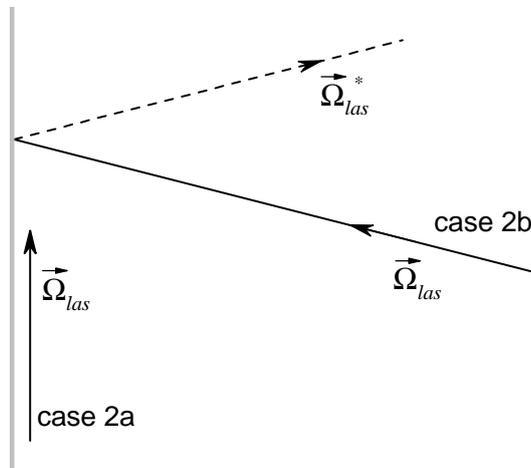


FIG. 5.7: Geometry of laser beamlets in the case of a single reflective boundary (rotation axis) along the  $y$ -axis.

**Case 2:** `irflty0 = 2`, possible for `iradial = 0, 1` with a reflective boundary (or the rotation axis) along the  $y$ -axis; see Fig. 5.7.

This case splits into the following two subcases:

2a: a cylindrical laser beam with

$$\Omega_{las,x} = 0, \quad -1 \leq \Omega_{las,y} \leq +1 \tag{5.24}$$

propagates along the symmetry (rotation)  $y$ -axis; `nocFblas(iblas) = 1`, the F-block is executed for only the incident beamlet `io = 1`;

2b: either a cylindrical (for `iradial = 0`) or a conical (for `iradial = 1`) laser beam propagates along a vector with components

$$\Omega_{las,x} < 0, \quad -1 < \Omega_{las,y} < +1; \tag{5.25}$$

$\text{nocFblas}(\text{iblas}) = 2$ , the F-block is executed for  $\text{io} = 1, 2$  (the incident and the reflected beamlets);

**Case 3:**  $\text{irflty0} = 3$ , possible for the values of  $\text{iradial} = 0, 1, 2$ , each of the  $x$ - and  $y$ -axis is either a reflective boundary or a rotational axis; see Fig. 5.8.

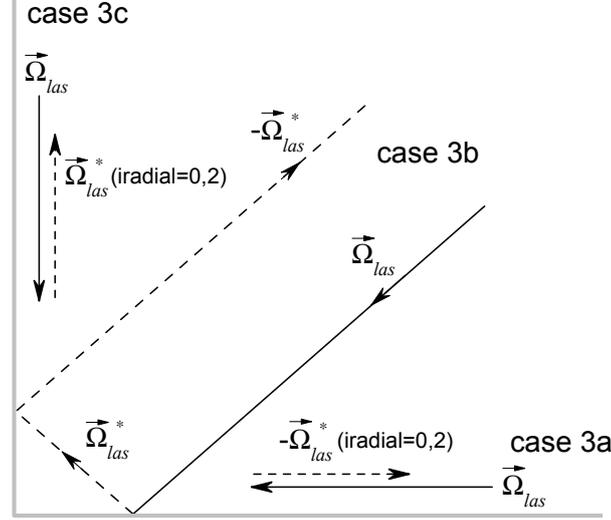


FIG. 5.8: Geometry of laser beamlets in the case of two mutually perpendicular reflective boundaries (or a reflective boundary perpendicular to the rotation axis).

This case splits into the following three subcases:

3a: either a cylindrical (for  $\text{iradial} = 0, 2$ ) or a conical (for  $\text{iradial} = 1$ ) laser beam with

$$\Omega_{las,y} = 0, \quad -1 \leq \Omega_{las,x} < 0 \quad (5.26)$$

propagates along the symmetry (rotation)  $x$ -axis;  $\text{nocFblas}(\text{iblas}) = 2$ , the F-block is executed for  $\text{io} = 1, 3$  (the incident and the opposite-reflected beamlets) when  $\text{iradial} = 0, 2$ , and for  $\text{io} = 1, 2$  (the incident and the reflected beamlets) when  $\text{iradial} = 1$ ;

3b: either a cylindrical ( $\text{iradial} = 0$ ) or a conical ( $\text{iradial} = 1, 2$ ) laser beam propagates along a vector with components

$$\Omega_{las,x} < 0, \quad \Omega_{las,y} < 0; \quad (5.27)$$

$\text{nocFblas}(\text{iblas}) = 4$ , the F-block is executed for  $\text{io} = 1, 2, 3, 4$  (the direct, the reflected, the opposite-reflected, and the opposite-direct beamlets);

3c: either a cylindrical ( $\text{iradial} = 0, 1$ ) or a conical ( $\text{iradial} = 2$ ) laser beam with

$$\Omega_{las,x} = 0, \quad -1 \leq \Omega_{las,y} < 0 \quad (5.28)$$

propagates along the symmetry (rotation)  $y$ -axis;  $\text{nocFblas}(\text{iblas}) = 2$ , the F-block is executed for  $\text{io} = 1, 2$  (the direct and the reflected beamlets) when  $\text{iradial} = 0, 2$ , and for  $\text{io} = 1, 3$  (the incident and the opposite-reflected beamlets) when  $\text{iradial} = 1$ .

3. Description of the LASDEPR subroutine

Deposition of energy by the laser beams is computed in subroutine LASDEPR, called by the main driver procedure HYDROCYC. Because, beside the laser deposition, other external sources of heat may be present, the principal task of the subroutine LASDEPR is to add the laser contribution to the final array  $qdepo(i)$  of the mass-specific energy deposition rate by external heat sources; the laser contribution should be added also to the block-specific  $wdepblk(iblk)$  and the global  $wdeptot$  heating powers by all external heat sources, that keep track of the global energy balance in the system. In addition, the corresponding laser-specific block-by-block,  $wlasblk(iblk)$ , and the total,  $wlastot$ , heating powers are computed in LASDEPR, as well as the total power  $hlasbcib(ib,iblk)$  of the laser energy flux across edge  $ib$  into block  $iblk$ , and the “nominal” power  $wlasnom$  of all incident laser beams summed together.

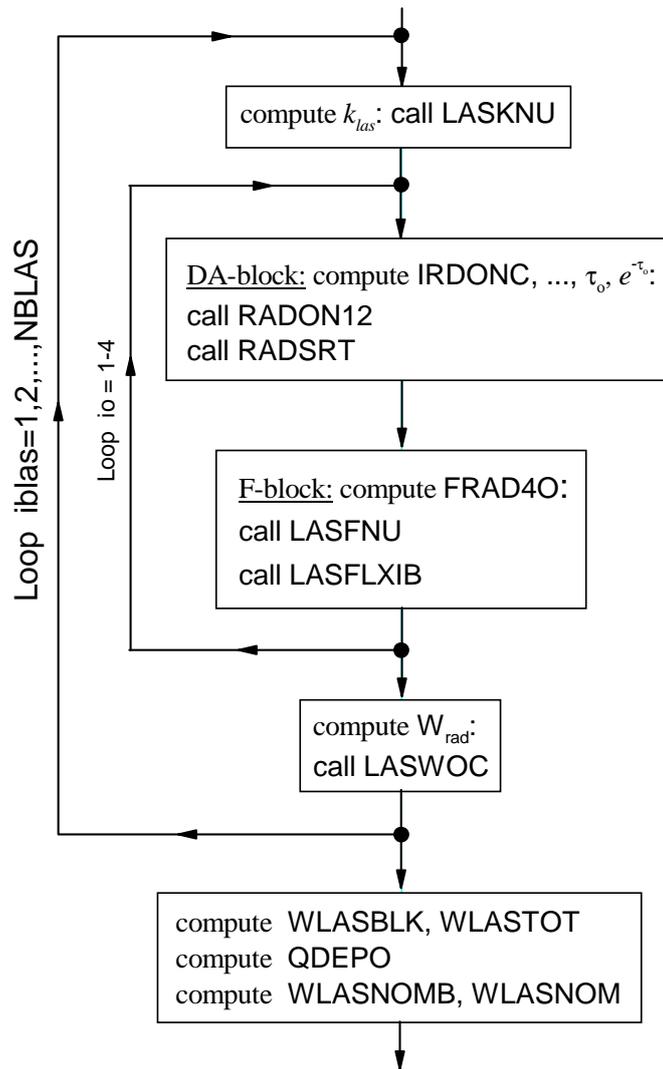


FIG. 5.9: Flowchart of subroutine LASDEPR.

If only one laser beam is present ( $nblas = 1$ ), the subroutine LASDEPR is executed in serial

mode by the master thread. If there are several laser beams (`nblas > 1`), the subroutine `LASDEPR` is executed in parallel mode, with different threads processing different laser beams in parallel.

The input parameters for the laser energy deposition are specified in the following way:

- The main control flag `iflasdep` (logical) is read from the `namelist/input/`: if `iflasdep=.false.` (the default value), then the subroutine `LASDEPR` is not called, and no laser energy deposition is computed. Starting from the RALEF code version `v11.07`, this flag is allowed to be changed at subsequent job restarts.
- All the other parameters of laser irradiation are defined by calling the subroutine `LASINPT` (file `'f10_taskinpt.f'`) after the `namelist/input/` has been read and `iflasdep=.true.` was found.
- The normalized temporal and spatial profiles of the irradiating laser beams must be programmed as functions `FLASPROT` and `FLASPROS` in file `'f10_taskinpt.f'`.

All the laser deposition parameters can be changed at subsequent job restarts by introducing the necessary changes into the subroutine `LASINPT`, the functions `FLASPROT` and `FLASPROS`, and by recompiling the code. The flowchart of subroutine `LASDEPR` is shown in Fig. 5.9.

#### 4. The hybrid refractive/reflective model of laser light transport

The hybrid model of laser light transport combines the approximation of geometric optics in strongly underdense plasma regions with the wave theory of light propagation across non-uniform one-dimensional plane-parallel plasma layers.

##### 1. General equations for monochromatic electromagnetic fields in dispersive media

*a. Maxwells' equations.* We assume that the magnetic permittivity is everywhere unity and  $\vec{B} = \vec{H}$ . Then the Maxwell equations in a continuous medium with non-zero polarizability in the absence of external currents and charges take the form [24, §77]

$$\nabla \times \vec{E} = -\frac{1}{c} \frac{\partial \vec{H}}{\partial t}, \quad \nabla \cdot \vec{D} = 0, \quad (5.29)$$

$$\nabla \times \vec{H} = \frac{1}{c} \frac{\partial \vec{D}}{\partial t}, \quad \nabla \cdot \vec{H} = 0. \quad (5.30)$$

For real  $\vec{E}$ ,  $\vec{H}$ ,  $\vec{B}$ , and  $\vec{D}$ , the principal equation of the energy balance reads

$$\frac{1}{4\pi} \left( \vec{E} \cdot \frac{\partial \vec{D}}{\partial t} + \vec{H} \cdot \frac{\partial \vec{B}}{\partial t} \right) + \nabla \cdot \left( \frac{c}{4\pi} \vec{E} \times \vec{H} \right) = 0. \quad (5.31)$$

The energy density (per unit volume) of the electromagnetic field is given by

$$U = \frac{1}{4\pi} \int \vec{E} \cdot d\vec{D} + \frac{\vec{H}^2}{8\pi}, \quad (5.32)$$

while

$$\vec{S} = \frac{c}{4\pi} \left( \vec{E} \times \vec{H} \right) \quad (5.33)$$

is the Poynting vector of the electromagnetic energy flux.

*b. Monochromatic fields.* Further on, we assume that the electromagnetic field of every particular laser beam is strictly monochromatic with the angular frequency  $\omega$ . In such a case, it is convenient to factor out the harmonic time dependence of the field variables as

$$\begin{aligned}\vec{E}(t, \vec{r}) &= \vec{E}_\omega(\vec{r}) e^{-i\omega t}, \\ \vec{D}(t, \vec{r}) &= \vec{D}_\omega(\vec{r}) e^{-i\omega t}, \\ \vec{H}(t, \vec{r}) &= \vec{H}_\omega(\vec{r}) e^{-i\omega t},\end{aligned}\tag{5.34}$$

where the space-dependent monochromatic components  $\vec{E}_\omega$ ,  $\vec{D}_\omega$ , and  $\vec{H}_\omega$  are now *complex* vectors, while the frequency of oscillation  $\omega$  is a *real* number. Below it is always assumed that the time  $\omega^{-1}$  is much shorter than any time scale of hydro motion or kinetic relaxation in a plasma.

When the plasma is an isotropic medium (which we assume always to be the case), the electric induction  $\vec{D}_\omega$  is proportional to the electric field  $\vec{E}_\omega$

$$\vec{D}_\omega = \varepsilon_\omega \vec{E}_\omega,\tag{5.35}$$

where

$$\varepsilon_\omega = \varepsilon(\omega) = \varepsilon'_\omega + i\varepsilon''_\omega\tag{5.36}$$

is the scalar complex dielectric permittivity at frequency  $\omega$ . When calculating the propagation and absorption of laser light,  $\varepsilon_\omega$  is assumed to be a function of time  $t$  and spatial coordinate  $\vec{r}$ . Its real and imaginary parts must obey the following general relationships [24, §77]:

$$\varepsilon'(-\omega) = \varepsilon'(\omega), \quad \varepsilon''(-\omega) = -\varepsilon''(\omega).\tag{5.37}$$

A specific semi-empirical formula for  $\varepsilon(\omega)$ , appropriate for plasmas and metals, is given in subsection 5.7.1. In the limit of low frequency  $\omega$ , the dielectric permittivity of a metal is given by

$$\varepsilon(\omega) = i \frac{4\pi\sigma}{\omega}.\tag{5.38}$$

At very high frequencies (exceeding all the atomic frequencies), the permittivity of any material becomes

$$\varepsilon(\omega) = 1 - \frac{\omega_p^2}{\omega^2}, \quad \omega_p^2 = \frac{4\pi n_e e^2}{m_e},\tag{5.39}$$

where  $n_e$  is the total number of all electrons (bound + free) per unit volume.

With the ansatz (5.34), Maxwells' equations (5.29), (5.30) are reduced to two wave equations [24, §88]

$$\nabla^2 \vec{E}_\omega - \nabla \left( \nabla \cdot \vec{E}_\omega \right) + \frac{\omega^2}{c^2} \varepsilon_\omega \vec{E}_\omega = 0,\tag{5.40}$$

$$\nabla^2 \vec{H}_\omega + \varepsilon_\omega^{-1} \nabla \varepsilon_\omega \times \left( \nabla \times \vec{H}_\omega \right) + \frac{\omega^2}{c^2} \varepsilon_\omega \vec{H}_\omega = 0.\tag{5.41}$$

Note that, because  $\nabla \cdot \vec{D}_\omega = \nabla \cdot (\varepsilon_\omega \vec{E}_\omega) = 0$ , Eq. (5.40) can be rewritten as

$$\nabla^2 \vec{E}_\omega + \nabla \left[ (\varepsilon_\omega^{-1} \nabla \varepsilon_\omega) \cdot \vec{E}_\omega \right] + \frac{\omega^2}{c^2} \varepsilon_\omega \vec{E}_\omega = 0.\tag{5.42}$$

The electric and the magnetic fields are related to each other as

$$\frac{i\omega}{c} \vec{H}_\omega = \nabla \times \vec{E}_\omega, \quad -\frac{i\omega}{c} \varepsilon_\omega \vec{E}_\omega = \nabla \times \vec{H}_\omega.\tag{5.43}$$

*c. Energy dissipation.* The energy dissipation by monochromatic fields is calculated as the time average over the rapid oscillations with frequency  $\omega$ : if  $F(\omega)$  is such an oscillating quantity, then

$$\langle F \rangle_t = \lim_{T \rightarrow \infty} \int_{-T}^{+T} F(t) dt. \quad (5.44)$$

Evidently,  $\langle e^{i\omega t} \rangle_t = 0$ . Then, the law of energy conservation for monochromatic fields can be written as [24, §80]

$$\nabla \cdot \langle \vec{S}_\omega \rangle_t + \langle Q_\omega \rangle_t = 0, \quad (5.45)$$

where

$$\langle \vec{S}_\omega \rangle_t = \frac{c}{8\pi} \Re \left( \vec{E}_\omega \times \vec{H}_\omega^* \right), \quad \langle Q_\omega \rangle_t = \frac{\omega}{8\pi} \Im(\varepsilon_\omega) \left| \vec{E}_\omega \right|^2 = \frac{\omega}{8\pi} \varepsilon_\omega'' \left| \vec{E}_\omega \right|^2. \quad (5.46)$$

The positive sign (under normal conditions) of the energy dissipation requires that  $\varepsilon_\omega'' > 0$ . As a further implication, the absorption coefficient of a planar monochromatic wave is defined as

$$k_a = \frac{\langle Q_\omega \rangle_t}{\left| \langle \vec{S}_\omega \rangle_t \right|} \quad [\text{cm}^{-1}]. \quad (5.47)$$

The energy density of the *quasi-monochromatic* light is given by [24, §83]

$$U_\omega = \frac{1}{v_g} \left| \langle \vec{S}_\omega \rangle_t \right|, \quad v_g = \frac{d\omega}{dk}. \quad (5.48)$$

*d. Planar harmonic waves in quasi-homogeneous media.* Here we assume that

$$\varepsilon_\omega = \varepsilon'_\omega + i\varepsilon''_\omega \quad (5.49)$$

is constant in space. Then we look for harmonic (periodic in space) solutions of the form

$$\vec{E}_\omega = \vec{E}_0 e^{i\vec{k} \cdot \vec{r}}, \quad \vec{H}_\omega = \vec{H}_0 e^{i\vec{k} \cdot \vec{r}}, \quad (5.50)$$

where  $\vec{E}_0$ ,  $\vec{H}_0$  and  $\vec{k}$  are complex constants. With (5.50) Maxwells' equations are reduced to

$$\vec{k} \times \vec{E}_0 = \frac{\omega}{c} \vec{H}_0, \quad \vec{k} \cdot \vec{E}_0 = 0, \quad (5.51)$$

$$\vec{k} \times \vec{H}_0 = -\frac{\omega}{c} \varepsilon_\omega \vec{E}_0, \quad \vec{k} \cdot \vec{H}_0 = 0, \quad (5.52)$$

and yield the dispersion relation

$$c^2 \vec{k}^2 = \omega^2 \varepsilon_\omega. \quad (5.53)$$

Having assumed that the real and imaginary components of the complex wave vector  $\vec{k}$  are colinear, we calculate

$$\vec{k} \stackrel{\text{def}}{=} \frac{c\vec{k}}{\omega} = (n' + in'')\vec{\Omega}, \quad (5.54)$$

where  $\vec{\Omega}$  is the unit real vector in the direction of wave propagation, and

$$n' + in'' = \sqrt{\varepsilon_\omega} \quad (5.55)$$

with

$$n' = \left[ \frac{\sqrt{(\varepsilon'_\omega)^2 + (\varepsilon''_\omega)^2} + \varepsilon'_\omega}{2} \right]^{1/2}, \quad n'' = \frac{\varepsilon''_\omega}{2n'} \quad (5.56)$$

is the complex refraction index. Note that both  $n'$  and  $n''$  are essentially non-negative.

From the above equations one calculates

$$\vec{S}_\omega = \frac{c}{8\pi} n' |\vec{E}_0|^2 \vec{\Omega}, \quad Q_\omega = \frac{\omega}{8\pi} \varepsilon''_\omega |\vec{E}_0|^2, \quad (5.57)$$

which leads to the following expression for absorption (attenuation) coefficient

$$k_a = \left( \frac{\omega}{c} \right) \frac{\varepsilon''_\omega}{n'} = \frac{2\omega}{c} n''. \quad (5.58)$$

When the Drude model is applied to a plasma, it leads to the following simple expression for the complex dielectric permittivity [25, §44, 48], [26, §5.1]

$$\varepsilon_\omega = 1 - \frac{\omega_{pe}^2}{\omega(\omega + i\nu_{ei})}, \quad (5.59)$$

where  $\omega_{pe}$  is the plasma frequency for free electrons, and  $\nu_{ei}$  is the electron-ion collision frequency. Once the classical expression (5.189) [25, §44] for the collision frequency is substituted into Eq. (5.59), one recovers (in the limit of  $\nu_{ei} \ll \omega$  and to the accuracy of the value of the Coulomb logarithm) from Eq. (5.58) the classical Kramers inverse-bremsstrahlung absorption coefficient (2.38) (with  $h\nu \equiv \hbar\omega$ ). For more details, see subsections 5 7 2 and 5 7 1 below.

## 2. Propagation of a monochromatic planar wave across a non-uniform planar layer

Consider propagation of a monochromatic electromagnetic wave with an angular frequency  $\omega$  across a planar layer of isotropic dispersive medium (a plasma layer) with a finite thickness. The complex dielectric permittivity  $\varepsilon_\omega = \varepsilon_\omega(z)$  of the plasma is assumed to vary along the  $z$  axis, perpendicular to the plasma layer at  $z_0 < z < z_\infty$ . The outside regions  $z < z_0$  and  $z > z_\infty$  are assumed to be uniform and have real constant values of  $\varepsilon_\omega = \varepsilon_0$  and  $\varepsilon_\omega = \varepsilon_\infty$ , respectively. When the outside regions are vacuum,  $\varepsilon_0 = \varepsilon_\infty = 1$ .

Let a planar monochromatic wave fall onto the considered plasma layer from the half-space  $z < z_0$  with the propagation vector

$$\vec{k}_0 = \frac{\omega}{c} \vec{\kappa}_0 = \frac{\omega}{c} (0, \kappa_{0y}, \kappa_{0z}) \quad (5.60)$$

in the  $yz$  plane. Since  $\varepsilon_0$  is real, the dimensionless wave vector  $\vec{\kappa}_0$  at  $z < z_0$  is also real. Equation (5.53) yields

$$\kappa_0 = \sqrt{\kappa_{0y}^2 + \kappa_{0z}^2} = \sqrt{\varepsilon_0} = n_0. \quad (5.61)$$

The angle of incidence  $\theta_0$  (with respect to the  $z$  axis) is defined by

$$\sin \theta_0 = \frac{\kappa_{0y}}{\sqrt{\kappa_{0y}^2 + \kappa_{0z}^2}}. \quad (5.62)$$

a. *The Helmholtz equation for s-polarization.* In the considered situation, the vector structure of the electromagnetic field in the entire space splits into two mutually fully independent waves (normal modes) [24, §88], [27, §1.6], [26, §5.5]: the s-polarized wave with

$$\text{s-polarization: } \vec{E}_\omega = (E_{\omega x}, 0, 0), \quad \vec{H}_\omega = (0, H_{\omega y}, H_{\omega z}) \quad (5.63)$$

and the p-polarized wave with

$$\text{p-polarization: } \vec{E}_\omega = (0, E_{\omega y}, E_{\omega z}), \quad \vec{H}_\omega = (H_{\omega x}, 0, 0). \quad (5.64)$$

For the s-polarization, the wave equation (5.40) becomes

$$\frac{\partial^2 E_{\omega x}}{\partial y^2} + \frac{\partial^2 E_{\omega x}}{\partial z^2} + \frac{\omega^2}{c^2} \varepsilon_\omega E_{\omega x} = 0 \quad (5.65)$$

(the *Helmholtz equation*), with the magnetic field given by

$$H_{\omega y} = -i \frac{c}{\omega} \frac{\partial E_{\omega x}}{\partial z}, \quad H_{\omega z} = i \frac{c}{\omega} \frac{\partial E_{\omega x}}{\partial y}. \quad (5.66)$$

It is convenient to introduce dimensionless coordinates

$$\bar{z} = \frac{\omega}{c} z = \frac{2\pi z}{\lambda}, \quad \bar{y} = \frac{\omega}{c} y = \frac{2\pi y}{\lambda}. \quad (5.67)$$

With respect to these variables, Eqs. (5.65) and (5.66) become

$$\frac{\partial^2 E_{\omega x}}{\partial \bar{y}^2} + \frac{\partial^2 E_{\omega x}}{\partial \bar{z}^2} + \varepsilon_\omega E_{\omega x} = 0, \quad (5.68)$$

$$H_{\omega y} = -i \frac{\partial E_{\omega x}}{\partial \bar{z}}, \quad H_{\omega z} = i \frac{\partial E_{\omega x}}{\partial \bar{y}}. \quad (5.69)$$

Because  $\varepsilon_\omega$  in Eq. (5.68) does not depend on  $\bar{y}$ , the dependence of  $E_{\omega x}$  on  $\bar{y}$  can be factored out as

$$E_{\omega x}(\bar{y}, \bar{z}) = \exp(i\kappa_y \bar{y}) E(\bar{z}), \quad (5.70)$$

$$H_{\omega y}(\bar{y}, \bar{z}) = \exp(i\kappa_y \bar{y}) H(\bar{z}), \quad (5.71)$$

with

$$\kappa_y = \kappa_{0y} = n_0 \sin \theta_0 \quad (5.72)$$

being a real constant. Finally, the Helmholtz equation for the s-polarized wave is reduced to

$$\frac{d^2 E}{d\bar{z}^2} + (\varepsilon_\omega - n_0^2 \sin^2 \theta_0) E = 0, \quad (5.73)$$

$$H = -i \frac{dE}{d\bar{z}}. \quad (5.74)$$

The general boundary conditions for the monochromatic fields  $\vec{E}_\omega$  and  $\vec{H}_\omega$  are deduced from the Maxwells' equations (5.43): for  $\vec{E}_\omega$  and  $\vec{H}_\omega$  to remain finite, their tangential components must be everywhere continuous. For the s-polarization this implies continuity of the

$E(\bar{z})$  and  $H(\bar{z})$  in Eqs. (5.73) and (5.74). The  $z$  component of the time-averaged Poynting vector (5.46) becomes

$$S_z \equiv \langle S_{\omega z} \rangle_t = \frac{c}{8\pi} \Re (E_{\omega x} H_{\omega y}^*) = \frac{c}{8\pi} \Re (EH^*). \quad (5.75)$$

The rate of energy dissipation per unit volume is

$$Q \equiv \langle Q_\omega \rangle_t = -\frac{dS_z}{dz} = \frac{\omega}{8\pi} \varepsilon_\omega'' |E|^2. \quad (5.76)$$

*b. The Helmholtz equation for p-polarization.* With respect to the dimensionless coordinates (5.67), the Helmholtz equations for the p-polarized wave, analogous to Eqs. (5.68), (5.69), become

$$\frac{\partial^2 H_{\omega x}}{\partial \bar{y}^2} + \frac{\partial^2 H_{\omega x}}{\partial \bar{z}^2} - \frac{d \ln \varepsilon_\omega}{d\bar{z}} \frac{\partial H_{\omega x}}{\partial \bar{z}} + \varepsilon_\omega H_{\omega x} = 0, \quad (5.77)$$

$$E_{\omega y} = \frac{i}{\varepsilon_\omega} \frac{\partial H_{\omega x}}{\partial \bar{z}}, \quad E_{\omega z} = -\frac{i}{\varepsilon_\omega} \frac{\partial H_{\omega x}}{\partial \bar{y}}. \quad (5.78)$$

Similarly to the s-wave, one can assume

$$H_{\omega x}(\bar{y}, \bar{z}) = \exp(i\kappa_y \bar{y}) H(\bar{z}), \quad (5.79)$$

$$E_{\omega y}(\bar{y}, \bar{z}) = \exp(i\kappa_y \bar{y}) E(\bar{z}), \quad (5.80)$$

where  $\kappa_y$  is real and given by Eq. (5.72). The  $z$  component of the electric field is given by

$$E_{\omega z}(\bar{y}, \bar{z}) = \frac{n_0 \sin \theta_0}{\varepsilon_\omega} \exp(i\kappa_y \bar{y}) H(\bar{z}). \quad (5.81)$$

Functions  $H(\bar{z})$  and  $E(\bar{z})$  are found from the 1D Helmholtz equation

$$\frac{d^2 H}{d\bar{z}^2} - \frac{d \ln \varepsilon_\omega}{d\bar{z}} \frac{dH}{d\bar{z}} + (\varepsilon_\omega - n_0^2 \sin^2 \theta_0) H = 0, \quad (5.82)$$

$$E = \frac{i}{\varepsilon_\omega} \frac{dH}{d\bar{z}}. \quad (5.83)$$

Note that Eq. (5.82) can be cast in the form

$$\frac{d}{d\bar{z}} \left( \frac{1}{\varepsilon_\omega} \frac{dH}{d\bar{z}} \right) + \left( 1 - \frac{n_0^2 \sin^2 \theta_0}{\varepsilon_\omega} \right) H = 0. \quad (5.84)$$

Again, the boundary conditions are reduced to the continuity of  $E(\bar{z})$  and  $H(\bar{z})$  in Eqs. (5.82) and (5.83).

Finally, the Poynting vector  $S_z$  and the volume-specific rate of power dissipation  $Q$  are given by

$$S_z = -\frac{c}{8\pi} \Re (H_{\omega x} E_{\omega y}^*) = -\frac{c}{8\pi} \Re (HE^*), \quad (5.85)$$

$$Q = -\frac{dS_z}{dz} = \frac{\omega}{8\pi} \varepsilon_\omega'' \left( |E|^2 + n_0^2 \sin^2 \theta_0 \left| \frac{H}{\varepsilon_\omega} \right|^2 \right). \quad (5.86)$$

Note that the two terms in the parentheses in the expression for  $Q$  represent the two components of the electric field  $|E_{\omega y}|^2$  and  $|E_{\omega z}|^2$ .

c. *Numerical solution by the finite-analytical method.* For numerical solution of Eqs. (5.73), (5.74), (5.82), and (5.83) on a finite interval  $\bar{z}_0 \leq \bar{z} \leq \bar{z}_N$ , we split this interval into  $N$  mesh cells  $[\bar{z}_{j-1}, \bar{z}_j]$ ,  $j = 1, \dots, N$ , as shown in Fig. 5.10, and assume the complex dielectric permittivity  $\varepsilon_\omega(\bar{z})$  to be constant within each of the  $j$ -th mesh zone,

$$\varepsilon_\omega(\bar{z}) = \varepsilon_j, \quad \bar{z}_{j-1} \leq \bar{z} \leq \bar{z}_j, \quad j = 1, \dots, N. \quad (5.87)$$

The wave vector

$$\vec{k} = \frac{\omega}{c} \vec{\kappa} \quad (5.88)$$

of the incident planar wave is assumed to lie in the  $yz$ -plane at an angle  $\theta_0$  with respect to the  $z$ -axis. Outside the treated domain  $\bar{z}_0 \leq \bar{z} \leq \bar{z}_N$  we assume real values of the dielectric permittivity

$$\varepsilon_\omega(\bar{z} < \bar{z}_0) = \varepsilon_0 = n_0^2, \quad \varepsilon_\omega(\bar{z} > \bar{z}_N) = \varepsilon_{N+1} \equiv \varepsilon_\infty = n_\infty^2, \quad \Im(\varepsilon_0) = \Im(\varepsilon_\infty) = 0. \quad (5.89)$$

This condition guarantees positiveness of the energy dissipation everywhere within  $\bar{z}_0 \leq \bar{z} \leq \bar{z}_N$ . Note that  $\varepsilon_\infty$  can, in principle, be negative, with the refractive index  $n_\infty$  purely imaginary.

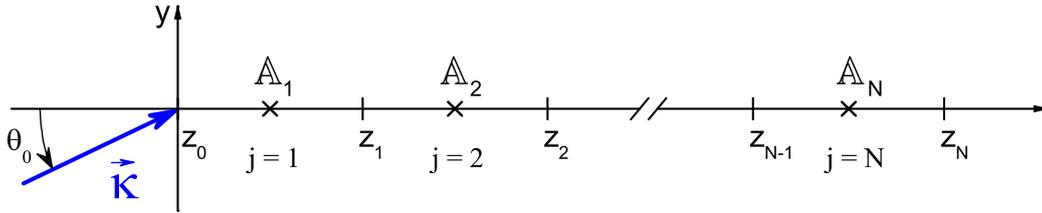


FIG. 5.10: Discrete mesh for numerical solution of the 1D Helmholtz equations (5.73), (5.74), (5.82), and (5.83). The blue arrow indicates the dimensionless wave vector  $\vec{\kappa} = (c/\omega)\vec{k}$  of the incident wave.

Our numerical scheme belongs to the class of *finite-analytical* numerical methods, where, instead of approximating the derivatives with finite differences (as in the classical *finite-difference* approach), we approximate the treated domain  $\bar{z}_0 \leq \bar{z} \leq \bar{z}_N$  as a stack of  $N$  uniform layers and solve the differential equations (5.73), (5.74), (5.82), (5.83) exactly on this stack. The corresponding analytical solution can be written as

$$\begin{cases} E_j = A_j^+ e^{i\kappa_j(\bar{z}-\bar{z}_{j-1})} + A_j^- e^{-i\kappa_j(\bar{z}-\bar{z}_{j-1})}, \\ H_j = \kappa_j [A_j^+ e^{i\kappa_j(\bar{z}-\bar{z}_{j-1})} - A_j^- e^{-i\kappa_j(\bar{z}-\bar{z}_{j-1})}], \end{cases} \quad \text{s-wave,} \quad (5.90)$$

$$\begin{cases} H_j = A_j^+ e^{i\kappa_j(\bar{z}-\bar{z}_{j-1})} + A_j^- e^{-i\kappa_j(\bar{z}-\bar{z}_{j-1})}, \\ E_j = -\frac{\kappa_j}{\varepsilon_j} [A_j^+ e^{i\kappa_j(\bar{z}-\bar{z}_{j-1})} - A_j^- e^{-i\kappa_j(\bar{z}-\bar{z}_{j-1})}], \end{cases} \quad \text{p-wave,} \quad (5.91)$$

where

$$\kappa_j \equiv \kappa_j' + i\kappa_j'' = \sqrt{\varepsilon_j - n_0^2 \sin^2 \theta_0}, \quad (5.92)$$

and  $A_j^\pm$  are integration constants to be determined from the boundary conditions. To include the boundary conditions at  $\bar{z} = \bar{z}_0$  and  $\bar{z} = \bar{z}_N$  into the general scheme, it is convenient to extend the variation of index  $j$  from  $j = 1, 2, \dots, N$  to  $j = 0, 1, 2, \dots, N, N + 1$  by assuming

$$\bar{z}_{-1} = \bar{z}_0, \quad \kappa_0 = n_0 \cos \theta_0, \quad \kappa_{N+1} \equiv \kappa_\infty = \sqrt{n_\infty^2 - n_0^2 \sin^2 \theta_0} \quad (5.93)$$

in Eqs. (5.90)–(5.92).

Having introduced a 2-vector

$$\mathbb{A}_j = \left\| \begin{array}{c} A_j^+ \\ A_j^- \end{array} \right\|, \quad j = 0, 1, \dots, N + 1, \quad (5.94)$$

the boundary conditions at  $\bar{z} = \bar{z}_0$  and  $\bar{z} = \bar{z}_N$  can be written as

$$\mathbb{A}_0 = \left\| \begin{array}{c} 1 \\ r \end{array} \right\|, \quad \mathbb{A}_{N+1} = \left\| \begin{array}{c} p \\ 0 \end{array} \right\|, \quad (5.95)$$

where  $r$  and  $p$  are unknown complex amplitudes of the reflected and transmitted waves; the amplitude of the incident wave is assumed to be 1. The boundary conditions at  $\bar{z} = \bar{z}_j$  can be cast in the form of linear recurrent relations

$$\mathbb{A}_{j+1} = \mathbb{G}_j \mathbb{A}_j = \mathbb{H}_j \mathbb{A}_0, \quad j = 0, 1, \dots, N, \quad (5.96)$$

where

$$\mathbb{H}_j = \mathbb{G}_j \cdot \mathbb{G}_{j-1} \cdot \dots \cdot \mathbb{G}_0, \quad (5.97)$$

$$\mathbb{G}_j = \left\| \begin{array}{cc} \frac{1}{2}(1 + \gamma_j)e^{\Delta_j} & \frac{1}{2}(1 - \gamma_j)e^{-\Delta_j} \\ \frac{1}{2}(1 - \gamma_j)e^{\Delta_j} & \frac{1}{2}(1 + \gamma_j)e^{-\Delta_j} \end{array} \right\|, \quad j = 0, 1, \dots, N, \quad (5.98)$$

$$\Delta_0 = 0, \quad \Delta_j = i\kappa_j \Delta \bar{z}_j \equiv i\kappa_j (\bar{z}_j - \bar{z}_{j-1}), \quad j = 1, \dots, N, \quad (5.99)$$

$$\gamma_j = \left\{ \begin{array}{ll} \frac{\kappa_j}{\kappa_{j+1}}, & s\text{-wave,} \\ \frac{\kappa_j}{\kappa_{j+1}} \frac{\varepsilon_{j+1}}{\varepsilon_j}, & p\text{-wave,} \end{array} \right\} \quad j = 0, 1, \dots, N. \quad (5.100)$$

Note that

$$\det \mathbb{G}_j = \gamma_j, \quad \det \mathbb{H}_j = \prod_{k=0}^j \gamma_k. \quad (5.101)$$

Because matrices  $\mathbb{G}_j$  contain both the growing and falling exponents, their successive multiplication may easily cause an arithmetic overflow. To avoid such overflows, one can factor out the growing exponents by introducing the corresponding scaled matrices  $\tilde{\mathbb{G}}_j$  and  $\tilde{\mathbb{H}}_j$ , defined as

$$\mathbb{G}_j = e^{\sigma_j} \tilde{\mathbb{G}}_j, \quad \mathbb{H}_j = e^{\Sigma_j} \tilde{\mathbb{H}}_j, \quad \tilde{\mathbb{H}}_j = \tilde{\mathbb{G}}_j \cdot \tilde{\mathbb{G}}_{j-1} \cdot \dots \cdot \tilde{\mathbb{G}}_0, \quad (5.102)$$

where

$$\Sigma_j = \sigma_0 + \sigma_1 + \dots + \sigma_j, \quad (5.103)$$

$$\sigma_0 = 0, \quad \sigma_j = \kappa_j'' (\bar{z}_j - \bar{z}_{j-1}) \geq 0, \quad j = 1, \dots, N. \quad (5.104)$$

The two unknown amplitudes  $r$  and  $p$  are found from two linear equations

$$\begin{pmatrix} p \\ 0 \end{pmatrix} = e^{\Sigma_N} \tilde{\mathbb{H}}_N \begin{pmatrix} 1 \\ r \end{pmatrix}, \quad (5.105)$$

which yield

$$r = -\frac{\tilde{\mathbb{H}}_{N,21}}{\tilde{\mathbb{H}}_{N,22}}, \quad p = -\frac{e^{-\Sigma_N} \prod_{j=0}^N \gamma_j}{\tilde{\mathbb{H}}_{N,22}}, \quad (5.106)$$

where  $\tilde{\mathbb{H}}_{N,jk}$  is the  $jk$ -th element of matrix  $\tilde{\mathbb{H}}_N$ . Note that in the limit of a very large “optical thickness”  $\Sigma_N \rightarrow \infty$  all the elements of the scaled matrices  $\tilde{\mathbb{G}}_j$  and  $\tilde{\mathbb{H}}_j$  vary within a limited range and, as a consequence, one automatically recovers the correct finite limit for  $r$  together with a zero value for the transmission amplitude  $p$  — which is not easy to achieve when solving the corresponding boundary-value problem with any finite-difference scheme. Once the reflection amplitude  $r$  is known, all the field amplitudes  $A_j^\pm$  are calculated from the recurrent relations (5.96).

To calculate the energy deposition in each mesh zone, we introduce the normalized  $z$ -component of the Poynting vector at zone interfaces

$$\bar{S}_j = \frac{8\pi}{c} \frac{S_{z=z_j}}{\cos \theta_0} \cdot \begin{cases} n_0^{-1}, & s\text{-wave,} \\ n_0, & p\text{-wave.} \end{cases} \quad (5.107)$$

Since at  $\bar{z} = \bar{z}_0$  we have

$$\bar{S}_0 = 1 - |r|^2, \quad (5.108)$$

the reflected fraction of the incident energy flux is given by

$$f_{rfl} = |r|^2. \quad (5.109)$$

Then the transmitted energy fraction is calculated as

$$f_{tr} \equiv \bar{S}_{N+1} = \frac{|p|^2}{\cos \theta_0} \cdot \begin{cases} n_0^{-1} \Re(\kappa_\infty), & s\text{-wave,} \\ n_0 \Re(\kappa_\infty/\varepsilon_\infty), & p\text{-wave.} \end{cases} \quad (5.110)$$

For the fraction of the incident energy flux  $f_{aj}$  absorbed in the  $j$ -th mesh zone we obtain

$$f_{aj} \equiv \bar{S}_{j-1} - \bar{S}_j = f_{aj,sec} + f_{aj,osc}, \quad (5.111)$$

$$f_{aj,sec} = \left[ |A_j^+|^2 \left( 1 - e^{-2\kappa_j' \Delta \bar{z}_j} \right) + |A_j^-|^2 \left( e^{2\kappa_j' \Delta \bar{z}_j} - 1 \right) \right] \cdot \begin{cases} \frac{\Re(\kappa_j)}{n_0 \cos \theta_0}, & s\text{-wave,} \\ \frac{n_0 \Re(\kappa_j/\varepsilon_j)}{\cos \theta_0}, & p\text{-wave,} \end{cases} \quad (5.112)$$

$$f_{aj,osc} = \Im \left[ A_j^+ A_j^{-*} \left( e^{2i\kappa_j' \Delta \bar{z}_j} - 1 \right) \right] \cdot \begin{cases} \frac{2\Im(\kappa_j)}{n_0 \cos \theta_0}, & s\text{-wave,} \\ \frac{2n_0 \Im(\kappa_j/\varepsilon_j)}{\cos \theta_0}, & p\text{-wave,} \end{cases} \quad (5.113)$$

where  $f_{aj,sec}$  is the secular (i.e. monotonically increasing with  $\Delta\bar{z}_j$ ) term, and  $f_{aj,osc}$  the oscillating (with the increasing  $\Delta\bar{z}_j$ ) term. The above finite-analytical scheme is conservative in the sense that it obeys the law of energy conservation

$$f_{rfl} + f_{tr} + \sum_{j=1}^N f_{aj} = 1. \quad (5.114)$$

## 5. Test problems for laser deposition: exact solutions

### 1. Absorption on a step profile (metallic surface) between two half-spaces

Consider a step profile of the dielectric permittivity

$$\varepsilon(z) = \begin{cases} 1, & z < 0, \\ \varepsilon_\infty, & z > 0, \end{cases} \quad (5.115)$$

between vacuum with  $\varepsilon_0 = 1$  and a half-space with a complex dielectric constant

$$\varepsilon_\infty = \varepsilon'_\infty + i\varepsilon''_\infty. \quad (5.116)$$

For typical metals one has  $\varepsilon'_\infty \ll -1$ , with  $|\varepsilon'_\infty|$  reaching the values of  $\simeq 100$ . In particular, for liquid tin (Sn) at  $T \approx 500^\circ\text{C}$  [28] one finds

$$\varepsilon_\infty = \begin{cases} (25.0 + 26.8i)^2 = -93.24 + 1340i, & \hbar\omega = 0.1 \text{ eV } (\lambda = 12.4 \text{ } \mu\text{m}), \\ (4.0 + 8.4i)^2 = -54.56 + 67.2i, & \hbar\omega = 1.2 \text{ eV } (\lambda = 1.03 \text{ } \mu\text{m}). \end{cases} \quad (5.117)$$

Reflection of a planar monochromatic wave, incident on a half-space at an angle  $\theta_0$ , is easily calculated by applying the solution of section 5.4.2 (with the boundary conditions of  $E$  and  $H$  being continuous across the interface  $z = 0$ ), which leads to the well known Fresnel formulae. In particular, the absorption fraction is calculated as

$$f_a = 1 - |r|^2, \quad r = \frac{\gamma_0 - 1}{\gamma_0 + 1}, \quad \gamma_0 = \begin{cases} \cos \theta_0 / \kappa_\infty, & s\text{-wave}, \\ (\varepsilon_\infty / \kappa_\infty) \cos \theta_0, & p\text{-wave}, \end{cases} \quad (5.118)$$

where

$$\kappa_\infty \equiv \kappa'_\infty + i\kappa''_\infty = \sqrt{\varepsilon_\infty - \sin^2 \theta_0}. \quad (5.119)$$

After some algebra, one obtains

$$f_{a,s}(\theta_0) = \frac{4\kappa'_\infty \cos \theta_0}{(\kappa'_\infty + \cos \theta_0)^2 + (\kappa''_\infty)^2}, \quad s\text{-wave}, \quad (5.120)$$

$$f_{a,p}(\theta_0) = \frac{4\mu'_\infty \cos \theta_0}{(1 + \mu'_\infty \cos \theta_0)^2 + (\mu''_\infty)^2 \cos^2 \theta_0}, \quad p\text{-wave}, \quad (5.121)$$

where

$$\mu_\infty \equiv \mu'_\infty + i\mu''_\infty = \frac{\varepsilon_\infty}{\kappa_\infty} = \kappa_\infty + \frac{\sin^2 \theta_0}{\kappa_\infty}, \quad (5.122)$$

$$\mu'_\infty = \kappa'_\infty \left[ 1 + \frac{\sin^2 \theta_0}{(\kappa'_\infty)^2 + (\kappa''_\infty)^2} \right], \quad \mu''_\infty = \kappa''_\infty \left[ 1 - \frac{\sin^2 \theta_0}{(\kappa'_\infty)^2 + (\kappa''_\infty)^2} \right], \quad (5.123)$$

$$\kappa'_\infty = \Re \left( \sqrt{\varepsilon_\infty - \sin^2 \theta_0} \right), \quad \kappa''_\infty = \Im \left( \sqrt{\varepsilon_\infty - \sin^2 \theta_0} \right). \quad (5.124)$$

The angular dependence  $f_a(\theta_0)$  of the absorption by liquid tin surface for the dielectric permittivity (5.117) is shown in Fig. 5.11.

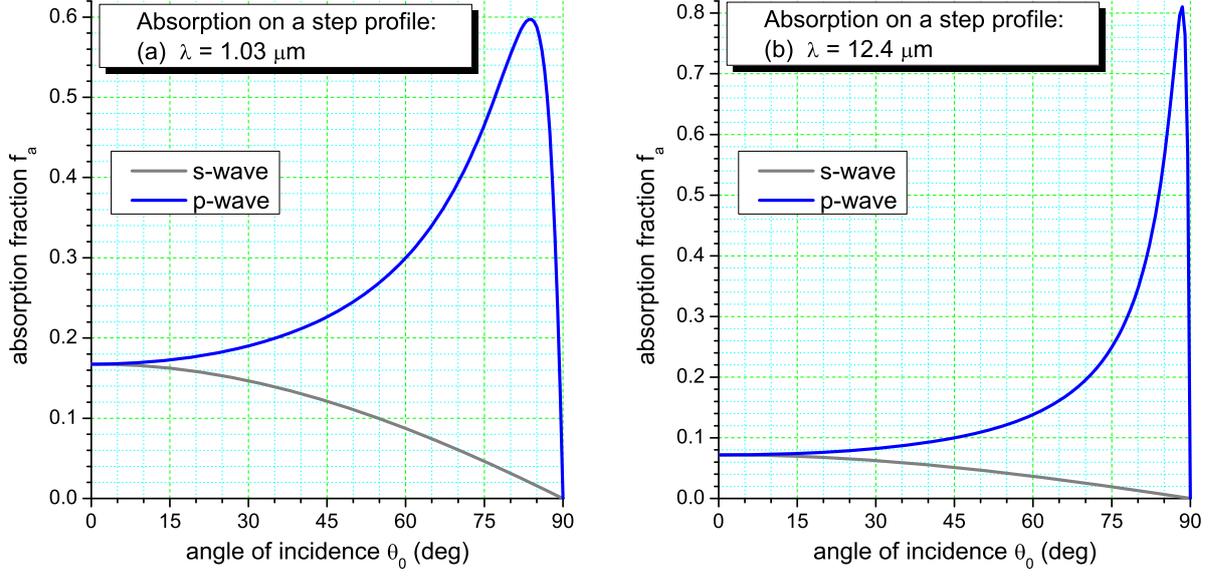


FIG. 5.11: Absorption coefficient  $f_a = 1 - |r|^2$  by a half-space of liquid tin with a step  $\varepsilon$  profile as a function of the incidence angle  $\theta_0$  for two values of the incident wavelength  $\lambda$ .

In a practically important limit of  $|\varepsilon'_\infty| \gg 1$  one readily ascertains that the angular dependence  $f_a(\theta_0)$  for the  $s$ -wave becomes a simple proportionality to  $\cos \theta_0$  because the weak dependence of  $\kappa'_\infty$ ,  $\kappa''_\infty$ , and of the denominator in Eq. (??) on  $\theta_0$  can be neglected.

For the  $p$ -polarization, of special interest is the limit of

$$\varepsilon_\infty = -|\varepsilon'_\infty| + i\varepsilon''_\infty, \quad |\varepsilon'_\infty| \gg 1, \quad \varepsilon''_\infty \ll 1 \quad (5.125)$$

because, when  $\varepsilon'_\infty(z)$  passes through zero with a finite derivative  $d\varepsilon'_\infty/dz$ , the absorption coefficient of the  $p$ -polarized light remains finite at any  $\theta_0 > 0$  (the *resonant absorption*) even when  $\varepsilon''_\infty \rightarrow 0$  [24, §88]. In the limit (5.125) equations (5.121)–(5.124) yield

$$\mu'_\infty \approx \kappa'_\infty \approx \frac{1}{2} \frac{\varepsilon''_\infty}{\sqrt{|\varepsilon'_\infty|}} \rightarrow 0, \quad \mu''_\infty \approx \kappa''_\infty \approx \sqrt{|\varepsilon'_\infty|} \gg 1, \quad (5.126)$$

$$f_{a,p}(\theta_0) \approx \frac{2\varepsilon''_\infty}{\sqrt{|\varepsilon'_\infty|}} \frac{\cos \theta_0}{1 + |\varepsilon'_\infty| \cos^2 \theta_0} \rightarrow 0. \quad (5.127)$$

From Eq. (5.127) we learn that (i) the resonant absorption of the  $p$ -wave vanishes on a step  $\varepsilon(z)$  profile when  $\varepsilon''_\infty \rightarrow 0$ , and (ii) the absorption coefficient  $f_{a,p}(\theta_0)$  attains maximum at the incidence angle  $\theta_0 = \theta_{0m}$  given by

$$\sin \left( \frac{\pi}{2} - \theta_{0m} \right) = \cos \theta_{0m} = \frac{1}{\sqrt{|\varepsilon'_\infty|}}. \quad (5.128)$$

Note that this estimate agrees well with the plots of  $f_{a,p}(\theta_0)$  in Fig. 5.11.

2. *S-wave transmission through a 3-layer slab*

a. *The solution.* Here we assume that all the distances are measured in units of  $c/\omega$ . Consider a planar slab, consisting of three adjacent layers (strata) of thicknesses  $h_1$ ,  $h_2$  and  $h_3$ , with the dielectric permittivity given by (see Fig. 5.12)

$$\varepsilon = \begin{cases} \varepsilon_0 = n_0^2, & z < z_1 = -h_1, & \text{stratum } j = 0, \\ \varepsilon_1 = (n'_1 + in''_1)^2, & -h_1 = z_1 < z < z_2 = 0, & \text{stratum } j = 1, \\ \varepsilon_2(z) = \varepsilon_{20} + z(\varepsilon_{21} - \varepsilon_{20})/h_2, & 0 = z_2 < z < z_3 = h_2, & \text{stratum } j = 2, \\ \varepsilon_3 = (n'_3 + in''_3)^2, & h_2 = z_3 < z < z_4 = h_2 + h_3, & \text{stratum } j = 3, \\ \varepsilon_4 = n_\infty^2, & h_2 + h_3 = z_4 < z, & \text{stratum } j = 4. \end{cases} \quad (5.129)$$

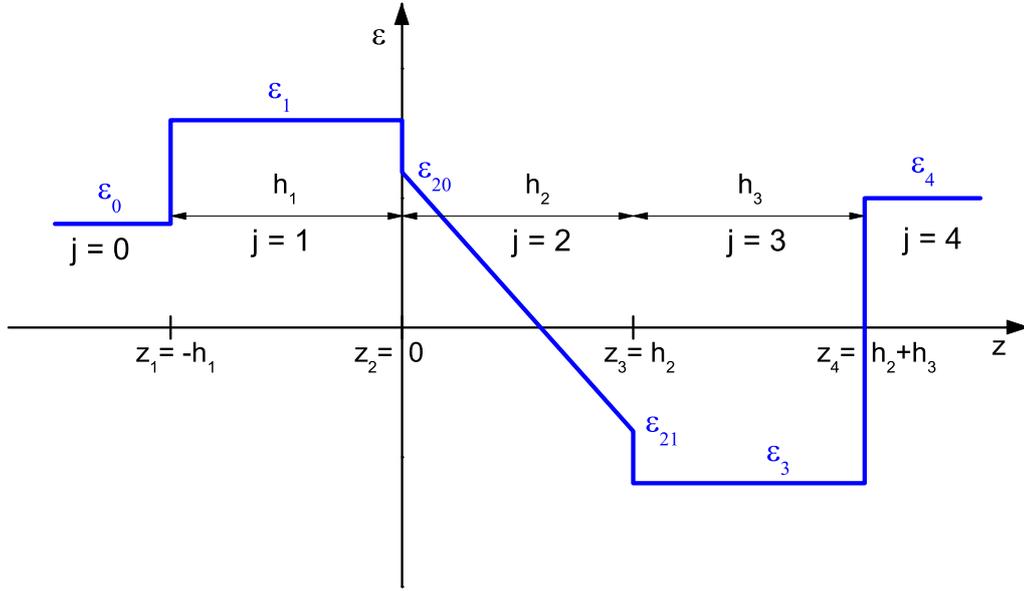


FIG. 5.12: Structure of a 3-layer planar region for testing reflection and transmission of the s- and p-polarized waves.

The general solution to the Helmholtz equations (5.73), (5.74) in the five above defined strata is given by

$$\begin{cases} E = A_0^+ e^{+i\kappa_{z0}(z-z_1)} + A_0^- e^{-i\kappa_{z0}(z-z_1)}, \\ H = \kappa_{z0} [A_0^+ e^{+i\kappa_{z0}(z-z_1)} - A_0^- e^{-i\kappa_{z0}(z-z_1)}], \end{cases} \quad z \in \text{stratum } j = 0, \quad (5.130)$$

$$\begin{cases} E = A_j^+ e^{+i\kappa_{zj}(z-z_j)} + A_j^- e^{-i\kappa_{zj}(z-z_j)}, \\ H = \kappa_{zj} [A_j^+ e^{+i\kappa_{zj}(z-z_j)} - A_j^- e^{-i\kappa_{zj}(z-z_j)}], \end{cases} \quad z \in \text{strata } j = 1, 3, 4, \quad (5.131)$$

$$\begin{cases} E = A_2^+ \text{Ai}(\xi) + A_2^- \text{Bi}(\xi), \\ H = i\kappa_{z2} [A_2^+ \text{Ai}'(\xi) + A_2^- \text{Bi}'(\xi)], \end{cases} \quad z \in \text{stratum } j = 2. \quad (5.132)$$

Here  $A_j^+$  and  $A_j^-$  are the integration constants in stratum  $j$ ,

$$\kappa_{zj} = \kappa'_{zj} + i\kappa''_{zj} = \begin{cases} n_0 \cos \theta_0, & j = 0, \\ \sqrt{\varepsilon_j - n_0^2 \sin^2 \theta_0}, & j = 1, 3, 4, \\ [(\varepsilon_{21} - \varepsilon_{20})/h_2]^{1/3}, & j = 2, \end{cases} \quad \kappa'_{zj} \geq 0, \kappa''_{zj} \geq 0, \quad (5.133)$$

$$\xi = -\frac{\kappa_{z2} h_2}{\varepsilon_{21} - \varepsilon_{20}} \left( \varepsilon_{20} + \frac{\varepsilon_{21} - \varepsilon_{20}}{h_2} z - n_0^2 \sin^2 \theta_0 \right). \quad (5.134)$$

Change of variable (5.134) in stratum  $j = 2$  reduces the Helmholtz equation (5.73) to the Airy-Stokes equation

$$\frac{d^2 E}{d\xi^2} - \xi E = 0, \quad (5.135)$$

whose two linearly independent solutions are the Airy functions  $\text{Ai}(\xi)$  and  $\text{Bi}(\xi)$ . Below we make use of the identity

$$W\{\text{Ai}(\xi), \text{Bi}(\xi)\} = \begin{vmatrix} \text{Ai}(\xi) & \text{Bi}(\xi) \\ \text{Ai}'(\xi) & \text{Bi}'(\xi) \end{vmatrix} = \frac{1}{\pi}. \quad (5.136)$$

The coefficients  $A_j^\pm$  are found from the boundary conditions at  $z \rightarrow \pm\infty$  and at the interfaces between the neighboring strata. It is convenient to introduce 2-vectors

$$\mathbb{A}_j = \begin{vmatrix} A_j^+ \\ A_j^- \end{vmatrix}, \quad j = 0, 1, 2, 3, 4, \quad (5.137)$$

and the coupling  $2 \times 2$  matrices  $\mathbb{G}_j$  such that

$$\mathbb{A}_{j+1} = \mathbb{G}_j \mathbb{A}_j. \quad (5.138)$$

In the left half-space  $z < z_1$  we assume to have an incident wave with a unit amplitude  $A_0^+ = 1$  and a reflected wave with an unknown amplitude  $A_0^- = r$ , i.e.

$$\mathbb{A}_0 = \begin{vmatrix} 1 \\ r \end{vmatrix}. \quad (5.139)$$

Then, from the conditions of continuity of  $E$  and  $H$  between the corresponding strata we find

$$\mathbb{G}_0 = \frac{1}{2} \begin{vmatrix} 1 + \gamma_0 & 1 - \gamma_0 \\ 1 - \gamma_0 & 1 + \gamma_0 \end{vmatrix}, \quad (5.140)$$

$$\mathbb{G}_1 = \pi \begin{vmatrix} +[\text{Bi}'(\xi_1) - \gamma_1 \text{Bi}(\xi_1)] e^{\Delta_1} & +[\text{Bi}'(\xi_1) + \gamma_1 \text{Bi}(\xi_1)] e^{-\Delta_1} \\ -[\text{Ai}'(\xi_1) - \gamma_1 \text{Ai}(\xi_1)] e^{\Delta_1} & -[\text{Ai}'(\xi_1) + \gamma_1 \text{Ai}(\xi_1)] e^{-\Delta_1} \end{vmatrix}, \quad (5.141)$$

$$\mathbb{G}_2 = \frac{1}{2} \begin{vmatrix} \text{Ai}(\xi_2) + \gamma_2 \text{Ai}'(\xi_2) & \text{Bi}(\xi_2) + \gamma_2 \text{Bi}'(\xi_2) \\ \text{Ai}(\xi_2) - \gamma_2 \text{Ai}'(\xi_2) & \text{Bi}(\xi_2) - \gamma_2 \text{Bi}'(\xi_2) \end{vmatrix}, \quad (5.142)$$

$$\mathbb{G}_3 = \frac{1}{2} \begin{vmatrix} (1 + \gamma_3) e^{\Delta_3} & (1 - \gamma_3) e^{-\Delta_3} \\ (1 - \gamma_3) e^{\Delta_3} & (1 + \gamma_3) e^{-\Delta_3} \end{vmatrix}, \quad (5.143)$$

where

$$\xi_1 = -\frac{\kappa_{z2}h_2}{\varepsilon_{21} - \varepsilon_{20}} (\varepsilon_{20} - n_0^2 \sin^2 \theta_0), \quad (5.144)$$

$$\xi_2 = -\frac{\kappa_{z2}h_2}{\varepsilon_{21} - \varepsilon_{20}} (\varepsilon_{21} - n_0^2 \sin^2 \theta_0), \quad (5.145)$$

$$\gamma_0 = \frac{\kappa_{z0}}{\kappa_{z1}}, \quad \gamma_1 = \frac{\kappa_{z1}}{i\kappa_{z2}}, \quad \gamma_2 = \frac{i\kappa_{z2}}{\kappa_{z3}}, \quad \gamma_3 = \frac{\kappa_{z3}}{\kappa_{z4}}, \quad (5.146)$$

$$\Delta_j = i\kappa_{zj}h_j = -\kappa_{zj}''h_j + i\kappa_{zj}'h_j, \quad j = 1, 3. \quad (5.147)$$

The boundary condition at  $z = z_4$  is the transmitted wave

$$\mathbb{A}_4 = \begin{Bmatrix} p \\ 0 \end{Bmatrix} \quad (5.148)$$

with an unknown amplitude  $p$ . The matrix equation, which allows to calculate the values of  $r$  and  $p$ , is

$$\begin{Bmatrix} p \\ 0 \end{Bmatrix} = \mathbb{G}_3\mathbb{G}_2\mathbb{G}_1\mathbb{G}_0 \begin{Bmatrix} 1 \\ r \end{Bmatrix} = \mathbb{H}_3 \begin{Bmatrix} 1 \\ r \end{Bmatrix}, \quad (5.149)$$

where

$$\mathbb{H}_3 = \begin{Bmatrix} H_{3,11}, & H_{3,12} \\ H_{3,21}, & H_{3,22} \end{Bmatrix} = \mathbb{G}_3\mathbb{G}_2\mathbb{G}_1\mathbb{G}_0. \quad (5.150)$$

From Eq. (5.149) we obtain

$$r = -\frac{H_{3,21}}{H_{3,22}}, \quad p = \frac{\det(\mathbb{H}_3)}{H_{3,22}}. \quad (5.151)$$

Note that the determinants of the above defined matrices  $\mathbb{G}_j$  and  $\mathbb{H}_3$  are equal to

$$\det(\mathbb{G}_0) = \gamma_0, \quad \det(\mathbb{G}_1) = -2\pi\gamma_1, \quad \det(\mathbb{G}_2) = -\frac{\gamma_2}{2\pi}, \quad \det(\mathbb{G}_3) = \gamma_3, \quad (5.152)$$

$$\det(\mathbb{H}_3) = \gamma_0\gamma_1\gamma_2\gamma_3 = \frac{\kappa_{z0}}{\kappa_{z4}} = \frac{n_0 \cos \theta_0}{\sqrt{n_\infty^2 - n_0^2 \sin^2 \theta_0}}. \quad (5.153)$$

In practice, the boundary-value problem for the Helmholtz equation must often be solved over a layer with very large optical thickness ( $10^3$  and more). Because matrices  $\mathbb{G}_j$  contain both the growing and falling exponents, their successive multiplication may easily cause arithmetic overflow. To avoid such overflows, one can factor out the growing exponents by introducing the corresponding scaled matrices  $\tilde{\mathbb{G}}_j$  and  $\tilde{\mathbb{H}}_j$

$$\mathbb{G}_j = e^{\sigma_j} \tilde{\mathbb{G}}_j, \quad \mathbb{H}_3 = e^{\Sigma_3} \tilde{\mathbb{H}}_3, \quad \sigma_j \geq 0, \quad \Sigma_3 = \sum_j \sigma_j \geq 0. \quad (5.154)$$

For the same reason, instead of the normal functions Ai and Bi, we will use scaled Airy functions  $\tilde{\text{Ai}}$  and  $\tilde{\text{Bi}}$ , and their first derivatives, defined by

$$\text{Ai}(\xi) = e^{-\zeta'} \tilde{\text{Ai}}(\xi), \quad \text{Ai}'(\xi) = e^{-\zeta'} \tilde{\text{Ai}}'(\xi), \quad (5.155)$$

$$\text{Bi}(\xi) = e^{|\zeta'|} \tilde{\text{Bi}}(\xi), \quad \text{Bi}'(\xi) = e^{|\zeta'|} \tilde{\text{Bi}}'(\xi), \quad \zeta' = \Re \left( \frac{2}{3} \xi^{3/2} \right). \quad (5.156)$$

Thus scaled Airy functions have  $|\widetilde{\text{Ai}}(\xi)| \sim 1$ ,  $|\widetilde{\text{Bi}}(\xi)| \sim 1$  over the entire complex plane of  $\xi$  values and, importantly, are continuous across the  $\arg(\xi) = \pm\pi$  cut. Finally, we get

$$\widetilde{\mathbb{G}}_0 = \mathbb{G}_0, \tag{5.157}$$

$$\widetilde{\mathbb{G}}_1 = \pi \left\| \begin{array}{l} \left[ \widetilde{\text{Bi}}'(\xi_1) - \gamma_1 \widetilde{\text{Bi}}(\xi_1) \right] e^{\Delta_1 + |\zeta'_1| - \sigma_1}, \quad \left[ \widetilde{\text{Bi}}'(\xi_1) + \gamma_1 \widetilde{\text{Bi}}(\xi_1) \right] e^{-\Delta_1 + |\zeta'_1| - \sigma_1} \\ \left[ -\widetilde{\text{Ai}}'(\xi_1) + \gamma_1 \widetilde{\text{Ai}}(\xi_1) \right] e^{\Delta_1 - \zeta'_1 - \sigma_1}, \quad \left[ -\widetilde{\text{Ai}}'(\xi_1) - \gamma_1 \widetilde{\text{Ai}}(\xi_1) \right] e^{-\Delta_1 - \zeta'_1 - \sigma_1} \end{array} \right\|, \tag{5.158}$$

$$\widetilde{\mathbb{G}}_2 = \frac{1}{2} \left\| \begin{array}{l} \left[ \widetilde{\text{Ai}}(\xi_2) + \gamma_2 \widetilde{\text{Ai}}'(\xi_2) \right] e^{-\zeta'_2 - |\zeta'_2|}, \quad \widetilde{\text{Bi}}(\xi_2) + \gamma_2 \widetilde{\text{Bi}}'(\xi_2) \\ \left[ \widetilde{\text{Ai}}(\xi_2) - \gamma_2 \widetilde{\text{Ai}}'(\xi_2) \right] e^{-\zeta'_2 - |\zeta'_2|}, \quad \widetilde{\text{Bi}}(\xi_2) - \gamma_2 \widetilde{\text{Bi}}'(\xi_2) \end{array} \right\|, \tag{5.159}$$

$$\widetilde{\mathbb{G}}_3 = \frac{1}{2} \left\| \begin{array}{l} (1 + \gamma_3) e^{\Delta_3 - \sigma_3}, \quad (1 - \gamma_3) e^{-\Delta_3 - \sigma_3} \\ (1 - \gamma_3) e^{\Delta_3 - \sigma_3}, \quad (1 + \gamma_3) e^{-\Delta_3 - \sigma_3} \end{array} \right\|, \tag{5.160}$$

$$\zeta'_1 = \Re\left(\frac{2}{3}\xi_1^{3/2}\right), \quad \zeta'_2 = \Re\left(\frac{2}{3}\xi_2^{3/2}\right), \tag{5.161}$$

$$\sigma_0 = 0, \quad \sigma_1 = \kappa''_{z1} h_1 + |\zeta'_1|, \quad \sigma_2 = |\zeta'_2|, \quad \sigma_3 = \kappa''_{z3} h_3. \tag{5.162}$$

In terms of the scaled values  $\widetilde{H}_{ij}$ , the reflected and transmitted amplitudes become

$$r = -\frac{\widetilde{H}_{3,21}}{\widetilde{H}_{3,22}}, \quad p = \frac{e^{-\Sigma_3} n_0 \cos \theta_0}{\widetilde{H}_{3,22} \sqrt{n_\infty^2 - n_0^2 \sin^2 \theta_0}}. \tag{5.163}$$

Because Maxwells' equations are linear, the normalization of the electric field  $E$  is arbitrary. Hence, it is convenient to define a normalized  $z$  component of the Poynting vector

$$\bar{S}_z(z) = \frac{\Re(EH^*)}{n_0 \cos \theta_0}, \quad \bar{S}_z = 1 - |r|^2 \quad \text{in stratum } j = 0, \tag{5.164}$$

so that in the general case

$$S_z(z) = F_0 \bar{S}_z(z), \tag{5.165}$$

where  $F_0$  is the problem-specific incident energy flux along the  $z$ -direction. Then the power dissipated per unit volume is given by

$$Q(z) = -\frac{\omega}{c} \frac{dS_z}{dz} = F_0 \frac{\omega}{c} \bar{Q}(z), \quad \bar{Q}(z) = -\frac{d\bar{S}_z}{dz} = \frac{\varepsilon''_\omega}{n_0 \cos \theta_0} |E|^2, \tag{5.166}$$

where  $E$  is calculated with the specific normalization (5.139). Note that the factor  $\omega/c$  after the first equality sign in Eq. (5.166) is due to the fact that  $z$  is assumed to be measured in  $c/\omega$ .

Finally, the transmitted power fraction is calculated as

$$\bar{S}_z(z_4) = \begin{cases} 0, & n_\infty^2 \leq n_0^2 \sin^2 \theta_0, \\ |p|^2 \frac{\sqrt{n_\infty^2 - n_0^2 \sin^2 \theta_0}}{n_0 \cos \theta_0} = \frac{e^{-2\Sigma_3} n_0 \cos \theta_0}{|\widetilde{H}_{3,22}|^2 \sqrt{n_\infty^2 - n_0^2 \sin^2 \theta_0}}, & n_\infty^2 > n_0^2 \sin^2 \theta_0. \end{cases} \tag{5.167}$$

*b. Specific examples.* As an illustrative example for the above described exact solution, consider reflection of a planar wave from the surface of liquid tin as a function of the incidence angle  $\theta_0$  for three different profiles of the dielectric permittivity  $\varepsilon$ : (i) a step profile between vacuum with  $\varepsilon_0 = n_0 = 1$  and a metallic half-space with

$$\varepsilon = \varepsilon_\infty = (n'_\infty + in''_\infty)^2, \quad n'_\infty = 25.0, \quad n''_\infty = 26.8, \quad (5.168)$$

(ii) a linear variation of  $\varepsilon$  between  $\varepsilon_0$  and  $\varepsilon_\infty$  over a distance of  $0.1\lambda$  ( $h_2 = 0.2\pi$ ) on top of a half-space of liquid tin, and (iii) a linear variation between  $\varepsilon_0$  and  $\varepsilon_\infty$  over a distance of  $100\lambda$  ( $h_2 = 200\pi$ ) on top of a half-space of liquid tin. The above numerical values for the real,  $n'_\infty$ , and imaginary,  $n''_\infty$ , parts of the refractive index of liquid tin at  $T \approx 500^\circ\text{C}$  are taken from Ref. 28 for  $\hbar\omega = 0.1$  eV ( $\lambda = 12.4 \mu\text{m}$ ). The calculated exact dependences of the absorption coefficient  $f_a(\theta_0) = 1 - f_{rfl}(\theta_0)$  are shown in Fig. 5.13. Note that a seemingly insignificant smearing of the sharp permittivity (i.e. electron density) profile over a distance of  $0.1\lambda$  leads to an increase of the absorption coefficient  $f_a$  by more than a factor of 2.

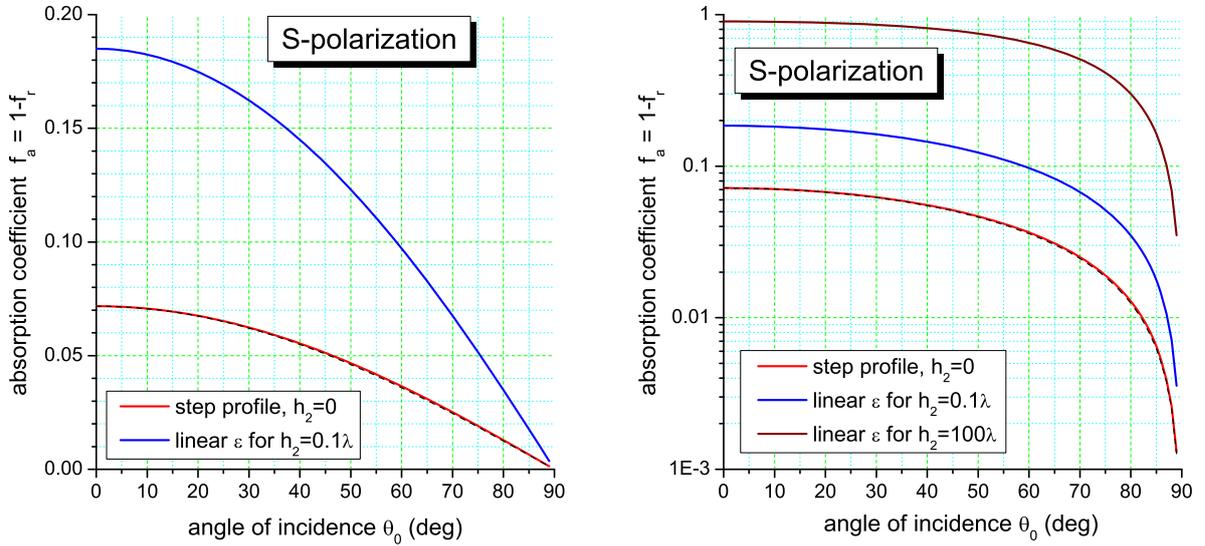


FIG. 5.13: Absorption coefficient  $f_a = 1 - |r|^2$  by a half-space of liquid tin as a function of the incidence angle  $\theta_0$  for a step  $\varepsilon$  profile (red), and two linearly ramped  $\varepsilon$  profiles over distances of  $0.1\lambda$  (blue) and  $100\lambda$  (wine). The black dashed curve is an approximation to the exact Fresnel formula (5.120) with the  $a \cos \theta_0$  function.

Because for  $|\Re(\varepsilon_\infty)| \gg 1$  the weak dependence of  $\kappa'_\infty$ ,  $\kappa''_\infty$ , and of the denominator in Eq. (5.120) on  $\theta_0$  can be neglected,  $f_a$  becomes simply proportional to  $\cos \theta_0$  — as is illustrated with the black dashed curve in Fig. 5.13. For specific applications to EUV lithography, the latter implies that the *s*-polarized component of a sub-picosecond prepulse with a very high contrast would deposit its energy along the surface of a spherical tin droplet in proportion to  $\cos^2 \theta$ , where  $\theta$  is the polar angle with respect to the laser axis (provided that the droplet radius  $R_{dr} \gg \lambda$  and the incident laser flux is constant across the focal spot).

Figure 5.14 illustrates how the reflection coefficient  $f_{rfl}$  of liquid tin at  $\lambda = 12.4 \mu\text{m}$  changes when the step spatial profile of  $\varepsilon$  is smeared by a linear ramp of variable extension  $h_2$

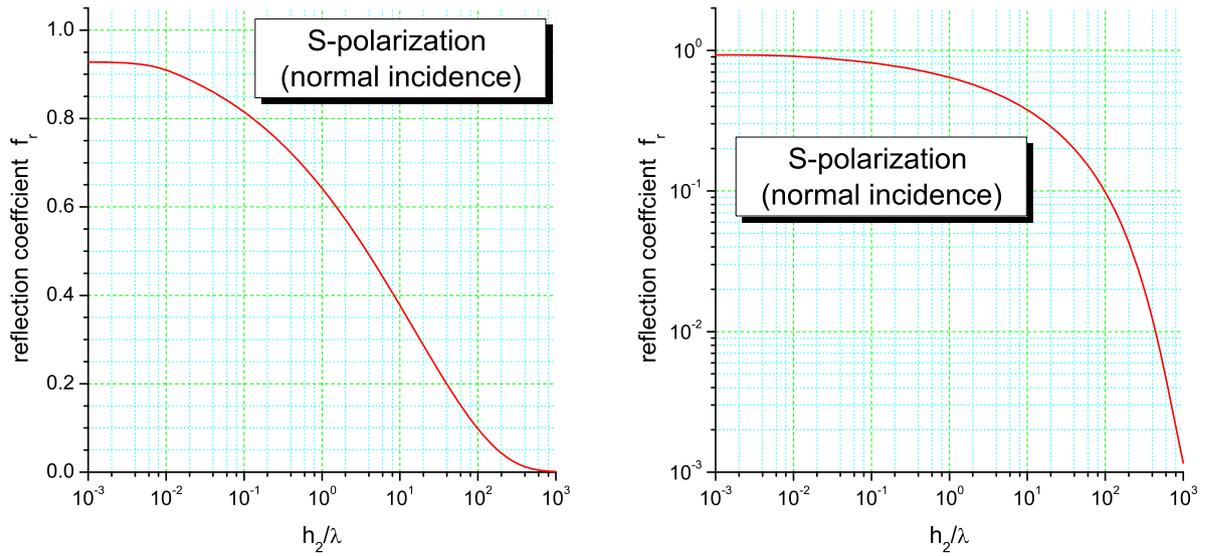


FIG. 5.14: Reflection coefficient  $f_r = |r|^2$  by a half-space of liquid tin as a function of the extension  $h_2$  of the linear ramp of  $\epsilon$  between vacuum and  $\epsilon_\infty$  in Eq. (5.168) for the case of normal incidence ( $\theta_0 = 0$ ).

(in units of the laser wavelength  $\lambda$ ). One sees that a spatial variation of  $\epsilon$  over a scale as short as  $h_2 = 0.1\lambda$  results in about a twofold increase of the absorption coefficient  $f_a = 1 - f_r f_l$ .

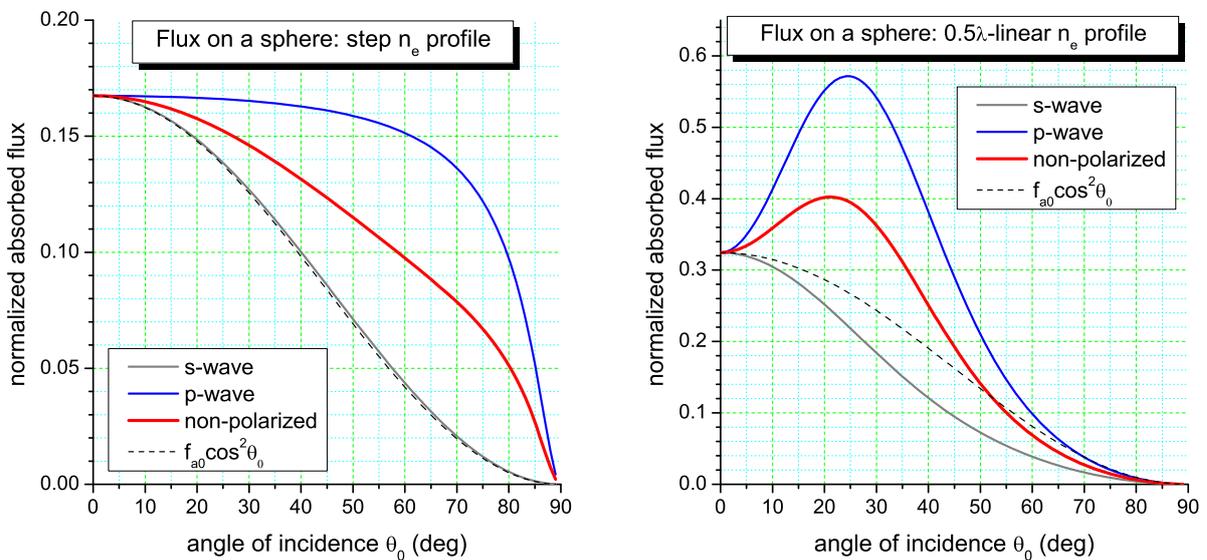


FIG. 5.15: Irradiation flux on a surface of a sphere as a function of the incidence angle  $\theta_0 = 0$ .

## 6. Test problems for laser deposition: the RALEF results

In all the RALEF runs discussed below, the laser wavelength  $\lambda$  was fixed at a value  $\lambda = 1 \mu\text{m}$ . However, because the dielectric permittivity was assigned separately — i.e. was fully decoupled from the  $\lambda = 2\pi c/\omega$  value, this value is physically irrelevant and serves only as an appropriate scale for other lengths and distances. The dimensionless quantities  $\bar{z}$  and  $\bar{h}_j$ , used in section 5.5 without the bar, are related to the corresponding dimensional quantities here and below as

$$\bar{z} = \frac{z\omega}{c} = 2\pi \frac{z}{\lambda}, \quad \bar{h}_j = 2\pi \frac{h_j}{\lambda}. \quad (5.169)$$

The independent variable  $z$ , used by derivation of the exact solutions, becomes the  $x$  coordinate in the RALEF system of notation.

For transition from the geometric optics to 1D wave equation, the criterion

$$\bar{n}_e + \beta_{wo} \lambda |\nabla \bar{n}_e| > \alpha_{gop} \quad (5.170)$$

is used, where  $\bar{n}_e = n_e/n_{e,cr}$  is the free electron density normalized to the critical density

$$n_{e,cr} = \frac{m_e \omega^2}{4\pi e^2}, \quad (5.171)$$

while for parameters  $\beta_{wo}$  and  $\alpha_{gop}$  the default values  $\beta_{wo} = 1$ ,  $\alpha_{gop} = 0.8$  are assumed. In all the test runs, the dielectric permittivity of liquid tin was set equal to

$$\varepsilon = \varepsilon_{Sn} = (n'_{Sn} + in''_{Sn})^2, \quad n'_{Sn} = 4.0, \quad n''_{Sn} = 8.4, \quad (5.172)$$

which corresponds to the laser wavelength  $\lambda = 1.064 \mu\text{m}$  of the YAG:Nd laser [28]. This choice was made keeping in mind application to irradiation of spherical droplets of tin with picosecond YAG-laser pulses, where adequate description of laser absorption at grazing incidence is a critical issue.

### 1. Test 1: reflection/absorption on a step profile

Reflection of a laser beam from a halfspace, filled with liquid tin with the dielectric permittivity (5.172), was simulated with the RALEF code on two types of numerical grid, namely, on the “square” and the “random” grids shown in Fig. 5.16. For simulations, a square domain of  $10 \mu\text{m} \times 10 \mu\text{m}$  was chosen, with one half of it occupied by the liquid tin. The numerical runs were performed with 1000 elementary rays in the incident laser beam.

The exact solution of this problem is a particular case of section 5.5 with  $n_0 = n_\infty = 1$ ,  $\varepsilon_0 = \varepsilon_1 = \varepsilon_4 = 1$ ,  $\varepsilon_3 = \varepsilon_{Sn}$ ,  $\bar{h}_2 = 0$ ,  $\bar{h}_1 = \bar{h}_3 = 10\pi$ . The transmission through layer  $\bar{h}_3$  is negligible because at normal incidence the electromagnetic fields decay by the factor  $e^{-\bar{h}_3 n''_{Sn}} = e^{-263.89}$ .

The results of a series of test runs for different values of the angle of incidence  $\theta_0 = 0, 10^\circ, \dots, 80^\circ$  are shown in Fig. 5.17. On the square mesh, the analytical values of the absorption coefficient  $f_a = 1 - f_{rfl}$ , given by the Fresnel formula (5.120), are reproduced *exactly* (for  $\theta_0 \leq 80^\circ$ ) — i.e. to at least 9 decimal digits — independent of the mesh size. In contrast, the random mesh exhibits finite deviations from the analytical solution, which become relatively

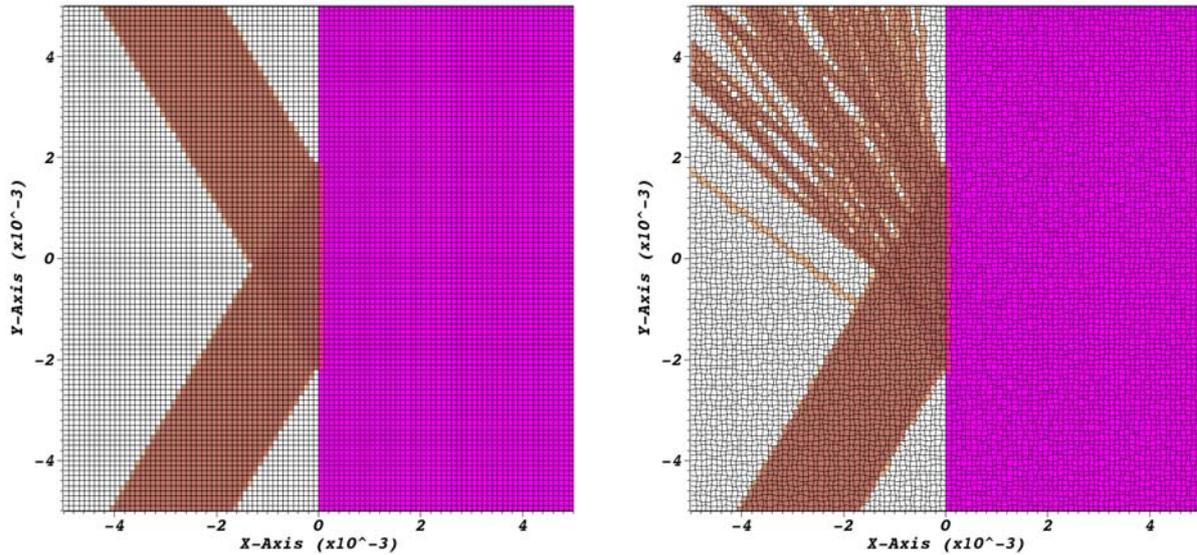


FIG. 5.16: Two mesh options used in test 1: the “random” grid on the right was obtained from the “square” grid on the left by imposing 20% random displacements to mesh nodes. The boundaries of the domain and the interface between the vacuum and liquid tin at  $x = 0$  have been preserved straight.

large (about a factor 1.5 at  $\theta_0 = 80^\circ$ ) at large incidence angles  $\theta > 60^\circ$  because of cell-to-cell fluctuations in the direction of the local normal to the critical surface near the surface of the Sn metal. No convergence to the exact solution is observed on the random mesh as the number of mesh cells is increased.

To understand the above described behavior, one has to recall that in our numerical algorithm (i) the jump in  $\varepsilon$  is smeared over two monolayers of cells along the vacuum-metal interface with large (in principle, unlimited) values of  $\nabla\varepsilon$  in the cell centers, and that (ii) Snell’s law is never applied at the cell interfaces inside the computational domain. Hence, a “perfect” (in this case rectangular) mesh, conforming to the material interface, reproduces the exact direction of the local normal to this interface, leading to a perfect specular reflection pattern and to the exact solution of the 1D Helmholtz equation on a step profile (which, in its turn, uses only the cell-centered values of  $\varepsilon$ ). The geometric optics with refraction does not corrupt the 1D wave solution (i.e. never applies within the boundary monolayers of cells) so long as the intermediate values of  $\varepsilon$  at the boundary nodes correspond to  $n_e/n_{e,cr} > \alpha_{gop}$  (or the mesh cell size is smaller than a certain fraction of the laser wavelength  $\lambda$ ).

On the random mesh, the numerically calculated local normal to the boundary fluctuates from cell to cell because in every cell it “feels” the positions of all its 4 nodes. This explains the fluctuation of the direction of reflection in the right-hand part of Fig. 5.16. Once the amplitude of mesh randomization is preserved, no convergence to the perfect specular reflection can be attained with mesh refinement.

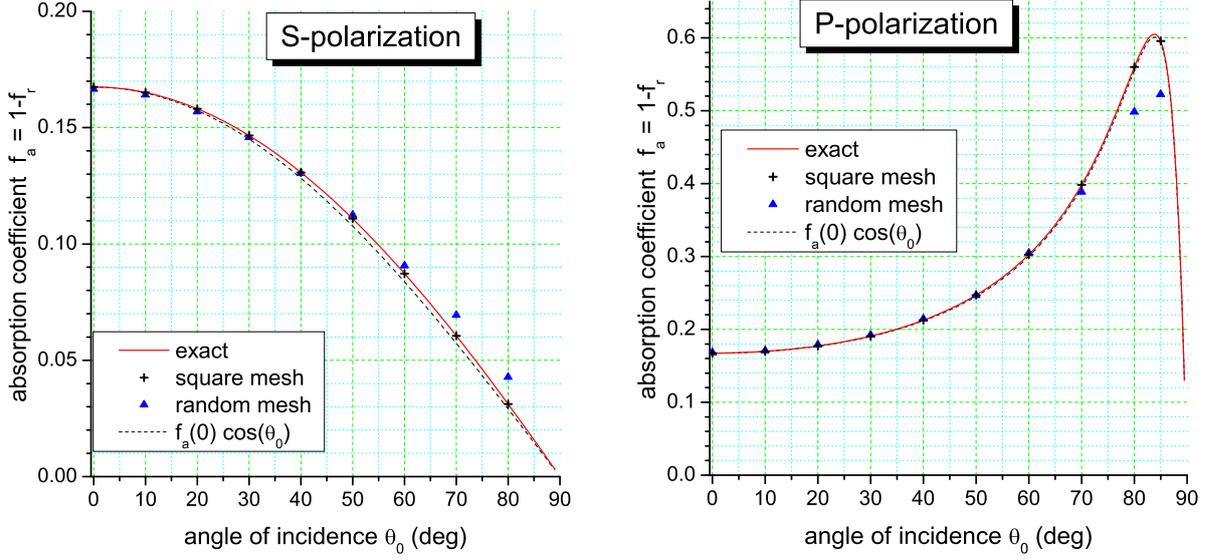


FIG. 5.17: Absorption fraction of a half-space of liquid tin with the dielectric permittivity (5.172) for a YAG laser pulse as a function of the incidence angle  $\theta_0$ . The exact solution is given by the Fresnel formula (5.120). The RALEF data were computed on a square (black plus signs) and a random (blue triangles) meshes with 10 cells per  $1 \mu\text{m}$ . The dashed curve is an approximation to the exact solution with the  $\cos \theta_0$  law.

## 2. Test 2: reflection/absorption on a linearly ramped profile

This test is conducted on a three-layer sandwiched structure, shown in Fig. 5.18, where the middle layer (layer 2) at  $0 < x < 2L$  has a linear profile of the dielectric permittivity

$$\varepsilon_2(x) = \varepsilon_1 + \frac{x}{2L}(\varepsilon_{21} - \varepsilon_1). \quad (5.173)$$

The left layer 1 at  $-1 \mu\text{m} < x < 0$  represents vacuum and has a constant permittivity of

$$\varepsilon_1 = (1 - 10^{-14} + 10^{-18}i)^2, \quad (5.174)$$

which deviates from the vacuum value of 1 by only a tiny amount to enable visualization of the incident and reflected laser beams.

The right-end value

$$\varepsilon_{21} = -1 + i\varepsilon_{21}'' \quad (5.175)$$

of the permittivity in the middle layer is chosen such that the critical surface, corresponding to  $\Re(\varepsilon_2) \equiv \varepsilon_2' = 0$ , lies exactly at the midpoint  $x = L$  of this layer. In other words, the length  $L$  represents the distance over which the free electron density  $n_e$  increases from  $n_e = 0$  to  $n_e = n_{e,cr}$  — which means that the left half of layer 2 is filled with the undercritical plasma, while its right half is supercritical. For different thicknesses  $L$ , different values of the imaginary component  $\varepsilon_{21}''$  are chosen so that the laser absorption fraction  $f_a$  does not deviate strongly from the middle value of  $f_a \approx 0.5$ .

The right layer 3 at  $2L < x < 2L + 1 \mu\text{m}$  has a constant dielectric permittivity of

$$\varepsilon_3 = (4.0 + 8.4i)^2 = -54.56 + 67.2i \quad (5.176)$$

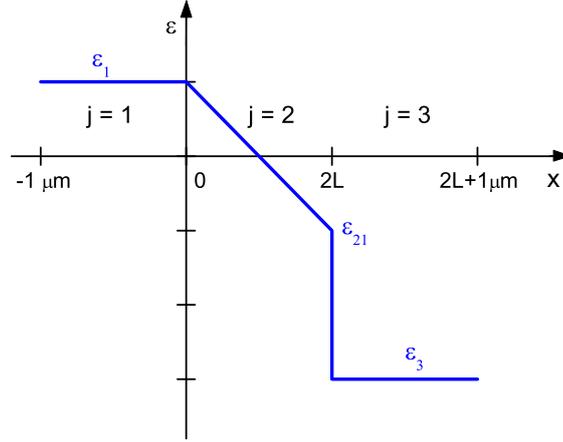


FIG. 5.18: The profile of the dielectric permittivity [shown is only the real part  $\varepsilon'(x)$ ] across a 3-layer slab used in test 2.

and represents liquid tin. Its thickness of  $1 \mu\text{m}$  is sufficient to ensure a negligible ( $f_{tr} < 10^{-20}$ ) overall laser transmission. Then the middle layer 2 may be thought of as representing the ablated tin plasma in the vicinity of the critical surface.

The test runs discussed below are split into two categories, namely, the steep-gradient cases with  $L \lesssim \lambda$  and the shallow-gradient cases with  $L \gg \lambda$ . Clearly, for *steep gradients* the adequate accuracy of our hybrid laser transport scheme must be ensured by its wave-optics part, for *shallow gradients* — by combination of both.

*a. Steep-gradient results.* On a steep profile with  $L \leq \lambda$  the condition (5.170) of transition to the 1D wave optics is fulfilled over the entire layer 2. Therefore, the numerical solution on a rectangular (!) grid must always converge to the exact one as the grid cell size

$$h = \frac{2L}{N_2} \quad (5.177)$$

goes to zero; here  $N_2$  is the number of cells along  $x$  in layer 2. The cell size along the  $y$  coordinate and in layers 1 and 3 does not affect the results — so long as the reflected beam comes out through layer 1 as is shown in Fig. 5.19.

The convergence to the exact solution is illustrated in Fig. 5.20 for two values ( $0^\circ$  and  $80^\circ$ ) of the incidence angle  $\theta_0$  on the profile with  $L = \lambda$ ,  $\varepsilon''_{21} = 0.1$ . Plotted is the relative error

$$\delta_a = \left| \frac{f_{a,h}}{f_{a,exact}} - 1 \right| \quad (5.178)$$

as a function of the grid size  $h$ ; here  $f_{a,h}$  is the absorption fraction calculated with RALEF for the cell size  $h$  in layer 2,  $f_{a,exact}$  is its exact value calculated as described in section 5.5.2.

In principle, the *finite-analytical* (as contrasted with the finite-difference) numerical scheme, implemented in RALEF for solution of the 1D Helmholtz equation, has the second convergence order, i.e. converges to the exact solution as  $\delta_a \propto h^2$ . This is confirmed by the two curves in Fig. 5.20, computed with  $\alpha_{gop} = 10^{-6}$  and marked with filled circles. From this plot one concludes also that a good accuracy of  $\delta_a \lesssim 10^{-2}$  can only be achieved when the numerical grid has at least 10 cells over the characteristic length scale  $L = |\nabla \bar{n}_e|^{-1}$ .

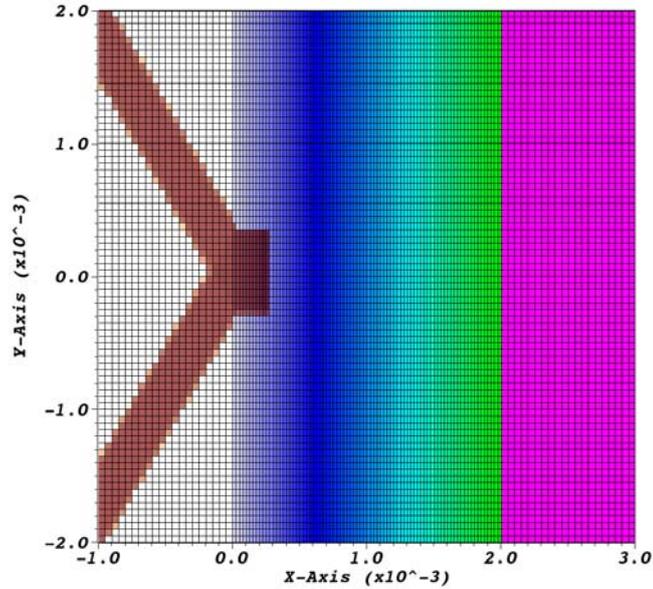


FIG. 5.19: The 2D domain in RALEF simulations of the laser reflection/absorption on the piecewise linear  $\epsilon$  profile of Fig. 5.18 with  $L = \lambda$  and  $\epsilon''_{21} = 0.1$ . The color palette indicates qualitatively the spatial variation of the dielectric permittivity  $\epsilon$ ; the laser beam is shown in brown.

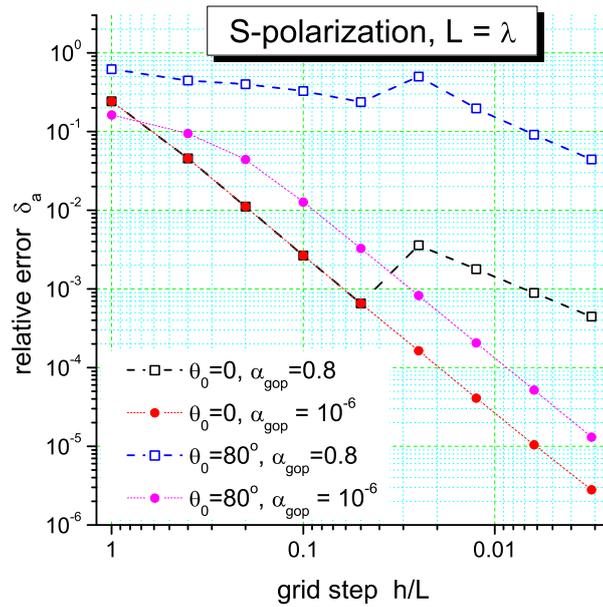


FIG. 5.20: Relative error  $\delta_a$  of the absorbed laser energy fraction  $f_a$  as a function of the grid cell size  $h$  along  $x$  in layer 2. When the geometric-optics part of the algorithm is fully blocked by setting  $\alpha_{gop} = 10^{-6}$ , the second-order convergence  $\delta_a \propto h^2$  to the exact solution is observed.

However, once the default value  $\alpha_{gop} = 0.8$  is used (the two curves in Fig. 5.20 marked with open squares), it leads to a degradation of the accuracy in two respects: (i) the error

$\delta_a$  becomes large at grazing incidence angles  $\theta_0 \gtrsim 80^\circ$ , and (ii) the rate of convergence to the exact solution is reduced to the 1st order. The main cause for both these effects is the fact that the jump of  $\nabla \bar{n}_e$  at  $x = 0$ , treated exactly in the exact analytical solution and in the finite-analytical solver of the 1D Helmholtz equation, is, within the general RALEF algorithm, smeared over the two monolayers of grid cells adjacent to the  $x = 0$  boundary. As a result, the condition (5.170) is not fulfilled in these two cell layers (in the  $x + 0$  layer — starting from  $h < 0.05$ ), and they are treated within the geometric optics approximation, which, due to ray refraction, introduces a noticeable additional error. Note that, due to this effect, a next smaller value of  $h$  may produce a less accurate result than the previous larger one.

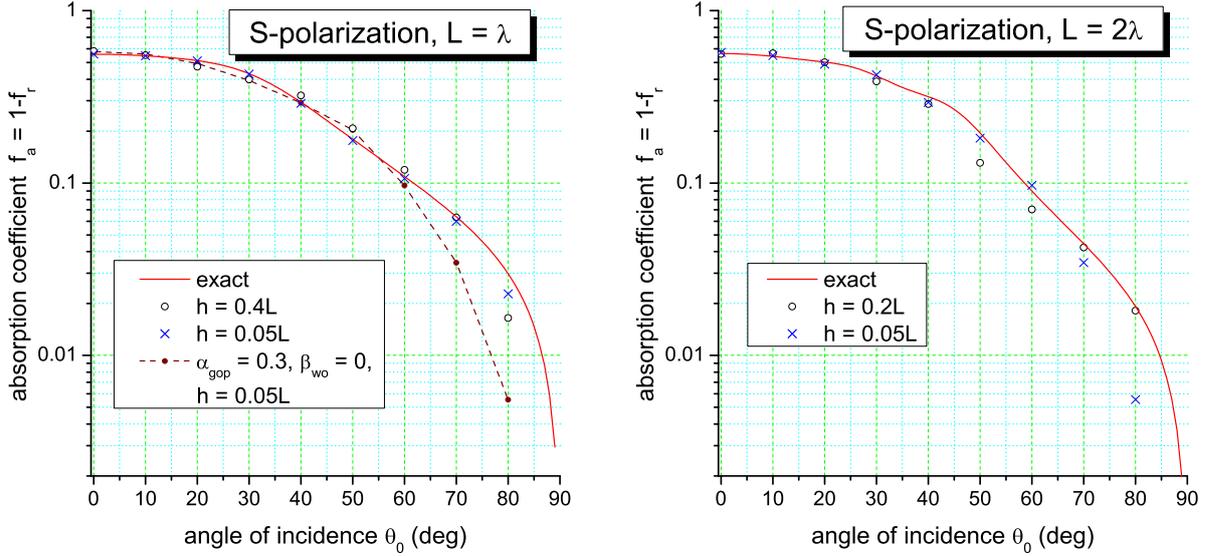


FIG. 5.21: Absorption fraction  $f_a$  as a function of the incidence angle  $\theta_0$  for two different slopes of  $\varepsilon_2(x)$ . Open circles: poor resolution of the gradient; crosses: moderate resolution of the gradient with 20 cells per  $L$ .

Figure 5.21 illustrates how the accuracy for the laser absorption depends on the angle of incidence  $\theta_0$  for two moderately steep gradients with  $L = \lambda$ ,  $\varepsilon_{21}'' = 0.1$  (left) and  $L = 2\lambda$ ,  $\varepsilon_{21}'' = 0.05$  (right). The main conclusion from the two parts of this figure is that, once the gradient of  $n_e$  is reasonably well resolved with 10-20 mesh intervals, our model typically demonstrates adequate accuracy ( $\delta_a \lesssim 10\%$ ) for all incidence angles  $\theta_0 \lesssim 60^\circ$ ; at the same time, a significant (sometimes by a factor 2-4 or bigger) degradation of accuracy may be observed at large incidence angles  $\theta_0 \gtrsim 80^\circ$ . Note that omission of the gradient from criterion (5.170) and using a smaller value of, say,  $\alpha_{gop} = 0.3$  would be a no better alternative in the general case — as is shown in the left part of Fig. 5.21.

The case of  $L = 2\lambda$  is of a particular interest because here the second term  $\beta_{wo}\lambda|\nabla \bar{n}_e|$  on the left-hand side of criterion (5.170) is equal to 0.5, and at least part of the laser beam trajectory in layer 2 is always treated within the geometric-optics approximation. As is shown in Fig. 5.22, for  $\theta_0 \geq 60^\circ$  already the whole beam trajectory falls into the domain of geometric-optics treatment. At the same time, with  $L = 2\lambda$  the gradient of  $\varepsilon_2(x)$  is still sufficiently steep to question the applicability of the geometric optics. The results in the right part of Figs. 5.21 and in Fig. 5.22 demonstrate that, for not too high values of

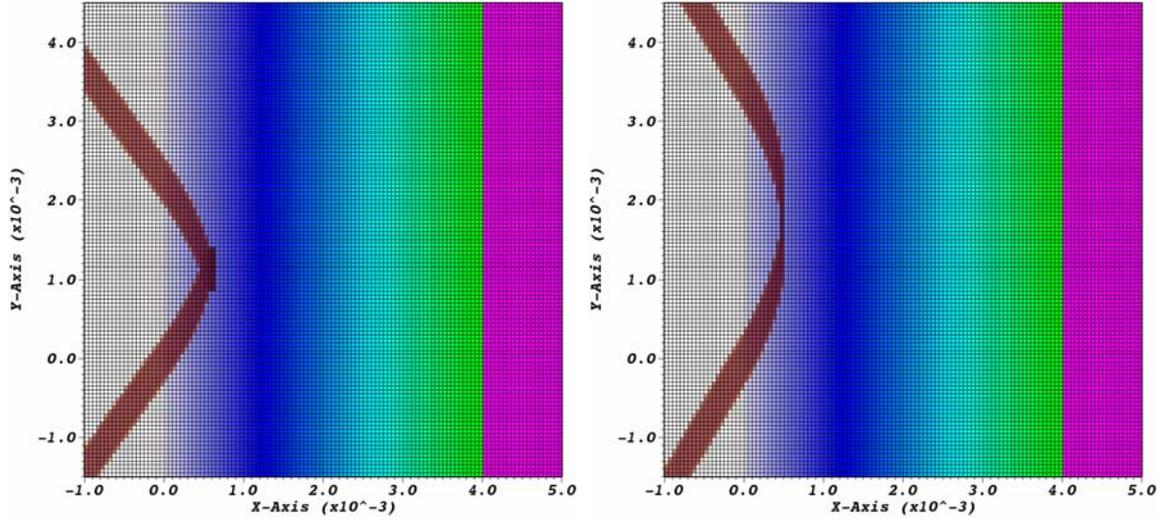


FIG. 5.22: Beam trajectories for  $\theta_0 = 55^\circ$  (on the left) and  $\theta_0 = 60^\circ$  (on the right) in the case of  $L = 2\lambda$ ,  $\varepsilon''_{21} = 0.05$ , calculated with the resolution of 40 cells per  $L$ .

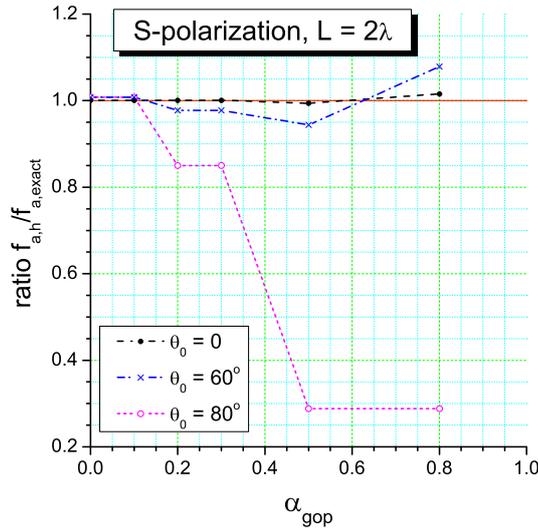


FIG. 5.23: Sensitivity to the transition parameter  $\alpha_{gop}$ : plotted is the ratio of the absorption coefficient  $f_{a,h}$ , calculated with  $N_2 = 40$ , to the exact value  $f_{a,exact}$  as a function of the transition parameter  $\alpha_{gop}$  for three different values of the incidence angle  $\theta_0 = 0, 60^\circ$  and  $80^\circ$  in the case of  $L = 2\lambda$ ,  $\varepsilon''_{21} = 0.05$ .

the incidence angle  $\theta_0 \lesssim 60^\circ-70^\circ$ , our hybrid model with the transition criterion (5.170) ensures adequate accuracy for  $f_a$ , and a relatively smooth transition between the geometric- and wave-optics approximations in what concerns the refractive displacement of the beam trajectory.

Figure 5.23 demonstrates explicitly the sensitivity of the absorbed energy fraction  $f_a$  to the chosen value of the transition parameter  $\alpha_{gop}$  in the criterion (5.170) for three different

incidence angles  $\theta_0$  in the case of  $L = 2\lambda$ . As already mentioned, a sufficiently low sensitivity is observed for only  $\theta_0 \lesssim 60^\circ$ . Ostensibly, the data for  $\theta_0 = 80^\circ$  suggest that one might significantly improve the accuracy at grazing incidence by choosing  $\alpha_{gop} = 0.2\text{--}0.3$  instead of  $\alpha_{gop} = 0.8\text{--}1.0$ . But then — as is demonstrated by the left part of Fig. 5.21 — the accuracy would still remain poor at steeper gradients with  $L \lesssim \lambda$ , and, more importantly, the refraction pattern in the case of shallow gradients (see below) would be too severely distorted.

*b. Shallow-gradient results.* Here we consider the case of a relatively shallow gradient of  $\varepsilon_2(x)$  with  $L = 10\lambda$  and  $\varepsilon''_{21} = 0.01$ . Similar to the case of steep gradients, Fig. 5.24 illustrates the convergence to the exact solution for the absorbed energy fraction  $f_a$  (left), and the overall dependence of the accuracy of numerically calculated  $f_a$  values on the incidence angle  $\theta_0$  (right); Fig. 5.25 shows how this accuracy depends on the choice of the transition parameter  $\alpha_{gop}$ . The conclusions that can be drawn from these figures are also similar to the case of steep gradients: adequate accuracy can be expected for not too large incidence angles  $\theta_0 \lesssim 70^\circ$  under the condition that the gradient of  $\varepsilon$  (or  $n_e$ ) is reasonably well resolved with  $\gtrsim 20$  mesh intervals over the characteristic scale  $L = |\nabla\varepsilon|^{-1} = |\nabla n_e|^{-1}$ . It may only be noted that the rate of convergence for grazing incidence ( $\theta_0 = 80^\circ$ ) is noticeably worse on a shallow gradient than on a steep one.

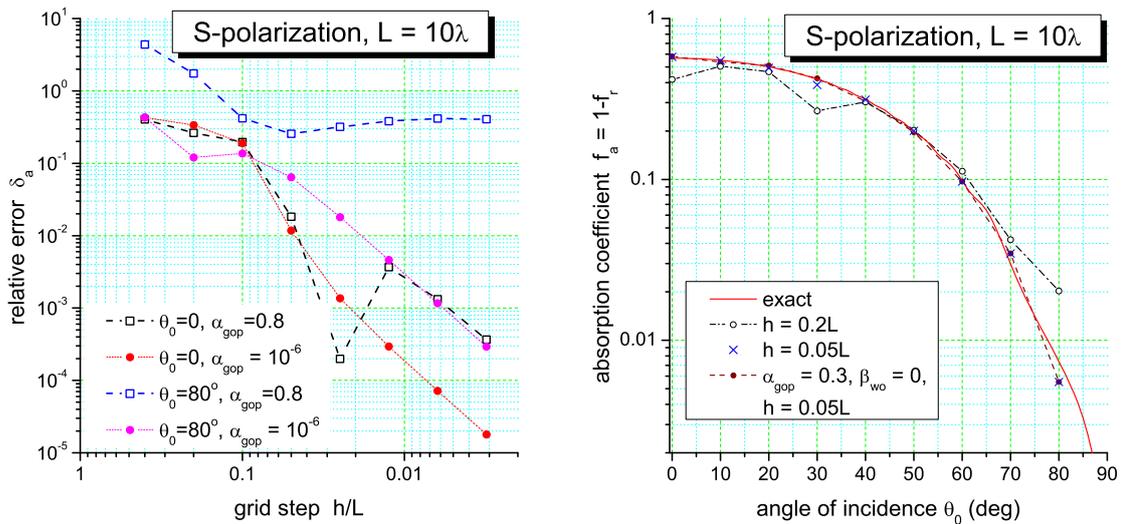


FIG. 5.24: Same as Fig. 5.20 (left) and Fig. 5.21 (right) but for the case of a shallow gradient with  $L = 10\lambda$ ,  $\varepsilon''_{21} = 0.01$ .

It is, however, not only the numerical error  $\delta_a$  in the overall absorption fraction  $f_a$  which matters. The plot in the right-hand part of Fig. 5.24 suggests that the version of the transition criterion (5.170) with  $\alpha_{gop} = 0.3, \beta_{wo} = 0$  is practically equivalent to that with  $\alpha_{gop} = 0.8, \beta_{wo} = 1$  (at least for weak gradients of  $\varepsilon$ ). But a very significant difference between these two options arises when one compares the spatial distribution of the laser energy deposition. Figure 5.26 compares two spatial trajectories of a laser beam, hitting the  $L = 10\lambda$  gradient at an angle  $\theta_0 = 40^\circ$ , computed with these two versions of the criterion (5.170). The two simulations produce close values of  $f_a = 0.314$  and  $f_a = 0.306$ . At the same time, one clearly sees that the reflected beam, computed with  $\alpha_{gop} = 0.8$ , comes out shifted

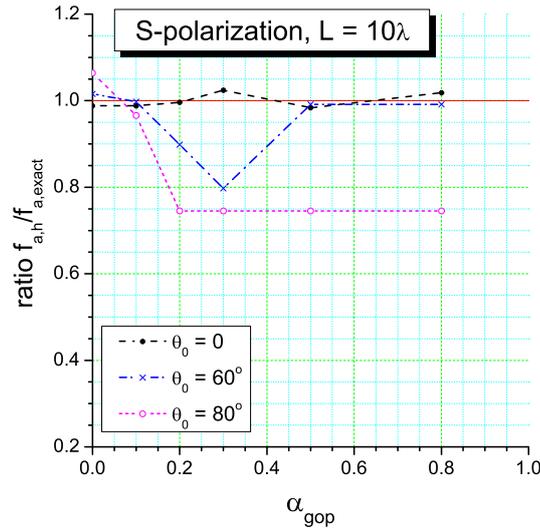


FIG. 5.25: Same as Fig. 5.23 but for the case of  $L = 10\lambda$ ,  $\varepsilon''_{21} = 0.01$ .

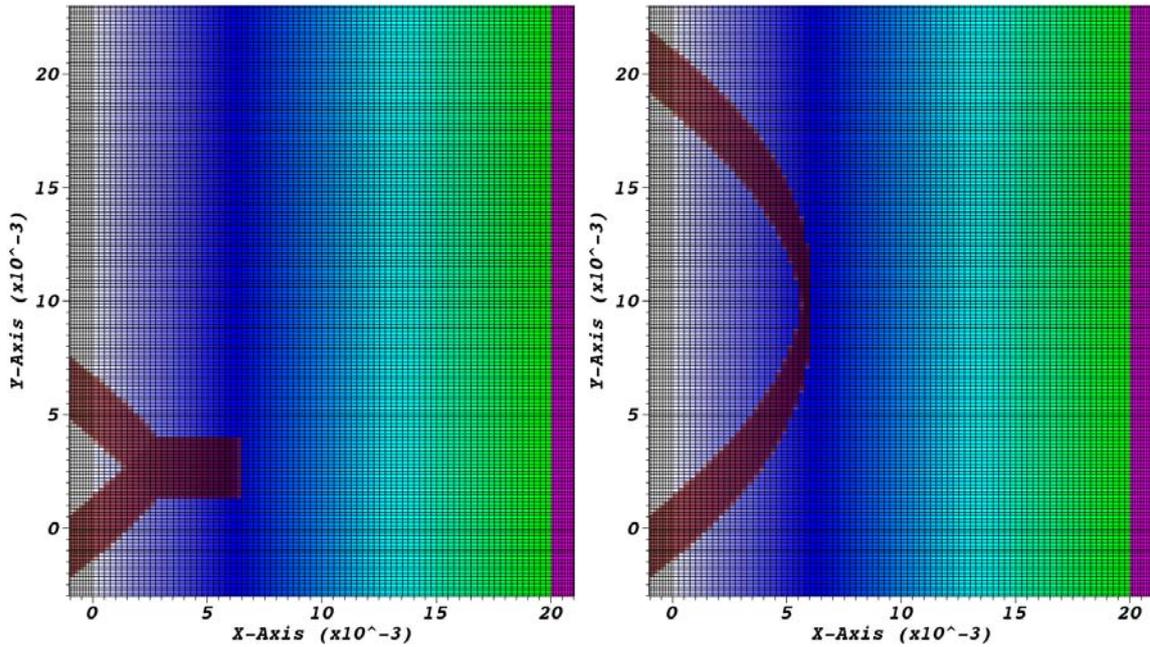


FIG. 5.26: Comparison of two beam trajectories for  $\theta_0 = 40^\circ$  in the case of a shallow gradient with  $L = 10\lambda$ ,  $\varepsilon''_{21} = 0.01$ , calculated with  $\alpha_{gop} = 0.3$ ,  $\beta_{wo} = 0$  (left) and with  $\alpha_{gop} = 0.8$ ,  $\beta_{wo} = 1$  (right); numerical resolution in both cases was 40 cells per  $L$ .

by about  $\Delta y \simeq 15 \mu\text{m} = 15\lambda$  with respect to the  $\alpha_{gop} = 0.3$  case. The latter implies that the spatial distributions of the laser energy deposition, computed with these two different options of criterion (5.170), may strongly differ from one another on a spatial scale of  $\simeq L$  — which in realistic cases may amount from several microns to several tens of microns (e.g. for a  $\text{CO}_2$  laser with  $\lambda = 10 \mu\text{m}$ ). In other words, the values of  $\alpha_{gop} \lesssim 0.5$  lead to a

severe underestimation of the effects of refraction on shallow gradients in the vicinity of the critical surface. This fact provides a decisive argument for choosing  $\alpha_{gop} \simeq 1$ . But then, to adequately describe the absorption fraction on steep gradients at oblique incidence, one needs the second  $\lambda|\nabla\bar{n}_e|$  term in the criterion (5.170).

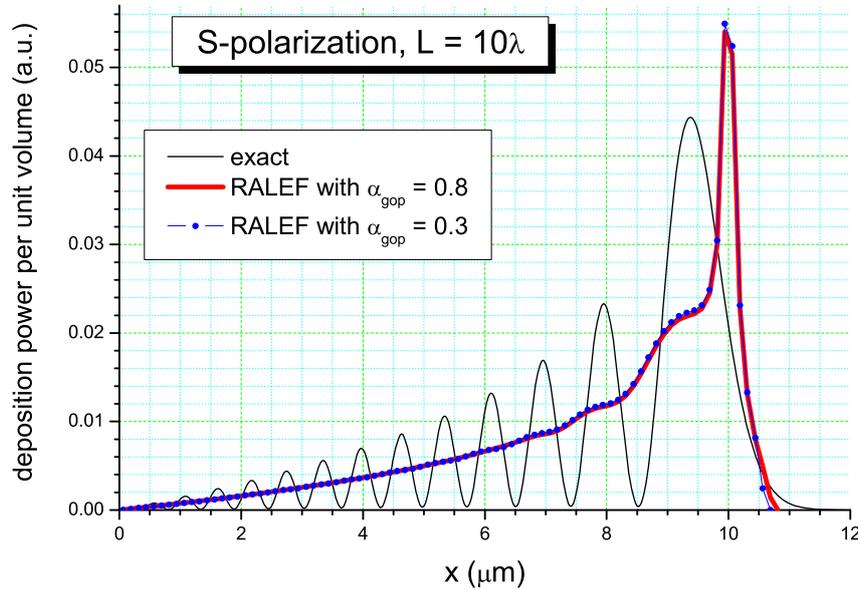


FIG. 5.27: Spatial profile of the volume-specific laser energy deposition for  $L = 10\lambda$  at normal incidence  $\theta_0 = 0$ . Thin black curve is the exact rigorous solution of the 1D Helmholtz equation. Thick red curve and blue dots represent the RALEF results (computed with  $N_2 = 80$  cells per  $L$ ) within the modified wave-optics approach.

Another important issue for the hybrid model, which deserves special attention, is the spatial distribution of the laser power deposition along the normal to the critical surface. The point of concern is that, due to the interference between the incident and the reflected waves, the exact solution to the 1D Helmholtz equation produces a standing wave, shown as a black strongly modulated curve in Fig. 5.27. In the geometric optics, on the contrary, the effect of wave interference is absent (averaged out in the approximation of  $\lambda \rightarrow 0$ ), and the deposition profiles exhibit no quasi-periodic modulation. Hence, a straightforward matching of the geometric-optics deposition profile to the wave-optics one produces, as a rule, an unphysical discontinuity at the matching point, whose position depends on the value of the transition parameter  $\alpha_{gop}$ ; as a result, the combined deposition profile becomes sensitive to the value of  $\alpha_{gop}$ . To avoid such unphysical and computationally troublesome behavior, in the RALEF code a modified wave-optics model is used, where (i) the reflection and transmission amplitudes are calculated by using the rigorous solution of the 1D Helmholtz equation, while (ii) the power deposition profile is calculated by simply omitting the oscillating interference term from the rigorous wave solution. And although such a modification appears as one more internal inconsistency of the model, it allows smooth matching of the geometric- and wave-optics deposition profiles at any value of  $\alpha_{gop} < 1$  — with the resulting combined profile being virtually independent of the particular choice of  $\alpha_{gop}$ , as is illustrated by Figs. 5.27 and 5.28 for different slopes of  $\varepsilon$ .

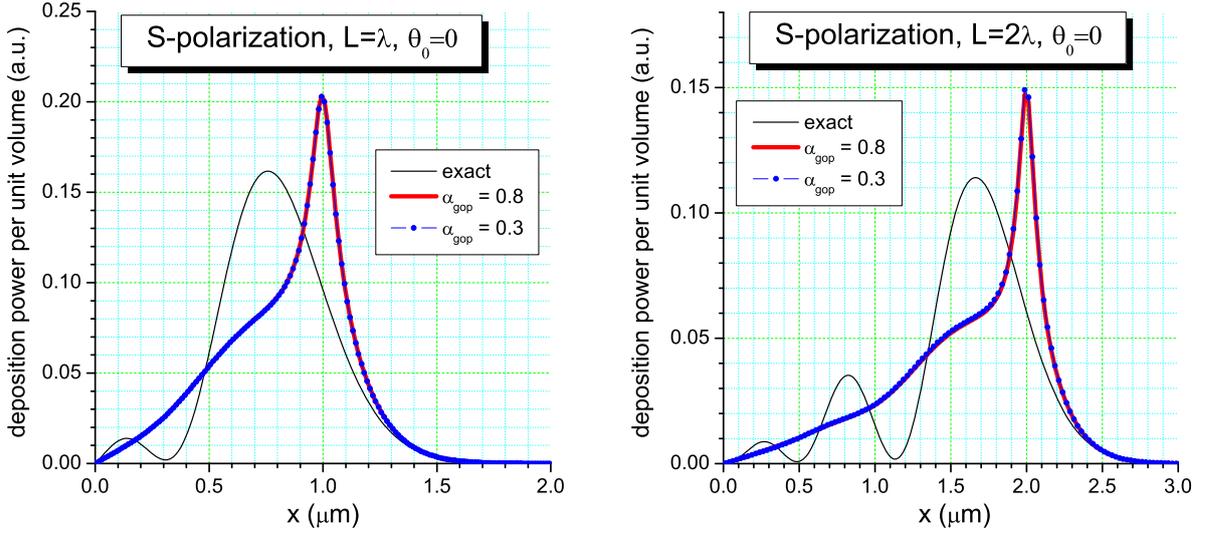


FIG. 5.28: Same as Fig. 5.27 but for moderately steep  $\varepsilon$  profiles of  $L = \lambda$  (left) and  $L = 2\lambda$  (right).

*c. Conclusions and recommendations.* The results of tests for the s-polarized laser light, presented above, may be summarized as follows:

- for not too large values of the angle of incidence  $\theta_0 \lesssim 60^\circ$ , the hybrid model demonstrates good accuracy for the laser absorption in the vicinity of the critical surface: typically, the relative error  $\delta_a$  for the absorbed power fraction  $f_a$  does not exceed 1–3% — provided that the gradient of  $n_e$  is reasonably well resolved with at least 10–20 mesh cells over the distance  $|\nabla \bar{n}_e|^{-1}$ ; the results are insensitive to the specific value of the transition parameter  $0 < \alpha_{gop} < 1$ ;
- at highly oblique incidence  $\theta_0 \gtrsim 70^\circ$ , the absorbed energy fraction  $f_a$  becomes sensitive to the details of numerics (to the quality of mesh, to the value of  $\alpha_{gop}$ , etc.) and may often be very inaccurate (in error by about a factor of 1.5–2 or larger);
- in contrast to the absorbed fraction  $f_a$ , the spatial distribution of the energy deposition by laser beams with narrow aperture (or high brightness variations across the focal spot) turns out to be rather sensitive to the  $\alpha_{gop}$  value even at moderate angles of incidence  $\theta_0 < 60^\circ$  because of refractive deflection of the incident and reflected beams near the critical surface — and especially so in the weak-gradient situations with  $\lambda |\nabla \bar{n}_e| \ll 1$ ; hence, an adequate account for the refraction effects requires  $\alpha_{gop} \simeq 0.7$ – $0.9$ .

As a corollary of the above conclusions, the following recommendation may be formulated for a particular case of very short (with duration of a few picoseconds or shorter) laser pulses:

- if a solid (or liquid) sample is irradiated by a very short laser pulse with a high contrast — so that the hydro expansion of the ablated plasma during the pulse remains of the order of the laser wavelength  $\lambda$  or smaller — the highest accuracy is expected to be achieved with some negligibly small value of  $\alpha_{gop} < 10^{-4}$ – $10^{-3}$ , guaranteeing that the whole layer of the ablated plasma is treated in the wave-optics approximation; in

particular, this choice should be most adequate for accurate description of the laser absorption at large angles of incidence  $\theta_0 > 60^\circ-70^\circ$ .

## 7. Laser absorption coefficient

### 1. Complex dielectric permittivity within the Drude model

a. *The Drude model.* Within the Drude model, the complex dielectric permittivity of a plasma

$$\varepsilon(\omega) = \varepsilon' + i\varepsilon'' \quad (5.179)$$

is expressed in terms of the effective collision frequency  $\nu_e$  of the free electrons as

$$\varepsilon' = 1 - \frac{\bar{\omega}_{pe}^2}{1 + \bar{\nu}_e^2}, \quad \varepsilon'' = \frac{\bar{\omega}_{pe}^2 \bar{\nu}_e}{1 + \bar{\nu}_e^2}, \quad (5.180)$$

where  $\varepsilon' > -\infty$  and  $\varepsilon'' > 0$  are real numbers,  $\omega$  is the angular frequency of the laser light,  $i = \sqrt{-1}$ ,

$$\bar{\nu}_e = \frac{\nu_e}{\omega}, \quad \bar{\omega}_{pe}^2 = \frac{\omega_{pe}^2}{\omega^2} = \frac{n_e}{n_{e,cr}}, \quad \omega_{pe}^2 = \frac{4\pi n_e e^2}{m_e}, \quad n_{e,cr} = \frac{m_e \omega^2}{4\pi e^2}, \quad (5.181)$$

$n_e$  is the number density of free electrons. Possible contribution of the bound electrons to the dielectric permittivity is neglected. Generally, collisions of free electrons with charged ions and neutral atoms are taken into account. Electron-electron collisions are ignored because they do not contribute to the dipole emission (absorption) of electromagnetic waves. In the RALEF package (the subroutines LASKNU and LASEPSN in file 'f06\_eos.f') the present model is implemented as model # 5.

The laser absorption coefficient  $k_{las}$  is expressed in terms of the imaginary part  $\kappa''$  of the complex refraction index  $\kappa = \sqrt{\varepsilon}$  as

$$k_{las} = 2 \frac{\omega}{c} \kappa'' \stackrel{R}{=} 6.3260586 \times 10^{16} [l][T] \nu_{las} \kappa'', \quad (5.182)$$

where  $\stackrel{R}{=}$  means that the RALEF-code units of measurement are used, and

$$\kappa' = \left( \frac{\sqrt{\varepsilon''^2 + \varepsilon'^2} + \varepsilon'}{2} \right)^{1/2} = \frac{\varepsilon''}{2\kappa''}, \quad (5.183)$$

$$\kappa'' = \left( \frac{\sqrt{\varepsilon''^2 + \varepsilon'^2} - \varepsilon'}{2} \right)^{1/2} = \frac{\varepsilon''}{2\kappa'}. \quad (5.184)$$

For numerical evaluation, the following implementation of the above formulae is used

Case  $\varepsilon' \geq 0$ :

$$\kappa' = \begin{cases} \sqrt{\varepsilon'}, & 10^4 \varepsilon'' \leq \varepsilon', \\ [0.5 (\sqrt{\varepsilon''^2 + \varepsilon'^2} + \varepsilon')]^{1/2}, & 0 \leq \varepsilon' < 10^4 \varepsilon'', \end{cases} \quad (5.185)$$

$$\kappa'' = 0.5 \varepsilon'' / \kappa';$$

Case  $\varepsilon' < 0$ :

$$\kappa'' = \begin{cases} \sqrt{-\varepsilon'}, & \varepsilon' \leq -10^4 \varepsilon'', \\ [0.5 (\sqrt{\varepsilon''^2 + \varepsilon'^2} - \varepsilon')]^{1/2}, & -10^4 \varepsilon'' < \varepsilon' < 0, \end{cases} \quad (5.186)$$

$$\kappa' = 0.5 \varepsilon'' / \kappa''.$$

*b. Maxwellian plasma of hydrogen-like ions.* First consider the simplest case of a plasma, composed of identical ions with a discrete charge  $z_i$  ( $1 \leq z_i \leq Z$ ) at a number density

$$n_i = n = \frac{\rho}{m_u A}, \quad (5.187)$$

and free electrons with a number density

$$n_e = n_i z_i = \frac{\rho z_i}{m_u A}. \quad (5.188)$$

Assuming ions to be infinitely heavy point-like charges, the electron collision frequency  $\nu_e$  is given by a well-known [25, §44, 48] expression

$$\nu_e \equiv \nu_{ei} = \frac{4\sqrt{2}\pi}{3} \frac{e^4 n_i z_i^2}{m_e^{1/2} T_e^{3/2}} L_{ei}, \quad (5.189)$$

where the Coulomb logarithm  $L_{ei}$  takes a different form in the low-frequency (an overdense plasma with  $\omega \ll \omega_{pe}$ ) [25, §44] and the high-frequency (an underdense plasma with  $\omega \gg \omega_{pe}$ ) [25, §48] limits: in particular, in the case of applicability of the quantum theory of Coulomb scattering

$$L_{ei} = \begin{cases} \ln \left( \frac{a T_e}{\hbar \omega_{pe}} \right), & \omega \ll \omega_{pe}, \\ \ln \left( \frac{4 T_e}{\gamma \hbar \omega} \right), & \omega \gg \omega_{pe}, \end{cases} \quad (5.190)$$

where  $\ln \gamma = 0.5772 \dots$  is Euler's constant, and  $a$  is an unknown numerical factor stemming from averaging the Landau collision integral. In the high-frequency limit the value of  $L_{ei}$  is obtained from the Gaunt factor in the first Born approximation.

To obtain a more general formula for the Coulomb logarithm  $L_{ei} = \ln \Lambda_{ei}$  of  $ei$  collisions, matching smoothly the classical and the quantum limits, we proceed as follows. We start with the formula

$$\Lambda_{ei} = \frac{v/\omega_{pe}}{[(z_i e^2/1.123 m_e v^2)^2 + (\hbar/2 m_e v)^2]^{1/2}}, \quad (5.191)$$

which combines the Bohr-Kramers classical formula with the Bethe-Lindhard-Larkin quantum formula for a fast ion  $+e z_i$  moving with velocity  $v$  past motionless plasma electrons. Next, we change to the ion rest frame and, following the original work of Spitzer [7], replace the adiabatic impact parameter  $v/\omega_{pe}$  in the numerator with the Debye length  $\lambda_D$ , given by

$$\lambda_D^{-2} = \frac{4\pi n_e e^2}{T_e} + \frac{4\pi n_i z_i^2 e^2}{T_i} = m_e \omega_{pe}^2 \left( \frac{1}{T_e} + \frac{z_i}{T_i} \right), \quad (5.192)$$

and replace  $m_e v^2$  in the denominator with  $3T_e$ . The resulting formula

$$\Lambda_{ei} = 2\sqrt{3} \frac{T_e}{\hbar} \left[ \left( \omega_{pe}^2 + \omega_{pe}^2 \frac{z_i T_e}{T_i} \right) \left( 1 + 1.0574 \frac{z_i^2 m_e e^4}{\hbar^2 T_e} \right) \right]^{-1/2}, \quad (5.193)$$

taken in the quantum limit of  $z_i e^2/\hbar v \ll 1$ , yields the low-frequency case  $\omega \ll \omega_{pe}$  in Eq. (5.190) with  $a = 2\sqrt{3}(1 + z_i)^{-1/2}$ .

To describe the high-frequency limit  $\omega \gg \omega_{pe}$ , we replace the first occurrence of  $\omega_{pe}^2$  in Eq. (5.193) with

$$\omega_{pe}^2 \rightarrow \omega_{pe}^2 + \beta_\omega^2 \omega^2, \quad \beta_\omega = \frac{\sqrt{3}}{2} \gamma \approx 1.542454. \quad (5.194)$$

In result, the final formula for  $\Lambda_{ei}$  takes the form

$$\Lambda_{ei} = 2\sqrt{3} \frac{T_e}{\hbar\omega} \left\{ \left[ \beta_\omega^2 + \bar{\omega}_{pe}^2 \left( 1 + \frac{z_i T_e}{T_i} \right) \right] \left( 1 + 1.0574 \frac{z_i^2 m_e e^4}{\hbar^2 T_e} \right) \right\}^{-1/2}. \quad (5.195)$$

*c. Monoatomic plasma of elements in the mean ion approximation.* Consider a plasma consisting of atoms (nuclei) of one sort with the atomic mass  $A$ , atomic number  $Z$ , and the total number density of nuclei (neutral atoms plus ions) given by

$$n = \frac{\rho}{m_u A}. \quad (5.196)$$

We assume that this plasma has a mean ionization degree  $0 \leq z_{ion} \leq Z$ , defined as

$$n_e = n z_{ion}, \quad z_{ion} = n^{-1} \sum_{j=1}^Z j n_{ij}, \quad (5.197)$$

where  $n_{ij}$  is the number density of ions with charge  $+ej$ . It is assumed that the values of  $z_{ion}(\rho, T)$  are provided by the EOS model.

Further on, within the approximation of mean ion we assume that for  $z_{ion} \geq 1$  all the plasma ions are identical point-like charges with a fractional charge  $+ez_{ion}$ , and the hydrogenic formulae from the previous subsection apply with  $z_i = z_{ion}$ . For  $z_{ion} < 1$  we assume the plasma to be a mixture of singly charged ions and neutral atoms with number densities

$$n_i = n_{i1} = n z_{ion} = n_e, \quad n_a = n_{i0} = n(1 - z_{ion}). \quad (5.198)$$

Accordingly, the electron collision frequency

$$\nu_e = \nu_{ei} + \nu_{ea} \quad (5.199)$$

becomes the sum of those with the singly charged ions, i.e.  $\nu_{ei}$  from Eq. (5.189) with  $n_i = n z_{ion}$ ,  $z_i = 1$ , and with the neutral atoms

$$\nu_{ea} = \sigma_{ea} n_a \left( \frac{T_e}{m_e} \right)^{1/2} = \sigma_{ea} n \left( \frac{T_e}{m_e} \right)^{1/2} \max(0, 1 - z_{ion}), \quad (5.200)$$

where for the cross-section of  $ea$  collisions we assume a fixed value of  $\sigma_{ea} = 10^{-15} \text{ cm}^{-2}$ . In the end, the two cases of  $z_{ion} \geq 1$  and  $z_{ion} < 1$  can be combined within a universal hydrogen-like formula for  $\nu_{ei}$  from the previous subsection by setting

$$n_e = n z_{ion}, \quad n_i z_i^2 = n_e \tilde{z}_i, \quad \tilde{z}_i = \max(1, z_{ion}), \quad (5.201)$$

and replacing all the alone-standing  $z_i$  under the Coulomb logarithm with  $\tilde{z}_i$ .

The effects of the electron degeneracy are accounted separately in the pre-logarithmic term in Eq. (5.189) and under the Coulomb logarithm. In the Coulomb logarithm  $\ln \Lambda_{ei}$ , it is sufficient to simply replace  $T_e$  in Eq. (5.195) with

$$T_F = \sqrt{T_e^2 + \left(\frac{2}{3}E_F\right)^2}, \quad (5.202)$$

where

$$E_F = \frac{\hbar^2}{2m_e} (3\pi^2 n_e)^{2/3} \stackrel{R}{=} 4.16628 \times 10^{-11} \frac{[m]^{2/3}}{[l]^2 [T]} \left(\frac{\rho z_{ion}}{A}\right)^{2/3} \quad (5.203)$$

is the Fermi energy, because in the degenerate case it is the electrons at the Fermi surface (in the narrow energy range  $\Delta(\frac{1}{2}m_e v_e^2) \simeq T_e \ll E_F$ ) which determine the plasma transport properties, i.e. the electrons with the mean energy equal to  $E_F$  instead of  $\frac{3}{2}T_e$  [12]. Also,  $T_e$  in Eq. (5.200) for the  $ea$  collision frequency should be replaced by  $T_F$ . At the same time, the term  $T_e^{3/2}$  in the denominator of Eq. (5.189) must be replaced by  $\tilde{T}_F^{3/2}$ , where

$$\tilde{T}_F = \sqrt{T_e^2 + (\beta_{\nu ei} E_F)^2}, \quad \beta_{\nu ei} = \left(\frac{4}{3\sqrt{\pi}}\right)^{2/3} \approx 0.82713. \quad (5.204)$$

The numerical coefficient  $\beta_{\nu ei}$  in Eq. (5.204) is adjusted such as to recover the correct value of  $\nu_{ei}$  in the degenerate limit of  $T_e \ll E_F$  for the Lorentzian plasma [12, 29]. Finally, we arrive at the following formulae

$$\bar{\omega}_{pe}^2 = \left(\frac{\rho z_{ion}}{A}\right) \left(\frac{\rho z_{ion}}{A}\right)_{cr}^{-1}, \quad \left(\frac{\rho z_{ion}}{A}\right)_{cr} \stackrel{R}{=} 4.691522 \times 10^{20} \frac{[T]^2 [l]^3}{[m]} \nu_{las}^2, \quad (5.205)$$

$$\bar{\nu}_e \stackrel{R}{=} 3.743144 \times 10^{-27} \frac{[m]}{[l]^3 [T]^{5/2}} \frac{\rho}{A \tilde{T}_F^{3/2} \nu_{las}} \left[ z_{ion} \tilde{z}_i L_{ei} + K_{ea} \tilde{T}_F^{3/2} T_F^{1/2} \max(0, 1 - z_{ion}) \right], \quad (5.206)$$

$$K_{ea} \stackrel{R}{=} 5.621421 \times 10^{21} [T]^2, \quad (5.207)$$

$$\Lambda_{ei} \stackrel{R}{=} 2\sqrt{3} \left(\frac{T_F}{\nu_{las}}\right) \left\{ \left[ \beta_\omega^2 + \bar{\omega}_{pe}^2 \left(1 + \frac{\tilde{z}_i T_F}{T_i}\right) \right] \left(1 + \frac{4.610 \times 10^{-11} \tilde{z}_i^2}{[T] T_F}\right) \right\}^{-1/2}. \quad (5.208)$$

For the Coulomb logarithm  $\ln \Lambda_{ei}$  of  $ei$  collisions we use an interpolation formula

$$L_{ei} = \ln \left[ 1 + \frac{\Lambda_{ei}(g_\epsilon + \Lambda_{ei})}{1 + \Lambda_{ei} + (6.5g_\epsilon \Lambda_{ei})^{-1}} \right], \quad (5.209)$$

which has one free parameter  $g_\epsilon$  for the limit of  $\Lambda_{ei} \ll 1$  [13], and approaches  $\ln \Lambda_{ei}$  in the limit of  $\Lambda_{ei} \gg 1$  independently of the  $g_\epsilon$  value. Formula (5.209) has a physically reasonable behavior  $L_{ei} \propto \Lambda_{ei}^2$  in the limit of  $\Lambda_{ei} \ll 1$ , which corresponds to the scattering of free electrons on thermal ion oscillations in a crystal lattice of a metal [13]. The fit parameter  $g_\epsilon$  is used to achieve agreement with the experimental reflection coefficient of a given metal near normal conditions for a given laser frequency  $\omega$ ; for Sn, a practically perfect agreement with the data from Ref. [28] at  $T = 335$  °C,  $\hbar\omega = 0.1$ -1 eV is achieved with  $g_\epsilon = 8.3$  ( $f_a = 7.00\%$  versus 6.95% in [28] for  $\hbar\omega = 0.1$  eV, and  $f_a = 15.17\%$  versus 15.54% in [28] for  $\hbar\omega = 1$  eV).

2. *The inverse-bremsstrahlung collisional absorption*

As a rule, the absorption of laser light with  $\hbar\omega \equiv h\nu_{las} \lesssim 3\text{--}10$  eV in hot undercritical ( $\omega \gg \omega_{pe}$ ) plasmas can be described by the monochromatic Kramers formula (2.38). In the subroutines `LASKNU` and `LASEPSN` (file ‘`f06_eos.f`’) of the RALEF package this model is implemented as model # 3. In the context of the mean ion approximation to the monoatomic plasma (see the previous subsection), the formula (2.38) can be rewritten as

$$k_{las,ff} \stackrel{R}{=} K_{ff} \left( \frac{\rho^2 z_{ion}}{A^2} \right) \frac{1 - e^{-\nu_{las}/T_e}}{T_e^{1/2} \nu_{las}^3} \left[ z_{ion} \tilde{z}_i \bar{g} + \frac{\sqrt{3}}{\pi} K_{ea} T_e^2 \max(0, 1 - z_{ion}) \right] \Gamma_{las}, \quad (5.210)$$

where

$$K_{ff} \stackrel{R}{=} 4.577350 \times 10^{-31} \frac{[m]^2}{[l]^5 [T]^{7/2}}, \quad (5.211)$$

and  $z_{ion}$ ,  $\tilde{z}_i$ , and  $K_{ea}$  are defined in the previous subsection. Equation (5.210) accounts for the collisions of free electrons with neutral atoms in the same simplified way as Eq. (5.206).

The ad hoc numerical factor  $\Gamma_{las}$  is introduced to imitate the enhanced laser absorption in the vicinity of the critical surface, where the free electron density  $n_e \simeq n_{e,cr}$ . In practice we use the approximation [30], [26, §5.1]

$$\Gamma_{las}(x) = \begin{cases} (1 - \bar{\omega}_{pe}^2)^{-1/2}, & \bar{\omega}_{pe}^2 < 1 - \Gamma_0^{-2}, \\ \Gamma_0, & \bar{\omega}_{pe}^2 \geq 1 - \Gamma_0^{-2}, \end{cases} \quad (5.212)$$

where  $\Gamma_0 \simeq 10\text{--}100$  is some reasonably large number. The value of  $\Gamma_0$  is loaded as the input variable `ovcrkblas(iblas)` in subroutine `LASINPT`, file ‘`f10_taskinpt.f`’.

The Maxwellian-mean Gaunt factor  $\bar{g}$  is most readily calculated in the Born approximation

$$\bar{g} = \frac{\sqrt{3}}{\pi} \exp\left(\frac{\hbar\omega}{2T_e}\right) K_0\left(\frac{\hbar\omega}{2T_e}\right), \quad (5.213)$$

where  $K_0(z)$  is the Macdonald function of order zero. In the high-temperature limit  $\hbar\omega \ll T$  one can use the corresponding asymptotic expansion of  $K_0(z)$ , which leads us to a simple approximation

$$\bar{g} \stackrel{R}{=} \max\left\{1; \frac{\sqrt{3}}{\pi} \ln\left(2.24584 \frac{T_e}{\nu_{las}}\right)\right\}. \quad (5.214)$$

## 6. EOS AND OPACITIES

### 1. “F17” file format for data exchange

Beginning from the year 2017, the EOS and opacity tables in the RALEF code are generated in the “f17” format, described below. This format has been introduced with the primary goal of providing a possibility to read and extract various numerical data from a file with a minimum amount of advance knowledge about the format, used for writing these data, combined with a possibility to add at any time any amount of new data of any type, supported by the fortran-90 language.

1. Any file `sample.f17` in the “f17” format is supposed to be of the *unformatted* form in all the I/O commands, i.e. written by the statements of the form

```
write(unit) a,b,...
```

2. Any file in the “f17” format is structured as a sequence of an arbitrary number of **items**; every individual **item** consists of two (and only two) elements, namely, of its **tag** and **entity**:

```
item # 1 ::= tag #1
           entity # 1
item # 2 ::= tag # 2
           entity # 2
...
```

3. The **entity** of an item is either a fortran *array*, or a fortran *scalar variable* of stored data, written into the `sample.f17` file as a single fortran *record*, i.e. by a single use of the `write` statement.
4. The **tag** of an item contains all the information about the fortran *type* and *shape* of the **entity**, associated with this item. Every **tag** is written into the `sample.f17` file as a single fortran *record*, i.e. by a single use of the `write` command. Every **tag** has the same fixed structure, namely, it is composed of two `character(32)` variables, followed by eight `integer(4)` numbers.

Example of code to write a single tag:

```
character(32) :: ent_name,ent_type
integer(4)   :: ent_shape(8)
integer     :: luw

luw=21
open(luw,file='sample.f17',form='unformatted')
ent_name='header'
ent_type='character(128)'
ent_shape=(/1,10,0,0,0,0,0,0/)
write(luw) ent_name,ent_type,ent_shape
```

Character variable `ent_name` is equal to the *name* of the **entity**, represented by the given **tag** (of no more than 32 characters). Character variable `ent_type` is equal to the fortran *type* of the **entity** (like `integer(4)`, `real(8)`, `complex(16)`, etc.; 32 characters are sufficient to describe any intrinsic fortran-90 type).

Integer `ent_shape(1)` is equal to the *rank* of the **entity**, represented by the given **tag**. Possible values are `ent_shape(1) = 0, 1, ..., 7`. The value `ent_shape(1) = 0` means that the **entity** is a *scalar variable*; the value `ent_shape(1) = 1` means that the **entity** is a *rank-1 array* with one dimension; `ent_shape(1) = 2` means that the **entity** is a *rank-2 array* with two dimensions; etc.

Integer `ent_shape(2)` is the *extent* of the **entity** array along dimension 1; integer `ent_shape(3)` is the *extent* of the **entity** array along dimension 2; etc.

In the above example, the **tag** represents an **entity** named **header**, which is a one-dimensional *array* of type `character(128)`, containing 10 *elements*.

5. *Example* of code for writing two items, namely, an *integer* array `ia(1:100,1:200)` and a *real(4)* array `pr(1:500,1:400,1:10)`:

```

character(32) :: ent_nt(2)
integer(4)   :: ent_sh(8)

integer :: luv,ia(100,200)
real(4)  :: pr(500,400,10)

luv=21
open(luv,file='sample.f17',form='unformatted')

ia= ...
ent_nt(1)='ia'
ent_nt(2)='integer'
ent_sh=(/2,100,200,0,0,0,0,0/)
write(luv) ent_nt,ent_sh
write(luv) ia

pr= ...
ent_nt(1)='pressure'
ent_nt(2)='real(4)'
ent_sh=(/3,500,400,10,0,0,0,0/)
write(luv) ent_nt,ent_sh
write(luv) pr

```

The only *a priori* information that a customer has to know in order to be able to use a given “f17” file, is the format of the **tag records**. Once the latter is known, one easily generates a list of all items in the given “f17” file; in the RALEF code this can be done by calling a service routine

```
call FINDALL_F17(lur,luv,ierr) ,
```

where `lur` is the *unit* number of the opened “f17” file to be explored, `luv` is the *unit* number of the opened output file (for the standard output `luv=6`), `ierr` is the `integer(4)` error flag. Having obtained the full list of **items**, one readily extracts the **entity** of a needed **item** by making use of the `FIND_F17(ent_name,lur,luv,ierr)` service routine as, for example, could be done for the previously written `sample.f17` file:

```

integer :: lur
real(4)  :: pr(500,400,10)

lur=21
open(luv,file='sample.f17',form='unformatted')

call FIND_F17('pressure',lur,6,ierr)
read(lur) pr

```

It is recommended that any “f17” file includes a `character(128)`-type **item** (a rank-1 array) with a standard name `header`, where all the explanations are given concerning the physical meaning, units of measurement, etc. for all the **items** stored in a given “f17” file. If `header` is the first **item** in a given “f17” file, the format of its **tags** can be explained in the first line of the `header`, and a customer could read it out as

```
integer :: lur
character(128) :: char

lur=21
open(luw,file='sample.f17',form='unformatted')
read(lur) char
```

(i.e. a customer has only to know that the first item is a `character(128)` variable or array), and then, having learned the format of **tags** from the contents of `char`, to read out the full list of items in the explored file.

To minimize the probability of errors by reading out **entities** from the “f17” files, one can add a special **control character** (or a **control word**) to the end of each **entity record**, like, for example,

```
character(32) :: ent_nt(2)
integer(4) :: ent_sh(8)

integer :: luw
real(4) :: pr(500,400,10)

character, parameter :: ccr='|'
luw=21
open(luw,file='sample.f17',form='unformatted')

pr= ...
ent_nt(1)='pressure'
ent_nt(2)='real(4)'
ent_sh=(/3,500,400,10,0,0,0,0/)
write(luw) ent_nt,ent_sh
write(luw) pr,ccr
```

This allows to check directly whether the corresponding `read` statement has attempted to read out less data (by attempting to read more data, an error message will be generated) than was originally written into the corresponding **entity**. The actual value of the **control character** (**control word**) is also recommended to be included into the first line of the `header`.

## 2. Tabular GLT equation of state

Here a new version of the tabular GLT-EOS (version GLT-EOS17), implemented in the year 2017, is described. All the data, required by the RALEF code to make use of the GLT-EOS in a given job, is written into a single file in the “f17” format with a fixed name `'input/tab-glteos17.f17'`. This file may contain the EOS data for an arbitrary number of materials, numbered sequentially by index

$$m = 1, 2, \dots, nmatgtes. \quad (6.1)$$

The value of `nmatgtes` is written into the `'tab-glteos17.f17'` file as a separate **item**.

The GLT-EOS data file `'tab-glteos17.f17'` is generated by running the **program** `GTESFILL17` (file `'tabeos17_GLT.f'`), belonging to a separate GLT-EOS code package, which is fully independent of the RALEF package. Unless the test **subroutine** `GTESTST17` (called from **program** `GTESFILL17`) is executed, the only communication between the GLT-EOS and the RALEF packages is by means of the **module** `COMGLT17`, which contains all the necessary common variables and arrays, and which is formally included into the file `'f00_comod.f'` from the RALEF package.

Once the GLT-EOS data file `'tab-glteos17.f17'` is available, the EOS itself is calculated by calling the **subroutine** `GLTEOS17`, which belongs to the RALEF code package (file `'f07_lnktbls.f'`). Note that this same subroutine is also called when running the test routine **subroutine** `GTESTST17`.

### 1. Units of measurement

All the physical quantities, written into file `'tab-glteos17.f17'`, are given in a separate system of user-defined units (the GLT-EOS units), which generally do not coincide with the RALEF units of measurements, chosen for a given particular job. This provides a possibility to freely change the 4 base RALEF units

$$\begin{aligned} \text{unilngth} &\equiv [l]_{RAL}, \\ \text{unitime} &\equiv [t]_{RAL}, \\ \text{unimass} &\equiv [m]_{RAL}, \\ \text{unitemp} &\equiv [T]_{RAL}, \end{aligned} \tag{6.2}$$

without the necessity to recalculate the GLT-EOS table.

The GLT-EOS units are set by assigning in **program** `GTESFILL17` (file `'tabeos17.f'`) the values of the four variables `ulengtes`, `utimgtes`, `umasgtes`, `utmpgtes`, which represent the chosen GLT-EOS units of length, time, mass, and temperature, expressed in terms of the CGS units (i.e. in [cm], [g], [sec]). The units for all other physical quantities are derived from these 4 base units. By default, the values

$$\begin{aligned} \text{ulengtes} &\equiv [l]_{GLT} = 1, \\ \text{utimgtes} &\equiv [t]_{GLT} = 10^{-6}, \\ \text{umasgtes} &\equiv [m]_{GLT} = 1, \\ \text{utmpgtes} &\equiv [T]_{GLT} = 1.60217733 \times 10^{-9} \end{aligned} \tag{6.3}$$

are used, i.e. 1 cm for length,  $10^{-6}$  sec for time, 1 gram for mass, and 1 keV for temperature; then, for example, the density  $\rho$  is measured in [g/cc], the pressure  $p$  in [Mbar], the mass-specific internal energy  $e$  in [100 kJ/g], etc. The values of `ulengtes`, `utimgtes`, `umasgtes`, `utmpgtes` are written into the file `'tab-glteos17.f17'` as a separate **item** under the name `units_gtes`.

Having read out the 4 base GLT-EOS units `ulengtes`, `utimgtes`, `umasgtes`, `utmpgtes`

from file 'tab-glteos17.f17', one easily calculates the three basic conversion factors

$$\begin{aligned} c_{f\rho} &= \frac{[m]_{GLT}[l]_{RAL}^3}{[m]_{RAL}[l]_{GLT}^3}, \\ c_{fT} &= \frac{[T]_{GLT}}{[T]_{RAL}}, \\ c_{fe} &= \frac{[l]_{GLT}^2[t]_{RAL}^2}{[l]_{RAL}^2[t]_{GLT}^2}, \end{aligned} \quad (6.4)$$

used to relate various thermodynamic quantities in the RALEF and GLT units, and defined by the equalities

$$\begin{aligned} \rho_{RAL} &= c_{f\rho} \cdot \rho_{GLT}, \\ T_{RAL} &= c_{fT} \cdot T_{GLT}, \\ e_{RAL} &= c_{fe} \cdot e_{GLT}; \end{aligned} \quad (6.5)$$

here  $f_{RAL}$  is the physical quantity  $f$  in RALEF units, and  $f_{GLT}$  is the same quantity in the GLT-EOS units. For the pressure  $p$ , sound speed squared  $c_s^2$ , and mass-specific heat capacity  $c_v = (\partial e / \partial T)_v$ , the corresponding conversion factors are given by

$$\begin{aligned} p_{RAL} &= c_{f\rho} c_{fe} \cdot p_{GLT}, \\ c_{s,RAL}^2 &= c_{fe} \cdot c_{s,GLT}^2, \\ c_{v,RAL} &= \frac{c_{fe}}{c_{fT}} \cdot c_{v,GLT}. \end{aligned} \quad (6.6)$$

If an EOS table was generated for an element  $(Z, A_1)$ , and one has to run a simulation for a different isotope  $(Z, A_2)$  of the same element, there is no need to generate a new table: one can simply apply the following *isotopic rescaling* to the existing table for  $(Z, A_1)$ . The isotopic rescaling is based on the observation that, once one has two thermodynamic states  $(\rho_1, e_1)$  and  $(\rho_2, e_2)$  which satisfy

$$\frac{\rho_1}{A_1} = \frac{\rho_2}{A_2}, \quad e_1 A_1 = e_2 A_2, \quad (6.7)$$

these two states will both have the same volume per atom, and the same energy per atom; hence, they should have the same pressure  $p_1 = p_2$  and the same temperature  $T_1 = T_2$ . As a result, a proper account of the isotopic rescaling is achieved after one adopts the following expressions for the three basic conversion factors

$$\begin{aligned} c_{f\rho} &= \frac{[m]_{GLT}[l]_{RAL}^3}{[m]_{RAL}[l]_{GLT}^3} \frac{A_{RAL}}{A_{GLT}}, \\ c_{fT} &= \frac{[T]_{GLT}}{[T]_{RAL}}, \\ c_{fe} &= \frac{[l]_{GLT}^2[t]_{RAL}^2}{[l]_{RAL}^2[t]_{GLT}^2} \frac{A_{GLT}}{A_{RAL}}; \end{aligned} \quad (6.8)$$

here  $A_{GLT} = A_1$  is the atomic mass, for which the EOS table was generated, and  $A_{RAL} = A_2$  is the actual atomic mass used in the RALEF simulation.

2. Principal arrays with the EOS data

The file 'tab-glteos17.f17' contains two main 1D arrays with tabulated thermodynamic EOS data, namely,

- the “warm” table `petab17(:)`, where all the needed thermodynamic quantities for all the `nmatgtes` materials are listed as functions of pairs  $(\rho_i, T_j)$  of the density and temperature on a 2D grid

$$\{\rho_i\} \otimes \{T_j\}, \quad i = 1, 2, \dots, N_r + 1; j = 0, 1, 2, \dots, N_T + 1,$$

and

- the “cold-curve” table `pexstab17(:)`, where the thermodynamic quantities for all materials are tabulated for a single value of the “cold” temperature  $T_x$  [which must lie below all the “warm” temperatures  $T_j$  with  $j = 1, 2, \dots, N_T + 1$  in the table `petab17(:)`] on an expanded density grid  $\{\rho_{ix}\}$ ,  $ix = 1, 2, \dots, N_{rx} + 1$ ,  $N_{rx} \geq N_r$ ; the “cold” temperature  $T_x$  is also added to the “warm” table `petab17(:)` as an extra value  $T_j$  with  $j = 0$ .

3. Density and temperature grids

The density and temperature grids for tables `petab17(:)` and `pexstab17(:)` are essentially arbitrary. The only constraint is that the grid nodes  $\rho_i, \rho_{ix}, T_j$  must be monotonically increasing with the corresponding indices. In all the interpolation formulae, it is the logarithms  $\ln \rho_i, \ln \rho_{ix}$ , and  $\ln T_j$  that are actually used as independent variables.

The density interval  $[\rho_{0x}, \rho_{N_{rx}+1}]$  for the cold curve is assumed to be generally much wider than the density interval  $[\rho_0, \rho_{N_r+1}]$  for the “warm” table, so that

$$[\rho_0, \rho_{N_r+1}] \in [\rho_{0x}, \rho_{N_{rx}+1}], \quad \rho_{0x} \ll \rho_0, \quad \rho_{N_{rx}+1} \gg \rho_{N_r+1}. \quad (6.9)$$

Within the “warm” density interval  $[\rho_0, \rho_{N_r+1}]$ , the “warm” grid nodes  $\{\rho_i\}$  must coincide with the corresponding “cold” grid nodes  $\{\rho_{ix}\}$ .

The “warm” density grids  $\{\ln \rho_i\}$  for all `nmatgtes` materials are packed into a single 1D array `rlngrigtes(:)`. Different materials  $m$  are distinguished by the values of  $i$ -index offsets

$$\begin{aligned} \text{morlngrigtes}(1) &= 0, \\ \text{morlngrigtes}(m+1) &= \sum_{k=1}^m [N_r(k) + 1], \quad m = 1, 2, \dots, \text{nmatgtes}, \end{aligned} \quad (6.10)$$

so that the  $\{\ln \rho_i\}$  grid for material  $m$  is given by the array section `rlngrigtes(morlngrigtes(m)+1:morlngrigtes(m+1))`. Accordingly, the shape of the array `rlngrigtes(:)` is `rlngrigtes(1:morlngrigtes(nmatgtes+1))`. Note that the values of  $N_r = N_r(m)$  in Eq. (6.10) [stored in the array `nrgtes(1:nmatgtes)`] are generally different for different materials  $m$ . The two arrays `nrgtes(1:nmatgtes)` and `rlngrigtes(1:morlngrigtes(nmatgtes+1))` are written into the file 'tab-glteos17.f17'.

Similarly, the “cold” density grids  $\{\ln \rho_{ix}\}$ , and the “warm” temperature grids  $\{\ln T_j\}$ ,  $j = 1, 2, \dots, N_T + 1$  for all `nmatgtes` materials are packed into single 1D arrays `rxlngrgtes(:)` and `Tlngrgtes(:)`, with the offsets for different materials given, respectively, by the elements of the arrays `morxlngrgtes(1:nmatgtes+1)` and `moTlngrgtes(1:nmatgtes+1)`,

$$\begin{aligned} \text{morxlngrgtes}(1) &= 0, \\ \text{morxlngrgtes}(m+1) &= \sum_{k=1}^m [N_{rx}(k) + 1], \quad m = 1, 2, \dots, \text{nmatgtes}, \end{aligned} \quad (6.11)$$

$$\begin{aligned} \text{moTlngrgtes}(1) &= 0, \\ \text{moTlngrgtes}(m+1) &= \sum_{k=1}^m [N_T(k) + 1], \quad m = 1, 2, \dots, \text{nmatgtes}; \end{aligned} \quad (6.12)$$

the values of  $N_{rx} = N_{rx}(m)$  and of  $N_T = N_T(m)$  are stored, respectively, in the arrays `nrxgtes(1:nmatgtes)` and `nTgtes(1:nmatgtes)`. Note that the “warm” temperature grid `Tlngrgtes(1:moTlngrgtes(nmatgtes+1))` does not include the “cold” temperature  $T_x = T_x(m) = T_0$ , which is stored separately as an array `Txgtes(1:nmatgtes)`. The arrays `nrxgtes(1:nmatgtes)`, `rxlngrgtes(1:morxlngrgtes(nmatgtes+1))`, `nTgtes(1:nmatgtes)`, `Tlngrgtes(1:moTlngrgtes(nmatgtes+1))`, and `Txgtes(1:nmatgtes)` are all written into the file ‘`tab-glteos17.f17`’.

#### 4. Structure of the principal arrays `petab17(:)` and `pextab17(:)`

The 1D array `petab17(:)` contains tabulated values of 7 thermodynamic quantities, namely, of the

mass-specific internal energy	$e,$	
pressure	$p,$	
ionization degree	$z,$	
heat capacity at constant volume	$c_v = \left( \frac{\partial e}{\partial T} \right)_\rho,$	(6.13)
$\rho$ -derivative of pressure	$p'_\rho = \left( \frac{\partial p}{\partial \rho} \right)_T,$	
$T$ -derivative of pressure	$p'_T = \left( \frac{\partial p}{\partial T} \right)_\rho,$	
strong-shock parameter	$A_s,$	

for all the `nmatgtes` materials; all the tabulated quantities are considered as functions of two independent variables — the density  $\rho$  and temperature  $T$ . The array `petab17(:)` is

composed as follows

$$\text{petab17}(:) ::= \left. \begin{array}{l} \{e_{i,0}\}, \{e_{i,1}\}, \dots, \{e_{i,N_T+1}\}, \\ \{p_{i,0}\}, \{p_{i,1}\}, \dots, \{p_{i,N_T+1}\}, \\ \{z_{i,0}\}, \{z_{i,1}\}, \dots, \{z_{i,N_T+1}\}, \\ \{c_{v; i,0}\}, \{c_{v; i,1}\}, \dots, \{c_{v; i,N_T+1}\}, \\ \{p'_{\rho; i,0}\}, \{p'_{\rho; i,1}\}, \dots, \{p'_{\rho; i,N_T+1}\}, \\ \{p'_{T; i,0}\}, \{p'_{T; i,1}\}, \dots, \{p'_{T; i,N_T+1}\}, \\ \{A_{s; i,0}\}, \{A_{s; i,1}\}, \dots, \{A_{s; i,N_T+1}\}, \end{array} \right\} m = 1, \quad (6.14)$$

$$\left. \begin{array}{l} \{e_{i,0}\}, \{e_{i,1}\}, \dots, \{e_{i,N_T+1}\}, \\ \{p_{i,0}\}, \{p_{i,1}\}, \dots, \{p_{i,N_T+1}\}, \\ \{z_{i,0}\}, \{z_{i,1}\}, \dots, \{z_{i,N_T+1}\}, \\ \{c_{v; i,0}\}, \{c_{v; i,1}\}, \dots, \{c_{v; i,N_T+1}\}, \\ \{p'_{\rho; i,0}\}, \{p'_{\rho; i,1}\}, \dots, \{p'_{\rho; i,N_T+1}\}, \\ \{p'_{T; i,0}\}, \{p'_{T; i,1}\}, \dots, \{p'_{T; i,N_T+1}\}, \\ \{A_{s; i,0}\}, \{A_{s; i,1}\}, \dots, \{A_{s; i,N_T+1}\}, \end{array} \right\} m = 2, \quad (6.15)$$

$$\dots \quad (6.16)$$

Here

$\{e_{i,0}\}$  is the  $i$ -sequence of  $N_r(m) + 1$  values of the “cold” energies  $e_{i,0} = e_x(\rho_i) = e(\rho_i, T_x)$  for material #  $m$ ;

$\{e_{i,j}\}$  for any  $1 \leq j \leq N_T(m) + 1$  is the  $i$ -sequence of  $N_r(m) + 1$  values of the thermal energy component

$$e_{i,j} = e_{th}(\rho_i, T_j) = e(\rho_i, T_j) - e_x(\rho_i) \quad (6.17)$$

for material #  $m$ ;

$\{p_{i,0}\}$  is the  $i$ -sequence of  $N_r(m) + 1$  values of the “cold” pressures  $p_{i,0} = p_x(\rho_i) = p(\rho_i, T_x)$  for material #  $m$ ;

$\{p_{i,j}\}$  for any  $1 \leq j \leq N_T(m) + 1$  is the  $i$ -sequence of  $N_r(m) + 1$  values of the “warm” pressure  $p_{i,j} = p(\rho_i, T_j)$  for material #  $m$ ;

$\{z_{i,j}\}$  for any  $0 \leq j \leq N_T(m) + 1$  is the  $i$ -sequence of  $N_r(m) + 1$  values of the ionization degree  $z_{i,j} = z(\rho_i, T_j)$  for material #  $m$ , where  $T_0 = T_x$ ; analogously for all the remaining quantities  $c_v$ ,  $p'_{\rho}$ ,  $p'_{T}$ , and  $A_s$ .

As of 2017-07-18, the values of  $A_s$  are all set equal to a universal constant  $A_s = 1.2$ .

To facilitate decomposition of the `petab17(:)` array, an auxiliary array `mogtes(1:nmatgtes+1)` of material offsets is generated:

$$\begin{aligned} \text{mogtes}(1) &= 0, \\ \text{mogtes}(m+1) &= \sum_{k=1}^m 7 * \text{msz1gtes}(k), \quad m = 1, 2, \dots, \text{nmatgtes}, \\ \text{msz1gtes}(m) &= [N_r(m) + 1][N_T(m) + 2] \equiv [\text{nrgtes}(m)+1] * [\text{nTgtes}(m)+2]. \end{aligned} \quad (6.18)$$

With the aid of these offsets, different physical quantities for material #  $m$  can be extracted as

$$\begin{aligned} e_{th}(\rho_i, T_j) &= \text{petab17}(\text{mogtes}(m) + j * \text{nr1} + i), \\ p(\rho_i, T_j) &= \text{petab17}(\text{mogtes}(m) + \text{msz1gtes}(m) + j * \text{nr1} + i), \\ z(\rho_i, T_j) &= \text{petab17}(\text{mogtes}(m) + 2 * \text{msz1gtes}(m) + j * \text{nr1} + i), \\ &\dots \end{aligned} \tag{6.19}$$

where  $\text{nr1} = \text{nrgtes}(m) + 1$ , and  $j = 1, 2, \dots, N_r + 1$ . Evidently, the size of array  $\text{petab17}(:)$  is given by the value of  $\text{mogtes}(\text{nmatgtes} + 1)$ .

Similarly, the 1D array  $\text{pexstab17}(:)$  contains the “cold” values of the 4 thermodynamic quantities

mass-specific internal energy	$e_x,$	
pressure	$p_x,$	
$\rho$ -derivative of pressure	$(p_x)'_{\rho} = \left( \frac{\partial p_x}{\partial \rho} \right)_T,$	(6.20)
ionization degree	$z_x,$	

for all the  $\text{nmatgtes}$  materials; all the tabulated quantities are considered as functions of the density  $\rho$  only. The array  $\text{pexstab17}(:)$  has the following structure

$$\begin{aligned} \text{pexstab17}(:) ::= & \left\{ e_{x; i}, \{ p_{x; i} \}, \{ (p_x)'_{\rho; i} \}, \{ z_{x; i} \}, \right\} & m = 1, \\ & \left\{ e_{x; i}, \{ p_{x; i} \}, \{ (p_x)'_{\rho; i} \}, \{ z_{x; i} \}, \right\} & m = 2, \\ & \dots \end{aligned} \tag{6.21}$$

where  $\{f_i\}$  is the sequence of  $N_{rx}(m) + 1$  values of the “cold” values of the quantity  $f$  for material #  $m$ . For decomposition of  $\text{pexstab17}(:)$  the material offsets

$$\begin{aligned} \text{moxgtes}(1) &= 0, \\ \text{moxgtes}(m+1) &= \sum_{k=1}^m [N_{rx}(k) + 1] \equiv \sum_{k=1}^m [\text{nrxgtes}(k) + 1] \end{aligned} \tag{6.22}$$

are used. The size of  $\text{pexstab17}(:)$  is given by the value of  $\text{moxgtes}(\text{nmatgtes} + 1)$ .

### 5. Interpolation scheme and the GLT-EOS in the caloric form $p = p(\rho, e)$ , $T = T(\rho, e)$

Once all the needed EOS data is read out from the file 'tab-glteos17.f17', the GLT-EOS is calculated by calling the subroutine `GLTEOS17(...)`, based on the following interpolation scheme. Let  $f(\rho, T)$  be any of the 7 thermodynamic quantities tabulated in the “warm” table  $\text{petab17}(:)$ , which provides the values  $f_{ij}$  at the nodes of the 2D grid  $\{\rho_i\} \otimes \{T_j\}$ . Then, for a given pair  $(\rho, T)$ , the value of  $f = f(\rho, T)$  is found by using a bilinear interpolation in the  $(\ln \rho, \ln T)$  coordinates

$$f = f_{i,j}(1 - \xi)(1 - \eta) + f_{i+1,j}\xi(1 - \eta) + f_{i,j+1}(1 - \xi)\eta + f_{i+1,j+1}\xi\eta \tag{6.23}$$

inside the corresponding grid cell  $[i, i + 1] \otimes [j, j + 1]$ . The fractional intra-cell coordinates  $\xi, \eta$  are given by

$$0 \leq \xi = \frac{\ln \rho - \ln \rho_i}{\ln \rho_{i+1} - \ln \rho_i} \leq 1, \quad 0 \leq \eta = \frac{\ln T - \ln T_j}{\ln T_{j+1} - \ln T_j} \leq 1. \quad (6.24)$$

To find the intervals  $[\ln \rho_i, \ln \rho_{i+1}]$  and  $[\ln T_j, \ln T_{j+1}]$ , containing given values of  $\ln \rho$  and  $\ln T$  on arbitrary grids  $\{\rho_i\}$  and  $\{T_j\}$ , a special service **subroutine** `FINDINTV(...)` is called, where the method of bisection of the full interval  $[1, N + 1]$  is applied to the logical (index) coordinates  $i$  (or  $j$ ).

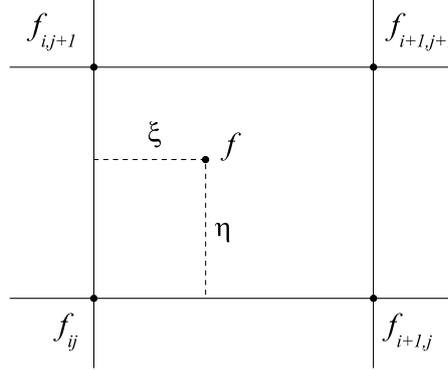


FIG. 6.1: Interpolation of quantity  $f$  inside one grid cell  $ij$  of the GLT table. Interpolated value  $f$  is requested at a point with fractional intra-cell coordinates  $0 \leq \xi \leq 1$  and  $0 \leq \eta \leq 1$ .

When a thermodynamic quantity  $f$  is requested not as a function of  $\rho$  and  $T$  but — as is the case in the RALEF code — as a function of  $\rho$  and  $e$  (the caloric form of EOS), then the interpolation algorithm, based on the necessary condition of the thermodynamic stability  $c_v > 0$ , is as follows. First of all, one finds the density index  $i$  such that  $\rho \in [\ln \rho_i, \ln \rho_{i+1}]$  (by calling the **subroutine** `FINDINTV(...)`) and calculates the fractional intra-cell coordinate  $\xi$ . Then, an auxiliary grid  $\{\tilde{e}_{th,j}\}$ ,  $j = 1, 2, \dots, N_T + 1$  is constructed along the energy axis, where

$$\tilde{e}_{th,j} = (1 - \xi)e_{th,i,j} + \xi e_{th,i+1,j}; \quad (6.25)$$

recall that it is the values of  $e_{th,i,j}$  that are listed in the “warm” table `petab17(:)`. Because  $c_v > 0$ , the values of  $\tilde{e}_{th,j}$  monotonically increase with  $j$ , and again one finds the required  $j$  such that  $e_{th} \equiv e - \tilde{e}_x \in [\tilde{e}_{th,j}, \tilde{e}_{th,j+1}]$  by calling the **subroutine** `FINDINTV(...)`; here

$$\tilde{e}_x = (1 - \xi)e_{x,i} + \xi e_{x,i+1} \quad (6.26)$$

is the density-interpolated value of the cold energy. Finally, the fractional intra-cell coordinate  $\eta$  is calculated as

$$\eta = \frac{e_{th} - \tilde{e}_{th,j}}{\tilde{e}_{th,j+1} - \tilde{e}_{th,j}}. \quad (6.27)$$

Note that, because  $e_{th}, e_{th,ij} > 0$ , more logical (but computationally more expensive) would be to use the formula

$$\eta = \frac{\ln e_{th} - \ln \tilde{e}_{th,j}}{\ln \tilde{e}_{th,j+1} - \ln \tilde{e}_{th,j}}. \quad (6.28)$$

With thus determined  $i, j$  and  $\xi, \eta$ , one calculates all the needed quantities from Eq. (6.23). In particular,

$$T = T(\rho, e) = \exp [(1 - \eta) \ln T_j + \eta \ln T_{j+1}], \quad (6.29)$$

$$p = p(\rho, e) = p_{i,j}(1 - \xi)(1 - \eta) + p_{i+1,j}\xi(1 - \eta) + p_{i,j+1}(1 - \xi)\eta + p_{i+1,j+1}\xi\eta, \quad (6.30)$$

etc.

#### 6. The Maxwell construction for the equilibrium (EQ) version of the GLT-EOS

The thermodynamic data for the main tables `petab17(:)` and `pexstab17(:)` are calculated in `program GTESFILL17` by using one of the several available source-EOS models. If the source EOS exhibits the van-der-Waals-like loops, it is only used to generate the metastable (MS) EOS on a chosen  $\{\rho_i\} \otimes \{T_j\}$  grid, which literally reproduces those loops. Because such an EOS has a region of absolute thermodynamic instability (below the spinodal), one would certainly like to have an option for an absolutely stable (equilibrium) EOS (the EQ-EOS), where the phase coexistence region below the binodal is bridged by the Maxwell construction. In the GLT-EOS17 this option can be inacted for any material  $m$  by assigning the logical flag `Maxwell(m)=.true.` (written into the file `'tab-glteos17.f17'`). In the present version of the GLT-EOS17, the EQ-EOS with the Maxwell construction is not invoked in the source EOS but is, rather, calculated separately starting from the metastable version of the GLT-EOS in the tabular form. In other words, any employed source EOS must not necessarily have an EQ option inside it in order to get an EQ version of the GLT-EOS in its final tabular form.

In summary, the general logical scheme of the algorithm, implemented in the `program GTESFILL17` to compute all the data written into file `'tab-glteos17.f17'`, is as follows:

1. compute the parameters  $\rho_{cp}, T_{cp}, p_{cp}$  of the liquid-vapor critical point by calling the `subroutine CRIPPOINT(...)` from the GLT-EOS package; this routine deals exclusively with the source EOS in its MS version; calculated values of  $\rho_{cp}, T_{cp}, p_{cp}$  are written into the file `'tab-glteos17.f17'`; the critical temperature  $T_{cp}$  is defined such that for all  $T > T_{cp}$  the isotherms  $p(\rho, T)$  are monotonically non-increasing functions of  $\rho$ ;
2. with the values of  $\rho_{cp}, T_{cp}, p_{cp}$  known, construct the main “warm” grid  $\{\rho_i\} \otimes \{T_j\}$  and the “cold” grid  $\{\rho_{ix}\}$  for the tabular GLT-EOS [for which a special `subroutine GENGRI1CP(...)` from the GLT-EOS package is used]; normally a finer grid resolution is required in the vicinity of the critical point;
3. compute all the EOS data under the assumption that an MS version (without the Maxwell construction) of the GLT-EOS is to be generated; ascertain that for every fixed  $i$  the internal energy  $e_{ij}$  of the computed MS EOS is a monotonically increasing function of the  $j$ -index; if the user-defined flag `Maxwell(m)=.false.`, finish the job by writing the computed data into the file `'tab-glteos17.f17'`;
4. if, however, `Maxwell(m)=.true.`, then, for every discrete MS isotherm  $\{p_{ij}\}$  (the  $i$ -sequence for a given fixed  $j$ ) with  $0 \leq j < \text{njbind}(m)$  lying below the critical point  $T_j < T_{cp}$ , do the following

- compute the vapor and liquid spinodal densities  $\rho_{v-sp,j}$ ,  $\rho_{l-sp,j}$ , defined, respectively, as the positions of the first local maximum (the last local minimum) on the  $\tilde{p}_j(\rho)$  isotherm along the increasing  $\rho$ , and the corresponding pressures  $p_{v-sp,j}$ ,  $p_{l-sp,j}$  (all written into the file 'tab-glteos17.f17'); all the calculations are done for the piece-wise linear (in  $\ln \rho$ ) function

$$\tilde{p}_j(\rho) = (1 - \xi)p_{ij} + \xi p_{i+1,j}, \quad \xi = \frac{\ln \rho - \ln \rho_i}{\ln \rho_{i+1} - \ln \rho_i} \quad \text{at } \rho_i \leq \rho \leq \rho_{i+1}, \quad (6.31)$$

where  $p_{ij} = p(\rho_i, T_j)$  are the initially tabulated MS-EOS pressures; it is designated as a “discrete” isotherm;

- compute the Maxwell shelves [there may be several of them on any given isotherm  $\tilde{p}_j(\rho)$ ] and the vapor and liquid binodal densities  $\rho_{v-bi,j}$ ,  $\rho_{l-bi,j}$ , defined, respectively, as the start (finish) points of the first (last) Maxwell shelf on the  $\tilde{p}_j(\rho)$  isotherm along the increasing  $\rho$ , and the corresponding pressures  $p_{v-bi,j}$ ,  $p_{l-bi,j}$  (all written into the file 'tab-glteos17.f17'); a single Maxwell shelf  $p = p_s$  is computed by solving the equation

$$p_s \left( \frac{1}{\rho_v} - \frac{1}{\rho_l} \right) = \int_{\rho_l^{-1}}^{\rho_v^{-1}} \tilde{p}_j(\rho) d\rho^{-1} \quad (6.32)$$

for the value of the shelf pressure  $p_s$ , where  $\rho_v = \rho_v(p_s)$  and  $\rho_l = \rho_l(p_s)$  are the vapor and the liquid roots of the equation  $\tilde{p}_j(\rho) = p_s$ ;

- modify the values of  $e_{ij}$ ,  $p_{ij}$ , and  $(\partial p / \partial \rho)_{ij}$  in the phase coexistence region  $\rho_{v-bi,j} \leq \rho \leq \rho_{l-bi,j}$  to their EQ-EOS values;
5. ascertain that for every fixed  $i$  the internal energy  $e_{ij}$  of the computed EQ EOS is a monotonically increasing function of the  $j$ -index; compute anew the temperature derivatives  $c_{v,ij} = (\partial e / \partial T)_{ij}$ ,  $(\partial p / \partial T)_{ij}$  in the phase coexistence region; finish the job by writing the computed data into the file 'tab-glteos17.f17'.

Note that Eq.(6.33) can be rewritten as

$$F(p_s) \equiv 1 - \frac{\rho_v}{\rho_l} - \int_{\ln \rho_v}^{\ln \rho_l} \left( \frac{\tilde{p}_j \rho_v}{p_s \rho} \right) d \ln \rho = 0. \quad (6.33)$$

With the EQ pressure  $p$  being constant, the EQ internal energy  $e(\rho)$  along the Maxwell shelf (a linear function of  $\rho^{-1}$ ) is calculated as

$$e(\rho) = \left[ e_v \left( \frac{\rho_v}{\rho} - \frac{\rho_v}{\rho_l} \right) + e_l \left( 1 - \frac{\rho_v}{\rho} \right) \right] \left( 1 - \frac{\rho_v}{\rho_l} \right)^{-1}. \quad (6.34)$$

In practice, instead of a single local maximum followed by a single local minimum, the “discrete” MS isotherms (6.31) may exhibit multiple local maxima and minima (usually, due to finite-difference errors in the algorithm implemented in a given source EOS), as is illustrated in Fig. 6.2 for the FEOS-16 equation of state of Sn. In such a case, the second substep in step 4 of the above algorithm is executed iteratively, until no local maxima (minima) are found on the corresponding Maxwell-bridged isotherm; the results are shown in the same Fig. 6.2. Note that the binodal and spinodal curves, obtained from such multi-humped “discrete” isotherms, may look rather ugly close to the critical point.

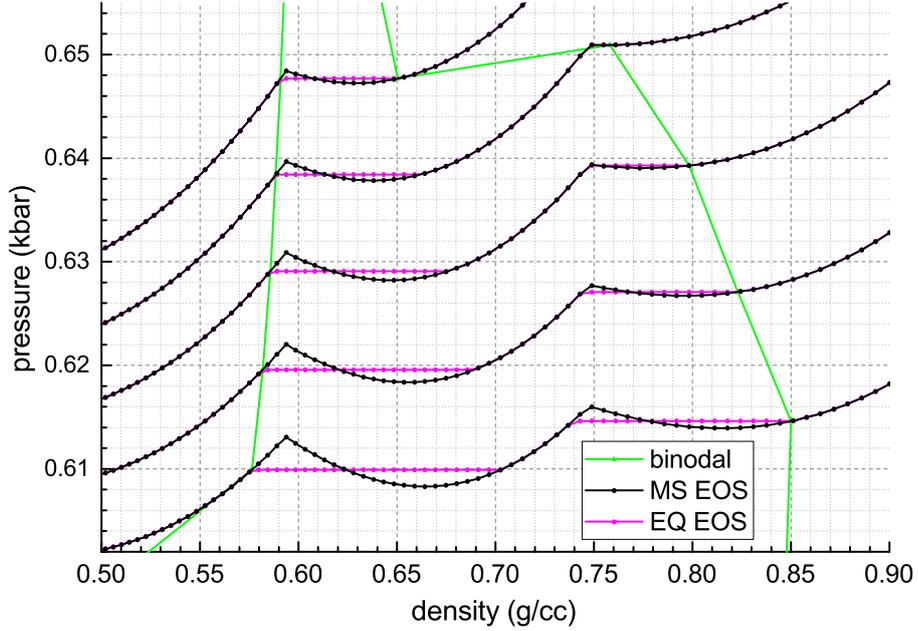


FIG. 6.2: Black: the “discrete” MS-EOS isotherms (6.31), calculated with the FEOS-16 source EOS for tin (Sn) immediately below the critical point of  $\rho_{cp} = 0.594$  g/cc,  $p_{cp} = 683$  bar. Magenta: the EQ-EOS isotherms, calculated with the present algorithm from the “discrete” MS-EOS isotherms. Green: the “discrete” binodal. Dots along each isotherm indicate the nodes of the density grid  $\{\rho_i\}$ .

### 7. Extrapolation beyond the main “warm” table `petab17(:)`

If the GLT-EOS is requested for a  $(\rho, T)$  point, which lies outside the main “warm” table `petab17(:)`, it is evaluated by using an extrapolation algorithm, based on the Mie-Grüneisen analytical EOS

$$e(\rho, T) = e_x(\rho) + c_{vo}T, \quad (6.35)$$

$$p(\rho, T) = p_x(\rho) + \Gamma c_{vo}\rho T, \quad (6.36)$$

$$T(\rho, e) = [e - e_x(\rho)] / c_{vo}, \quad (6.37)$$

$$p(\rho, e) = p_x(\rho) + \Gamma\rho [e - e_x(\rho)], \quad (6.38)$$

$$c_s^2(\rho, e) = \frac{dp_x}{d\rho} + \Gamma(\Gamma + 1)\rho [e - e_x(\rho)]. \quad (6.39)$$

Here the factors  $c_{vo}$  and  $\Gamma$  are assumed to be constant, i.e. independent of  $\rho$  and  $T$  (or of  $\rho$  and  $e$ ). The cold pressure  $p_x(\rho)$  and the cold energy  $e_x(\rho)$  are supposed to be available from the “cold” table `pexstab17(:)` for any value of  $\rho$  that might be requested by the hydro code. In the caloric form of EOS, it must, of course, be ensured that  $e \geq e_x(\rho)$  for any requested value of  $\rho$ .

To calculate the two unknown parameters  $c_{vo}$  and  $\Gamma$  in Eqs. (6.35)-(6.39), one can make use of the values of thermodynamic variables at the point  $b$  of the outer boundary of the “warm” table in the  $(\ln \rho, \ln T)$  plane, which is the closest to the requested point  $(\rho, T)$ . Figure 6.3 shows schematically how the coordinates  $(\rho_b, T_b)$  are established, when the requested

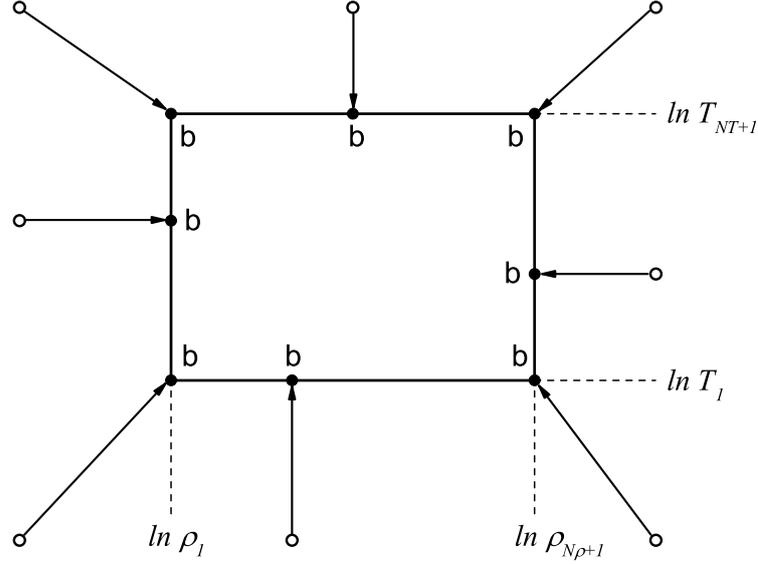


FIG. 6.3: Schematics of projection of an arbitrary point  $(\rho, T)$  (open circles), lying outside the main “warm” table, onto its outer boundary to the closest point  $b$  (black dots) with  $\rho = \rho_b$ ,  $T = T_b$ .

point  $(\rho, T)$  lies at different locations relative to the rectangular domain of the “warm” table in the  $(\ln \rho, \ln T)$  plane. With the  $b$ -point found, one obtains

$$c_{vo} = \frac{e(\rho_b, T_b) - e_x(\rho_b)}{T_b}, \quad \Gamma = \frac{p(\rho_b, T_b) - p_x(\rho_b)}{\rho_b [e(\rho_b, T_b) - e_x(\rho_b)]}. \quad (6.40)$$

Strictly speaking, the extrapolated EOS (6.35)-(6.40) is thermodynamically not consistent because the position of the boundary point  $(\rho_b, T_b)$  — hence the values of the parameters  $c_{vo}$ ,  $\Gamma$  — depend on  $\rho$  and  $T$ . In reality, however, when the “warm” table occupies a sufficiently wide domain, variation of  $c_{vo}$  and  $\Gamma$  with  $\rho$  and  $T$  is often rather weak. At the same time, the main goal of the extrapolation (6.35)-(6.40) is to provide a simple and physically reasonable — if not particularly accurate — estimate for the thermodynamic parameters outside the domain of the main table. Clearly, the farther the requested point  $(\rho, T)$  from the table domain, the less accurate will such an extrapolation be.

Extrapolation of the mean ionization degree  $z$ , which does not directly enter the equations of hydrodynamics but is needed for evaluation of various transport coefficients, should be considered separately. In the present version of the GLT-EOS17, the following extrapolation scheme for  $z(\rho, T)$  has been adopted.

- Low-temperature limit,  $T \leq T_{NT+1}$ :

$$z(\rho, T) = \max [z_x(\rho), z(\rho_b, T_b)], \quad (6.41)$$

where

$$z_x(\rho) = z(\rho, T_x) \quad (6.42)$$

is the cold ionization degree.

- High-temperature limit,  $T > T_{N_T+1}$ :

$$z(\rho, T) = \max [z_x(\rho), \min(Z, \tilde{z}_T)], \quad (6.43)$$

where

$$\tilde{z}_T = z(\rho_b, T_b) \left( \frac{T}{T_b} \right)^\alpha, \quad (6.44)$$

$Z$  is the atomic number of the considered element material, and the exponent  $\alpha$  is determined from the nearest boundary point  $b$ ,

$$\alpha = \frac{\ln z_{i,N_T+1} - \ln z_{i,N_T}}{\ln T_{N_T+1} - \ln T_{N_T}}; \quad (6.45)$$

$i$  is the index along the  $\{\ln \rho_i\}$  grid of the cell in the `petab17(:)` table, which contains the boundary point  $b$ .

### 3. Tabular GLT conduction coefficient and opacities

In the new version “GLT-TC-OPA17” of the GLT tables for thermal conduction coefficient  $\kappa_e$  and opacities (including, in particular, the spectral absorption coefficient  $k_\nu$  and the non-LTE spectral source function  $G_\nu$ ), implemented in the year 2017, all the data, required by the RALEF code for a given job, is written into a single file in the “f17” format with a fixed name `'input/tab-gltcrad.f17'`. This file may contain the TC-OPA data for an arbitrary number of materials, numbered sequentially by index

$$\begin{aligned} m = 1, 2, \dots, \text{nmatgtc} &\rightarrow \text{for conduction,} \\ m = 1, 2, \dots, \text{nmatgtr} &\rightarrow \text{for opacities.} \end{aligned} \quad (6.46)$$

The values of `nmatgtc`, `nmatgtr` are written into the `'tab-gltcrad.f17'` file as a separate item `nmat_gtcr`.

The GLT-TC-OPA data file `'tab-gltcrad.f17'` is generated by running the program `GTCRFILL17` (file `'tabtcr.GLT.f'`), belonging to a separate GLT-TC-OPA code package, which is fully independent of the RALEF package. Unless the test subroutine `GTCRTST17` (called from program `GTCRFILL17`) is executed, the only communication between the GLT-TC-OPA and the RALEF packages is by means of the module `COMGLT17`, which contains all the necessary common variables and arrays, and which is included into the file `'f00_comod.f'` from the RALEF package.

...

The general logarithmic table (the GLT-TCRAD table) for thermal conduction coefficient  $\kappa = \kappa(\rho, T)$ , mean Rosseland,  $k_{Ros} = k_{Ros}(\rho, T)$ , mean Planckian,  $k_{Pl} = k_{Pl}(\rho, T)$ , and group

averaged spectral,  $k_{[k]} = k_{[k]}(\rho, T)$ , absorption coefficients contains an array of tabular values  $f_{ij}$ , where

$$f = \begin{cases} \ln \kappa, \\ \ln (k_{Ros}/\rho), \\ \ln (k_{Pl}/\rho), \\ \ln (k_{[k]}/\rho), \quad [k] = 1, 2, \dots, N_\nu. \end{cases} \quad (6.47)$$

Tabulated are the values  $f_{ij}$  on a uniform grid

$$\begin{aligned} \ln \rho_i &= \ln \rho_0 + (i - 1)d_{lr}, \quad i = 1, 2, \dots, N_r + 1, \\ \ln T_j &= \ln T_0 + (j - 1)d_{lT}, \quad j = 1, 2, \dots, N_T + 1, \end{aligned} \quad (6.48)$$

of  $\ln \rho$  and  $\ln T$  values. Table dimensions  $N_r$  and  $N_T$  may differ between the tables for the conduction coefficient and opacities, and between different materials. The table bounds are given by

$$\begin{aligned} \rho_0 &\leq \rho \leq \rho_1 = \rho_0 \exp(N_r d_{lr}) \\ T_0 &\leq T \leq T_1 = T_0 \exp(N_T d_{lT}). \end{aligned} \quad (6.49)$$

### 1. Interpolation method

For a given pair  $(\rho, T)$ , the value of  $f = f(\rho, T)$  is found from the corresponding table by using a bilinear interpolation in the  $(\ln \rho, \ln T)$  coordinates. First, the indices  $i$  and  $j$  of the corresponding table-grid cell  $ij$  are found:  $i$  and  $j$  are the indices of the lower-left corner of the grid cell containing the point  $(\ln \rho, \ln T)$  (see Fig. 6.1):

$$x = \frac{\ln \rho - \ln \rho_0}{d_{lr}}, \quad y = \frac{\ln T - \ln T_0}{d_{lT}}, \quad (6.50)$$

$$i = \begin{cases} \min(N_r, [x] + 1), & x > 0, \\ 1, & x \leq 0, \end{cases} \quad j = \begin{cases} \min(N_T, [y] + 1), & y > 0, \\ 1, & y \leq 0, \end{cases} \quad (6.51)$$

where  $[x]$  is the integer part of  $x$ ,  $[y]$  is the integer part of  $y$ . Then the interpolated value  $f = f(\xi, \eta)$  is given by

$$f(\xi, \eta) = f_{ij}(1 - \xi)(1 - \eta) + f_{i+1,j}\xi(1 - \eta) + f_{i,j+1}(1 - \xi)\eta + f_{i+1,j+1}\xi\eta, \quad (6.52)$$

where

$$\xi = x - i + 1, \quad \eta = y - j + 1. \quad (6.53)$$

Note that the above interpolation formula (6.52) can be automatically extended beyond the table limits, i.e. for  $\xi < 0$  or  $\xi > 1$ , and for  $\eta < 0$  or  $\eta > 1$ .

### 2. Extrapolation beyond table limits for thermal conduction

If the requested point  $(\rho, T)$  is outside the table grid but the condition

$$\rho \leq \rho_1 \quad \text{and} \quad T \leq T_1 \quad (6.54)$$

is fulfilled, we evaluate the thermal conduction coefficient  $\kappa = \exp(f)$  by using the basic bilinear interpolation (6.52), i.e. by using a power-law extrapolation  $\kappa = A\rho^\alpha T^\beta$  with constants  $A$ ,  $\alpha$  and  $\beta$  taken from the nearest table-grid cell. For high densities and temperatures, i.e. when the condition

$$\rho > \rho_1 \quad \text{or} \quad T > T_1 \quad (6.55)$$

is fulfilled, we use a simple asymptotic formula

$$\begin{aligned} \kappa_{as}(\rho, T) &= \frac{3 \cdot 5^3}{2^7 \sqrt{\pi}} \frac{1 + \frac{433}{180\sqrt{2}} Z}{1 + \frac{151}{36\sqrt{2}} Z + \frac{217}{288} Z^2} \frac{T \left[ T^{3/2} + \beta_F \epsilon_F^{3/2} \right]}{m_e^{1/2} e^4 L} \\ &= 1.470 \times 10^{50} \frac{1 + 1.70 Z}{1 + 2.966 Z + 0.7535 Z^2} T^{5/2} \left( 1 + 4.9 \times 10^{-17} \frac{\rho Z}{AT^{3/2}} \right), \end{aligned} \quad (6.56)$$

which combines the results of L. Spitzer, R. Härm [7] and V.S. Imshennik [8] for a non-degenerate plasma, and of M. Lampe [9, 10] for a degenerate plasma. In Eq. (6.56)  $Z$  is the atomic number of the element (full ionization is assumed),  $A$  is its atomic mass,  $m_e$  is the electron mass,  $e$  is the electron charge,  $L$  is the Coulomb logarithm, which is set equal to a fixed value  $L = 7$ ,

$$\epsilon_F = \frac{\hbar^2}{2m_e} (3\pi^2 n_e) \quad (6.57)$$

is the Fermi energy,

$$\beta_F = \frac{\pi^{5/2}}{96} = 0.1822 \quad (6.58)$$

is the matching constant between the Maxwellian and Fermi asymptotics for the Lorentzian plasma.

A smooth transition from the regular interpolation (6.52) to the asymptotic value (6.56) is effected by a simple formula

$$\ln \kappa = \frac{f + [(0.5\Delta_\rho)^2 + (0.5\Delta_T)^2] \ln \kappa_{as}}{1 + [(0.5\Delta_\rho)^2 + (0.5\Delta_T)^2]}, \quad (6.59)$$

where

$$\Delta_\rho = \max(0, \ln \rho - \ln \rho_1), \quad \Delta_T = \max(0, \ln T - \ln T_1), \quad (6.60)$$

and  $f$  is a regular bilinear interpolation (6.52).

### 3. Extrapolation beyond table limits for radiation absorption coefficients

For  $\ln k_{Ros}$ ,  $\ln k_{Pl}$ , and  $\ln k_{[k]}$  extrapolation beyond the table limits is performed in two steps: first along the  $\ln \rho$  axis, and then along the  $\ln T$  axis.

Extrapolation along the  $\ln \rho$  axis starts from the initial value

$$f_b = f(\xi_{in}, \eta_{in}), \quad (6.61)$$

calculated from Eq. (6.52) at a point

$$\xi_{in} = \min [1, \max(0, \xi)], \quad \eta_{in} = \min [1, \max(0, \eta)], \quad (6.62)$$

on the table boundary [in cell  $ij$  given by Eq. (6.51)], which is the point within the table limits closest to the outside point  $(\ln \rho, \ln T)$  in question. Extrapolation along the  $\ln \rho$  axis is done by using the logarithmic derivative

$$f'_{b\rho} = [(f_{i+1,j} - f_{ij})(1 - \eta_{in}) + (f_{i+1,j+1} - f_{i,j+1})\eta_{in}] / d_{l\rho}, \quad (6.63)$$

evaluated from the nearest table-border cell  $ij$  and constrained to the interval  $[0, 1]$ , by using the formula

$$f_{e1} = f_b + f'_{b\rho,lim} d_{l\rho} \xi_{out}, \quad (6.64)$$

where

$$f'_{b\rho,lim} = \min [1, \max(0, f'_{b\rho})], \quad (6.65)$$

$$\xi_{out} = \begin{cases} \xi, & \xi < 0, \\ 0, & 0 \leq \xi \leq 1, \\ \xi - 1, & \xi > 1. \end{cases} \quad (6.66)$$

Equation (6.65) constrains the logarithmic derivatives  $d \ln k_{Ros} / d \ln \rho$ ,  $d \ln k_{Pl} / d \ln \rho$ , and  $d \ln k_{[k]} / d \ln \rho$  to the interval  $[1, 2]$ , which agrees with the physical models for the free-free, free-bound and Compton-scattering components of the opacity.

Now, the extrapolation along the  $\ln T$  axis starts from the value  $f_{e1}$ . It is based on different formulae for the low-temperature,  $T < T_0$ , and the high-temperature,  $T > T_1$ , regions. For the low-temperature region we use the extrapolation formula

$$f = f_{e1} + \frac{f'_{bT}}{1 + |y|} y, \quad T < T_0, \quad (6.67)$$

where

$$y = \ln \left( \frac{T}{T_0} \right), \quad (6.68)$$

and

$$f'_{bT} = [(f_{i,j+1} - f_{ij})(1 - \xi_{in}) + (f_{i+1,j+1} - f_{i+1,j})\xi_{in}] / d_{lT} \quad (6.69)$$

is the logarithmic derivative, evaluated from the nearest table-border cell  $ij$  [as given by Eq. (6.51)]. Formula (6.67) ensures a smooth transition to the temperature-independent radiation absorption coefficients in the limit of very low temperatures  $T \ll T_0$ .

For high temperatures  $T > T_1$  we make a smooth transition to an asymptotic opacity of a fully ionized mono-element (atomic number  $Z$ , atomic mass  $A$ ) plasma by applying an extrapolation formula

$$f = \frac{f_{e1} + f_{ff} \tilde{y}^2}{1 + \tilde{y}^2} + \frac{f'_{bT} - \mu y^2}{1 + 2y^2} y, \quad (6.70)$$

where  $f_{ff}$  is the asymptotic free-free value for the corresponding opacity component,

$$\mu = \begin{cases} 7, & f = \ln(k_{Ros}/\rho) \quad \text{or} \quad \ln(k_{Pl}/\rho), \\ 1, & f = \ln(k_{[k]}/\rho), \end{cases} \quad (6.71)$$

$$\tilde{y} = \begin{cases} 0, & T \leq T_Z, \\ \ln(T/T_Z), & T > T_Z > T_1, \\ y, & T > T_1 \geq T_Z, \end{cases} \quad (6.72)$$

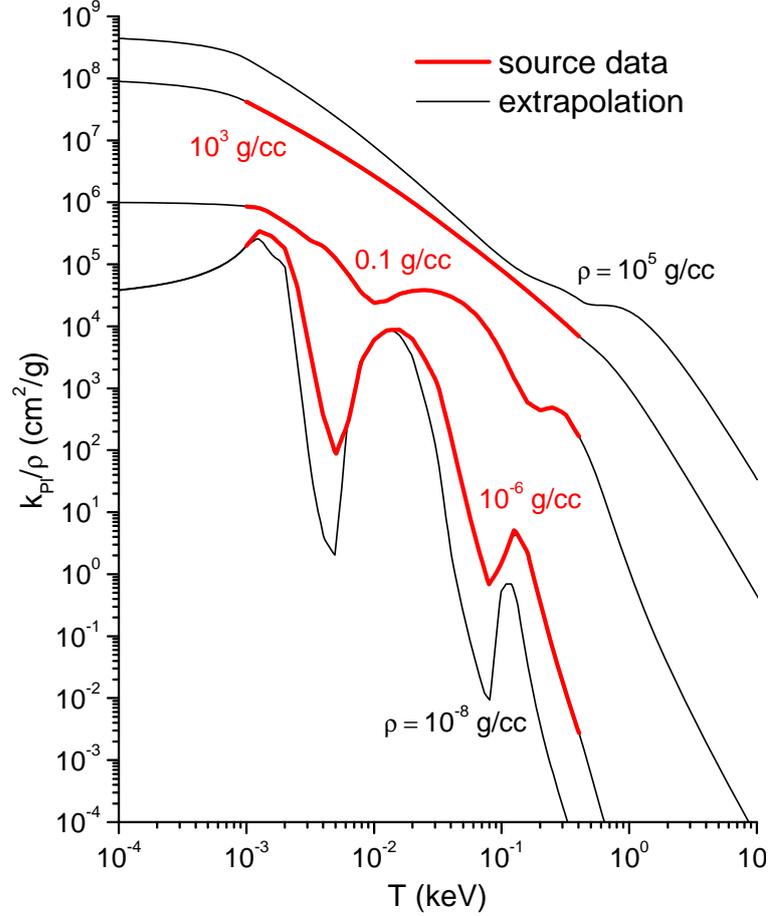


FIG. 6.4: Extrapolation of the Planckian mean absorption coefficient  $k_{pl}$  of aluminum beyond the limits  $\rho_0 = 10^{-6} \text{ g cm}^{-3}$ ,  $\rho_1 = 10^3 \text{ g cm}^{-3}$ ,  $T_0 = 0.001 \text{ keV}$ ,  $T_1 = 0.4 \text{ keV}$ .

and

$$T_Z = 2Z^2 \text{ [eV]} \approx \frac{1}{7} I_H Z^2; \tag{6.73}$$

$I_H = 13.6 \text{ eV}$  is the ionization threshold of hydrogen. We use the Kramers formula

$$\begin{aligned} k_{ff,\nu} &= \frac{32\pi^3}{3\sqrt{6\pi}} \alpha a_0^5 \left(\frac{e^2/a_0}{T}\right)^{1/2} \left(\frac{e^2/a_0}{h\nu}\right)^3 n_i^2 Z^3 (1 - e^{-h\nu/T}) \\ &= \frac{2.7805}{T_{keV}^{1/2} \nu_{keV}^3} \frac{\rho^2 Z^3}{A^2} (1 - e^{-h\nu/T}) \text{ [cm}^{-1}] \end{aligned} \tag{6.74}$$

for fully ionized plasma of element  $Z$  as the asymptotic limit for the spectral absorption

coefficient  $k_\nu$ , and the corresponding Kramers formulae

$$\begin{aligned} k_{ff,Ros} &= \frac{32\pi^3}{3\sqrt{6\pi}} \alpha a_0^5 \left( \frac{e^2/a_0}{T} \right)^{7/2} n_i^2 Z^3 \left[ \frac{15}{4\pi^4} \int_0^\infty \frac{x^7 e^{-x} dx}{(1 - e^{-x})^3} \right]^{-1} \\ &= \frac{1}{70.677} \frac{\rho^2 Z^3}{A^2 T_{keV}^{7/2}} [\text{cm}^{-1}], \end{aligned} \quad (6.75)$$

$$k_{ff,Pl} = \frac{160}{\pi\sqrt{6\pi}} \alpha a_0^5 \left( \frac{e^2/a_0}{T} \right)^{7/2} n_i^2 Z^3 = \frac{1}{2.3355} \frac{\rho^2 Z^3}{A^2 T_{keV}^{7/2}} [\text{cm}^{-1}], \quad (6.76)$$

for the mean Rosseland and Planckian absorption coefficients. In the above formulae  $e$  is the elementary charge,  $\alpha = e^2/\hbar c = 1/137.036$  is the fine structure constant,  $a_0 = \hbar^2/m_e e^2 = 0.52918 \times 10^{-8}$  cm is the Bohr radius,  $n_i$  is the number of plasma ions per unit volume. In Eq. (6.70)  $f_{ff}$  is either  $\ln(k_{ff,Ros}/\rho)$ , or  $\ln(k_{ff,Pl}/\rho)$ , or  $\ln(k_{ff,\nu}/\rho)$  calculated at  $T = T_1$ .

Figure 6.4 illustrates the above described extrapolation algorithm on the example of the Planckian mean absorption coefficient  $k_{Pl}$  of aluminum.

#### 4. Ionization degree at low energy densities

In many EOS models, based on the Thomas-Fermi (TF) approximation, the TF model produces absolutely unacceptable values of the mean equilibrium ionization degree at low temperatures ( $T \lesssim 0.5$  eV) and densities ( $\rho \lesssim 0.1\text{--}1$  g/cc), and especially so when the Maxwell construction is used in the liquid-vapor phase coexistence region. Here we propose a simple correction for mono-atomic materials, which automatically ensures a smooth transition to the Saha limit at low densities and  $T \ll I_1$ , where  $I_1$  is the first ionization potential. Presently, this correction is implemented in the RALEF code for the FEOS source-EOS model.

At low energy densities, the TF ionization degree is artificially suppressed by a factor  $f_{zcor} = f_{zcor}(\rho, T) \leq 1$ , which is a continuous function of density  $\rho$  and temperature  $T$ . It is calculated as

$$f_{zcor}(\rho, T) = \min \left\{ 1, \sqrt{\phi_\rho^2(\rho) + \phi_T^2(T)} \right\}, \quad (6.77)$$

$$\phi_\rho(\rho) = \begin{cases} 1, & \rho \geq \rho_{led}, \\ \exp(1 - \rho_{led}/\rho), & \rho < \rho_{led}, \end{cases} \quad (6.78)$$

$$\phi_T(T) = \begin{cases} 1, & T \geq T_{led}, \\ \exp(I_1/T_{led} - I_1/T), & T < T_{led}, \end{cases} \quad (6.79)$$

where  $\rho_{led}$  and  $T_{led}$  are user-defined threshold density and temperature values.

With thus defined suppression factor  $f_{zcor}$ , the original TF ionization degree  $z_{ion,TF}$  is replaced by

$$z_{ion} = f_{zcor} z_{ion,TF} + (1 - f_{zcor}) z_{ion,1}, \quad (6.80)$$

where  $z_{ion,1}$  is the Saha ionization degree, calculated under the assumption that only neutral

atoms and singly ionized ions are present in the plasma [31, §106]

$$z_{ion,1} = \frac{2}{1 + \sqrt{1 + \Omega}}, \tag{6.81}$$

$$\begin{aligned} \Omega &= 2n \left( \frac{2\pi\hbar^2}{m_e T} \right)^{3/2} \exp\left(\frac{I_1}{T}\right) = \\ &= 8.09182396 \times 10^{-16} \frac{[m]}{[l]^3 [T]^{3/2}} \frac{\rho e^{I_1/T}}{A T^{3/2}} = 399.007 \frac{\rho_{g/cc} e^{I_1/T}}{A T_{eV}^{3/2}}, \end{aligned} \tag{6.82}$$

where  $n = \rho/(m_u A)$  is the total number of nuclei per unit volume. As  $\rho$  and  $T$  decrease below their thresholds of  $\rho_{led}$  and  $T_{led}$ , the above formulae ensure a smooth transition to the Saha limit of a singly ionized plasma.

From physical considerations it is clear that  $T_{led}$  must have a value about  $(0.1-0.2)I_1$ , while  $\rho_{led}$  must be approximately the lowest density of a liquid metal where the liquid-vapor phase separation surface can still exist, i.e. about twice the critical density for the corresponding liquid state. Presently, in RALEF the default general values of

$$T_{led} = 0.2I_1, \quad I_1 = 10 \text{ eV}, \quad \rho_{led} = 0.02A \tag{6.83}$$

are used unless better values are known from the experiments. For tin (Sn), for example, the values

$$T_{led} = 0.2I_1, \quad I_1 = 7.3 \text{ eV}, \quad \rho_{led} = 1.2 \text{ g/cc} \tag{6.84}$$

are used.

---

*Correspondence with the code variables:*

$N_r =$	<code>nrgtc(m)</code>	number of grid cells along $\ln \rho$ for material <b>m</b> in the GLT-TC table for thermal conduction coefficient $\kappa$ ;
	<code>nrgtr(m)</code>	number of grid cells along $\ln \rho$ for material <b>m</b> in the GLT-RAD table for radiation absorption coefficients $k_{Ros}$ , $k_{Pl}$ , $k_{[k]}$ ;
$N_T =$	<code>nTgtc(m)</code>	number of grid cells along $\ln T$ for material <b>m</b> in the GLT-TC table for thermal conduction coefficient $\kappa$ ;
	<code>nTgtr(m)</code>	number of grid cells along $\ln T$ for material <b>m</b> in the GLT-RAD table for radiation absorption coefficients $k_{Ros}$ , $k_{Pl}$ , $k_{[k]}$ ;
$\rho_0 =$	<code>rho0c(m)</code>	lower table bound along $\ln \rho$ for material <b>m</b> in the GLT-TC table for thermal conduction coefficient $\kappa$ ;
	<code>rho0r(m)</code>	lower table bound along $\ln \rho$ for material <b>m</b> in the GLT-RAD table for radiation absorption coefficients $k_{Ros}$ , $k_{Pl}$ , $k_{[k]}$ ;
$\rho_1 =$	<code>rho1c(m)</code>	upper table bound along $\ln \rho$ for material <b>m</b> in the GLT-TC table for thermal conduction coefficient $\kappa$ ;
	<code>rho1r(m)</code>	upper table bound along $\ln \rho$ for material <b>m</b> in the GLT-RAD table for radiation absorption coefficients $k_{Ros}$ , $k_{Pl}$ , $k_{[k]}$ ;
$T_0 =$	<code>temp0c(m)</code>	lower table bound along $\ln T$ for material <b>m</b> in the GLT-TC table for thermal conduction coefficient $\kappa$ ;
	<code>temp0r(m)</code>	lower table bound along $\ln T$ for material <b>m</b> in the GLT-RAD table for radiation absorption coefficients $k_{Ros}$ , $k_{Pl}$ , $k_{[k]}$ ;
$T_1 =$	<code>temp1c(m)</code>	upper table bound along $\ln \rho$ for material <b>m</b> in the GLT-TC table for thermal conduction coefficient $\kappa$ ;
	<code>temp1r(m)</code>	upper table bound along $\ln \rho$ for material <b>m</b> in the GLT-RAD table for radiation absorption coefficients $k_{Ros}$ , $k_{Pl}$ , $k_{[k]}$ ;

---

## 7. MESH REZONING AND REMAPPING OF PRINCIPAL VARIABLES

### 1. General scheme

By the end of the Lagrangian phase of the  $(n + 1)$ -th hydrocycle, known are the new values  $m_i^{L,n+1} = m_i^n$  (cell masses),  $\vec{u}_{c,i}^{L,n+1}$  (cell-centered fluid velocities),  $e_i^{L,n+1}$  (mass-specific total energies) of the three principal hydro variables, which must be ascribed to mesh cells comoving with the fluid. Once the fluid velocities  $\vec{u}_i$  at the mesh nodes are calculated in subroutine LAGVEL, the mesh is advanced from its old ALE configuration  $\vec{x}_i^n$  to the new Lagrangian configuration as

$$\vec{x}_i^{L,n+1} = \vec{x}_i^n + \vec{u}_i \cdot \Delta t, \quad (7.1)$$

where  $\Delta t$  is the hydrodynamic time step of the  $(n+1)$ -th hydrocycle. For the pure Lagrangian mode (`alecoef`  $\equiv \alpha_{ALE} = 1.0$ ) no additional rezoning/remapping is needed.

In the ALE and pure Eulerian modes ( $0.0 \leq \alpha_{ALE} < 1.0$ ), the new Lagrangian mesh  $\vec{x}_i^{L,n+1}$  is reshaped (rezoned) to a new ALE configuration  $\vec{x}_i^{n+1}$  in subroutine REZONE by applying an iterative algorithm, based on the Winslow method — as described in the original CAVEAT report [1]. The resulting mesh displacement at the rezoning step is expressed in terms of the velocity  $\vec{u}_{r,i}$  of mesh motion, defined as

$$\vec{u}_{r,i} = (\vec{x}_i^{n+1} - \vec{x}_i^{L,n+1})/\Delta t. \quad (7.2)$$

Thus defined *rezoning velocities* are used to perform the advection step, where the principal variables  $m_i^{L,n+1}$ ,  $\vec{u}_{c,i}^{L,n+1}$ ,  $e_i^{L,n+1}$  are remapped (advected) from the Lagrangian mesh  $\vec{x}_i^{L,n+1}$  to the new ALE mesh  $\vec{x}_i^{n+1}$ . In a pure Eulerian calculation with  $\alpha_{ALE} = 0.0$  all the new Lagrangian node positions  $\vec{x}_i^{L,n+1}$  are returned to the old values  $\vec{x}_i^n$  by setting the rezoning velocities to  $\vec{u}_{r,i} = -\vec{u}_i$ . The remapping is performed in subroutine ADVECT by using the fluxing volumes, calculated in subroutine ADVFLUX.

In the RALEF code, the remapping can be performed in either one pass (i.e. with one step  $\Delta t$  in time), or in two passes (i.e. with two fractional steps  $0.5\Delta t$  in time), or in four passes (i.e. with four fractional steps  $0.25\Delta t$  in time), depending on whether the corresponding criterion is fulfilled; in the CAVEAT code no more than two passes were used. In each pass the method of directional splitting is applied, i.e. advection is first performed along mesh direction 1, and then along mesh direction 2 (or, alternatively, in the reverse order). The latter means that the advection step is actually split into 2, 4, or 8 substeps.

### 2. General features of the rezoning algorithm

The CAVEAT rezoning algorithm proved to be quite powerful and efficient in generating smooth and well adapted meshes — provided that the boundaries and internal material (i.e. Lagrangian) interfaces remain sufficiently smooth. In practice, however, often zigzag-like features develop along certain Lagrangian interfaces, which cause inversion of mesh directions and its eventual tangling (mesh crash). To cope with such situations, additional smoothing algorithms, described in the subsections below, have been implemented in the RALEF code.

The general scheme of the driver routine REZONE for mesh rezoning is as follows:

- starting from the new Lagrangian mesh  $\vec{x}_i^{L,n+1}$ , an initial state  $\vec{x}_i^{0,n+1}$  for rezoning is prepared in subroutine REZINIT by subtracting the fraction  $(1 - \alpha_{ALE})$  of the coordinate increment on the right-hand side of Eq. (7.1) for all the “regular” vertices not marked with flags "fix", or "lvx", or "ivx"; vertices that must remain fixed in space (i.e. marked with the "fix" flag) are returned to their original positions  $\vec{x}_i^{0,n+1} = \vec{x}_i^n$ ; Lagrangian vertices, marked with the "lvx" flag, retain their new Lagrangian positions  $\vec{x}_i^{0,n+1} = \vec{x}_i^{L,n+1}$ ; for vertices along the Lagrangian interfaces (i.e. marked with the "ivx" flag) the subtraction of the  $(1 - \alpha_{ALE})$  displacement fraction is performed tangentially along the corresponding mesh direction; the nodes on the inflow/outflow boundaries are subject to a special treatment: they are returned to these boundaries on the assumption that the latter are given by surfaces that are fixed in space;
- the cell-centered weight coefficients  $w_i$  are computed in subroutines WEIGHT and WTGLOBL;
- block edges marked by the values  $\text{ibcrezn}(\text{ib}, \text{iblk}) \pmod{10} > 0$  are subject to a special smoothing procedure (as described below) via calls to subroutines REZSMOO and/or REZSTRE; subroutine REZSTRE is a slight modification of the original REZTANG routine from CAVEAT; as a result, the initial state  $\vec{x}_i^{0,n+1}$  is modified to  $\vec{x}_i^{1,n+1}$ ;
- rezoning coefficients  $\text{arzn}(\text{i}, 1:3)$ ,  $\text{brzn}(\text{i}, 1:2)$  are calculated in subroutine REZCOEF for all physical vertices by using the initial mesh configuration  $\vec{x}_i^{1,n+1}$  and the weights  $w_i$ ;
- an intermediate rezoned mesh  $\vec{x}_i^{w,n+1}$  is calculated by performing `itrezn` iterations on the initial state  $\vec{x}_i^{1,n+1}$  in subroutine REZITER; at each iteration the belts of ghost cells with a 100% relative width must be reconstructed; also, a special action is needed at 3-block meeting points to equalize the coordinates of the corresponding three block corners, which is not ensured by the adopted iteration algorithm and is accomplished in subroutine BCRNMRG;
- the final rezoned mesh configuration

$$\vec{x}_i^{n+1} = \vec{x}_i^{L,n+1} + \min \left\{ 1; \frac{a_{mdx} d_i}{|\vec{x}_i^{w,n+1} - \vec{x}_i^{L,n+1}|} \right\} (\vec{x}_i^{w,n+1} - \vec{x}_i^{L,n+1}) \quad (7.3)$$

is obtained by constraining the values  $\vec{x}_i^{w,n+1}$  in subroutine REZLIM in such a way as to ensure that the mesh displacement  $|\vec{x}_i^{n+1} - \vec{x}_i^{L,n+1}|$  is no larger than  $a_{mdx}$  times the distance  $d_i$  to the nearest of the four neighbor vertices. Parameter  $a_{mdx}$  is assigned a fixed value of  $a_{mdx} = 0.7$  in subroutine REZLIM. In order to eliminate possible deviations from spatially fixed boundaries, corresponding boundary conditions on reflecting, center-of-convergence, and inflow/outflow boundaries are imposed on the values of  $\vec{u}_{r,i}$  in subroutine BNDUMRZ.

### 3. Defining purely Lagrangian parts of the mesh

The user is allowed to declare any part `ip` of any block `iblk` to be purely Lagrangian with respect to the mesh rezoning process by setting up the flag

$$\text{iflagprt}(\text{ip}, \text{iblk}) = \text{.true.} \quad (7.4)$$

in the ‘`in2d`’ input file. In this case the “`lvx`” bit (bit # 8) in the global flag array `iflg(I)`, identified with the mask `msklvx`, is set equal to 1 (in subroutine `FLAGSET`, file ‘`f02_init.f`’) for all physical vertices belonging to the part `ip` of block `iblk`. The latter means that during the rezoning procedure all these vertices will be treated as purely Lagrangian, i.e. excluded from the rezoning procedure. To make the rezoned parts of the mesh match smoothly the neighboring Lagrangian part, one should avoid jumps of the weight function at the boundaries of the Lagrangian part.

The default value of the flag `iflagprt` is `iflagprt(ip,iblk) = .false.` for all `ip` and `iblk`.

### 4. Algorithm for smoothing corrugated interfaces

In cases, where tangential rezoning tends to produce unwanted numerical corrugation of Lagrangian interfaces, it is proposed to use the following special smoothing procedure. Let  $\vec{x}_i$  be the original coordinates of vertices along an interface to be smoothed. Then, the new coordinates of the smoothed interface are calculated as

$$\vec{x}_{sm,i} = \vec{x}_i + c_{sm} (\vec{x}_{s,i} - \vec{x}_i), \quad (7.5)$$

where  $\vec{x}_{s,i}$  are the “ideal” (desired) coordinates of vertex  $i$ , and  $c_{sm}$  is a free scaling parameter on the order of unity. The ideal location  $\vec{x}_{s,i}$  of vertex  $i$  is determined by a symmetric local algorithm, based on a 5-point stencil  $\{\vec{x}_{i-2}, \vec{x}_{i-1}, \vec{x}_i, \vec{x}_{i+1}, \vec{x}_{i+2}\}$ .

As a first step, we calculate the turn angle  $\theta_i$  at each vertex  $i$  (see Fig. 7.1) as

$$\theta_i = \text{ATAN2}(\Delta y_i, \Delta x_i), \quad (7.6)$$

where

$$\Delta x_i = (\vec{x}_{i+1} - \vec{x}_i) \cdot (\vec{x}_i - \vec{x}_{i-1}), \quad (7.7)$$

$$\Delta y_i = (\vec{x}_{i+1} - \vec{x}_i) \times (\vec{x}_i - \vec{x}_{i-1}). \quad (7.8)$$

Here  $\theta_i$  is the rotation angle of segment  $\vec{x}_{i+1} - \vec{x}_i$  with respect to the preceding segment  $\vec{x}_i - \vec{x}_{i-1}$  (in the counterclockwise direction) in radians; the FORTRAN intrinsic function `ATAN2(y, x)` yields the principal value of the argument of a complex number  $x + iy$ .

Next, we assume that the ideal location  $\vec{x}_{s,i}$  of vertex  $i$  would correspond to a uniform distribution of the five stencil vertices  $\{\vec{x}_{i-2}, \vec{x}_{i-1}, \vec{x}_i, \vec{x}_{i+1}, \vec{x}_{i+2}\}$  along a common circumscribing circle. In such an ideal case vertex  $i$  would lie on a normal  $\vec{a}_*$  to the segment  $\vec{a} = \vec{x}_{i+1} - \vec{x}_{i-1}$ , passing through its midpoint  $\frac{1}{2}(\vec{x}_{i+1} + \vec{x}_{i-1})$ , at a distance

$$s = \frac{1}{2} |\vec{a}| \cdot \tan \delta \quad (7.9)$$

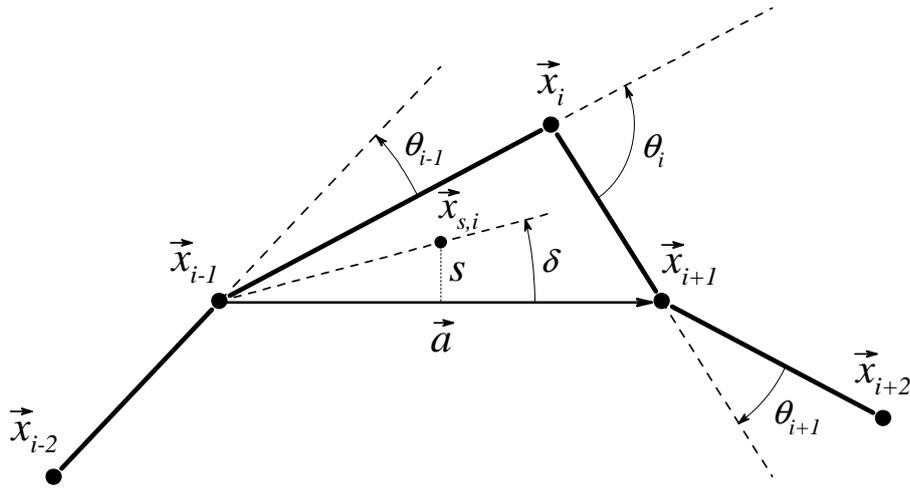


FIG. 7.1: Rezoning scheme for smoothing a corrugated interface.

from this midpoint, i.e. we would have

$$\vec{x}_{s,i} = \frac{1}{2} (\vec{x}_{i-1} + \vec{x}_{i+1}) + \frac{s}{|\vec{a}|} \vec{a}_*, \quad (7.10)$$

where the vectors  $\vec{a}$  and  $\vec{a}_*$  are defined as

$$\vec{a} \equiv \{a_x, a_y\} = \vec{x}_{i+1} - \vec{x}_{i-1}, \quad \vec{a}_* = \{-a_y, a_x\}, \quad (7.11)$$

and the angle  $\delta$  would be given by

$$\delta = \frac{1}{2}\theta_i = \frac{1}{2}\theta_{i-1} = \frac{1}{2}\theta_{i+1}. \quad (7.12)$$

In a real non-ideal case of an arbitrary polygonal line, where the turn angles  $\theta_{i-1} \neq \theta_i \neq \theta_{i+1}$ , we can use the average

$$\delta = \frac{1}{6} (\theta_{i-1} + \theta_i + \theta_{i+1}) \quad (7.13)$$

as the best approximation to the desired value of angle  $\delta$ . As a result, we arrive at the following smoothing correction to the coordinates of vertex  $i$

$$\vec{\tilde{x}}_i = \vec{x}_i + c_{sm} \left[ \frac{1}{2} (\vec{x}_{i-1} + \vec{x}_{i+1}) - \vec{x}_i + \frac{1}{2} \vec{a}_* \tan \delta \right], \quad (7.14)$$

where  $\delta$  is given by Eq. (7.13).

To obtain an a priori estimate for the coefficient  $c_{sm}$ , we may consider a simple particular case of all vertices  $\vec{x}_i$  lying on a straight line but being non-uniformly spaced. In this case  $\delta = 0$  because all  $\theta_i = 0$ , and our smoothing procedure will push the vertices toward a more uniform spacing along the interface. If, say, we consider a distribution that is extremely non-uniform around one point  $x = 0$  with vertices  $\dots, x_{-2} = -2, x_{-1} = -1, x_0 = x_1 = 0, x_2 = 1, x_3 = 2, \dots$ , then after a single smoothing iteration we would like to have the values close to  $x_0 = -1/3, x_1 = +1/3$ . One immediately verifies that the latter is obtained from Eq. (7.14)

with  $c_{sm} = 2/3$ . For practical applications, to avoid overshooting, somewhat smaller values of  $c_{sm} = 0.4-0.5$  can be recommended. Figure 7.2 shows an example of application of the above smoothing algorithm to a strongly distorted boundary, where further simulation without such smoothing would have been impossible.

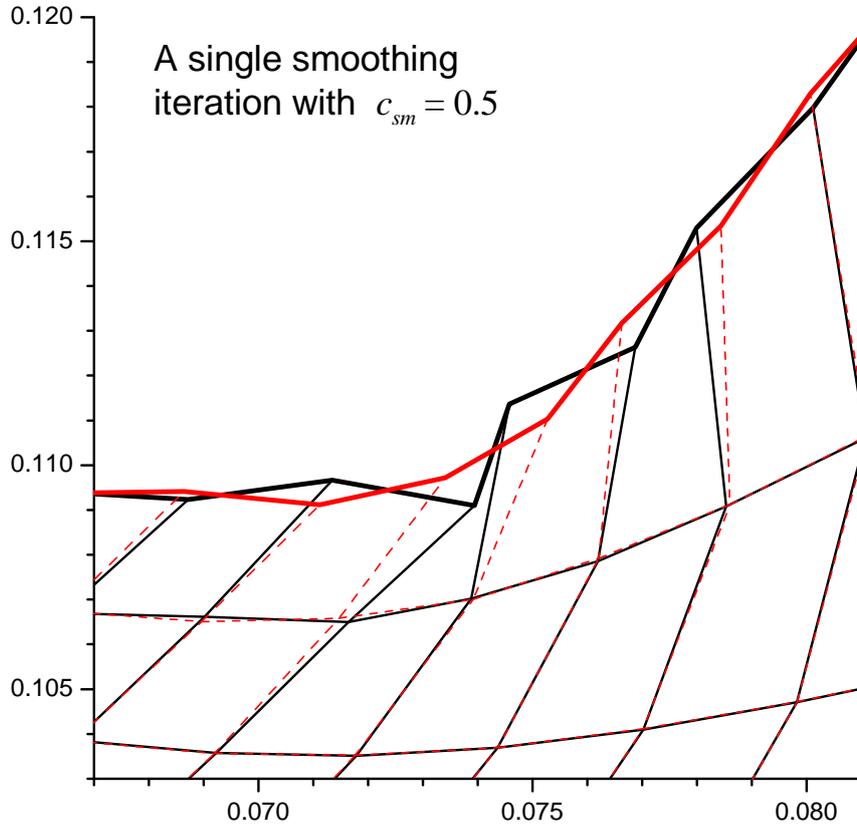


FIG. 7.2: Example of application of a single smoothing iteration with  $c_{sm} = 0.5$  to a corrugated external boundary.

It should be noted that, strictly speaking, the above described rezoning algorithm does not conserve the area of the region, bounded by the rezoned interface. However, due to high order of approximation, the inflicted changes in this area are quite small and usually have no practical significance — which is clearly seen in Fig. 7.2.

Practical tests show that unconditional application of the smoothing algorithm to selected interfaces at every time step may sometimes significantly slow down simulation not because of an extra time required by the smoothing algorithm itself but because too fine and/or skewed mesh is generated at certain locations. To circumvent this undesirable property, additional conditions are introduced into the algorithm that are checked before applying the smoothing procedure to a selected interface. These conditions are formulated as two criteria controlled by the value of parameter  $p_{sm}$ : for  $p_{sm} > 1$  criterion 1 is applied, whereas for  $0 < p_{sm} < 1$  criterion 2 is applied; for  $p_{sm} = 1$  the smoothing is performed unconditionally at every time step.

According to **criterion 1**, the entire interface is subject to a smoothing iteration if for

at least one of its vertices either

$$|\theta_i| > 0.8, \tag{7.15}$$

or

$$3 (\theta_{i-1}^2 + \theta_i^2 + \theta_{i+1}^2 - 3\theta_0^2) > p_{sm} (\theta_{i-1} + \theta_i + \theta_{i+1})^2, \tag{7.16}$$

or

$$\max (|\vec{x}_i - \vec{x}_{i-1}|, |\vec{x}_{i+1} - \vec{x}_i|) > (2p_{sm} - 1) \min (|\vec{x}_i - \vec{x}_{i-1}|, |\vec{x}_{i+1} - \vec{x}_i|). \tag{7.17}$$

According to **criterion 2**, the entire interface is subject to a smoothing iteration if for at least one of its vertices either

$$|\theta_i| > 0.8, \tag{7.18}$$

or

$$\theta_{i-1} \cdot \theta_i < 0 \text{ and } \theta_{i-1} \cdot \theta_{i+1} > 0 \text{ and } \min (|\theta_{i-1}|, |\theta_i|, |\theta_{i+1}|) > p_{sm}. \tag{7.19}$$

Criterion 1 contains an additional free parameter: a sensitivity threshold  $\theta_0$  for the turn angles. The larger the values of  $p_{sm}$  and  $\theta_0$ , the weaker the criterion, and the less frequently is the smoothing procedure applied. High quality smooth interfaces without adverse side effects are obtained for  $p_{sm} = 1.5\text{--}2$  and  $\theta_0 = 0.02\text{--}0.04$ . Criterion 2 is generally less restrictive than criterion 1; good results are usually obtained for  $p_{sm} = 0.01\text{--}0.1$ .

All block corners are excluded from the smoothing procedure. Also excluded (because of the 5-point stencil) are the next neighbors to the block corners unless the second meeting block edge at the corresponding corner is either a reflective or an interblock boundary — for which ghost cells have “real” sizes.

Application of the described smoothing algorithm is realized by calling the subroutine REZSM00 and is controlled by the value of flag `ibcrezn(ib,iblk)`: the criteria for application of the algorithm to an edge `ib` of block `iblk` are tested only when `ibcrezn(ib,iblk) (mod 10) = 1, 2, 4, 5, ..., 9`.

### 5. Algorithm for pocket filling where an interface tends to fold up

Because the above described smoothing algorithm is local (based on a 5-point stencil along one dimension), it cannot prevent formation of deep bottle-shaped pockets, which encompass significantly more than 5 cell faces, and where the interface eventually folds up and mesh discretization of the computational domain becomes multivalued; see Fig. 7.3. To straighten up such a pocket forming at a Lagrangian interface before it closes up in the neck area, the following “pocket-fill” algorithm has been implemented.

First of all, formation of a potentially dangerous pocket is probed by summing up the turn angles  $\theta_{i-j}$  and  $\theta_{i+j}$ ,  $j = 0, 1, 2, \dots, n_{bay} + k_{jt} - 1$ , for  $n_{bay} + k_{jt} - 1$  vertices “down-stream” and “up-stream” from a considered vertex  $i$ , and finding the two maximum values of the accumulated turn angle. In other words, for each vertex  $i$  we calculate the two maximum accumulated turn angles as

$$\Delta\theta_{m,i}^- = \max_k \left\{ - \sum_{j=0}^k \delta_{ed} \theta_{i-j} \right\}, \quad \Delta\theta_{m,i}^+ = \max_k \left\{ - \sum_{j=0}^k \delta_{ed} \theta_{i+j} \right\}, \tag{7.20}$$

where  $k = 0, 1, 2, \dots, n_{bay} + k_{jt} - 1$ , and the factor

$$\delta_{ed} = \text{iend}(\text{ib}) = \begin{cases} +1, & \text{for the left and top block edges (ib = 2, 3),} \\ -1, & \text{for the bottom and right block edges (ib = 1, 4)} \end{cases} \tag{7.21}$$

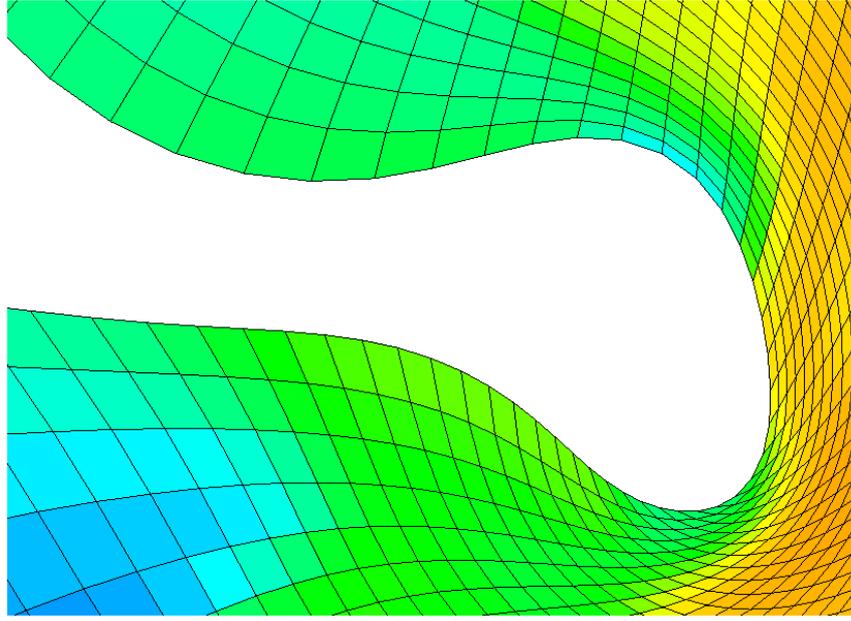


FIG. 7.3: Example of a bottle-shaped vacuum pocket along a corrugated external boundary.

indicates the mesh direction towards the primary corner of block edge  $ib$ . Parameter  $n_{bay}$  is user defined (typical values are  $n_{bay} = 6-12$ ), whereas the “jitter” shift  $k_{jt}$  — which takes on the values  $k_{jt} = \text{ncyc} \pmod{3} = 0, 1, \text{ or } 2$  — is introduced with the purpose to avoid sharp corners at a distance  $\pm n_{bay}$  from the bottom of the filled pocket. If both  $\Delta\theta_{m,i}^-$  and  $\Delta\theta_{m,i}^+$  are positive and large, we have a vacuum pocket with a bottom near vertex  $i$  (if the block edge  $ib$  does border vacuum) — as it is shown in Fig. 7.3.

Once the quantities  $\Delta\theta_{m,i}^\pm$  have been determined, they are compared with the threshold value  $\theta_*$  (a user-defined parameter), and, if the condition

$$\theta_{bay} \equiv \min\{\Delta\theta_{m,i}^-, \Delta\theta_{m,i}^+\} > \theta_* \tag{7.22}$$

is fulfilled, then a “pocket-fill” shift is applied to vertex  $i$ . In practice, good results are usually obtained with  $\theta_* = 0.6-1.2$ .

In the process of the “pocket-fill” shift the old position  $\vec{x}_i$  of vertex  $i$  is shifted to a new one  $\vec{x}_{pf,i}$  along a vector

$$\vec{g} = \delta_{ed} (\vec{n}_a + \vec{n}_b), \tag{7.23}$$

where

$$\vec{a} \equiv \{a_x, a_y\} = \vec{x}_{i+1} - \vec{x}_i, \quad \vec{n}_a = \left\{ \frac{-a_y}{\sqrt{a_x^2 + a_y^2}}, \frac{a_x}{\sqrt{a_x^2 + a_y^2}} \right\}, \tag{7.24}$$

$$\vec{b} \equiv \{b_x, b_y\} = \vec{x}_i - \vec{x}_{i-1}, \quad \vec{n}_b = \left\{ \frac{-b_y}{\sqrt{b_x^2 + b_y^2}}, \frac{b_x}{\sqrt{b_x^2 + b_y^2}} \right\}; \tag{7.25}$$

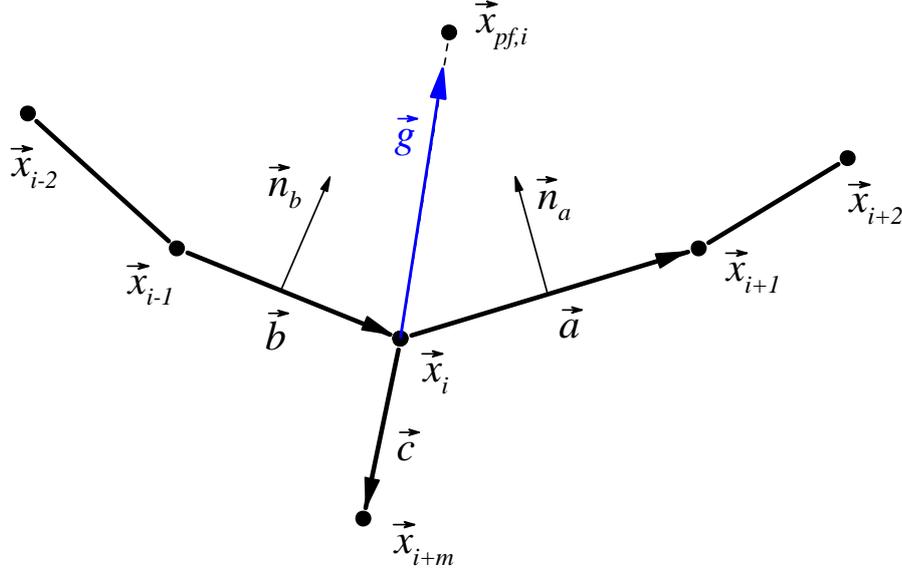


FIG. 7.4: Notation scheme for the “pocket-fill” shift of vertex  $i$ .

see Fig. 7.4. The magnitude of this shift is calculated as

$$\vec{x}_{pf,i} = \vec{x}_i + \alpha_{bay} \min\{|\vec{a}|, |\vec{b}|, |\vec{c}|\} \frac{\vec{g}}{|\vec{g}|}, \quad (7.26)$$

where

$$\alpha_{bay} = c_{bay} h_{bay} h_\theta, \quad (7.27)$$

$$h_{bay} = \min \left\{ 1, \max \left[ 0, \delta_{ed} \frac{\vec{a}_{bay} \times \vec{b}_{bay}}{|\vec{b}_{bay}|^2} \right] \right\}, \quad (7.28)$$

$$h_\theta = \min \left\{ 1, \max \left[ 0, \frac{\theta_{bay} - \theta_*}{\pi/2 - \min(1.4, \theta_*)} \right] \right\}. \quad (7.29)$$

Here

$$\vec{c} = \vec{x}_{i+m} - \vec{x}_i \quad (7.30)$$

is a vector connecting vertex  $i$  to its neighbor  $i + m$  inside the considered block along the mesh direction, other than the mesh direction from  $i - 1$  to  $i$  to  $i + 1$  (see Fig. 7.4);  $c_{bay} \leq 0.8$  is the user defined amplitude of the “pocket-fill” coordinate shift;  $h_{bay}$  is the dimensionless depth of the forming pocket;  $h_\theta$  is a relative measure of how concave the forming pocket is; vector

$$\vec{a}_{bay} = \vec{x}_i - \vec{x}_{im-} \quad (7.31)$$

connects vertex  $im- = i - k_{m-}$ , where the maximum  $\Delta\theta_{m,i}^-$  [see Eq. (7.20)] is reached, with vertex  $i$ ; vector

$$\vec{b}_{bay} = \vec{x}_{im+} - \vec{x}_{im-} \quad (7.32)$$

connects vertex  $im-$  with the vertex  $im+ = i + k_{m+}$ , where the maximum  $\Delta\theta_{m,i}^+$  is reached; see Fig. 7.5.

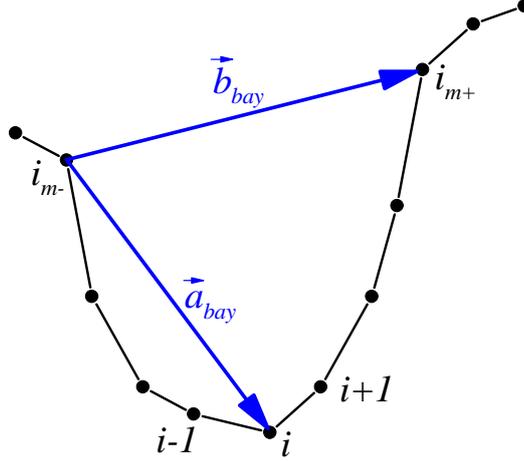


FIG. 7.5: Vectors  $\vec{a}_{bay}$  and  $\vec{b}_{bay}$  used to evaluate the relative depth of a vacuum pocket around vertex  $i$ .

TABLE 7.2: The values of the flag array `ibcrezn(ib,iblk)` used to control application of the smoothing, “pocket-fill”, and stretching algorithms to an edge `ib` of block `iblk`.

<code>ibcrezn(ib,iblk) (mod 10)</code>	smoothing	pocket-fill	stretching
0	no	no	no
1	yes	no	no
2	yes	yes	no
3	no	no	yes
4	yes	no	yes
5	yes	yes	yes
$\geq 6$	yes	no	no

The “pocket-fill” algorithm is applied always in combination with the smoothing algorithm of subsection 7.4 by calling the subroutine `REZSM00`, and it always precedes the smoothing algorithm. The latter means that, if the vertex coordinates  $\vec{x}_i$  along a certain interface were modified to  $\vec{x}_{pf,i}$  by the “pocket-fill” algorithm, it is the modified values  $\vec{x}_{pf,i}$  rather than  $\vec{x}_i$  that serve as an input to Eq. (7.5) in the smoothing algorithm; for this, the values of the turn angles  $\theta_i$  in Eq. (7.6) used in the smoothing algorithm are recalculated for the modified interface configuration  $\vec{x}_{pf,i}$ . To avoid too strong mesh deformations, the “pocket-fill” algorithm is applied no more frequently than every second hydro cycle. Application of the “pocket-fill” and the smoothing algorithms to an edge `ib` of block `iblk` is controlled via user defined values to the flag `ibcrezn(ib,iblk)`, as it is explained in Table 7.2.

## 6. Second-order remapping

In the RALEF code the spatial order of the remapping scheme is controlled by a separate user-defined parameter `iordadv`. If the user specifies `iordadv = 1` (or 2), the first-order (or

second-order) advection scheme is used independent of the value of the control parameter `iorder` for the Riemann solver. If `iordadv` remains unspecified by the user, its value is set equal to that of `iorder` (the original CAVEAT option).

### 1. Gradient limiting

The second-order remapping scheme uses the cell-centered gradients of three volume-specific quantities — namely, of the density  $\rho$ , the volume-specific momentum  $\rho\vec{u}$ , and the volume specific total energy  $\rho e = \rho\epsilon + \frac{1}{2}\rho u^2$  — that are calculated in the subroutine `GRADIENT`. An important role belongs to the method of limiting the calculated gradients, which helps to avoid unphysical values of remapped quantities near sharp fronts.

The method of gradient limiting is controlled by the user-defined parameter `limgrad`. In the RALEF code the following options are available:

- `limgrad = -1` : the unlimited vertex-centered gradients are calculated for all quantities in subroutine `GRADIENT`;
- `limgrad = 0` : the method of monotonic limiting is applied to all fields: the limited cell-centered gradient is the minimum of the 4 gradients at 4 vertices if they all are positive, the maximum of the 4 if they all are negative, and zero otherwise; all the gradients calculated in subroutine `GRADIENT` are cell-centered;
- `limgrad = +1` : the *regular van Leer* limiting method is applied to the scalar fields (density, pressure, ...), while the gradients of the vector fields (velocity, momentum, ...) are left unlimited; all the gradients calculated in subroutine `GRADIENT` are cell-centered;
- `limgrad = +2` : the *regular van Leer* limiting method is applied to all fields; all the calculated gradients are cell-centered;
- `limgrad = +3` : the *extended van Leer* limiting method is applied to all fields; all the calculated gradients are cell-centered.

The original CAVEAT code had only the monotonic and the regular van Leer gradient limiting, employed with the options `limgrad = 1` and `2`. The extended van Leer limiting with the option `limgrad = 3` is a new development in the RALEF code implemented in November 2011.

When a cell-centered gradient  $\vec{g}_i$  of a quantity  $q$  is calculated (see [1, §3.1.2]), the latter means that the spatial variation of  $q$  across cell  $i$  is approximated by a linear function

$$\tilde{q}(\vec{x}) = q_i + \vec{g}_i \cdot (\vec{x} - \vec{x}_{c,i}), \quad (7.33)$$

where  $q_i$  is the value of  $q$  at the center of cell  $i$  with coordinates  $\vec{x}_{c,i}$ . When the van Leer method of gradient limiting is applied, the gradient  $\vec{g}_i$  in Eq. (7.33) is replaced by the product  $\alpha_i \vec{g}_i$ , where the limiting factor  $0 \leq \alpha_i \leq 1$  is given by

$$\alpha_i = \min\{1, \alpha_{max}, \alpha_{min}\}, \quad (7.34)$$

with

$$\alpha_{max} = \max\left\{0, \frac{q_{8n,max} - q_i}{\tilde{q}_{max} - q_i}\right\}, \quad \alpha_{min} = \max\left\{0, \frac{q_{8n,min} - q_i}{\tilde{q}_{min} - q_i}\right\}. \quad (7.35)$$

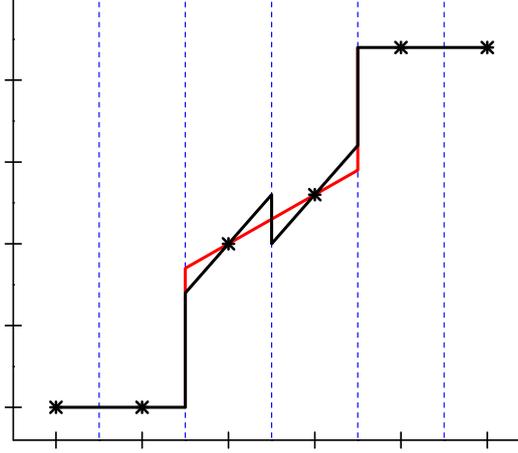


FIG. 7.6: The regular van Leer (black) and the extended van Leer (red) methods of gradient limiting.

In the above equations

$$q_{8n,max} = \max_{l=1,2,\dots,8} \{q_{i+k_{8,l}}\}, \quad q_{8n,min} = \min_{l=1,2,\dots,8} \{q_{i+k_{8,l}}\}, \quad (7.36)$$

are, respectively, the maximum and the minimum values of  $q_i$  from the 8 neighboring cells surrounding the given cell  $i$ , which participate in calculating the gradient  $\vec{g}_i$ ;  $k_{8,l}$  are the local index offsets with respect to  $i$  for these 8 neighbors. In the *regular van Leer* limiting algorithm (`limgrad=1,2`) the maximum,  $\tilde{q}_{max}$ , and the minimum,  $\tilde{q}_{min}$ , values of the linear function (7.33) (with unlimited gradients  $\vec{g}_i$ ) are calculated by only considering its variation *inside* the cell  $i$ , i.e. among the four corner values

$$\tilde{q}_{max} = \max_{l=1,2,3,4} \{\tilde{q}(\vec{x}_{i+k_{4,l}})\}, \quad \tilde{q}_{min} = \min_{l=1,2,3,4} \{\tilde{q}(\vec{x}_{i+k_{4,l}})\}, \quad (7.37)$$

where  $k_{4,l}$  are the local index offsets with respect to  $i$  for the 4 corner vertices of cell  $i$ , and  $\vec{x}_i$  are the coordinates of cell vertex  $i$ .

As mentioned in the original CAVEAT report [1], the regular van Leer limiting sometimes produces locally non-monotonic sawtooth-like variation of the approximated quantity  $q(\vec{x})$  at cell interfaces, as is illustrated with the black polygonal line in Fig. 7.6. Quite often this violation of local monotonicity becomes prohibitive for using the second-order in the advection scheme. As a simple and effective remedy, the following modification of the regular van Leer limiting has been introduced in the RALEF code: when calculating the quantities  $\tilde{q}_{max}$  and  $\tilde{q}_{min}$ , the domain of variation of the linear interpolation (7.33) must be extended to include the centers of the 8 surrounding neighbor cells, i.e. to modify Eq. (7.37) to

$$\begin{aligned} \tilde{q}_{max} &= \max_{l=1,2,3,4; l'=1,2,\dots,8} \{\tilde{q}(\vec{x}_{c,i+k_{4,l}}), \tilde{q}(\vec{x}_{i+k_{8,l'}})\}, \\ \tilde{q}_{min} &= \min_{l=1,2,3,4; l'=1,2,\dots,8} \{\tilde{q}(\vec{x}_{i+k_{4,l}}), \tilde{q}(\vec{x}_{c,i+k_{8,l'}})\}. \end{aligned} \quad (7.38)$$

As a result, we obtain a more restrictive *extended van Leer* method of gradient limiting, illustrated in Fig. 7.6 with the red polygonal line.

It is easy to see that for 1D interpolation the extended van Leer method is in fact equivalent to the monotonic limiting (`limgrad=0`) already implemented in the original CAVEAT code — which, however, is not the case in two and three dimensions! Practical tests have demonstrated that in many cases, where both the regular van Leer and the monotonic gradient limiting algorithms fail (resulting, typically, in negative temperatures and/or unphysical reduction of the hydrodynamic time step by many orders of magnitude), the extended van Leer limiting facilitates a smooth simulation with the second-order advection scheme. Therefore, the default value of the `limgrad` parameter is set equal to `limgrad=3`.

2. *Limits on the 2nd-order advection terms*

The advection scheme inherited from the CAVEAT code is conservative with respect to the three principal advected variables: mass, momentum and total (kinetic + internal) energy. The internal energy is then calculated as the difference between the total and the kinetic energy components. Clearly, in certain cases this may lead to spurious unphysical drops in the value of the internal energy, accompanied by unphysical reduction of temperature (often below zero) and practical stoppage of the simulation. Because the total energy is conserved, we can call such behavior a *negative advective dissipation*, or consider it as a manifestation of a *negative numerical viscosity*.

The key quantity for computing the amount of advection in cell  $i$  for any of the principal variables at every advection substep is the *fluxing volume*  $V_{im}^a$ , associated with face  $im$ . The sign of  $V_{im}^a$  is defined such that the change in volume of cell  $i$  is given by

$$V_i = V_i^L + V_{im}^a. \tag{7.39}$$

Hence, if at a given advection substep face  $i1$  ( $i2$ ) is displaced as a whole in the positive mesh direction 2 (1) along the normal  $\vec{n}_{f,i1}$  ( $\vec{n}_{f,i2}$ ), we have  $V_{im}^a < 0$  because such a displacement results in the reduction of the volume  $V_i$  of cell  $i$ ; see Fig. ??.

First of all, consider the core part of all the advection terms, which is obtained in the 1-st order with the full donor treatment. The extent of the donor-cell treatment is controlled by the user defined parameter  $0 \leq f_d \leq 1$ :  $f_d$  is the weight coefficient for the donor-cell value of the advected quantity,  $1 - f_d$  is the weight coefficient for the acceptor-cell value of the advected quantity. In the case of the full donor treatment we have  $f_d = 1.0$ , which means a 100% upstream advection.

Here we prove the following statement:

*the artificial advective dissipation of the kinetic energy is always non-negative in the 1-st order with the full (100% upstream) donor treatment.*

Indeed, it is quite clear that for  $V_{im}^a < 0$  we have zero advective dissipation in cell  $i$  because the donor cell is the cell  $i$  itself, and the reductions in mass, momentum, total, kinetic and internal energies of cell  $i$  are all strictly proportional to the fractional reduction of its volume (i.e. are all altered in one and the same proportion).

Now consider a less evident case of  $V_{im}^a > 0$ . Here the cell  $i$  receives a certain amount of mass

$$\Delta m = \rho_{i_d} V_{im}^a > 0 \tag{7.40}$$

from the neighbor (across face  $im$ ) donor cell  $i_d = i_o$  (an equal amount of mass is, of course, lost by the donor cell). With this mass the acceptor cell  $i$  receives also the amount of

momentum

$$\Delta \vec{P} = \Delta m \vec{u}_{i_d}, \quad (7.41)$$

and the amount of total energy

$$\Delta E = \Delta m e_{i_d} = \Delta m \left( \epsilon_{i_d} + \frac{1}{2} u_{i_d}^2 \right), \quad (7.42)$$

where  $\vec{u}_{i_d}$  is the cell-center fluid velocity in the donor cell  $i_d$ ,  $e_{i_d}$  is its mass-specific total energy, and  $\epsilon_{i_d}$  is its mass-specific internal energy.

Because the internal energy  $\mathcal{E}_i$  of any cell  $i$  is calculated as

$$\mathcal{E}_i = E_i - \frac{P_i^2}{2m_i}, \quad (7.43)$$

where  $E_i$  is its total energy, and  $P_i$  is its total momentum, the advective increment of the internal energy in cell  $i$  is given by

$$\Delta \mathcal{E} = \Delta E + \frac{P_i^2}{2m_i} - \frac{(\vec{P}_i + \Delta \vec{P})^2}{2(m_i + \Delta m)}. \quad (7.44)$$

To find the amount of artificial advective dissipation, we have to compare (7.44) with the direct advective increment of the internal energy

$$\Delta \mathcal{E}_0 = \Delta m \epsilon_{i_d}, \quad (7.45)$$

which cell  $i$  would have received with mass  $\Delta m$  in the absence of any dissipation. As a result, the amount of artificial kinetic energy dissipation due to the first-order upstream advection across face  $im$  is found to be

$$d\mathcal{E}_a \equiv \Delta \mathcal{E} - \Delta \mathcal{E}_0 = \frac{m_i \Delta m}{2(m_i + \Delta m)} (\vec{u}_{i_d} - \vec{u}_i)^2 \geq 0. \quad (7.46)$$

Thus, negative advective dissipation may arise only due to a partial donor treatment ( $f_d < 1.0$ ) and/or second-order terms. In this general case the advection terms  $\Delta m$ ,  $\Delta \vec{P}$ , and  $\Delta E$ , associated with face  $im$  and added to cell  $i$ , are calculated as follows. Let  $i_o$  be the neighbor cell of cell  $i$ , which lies across the face  $im$ . Then the donor cell  $i_d$  and the acceptor cell  $i_a$  are determined as

$$i_d = \begin{cases} i, & V_{im}^a < 0, \\ i_o, & V_{im}^a \geq 0, \end{cases} \quad i_a = \begin{cases} i, & V_{im}^a \geq 0, \\ i_o, & V_{im}^a < 0. \end{cases} \quad (7.47)$$

The principal advection terms are given by the sum of two components

$$\Delta m = \Delta m^{(1)} + \lambda \Delta m^{(2)}, \quad (7.48a)$$

$$\Delta \vec{P} = \Delta \vec{P}^{(1)} + \lambda \Delta \vec{P}^{(2)}, \quad (7.48b)$$

$$\Delta E = \Delta E^{(1)} + \lambda \Delta E^{(2)}, \quad (7.48c)$$

where

$$\Delta m^{(1)} = \rho_{i_d} V_{im}^a, \quad (7.49a)$$

$$\Delta \vec{P}^{(1)} = \rho_{i_d} \vec{u}_{i_d} V_{im}^a = \vec{u}_{i_d} \Delta m^{(1)}, \quad (7.49b)$$

$$\Delta E^{(1)} = \rho_{i_d} e_{i_d} V_{im}^a = e_{i_d} \Delta m^{(1)}, \quad (7.49c)$$

are the first-order full-donor contributions, whose numerical dissipation is always positive; also, for a non-inverted mesh,  $\Delta m^{(1)}$  cannot lead to a negative mass of either donor or acceptor cell.

The second components in Eq. (7.48),

$$\Delta m^{(2)} = [\Delta \rho^{(12)} + \Delta \rho^{(2)}] V_{im}^a, \quad (7.50a)$$

$$\Delta \vec{P}^{(2)} = [\Delta(\rho \vec{u})^{(12)} + \Delta(\rho \vec{u})^{(2)}] V_{im}^a, \quad (7.50b)$$

$$\Delta E^{(2)} = [\Delta(\rho e)^{(12)} + \Delta(\rho e)^{(2)}] V_{im}^a, \quad (7.50c)$$

represent the sum of the partial-donor [upper index (12)] and the second-order [upper index (2)] contributions;  $0 \leq \lambda \leq 1$  is the limiting factor associated with the given face  $im$ , which has been introduced in the RALEF code (November 2011) to suppress possible negative numerical viscosity and too high mass advection. The contributions from the partial-donor treatment are defined by

$$\Delta \rho^{(12)} = \frac{1}{2} \text{sign}(V_{im}^a) (1 - f_d) (\rho_i - \rho_{i_o}), \quad (7.51a)$$

$$\Delta(\rho \vec{u})^{(12)} = \frac{1}{2} \text{sign}(V_{im}^a) (1 - f_d) (\rho_i \vec{u}_i - \rho_{i_o} \vec{u}_{i_o}), \quad (7.51b)$$

$$\Delta(\rho e)^{(12)} = \frac{1}{2} \text{sign}(V_{im}^a) (1 - f_d) (\rho_i e_i - \rho_{i_o} e_{i_o}). \quad (7.51c)$$

The second-order contribution to  $\Delta m^{(2)}$  in Eq. (7.50a) is defined by

$$\Delta \rho^{(2)} = \frac{1}{2} [\delta \rho_i^{(2)} + \delta \rho_{i_o}^{(2)}] - \frac{f_d}{2} \text{sign}(V_{im}^a) [\delta \rho_i^{(2)} - \delta \rho_{i_o}^{(2)}], \quad (7.52)$$

where

$$\delta \rho_i^{(2)} = \alpha_i \vec{g}_{\rho_i} \cdot (\vec{x}_{im} - \vec{x}_{c,i}), \quad (7.53a)$$

$$\delta \rho_{i_o}^{(2)} = \alpha_{i_o} \vec{g}_{\rho_{i_o}} \cdot (\vec{x}_{im} - \vec{x}_{c,i_o}), \quad (7.53b)$$

$\vec{g}_{\rho_i}$  and  $\vec{g}_{\rho_{i_o}}$  are, respectively, the cell-centered gradients of density  $\rho$  in cells  $i$  and  $i_o$ ,  $\vec{x}_{im}$  is the center position of face  $im$ ,  $\vec{x}_{c,i}$  and  $\vec{x}_{c,i_o}$  are the center positions of cells  $i$  and  $i_o$ . For the fractional interpolation lengths  $\alpha_i$  and  $\alpha_{i_o}$  the following expressions are used in the RALEF code

$$\alpha_i = \max \left\{ 0, \min \left\{ 1, 1 + \frac{V_{im}^a}{V_i} \right\} \right\}, \quad (7.54a)$$

$$\alpha_{i_o} = \max \left\{ 0, \min \left\{ 1, 1 - \frac{V_{im}^a}{V_{i_o}} \right\} \right\}, \quad (7.54b)$$

which differ from the original CAVEAT formulae

$$\alpha_{CAV,i} = 1 - \left[ \frac{1}{2} - \frac{f_d}{2} \text{sign}(V_{im}^a) \right] \frac{|V_{im}^a|}{V_i}, \quad (7.55a)$$

$$\alpha_{CAV,i_o} = 1 - \left[ \frac{1}{2} + \frac{f_d}{2} \text{sign}(V_{im}^a) \right] \frac{|V_{im}^a|}{V_{i_o}}. \quad (7.55b)$$

Since for  $f_d = 1$  Eqs. (7.54) and (7.55) yield the same result, the formal difference between them can hardly be of any practical importance because it does not make much sense to combine  $f_d < 1$  with the second-order scheme. Nevertheless, for  $f_d = 0$  Eq. (7.54) gives more bias to the donor cell than (7.55), which appears physically more reasonable. Note also that Eq. (7.54) does not allow non-physical values of the interpolated quantity even for a weaker regular van Leer gradient limiting with `limgrad` = 1 and 2 by constraining the interpolation region within the corresponding cell. Expressions for  $\Delta(\rho\vec{u})^{(2)}$  and  $\Delta(\rho e)^{(2)}$  are fully analogous to Eqs. (7.52) and (7.53) with the only difference that the gradients of the corresponding volume-specific quantity (either  $\rho\vec{u}$  or  $\rho e$ ) are used.

The limiting factor  $\lambda$  in Eqs. (7.48) is calculated in two steps, and, respectively, its value is controlled by two user defined parameters  $\varepsilon_{a2m} \geq 0$  and  $\varepsilon_{a2e} \geq 0$ . At the first step the limit  $\lambda_m$  is calculated by constraining mass depletion in the donor cell due to the  $\Delta m^{(2)}$  term. More specifically, the value of  $\lambda_m$  is chosen such as to ensure that the  $\lambda_m \Delta m^{(2)}$  term carries out no more mass from the donor cell than a fraction  $\varepsilon_{a2m}$  of what is left after the term  $|\Delta m^{(1)}|$  is subtracted, i.e. no more mass than  $\varepsilon_{a2m} (V_{i_d} - |V_{im}^a|)$ , which yields

$$\lambda_m = \begin{cases} \max \left\{ 0, \min \left\{ 1, \varepsilon_{a2m} \frac{\rho_{i_d}}{\Delta\rho} \left( \frac{V_{i_d}}{|V_{im}^a|} - 1 \right) \right\} \right\}, & \Delta\rho > 0 \text{ and } 0 \leq \varepsilon_{a2m} < 1, \\ 1, & \Delta\rho \leq 0 \text{ or } \varepsilon_{a2m} \geq 1, \end{cases} \quad (7.56)$$

where

$$\Delta\rho = \Delta\rho^{(12)} + \Delta\rho^{(2)}. \quad (7.57)$$

Note that the user can easily turn off the limiting due to mass depletion by choosing  $\varepsilon_{a2m} \geq 1$ . Typically, values  $\varepsilon_{a2m} = 0.2\text{--}0.5$  would be used.

At the second step the limit  $\lambda_e$  constraining the amount of negative advective dissipation  $d\tilde{\varepsilon}_a$  of the kinetic energy is evaluated as

$$\lambda_e = \begin{cases} -\varepsilon_{a2e} \frac{\max\{\frac{1}{2}\tilde{u}_i^2, \frac{1}{2}\tilde{u}_{i_o}^2\}}{d\tilde{\varepsilon}_a}, & d\tilde{\varepsilon}_a < 0 \text{ and } 0 \leq \varepsilon_{a2e} < 1, \\ 1, & d\tilde{\varepsilon}_a \geq 0 \text{ or } \varepsilon_{a2e} \geq 1, \end{cases} \quad (7.58)$$

where

$$d\tilde{\varepsilon}_a = \max \{ \tilde{\varepsilon}_i - \tilde{\varepsilon}_{0,i}, \tilde{\varepsilon}_{i_o} - \tilde{\varepsilon}_{0,i_o} \}. \quad (7.59)$$

Again, the user can easily turn off this limiting step by choosing  $\varepsilon_{a2e} \geq 1$ . The intermediate values of the mass-specific kinetic energy  $\frac{1}{2}\tilde{u}^2$  and internal energy  $\tilde{\varepsilon}$  in cells  $i$  and  $i_o$  are calculated by using the intermediate advection terms (7.48) with  $\lambda = \lambda_m$ , namely,

$$\tilde{u}_i^2 = \left( \frac{\vec{P}_i + \Delta\vec{P}}{m_i + \Delta m} \right)^2, \quad \tilde{u}_{i_o}^2 = \left( \frac{\vec{P}_{i_o} - \Delta\vec{P}}{m_{i_o} - \Delta m} \right)^2, \quad (7.60)$$

$$\tilde{\varepsilon}_i = \frac{E_i + \Delta E}{m_i + \Delta m} - \frac{1}{2}\tilde{u}_i^2, \quad \tilde{\varepsilon}_{i_o} = \frac{E_{i_o} - \Delta E}{m_{i_o} - \Delta m} - \frac{1}{2}\tilde{u}_{i_o}^2. \quad (7.61)$$

The reference dissipationless values  $\tilde{\epsilon}_0$  of the mass-specific internal energy in cells  $i$  and  $i_o$  are given by

$$\tilde{\epsilon}_{0,i_d} = \frac{E_{i_d}}{m_{i_d}} - \frac{1}{2} \left( \frac{\vec{P}_{i_d}}{m_{i_d}} \right)^2, \quad (7.62)$$

$$\tilde{\epsilon}_{0,i_a} = \frac{E_{i_a} - \vec{P}_{i_a}^2/2m_{i_a} + |\Delta m| \tilde{\epsilon}_{0,i_d}}{m_{i_a} + |\Delta m|}, \quad (7.63)$$

where again  $\Delta m$  is calculated from Eq. (7.48a) with  $\lambda = \lambda_m$ . In the end, the minimum

$$\lambda = \min\{\lambda_m, \lambda_e\} \quad (7.64)$$

of the two  $\lambda$ -values is used to calculate the final advection terms (7.48).

*Practical recommendations:* It is recommended to always start with the default values  $f_d \equiv \text{fdonor} = 1.0$ ,  $\text{limgrad} = 3$ ,  $\epsilon_{a2m} = 0.5$ , and  $\epsilon_{a2e} = 2.0$  when the second-order advection scheme is used. If this option fails with the message of a negative cell mass after advection and the mesh is not too strongly distorted, one can try a smaller value of  $\epsilon_{a2m} \approx 0.1$ . If the default option fails because of persistent negative advective dissipation, which sometimes develops into a sort of numerical instability and leads to negative temperatures, one should set a sufficiently small value of  $\epsilon_{a2e} \ll 1$ ; numerical tests have demonstrated that  $\epsilon_{a2e} = 10^{-3}$  usually fixes the problem. As a final resort, one can set  $\epsilon_{a2e} = 0$ .

---

*Correspondence with the code variables:*

$\alpha_{ALE}$	= <code>alecoef</code>	ALE coefficient in range $0 \leq \alpha_{ALE} \leq 1$ ;
$\Delta t$	= <code>dthydro</code>	time increment in the current hydrocycle;
$0 \leq f_d \leq 1$	= <code>fdonor</code>	user-defined parameter which defines the amount of donor-cell treatment by advection;
$\epsilon_{a2m} \geq 0$	= <code>adv2lim</code>	user-defined scaling factor for limiting the second-order advection to prevent excessive mass depletion in the donor cell;
$\epsilon_{a2e} \geq 0$	= <code>epsie2adv</code>	user-defined scaling factor for limiting the second-order advection to suppress negative numerical dissipation of the kinetic energy;
$\vec{u}_i$	= <code>um(i,m)</code> , $m=1,2$	mesh node velocity;
$\vec{x}_i^{L,n+1}$	= <code>xv(i,m)</code>	new Lagrangian coordinates of mesh vertices at $(n+1)$ -th hydro cycle;
$m_i^{L,n+1} = m_i^n$	= <code>cm(i)</code>	mesh-cell masses after the Lagrangian phase of the $(n+1)$ -th hydro cycle;
$\vec{u}_{c,i}^{L,n+1}$	= <code>ucl(i)</code>	cell-centered fluid velocities after the Lagrangian phase of the $(n+1)$ -th hydro cycle;
$e_i^{L,n+1}$	= <code>tel(i)</code>	mass-specific total fluid energies after the Lagrangian phase of the $(n+1)$ -th hydro cycle;
$w_i$	= <code>wt(i)</code>	weight coefficients in the Winslow rezoning algorithm; cell-centered, $I$ -numbered;
$\vec{x}_i^{0,n+1}$	= <code>xn(i,m)</code>	initial configuration of rezoned mesh at $(n+1)$ -th hydro cycle after the <code>REZINIT</code> subroutine;
$\vec{x}_i^{1,n+1}$	= <code>xn(i,m)</code>	initial configuration of rezoned mesh after special smoothing of selected block boundaries in subroutines <code>REZSMO</code> and <code>REZSTRE</code> ;

$\vec{x}_i^{w,n+1}$	= <code>xn(i,m)</code>	almost final configuration of rezoned mesh after <code>itrezn</code> rezoning iterations with the <code>REZITER</code> subroutine;
$\vec{u}_{r,i}$	= <code>umr(i,m)=um(i,m)</code>	mesh rezoning velocities; vertex-centered, $I$ -numbered;
$a_{mdx}$	= <code>amaxdx</code>	parameter defined in subroutine <code>REZLIM</code> , which controls the relative coordinate shift from the Lagrangian to rezoned mesh in the boundary smoothing algorithm; default value is 0.7.
$c_{sm}$	= <code>crezsm</code>	coefficient controlling the amplitude of coordinate shift in the boundary smoothing algorithm;
$p_{sm}$	= <code>prezsm</code>	parameter controlling application of the boundary smoothing algorithm;
$\theta_0$	= <code>arezsm</code>	sensitivity threshold for turn angles in criterion 1 for application of the boundary smoothing algorithm;
$n_{bay}$	= <code>nrezbay</code>	vertex-number span when probing block edges for the presence of a “vacuum” pocket;
$\theta_*$	= <code>arezbay</code>	threshold value for $\Delta\theta_{m,i}^\pm$ to apply the pocket-fill algorithm;
$c_{bay}$	= <code>crezbay</code>	amplitude of the “pocket-fill” coordinate shift;

---

## 8. PARALLELIZATION SCHEME

In the RALEF-2D code, only the parts of the algorithm dealing with the radiation transport and laser deposition have a parallel version. A two-level parallelization scheme, based on the MPI and OpenMP standards, has been implemented. Parallel calculations can be performed for different radiation frequencies  $\nu$  and different photon propagation directions  $\vec{\Omega}$ . Because not all of the total  $4N_\Omega$  processed angular directions are independent of one another, we introduce a minimum unit (collection) of directions, which can be processed independently, and denote it as an  $\Omega$ -*bundle*. Thus, the minimum amount of work that a single OpenMP thread of a single MPI task can process is to calculate the contribution of a single frequency group  $[k]$  and a single  $\Omega$ -bundle to the radiative heating rate  $Q_r$ .

- Any  $\Omega$ -bundle can consist of up to two independent  $\Omega$ -groups.

## 9. NUMERICAL DIAGNOSTICS

### 1. Angular distribution of emission in selected spectral band

In certain cases it is important to know the angular distribution of thermal x-ray emission from the entire simulated plasma. The corresponding numerical diagnostics is activated by setting the parameter `nnusetet`  $\geq 1$ , as is described in the user-guide report on RALEF. Within this diagnostics the angular distribution of the target emission in the user-defined selected spectral band

$$[k] \in [\Delta k], \text{ i.e. for } k1nuse1 \leq [k] \leq k2nuse1 \tag{9.1}$$

is calculated and written out into file ‘`enusoutet.dat`’ at user-defined times `twfilm(i)`. More specifically, calculated are (i) the specific (per steradian of the solid angle) emission power

$$J_{[\Delta k]}(\theta) = \frac{dW_{[\Delta k]}}{d\Omega} \begin{cases} [\text{erg cm}^{-1} \text{ s}^{-1} \text{ ster}^{-1}], & \text{iradial} = 0, \\ [\text{erg s}^{-1} \text{ ster}^{-1}], & \text{iradial} = 1, 2, \end{cases} \tag{9.2}$$

of the whole target, and the (ii) the specific (per steradian of the solid angle) emitted energy

$$\frac{dE_{[\Delta k]}}{d\Omega} = \int_0^t J_{[\Delta k]}(\theta) dt \quad \begin{cases} [\text{erg cm}^{-1} \text{ ster}^{-1}], & \text{iradial} = 0, \\ [\text{erg ster}^{-1}], & \text{iradial} = 1, 2, \end{cases} \quad (9.3)$$

in the selected spectral band  $[k] \in [\Delta k]$  as functions of a single angle  $\theta$ . In the  $xy$  geometry ( $\text{iradial} = 0$ )  $\theta$  is the azimuthal angle measured in the equatorial plane, perpendicular to the  $Z$ -axis; in the  $rz$  geometry ( $\text{iradial} = 1, 2$ )  $\theta$  is the polar angle measured from the reference axis, which is assumed to be colinear with the rotation  $Z$ -axis. In the  $xy$  case the reference axis is assumed to be colinear with either the  $X$ -axis or  $Y$ -axis.

Because the volumetric heating of every individual mesh cell is calculated by combining two opposite directions from the  $S_n$  set, the values of  $J_{[\Delta k]}(\theta)$  can only be calculated by integrating the radiation intensity  $I_{[k],L}$  along the outer physical boundary (boundary contours **bc1+bc2**). Then, once we choose to use the known intensities  $I_{[k],L}$ , the discrete  $\theta$ -mesh for the angular dependence of  $J_{[\Delta k]}(\theta)$  is dictated by the currently used  $S_n$  (or  $ES_n$ ) quadrature with  $k_\Omega(k_\Omega + 1)/2$  nodes per octant, where  $n = 2k_\Omega$ . As a consequence, in the  $xy$  geometry only a subset of  $k_\Omega$  nodes (per octant) from the nearest to the equator  $z$ -tier is used.

As a starting point for the computation formulae, we recall the flux-based formula

$$W_{[\Delta k],fl} = \int_{S_{bc}} dS \int_{4\pi} (\vec{\Omega} \cdot \vec{n}) d\vec{\Omega} \int I_\nu d\nu = \int_{S_{bc}} dS \left[ 2 \sum_{L=1}^{4N_\Omega} w_L (\vec{\Omega}_L \cdot \vec{n}) \sum_{[k] \in [\Delta k]} I_{[k],L} \right] \quad (9.4)$$

for the total (over the entire solid angle) emission power into the selected spectral band. Here

$$dS = \begin{cases} d\lambda, & \text{iradial} = 0, \\ R d\lambda, & \text{iradial} = 1, 2, \end{cases} \quad (9.5)$$

is the area element along the physical outer boundary surface  $S_{bc}$ ,  $\vec{n}$  is the outward unit normal to this boundary surface.

Then in the  $xy$  geometry, where every discrete interval  $[\theta_j, \theta_{j+1}]$  is associated with a single  $S_n$  direction  $L_j$ , we have

$$J_{[\Delta k],j} = \int_{S_{bc}} d\lambda (\vec{\Omega}_{L_j} \cdot \vec{n}) \sum_{[k] \in [\Delta k]} I_{[k],L_j}, \quad \text{for iradial} = 0. \quad (9.6)$$

In the  $rz$  geometry, however, every discrete interval  $[\theta_j, \theta_{j+1}]$  is associated with a single  $Z$ -tier of  $ES_n$  beamlets — i.e. with a single pair of “quantum numbers”  $(i_{oZ}, m_Z)_j$  — which combines a subset of  $2(k_\Omega + 1 - m_Z)$   $ES_n$ -directions. In this case

$$\begin{aligned} J_{[\Delta k],j} &= 2\pi \left\{ \int_{S_{bc}} dS \left[ 2 \sum_{L \in (i_{oZ}, m_Z)_j} w_L (\vec{\Omega}_L \cdot \vec{n}) \sum_{[k] \in [\Delta k]} I_{[k],L} \right] \right\} \left( 2 \sum_{L \in (i_{oZ}, m_Z)_j} w_L \right)^{-1} \\ &= \frac{\pi}{k_\Omega + 1 - m_{Z,j}} \int_{S_{bc}} R d\lambda \sum_{L \in (i_{oZ}, m_Z)_j} (\vec{\Omega}_L \cdot \vec{n}) \sum_{[k] \in [\Delta k]} I_{[k],L}, \quad \text{iradial} = 1, 2. \end{aligned} \quad (9.7)$$

Here we used the fact that in the  $ES_n$  quadrature (always applied in the  $rz$  case) all the weights  $w_l$  are equal between themselves,

$$w_l = w_0 = \frac{\pi}{k_\Omega(k_\Omega + 1)}. \quad (9.8)$$

When the simulated configuration has one (or two) global reflective boundaries (i.e. when `ifrlty1`  $\geq 1$ ), one has for every angular bin  $j$  combine two (or four) mutually conjugate beamlets. For the  $rz$  geometry this means that, when `ifrlty1` = 1 or 2, one has for every bin  $j$  sum up the values of  $J_{[\Delta k],j}$  for  $i_{oZ} = +1$  and  $i_{oZ} = -1$ .

Note that, to provide an extra control, in addition to  $J_{[\Delta k],j}$  and  $dE_{[\Delta k]}/d\Omega$  the “check sums”

$$W_{[\Delta k],chk} = \begin{cases} 2w_0 \sum_{j=1}^{4k_\Omega} J_{[\Delta k],j} = \frac{2\pi}{k_\Omega(k_\Omega + 1)} \sum_{j=1}^{4k_\Omega} J_{[\Delta k],j}, & \text{iradial} = 0, \\ \frac{2w_0}{2\pi} \sum_{j=1}^{2k_\Omega} 2(k_\Omega + 1 - m_{Z,j}) J_{[\Delta k],j} = \sum_{j=1}^{2k_\Omega} \frac{2(k_\Omega + 1 - m_{Z,j})}{k_\Omega(k_\Omega + 1)} J_{[\Delta k],j}, & \text{iradial} = 1, 2, \end{cases} \quad (9.9)$$

are also printed into file ‘`enusoutet.dat`’ together with the corresponding values of  $E_{[\Delta k],chk}$ . In the  $rz$  case the “check sums”  $W_{[\Delta k],chk}$  ( $E_{[\Delta k],chk}$ ) must coincide with the total (summed up over all the  $ES_n$ -directions) flux-based values  $W_{[\Delta k],fl}$  ( $E_{[\Delta k],fl}$ ) (per radian of the azimuth angle); in the  $xy$  case the “check sums” represent only two equatorial bands from the full set of the  $ES_n$ -directions and are always smaller than the corresponding total flux-based values (per unit length along the  $Z$ -axis).

It should be noted also that, because the angular distributions  $J_{[\Delta k],j}$  and  $dE_{[\Delta k]}/d\Omega$  are calculated by applying the flux-based formulae (i.e. by integrating the radiation intensity along the outer boundary), they should subsequently, for the final presentation, be renormalized by the factors

$$f_{rnW} = \frac{W_{[\Delta k],vol}}{W_{[\Delta k],fl}}, \quad f_{rnE} = \frac{E_{[\Delta k],vol}}{E_{[\Delta k],fl}}, \quad (9.10)$$

where  $W_{[\Delta k],vol}$  ( $E_{[\Delta k],vol}$ ) is the total (summed up over all the  $ES_n$ -directions) emission power (energy) in the selected spectral band, computed by applying the volumetric algorithm (which always yields more accurate results in the RALEF code), and  $W_{[\Delta k],fl}$  ( $E_{[\Delta k],fl}$ ) is the emission power (energy) obtained by applying the Gauss theorem to the entire computational domain.

## 10. TEST PROBLEMS

In this section we investigate the quality of our numerical scheme for radiation transport by applying it to several test problems, each of which has an easily computable exact solution. The corresponding exact solution may be either analytical, or in a form of explicit integrals that can be evaluated numerically with a high degree of accuracy (with at least 6–8 digits). The convergence of the RALEF numerical solutions to the exact solution is studied by examining three types of numerical errors.

When the compared quantity is represented by a single number (assumed to be generally non-zero) for the entire simulated 2D configuration — like for example the total power  $W_r$ ,

of radiative cooling — then we use a simple *relative error* for this quantity, defined as

$$\delta(W_r) = \frac{W_r^{(n)} - W_r}{W_r}, \quad (10.1)$$

where  $W_r^{(n)}$  is the numerical value from the RALEF simulation, and  $W_r$  is the exact value calculated from the corresponding exact solution. Note that the relative error  $\delta(W_r)$  of any “scalar” quantity  $W_r$  can be both positive and negative.

For any field dependent variable  $F$  [like the radiation intensity  $I_\nu(\vec{x}, \vec{\Omega}, \nu)$ , the volume specific radiative heating rate  $Q_r(\vec{x})$ , etc.), which is a function of one or more continuous independent variables, we introduce two different errors, namely, the  $L_2$ -norm error  $\delta_{L_2}(F)$  and the *Chebyshev-norm* (or the *supremum-norm*) error  $\delta_s(F)$ . Let for example the field quantity  $F$  be a function of position  $\vec{x}$  and of the photon propagation direction  $\vec{\Omega}$ . Then the  $L_2$ -norm error is defined as

$$\delta_{L_2}(F) = \left\{ \frac{\sum_{i,L} [F_{i,L}^{(n)} - F(\vec{x}_i, \vec{\Omega}_L)]^2}{\sum_{i,L} [F(\vec{x}_i, \vec{\Omega}_L)]^2} \right\}^{1/2}, \quad (10.2)$$

where  $F(\vec{x}_i, \vec{\Omega}_L)$  are the exact values of  $F$  at discrete collocation points  $\vec{x}_i$  and  $\vec{\Omega}_L$  for the quantity  $F$ ,  $F_{i,L}^{(n)}$  are the numerical values from the RALEF code, and the summation is performed over all the corresponding discrete points in space and angles. The supremum-norm error  $\delta_s$  is defined as

$$\delta_s(F) = \frac{\max_{i,L} \left\{ \left| F_{i,L}^{(n)} - F(\vec{x}_i, \vec{\Omega}_L) \right| \right\}}{\max_{i,L} \left\{ \left| F(\vec{x}_i, \vec{\Omega}_L) \right| \right\}}. \quad (10.3)$$

Both the  $L_2$ -norm and the supremum-norm errors are non-negative.

### 1. Searchlight beam in vacuum ( $xy$ geometry)

First of all we consider propagation of a mono-directional light beam (a searchlight beam) in vacuum (i.e. with the absorption coefficient  $k_{[k]}$  set equal to zero). This test problem reveals how diffusive is our transport algorithm, and what are its properties with respect to global flux conservation. The beam propagation is tested across a slab of unit thickness on three types of grid shown in Fig. 10.1. The mesh finesse is characterized by the numbers of cells  $n_x$ ,  $n_y$  per unit length along the  $x$  and  $y$  directions. All the results are given for a beam direction  $(\Omega_x, \Omega_y) = (0.1915, 0.6940)$  from the  $S_6$  set, which exhibits about the largest transverse dispersion on all types of mesh; note that beams with  $\Omega_x = 0$ , or  $\Omega_y = 0$ , or  $\Omega_x = \Omega_y$  have no dispersion on a square grid.

Figure 10.2 shows how a beam with sharp edges (i.e. with a box-like initial profile) spreads due to numerical diffusion upon exit from a slab on the random and the Kershaw grids; the results for the square grid are very similar to those of the random grid. The beam dispersion on the square and the random grids is moderate and similar to that described in Refs. [19, 32], but becomes very large on a strongly distorted Kershaw grid. On the square

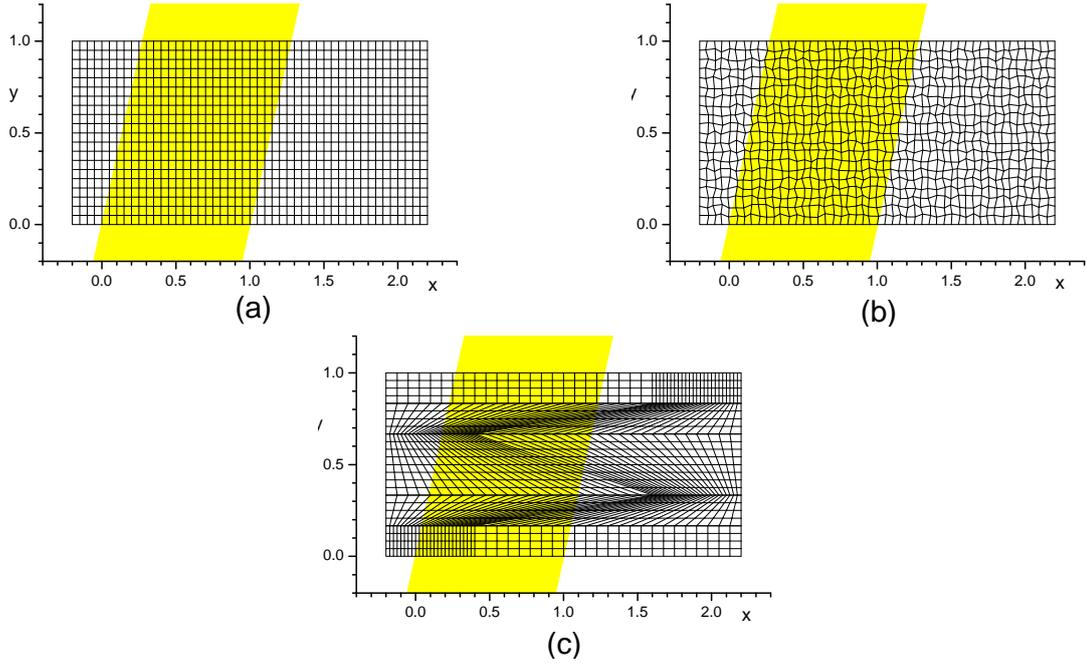


FIG. 10.1: A searchlight beam traversing a slab with (a) a square mesh, (b) a random mesh, and (c) a 1:3 Kershaw mesh.

grid it roughly amounts to  $0.3\sqrt{N}$  transverse cells after the beam propagates over  $N$  cells in the longitudinal direction. The latter implies rather slow convergence  $\sim h^{1/2}$  to the exact solution for sharp-edged beams as the mean cell size  $h = (n_x n_y)^{-1/2}$  is decreased. This convergence is documented in Table 10.1 with the values of the  $L_2$  error

$$\delta F_{L2} = \left[ \int (F_h - F_{ex})^2 dx \right]^{1/2}, \tag{10.4}$$

where  $F_h$  is the numerical solution on a mesh with cell size  $h$ , and  $F_{ex}$  is the exact solution satisfying the condition

$$\int F_{ex} dx = 1. \tag{10.5}$$

Figure 10.3 illustrates the numerical diffusion of a beam with the Gaussian initial profile,

$$F(x)|_{y=0} = \frac{2}{\sqrt{\pi}} \exp \left[ - \left( \frac{x - 0.5}{0.5} \right)^2 \right]. \tag{10.6}$$

Here, due to smooth spatial variation of the beam intensity  $F$ , we observe faster convergence (at least the first-order with respect to  $h$ ) and considerably lower errors than for a box-like beam profile.

Because our numerical scheme does not in general conserve radiative flux in vacuum, the total flux  $H_{out}$  of radiative energy escaping from the upper slab boundary is not equal to the incident flux  $H_{in}$ . Exact flux conservation is observed only in some particular cases like

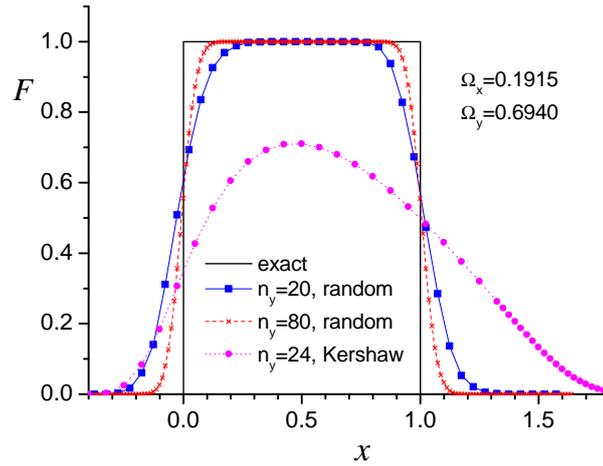


FIG. 10.2: Diffusive spreading of a searchlight beam with a box-like initial profile, propagating in vacuum along the random and the Kershaw grids.

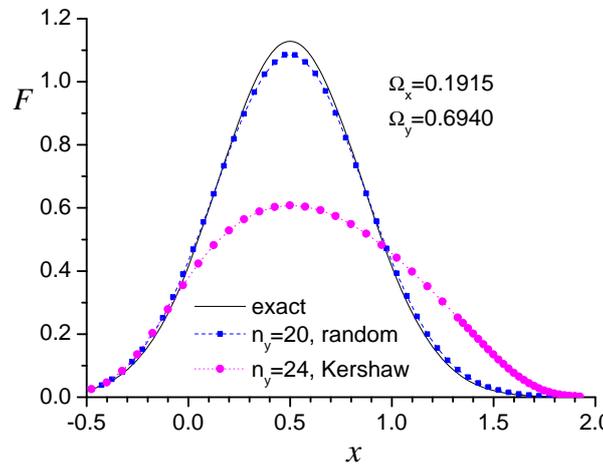


FIG. 10.3: Diffusive spreading of a searchlight beam with a Gaussian initial profile, propagating in vacuum along the random and the Kershaw grids.

between two parallel planes on an orthogonal mesh. Table 10.1 gives the values of relative flux deficit

$$\frac{\delta H}{H} = \frac{H_{out} - H_{in}}{H_{in}}, \tag{10.7}$$

calculated for a box-like spatial beam profile. One sees that the flux deficit is rather small on the random mesh but quite considerable on the strongly distorted Kershaw mesh. As mesh is refined, the flux deficit generally decreases in proportion to the first power of  $h$ .

TABLE 10.1: Test results for a searchlight beam in vacuum with two different (box-like and Gaussian) transverse spatial profiles.

	$n_x$	$n_y$	$\delta F_{L2}$ (box)	$\delta F_{L2}$ (Gauss)	$\delta H/H$ (box)
Square grid	10	10	0.278	0.055	$+2.1 \times 10^{-16}$
	20	20	0.224	0.029	$-4.4 \times 10^{-16}$
	40	40	0.190	0.0148	$+2.3 \times 10^{-16}$
	80	80	0.156	0.0075	$-1.1 \times 10^{-16}$
	160	160	0.130	0.0038	$+1.5 \times 10^{-15}$
Random grid	10	10	0.280	0.054	-0.0085
	20	20	0.221	0.029	+0.0074
	40	40	0.191	0.0146	-0.0043
	80	80	0.156	0.0074	-0.0011
	160	160	0.129	0.0037	+0.0013
Kershaw grid	10	12	0.465	0.48	+0.33
	20	24	0.465	0.36	+0.21
	40	48	0.382	0.25	+0.116
	80	96	0.316	0.161	+0.061
	160	192	0.266	0.0095	+0.031

## 2. Thermal emission from a slab ( $xy$ geometry)

In this problem we calculate thermal emission from a planar slab, which has a finite optical thickness  $\tau_0$  and a prescribed temperature distribution. Consider a planar layer  $-\infty < x, z < +\infty$  of a unit thickness  $0 \leq y \leq y_0 = 1$  along the  $y$ -axis, as is shown in Fig. 10.4. Assume that the temperature  $T = T(y)$  and the absorption coefficient  $k = k(y)$  are prescribed functions of  $y$  only; in addition, let the absorption coefficient  $k(y)$  be independent of the radiation frequency (a grey atmosphere). Then, the frequency-integrated radiation intensity  $I = I(y, \theta)$  is a function of only two variables, namely, of  $y$  and of the polar angle  $\theta$  with respect to the  $y$ -axis.

The transport equation for  $I$  can be written as

$$\mu \frac{\partial I}{\partial \tau} = B(\tau) - I(\tau, \mu), \tag{10.8}$$

where

$$\mu = \cos \theta, \quad \tau = \tau(y) = \int_0^y k(y) dy \tag{10.9}$$

is the optical thickness along the  $y$ -axis, and

$$B(\tau) = \frac{\sigma_{SB}}{\pi} T^4(\tau) \tag{10.10}$$

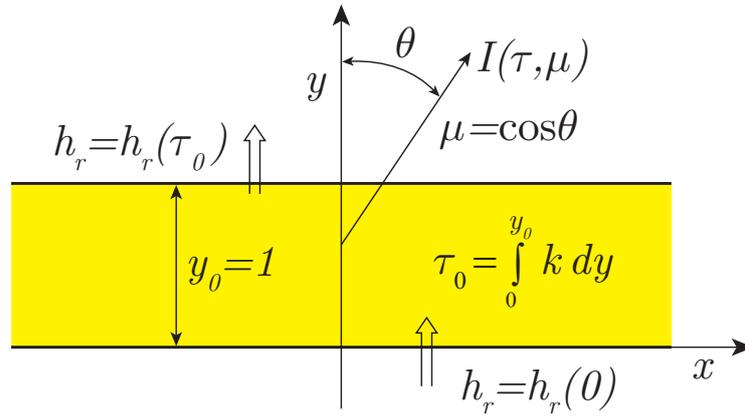


FIG. 10.4: Thermal emission from a slab.

is the Planckian source function. Its solution is given by

$$I(\tau, \mu) = \begin{cases} \frac{1}{\mu} \int_0^\tau B(\tau') \exp\left(\frac{\tau' - \tau}{\mu}\right) d\tau', & \mu > 0, \\ -\frac{1}{\mu} \int_\tau^{\tau_0} B(\tau') \exp\left(\frac{\tau' - \tau}{\mu}\right) d\tau', & \mu < 0, \end{cases} \quad (10.11)$$

where

$$\tau_0 = \int_0^1 y_0 k(y) dy. \quad (10.12)$$

The local volume-specific heating rate  $Q_r$  [erg cm<sup>-3</sup> s<sup>-1</sup>] of the fluid by radiation can be calculated as

$$\begin{aligned} Q_r(y) &= 4\pi k(y) \left[ \frac{1}{2} \int_{-1}^{+1} I(\tau, \mu) d\mu - B(\tau) \right] \\ &= 4\pi k(y) \left[ \frac{1}{2} \int_0^{\tau_0} B(\tau') E_1(|\tau - \tau'|) d\tau' - B(\tau) \right], \end{aligned} \quad (10.13)$$

where

$$E_n(x) = \int_1^\infty \frac{e^{-xt}}{t^n} dt \quad (10.14)$$

is the exponential integral of order  $n$ , which has the following basic properties

$$\frac{dE_{n+1}}{dx} = -E_n(x), \quad nE_{n+1}(x) = e^{-x} - xE_n(x). \quad (10.15)$$

The outgoing thermal flux, emerging from such a planar atmosphere at  $\tau = 0$  and  $\tau = \tau_0$ , can be calculated as

$$h_r(0) = 2\pi \int_{-1}^0 \mu I(0, \mu) d\mu = -2\pi \int_0^{\tau_0} E_2(\tau') B(\tau') d\tau', \quad (10.16)$$

$$h_r(\tau_0) = 2\pi \int_0^1 \mu I(\tau_0, \mu) d\mu = 2\pi \int_0^{\tau_0} E_2(\tau_0 - \tau') B(\tau') d\tau'. \quad (10.17)$$

One can easily prove the following identity (obvious from the physical point of view)

$$\int_0^{y_0} Q_r(y) dy = h_r(0) - h_r(\tau_0). \quad (10.18)$$

### 1. Isothermal slab

The simplest case would correspond to a uniform isothermal layer with constant values

$$k(y) = k_0 = \tau_0, \quad B(\tau) = B_0. \quad (10.19)$$

We simulate this case on a rectangle  $(x, y) \in [0, x_0] \times [0, y_0] = [0, 20] \times [0, 1]$ , and use the data from the reference unit square  $(x, y) \in [0, 1] \times [0, 1]$  for the analysis. The plane  $x = 0$  (in 3D space) is assumed to be a plane of symmetry. The full aspect ratio  $2x_0 : y_0 = 40 : 1$  of the represented physical region is high enough to be considered as infinite. The simulated region  $(x, y) \in [0, 20] \times [0, 1]$  is divided into two blocks, one at  $0 \leq x \leq 1$ , and the other at  $1 \leq x \leq 20$ , with the mesh being uniform in the first block, and progressive along the  $x$ -axis (with  $\Delta x_{i+1}/\Delta x_i = 1.05$ ) in the second block; see Fig. 10.5. The optical thickness  $\tau_0$  of the slab is varied by changing the value of the absorption coefficient  $k_0$ . For the boundary conditions we use the options `IRADFBC = 0` (zero incident intensity) and `IRADBEC = 2` [specified boundary  $B$ -temperature  $T_{B,bc} = (\pi B_0/\sigma_{SB})^{1/4}$ ] along all non-symmetry boundaries.

In the present test we explore the properties of our numerical algorithm by comparing the total radiative cooling power

$$W_{11} = - \sum_{0 < x, y < 1} W_i^r \quad (10.20)$$

[erg cm<sup>-1</sup> s<sup>-1</sup>] of the reference square  $(x, y) \in [0, 1] \times [0, 1]$  with the exact solution for the infinite (along  $x$ ) planar slab. In Eq. (10.20) the cooling rate  $W_{11}$  is calculated by summing up the radiative heating powers of individual cells  $W_i^r$  given by Eq. (??). Parallel to the volumetric quantity  $W_{11}$ , we can evaluate the radiative cooling rate of the reference region by summing up the integral energy fluxes across the lower,  $y = 0$ , and the upper,  $y = 1$ , boundaries of the computational block 1, i.e. by evaluating the surface quantity

$$H_{11} = -\mathcal{H}_1^r + \mathcal{H}_2^r. \quad (10.21)$$

The surface-associated quantity  $H_{11}$  is computed from the values of the radiation intensity  $I$  at mesh nodes along the boundaries. In the exact solution one has

$$\begin{aligned} W_{11} = H_{11} \equiv H_{ex} &= 2\pi B_0 [E_3(0) - E_3(\tau_0)] \\ &= 2\pi B_0 [1 - (1 - \tau_0)e^{-\tau_0} - \tau_0^2 E_1(\tau_0)]. \end{aligned} \quad (10.22)$$

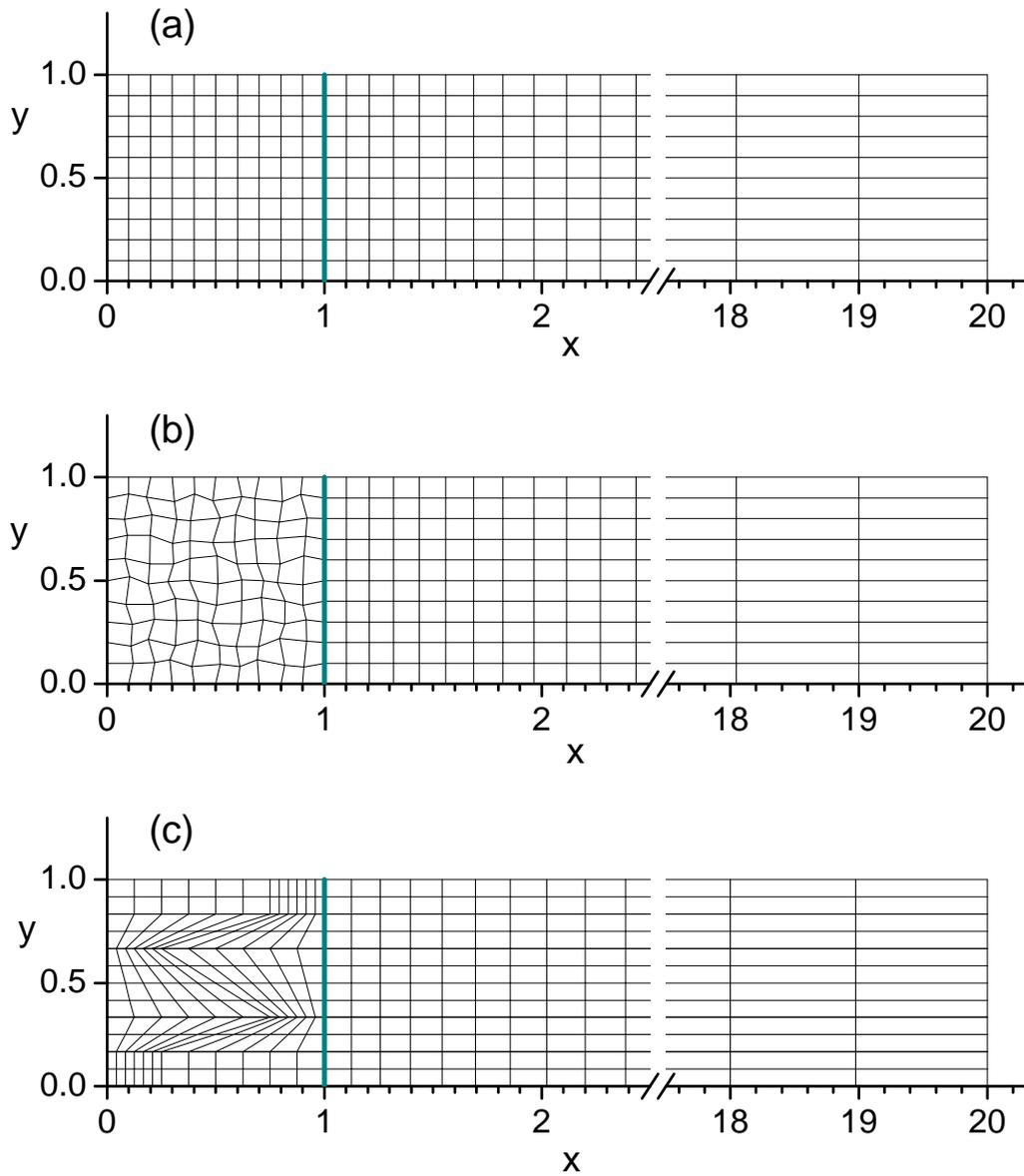


FIG. 10.5: Orthogonal (a), random (b), and Kershaw (c) numerical grids on a slab with an aspect ratio  $x_0 : y_0 = 20 : 1$  and a symmetry axis along the  $y$ -direction.

In the numerical solution the quantities  $W_{11}$  and  $H_{11}$  differ from one another, and from  $H_{ex}$ . Recall that in our algorithm it is the volumetric quantity  $W_{11}$  which is coupled to hydrodynamics. The difference between the values of  $W_{11}$  and  $H_{11}$  characterizes the degree of global energy non-conservation in our numerical scheme.

Figure 10.6 compares the values of  $W_{11}$  and  $H_{11}$ , calculated numerically with the  $S_6$  method (24 angular directions) on an orthogonal grid with  $n_x = n_y = 10$  mesh intervals along each axis inside the unit square  $[0, 1] \times [0, 1]$ , with the exact solution (10.22). Table 10.2 lists the relative errors for the same quantities, defined as

$$\delta H_{11} = H_{11}/H_{ex} - 1, \quad \delta W_{11} = W_{11}/H_{ex} - 1, \quad (10.23)$$

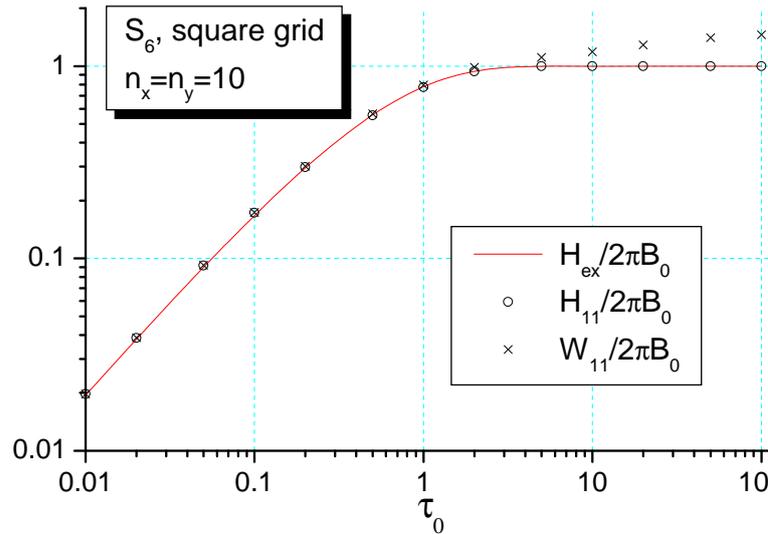


FIG. 10.6: Volumetric,  $W_{11}$ , and surface,  $H_{11}$ , radiative cooling rates (per unit surface area and normalized to  $2\pi B_0$ ) of a uniform planar slab as functions of its optical thickness  $\tau_0$ . Numerical results, obtained with the  $S_6$  method on a square grid with  $n_x \times n_y = 10 \times 10$  cells inside the unit square, are compared with the exact formula (10.22).

that are calculated with the same number of angular directions but on a finer grid with  $n_x = n_y = 40$ . In all cases the numerical errors on the random and square grids are pretty close to one another.

TABLE 10.2: Relative errors in the  $W_{11}$  and  $H_{11}$  values, calculated with the  $S_6$  method on a square (sm) and random (rm) grids with  $n_x = n_y = 40$  intervals inside the unit square for different values of the optical thickness  $\tau_0$ .

$\tau_0$	0.01	0.1	1.0	10	100	$10^8$
$\delta(H_{11})$ (sm)	1.27%	3.33%	-0.44%	$4.1 \times 10^{-8}$	$2.2 \times 10^{-16}$	$6.7 \times 10^{-16}$
$\delta(H_{11})$ (rm)	1.27%	3.32%	-0.46%	$-4.6 \times 10^{-8}$	$6.7 \times 10^{-16}$	$6.7 \times 10^{-16}$
$\delta(W_{11})$ (sm)	1.28%	3.40%	0.24%	5.84%	31.9%	50%
$\delta(W_{11})$ (rm)	1.28%	3.40%	0.30%	6.32%	31.6%	49%

Because our algorithm is built upon exact integration of the transport equation along each direction  $\vec{\Omega}_L$ , the numerical errors in the present uniform-slab test originate only from (i) the discretized angular integration in the  $S_n$  method, and from (ii) the linear interpolation of the residual intensity  $F = I - B$  along the edges of mesh quadrilaterals. The  $S_n$  error dominates at small optical thicknesses  $\tau_0 < 1$  — most notably at  $\tau_0 = 0.1$ , where it practically saturates as one increases  $n_x, n_y$  beyond  $n_x = n_y = 10$ ; in particular, this fact manifests itself in that the two errors  $\delta H_{11}$  and  $\delta W_{11}$  become practically equal for  $\tau_0 \leq 0.1$ . Table 10.3 illustrates the convergence of the  $S_n$  method for  $W_{11}$  at  $\tau_0 = 0.1$ . It is seen that generally this convergence

is rather slow.

TABLE 10.3: Convergence of the  $S_n$  method for the volumetric cooling rate  $W_{11}$  of a planar slab with  $\tau_0 = 0.1$  on a square grid with  $n_x = n_y = 40$ .

$S_n$	$S_2$	$S_4$	$S_6$	$S_8$	$S_{12}$	$S_{24}$	$S_{48}$	$S_{72}$
$\delta(W_{11})$	9.7%	6.9%	3.4%	3.0%	2.1%	1.35%	0.46%	0.29%

The linear-interpolation error becomes dominant at  $\tau_0 \gg 1$ , where all the radiative cooling occurs from the two mono-layers of optically thick (i.e. with  $k_0\Delta y \gg 1$ ) grid cells along the two boundaries. In the limit of  $\tau_0 \rightarrow \infty$  the numerical error  $\delta H_{11}$  essentially vanishes (see Table 10.2) for the  $S_6$  and higher  $S_n$  simply because the outgoing radiation intensity is accurately calculated as  $I = B_0$  while the  $S_n$  angular integration with  $n \geq 6$  reproduces exactly the first angular half-moment (see section 4.2). At the same time, the error  $\delta(W_{11})$  for the volumetric cooling rate reaches its maximum as  $\tau_0 \rightarrow \infty$ . For an orthogonal mesh, this limiting error is given by

$$\delta(W_{11,max}) = \frac{1}{2} \frac{\Delta y}{\Delta x}, \tag{10.24}$$

where  $\Delta y/\Delta x$  is the aspect ratio of mesh cells in the outer cell layer along the boundary with vacuum. On a square mesh with  $\Delta y = \Delta x$  it becomes 50%, as quoted in Table 10.2. The fact that the  $\delta(W_{11})$  error is finite (rather than infinite) for optically infinitely thick grid cells is a direct consequence of our specific integration procedure over the cell volume incorporated in Eqs. (??)-(??) for  $W_i^r$ .

Thus, our numerical algorithm for radiation transport has an important drawback of generating significant energy disbalance along an interface with a sharp transition from an optically very thick to an optically thin region. This disbalance can be significantly reduced by a rezoning procedure that tends to flatten the mesh along such interfaces on the opaque side. When, for example, we recalculate the case of  $\tau_0 = 10^8$  from Table 10.2 on the  $n_x \times n_y = 10 \times 80$  random mesh, we obtain  $\delta(W_{11}) = 6.15\%$  instead of  $\delta(W_{11}) = 49\%$ .

2. Slab with a sine B-profile: optically thin mesh

Our numerical scheme for radiation transport is best suited for optically thin grids, i.e. for grids where the optical thickness of every individual cell is below unity. In this subsection we test our code against a non-uniform temperature distribution

$$k(y) = k_0 = \tau_0, \quad B(\tau) = \sin\left(\frac{\pi\tau}{\tau_0}\right) = \sin(\pi y) \tag{10.25}$$

with a fixed value  $\tau_0 = 2$  of the slab optical thickness  $\tau_0$ . Simulations have been performed on the three types of numerical grid shown in Fig. 10.5. Along the non-symmetry boundaries the boundary conditions of zero incident intensity ( $IRADFBC = 0$ ) and of specified boundary  $B$ -temperature  $T_{B,bc} = 0$  ( $IRADBBC = 2$ ) were applied.

Figure 10.7 illustrates how our algorithm reproduces the distribution of the volume-specific radiative cooling rate  $-Q_r(y)$  across the slab thickness  $y$ ; the exact values of  $Q_r(y)$

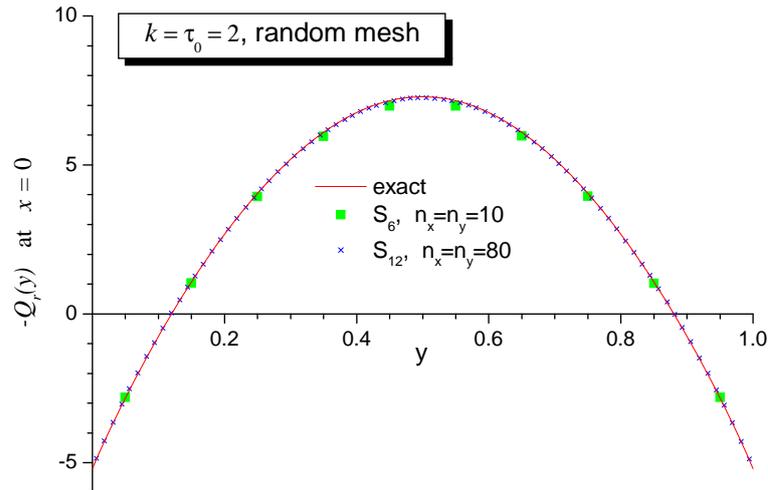


FIG. 10.7: Volume-specific radiative cooling rate  $-Q_r$  of a planar slab with the source-function profile (10.25) as a function of its depth  $y$  for the value of  $k_0 = \tau_0 = 2$ . Two sets of numerical results are shown, one obtained with the  $S_6$  method on a random  $n_x \times n_y = 10 \times 10$  grid (inside the reference unit square), and the other computed with the  $S_{12}$  method on a random  $n_x \times n_y = 80 \times 80$  grid.

were computed from Eq. (10.13). The plotted numerical values were obtained by calculating the ratio of the integral cell heating power  $W_i^r$  to the cell volume  $V_i$ ,

$$Q_i^r = \frac{W_i^r}{V_i}, \quad (10.26)$$

and ascribing this quantity to the geometrical cell center. Only the data from the physical cells along the symmetry boundary  $x = 0$  are plotted in Fig. 10.7. It is seen that general agreement with the exact solution is quite good even for a relatively coarse  $n_x = n_y = 10$  grid and the  $S_6$  angular discretization.

More detailed information on the accuracy and spatial convergence of our scheme on optically thin grids is presented in Table 10.4, where the  $L_2$ -norm errors  $\delta_{L_2}(F)$  and  $\delta_{L_2}(Q_r)$  [as defined by Eq. (10.2)] for the quantities  $F = I - B$  and  $Q_r$  are given; here  $F_{i,L}$  is the value of  $F$  at mesh node  $i$  with the  $y$ -coordinate  $y_i$  for propagation direction  $L$ ,  $Q_{i,L}^r$  is the value of  $Q_r$  in cell  $i$  [as given by Eq. (10.26)],  $y_{c,i}$  is the  $y$ -coordinate of the cell center  $i$ ; the exact values  $F(y_i, \mu_L) = I(y_i, \mu_L) - B(y_i)$ ,  $Q_r(y_{c,i})$  are calculated from Eqs. (10.11) and (10.13); summation in Eq. (10.2) is performed over all the physical nodes (cells) inside the unit square  $0 \leq x, y \leq 1$ , and over all angular directions  $L$  for  $\delta F_{L2}$ . Also given are the relative errors  $\delta(H_{11})$ ,  $\delta(W_{11})$  in the integral cooling rates  $H_{11}$ ,  $W_{11}$  as defined in Eqs. (10.20), (10.21) and (10.23).

From Table 10.4 one learns that the errors in radiative quantities are not sensitive to distortions of optically thin grids, i.e. are practically the same on the square, random, and Kershaw grids. The error  $\delta F_{L2}$  in the residual intensity  $F$  does not depend on angular integration and demonstrates a monotonous 1-st-order convergence as one refines the mesh — in full agreement with the linear interpolation of  $B$  and  $F$  along the edges of mesh quadrilaterals, and linear interpolation of  $B$  from cell centers to nodes [note that the exact

TABLE 10.4:  $L_2$  errors  $\delta F_{L2}$ ,  $\delta Q_{L2}$  for the residual intensity  $F = I - B$  and the volume-specific radiative heating rate  $Q_r$  inside the reference  $[0, 1] \times [0, 1]$  square, as well as relative errors  $\delta H_{11}$ ,  $\delta(W_{11})$  for the bulk radiative cooling rate [as defined in Eq. (10.23)] calculated with the  $S_{12}$  method on different grids for the  $\tau_0 = 2$  case.

	$n_x = n_y$	$\delta F_{L2}$	$\delta_{L2}(Q_r)$	$\delta(H_{11})$	$\delta(W_{11})$
Square grid	10	$3.97 \times 10^{-2}$	$1.67 \times 10^{-2}$	$-2.56 \times 10^{-2}$	$-8.82 \times 10^{-3}$
	20	$1.99 \times 10^{-2}$	$3.76 \times 10^{-3}$	$-1.08 \times 10^{-2}$	$-4.24 \times 10^{-4}$
	40	$1.01 \times 10^{-2}$	$3.98 \times 10^{-3}$	$-6.07 \times 10^{-3}$	$+2.81 \times 10^{-3}$
	80	$5.05 \times 10^{-3}$	$5.26 \times 10^{-3}$	$-4.31 \times 10^{-3}$	$-1.01 \times 10^{-3}$
Random grid	10	$4.20 \times 10^{-2}$	$2.01 \times 10^{-2}$	$-2.67 \times 10^{-2}$	$-6.30 \times 10^{-3}$
	20	$2.14 \times 10^{-2}$	$6.93 \times 10^{-3}$	$-1.15 \times 10^{-2}$	$+2.41 \times 10^{-3}$
	40	$1.10 \times 10^{-2}$	$4.81 \times 10^{-3}$	$-6.42 \times 10^{-3}$	$+1.09 \times 10^{-3}$
	80	$5.60 \times 10^{-3}$	$5.21 \times 10^{-3}$	$-4.50 \times 10^{-3}$	$-6.14 \times 10^{-4}$
Kershaw grid	12	$3.60 \times 10^{-2}$	$1.52 \times 10^{-2}$	$-2.80 \times 10^{-2}$	$+1.22 \times 10^{-4}$
	24	$1.79 \times 10^{-2}$	$6.25 \times 10^{-3}$	$-1.34 \times 10^{-2}$	$+3.72 \times 10^{-3}$
	48	$8.98 \times 10^{-3}$	$5.64 \times 10^{-3}$	$-7.71 \times 10^{-3}$	$+1.61 \times 10^{-3}$
	96	$4.50 \times 10^{-3}$	$6.08 \times 10^{-3}$	$-5.24 \times 10^{-3}$	$-4.03 \times 10^{-4}$

initial values of the source function  $B$  are loaded at the cell centers  $(x_{c,i}, y_{c,i})$ . The errors  $\delta Q_{L2}$ ,  $\delta(W_{11})$  in radiative cooling rate demonstrate a somewhat different behavior: they rapidly drop to a level of about 0.5% as the optical thickness of mesh cells along the  $y$ -axis drops below  $k_0 \Delta y \lesssim 0.1$ , i.e. for  $n_x, n_y \gtrsim 20$ , where they saturate due to numerical errors of the  $S_{12}$  angular quadratures.

### 3. Slab with a sine $B$ -profile: optically thick mesh and the diffusion limit

An important issue for any radiation-transport numerical scheme is whether it recovers the diffusion limit on optically thick grids [18], i.e. on grids with optical thickness of individual cells much larger than unity. We can easily test the diffusion limit in the slab geometry by considering the limit  $k_0 y_0 \gg n_y$  for the same sine-like source function profile (10.25). In the diffusion limit, the volume-specific radiative heating rate  $Q_r$  in a grey atmosphere is given by

$$Q_r = -\operatorname{div}(\vec{h}_r) = \frac{4\pi}{3} \operatorname{div} \left( \frac{1}{k} \nabla B \right) = \frac{4\pi^3}{3\tau_0} \sin(\pi y). \quad (10.27)$$

Numerical simulations for  $\tau_0 \gg 1$  have been performed on the same grids and with the same boundary conditions as for the case of  $\tau_0 = 2$ , described in the previous subsection.

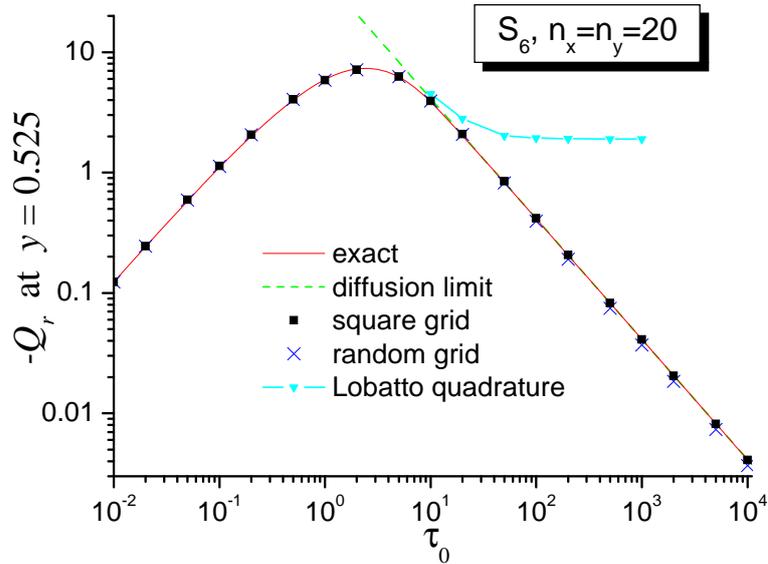


FIG. 10.8: Radiative cooling rate  $-Q_i^r$  in the central cell [i.e. at  $(x_{c,i}, y_{c,i}) = (0.025, 0.525)$ ] on the square grid with  $n_x = n_y = 20$ ] of a planar slab as a function of its optical thickness  $\tau_0$ . The diffusion solution is shown with a straight dashed line. Simulations have been performed on a square and random grids with  $n_x \times n_y = 20 \times 20$  cells in the reference square  $(x, y) \in [0, 1] \times [0, 1]$  with the  $S_6$  method.

Figure 10.8 shows variation of the volume-specific radiative cooling rate  $-Q_r$  at the mid-point of the simulated slab with the slab optical thickness  $\tau_0$ . The numerical values  $-Q_i^r$  were taken from a grid cell with the center coordinates  $(x_{c,i}, y_{c,i})$  closest to the slab mid-point  $(x, y) = (0, 0.5)$ . It is clearly seen that our algorithm does reproduce the diffusion limit, although the numerical errors on a random grid are noticeably higher than on a square grid (see Table 10.5). The curve marked “Lobatto quadrature” depicts the situation where the diffusion limit is not reproduced and a very large error in the heating/cooling rate of optically thick grid cells can be made. Such situation occurs, for example, when we choose to integrate the transport equation along short characteristics by employing an implicit Runge-Kutta method — as is recommended in Ref. [19], namely, the unconditionally positive and stable 2-nd order Lobatto IIIC formula [33].

More information on the numerical errors in the diffusion limit is presented in Fig. 10.9 and Table 10.5. First of all note that the relative error in the integral quantity  $W_{11}$  reaches some 30% as  $\tau_0 \rightarrow \infty$  due to the reasons explained in subsection 10.2.1. This error originates entirely from the two layers of optically thick cells along the lower,  $y = 0$ , and the upper,  $y = 1$ , boundaries, where the diffusion approximation does not apply. Hence, to make the  $\delta Q_{L2}$  error representative of the diffusion limit, these two layers of boundary cells are excluded from the summation in Eq. (10.2), the corresponding quantity is denoted as  $\delta_{L2}(Q')$  and listed in Table 10.5.

From Table 10.5 we learn that in the diffusion limit the numerical error  $\delta_{L2}(Q')$  in the local heating rate becomes sensitive to mesh distortions: for the random and Kershaw meshes

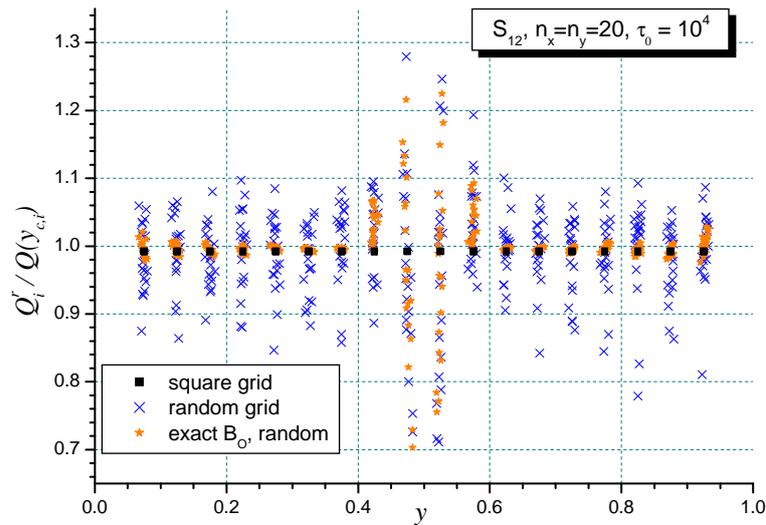


FIG. 10.9: Ratios of  $Q'_i$ , obtained in the RALEF simulation with the  $S_{12}$  method on  $20 \times 20$  square and random grids, to the exact values  $Q_r(y_{c,i})$  for all grid cells inside the unit square  $(x, y) \in [0, 1] \times [0, 1]$ . The optical thickness of the slab is fixed at  $\tau_0 = 10^4$ , i.e. each grid cell has an optical thickness of  $\approx 500$ .

it turns out to be about a factor 10 larger (around 10%) than on the square mesh (below 1%) — which also immediately catches the eye in Fig. 10.9. The main cause of this error is the linear interpolation of  $B$  and  $F$  along the edges of quadrilaterals, and from cell centers to nodes for the source function  $B$ . This is confirmed by a separate run on a random grid, where the exact values of  $B$  have been used at all stages of the algorithm; the results of this simulation are shown as red stars in Fig. 10.9. The scatter of stars around unity in Fig. 10.9 is close to  $\pm 1\%$  for major portion of the slab, except for a region near the midplane where the source function  $B$  reaches its maximum and the strictly monotonic second-order interpolation of  $B$  along the characteristics becomes rather inaccurate.

An important fact for our transport algorithm is that, because of the linear interpolation of the source function  $B$  on the mesh, we loose spatial convergence for the radiative heating rate  $Q_r$  in the diffusion limit: as we refine the mesh by increasing  $n_x, n_y$ , the  $L_2$ -norm error  $\delta Q'_{L2}$  remains practically unchanged. This is explained by the fact that  $Q_r$  in the diffusion limit is proportional to the second spatial derivative of the source function  $B$  [see Eq. 10.27]. The particular value of the mean error in  $Q_r$  in the diffusion limit depends on the degree of mesh distortion, and for our two examples of the random and Kershaw meshes amounts to 7–10%.

### 3. Radiation transport in a cylindrical cavity ( $xy$ geometry)

From the point of view of application to ICF hohlraums [34], of particular interest would be a test problem where radiation propagates across an optically thin spherical or cylindrical cavity between two concentric opaque layers. Here we consider one of the simplest such configurations in the  $(x, y)$  geometry, where radiation is emitted by a hot and optically thick cylindrical rod of radius  $R_1$  into a surrounding cylindrical cavity with the outer radius

TABLE 10.5: Same as Table 10.4 but for  $n_x = n_y = 20$  and different values of  $\tau_0 \gg 1$ .

	$\tau_0$	$\delta_{L_2}(F)$	$\delta_{L_2}(Q'_r)$	$\delta(H_{11})$	$\delta(W_{11})$
Square grid	10	$1.58 \times 10^{-2}$	$3.25 \times 10^{-2}$	$-4.85 \times 10^{-3}$	+0.0125
	$10^2$	$7.65 \times 10^{-3}$	$1.92 \times 10^{-2}$	$-6.63 \times 10^{-3}$	-0.185
	$10^3$	$7.18 \times 10^{-3}$	$5.26 \times 10^{-3}$	$-7.17 \times 10^{-3}$	-0.309
	$10^4$	$7.18 \times 10^{-3}$	$7.96 \times 10^{-3}$	$-7.17 \times 10^{-3}$	-0.321
Random grid	10	$1.68 \times 10^{-2}$	$3.93 \times 10^{-2}$	$-5.08 \times 10^{-3}$	+0.0218
	$10^2$	$1.30 \times 10^{-2}$	$6.53 \times 10^{-2}$	$-6.33 \times 10^{-3}$	-0.1607
	$10^3$	$1.40 \times 10^{-2}$	$8.16 \times 10^{-2}$	$-6.85 \times 10^{-3}$	-0.2995
	$10^4$	$1.42 \times 10^{-2}$	$8.44 \times 10^{-2}$	$-6.86 \times 10^{-3}$	-0.3136
Kershaw grid	10	$2.12 \times 10^{-2}$	$5.38 \times 10^{-2}$	$-5.09 \times 10^{-3}$	+0.064
	$10^2$	$1.70 \times 10^{-2}$	0.201	$-4.49 \times 10^{-3}$	+0.049
	$10^3$	$1.33 \times 10^{-2}$	$7.99 \times 10^{-2}$	$-4.98 \times 10^{-3}$	-0.264
	$10^4$	$1.32 \times 10^{-2}$	$9.72 \times 10^{-2}$	$-4.99 \times 10^{-3}$	-0.318

$R_2 > R_1$ . The transverse section of such a cylindrical configuration in the  $(x, y)$ -plane is shown schematically in Fig. 10.10.

We adopt a box-like radial profile

$$B = B(r) = \begin{cases} B_0, & 0 \leq r \leq R_1, \\ 0, & R_1 < r \leq R_2, \end{cases} \quad (10.28)$$

of the spectrally integrated source function  $B = \int_0^\infty B_\nu$ , and assume that at all frequencies  $\nu$  the absorption coefficient

$$k_\nu = k_\nu(r) = \begin{cases} \infty, & 0 \leq r \leq R_1, \\ 0, & R_1 < r \leq R_2, \end{cases} \quad (10.29)$$

is infinitely large in the hot emitting region  $r \leq R_1$ , and zero outside it; here  $r = (x^2 + y^2)^{1/2}$  is the cylindrical radius. Then, the exact solution of the transfer equation for the spectrally integrated radiation intensity  $I = \int_0^\infty I_\nu d\nu$  is given by

$$I = I(r, \mu, \phi) = \begin{cases} B_0, & 0 \leq r < R_1, \\ B_0, & r > R_1, |\phi| < \arcsin(R_1/r), \\ 0, & r > R_1, |\phi| > \arcsin(R_1/r). \end{cases} \quad (10.30)$$

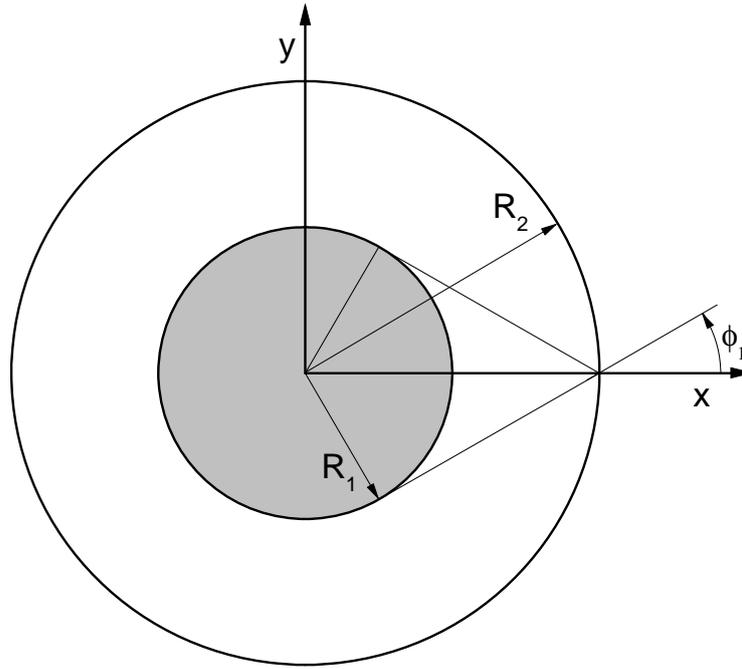


FIG. 10.10: The  $(x, y)$  cross-section of a transparent cylindrical cavity between a “hot” opaque rod of radius  $R_1$  and the outer vacuum boundary at  $r = R_2$ . At  $r = R_2$  the radiation intensity is non-zero only within the range of azimuth angles  $-\phi_1 < \phi < +\phi_1$ , where  $\sin \phi_1 = R_1/R_2$ .

Here the azimuth angle  $\phi \in [-\pi, +\pi]$ , measured in the  $(x, y)$ -plane with respect to the  $x$ -axis, and the polar angle  $\theta = \arccos \mu \in [0, \pi]$ , measured with respect to the cylinder axis (the  $z$ -axis), specify the direction of the 3D photon propagation vector

$$\vec{\Omega} = \left\{ \sqrt{1 - \mu^2} \cos \phi; \sqrt{1 - \mu^2} \sin \phi; \mu \right\}. \quad (10.31)$$

From Eq. (10.30) one readily calculates the exact values of the total integral intensity

$$U = U(r) = 2 \int_{-1}^{+1} d\mu \int_0^\pi I(r, \mu, \phi) d\phi = \begin{cases} 4\pi B_0, & 0 \leq r < R_1, \\ 4B_0 \arcsin(R_1/r), & R_1 < r, \end{cases} \quad (10.32)$$

and of the specific energy flux

$$h_r = h_r(r) = 2 \int_{-1}^{+1} \sqrt{1 - \mu^2} d\mu \int_0^\pi I(r, \mu, \phi) \cos \phi d\phi = \begin{cases} 0, & 0 \leq r < R_1, \\ \pi B_0 \left( \frac{R_1}{r} \right), & R_1 < r. \end{cases} \quad (10.33)$$

Numerical simulations have been performed for a quarter-circle segment of the considered cylindrical region  $r \leq R_2$ , shown in Fig. 10.11 for  $R_1 = 1$ ,  $R_2 = 4$ , with the reflective boundary conditions along the  $x$ - and  $y$ -axes. The simulation domain was covered by a polar mesh, consisting of two blocks: block 1 for the inner opaque region  $r < R_1$ , and block 2 for the vacuum cavity at  $R_1 < r < R_2$ . The value of  $R_1$  was fixed at  $R_1 = 1$ , while

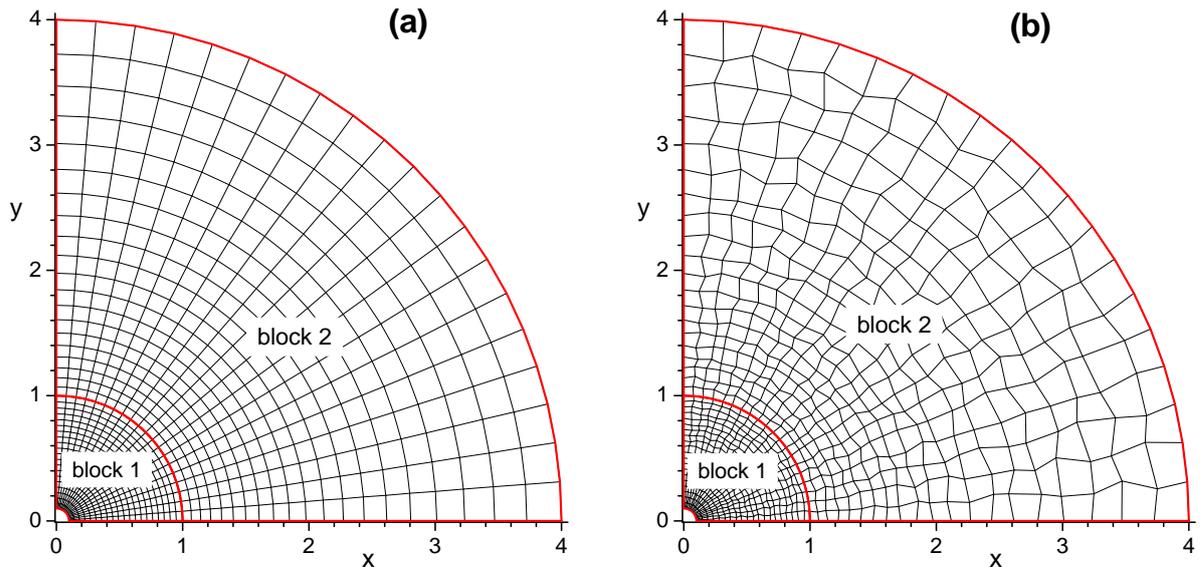


FIG. 10.11: Two-block quasi-square (a) and random (b) polar meshes with  $n_h = 20$  cells along each direction in each block over the quarter-circle simulation domain. Block 1 covers the opaque central region at  $r < R_1 = 1$ , block 2 covers the vacuum cavity at  $R_1 < r < R_2 = 4$ .

different values of  $R_2$  were tested. Equal number of cells  $n_h$  was assumed for both the radial and azimuthal directions in both blocks. In block 1 all cells had the same radial size, while in block 2 the radial cell size increased in geometric progression, so that  $(\Delta r)_n / (\Delta r)_1 = R_2 / R_1$ . In this way a practically uniform aspect ratio has been ensured for all cells in block 2 (a quasi-square mesh). The random mesh of Fig. 10.11b was obtained by adding a 20% (with respect to the local cell-edge length) randomly directed displacement to every inner node in blocks 1 and 2 of the quasi-square mesh of Fig. 10.11a.

An artificial inner boundary at  $r = 0.1$  of block 1, assumed to be transparent for radiation, had no effect whatsoever on the results of simulations because of the infinitely large opacity in this block. For the radiation opacity, fixed values

$$k_\nu = \begin{cases} 10^{12}, & \text{block 1,} \\ 10^{-20}, & \text{block 2,} \end{cases} \quad (10.34)$$

were used in all the simulations. At the outer boundary  $r = R_2$ , the boundary condition of zero incident external radiation was used.

### 1. Total flux balance

Because our algorithm for radiation transport does not ensure flux conservation in vacuum (see section 33), we begin by checking the global radiation energy balance between the two concentric surfaces  $r = R_1$  and  $r = R_2$ , i.e. between the edges  $\mathbf{ib} = 1$  and  $\mathbf{ib} = 2$  of block 2. According to Eq. (10.33) the exact total radiative energy flux (per unit cylinder length)

across either of these two edges is

$$\mathcal{H}_r = \frac{\pi}{2} R_1 h_r(R_1) = \frac{\pi^2}{2} R_1 B_0. \quad (10.35)$$

Table 10.6 lists the relative errors  $\delta(\mathcal{H}_1)$  and  $\delta(\mathcal{H}_2)$  [as defined by Eq. (10.1)] of the numerically calculated fluxes

$$\mathcal{H}_1^{(n)} = \sum_{i \in \text{ib}=1} H_{i1}^r, \quad \mathcal{H}_2^{(n)} = \sum_{i \in \text{ib}=2} H_{i1}^r, \quad (10.36)$$

for different values of the cavity aspect ratio  $R_2/R_1$ , and different numbers  $n_g$  of cells along the  $r$  and  $\phi$  mesh directions for the quasi-square mesh of Fig. 10.11a; the fluxes  $H_{im}^r$  across the edges of individual cells are calculated according to Eq. (??). Results obtained for the random mesh of Fig. 10.11b are very close to those for the quasi-square mesh and are not given here.

The error  $\delta(\mathcal{H}_1)$  is typically rather small and can be ignored: it characterizes the accuracy of the  $S_n$  method in evaluating the angular half-moments (see section 4.2.2). The relative disbalance of the radial radiation-energy flux is represented by the difference  $\delta(\mathcal{H}_2) - \delta(\mathcal{H}_1)$ , which is dominated by the second error  $\delta(\mathcal{H}_2)$ . This error depends strongly on the cavity aspect ratio  $R_2/R_1$  and the fineness of the numerical grid (represented by the parameter  $n_g$ ), but is not sensitive to the order of the  $S_n$  quadrature. As is seen from Table 10.6, the flux disbalance can be rather large in cavities with large aspect ratios  $R_2/R_1 \gg 1$ , which is a clear indication of poor overall accuracy of our transport algorithm for such cases. To improve the accuracy, one has to refine the grid, i.e. to increase  $n_g$ . Table 10.6 clearly demonstrates that the relative flux disbalance decreases in inverse proportion to  $n_g$  — in accord with the first order of the numerical scheme for the transfer equation. In general, one has to use a mesh with  $n_g \gtrsim (20-30)(R_2/R_1)$  to ensure a good accuracy  $\sim 1\%$  for radiation transport across a vacuum cavity with a not too large aspect ratio  $R_2/R_1$ .

## 2. The ray effect

In the  $S_n$  method the discretized directions of radiation propagation are fixed in space. As a result, strong spurious non-uniformities can develop in the spatial distribution of the radiation intensity where radiation propagates over extended transparent (or semi-transparent) regions — the so called *ray effect*. In our test problem the ray effect manifests itself as the angular variation of the radiation energy flux  $h_r = h_r(r, \phi)$  (or the energy density) along any cylindrical surface with a fixed radius  $r > R_1$  — despite the fact that in the “hot” emitting region at  $r \leq R_1$  the radiation field is perfectly uniform along the  $\phi$  coordinate. Figure 10.12 provides a vivid illustration of the ray effect observed for the  $S_6$  angular quadrature in a cylindrical cavity with an aspect ratio  $R_2/R_1 = 4$ . Recall once again that the material parameters, numerical grid and the radiation source function have been perfectly cylindrically symmetric in the corresponding simulation.

As a quantitative measure of the ray effect, we consider the angular variation of the specific radiation-energy flux  $h_r(r, \phi)$  at the outer boundary  $r = R_2$  of the considered cylindrical cavity. Caused by the ray effect, this variation is always smoothed to a certain extent by the numerical diffusion of the short characteristic method (see the test problem in section 10.1). To reduce the diffusive smoothing to a negligible level, we performed simulations

TABLE 10.6: Relative errors for the total fluxes  $\mathcal{H}_1^{(n)}$ ,  $\mathcal{H}_2^{(n)}$  across the boundaries  $r = R_1$ ,  $r = R_2$  of the cylindrical vacuum cavity, calculated for different numbers  $n_g$  of the quasi-square mesh of Fig. 10.11a with the  $S_{24}$  angular quadratures.

	$n_g$	$\delta(\mathcal{H}_1)$	$\delta(\mathcal{H}_2)$	$\delta(\mathcal{H}_2) - \delta(\mathcal{H}_1)$
$R_2/R_1 = 2$	20	$-1.95 \times 10^{-3}$	$-5.81 \times 10^{-3}$	$-3.87 \times 10^{-3}$
	40	$-6.37 \times 10^{-4}$	$-1.96 \times 10^{-3}$	$-1.32 \times 10^{-3}$
	80	$-2.98 \times 10^{-4}$	$-8.11 \times 10^{-4}$	$-5.13 \times 10^{-4}$
	160	$-2.19 \times 10^{-4}$	$-4.38 \times 10^{-4}$	$-2.20 \times 10^{-4}$
	320	$-1.99 \times 10^{-4}$	$-3.00 \times 10^{-4}$	$-1.01 \times 10^{-4}$
$R_2/R_1 = 4$	20	$-1.95 \times 10^{-3}$	$+3.05 \times 10^{-2}$	$+3.25 \times 10^{-2}$
	40	$-6.37 \times 10^{-4}$	$+1.67 \times 10^{-2}$	$+1.72 \times 10^{-2}$
	80	$-2.98 \times 10^{-4}$	$+8.61 \times 10^{-3}$	$+8.91 \times 10^{-3}$
	160	$-2.19 \times 10^{-4}$	$+4.30 \times 10^{-3}$	$+4.52 \times 10^{-3}$
	320	$-1.99 \times 10^{-4}$	$+2.08 \times 10^{-3}$	$+2.27 \times 10^{-3}$
$R_2/R_1 = 8$	20	$-1.82 \times 10^{-3}$	+0.119	+0.121
	40	$-5.31 \times 10^{-4}$	$+6.21 \times 10^{-2}$	$+6.26 \times 10^{-2}$
	80	$-2.06 \times 10^{-4}$	$+3.15 \times 10^{-2}$	$+3.17 \times 10^{-2}$
	160	$-1.24 \times 10^{-4}$	$+1.58 \times 10^{-2}$	$+1.59 \times 10^{-2}$
	320	$-1.99 \times 10^{-4}$	$+7.80 \times 10^{-3}$	$+8.00 \times 10^{-3}$

with a sufficiently large number of mesh cells  $n_g = 320$  for  $R_2/R_1 = 4$ , and  $n_g = 640$  for  $R_2/R_1 = 8$ . Figure 10.13 shows thus calculated variation  $h_r(R_2, \phi)$  in a cavity with the aspect ratio  $R_2/R_1 = 4$  for different  $n$  values in the  $S_n$  method: plotted is the normalized flux

$$\tilde{h}_r = \tilde{h}_r(\phi) = \left(\frac{R_2}{R_1}\right) \frac{h_r(R_2, \phi)}{\pi B_0}, \tag{10.37}$$

whose exact value, according to Eq. (10.33), is  $\tilde{h}_r = 1$ . Table 10.7 lists the relative errors

$$\delta_s(h_2) = \max_{i \in [r_i=R_2]} |\tilde{h}_{r,i} - 1|, \quad \delta_{L_2}(h_2) = \left[ (n_h + 1)^{-1} \sum_{i \in [r_i=R_2]} (\tilde{h}_{r,i} - 1)^2 \right]^{1/2} \tag{10.38}$$

for the numerically calculated values  $h_{r,i}$  at nodes  $i$  along the outer boundary  $r = R_2$ . The amplitude of angular variation of the integral intensity  $U_r = U_r(r, \phi)$ , not given here, is

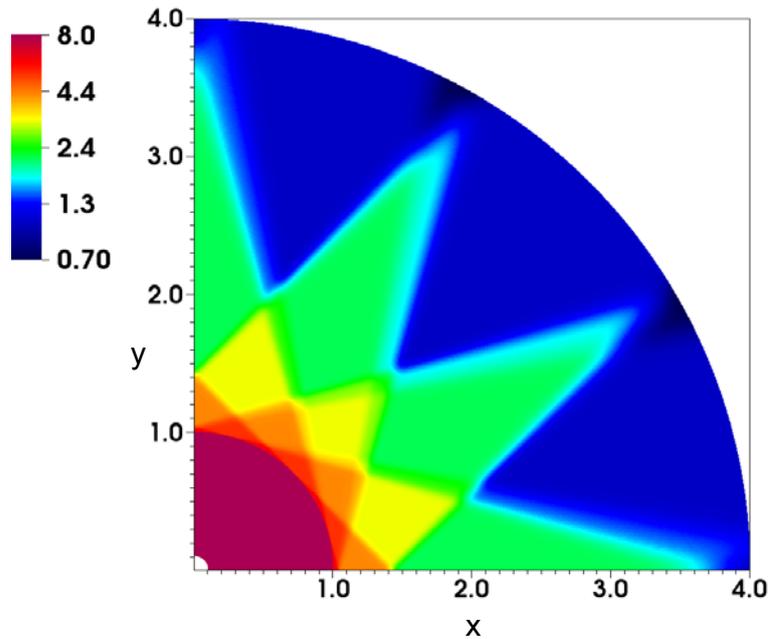


FIG. 10.12: 2D color plot of the integral radiation intensity  $U_r = U_r(r, \phi)$  in a cylindrical configuration with a vacuum cavity at  $R_1 < r < R_2$ , obtained for  $R_2/R_1 = 4$  with the  $S_6$  method on a quasi-square grid with  $n_h = 160$ .

similar to that of  $h_r(r, \phi)$ , and typically exceeds the latter by some 20–50%.

TABLE 10.7: Relative errors for the specific radiation-energy flux  $h_r$  at the outer boundary  $r = R_2$  of the cylindrical cavity at  $R_1 < r < R_2$ , obtained with different  $S_n$  quadratures on a quasi-square grid with  $n_h = 320$  for  $R_2/R_1 = 4$ , and  $n_h = 640$  for  $R_2/R_1 = 8$ .

	$S_6$	$S_{12}$	$S_{24}$	$S_{48}$	$S_{96}$
$R_2/R_1 = 4 : \delta_s(h_2)$	0.41	0.18	$6.8 \times 10^{-2}$	$1.9 \times 10^{-2}$	$4.7 \times 10^{-3}$
$\delta_{L_2}(h_2)$	0.15	$8.5 \times 10^{-2}$	$2.4 \times 10^{-2}$	$5.2 \times 10^{-3}$	$2.3 \times 10^{-3}$
$R_2/R_1 = 8 : \delta_s(h_2)$	1.16	0.64	0.21	$7.0 \times 10^{-2}$	$1.9 \times 10^{-2}$
$\delta_{L_2}(h_2)$	0.90	0.20	$6.9 \times 10^{-2}$	$1.8 \times 10^{-2}$	$5.0 \times 10^{-3}$

A straightforward way to suppress the ray effect is to increase the order  $n$  of the  $S_n$  quadrature. For cavities with moderate aspect ratios  $R_2/R_1 \lesssim 4$  good results are obtained with  $n \gtrsim 36$ , where the  $L_2$ -norm error  $\delta_{L_2}(h_2)$  falls below 1%. Note that, for a fixed ratio  $R_2/R_1$ , the amplitude of spurious angular variations in  $h_r$  and  $U_r$  due to the ray effect rapidly decreases with the increasing order of the  $S_n$  quadrature, approximately  $\propto n^{-2}$ . At the same time, if we fix the accuracy threshold, the minimum acceptable value of  $n$  increases in direct proportion to the aspect ratio  $R_2/R_1$ .

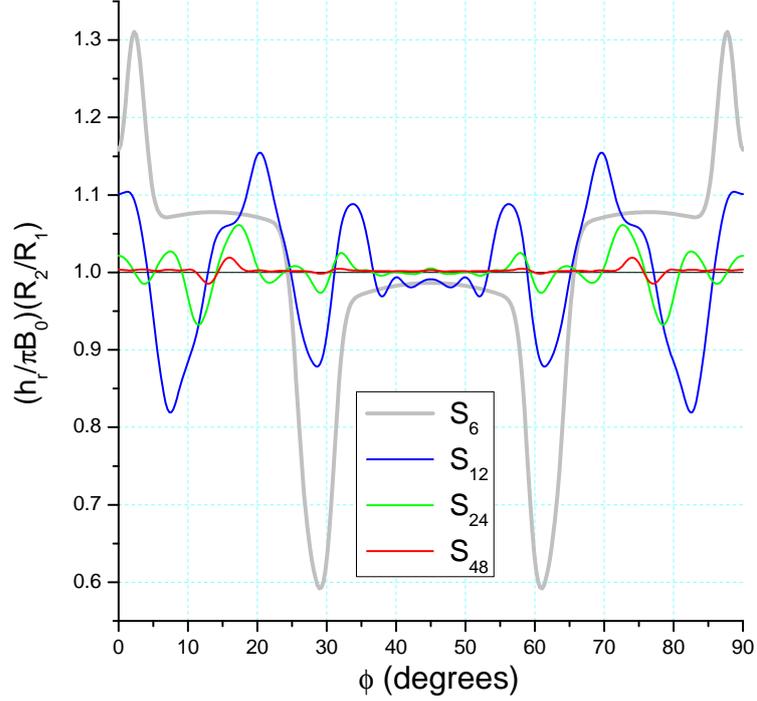


FIG. 10.13: Distribution of the normalized radiation flux along the outer boundary  $r = R_2$  of the cylindrical cavity at  $R_1 < r < R_2$ , obtained for  $R_2/R_1 = 4$  with different  $S_n$  quadratures on a quasi-square grid with  $n_h = 320$ .

#### 4. Thermal emission from a sphere ( $rz$ geometry)

##### 1. Analytical solution

In this 1D test problem we consider radiation transport in a sphere  $0 < r < R_0$  with a prescribed profile  $B(r)$  of the frequency-integrated source function  $B$ , where

$$r = \sqrt{R^2 + Z^2} \quad (10.39)$$

is the spherical radius. The sphere is assumed to be a grey medium with an absorption coefficient  $k$ , which is constant over the entire spherical volume. Then the frequency-integrated radiation intensity is a function of only two variables,  $I = I(r, \mu)$ , where

$$\mu = \cos \theta, \quad (10.40)$$

and  $\theta$  is the local polar angle between the photon propagation direction  $\vec{\Omega}$  and the local normal to the spherical surface; see Fig. 10.15. Without loss of generality, we can assume that the observation point  $O$  at radius  $r$  lies on the rotation axis  $Z$ , as is shown in Fig. 10.15.

The transfer equation in this case takes the form

$$\mu \frac{\partial I}{\partial r} + \frac{1 - \mu^2}{r} \frac{\partial I}{\partial \mu} = k(B - I). \quad (10.41)$$

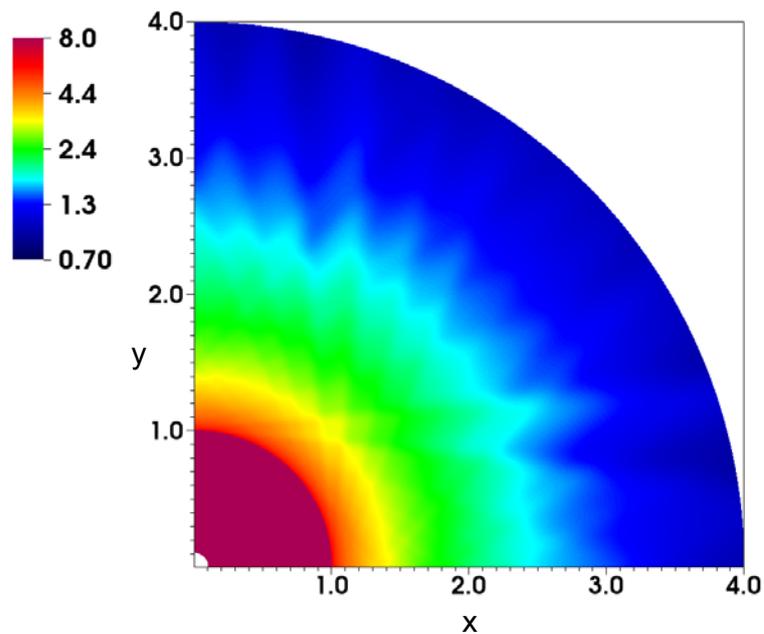


FIG. 10.14: 2D color plot of the integral radiation intensity  $U_r = U_r(r, \phi)$  in a cylindrical configuration with a vacuum cavity at  $R_1 < r < R_2$ , obtained for  $R_2/R_1 = 4$  with the  $S_{24}$  method on a quasi-square grid with  $n_g = 160$ .

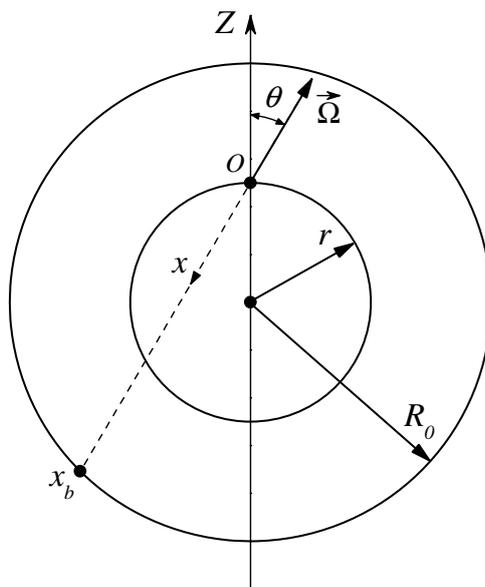


FIG. 10.15: Radiation transport in a spherical volume  $r \leq R_0$ . Radiation intensity  $I = I(r, \mu)$  is obtained by integrating back along the characteristic  $x_b \leq x \leq 0$ , starting at the observation point  $O$ .

Its solution is given by

$$I(r, \mu) = k \int_0^{x_b} B(r') e^{-kx} dx, \quad (10.42)$$

where

$$r' = r'(r, \mu, x) = \sqrt{r^2 - 2r\mu x + x^2}, \quad (10.43)$$

$$x_b = x_b(r, \mu) = r\mu + \sqrt{R_0^2 - r^2(1 - \mu^2)}. \quad (10.44)$$

The net volume-specific power of radiative heating is given by the expression

$$Q_r(r) = 2\pi k \int_{-1}^{+1} [I(r, \mu) - B(r)] d\mu. \quad (10.45)$$

Note that in terms of the cylindrical coordinates  $(R, Z)$  the ‘‘spherical’’ cosine  $\mu$  is given by

$$\mu = \frac{R\Omega_R + Z\Omega_Z}{\sqrt{R^2 + Z^2}}, \quad (10.46)$$

where  $\Omega_R$  and  $\Omega_Z$  are the projections of the unit propagation vector  $\vec{\Omega}$  onto the  $R$ - and  $Z$ -axis.

The outward bound energy flux  $h_r$  [erg cm<sup>-2</sup> s<sup>-1</sup>] at  $r = R_0$  is calculated as

$$h_r = 2\pi \int_0^1 \mu I(R_0, \mu) d\mu. \quad (10.47)$$

The total (i.e. integrated over the entire sphere surface) net radiative heating power (per radian of the global azimuth angle  $\phi$ ) due to the outgoing flux is given by

$$\mathcal{H}_r = -\frac{4\pi R_0^2 h_r}{2\pi} = -4\pi R_0^2 \int_0^1 \mu I(R_0, \mu) d\mu. \quad (10.48)$$

## 2. Radiative cooling of an isothermal sphere

*a. Exact results.* For a uniform isothermal sphere with  $B(r) = B_0 = \text{const}$  all the relevant quantities can be calculated analytically. The radiation intensity is given by

$$I(r, \mu) = B_0 [1 - \exp(-\tau_b)], \quad (10.49)$$

where

$$\tau_b = \tau_b(\tau, \mu) = \tau\mu + \sqrt{\tau_0^2 - \tau^2(1 - \mu^2)}, \quad (10.50)$$

$$\tau = kr, \quad \tau_0 = kR_0. \quad (10.51)$$

The net volume-specific heating power is calculated as

$$\begin{aligned}
 Q_r(r) &= -2\pi B_0 k \int_{-1}^{+1} \exp(-\tau_b) d\mu = \\
 &= -\frac{\pi B_0}{r} \left\{ [(1 + \tau_0 + \tau)e^\tau - (1 + \tau_0 - \tau)e^{-\tau}] e^{-\tau_0} + \right. \\
 &\quad \left. (\tau_0^2 - \tau^2) [E_1(\tau_0 + \tau) - E_1(\tau_0 - \tau)] \right\}, \tag{10.52}
 \end{aligned}$$

where  $E_1(x) = \int_x^\infty t^{-1} e^{-t} dt$  is the integral exponent. In particular, for the center of the sphere we obtain

$$Q_r(0) = -4\pi B_0 k e^{-\tau_0}. \tag{10.53}$$

The outgoing energy flux at  $r = R_0$  is

$$h_r = 2\pi \int_0^1 \mu I d\mu = \pi B_0 \left[ 1 - \frac{1 - (1 + 2\tau_0)e^{-2\tau_0}}{2\tau_0^2} \right]. \tag{10.54}$$

From this expression one readily obtains the total (i.e. integrated over the total spherical volume) net radiative heating power of an isothermal sphere per radian of the azimuth angle  $\phi$

$$W_r \equiv \frac{4\pi}{2\pi} \int_0^{R_0} Q_r r^2 dr = \mathcal{H}_r = -2\pi B_0 R_0^2 \left[ 1 - \frac{1 - (1 + 2\tau_0)e^{-2\tau_0}}{2\tau_0^2} \right]. \tag{10.55}$$

*b. Numerical results.* All the numerical test runs were performed for  $R_0 = B_0 = 1$  in the  $rz$ -mode of the RALEF code with `iradial = 1`. The structure of the mesh used in the simulations (constructed with the mesh option `IGEOM = 52`) is shown in Fig. 10.16. Individual runs differed in the values of the total number  $n_g$  of the mesh cells along the spherical radius  $r$ , of the absorption coefficient  $k$ , and of the order  $n$  of the  $S_n$  angular quadrature. For the boundary condition the default option `IRADDFBC = 0` (zero incident intensity) and `IRADBBC = 1` (reflective boundary condition) was used.

Figure 10.17 shows the calculated dependence of the total volumetric,

$$-W_r^{(n)} = -\sum_i W_i^r, \tag{10.56}$$

and surface,

$$-\mathcal{H}_r^{(n)} = -(\mathcal{H}_{2,2}^r + \mathcal{H}_{2,3}^r + \mathcal{H}_{2,4}^r), \tag{10.57}$$

radiative cooling rates (per radian of the azimuth angle) on the optical thickness  $\tau_0$ . The lower index (*ib, iblk*) by the integrated flux  $\mathcal{H}_{ib,iblk}^r$ , defined in Eq. (4.153), indicates the edge number *ib* and the block number *iblk* for which it was calculated. Table 10.8 lists the relative errors  $\delta(W_r)$  and  $\delta(\mathcal{H}_r)$  [as defined by Eq. (10.1)] for the corresponding radiative heating rates calculated on a finer grid with  $n_g = 40$ .

On the total, the results obtained in this test are very similar to those calculated for a planar slab and discussed in section 10.2.1. For  $\tau_0 \gg 1$ , when the optical thickness of individual mesh cells near the outer boundary becomes larger than unity, the error  $\delta W_r$ ,

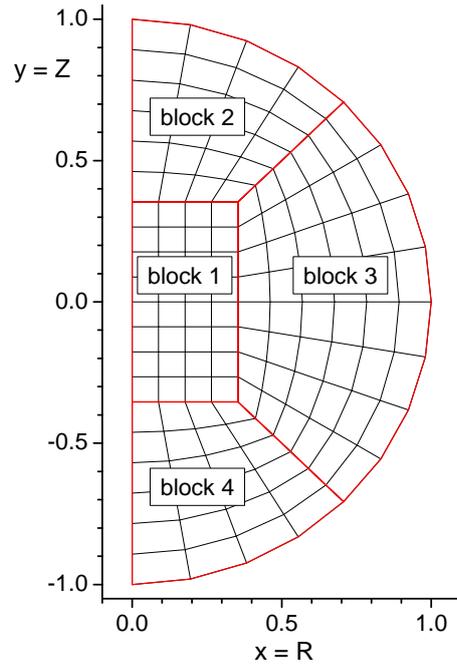


FIG. 10.16: Numerical grid with  $n_g = 10$  cells along the radial direction used to simulate radiation transfer in a spherical region of radius  $R_0 = 1$ . The rotation axis is along the vertical  $y = Z$  axis, while the horizontal  $x = R$  axis is the cylindrical radius  $R$ .

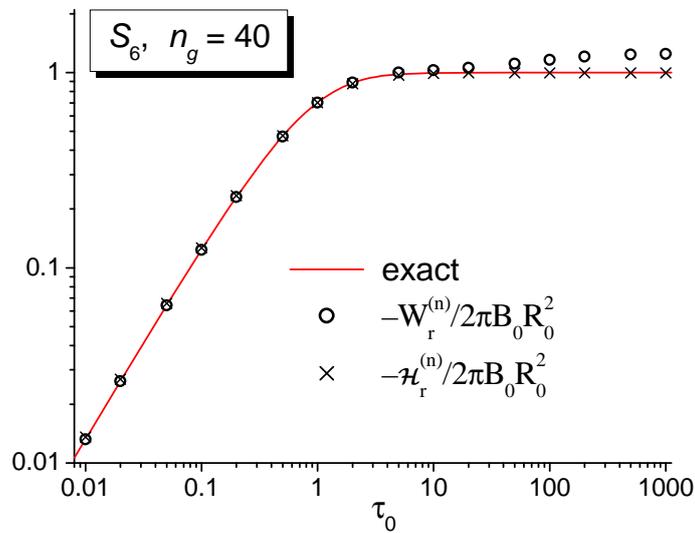


FIG. 10.17: Numerically calculated the volumetric,  $-W_r^{(n)}$ , and the surface,  $-\mathcal{H}_r^{(n)}$ , radiative cooling powers (per radian of the global azimuth angle) of a uniform sphere as functions of its optical thickness  $\tau_0$ . Numerical results, obtained with the  $S_6$  method on a grid from Fig. 10.16 with  $n_g = 40$  are compared with the exact formula (10.55).

in the volumetric heating rate  $W_r$  significantly increases, and for  $\tau_0 \rightarrow \infty$  saturates at a finite level (some 25% in our case) which depends on the aspect ratio of the boundary cells.

TABLE 10.8: Relative errors for  $W_r$  and  $\mathcal{H}_r$ , calculated with the  $S_6$  method on a grid of Fig. 10.16 with  $n_g = 40$  for different values of the optical thickness  $\tau_0$ .

$\tau_0$	0.01	0.1	1.0	10	100	$10^8$
$\delta(W_r)$	-0.00073	-0.0019	-0.0039	0.035	0.163	0.257
$\delta(\mathcal{H}_r)$	0.025	0.021	-0.0032	-0.0061	-0.0016	-0.0016

This error stems from the linear interpolation of the residual intensity  $F = I - B$  along the edges of optically thick cells. The relative difference between the values of  $-W_r$  and  $-\mathcal{H}_r$  in Fig. 10.17 at  $\tau_0 \gg 1$  gives a feeling for numerical errors in the energy balance that may be generated by our algorithm at interfaces with a sharp transition from an optically thin to an optically thick region.

3. Sphere with a cosine  $B$ -profile: optically thin mesh

Here we assume that the radial profile of the source function is given by

$$B(r) = B_0 \cos\left(\frac{\pi r}{2 R_0}\right). \tag{10.58}$$

The simulations were performed with the values  $R_0 = 1$ ,  $B_0 = 1$  on a randomized grid, shown in Fig. 10.18 for  $n_g = 10$  cells along the radial direction, in the  $rz$ -mode with `iradial` = 1. In all the test runs the exact values of the source function (10.58) were assigned at cell centers by loading the corresponding values of the cell-centered matter temperature  $T = (\pi B / \sigma_{SB})^{1/4}$ . The boundary condition was assigned with the `IRADFBC` = 0 (zero incident intensity), `IRADBCC` = 2 (fixed boundary  $B$ -temperature equal to zero) option.

Tables 10.9 and 10.10 list the  $L_2$ -norm errors  $\delta_{L_2}$  [as defined by Eq. (10.2)] for  $B_i$  (the source function at cell vertices),  $B'_{i,L}$  [the directional  $\tau$ -derivative of the source function (4.65) at cell vertices], the radiation intensity  $I_{i,L}$  at cell vertices, and the volume-specific net heating power  $Q_i^r = W_i^r / V_i$  at cell centers calculated with different numbers  $n_g$  of mesh cells in radial direction and different  $S_n$  order  $n = 2k_\Omega$ . For  $I$  and  $Q_r$  also the supremum-norm errors  $\delta_s$  [defined in Eq.(10.3)] were calculated, as well as the two relative errors  $\delta(W_r)$  and  $\delta(\mathcal{H}_r)$  for the total radiative heating power  $W_r$ .

The first row in Table 10.9 demonstrates that the bilinear interpolation of the source function  $B$  ensures the second-order accuracy for the vertex-centered values of  $B$ : the error  $\delta_{L_2}(B_v)$  decreases in proportion to  $n_g^{-2}$  with the increasing  $n_g$ . The error in the  $\tau$ -derivative  $B' = dB/d\tau$  is also determined exclusively by the finesse of the grid and decreases roughly as  $n_g^{-1.5}$  with the increasing  $n_g$ . Thus, with respect to these two quantities we observe a good convergence as the spatial mesh is refined even when the  $S_n$  order is fixed.

As might be expected, this however is not the case for the radiation intensity  $I$  and the heating rates  $Q_r$ ,  $W_r$ ,  $\mathcal{H}_r$ . Similar to the planar case in the  $xy$  geometry, at large  $n_g$  the errors for  $Q_r$ ,  $W_r$  and  $\mathcal{H}_r$  saturate (though at a comfortable level of a fraction of a percent, except for a  $\approx 2\%$  error for  $\mathcal{H}_r$ ) because of the fixed order of the angular quadrature and, in addition, because in the  $rz$  case one has to perform interpolation along the angular

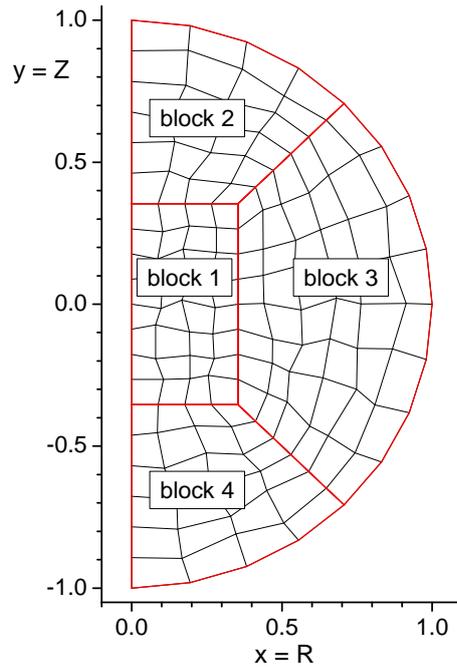


FIG. 10.18: Randomized numerical grid with  $n_g = 10$  cells along the radial direction in a spherical region of radius  $R_0 = 1$ . The rotation axis is along the vertical  $y = Z$  axis, while the horizontal  $x = R$  axis is the cylindrical radius  $R$ .

TABLE 10.9: Table of errors for a series of runs with the  $S_n$  quadrature fixed at  $S_6$  and different numbers of cells in radial direction  $n_g$  for a sphere with the cosine  $B$ -profile and absorption coefficient  $k = 1$ , i.e. with the radial optical thickness  $kR_0 = 1$ .

$n_g/S_n$	$10/S_6$	$20/S_6$	$40/S_6$	$80/S_6$	$160/S_6$
$\delta_{L_2}(B_v)$	0.0082	0.0022	0.00056	0.00014	0.000036
$\delta_{L_2}(B')$	0.069	0.027	0.0097	0.0038	0.0015
$\delta_{L_2}(I)$	0.018	0.0074	0.0046	0.0049	0.0054
$\delta_s(I)$	0.032	0.015	0.019	0.022	0.025
$\delta_{L_2}(Q_r)$	0.023	0.0063	0.0034	0.0034	0.0037
$\delta_s(Q_r)$	0.033	0.017	0.0071	0.0067	0.0067
$\delta(W_r)$	-0.024	-0.0071	-0.0063	-0.0080	-0.0094
$\delta(\mathcal{H}_r)$	-0.034	-0.0029	0.012	0.019	0.023

coordinate  $\Omega_R$  for the initial  $F_O$  values when integrating along a short characteristic; see section 4.3.2. In addition, unlike in the  $xy$  case, we loose convergence for the radiation intensity  $I$  (hence, for the residual intensity  $F = I - B$ ) as  $n_g \rightarrow \infty$  and  $S_n$  remains fixed — again, because of the necessity to interpolate along the angular coordinate  $\Omega_R$ .

TABLE 10.10: Table of errors for a series of runs where both the  $S_n$  order and the number of cells  $n_g$  increase. The source function  $B$  has a cosine radial profile, the absorption coefficient is fixed at  $k = 1$ , i.e. the radial optical thickness of the considered sphere is  $kR_0 = 1$ .

$n_g/S_n$	10/ $S_6$	20/ $S_{12}$	40/ $S_{24}$	80/ $S_{48}$	160/ $S_{96}$	320/ $S_{192}$
$\delta_{L_2}(I)$	0.018	0.0074	0.0033	0.0015	0.00073	0.00036
$\delta_s(I)$	0.032	0.015	0.0069	0.0039	0.0017	0.00082
$\delta_{L_2}(Q_r)$	0.023	0.0057	0.0018	0.00086	0.00047	0.00025
$\delta_s(Q_r)$	0.033	0.017	0.0059	0.0030	0.0012	0.00073
$\delta(W_r)$	-0.024	-0.0049	-0.00013	0.0060	0.0047	0.00028
$\delta(\mathcal{H}_r)$	-0.034	-0.013	-0.0059	-0.0029	-0.0015	-0.00073

Table 10.10 in its turn demonstrates that, once we begin to increase the  $S_n$  order in pace with  $n_g$ , we recover the first-order convergence for all the relevant quantities. The latter indicates that our scheme for radiation transport has the overall first order of accuracy on optically thin meshes. Note that the errors for  $B_v$  and  $B'_\tau$  practically do not depend on the  $S_n$  order and can be inferred from Table 10.9.

4. Sphere with a cosine  $B$ -profile: optically thick mesh and the diffusion limit

a. Linear inter-node interpolation (4.80) of the source function  $B$ . Here all the simulation parameters were the same as in the previous subsection except that in the process of variation of  $n_g$  and  $S_n$  the radial optical thickness of numerical cells was kept constant and equal to  $\tau_c = 100$ . Under this condition the exact solution of the transfer equation approaches the solution of the diffusion equation everywhere except near the boundary within a narrow outer layer with the optical thickness of  $\tau \simeq 1$ . Therefore, to test how our numerical scheme behaves in the diffusion limit, we excluded the outermost boundary cell layer when calculating the errors for the volumetric heating rate  $Q_r$ , having designated thus calculated errors as  $\delta_{L_2}(Q'_r)$  and  $\delta_s(Q'_r)$ . The results obtained are presented in Table 10.11.

As is seen from this Table, we observe fair convergence (roughly of the order of 1.4) to the exact solution — hence to the diffusion limit — for  $B'$  and the residual intensity  $F$  but not for the energy coupling rate  $Q_r$ . Nevertheless, since the  $L_2$ -norm error  $\delta_{L_2}(Q'_r)$  saturates at a practically quite acceptable level of  $\simeq 3\%$ , we conclude that from the practical point of view we do recover the diffusion limit (on moderately distorted meshes like the one shown in Fig. 10.18) even with the linear inter-node interpolation scheme (4.80) for the source function.

5. Thermal emission from a narrow opaque core ( $xy$  and  $rz$  geometries)

In this 1D problem we consider either a cylindrical or a spherical volume  $0 \leq r \leq R_0$  of radius  $R_0$ , where the source function

$$B(r) = B_0 \tag{10.59}$$

TABLE 10.11: Table of errors for a series of runs where the radial optical thickness of mesh cells was fixed at  $\tau_c = 100$  and both  $S_n$  and  $n_g$  varied. The absorption coefficient  $k = \tau_c n_g$  changed in proportion to the radial cell number  $n_g$  to keep  $\tau_c$  constant.

$n_g/S_n$	10/ $S_6$	20/ $S_6$	40/ $S_6$	80/ $S_6$	160/ $S_6$	20/ $S_{12}$	40/ $S_{24}$	80/ $S_{48}$	160/ $S_{96}$
$\delta_{L_2}(B')$	0.061	0.023	0.0084	0.0033	0.0013	0.025	0.010	0.0038	0.0015
$\delta_{L_2}(F)$	0.049	0.017	0.0066	0.0027	0.0012	0.020	0.0087	0.0033	0.0013
$\delta_s(F)$	0.248	0.101	0.031	0.016	0.0079	0.292	0.173	0.145	0.114
$\delta_{L_2}(Q'_r)$	0.050	0.034	0.029	0.029	0.028	0.034	0.030	0.029	0.027
$\delta_s(Q'_r)$	0.135	0.126	0.155	0.161	0.182	0.134	0.123	0.163	0.186

is constant and the radiation absorption coefficient  $k(r)$  has a Lorentzian profile

$$k(r) = \frac{k_0}{1 + (r/r_*)^2}, \tag{10.60}$$

with  $r_*$  being the effective radius of the “opaque” core;  $r = \sqrt{x_1^2 + x_2^2}$  is either spherical or cylindrical radius. Constants  $B_0$ ,  $k_0$  and  $r_*$  are independent free parameters of the problem. Instead of  $k_0$  one can adopt the radial optical thickness

$$\tau_0 = \int_0^{R_0} k dr = k_0 r_* \arctan \left( \frac{R_0}{r_*} \right) \tag{10.61}$$

of the considered sphere (cylinder) as an alternative independent free parameter of the problem.

This particular test problem is of special interest for our non-conservative numerical scheme of radiation transfer, which in its discretized form does not obey the divergence (Gauss-Ostrogradsky) theorem. As a consequence, we always calculate two different values of the net radiative heating power of any closed volume, namely, the *volumetric* heating power  $W_r^{(n)}$ , obtained by summing up the heating powers  $W_i^r$  of all mesh cells in this volume, and the *fluxing* heating power  $\mathcal{H}_r^{(n)}$ , obtained by integrating the outgoing flux  $h_r$  over the outer surface of the simulated volume. The difference between the two provides a certain figure of merit for the overall accuracy of the radiation transport treatment. In the present test, once we choose  $r_* \ll R_0$  and  $\tau_0 \simeq 1$ , the regions of the main contribution to the values of  $W_r^{(n)}$  and  $\mathcal{H}_r^{(n)}$  happen to be well separated in space: while the main contribution to  $W_r^{(n)}$  comes from the central region  $r \lesssim r_*$  of strong coupling between matter and radiation, the fluxing quantity  $\mathcal{H}_r^{(n)}$  is calculated by using the radiation intensity along the outer boundary  $r = R_0$ . In such a situation we can probe directly the practical worthiness of the accuracy assessment from the difference  $W_r^{(n)} - \mathcal{H}_r^{(n)}$  by comparing the numerically calculated values  $W_r^{(n)}$  and  $\mathcal{H}_r^{(n)}$  between themselves, and with the known exact value  $W_r = \mathcal{H}_r$ .

1. Analytical solution for a sphere in the  $rz$ -geometry

Below we use the same notation as in the subsection 10.4.1. The exact solution for the radiation intensity is given by

$$I(r, \mu) = B_0(1 - e^{-\tau}), \tag{10.62}$$

$$\tau(r, \mu) = \frac{k_0 r_*^2}{a} \left[ \arctan\left(\frac{r\mu}{a}\right) + \arctan\left(\frac{\sqrt{R_0^2 - r^2(1 - \mu^2)}}{a}\right) \right], \tag{10.63}$$

$$a(r, \mu) = \sqrt{r_*^2 + r^2(1 - \mu^2)}. \tag{10.64}$$

Here

$$\mu = \frac{R\Omega_R + Z\Omega_Z}{\sqrt{R^2 + Z^2}} = \frac{x_1\Omega_{x1} + x_2\Omega_{x2}}{\sqrt{x_1^2 + x_2^2}} \tag{10.65}$$

is the cosine of the local polar angle  $\theta$ , and  $\tau$  is the optical length between the observation point  $O$  and the backward exit point  $x_b$  — as is shown in Fig. 10.15. The exact volumetric radiative heating rate is

$$Q_r(r) = -\frac{2\pi k_0 B_0}{1 + (r/r_*)^2} \int_{-1}^{+1} e^{-\tau} d\mu. \tag{10.66}$$

For the outgoing radiative energy flux across the sphere surface we find

$$h_r = 2\pi \int_0^1 \mu I(R_0, \mu) d\mu = 2\pi B_0 \int_0^1 \mu \{1 - \exp[-\tau(R_0, \mu)]\} d\mu. \tag{10.67}$$

By virtue of the divergence theorem the total radiative heating power of the considered sphere (per radian of the global azimuth angle  $\phi$ ) can be expressed as

$$W_r = 2 \int_0^{R_0} Q_r(r) r^2 dr = \mathcal{H}_r = -2R_0^2 h_r. \tag{10.68}$$

For numerical checks, certain particular values of the above listed quantities might be of interest

$$\tau(0, \mu) = \tau_0, \quad \tau(R_0, 1) = 2\tau_0, \tag{10.69}$$

$$Q_r(0) = -4\pi k_0 B_0 e^{-\tau_0} = -\frac{4\pi B_0 \tau_0 e^{-\tau_0}}{r_* \arctan(R_0/r_*)}, \tag{10.70}$$

$$Q_r(R_0) \xrightarrow{R_0 \gg r_*} -\frac{8B_0 \tau_0 r_*}{R_0^2}. \tag{10.71}$$

2. Analytical solution for a cylinder in the  $xy$ -geometry

Now, if we consider an infinite cylinder where nothing depends on coordinate  $Z$  along the rotation axis (see Fig. 10.19), we obtain the following solution

$$I(r, \theta, \varphi) = B_0(1 - e^{-\tau}), \tag{10.72}$$

$$\tau(r, \theta, \varphi) = \frac{k_0 r_*^2}{a \sin \theta} \left[ \arctan \left( \frac{r \cos \varphi}{a} \right) + \arctan \left( \frac{\sqrt{R_0^2 - r^2 \sin^2 \varphi}}{a} \right) \right], \tag{10.73}$$

$$a(r, \varphi) = \sqrt{r_*^2 + r^2 \sin^2 \varphi}, \tag{10.74}$$

$$\cos \theta = \Omega_z, \quad \sin \theta = \sqrt{1 - \Omega_z^2} = \sqrt{\Omega_x^2 + \Omega_y^2}, \tag{10.75}$$

$$r = \sqrt{x^2 + y^2}, \quad \cos \varphi = \frac{x \Omega_x + y \Omega_y}{r \sin \theta}. \tag{10.76}$$

In the Cartesian cylindrical geometry we use the usual simplified notation  $x_1 = x$ ,  $x_2 = y$ .

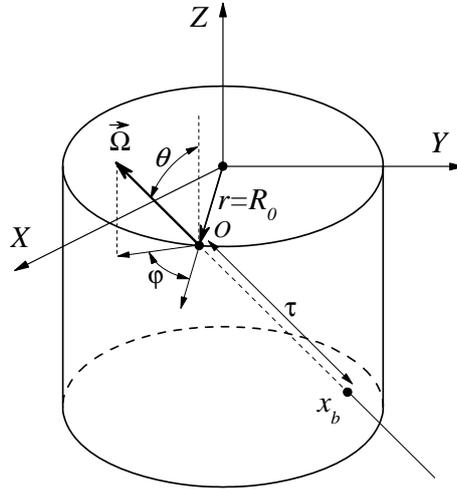


FIG. 10.19: Radiation transport in a cylindrical volume  $r \leq R_0$ . Radiation intensity  $I = I(r, \theta, \varphi)$  is obtained by integrating back along the characteristic  $x_b \leq x \leq 0$ , starting at the observation point  $O$ .

The exact volumetric radiative heating rate  $Q_r(r)$  and the outgoing radiative energy flux  $h_r$  across the outer cylinder surface are given by the double integrals

$$Q_r(r) = -\frac{4k_0 B_0}{1 + (r/r_*)^2} \int_0^1 d \cos \theta \int_0^\pi e^{-\tau} d\varphi, \tag{10.77}$$

$$h_r = 4B_0 \int_0^1 \sin \theta d \cos \theta \int_0^\pi (1 - e^{-\tau}) \cos \varphi d\varphi. \tag{10.78}$$

Numerical integration in the above formulae is somewhat simplified after we change to a

different pair of angular variables

$$\Omega_r = \frac{x\Omega_x + y\Omega_y}{\sqrt{x^2 + y^2}} = \sin \theta \cos \varphi, \quad (10.79)$$

$$\cos \psi = \sqrt{\frac{1 - \Omega_z^2 - \Omega_r^2}{1 - \Omega_r^2}} = \frac{\sin \theta \sin \varphi}{\sqrt{\sin^2 \varphi + \cos^2 \theta \cos^2 \varphi}}, \quad (10.80)$$

in terms of which we get

$$Q_r(r) = -\frac{4k_0B_0}{1 + (r/r_*)^2} \int_{-1}^1 d\Omega_r \int_0^{\pi/2} e^{-\tau} d\psi, \quad (10.81)$$

$$h_r = 4B_0 \int_0^1 \Omega_r d\Omega_r \int_0^{\pi/2} (1 - e^{-\tau}) d\psi. \quad (10.82)$$

Again, by virtue of the divergence theorem the total radiative heating power of the considered cylinder (per unit length along the  $Z$ -axis) is given by

$$W_r = 2\pi \int_0^{R_0} Q_r(r) r dr = \mathcal{H}_r = -2\pi R_0 h_r. \quad (10.83)$$

The particular values on the cylinder axis  $r = 0$  are

$$Q_r(0) = -4\pi k_0 B_0 \int_0^{\pi/2} \exp\left(-\frac{\tau_0}{\sin \theta}\right) \sin \theta d\theta, \quad \tau(0, \theta, \varphi) = \frac{\tau_0}{\sin \theta}. \quad (10.84)$$

### 3. Numerical results

Numerical simulations were performed with  $R_0 = 1$ ,  $B_0 = 1$  on a randomized grid, shown in Fig. 10.18 for  $n_g = 10$  cells in radial direction. In all test runs the exact values of the source function  $B = 1$  were assigned at cell centers by loading the corresponding values of the cell-centered matter temperature  $T = (\pi B / \sigma_{SB})^{1/4}$ . The effective radius of the opaque core was fixed at a value  $r_* = 0.05$ , the total radial optical thickness at  $\tau_0 = 1$ . The boundary condition was assigned by setting  $\text{IRAD}FBC = 0$  (zero incident intensity) and  $\text{IRAD}BBC = 1$  (a reflective boundary for the  $B$ -temperatures).

Table 10.12 lists the  $L_2$ -norm and the supremum-norm errors for the radiation intensity  $I$  and the volume-specific radiative heating rate  $Q_r$ , as well as the relative errors for the volumetric,  $W_r$ , and the fluxing,  $\mathcal{H}_r$  total net radiative heating powers calculated in the  $xy$ -geometry for a cylindrical volume. Table 10.13 presents the same quantities calculated in the  $rz$ -geometry for a spherical volume. These errors were calculated in two series of numerical runs: one with a fixed angular quadrature  $S_6$  and consecutively increasing number  $n_g$  of cells in radial direction, the other with both the  $S_n$  order and  $n_g$  increasing in geometric progression. When analyzing the behavior of the numerical errors along these series, one has

TABLE 10.12: Table of errors calculated for a cylinder ( $\text{iradial} = 0$ ) with a narrow opaque core  $r_* = 0.05R_0$  in two series of runs: one where  $S_n = S_6$  is fixed and only the number of grid cells  $n_g$  increases, and the other with both  $S_n$  and  $n_g$  increasing.

$n_r/S_n$	20/ $S_6$	40/ $S_6$	80/ $S_6$	160/ $S_6$	320/ $S_6$	640/ $S_6$	20/ $S_{12}$	40/ $S_{24}$	80/ $S_{48}$	160/ $S_{96}$	320/ $S_{192}$
$\delta_{L_2}(I)$	0.056	0.033	0.018	0.0099	0.0052	0.0027	0.056	0.033	0.018	0.010	0.0052
$\delta_s(I)$	0.202	0.131	0.084	0.050	0.028	0.015	0.190	0.133	0.088	0.052	0.059
$\delta_{L_2}(Q_r)$	0.053	0.025	0.011	0.0062	0.0041	0.0035	0.052	0.025	0.011	0.0055	0.0026
$\delta_s(Q_r)$	0.092	0.059	0.028	0.014	0.0070	0.0036	0.090	0.059	0.028	0.014	0.0065
$\delta(W_r)$	0.012	0.0059	0.0033	0.0023	0.0017	0.0015	0.010	0.0046	0.0021	0.0011	0.00053
$\delta(\mathcal{H}_r)$	0.015	0.0070	0.0043	0.0025	0.0018	0.0015	0.0089	0.0035	0.0017	0.00082	0.00038

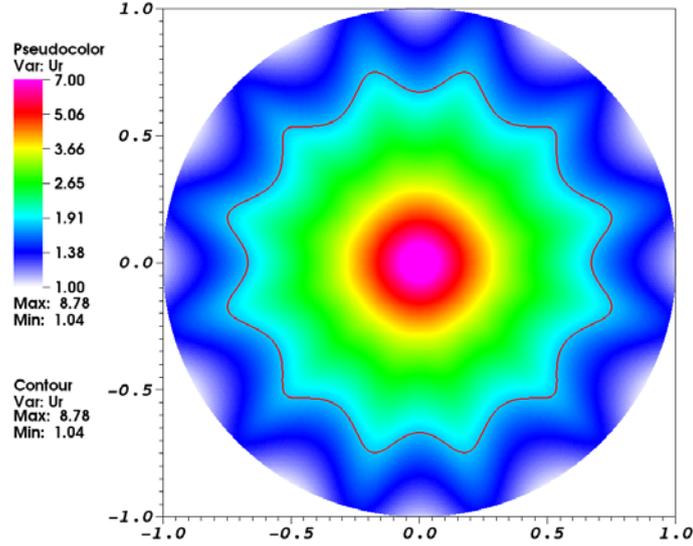


FIG. 10.20: 2D color-contour plot of the integral intensity  $U_r = \int_{4\pi} I d\vec{\Omega}$  calculated for a cylinder ( $\text{iradial} = 0$ ) with a narrow opaque core  $r_* = 0.05R_0$  with the  $S_6$  angular quadrature and  $n_g = 160$  mesh cells along the radius.

to keep in mind that, because the radiation field is mainly generated in the narrow opaque core  $r \lesssim r_*$ , numerical results for this problem exhibit a pronounced ray-effect, displayed in Figs. 10.20 and 10.21 for the total integral radiation intensity  $U_r$ .

The data in Table 10.12 clearly demonstrate that in the  $xy$  case all the numerical errors depend primarily on how well the opaque core is resolved, i.e. on the number of grid cells along the two spatial directions (represented by the parameter  $n_g$ ). For the radiation intensity  $I$  we have a clear first-order convergence independent of the  $S_n$ -quadrature order  $n$ . This is not surprising because in the Cartesian  $xy$  geometry the photons propagate along fixed global directions and no interpolation in the  $\Omega$ -space is involved. However, because of the ray effect, the errors for the energy coupling quantities  $Q_r$ ,  $W_r$  and  $\mathcal{H}_r$ , calculated in

the  $S_6$  series, saturate — though at a relatively low level of  $\simeq (1-3) \times 10^{-3}$  due to a high accuracy of the  $S_6$  quadrature for smooth functions of  $\vec{\Omega}$  — and the true convergence is observed only in the second series, where both the  $n$  and the  $n_g$  increase.

Since our numerical scheme does not obey the divergence theorem, we would like to use the relative discrepancy

$$\delta_G = \frac{W_r^{(n)} - \mathcal{H}_r^{(n)}}{0.5 |W_r^{(n)} + \mathcal{H}_r^{(n)}|} \tag{10.85}$$

between the numerically calculated volumetric,  $W_r^{(n)}$ , and fluxing,  $\mathcal{H}_r^{(n)}$ , net total heating powers as an indication for the numerical error in  $W_r$ . (Note that it is the value of  $W_r$  and not of  $\mathcal{H}_r$  which represents the energy coupling with the fluid.) Because  $|\delta(W_r)| \ll 1$  and  $|\delta(\mathcal{H}_r)| \ll 1$ , the values of  $\delta_G$  can be easily evaluated from Table 10.12 as  $\delta_G = \delta(W_r) - \delta(\mathcal{H}_r)$ . Such evaluation reveals that we always have  $|\delta_G| \ll \max\{|\delta(W_r)|, |\delta(\mathcal{H}_r)|\}$ . The latter means that in the  $xy$  Cartesian case the accuracy criterion based on the value of  $\delta_G$  turns out to be too optimistic — typically by about a factor 3–10 in this particular test problem.

TABLE 10.13: Table of errors calculated for a sphere ( $\text{iradial} = 1$ ) with a narrow opaque core  $r_* = 0.05R_0$  in two series of runs: one with the  $S_n = S_6$  fixed and only the number of grid cells  $n_g$  increasing, the other with both the  $S_n$  order and the  $n_g$  increasing. The values in the parentheses (the second row) were calculated with the 1st-order interpolation along the component  $\Omega_R$  of the photon propagation vector  $\vec{\Omega}$ .

$n_r/S_n$	20/ $S_6$	40/ $S_6$	80/ $S_6$	160/ $S_6$	20/ $S_{12}$	40/ $S_{24}$	80/ $S_{48}$	160/ $S_{96}$
$\delta_{L_2}(I)$	0.090	0.082	0.081	0.083	0.064	0.034	0.017	0.0077
	(0.124)	(0.123)	(0.128)	(0.132)	(0.078)	(0.047)	(0.028)	(0.016)
$\delta_s(I)$	0.213	0.229	0.243	0.252	0.226	0.150	0.146	0.208
	(0.333)	(0.354)	(0.375)	(0.385)	(0.226)	(0.150)	(0.146)	(0.208)
$\delta_{L_2}(Q_r)$	0.064	0.030	0.012	0.0055	0.064	0.030	0.012	0.0053
	(0.063)	(0.029)	(0.012)	(0.079)	(0.064)	(0.030)	(0.012)	(0.0053)
$\delta_s(Q_r)$	0.107	0.066	0.030	0.014	0.104	0.069	0.031	0.014
$\delta(W_r)$	0.00089	-0.0030	-0.0046	-0.0054	0.0028	0.00063	0.00021	0.00011
	(-0.0028)	(-0.0078)	(-0.010)	(-0.012)	(0.00092)	(-0.00083)	(-0.00079)	(-0.00050)
$\delta(\mathcal{H}_r)$	0.127	0.140	0.149	0.153	0.072	0.038	0.015	0.0048
	(0.256)	(0.280)	(0.295)	(0.303)	(0.145)	(0.089)	(0.050)	(0.027)

In contrast to the Cartesian  $xy$ -case, calculation of the radiation field in the  $rz$ -geometry requires interpolation along the angular coordinate  $\Omega_R$ . As a consequence, we loose convergence at  $n_g \rightarrow \infty$  for the radiation intensity  $I$  when  $S_n$  is fixed (see the data in the first four columns of Table 10.13): with  $S_n = S_6$  the error  $\delta_{L_2}(I)$  stabilizes at a rather high level of  $\delta_{L_2}(I) \approx 10\%$ . On the other hand, as is demonstrated by Fig. 10.21, the amplitude of the ray effect is noticeably reduced as compared to the  $xy$  case due to the numerical diffusion

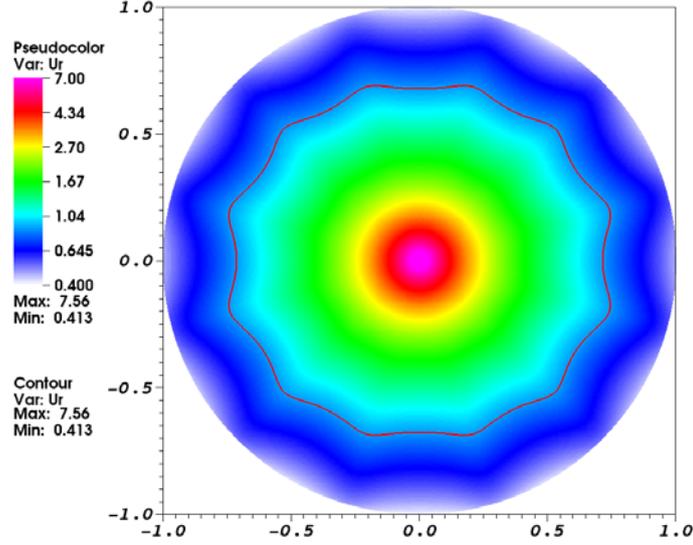


FIG. 10.21: 2D color-contour plot of the integral intensity  $U_r = \int_{4\pi} I d\vec{\Omega}$  calculated for a sphere ( $\text{iradial} = 1$ ) with a narrow opaque core  $r_* = 0.05R_0$  with the  $S_6$  angular quadrature and  $n_g = 160$  mesh cells along the radius.

in the angular space. Also, in the first series with  $S_n = S_6$  and  $n_g \rightarrow \infty$  the errors  $\delta_{L_2}(Q_r)$ ,  $\delta_s(Q_r)$  for the radiation-matter energy coupling rate  $Q_r$  saturate at levels that are even below those for the  $xy$ -case (due, apparently, to the reduced amplitude of the ray effect). Large difference between the saturated errors for the radiation intensity  $\delta_{L_2}(I)$  and the radiative heating rate  $\delta_{L_2}(Q_r)$  is explained by the fact that the main contribution to  $\delta_{L_2}(Q_r)$  comes from the narrow opaque core  $r \lesssim r_*$ , where the radiation intensity  $I(r, \vec{\Omega})$  has a much more uniform angular dependence than in the outer layers at  $r \gg r_*$ , which make a large contribution to  $\delta_{L_2}(I)$ .

Most striking, however, is the difference between the values of  $|\delta(W_r)|$  and  $|\delta(\mathcal{H}_r)|$  in Table 10.13. While the error  $|\delta(W_r)|$  for the total net radiative heating power in the  $S_n = S_6$  series saturates at a quite satisfactory level of  $|\delta(W_r)| \simeq 0.6\%$ , the error  $|\delta(\mathcal{H}_r)|$  for the fluxing quantity  $\mathcal{H}_r$  remains about a factor 30 higher. The latter implies that any judgement on the accuracy of radiation transport treatment based on the value of  $\delta_G \approx \delta(W_r) - \delta(\mathcal{H}_r)$  would be far too pessimistic. This remains true even in the second series of test runs, where both the  $S_n$  order and the grid cell number  $n_g$  are increased and the first-order convergence (at least for the  $L_2$ -norm) to the exact solution for all the relevant quantities is observed.

Note also that the supremum-norm error  $\delta_s(I)$  in the  $rz$ -case remains rather high and does not decrease as the  $S_n$  order and  $n_g$  are increased. This large error is calculated for a small fraction of vertices at large radii where the sharp dependence of  $I(r, \mu)$  on  $\mu$  is poorly reproduced by interpolation along the angular coordinate  $\Omega_R$ . Table 10.13 demonstrates also that introduction of the second-order interpolation along  $\Omega_R$  results in a significant reduction of the discrepancy  $W_r^{(n)} - \mathcal{H}_r^{(n)}$  in the discrete version of the divergence theorem.

In conclusion, the present test clearly demonstrates that in the  $rz$ -geometry large discrepancy  $W_r^{(n)} - \mathcal{H}_r^{(n)}$  between the volumetric and the fluxing total net radiative heating powers does not necessarily mean poor accuracy in the overall description of the radiation energy transport — especially (as is often the case in practice) when the outer boundary of

the simulated domain is far away from the inner region of strong coupling between matter and radiation.

**APPENDIX A: THE  $ES_n$  ANGULAR ORDINATES**

The main option for the  $S_n$  method in the RALEF code is based on the  $ES_n$  angular ordinates proposed by B.G. Carlson [21], which have an important advantage of being easily generated for any order  $n = 2k_\Omega$ ,  $k_\Omega = 1, 2, 3, \dots$ . The  $ES_n$  ordinates have a lower degree of symmetry than the classical  $S_n$  ordinates; its nodes are assigned equal weights

$$w_l = w_0 = \frac{\pi}{k_\Omega(k_\Omega + 1)} = \frac{\pi}{\bar{l}_{max}(\bar{l}_{max} + 1)} \tag{A.1}$$

and arranged in  $k_\Omega$  tiers (per octant) around the  $Z$ -axis, which plays a special role in the construction. For historical reasons and to agree with the notation of Ref. [35], used in the corresponding module SN\_NODES of the RALEF code, in this section we use an alternative system of notation which relates to the variables described in section 4.2.3 as

$$\bar{l} = m_Z, \quad \bar{m} = m_R, \quad \bar{l}_{max} = k_\Omega. \tag{A.2}$$

If we define

$$\mu = \cos \theta \tag{A.3}$$

to be the cosine of the polar angle  $\theta$  between the unit vector  $\vec{\Omega}$  and the  $Z$ -axis, and  $\phi$  to be the azimuth angle in the  $XY$ -plane with respect to the  $X$ -axis, then the  $N_\Omega = \bar{l}_{max}(\bar{l}_{max} + 1)/2$  primary  $ES_n$  nodes in the first octant  $\mu \in [0, 1]$ ,  $\phi \in [0, \pi/2]$  are identified by the pairs

$$\vec{\Omega}_{\bar{l}\bar{m}} = (\mu_{\bar{l}}, \phi_{\bar{l}\bar{m}}), \tag{A.4}$$

where the indices  $\bar{l}, \bar{m}$  run through

$$\bar{m} = 1, 2, \dots, \bar{l}_{max} - \bar{l} + 1, \quad \bar{l} = 1, 2, \dots, \bar{l}_{max}. \tag{A.5}$$

Note that, because the value of  $\mu$  in the  $rz$ -geometry is preserved along the photon path, the fact that the discrete values  $\mu_{\bar{l}}$  do not depend on index  $\bar{m}$  enables easy code parallelization with respect to independent Omega-blocks having different values of index  $\bar{l}$ .

Since the nodes  $\vec{\Omega}_{\bar{l}\bar{m}}$  of the  $ES_n$  quadrature represent unit-sphere elements with equal areas of  $w_0$ , one readily calculates the area of the  $\bar{l}$ -th tier around the  $Z$ -axis to be  $(\pi/2)w_{\bar{l}}$ , where

$$w_{\bar{l}} = \frac{2(\bar{l}_{max} - \bar{l} + 1)}{\bar{l}_{max}(\bar{l}_{max} + 1)}, \quad \sum_{\bar{l}=1}^{\bar{l}_{max}} w_{\bar{l}} = 1. \tag{A.6}$$

The  $\mu_{\bar{l}}$  nodes are chosen as

$$\mu_{\bar{l}} = \hat{\mu}_{\bar{l}} + f \cdot \mu_{\bar{l}-1/2}, \tag{A.7}$$

where

$$\hat{\mu}_{\bar{l}} = 1 - \frac{(\bar{l}_{max} - \bar{l} + 1)^2}{\bar{l}_{max}(\bar{l}_{max} + 1)} \tag{A.8}$$

is the mid-point of the  $\bar{l}$ -th tier,

$$\mu_{\bar{l}-1/2} = 1 - \frac{(\bar{l}_{max} - \bar{l} + 1)(\bar{l}_{max} - \bar{l} + 2)}{\bar{l}_{max}(\bar{l}_{max} + 1)} \tag{A.9}$$

is its lower boundary, and  $f$  is a fitting parameter. The  $\phi_{\bar{l}\bar{m}}$  nodes are chosen as

$$\phi_{\bar{l}\bar{m}} = \frac{\pi}{4} \left( \frac{2\bar{m} - 1}{\bar{l}_{max} - \bar{l} + 1} A_n + 1 - A_n \right), \quad (\text{A.10})$$

where  $A_n$  is another fitting parameter. Note that, for any fixed  $\bar{l}$ , the values of  $\phi_{\bar{l}\bar{m}}$  are equidistant except for the end intervals encompassing the values  $\phi = 0$  and  $\phi = \pi/2$ , whose lengths are stretched by a factor  $1 + (1 - A_n)(\bar{l}_{max} - \bar{l})$ .

The two fitting factors  $f$  and  $A_n$  are used to ensure the exact values of the two lower momenta

$$\int_0^1 d\mu \int_0^{\pi/2} \gamma^k d\phi \quad \text{with} \quad \gamma = \left\{ \begin{array}{l} \mu \\ \sqrt{1 - \mu^2} \sin \phi \\ \sqrt{1 - \mu^2} \cos \phi \end{array} \right\}, \quad k = 1, 2. \quad (\text{A.11})$$

Conditions (A.11) boil down to two independent equations

$$\sum_{\bar{l}=1}^{\bar{l}_{max}} w_{\bar{l}} \mu_{\bar{l}}^2 = \frac{1}{3}, \quad (\text{A.12})$$

$$w_0 \sum_{\bar{l}=1}^{\bar{l}_{max}} \sqrt{1 - \mu_{\bar{l}}^2} \sum_{\bar{m}=1}^{\bar{l}_{max} - \bar{l} + 1} \cos \phi_{\bar{l}\bar{m}} = \frac{\pi}{2} \sum_{\bar{l}=1}^{\bar{l}_{max}} w_{\bar{l}} \mu_{\bar{l}}, \quad (\text{A.13})$$

which yield the values of the two independent fitting parameters  $f$  and  $A_n$ . In particular, Eq. (A.12) yields

$$f = \frac{-b + \sqrt{b^2 - a(c - 1/3)}}{a}, \quad (\text{A.14})$$

where

$$a = \sum_{\bar{l}=1}^{\bar{l}_{max}} w_{\bar{l}} \hat{\mu}_{\bar{l}}^2, \quad b = \sum_{\bar{l}=1}^{\bar{l}_{max}} w_{\bar{l}} \hat{\mu}_{\bar{l}} \mu_{\bar{l}-1/2}, \quad c = \sum_{\bar{l}=1}^{\bar{l}_{max}} w_{\bar{l}} \hat{\mu}_{\bar{l}-1/2}^2. \quad (\text{A.15})$$

Equation (A.13) reduces to

$$\frac{\sqrt{2} \sin(\pi A_n/4)}{\bar{l}_{max}(\bar{l}_{max} + 1)} \sum_{\bar{l}=1}^{\bar{l}_{max}} \frac{\sqrt{1 - \mu_{\bar{l}}^2}}{\sin[\pi A_n/4(\bar{l}_{max} - \bar{l} + 1)]} = \sum_{\bar{l}=1}^{\bar{l}_{max}} w_{\bar{l}} \mu_{\bar{l}}. \quad (\text{A.16})$$

Table A.1 lists the values of parameters  $f$  and  $A_n$  calculated for a  $\bar{l}_{max} = 2, 3, 6, 12$  and 100.

TABLE A.1: Calculated values of the two fitting parameters  $f$  and  $A_n$  of the  $ES_n$  quadrature.

$ES_n$	$ES_4$	$ES_6$	$ES_{12}$	$ES_{24}$	$ES_{200}$
$f$	0.073	0.0288	0.00688	0.00172	0.000025
$A_n$	1.0798	1.0277	1.00611	1.00151	1.000022

**APPENDIX B: INTERSECTION OF A RAY WITH A CONICAL SEGMENT IN THE AXISYMMETRIC  $rz$ -GEOMETRY**

**1. Basic formulae**

Consider a straight ray

$$X(s) = X_0 + s\Omega_X, \quad Y(s) = Y_0 + s\Omega_Y, \quad Z(s) = Z_0 + s\Omega_Z, \quad (\text{B.1})$$

in the global Cartesian 3D coordinates  $(X, Y, Z)$  [as shown in Fig. B.1], where  $(X_0, Y_0, Z_0)$  is the ray starting point,  $\vec{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$  is a fixed unit 3D vector with

$$\Omega_X^2 + \Omega_Y^2 + \Omega_Z^2 = 1, \quad (\text{B.2})$$

and  $-\infty < s < +\infty$  is the length along the ray measured from the starting point. We are looking for intersection points of the ray (B.1) with a segment of a conical surface, whose axis coincides with the  $Z$ -axis of the coordinate system, and the generatrix passes through two given points  $(R_k, Z_k)$  and  $(R_{k+1}, Z_{k+1})$  in the  $(R, Z)$  plane; here and below

$$R = \sqrt{X^2 + Y^2} \quad (\text{B.3})$$

is the cylindrical radius around the  $Z$ -axis. The considered segment of a cone surface can be parametrized as

$$R(t) = R_k + td_R, \quad Z(t) = Z_k + td_Z, \quad (\text{B.4})$$

where the 2D vector  $\vec{d} = (d_R, d_Z)$  (non-normalized) is given by its  $(R, Z)$  components

$$d_R = R_{k+1} - R_k, \quad d_Z = Z_{k+1} - Z_k, \quad (\text{B.5})$$

and  $0 \leq t \leq 1$  is a parameter. Everywhere below we assume that the 2D vector  $\vec{d}$  has a non-zero length

$$d = \sqrt{d_R^2 + d_Z^2} > 0. \quad (\text{B.6})$$

To find the intersection points, we have to solve the system of equations

$$\begin{cases} Z_k + td_Z = Z_0 + s\Omega_Z, \\ (R_k + td_R)^2 = (X_0 + s\Omega_X)^2 + (Y_0 + s\Omega_Y)^2, \end{cases} \quad (\text{B.7})$$

for the values of  $t$  and  $s$ . Normally we are looking for solutions with  $0 \leq t \leq 1$  and  $s \geq 0$ , which comprise a subset of the full set of solutions with all possible  $-\infty < s < +\infty$  and  $-R_k < td_R < +\infty$ . Below we call the full set of solutions with  $s \in (-\infty, +\infty)$  the *line-cone* solutions, and the restricted subset of solutions with  $t \in [0, 1]$ ,  $s \in [0, +\infty)$  the *ray-cone-segment* solutions.

From Eqs. (B.7) one readily obtains a quadratic equation

$$at^2 - 2bt + c = 0 \quad (\text{B.8})$$

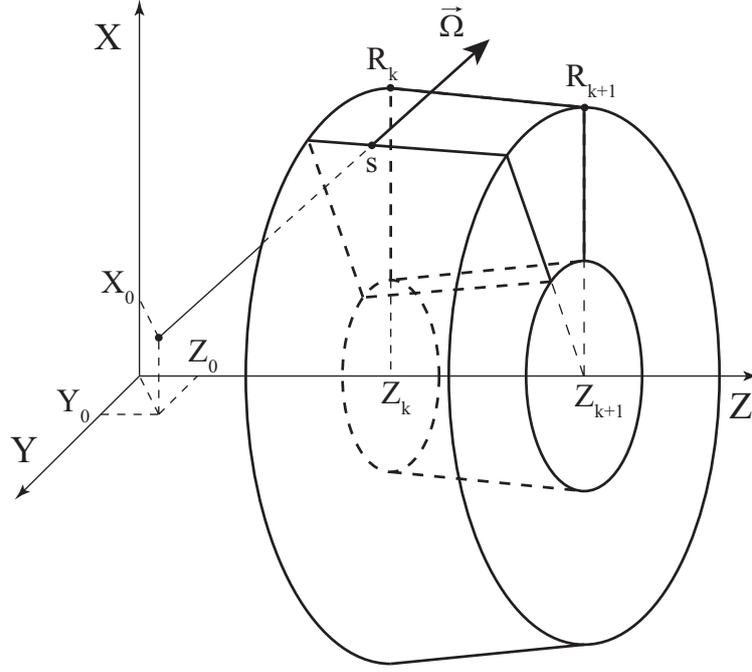


FIG. B.1: Intersection of a ray with a conical surface in global Cartesian coordinates  $(X, Y, Z)$ .

for the value of  $t$ , where

$$a = d_R^2 \Omega_Z^2 - d_Z^2 \varpi_R^2, \quad (\text{B.9})$$

$$b = d_Z [p_0 \Omega_Z + (Z_k - Z_0) \varpi_R^2] - R_k d_R \Omega_Z^2, \quad (\text{B.10})$$

$$c = (R_k^2 - R_0^2) \Omega_Z^2 - (Z_k - Z_0) [(Z_k - Z_0) \varpi_R^2 + 2p_0 \Omega_Z], \quad (\text{B.11})$$

$$\varpi_R^2 = \Omega_X^2 + \Omega_Y^2 = 1 - \Omega_Z^2, \quad (\text{B.12})$$

$$p_0 = X_0 \Omega_X + Y_0 \Omega_Y, \quad (\text{B.13})$$

$$R_0^2 = X_0^2 + Y_0^2. \quad (\text{B.14})$$

Generally Eq. (B.8) has two roots given by

$$t_{1,2} = \frac{b \pm \Omega_Z \sqrt{D}}{a}, \quad s_{1,2} = \frac{d_R v_d \Omega_Z + p_0 d_Z^2 \pm d_Z \sqrt{D}}{a}, \quad (\text{B.15})$$

where

$$D = d_R^2 R_0^2 \Omega_Z^2 + 2p_0 d_R v_d \Omega_Z + v_d^2 \varpi_R^2 - d_Z^2 v_0^2, \quad (\text{B.16})$$

$$v_d = d_R (Z_k - Z_0) - R_k d_Z, \quad (\text{B.17})$$

$$v_0 = X_0 \Omega_Y - Y_0 \Omega_X. \quad (\text{B.18})$$

Note the identity

$$p_0^2 + v_0^2 = R_0^2 \varpi_R^2. \quad (\text{B.19})$$

Representation (B.15) has the advantage of being regular with respect to the components  $\Omega_X$ ,  $\Omega_Y$ ,  $\Omega_Z$  of the unit vector  $\vec{\Omega}$  along the ray in two alternative limits of  $\Omega_Z \rightarrow 0$  and  $\varpi_R \rightarrow 0$ .

In the axisymmetric geometry, where all the computations are performed in the  $(R, Z)$  plane [with either  $(x_1, x_2) = (R, Z)$  for `iradial = 1`, or  $(x_1, x_2) = (Z, R)$  for `iradial = 2`], the 3D unit vector  $\vec{\Omega}$  is represented by its 2D components  $\vec{\tau} = (\Omega_R, \Omega_Z)$ . For a given fixed 3D vector  $\vec{\Omega}$ , the value of its radial projection  $\Omega_R$  on the  $(R, Z)$ -plane depends on the spatial location in the 3D space and is generally given by

$$\Omega_R = \frac{X\Omega_X + Y\Omega_Y}{\sqrt{X^2 + Y^2}}. \quad (\text{B.20})$$

Note that the 2D vector  $\vec{\tau}$  is generally not a unit 2D vector, with its length satisfying the inequality

$$\Omega_R^2 + \Omega_Z^2 \leq 1. \quad (\text{B.21})$$

In Eq. (B.21) the sign of equality holds if and only if

$$X\Omega_Y - Y\Omega_X = 0. \quad (\text{B.22})$$

When point  $(X, Y, Z)$  belongs to the ray (B.1), condition (B.22) is equivalent to  $v_0 = 0$ .

Once the intersection point  $s$  has been found from Eqs. (B.7) and (B.15), one can calculate the radial component  $\Omega_{Rs}$  of the propagation vector  $\vec{\Omega}$  at this point as

$$\Omega_{Rs} = \frac{X_s\Omega_X + Y_s\Omega_Y}{R_s} = \frac{p_0 + s\varpi_R^2}{R_k + td_R}, \quad (\text{B.23})$$

where

$$X_s = X_0 + s\Omega_X, \quad Y_s = Y_0 + s\Omega_Y, \quad R_s = \sqrt{X_s^2 + Y_s^2} = R_k + td_R. \quad (\text{B.24})$$

For numerical applications, however, an alternative representation

$$\Omega_{Rs} = \frac{p_0 + s\varpi_R^2}{\sqrt{R_0^2 + s(2p_0 + s\varpi_R^2)}} \quad (\text{B.25})$$

of this same formula may be preferable because for  $s \geq 0$  it guarantees  $\Omega_{R0} \leq \Omega_{Rs} < 1$ .

## 2. Classification of solutions and selection rules

The principal algorithm for finding the ray-cone-segment solutions is implemented in the subroutine `RAYx3D`, file `'f08_util.f'`. Also, a somewhat shorter particular version of this algorithm makes the basis of the subroutine `R0UTIRZ` in file `'f09_rad.f'`. Mathematically, the problem of intersection of a ray with a segment of a conical surface may have 0, 1, 2, or a continuum solutions — depending on the degree of degeneracy of the quadratic equation (B.8). The algorithm in the subroutine `RAYx3D` is constructed in such a way as to produce either one or none solutions for the intersection point. To be able to select an appropriate single solution from eventual multiple possibilities, we have to introduce an adequate classification scheme for different solution types and appropriate selection rules.

Algebraically, all the solutions of Eq. (B.8) can be sorted out between the following degeneracy cases for the coefficients of the quadratic equation (B.8):

**IF** ( $a \neq 0$ ) **THEN** [*Case A*: 0, 1, or 2 roots, depending on the value of  $D$ ]  
 the relevant solution may be of either the *crossing* or the *tangent* type;

$$t_{1,2} = \frac{b \pm \Omega_Z \sqrt{D}}{a}, \quad s_{1,2} = \frac{d_R v_d \Omega_Z + p_0 d_Z^2 \pm d_Z \sqrt{D}}{a}; \quad (\text{B.26})$$

**ELSEIF** ( $b \neq 0$ ) **THEN** [*Case B*:  $(a = 0) \wedge (b \neq 0) \Rightarrow (\Omega_Z \neq 0)$ , a single root]  
 the relevant solution may be of the *crossing* type only;

$$t = \frac{c}{2b}, \quad s = \frac{Z_k - Z_0 + t d_Z}{\Omega_Z}. \quad (\text{B.27})$$

**ELSEIF** ( $c \neq 0$ ) **THEN** [*Case C*:  $(a = b = 0) \wedge (c \neq 0)$ , no roots]  
 no solution

**ELSE** [*Case D*:  $a = b = c = 0$ , *colinear* solutions, continuum number of roots]

**if** ( $\Omega_Z \neq 0$ ) **then** [*Subcase D1*:  $(\Omega_Z \neq 0) \Rightarrow (d_Z \neq 0)$ ]

in this case the point of origin  $(X_0, Y_0, Z_0)$  lies on the considered conical surface, and the ray runs along its generatrix; the relevant solution may be of the *colinear* type only and belongs to the continuum  $t \in [0, 1]$ ,  $s \in [\min(s_1, s_2), \max(s_1, s_2)]$  (see Fig. B.3), where

$$t_1 = 0, \quad s_1 = \frac{Z_k - Z_0}{\Omega_Z}, \quad (\text{B.28})$$

$$t_2 = 1, \quad s_2 = \frac{Z_{k+1} - Z_0}{\Omega_Z}. \quad (\text{B.29})$$

**else** [*Subcase D2*:  $\Omega_Z = 0 \Rightarrow (d_Z = 0), (d_R \neq 0), (Z_k = Z_{k+1} = Z_0)$ ]

in this case the ray crosses the annular region between the two circles with radii  $R_k$  and  $R_{k+1}$  in the  $(X, Y)$  plane; the relevant solution may be of either the *colinear* or the *tangent* type and belongs to the continuum

$$t \in [0, 1], \quad s \in [\min(s_{11}, s_{21}), \max(s_{11}, s_{21})] \cup [\min(s_{12}, s_{22}), \max(s_{12}, s_{22})] \quad (\text{B.30})$$

where

$$t_1 = 0, \quad s_{11,12} = -p_0 \mp \sqrt{R_k^2 - v_0^2}, \quad (\text{B.31})$$

$$t_2 = 1, \quad s_{21,22} = -p_0 \mp \sqrt{R_{k+1}^2 - v_0^2}; \quad (\text{B.32})$$

the solution is classified as a *tangent* one when it corresponds to the merged either  $s_{11,12}$  or  $s_{21,22}$  roots [in reality, this makes sense only for an exit-type solution at  $R = \max(R_k, R_{k+1})$ ].

**endif**

**ENDIF**

An appropriate geometrical classification of different types of the ray-cone-segment solutions can be introduced by considering a projection onto the computational  $(R, Z)$ -plane (see Figs. B.2–B.4) — where the considered straight 3D ray (B.1) becomes a hyperbola — augmented in some cases by a projection onto the perpendicular  $(X, Y)$ -plane (Fig. B.4b). The key quantity for such a classification is the 2D vector product

$$\vec{\tau}_s \times \vec{d} = (2*\text{iradial} - 3)(\Omega_Z d_R - \Omega_{Rs} d_Z), \quad (\text{B.33})$$

calculated at the intersection point  $s$ . Respectively, a ray-cone-segment solution is classified as

- a *crossing* solution (identified by setting `isotyp = 3`) if and only if  $\vec{\tau}_s \times \vec{d} \neq 0$ ; this is only possible in cases A and B; Figure B.2 provides an illustration for such solutions in case A;
- a *colinear* solution (identified by setting `isotyp = 2`) when it belongs to a continuum set of line-cone solutions; this is only possible in case D of  $a = b = c = 0$ ; all colinear solutions have  $\vec{\tau}_s \times \vec{d} = 0$ ; several possible variants of colinear solutions are illustrated in Figs. B.3 and B.4;
- a *tangent* solution (identified by setting `isotyp = 1`) when it either
  - (i) is not a colinear solution (i.e. is obtained in case A or B) but has  $\vec{\tau}_s \times \vec{d} = 0$  (an illustration for such a solution is given in Fig. B.2), or
  - (ii) is a colinear solution with  $\Omega_Z = d_Z = 0$  (i.e. is obtained in subcase D2) and has  $\Omega_{Rs} = 0$  (see Fig. B.4b).

If no solutions exist, the parameter `isotyp` is set equal to `isotyp = 0`. Crossing solutions are always preferred to colinear and tangent ones.

The algorithm for solving the ray-cone-segment problem is usually applied to find an exit from, or an entry to a particular mesh cell (or computational domain). The latter means that acceptable are only such solutions where the ray crosses the segment  $\vec{d}$  in a specific direction — for example, from right to left when moving along  $\vec{d}$  if an entry to the region on the left-hand side of  $\vec{d}$  is sought. The direction of the crossing is defined by the sign of the 2D vector product (B.33). Hence, in the `subroutine RAYx3D` the convention is adopted that only those crossing-type solutions are accepted for which

$$\vec{\tau}_s \times \vec{d} < 0, \quad (\text{B.34})$$

i.e. for which the ray *enters* the domain on the left from  $\vec{d}$ . If an exit from this domain is to be found, the `subroutine RAYx3D` has to be called with the points  $k$  and  $k+1$  interchanged. In the `subroutine ROUTIRZ` the convention is just the opposite.

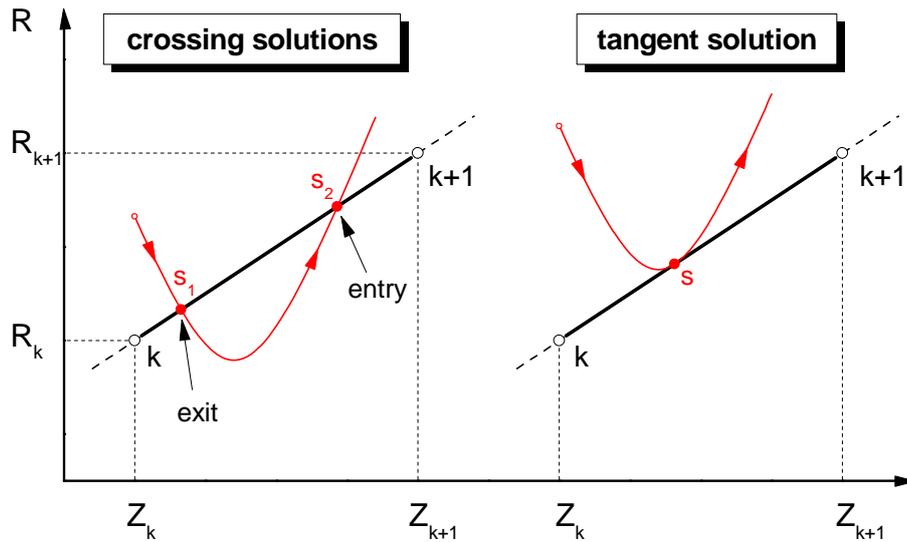


FIG. B.2: Possible types of the ray-cone-segment solutions in case A. The projection of ray (B.1) onto the  $(Z, R)$ -plane (for  $\text{iradial} = 2$ ) is shown as a red hyperbola. The crossing solution  $s_1$  is of the exit-type, and the solution  $s_2$  is of the entry-type with respect to the region on the left when moving from  $(Z_k, R_k)$  to  $(Z_{k+1}, R_{k+1})$ .

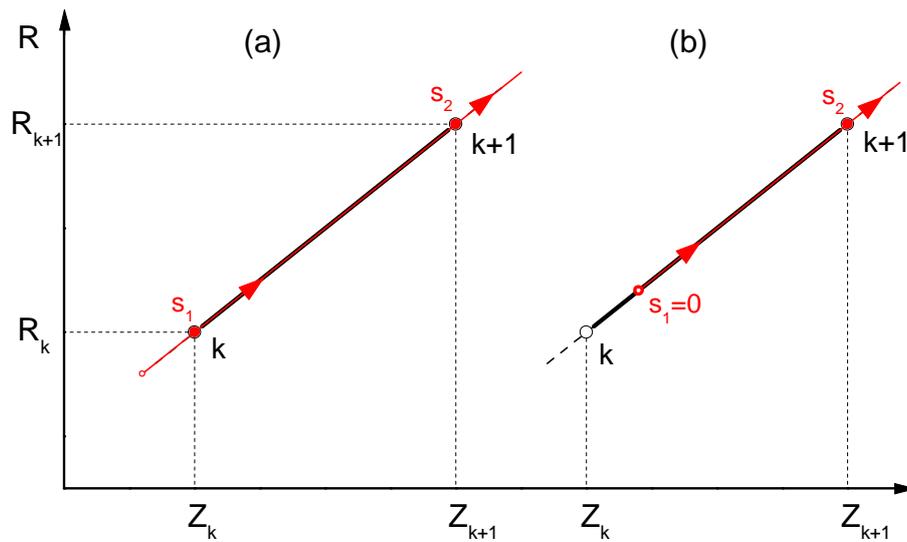


FIG. B.3: Two variants of colinear solutions in case D1. The projection of ray (B.1) onto the  $(Z, R)$ -plane (for  $\text{iradial} = 2$ ) is shown as a red line. (a) The ray starts outside the conical segment  $[k, k + 1]$ ; (b) the ray starts on the conical segment  $[k, k + 1]$ . In both cases, out of the continuum  $s \in [s_1, s_2]$  of possible solutions  $s_1$  is selected as the entry-type solution, and  $s_2$  as the exit-type solution.

The situation becomes more complex for the colinear and tangent solutions, where  $\vec{\tau}_s \times \vec{d} = 0$ . Here we have to specify explicitly whether we are looking for an *entry-type* or an *exit-*

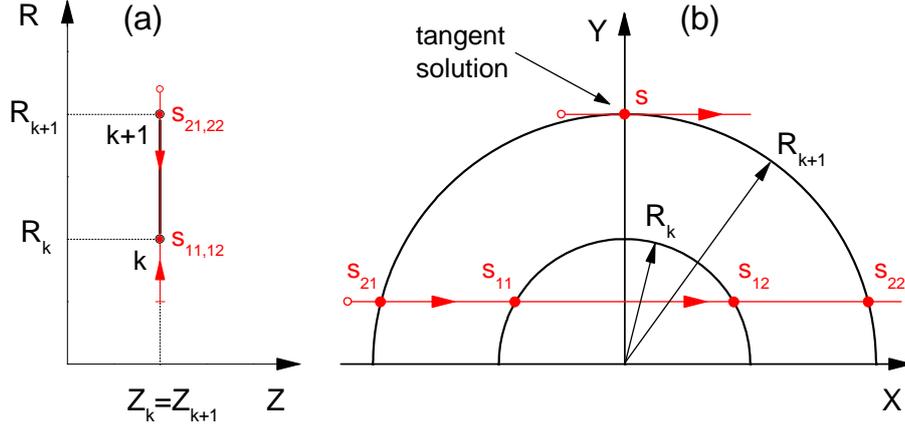


FIG. B.4: Colinear and tangent solutions in case D2. The projection of ray (B.1) onto the (a)  $(Z, R)$ - and (b)  $X, Y$ -planes (for  $\text{iradial} = 2$ ) is shown as a red line. Once the ray starts outside the annulus  $[k, k + 1]$  and all the four crossing points  $s_{11}, s_{12}, s_{21}, s_{22}$  are present, the crossing  $s_{21}$  is selected as the entry-type solution, and the crossing  $s_{11}$  as the exit-type solution (when  $R_k < R_{k+1}$ ). The solution is classified as a tangent one when it occurs at merged either  $s_{11,12}$  or  $s_{21,22}$  points.

*type* solution. This is done by setting a separate logical parameter `ifentry` either `.true.` or `.false.` The entry (exit) domain itself remains undefined. From the multiple colinear and/or tangent solutions, allowed by Eq. (B.8), always the one with the minimum non-negative  $s$  is selected for the entry-type solution, and the one with the maximum non-negative  $s$  for the exit-type solution. Such a selection rule allows to correctly perform the ray-tracing along such special lines as the rotation axis or a reflective boundary, and to avoid cyclic traps (by setting appropriate small safety margins) due to intermittent exit and entry through the same point at a cell boundary.

### 3. Rounding errors and safety margins

To construct a stable algorithm, which correctly identifies all degenerate cases, we have to introduce certain margins into the corresponding mathematical conditions to account for the effect of the rounding errors. For example, in their finite-difference form the mathematical conditions  $a = 0, b = 0, c = 0, 0 \leq t \leq 1$ , and  $s \geq 0$  are replaced by

$$|a| > 0 \quad \Rightarrow \quad |a| > \Delta_a, \quad (\text{B.35})$$

$$|b| > 0 \quad \Rightarrow \quad |b| > \Delta_b, \quad (\text{B.36})$$

$$|c| > 0 \quad \Rightarrow \quad |c| > \Delta_c, \quad (\text{B.37})$$

$$0 \leq t \leq 1 \quad \Rightarrow \quad -\Delta_t \leq t \leq 1 + \Delta_t, \quad (\text{B.38})$$

$$s \geq 0 \quad \Rightarrow \quad s \geq -\Delta_s, \quad (\text{B.39})$$

where small positive numbers  $\Delta_a > 0, \Delta_b > 0, \Delta_c > 0, \Delta_t > 0, \Delta_s > 0$  are sort of safety margins which should in principle guarantee that, despite ever present rounding errors, an appropriate and a fairly accurate solution to the ray-cone-segment problem is found under

any circumstances. In this sense, the original “sharp” mathematical conditions are made somewhat “fuzzy”. The values of all the safety margins are controlled by the following basic three user-defined parameters:

$$\begin{aligned}
 \delta = 10^{-16} - 10^{-15} &\rightarrow \text{an upper bound to the expected base relative level of rounding errors in arithmetic operations; loaded in module COMPAR;} \\
 \delta_{tmx} = 10^{-6} - 10^{-4} &\rightarrow \text{maximum allowed relative error level for calculated values of the fractional position } 0 < t < 1 \text{ on the crossed segment } \vec{d}; \text{ loaded in subroutines RAYx3D and ROUTIRZ;} \\
 f_{sf} = 2 - 10 &\rightarrow \text{an additional safety factor used under certain conditions that might require additional broadening of the corresponding safety margin; loaded in subroutines RAYx3D and ROUTIRZ.}
 \end{aligned} \tag{B.40}$$

To set the margins  $\Delta_a, \Delta_b, \Delta_c, \Delta_t, \Delta_s$ , we introduce two fiducial values

$$R_f = \max(|X_0|, |Y_0|, R_k, R_{k+1}), \quad Z_f = \max(|Z_0|, |Z_k|, |Z_{k+1}|) \tag{B.41}$$

of the radial and axial coordinates of the three relevant points in the ray-cone-segment problem. Then, we evaluate the upper bounds for the rounding errors of the corresponding quantities. Below the upper bound to the absolute rounding error of any quantity  $q$  is denoted as  $\delta_q$ . The absolute error introduced by the components of the unit vector  $\Omega$  is assumed to be  $\leq \delta$ .

Noting that the rounding errors for  $d_R$  and  $d_Z$  are, respectively,  $R_f\delta$  and  $Z_f\delta$ , from Eqs. (B.9)–(B.11) the absolute rounding errors for coefficients  $a, b$ , and  $c$  can be evaluated as

$$\delta_a = \max \{ |d_R| \Omega_Z^2 (R_f \delta); |d_Z| \varpi_R^2 (Z_f \delta); d_{RZ}^2 \delta \}, \tag{B.42}$$

$$\delta_b = \max (\delta_b^R, \delta_b^Z, \delta_b^\Omega),$$

$$\delta_b^R = \max [|d_Z \Omega_Z|, R_k \Omega_Z^2] (R_f \delta),$$

$$\delta_b^Z = \max [|d_Z| \varpi_R^2, |p_0 \Omega_Z + (Z_k - Z_0) \varpi_R^2|] (Z_f \delta),$$

$$\delta_b^\Omega = \max [|d_Z| (|p_0| + |Z_k - Z_0|) + |R_k d_R|] \delta, \tag{B.43}$$

$$\delta_c = \max \{ \delta_c^R; \delta_c^Z; [R_f^2 + (Z_k - Z_0)^2] \delta \},$$

$$\delta_c^R = \max (R_f \Omega_Z^2, |(Z_k - Z_0) \Omega_Z|) (R_f \delta),$$

$$\delta_c^Z = \max (|Z_k - Z_0| \varpi_R^2, |(Z_k - Z_0) \varpi_R^2 + 2p_0 \Omega_Z|) (Z_f \delta), \tag{B.44}$$

where

$$d_{RZ}^2 = d_R^2 + d_Z^2. \tag{B.45}$$

Then the safety margins in Eqs. (B.35)–(B.37) are defined as

$$\Delta_a = \frac{\delta_a}{\delta_{tmx}}, \quad \Delta_b = \frac{\delta_b}{\delta_{tmx}}, \quad \Delta_c = f_{sf} \frac{\delta_c}{\delta_{tmx}}. \tag{B.46}$$

Having chosen  $\delta_{tmx} \ll 1$ ,  $\delta_{tmx} \gg \delta$ , we can ensure that the maximum error for  $t$  in Eqs. (B.26) and (B.27) does not exceed a small value  $\simeq \delta_{tmx} \ll 1$ . The margin  $\Delta_c$  is increased by a factor  $f_{sf} > 1$  relative to the “nominal” value in order to reduce the probability of case C, where no solution is found.

Further on, we need the error estimates

$$\begin{aligned} \delta_D &= \max \{ \delta_D^R; \delta_D^Z; [R_f^2 + (Z_k - Z_0)^2] d_{RZ}^2 \delta \}, \\ \delta_D^R &= \max \{ R_f (|d_R| R_f \Omega_Z^2 + d_Z^2); (|d_R| + |p_0|) |v_d \Omega_Z| \} (R_f \delta) \\ \delta_D^Z &= \max \{ R_f (|d_R p_0 \Omega_Z| + |v_d| \varpi_R^2), |d_Z| v_0^2 \} (Z_f \delta), \end{aligned} \tag{B.47}$$

$$\delta_{\sqrt{D}} = \begin{cases} \delta_D / \sqrt{D}, & D \geq \delta_D \\ \sqrt{\delta_D}, & D < \delta_D. \end{cases} \tag{B.48}$$

for the discriminant  $D$  and its square root  $\sqrt{D}$ . Then, in case A from Eq. (B.26) we obtain

$$\Delta_t = f_{sm} \delta_t = \frac{f_{sm}}{|a|} (\delta_b + \sqrt{D} \delta + |\Omega_Z| \delta_{\sqrt{D}} + |t| \delta_a), \tag{B.49}$$

$$\begin{aligned} \Delta_s &= f_{sm} \delta_t = \frac{f_{sm}}{|a|} (\delta_s^{(1)} + |d_Z| \delta_{\sqrt{D}} + |s| \delta_a), \\ \delta_s^{(1)} &= \max \{ \max (d_Z^2, |v_d \Omega_Z|) (R_f \delta); |d_R v_d| \delta; \\ &\quad \max (R_f |d_R \Omega_Z|, |p_0 d_Z|, \sqrt{D}) (Z_f \delta) \}. \end{aligned} \tag{B.50}$$

Similarly, in case B from Eq. (B.27) we get

$$\Delta_t = f_{sm} \delta_t = \frac{f_{sm}}{|b|} (\delta_c + |t| \delta_b), \tag{B.51}$$

$$\Delta_s = f_{sm} \delta_t = \frac{f_{sm}}{|\Omega_Z|} (Z_f \delta + |d_Z| \delta_t). \tag{B.52}$$

The safety margin  $\Delta_{\tau d}$  to be used when applying inequality (B.34) can be evaluated as

$$\Delta_{\tau d} = f_{sm} (|d_R| + |d_Z|) \sqrt{\delta}. \tag{B.53}$$

---

*Correspondence with the code variables:*

$R_f$	= <b>rfidu</b>	fiducial value of the $(x_1, x_2)$ coordinate values in the ray-cone-segment problem;
$\delta$	= <b>dfuzz</b>	$\simeq 10^{-15}$ , user-defined upper bound to rounding errors in a single arithmetic operation;
$\delta_{tmx}$	= <b>dfuzmx</b>	$\simeq 10^{-5}$ , user-defined upper bound for the errors in $t$ and $s$ in Eqs. (B.26), (B.27);
$f_{sf}$	= <b>fsmarg</b>	$\simeq 2-4$ , an additional safety factor for broadening some safety margins; loaded in subroutines RAYx3D and ROUTIRZ
$\delta_a$	= <b>d_a</b>	
$\delta_b$	= <b>d_b</b>	
$\delta_c$	= <b>d_c</b>	
$\delta_D$	= <b>d_D</b>	
$\Delta_t$	= <b>tfuzz</b>	
$\Delta_s$	= <b>sfuzz</b>	
$\Delta_{\tau d}$	= <b>vpfuzz</b>	
$\Omega_{Rs}$	= <b>taur</b>	radial component of the 2D vector $\vec{\tau}_s$ ;

---

APPENDIX C: SECOND-ORDER INTERPOLATION SCHEMES

1. Positive piece-wise parabolic oscillation-suppressed interpolation on a 3-point stencil (PPOS3P)

Suppose we know the values of a function  $f(x)$  at 3 points  $x = x_-$ ,  $x = 0$ , and  $x = x_+$  such that  $x_- < 0 < x_+$ :

$$f_- = f(x_-), \quad f_0 = f(0), \quad f_+ = f(x_+). \tag{C.1}$$

We want to construct a smooth quadratic interpolation to this function on the interval  $[x_-, x_+]$  that would be strictly positive and have a limited amplitude of non-monotonic variation that could be easily brought down to zero. The formulae below are simplified after we introduce the notation

$$\bar{f}(x) = f(x) - f_0, \quad \bar{f}_- = f_- - f_0, \quad \bar{f}_+ = f_+ - f_0. \tag{C.2}$$

The single quadratic polynomial that passes through the 3 points  $\bar{f}_-$ , 0, and  $\bar{f}_+$  is

$$\bar{f}_r(x) = \bar{f}_- \frac{x}{x_-} \frac{x_+ - x}{x_+ - x_-} + \bar{f}_+ \frac{x}{x_+} \frac{x - x_-}{x_+ - x_-}, \tag{C.3}$$

which would give us a regular interpolation formula. Generally, it is not monotonic and not positive [with respect to  $f_r(x) = \bar{f}_r(x) + f_0$ ]. Its 1st derivative is given by

$$\bar{f}'_r(x) = \frac{\bar{f}_-}{x_-} \frac{x_+ - 2x}{x_+ - x_-} + \frac{\bar{f}_+}{x_+} \frac{2x - x_-}{x_+ - x_-}, \tag{C.4}$$

$$\sigma_r \stackrel{\text{def}}{=} \bar{f}'_r(0) = \frac{1}{x_+ - x_-} \left( \frac{\bar{f}_-}{x_-} x_+ - \frac{\bar{f}_+}{x_+} x_- \right). \tag{C.5}$$

To make the interpolation positive and suppress the amplitude of its non-monotonic variation, we use slope correction at  $x = 0$  and construct a piece-wise parabolic interpolation of the form

$$\bar{f}(x) = \begin{cases} x \left[ \sigma \left( 1 - \frac{x}{x_-} \right) + \frac{\bar{f}_-}{x_-} \frac{x}{x_-} \right], & x_- \leq x \leq 0, \\ x \left[ \sigma \left( 1 - \frac{x}{x_+} \right) + \frac{\bar{f}_+}{x_+} \frac{x}{x_+} \right], & 0 \leq x \leq x_+, \end{cases} \tag{C.6}$$

where  $\sigma = \bar{f}'(0)$  is the slope to be corrected. The piece-wise parabolic interpolation (C.6) has a continuous 1st derivative on  $[x_-, x_+]$ , but generally a discontinuous (at  $x = 0$ ) 2nd derivative — which becomes continuous only for  $\sigma = \sigma_r$ . The 1st derivative of (C.6) is given

by

$$\bar{f}'(x) = \begin{cases} \sigma \left(1 - \frac{2x}{x_-}\right) + 2 \frac{\bar{f}_-}{x_-} \frac{x}{x_-}, & x_- \leq x \leq 0, \\ \sigma \left(1 - \frac{2x}{x_+}\right) + 2 \frac{\bar{f}_+}{x_+} \frac{x}{x_+}, & 0 \leq x \leq x_+, \end{cases} \quad (\text{C.7})$$

$$\bar{f}'(x_-) = -\sigma + 2 \frac{\bar{f}_-}{x_-}, \quad (\text{C.8})$$

$$\bar{f}'(x_+) = -\sigma + 2 \frac{\bar{f}_+}{x_+}. \quad (\text{C.9})$$

To ensure positiveness, we introduce a dimensionless control parameter  $0 \leq \varepsilon_d < 1$  defined such that

$$f(x) = f_0 + \bar{f}(x) \geq (1 - \varepsilon_d) \min(f_-, f_0, f_+). \quad (\text{C.10})$$

According to this inequality, the maximum downward deviation of  $\bar{f}(x)$  from the monotonic variation is limited to

$$h_d = \varepsilon_d \min(f_-, f_0, f_+) \geq 0. \quad (\text{C.11})$$

To suppress oscillatory behavior, we introduce a second dimensionless control parameter  $0 \leq \varepsilon_u < +\infty$ , defined such that the maximum allowed deviation (both upward and downward) from the monotonic interpolation does not exceed

$$h_{u-} = \frac{1}{4} \varepsilon_u \left| x_- \frac{\bar{f}_+}{x_+} \right| \quad \text{at } x_- \leq x \leq 0, \quad \text{and} \quad (\text{C.12})$$

$$h_{u+} = \frac{1}{4} \varepsilon_u \left| x_+ \frac{\bar{f}_-}{x_-} \right| \quad \text{at } 0 \leq x \leq x_+.$$

This condition translates into applying the limiting slopes

$$\lambda_- = \varepsilon_u \frac{\bar{f}_-}{x_-}, \quad \lambda_+ = \varepsilon_u \frac{\bar{f}_+}{x_+}. \quad (\text{C.13})$$

For  $\varepsilon_d = \varepsilon_u = 0$  we obtain a strictly positive and monotonic interpolation formula. For  $\varepsilon_u \leq 1$  the curvature of the interpolation formula (C.6) does not change its sign when experiencing a jump at  $x = 0$ .

Evaluation of the slope  $\sigma$  splits into the following 4 cases.

*Case 1:*  $\bar{f}_- \leq 0 \leq \bar{f}_+$ . In this case we have  $\sigma_r \geq 0$ ,  $\lambda_- \geq 0$ ,  $\lambda_+ \geq 0$ . Here, in the situation shown in Fig. C.1a, we limit the amplitude of non-monotonicity by demanding  $\bar{f}'(x_-) \geq 4h_d/x_- \wedge \bar{f}'(x_-) \geq -\lambda_+$ , and in the case of Fig. C.1b by demanding  $\bar{f}'(x_+) \geq -\lambda_-$ , which results in the following slope correction

$$\sigma = \min(\sigma_r, \sigma_{d-}, \sigma_{u+}), \quad (\text{C.14})$$

$$\sigma_{d-} = 2 \frac{\bar{f}_-}{x_-} + \min\left(\lambda_+, -\frac{4h_d}{x_-}\right), \quad (\text{C.15})$$

$$\sigma_{u+} = 2 \frac{\bar{f}_+}{x_+} + \lambda_-. \quad (\text{C.16})$$

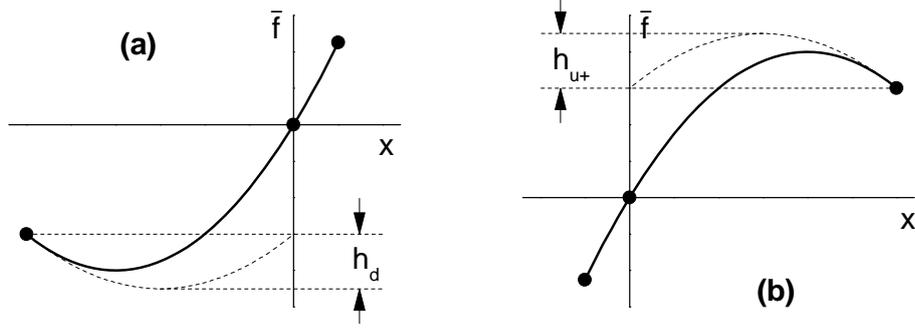


FIG. C.1: Case 1 of the PPOS3P interpolation.

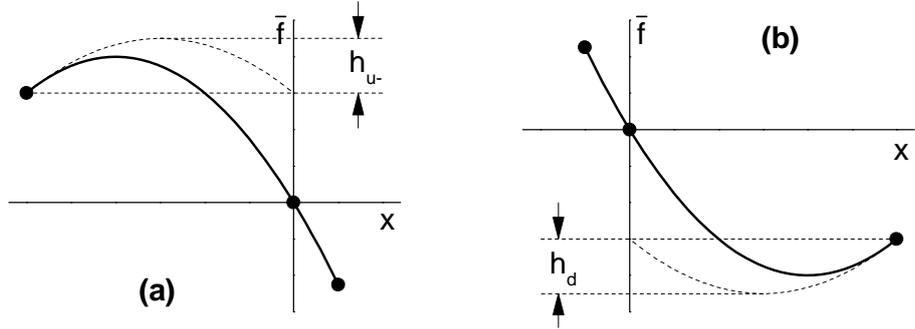


FIG. C.2: Case 2 of the PPOS3P interpolation.

*Case 2:*  $\bar{f}_- \geq 0 \geq \bar{f}_+$ . In this case we have  $\sigma_r \leq 0$ ,  $\lambda_- \leq 0$ ,  $\lambda_+ \leq 0$ . Here, in the situation shown in Fig. C.2a, we limit the amplitude of non-monotonicity by demanding  $\bar{f}'(x_-) \leq -\lambda_+$ , and in the case of Fig. C.2b by demanding  $\bar{f}'(x_+) \leq 4h_d/x_+ \wedge \bar{f}'(x_+) \leq -\lambda_-$ , which results in the following slope correction

$$\sigma = \max(\sigma_r, \sigma_{d+}, \sigma_{u-}), \tag{C.17}$$

$$\sigma_{d+} = 2\frac{\bar{f}_+}{x_+} + \max\left(\lambda_-, -\frac{4h_d}{x_+}\right), \tag{C.18}$$

$$\sigma_{u-} = 2\frac{\bar{f}_-}{x_-} + \lambda_+. \tag{C.19}$$

*Case 3:*  $\bar{f}_- \leq 0 \wedge \bar{f}_+ \leq 0$ . In this case we have  $\lambda_- \geq 0$ ,  $\lambda_+ \leq 0$ . Here, in the situation shown in Fig. C.3a, we limit the amplitude of non-monotonicity by demanding  $\sigma \geq \lambda_+$ , and in the case of Fig. C.3b by demanding  $\sigma \leq \lambda_-$ , which results in the following slope correction

$$\sigma = \max[\lambda_+, \min(\sigma_r, \lambda_-)]. \tag{C.20}$$

*Case 4:*  $\bar{f}_- \geq 0 \wedge \bar{f}_+ \geq 0$ . In this case we have  $\lambda_- \leq 0$ ,  $\lambda_+ \geq 0$ . Here, in the situation shown in Fig. C.4b we limit the amplitude of non-monotonicity by demanding  $\sigma \leq \lambda_+ \wedge \sigma \leq$

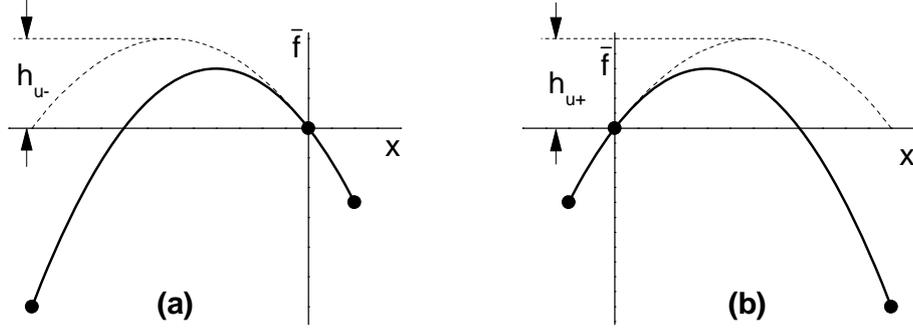


FIG. C.3: Case 3 of the PPOS3P interpolation.

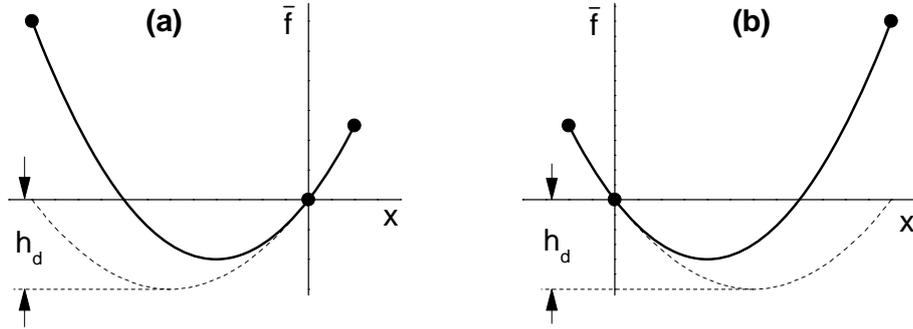


FIG. C.4: Case 4 of the PPOS3P interpolation.

$-4h_d/x_-$ , and in the case of Fig. C.4b by demanding  $\sigma \geq \lambda_- \wedge \sigma \geq -4h_d/x_+$ , which results in the following slope correction

$$\sigma = \max \left[ \lambda_-, -\frac{4h_d}{x_+}, \min \left( \sigma_r, \lambda_+, -\frac{4h_d}{x_-} \right) \right]. \tag{C.21}$$

### 2. Symmetric 4-point piece-wise parabolic interpolation

Consider a function  $f(\xi)$ , for which we seek a parabolic interpolation on a normalized interval  $0 \leq \xi \leq 1$ . We want to construct this interpolation by using four values of this function

$$f_- = f(-\Delta_-), \quad f_0 = f(0), \quad f_1 = f(1), \quad f_+ = f(1 + \Delta_+), \tag{C.22}$$

known at four discrete points  $\xi = -\Delta_-, 0, 1,$  and  $1 + \Delta_+$ , where  $\Delta_- > 0, \Delta_+ > 0$ . The interpolation must be symmetric with respect to the values  $f_-$  and  $f_+$ .

We start with a symmetric form

$$f(\xi) = f_0(1 - \xi) + f_1\xi - A\xi(1 - \xi) = f_0 + (f_1 - f_0 - A)\xi + A\xi^2, \tag{C.23}$$

where a constant  $A$  represents an increment of the first derivative  $df/d\xi = f'(\xi)$  between the points  $\xi = 0$  and  $\xi = 1$ . This increment

$$2A = f'(1) - f'(0) \tag{C.24}$$

can be evaluated by using the following second-order 3-point formulae for numerical derivatives

$$f'(0) = -\frac{\Delta_-^{-2}}{1 + \Delta_-^{-1}}f_- + (\Delta_-^{-1} - 1)f_0 + \frac{1}{1 + \Delta_-^{-1}}f_1, \quad (\text{C.25a})$$

$$f'(1) = -\frac{1}{1 + \Delta_+^{-1}}f_0 + (1 - \Delta_+^{-1})f_1 + \frac{\Delta_+^{-2}}{1 + \Delta_+^{-1}}f_+. \quad (\text{C.25b})$$

From Eqs. (C.24) and (C.25) we obtain

$$2A = \frac{1}{\Delta_- + \Delta_-^2}f_- + \left(\frac{1}{1 + \Delta_+} - \Delta_-^{-1}\right)f_0 + \left(\frac{1}{1 + \Delta_-} - \Delta_+^{-1}\right)f_1 + \frac{1}{\Delta_+ + \Delta_+^2}f_+. \quad (\text{C.26})$$

Also, thus constructed interpolation can be cast in the form

$$f(\xi) = \gamma_-f_- + \gamma_0f_0 + \gamma_1f_1 + \gamma_+f_+, \quad (\text{C.27})$$

where

$$\gamma_- = -\frac{1}{2}\frac{\xi(1-\xi)}{\Delta_- + \Delta_-^2}, \quad \gamma_+ = -\frac{1}{2}\frac{\xi(1-\xi)}{\Delta_+ + \Delta_+^2}, \quad (\text{C.28a})$$

$$\gamma_1 = \xi + \Delta_- \gamma_- - (1 + \Delta_+) \gamma_+ \quad (\text{C.28b})$$

$$1 = \gamma_- + \gamma_0 + \gamma_1 + \gamma_+. \quad (\text{C.28c})$$

The error estimate for this interpolation can readily be obtained in the particular case of equally spaced discrete abscissae, i.e. for  $\Delta_- = \Delta_+ = 1$ . In this case, the exact Taylor expansion around  $\xi = 0$  is given by

$$f_{ex}(\xi) = f_0 + \xi f'_0 + \frac{1}{2}\xi^2 f''_0 + \frac{1}{6}\xi^3 f'''_0 + \dots \quad (\text{C.29})$$

while the interpolation formula (C.27) yields

$$f_{ap}(\xi) = f_0 + \xi f'_0 + \frac{1}{2}\xi^2 f''_0 + \frac{1}{6}\left(\frac{3}{2}\xi^2 - \frac{1}{2}\xi\right) f'''_0 + \dots \quad (\text{C.30})$$

Here all the derivatives  $f'_0, f''_0, \dots$  at point  $\xi = 0$  are with respect to the normalized variable  $\xi$ . The error of the interpolation (C.27) is given by

$$Err \equiv |f_{ex}(\xi) - f_{ap}(\xi)| = \frac{1}{12} |2\xi^3 - 3\xi^2 + \xi| |f'''_0| + \dots < \frac{1}{120} |f'''_0| + \dots \quad (\text{C.31})$$

Thus, as the separation  $h$  between the non-normalized discrete abscissae  $x_i$  is diminished, the error of our interpolation formula decreases as

$$Err \approx \frac{h^3}{120} |f'''_{xxx}|. \quad (\text{C.32})$$

### 3. Positive piece-wise parabolic interpolation

The general positive second-order interpolation scheme is constructed as follows [36]. We start with a parabolic approximation of the form

$$f(\xi) = f_0 + a\xi + \frac{1}{2}(b - a)\xi^2 \tag{C.33}$$

to a positive function  $f(\xi) \geq 0$  on a normalized interval  $\xi \in [0, 1]$ ; known are the values  $f_0 = f(0) \geq 0$ ,  $f_1 = f(1) \geq 0$ , and  $f_2 = f(\xi_2) \geq 0$ . The value  $\xi_2$  must be different from 0 and 1, otherwise arbitrary. The first derivatives  $f'(0) = a$  and  $f'(1) = b$  are represented by unknown coefficients  $a$  and  $b$ , which obey the relationship

$$a + b = 2(f_1 - f_0). \tag{C.34}$$

To ensure positiveness of  $f(\xi)$  for all  $\xi \in [0, 1]$ , it is sufficient to impose the following constraints on the slopes  $a$  and  $b$

$$a \geq -f_0, \quad b \leq f_1, \tag{C.35}$$

which appear to be “physically most reasonable”, given the values  $f_0 \geq 0$  and  $f_1 \geq 0$ , and on the premise that  $f(\xi)$  must be non-negative for  $\xi \in [0, 1]$ . As a consequence, we arrive at the following slope-corrector formulae

$$a = \begin{cases} 2(f_1 - f_0) - b, & f_0 < f_1, \\ \max \left\{ \frac{-f_2 + f_0(1 - \xi_2^2) + f_1\xi_2^2}{\xi_2(\xi_2 - 1)}; -f_0 \right\}, & f_0 \geq f_1, \end{cases} \tag{C.36}$$

$$b = \begin{cases} \min \left\{ \frac{f_2 - f_0(1 - \xi_2)^2 - f_1(2\xi_2 - \xi_2^2)}{\xi_2(\xi_2 - 1)}; f_1 \right\}, & f_0 < f_1, \\ 2(f_1 - f_0) - a, & f_0 \geq f_1. \end{cases} \tag{C.37}$$

Note that for  $f_0 < f_1$  inequality  $b \leq f_1$  implies  $a > -f_0$ , whereas for  $f_0 \geq f_1$  inequality  $a \geq -f_0$  implies  $b \leq f_1$ .

### 4. Monotonic piece-wise parabolic interpolation

Here we construct a second-order interpolation to a function  $f(\xi)$ , which is monotonic on the normalized interval  $\xi \in [0, 1]$ . Without loss of generality, the value  $f_0 = f(0)$  can be set equal to zero,  $f_0 = 0$ . Then, we are looking for an interpolation of the form

$$f(\xi) = a\xi + \frac{1}{2}(b - a)\xi^2, \tag{C.38}$$

subject to constraints

$$0 \leq f(\xi) \leq f_1 \quad \text{for} \quad f_1 \geq 0, \tag{C.39}$$

$$f_1 \leq f(\xi) \leq 0 \quad \text{for} \quad f_1 < 0; \tag{C.40}$$

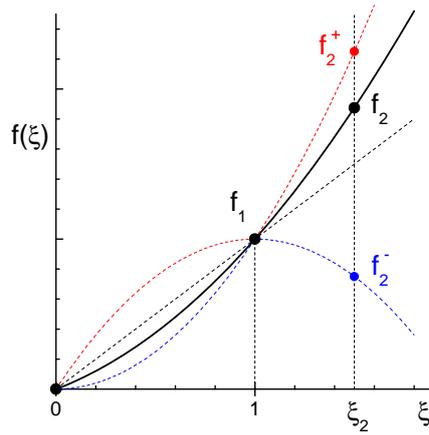


FIG. C.5: Second-order monotonic interpolation for  $f(\xi)$  on a normalized interval  $\xi \in [0, 1]$ .

known are the values  $f_1 = f(1)$  and  $f_2 = f(\xi_2)$ . The point  $\xi = \xi_2$  must be different from points  $\xi = 0$  and  $\xi = 1$ , otherwise arbitrary. The first derivatives  $f'(0) = a$  and  $f'(1) = b$  are represented by unknown coefficients  $a$  and  $b$ , which obey the relationship

$$a + b = 2f_1. \tag{C.41}$$

The necessary and sufficient conditions for the parabolic interpolation (C.38) to be monotonic at  $\xi \in [0, 1]$  are

$$a \geq 0 \text{ and } b \geq 0 \quad \text{for } f_1 \geq 0, \tag{C.42}$$

$$a \leq 0 \text{ and } b \leq 0 \quad \text{for } f_1 < 0. \tag{C.43}$$

As a consequence, we obtain the following formulae for  $a$  and  $b$ :

$$a = \frac{f_1 \xi_2^2 - \tilde{f}_2}{\xi_2(\xi_2 - 1)}, \tag{C.44}$$

$$b - a = 2 \frac{\tilde{f}_2 - f_1 \xi_2}{\xi_2(\xi_2 - 1)}, \tag{C.45}$$

where

$$\tilde{f}_2 = \max \{ f_2^-; \min \{ f_2^+; f_2 \} \}, \tag{C.46}$$

$$f_2^- = \min \{ f_1 \xi_2 (2 - \xi_2); f_1 \xi_2^2 \}, \tag{C.47}$$

$$f_2^+ = \max \{ f_1 \xi_2 (2 - \xi_2); f_1 \xi_2^2 \}. \tag{C.48}$$

Figure C.5 illustrates formulae (C.44)–(C.48) for the case of  $\xi_2 > 1$ ,  $f_1 > 0$ .

Sometimes it is more convenient to present the interpolation formula (C.38) as an explicit linear function of  $f_1$  and  $f_2$  in the form

$$f(\xi) = a_1 f_1 + a_2 f_2, \tag{C.49}$$

where coefficients  $a_1$  and  $a_2$  are given by

1)  $f_2^- \leq f_1 \leq f_2^+$ :

$$a_1 = \frac{\xi(\xi_2 - \xi)}{\xi_2 - 1}, \quad a_2 = \frac{\xi(\xi - 1)}{\xi_2(\xi_2 - 1)}; \quad (C.50)$$

2)  $[f_1 > 0 \wedge f_2 > f_1\xi_2^2]$  or  $[f_1 \leq 0 \wedge f_2 < f_1\xi_2^2]$  :

$$a_1 = \xi^2, \quad a_2 = 0; \quad (C.51)$$

3)  $[f_1 > 0 \wedge f_2 < f_1\xi_2(2 - \xi_2)]$  or  $[f_1 \leq 0 \wedge f_2 > f_1\xi_2(2 - \xi_2)]$  :

$$a_1 = \xi(2 - \xi), \quad a_2 = 0. \quad (C.52)$$

The above formulae can be programmed as

**if**  $[(f_2 - f_1\xi_2^2)(f_2 - f_1\xi_2(2 - \xi_2)) \leq 0]$  **then**

$$a_1 = \frac{\xi(\xi_2 - \xi)}{\xi_2 - 1}, \quad a_2 = \frac{\xi(\xi - 1)}{\xi_2(\xi_2 - 1)},$$

**elseif**  $[f_1f_2 > 0 \wedge (|f_2| > |f_1\xi_2^2|)]$  **then**

$$a_1 = \xi^2, \quad a_2 = 0,$$

**else**

$$a_1 = \xi(2 - \xi), \quad a_2 = 0.$$

**endif**

The present interpolation scheme has a disadvantage of being non-symmetric with respect to point  $\xi = 1$ . Indeed, assume for example that  $\xi_2 > 1$  and  $f_1 = f_2 > 0$ . Then one readily verifies that our interpolation, being monotonic at  $0 < \xi < 1$ , is non-monotonic at  $1 < \xi < \xi_2$ . The latter means that if we construct this interpolation from right to left, i.e. for the function  $\tilde{f}(\eta) = f(\xi_2 - \eta\xi_2 + \eta)$ , where  $\eta = (\xi_2 - \xi)/(\xi_2 - 1)$ , and make it monotonic in the interval  $\eta \in [0, 1]$ , we will obtain different values of  $f(\xi)$  for  $\xi \in [0, 1]$ , and, in particular, a different value of the derivative  $f'(1)$ . In many cases this may be unacceptable from the physics point of view.

### 5. Symmetric monotonic piece-wise parabolic interpolation

*a. Case of  $\xi_2 > 1$ .* Similar to the previous subsection, assume that a function  $f(\xi)$  of a normalized variable  $\xi$  is known at three points

$$f(0) = 0, \quad f(1) = f_1, \quad f(\xi_2) = f_2, \quad (C.53)$$

and that  $\xi_2 > 1$ . We want to construct a piece-wise parabolic interpolation to  $f(\xi)$ , which is

- (i) continuous on the interval  $[0, \xi_2]$  and passes through all the three values (C.53);
- (ii) monotonic on each of the two intervals  $[0, 1]$  and  $[1, \xi_2]$ , i.e.  $\min\{0, f_1\} \leq f(\xi) \leq \max\{0, f_1\}$  for  $\xi \in [0, 1]$ , and  $\min\{f_1, f_2\} \leq f(\xi) \leq \max\{f_1, f_2\}$  for  $\xi \in [1, \xi_2]$  (a *strictly monotonic* interpolation);

- (iii) is smooth in the sense that it has a continuous derivative  $f'(\xi)$  at  $\xi = 1$  (this is needed for reconstruction of the diffusion limit when the source function  $B$  is interpolated along the optical depth  $\tau$ );
- (iv) is symmetric in the sense that if such an interpolation is constructed starting from  $\xi = \xi_2$  rather than from  $\xi = 0$  [i.e. with respect to variable  $\eta = (\xi_2 - \xi)/(\xi_2 - 1)$ ], it should yield the same values of  $f(\xi)$  for all  $0 \leq \xi \leq \xi_2$ .

Note that continuity of the second derivative at  $\xi = 1$  cannot be guaranteed under the conditions (ii)–(iv). Clearly, inside the normalized interval  $\xi \in [0, 1]$  our interpolation can be written in the form

$$f(\xi) = a\xi + \frac{1}{2}(b - a)\xi^2, \tag{C.54}$$

where

$$a + b = 2f_1. \tag{C.55}$$

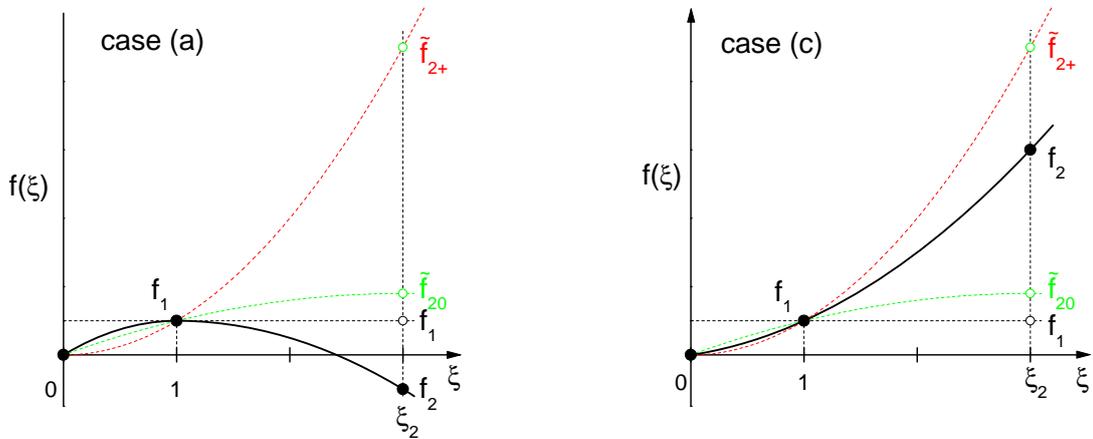


FIG. C.6: Symmetric piece-wise parabolic monotonic interpolation for  $f(\xi)$  on a normalized interval  $\xi \in [0, 1] + [1, \xi_2]$ . Shown are the cases (a) and (c).

The required interpolation is constructed by combining the following four cases:

- (a) Case  $f_2 \leq f_1$  and  $f_1 \geq 0$ , or  $f_2 \geq f_1$  and  $f_1 < 0$ :

$$f(\xi) = \begin{cases} f_1 \xi(2 - \xi), & 0 \leq \xi \leq 1, \\ f_1 + (f_2 - f_1) \left( \frac{\xi - 1}{\xi_2 - 1} \right)^2, & 1 < \xi \leq \xi_2; \end{cases} \tag{C.56}$$

in this case  $f'(1) = 0$ .

- (b) Case  $|f_1| < |f_2| < \frac{\xi_2^2}{2\xi_2 - 1}|f_1| \equiv |\tilde{f}_{20}|$ :

$$f(\xi) = \begin{cases} 2 \left( f_1 - \frac{f_2 - f_1}{\xi_2 - 1} \right) \xi + \left( 2 \frac{f_2 - f_1}{\xi_2 - 1} - f_1 \right) \xi^2, & 0 \leq \xi \leq 1, \\ f_1 + 2 \frac{f_2 - f_1}{\xi_2 - 1} (\xi - 1) \left( 1 - \frac{1}{2} \frac{\xi - 1}{\xi_2 - 1} \right), & 1 < \xi \leq \xi_2. \end{cases} \tag{C.57}$$

(c) Case  $|\tilde{f}_{20}| \equiv \frac{\xi_2^2}{2\xi_2 - 1}|f_1| \leq |f_2| \leq \xi_2^2|f_1| \equiv |\tilde{f}_{2+}|$ :

$$f(\xi) = \frac{f_1\xi_2^2 - f_2}{\xi_2(\xi_2 - 1)}\xi + \frac{f_2 - f_1\xi_2}{\xi_2(\xi_2 - 1)}\xi^2, \quad 0 \leq \xi \leq \xi_2; \quad (\text{C.58})$$

in this and only in this case the second derivative  $f''(\xi)$  is continuous at  $\xi = 1$ .

(d) Case  $|\tilde{f}_{2+}| \equiv \xi_2^2|f_1| < |f_2|$ :

$$f(\xi) = \begin{cases} f_1\xi^2, & 0 \leq \xi \leq 1, \\ f_1 + (\xi - 1) \left[ 2f_1 + \frac{f_2 - f_1(2\xi_2 - 1)}{(\xi_2 - 1)^2}(\xi - 1) \right], & 1 < \xi \leq \xi_2. \end{cases} \quad (\text{C.59})$$

Cases (b) and (c) match along the parabola 20 (green dashes in Fig. C.6), given by

$$f_{20}(\xi) = f_1 \frac{\xi(2\xi_2 - \xi)}{2\xi_2 - 1}, \quad (\text{C.60})$$

which reaches the value

$$\tilde{f}_{20} = \frac{f_1\xi_2^2}{2\xi_2 - 1} \quad (\text{C.61})$$

at  $\xi = \xi_2$ . Cases (c) and (d) match along the parabola 2+ (red dashes in Fig. C.6), given by

$$f_{2+}(\xi) = f_1\xi^2, \quad (\text{C.62})$$

which reaches the value

$$\tilde{f}_{2+} = f_1\xi_2^2 \quad (\text{C.63})$$

at  $\xi = \xi_2$ .

*b. Case of  $\xi_2 < 0$ .* When we have to treat the case of  $\xi_2 < 0$  (as in the case of interpolation along the  $\Omega_R$  projection of the photon propagation direction  $\vec{\Omega}$  in the  $rz$ -geometry; see Fig. C.7), the same interpolation inside the interval  $0 \leq \xi \leq 1$  can be cast in the following form

$$f(\xi) = a_1f_1 + a_2f_2, \quad (\text{C.64})$$

where coefficients  $a_1$  and  $a_2$  are given by

1) for  $[f_1 \geq 0 \wedge f_2 \leq f_2^* \equiv f_1\xi_2(2 - \xi_2)]$  or  $[f_1 \leq 0 \wedge f_2^* \leq f_2]$

$$a_1 = \xi(2 - \xi), \quad a_2 = 0; \quad (\text{C.65})$$

2) for  $[f_1 \geq 0 \wedge f_2^* \leq f_2 \leq f_2^0 \equiv f_1\xi_2^2/(2\xi_2 - 1)]$  or  $[f_1 \leq 0 \wedge f_2^0 \leq f_2 \leq f_2^*]$

$$a_1 = \frac{\xi(\xi_2 - \xi)}{\xi_2 - 1}, \quad a_2 = \frac{\xi(\xi - 1)}{\xi_2(\xi_2 - 1)}; \quad (\text{C.66})$$

3) for  $[f_1 \geq 0 \wedge f_2^0 \leq f_2 \leq 0]$  or  $[f_1 \leq 0 \wedge 0 \leq f_2 \leq f_2^0]$

$$a_1 = \xi^2, \quad a_2 = \frac{2\xi(1 - \xi)}{\xi_2}; \quad (\text{C.67})$$

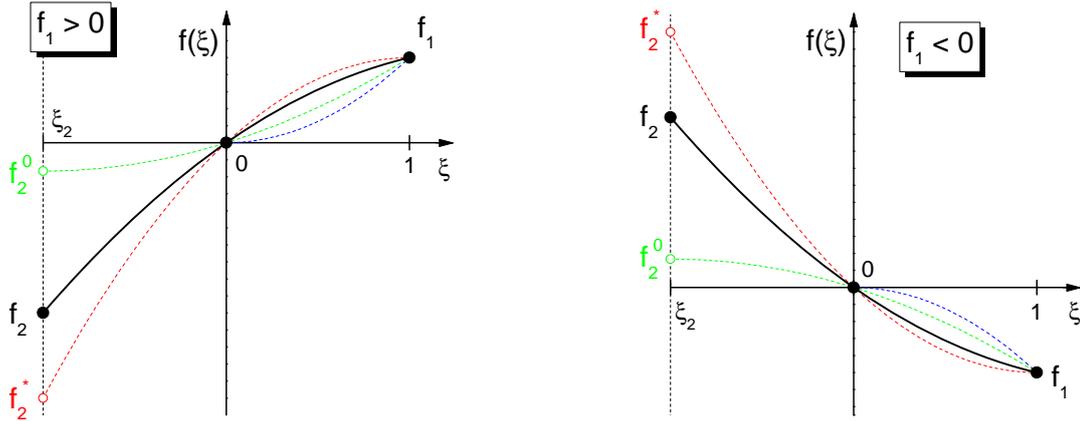


FIG. C.7: Symmetric piece-wise parabolic monotonic interpolation for  $f(\xi)$  on a normalized interval  $\xi \in [0, 1]$ .

4) for  $[f_1 \geq 0 \wedge 0 \leq f_2]$  or  $[f_1 \leq 0 \wedge f_2 \leq 0]$

$$a_1 = \xi_2^2, \quad a_2 = 0. \tag{C.68}$$

The hierarchy of the four relevant cases is illustrated in Fig. C.7. Cases 1) and 2) match at

$$f_2 = f_2^* = f_1 \xi_2 (2 - \xi_2) \tag{C.69}$$

along the parabola

$$f_{b^*}(\xi) = f_1 \xi (2 - \xi) \tag{C.70}$$

shown in Fig. C.7 with red dashes. Cases 2) and 3) match at

$$f_2 = f_2^0 = f_1 \frac{\xi_2^2}{2\xi_2 - 1} \tag{C.71}$$

along the parabola

$$f_{b0}(\xi) = f_1 \frac{\xi(2\xi_2 - \xi)}{2\xi_2 - 1} \tag{C.72}$$

shown in Fig. C.7 with green dashes. Cases 3) and 2) match at

$$f_2 = 0 \tag{C.73}$$

along the parabola

$$f_{b00}(\xi) = f_1 \xi^2 \tag{C.74}$$

shown in Fig. C.7 with blue dashes. Note that the second derivative  $f''(\xi)$  is continuous at  $\xi = 0$  only in case 2); the interpolation parabola (C.64) in this case is shown as a solid black curve in Fig. C.7.

The above formulae can be programmed as

**if**  $[(f_2 - f_1\xi_2(2 - \xi_2)) \cdot (f_2 - f_1\xi_2^2/(2\xi_2 - 1))] \leq 0$  **then**

$$a_1 = \frac{\xi(\xi_2 - \xi)}{\xi_2 - 1}, \quad a_2 = \frac{\xi(\xi - 1)}{\xi_2(\xi_2 - 1)},$$

**elseif**  $[f_2 \cdot (f_2 - f_1\xi_2^2/(2\xi_2 - 1))]$  **then**

$$a_1 = \xi^2, \quad a_2 = \frac{2\xi(1 - \xi)}{\xi_2},$$

**elseif**  $[f_1 \cdot f_2 \geq 0]$  **then**

$$a_1 = \xi^2, \quad a_2 = 0,$$

**else**

$$a_1 = \xi(2 - \xi), \quad a_2 = 0.$$

**endif**

- [1] F. L. Addessio, J. R. Baumgardner, J. K. Dukowicz, N. L. Johnson, B. A. Kashiwa, R. M. Rauenzahn, and C. Zemach, Report LA-10613-MS-Rev. 1, UC-32, Los Alamos National Laboratory (1992).
- [2] M. Basko, J. A. Maruhn, and A. Tauschwitz, Tech. Rep. 2007-05, GSI Helmholtzzentrum für Schwerionenforschung GmbH (2007).
- [3] M. M. Basko, J. A. Maruhn, and A. Tauschwitz, *Journal of Computational Physics* **228**, 2175 (2009), ISSN 0021-9991, URL <http://www.sciencedirect.com/science/article/pii/S0021999108006232>.
- [4] D. Mihalas and B. W. Mihalas, *Foundations of Radiation Hydrodynamics* (Oxford University Press, Oxford, 1984), ISBN 978-3764321833.
- [5] M. M. Basko, *DEIRA: A 1-D 3-T hydrodynamic code for simulating ICF targets driven by fast ion beams. Version 4* (2001; <http://www.basko.net/mm/deira>).
- [6] Y. B. Zeldovich and Y. P. Raizer, *Physics of Shock-Waves and High-Temperature Hydrodynamic Phenomena* (Dover Publ Inc, 2002), illustrated ed., ISBN 978-0486420028.
- [7] L. Spitzer, *Physics of Fully Ionized Gases* (Interscience, New York, 1962), 2nd ed.
- [8] V. S. Imshennik, *Soviet Astronomy – AJ* **5**, 495 (1962).
- [9] M. Lampe, *Phys. Rev.* **170**, 306 (1968).
- [10] M. Lampe, *Phys. Rev.* **174**, 276 (1968).
- [11] N. A. Bobrova and P. V. Sasorov, *Plasma Phys. Rep.* **19**, 409 (1993).
- [12] H. Brysk, P. M. Campbell, and P. Hammerling, *Plasma Physics* **17**, 473 (1975).
- [13] M. M. Basko, T. Löwer, V. N. Kondrashov, A. Kendl, R. Sigel, and J. Meyer-ter Vehn, *Physical Review E* **56**, 1019 (1997).
- [14] R. Ramis, J. Meyer-ter Vehn, and J. Ramírez, *Comp. Phys. Comm.* **180**, 977 (2009).
- [15] E. Livne and A. Glasner, *Journal of Computational Physics* **58**, 59 (1985), ISSN 0021-9991.
- [16] B. G. Carlson, in *Methods in Computational Physics*, edited by B. Alder, S. Fernbach, and M. Rotenberg (Academic Press, New York, 1963), vol. 1, pp. 1–42.
- [17] K. D. Lathrop, *Journal of Computational Physics* **4**, 475 (1969), ISSN 0021-9991.

- [18] E. W. Larsen, J. E. Morel, and W. F. Miller Jr., *Journal of Computational Physics* **69**, 283 (1987), ISSN 0021-9991.
- [19] A. Dedner and P. Vollmöller, *Journal of Computational Physics* **178**, 263 (2002), ISSN 0021-9991.
- [20] D. Balsara, *JQSRT* **69**, 671 (2001), ISSN 0022-4073.
- [21] B. Carlson, *Nucl. Sci. Eng.* **61**, 408 (1976).
- [22] J. Bruls, P. Vollmöller, and M. Schüssler, *Astron. Astrophys.* **348**, 233 (1999).
- [23] S. Chandrasekhar, *Radiative Transfer* (Dover, New York, 1960).
- [24] L. D. Landau and E. M. Lifshitz, *Electrodynamics of Continuous Media* (Pergamon Press, 1984), 2nd ed., ISBN 978-0750626347.
- [25] L. Pitaevskii and E. Lifshitz, *Physical Kinetics*, v. 10 (Elsevier Science, 2012), ISBN 9780080570495, URL <http://books.google.ru/books?id=DTHxPDfV0fQC>.
- [26] S. Eliezer, *The Interaction of High-Power Lasers with Plasmas*, Series in Plasma Physics (CRC Press, 2002), ISBN 9781420033380, URL <https://books.google.ru/books?id=Hqzjvpu4kEUC>.
- [27] M. Born, E. Wolf, A. Bhatia, P. Clemmow, D. Gabor, A. Stokes, A. Taylor, P. Wayman, and W. Wilcock, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light* (Cambridge University Press, 1999), ISBN 9781139643405, URL <https://books.google.ru/books?id=kURdAAAAQBAJ>.
- [28] G. Cisneros, J. S. Helman, and C. N. J. Wagner, *Phys. Rev. B* **25**, 4248 (1982).
- [29] H. Brysk, *Plasma Physics* **16**, 927 (1974).
- [30] W. Kruer, *The physics of laser plasma interactions*, Frontiers in physics (Addison-Wesley, 1988), ISBN 9780201156720, URL <http://books.google.ru/books?id=csDvAAAAAAAJ>.
- [31] L. D. Landau and E. M. Lifshitz, *Statistical Physics* (Butterworth Heinemann, 1996), 3rd ed., ISBN 978-0750633727.
- [32] P. Kunasz and L. Auer, *JQSRT* **39** (1988).
- [33] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, vol. 14 of *Springer Series in Computational Mathematics* (Springer-Verlag, 1991).
- [34] J. D. Lindl, *Inertial Confinement Physics* (Springer-Verlag, New York, 1998).
- [35] L. P. Bass, A. M. Voloshchenko, and T. A. Germogenova, KIAM report, Keldysh Institute of Applied Mathematics, Acad. Sci. USSR (1986), (In Russian).
- [36] F. Filbet, E. Sonnendrücker, and P. Bertrand, *Journal of Computational Physics* **172**, 166 (2001).