

MULTI2D – a computer code for two-dimensional radiation hydrodynamics [☆]

R. Ramis ^{a,*}, J. Meyer-ter-Vehn ^b, J. Ramírez ^a

^a E.T.S.I. Aeronáuticos, Universidad Politécnica de Madrid, Spain

^b Max-Planck-Institut für Quantenoptik, Garching, Germany

ARTICLE INFO

Article history:

Received 24 July 2008

Received in revised form 17 December 2008

Accepted 18 December 2008

Available online 25 December 2008

PACS:

47.11.-j

47.70.Mc

52.57.-z

52.57.Fg

Keywords:

Inertial confinement

Radiative transfer

Lagrangian hydrodynamics

Plasma physics

ABSTRACT

Simulation of radiation hydrodynamics in two spatial dimensions is developed, having in mind, in particular, target design for indirectly driven inertial confinement energy (IFE) and the interpretation of related experiments. Intense radiation pulses by laser or particle beams heat high-Z target configurations of different geometries and lead to a regime which is optically thick in some regions and optically thin in others. A diffusion description is inadequate in this situation. A new numerical code has been developed which describes hydrodynamics in two spatial dimensions (cylindrical R - Z geometry) and radiation transport along rays in three dimensions with the 4π solid angle discretized in direction. Matter moves on a non-structured mesh composed of trilateral and quadrilateral elements. Radiation flux of a given direction enters on two (one) sides of a triangle and leaves on the opposite side(s) in proportion to the viewing angles depending on the geometry. This scheme allows to propagate sharply edged beams without ray tracing, though at the price of some lateral diffusion. The algorithm treats correctly both the optically thin and optically thick regimes. A symmetric semi-implicit (SSI) method is used to guarantee numerical stability.

Program summary

Program title: MULTI2D

Catalogue identifier: AECV_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AECV_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: Standard CPC licence, <http://cpc.cs.qub.ac.uk/licence/licence.html>

No. of lines in distributed program, including test data, etc.: 151 098

No. of bytes in distributed program, including test data, etc.: 889 622

Distribution format: tar.gz

Programming language: C

Computer: PC (32 bits architecture)

Operating system: Linux/Unix

RAM: 2 Mbytes

Word size: 32 bits

Classification: 19.7

External routines: X-window standard library (libX11.so) and corresponding heading files (X11/*.h) are required.

Nature of problem: In inertial confinement fusion and related experiments with lasers and particle beams, energy transport by thermal radiation becomes important. Under these conditions, the radiation field strongly interacts with the hydrodynamic motion through emission and absorption processes.

Solution method: The equations of radiation transfer coupled with Lagrangian hydrodynamics, heat diffusion and beam tracing (laser or ions) are solved, in two-dimensional axial-symmetric geometry (R - Z coordinates) using a fractional step scheme. Radiation transfer is solved with angular resolution. Matter properties are either interpolated from tables (equations-of-state and opacities) or computed by user routines (conductivities and beam attenuation).

Restrictions: The code has been designed for typical conditions prevailing in inertial confinement fusion (ns time scale, matter states close to local thermodynamical equilibrium, negligible radiation pressure,

[☆] This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author.

E-mail address: rafael.ramis@upm.es (R. Ramis).

...). Although a wider range of situations can be treated, extrapolations to regions beyond this design range need special care.

Unusual features: A special computer language, called **r94**, is used at top levels of the code. These parts have to be converted to standard C by a translation program (supplied as part of the package). Due to the complexity of code (hydro-code, grid generation, user interface, graphic post-processor, translator program, installation scripts) extensive manuals are supplied as part of the package.

Running time: 567 seconds for the example supplied.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Indirectly irradiated targets for Inertial Confinement Fusion (ICF) [1,2] rely on radiative energy transport to induce the implosion of thermonuclear fuel to ignition conditions. Intense laser or ion beam power is converted into thermal radiation and is confined in a cavity (hohlraum) having high-Z walls. The almost isotropic radiation field produced in this way uniformly heats the external layers of a spherical capsule, filled with a deuterium–tritium mixture, and induces its hydrodynamic implosion to thermonuclear conditions. To quantitatively understand this process, as well as a great variety of related experiments [2–5], numerical simulations are required. Very different scales of length (from cm (cavity size) to μm (cavity wall thickness)), density (from 10^{-3} (laser absorption) to 10^3 g cm^{-3} (compressed fuel at ignition)), and temperature (from 1 (imploding fuel) to 10^5 eV (burning temperature)), are present during these processes and are relevant. In this paper, we deal with the numerical problem of multi-dimensional radiation transport. A new method is presented to treat this problem, which is robust, flexible, and efficient. For the basic physics of the problem, we refer the reader to [6].

In [7], MULTI, a one-dimensional code including hydrodynamics, radiative transfer, laser light deposition, and electronic heat flux, was presented. Although all the basic physical processes involved in ICF were included, the multi-dimensional aspects were missing, e.g., the radiation symmetrization inside cavities or the growth of hydrodynamics instabilities. In this paper we describe the most recent version of the two-dimensional code MULTI2D [8–10]. It treats (in axial-symmetric R – Z geometry) plasma hydrodynamic motion together with various energy transfer mechanisms: electronic heat conduction, thermal radiation transport, and laser or ion beam deposition. It uses unstructured grids composed of triangular elements. The different processes are organized in a fractional step scheme; state variables are advanced in sub-steps, in which only some of the physical processes are active. Special emphasis is put on conservation of physical quantities. This is particularly important in view of limited computer resources. Typically, the computational grid has to be chosen rather coarse in comparison with the scale length of the problem; for example, the sharp jump of fluids variables across shock fronts, whose physical thickness is several orders of magnitude smaller than grid spacing, are smoothed artificially by the numerical method to occur across several computational cells. The apparent shock thickness depends on the numerical method used (2–3 cells in our simulations) and can be reduced further only by increasing the number of grid elements. Under these circumstances, conservation principles warrant correct evaluation of global quantities, for example, the shock wave velocity. In this way, one avoids that numerical errors at some special grid locations spoil the results of the simulation as a whole.

In order to allow a maximum flexibility, the code has been organized as a package of subroutines. It is left to the user to design the main program: grid generation, initialization, and evaluation of matter properties by user routines: equations-of-state, opacities, attenuation coefficients for laser/ion beam interactions, and heat conductivities. Once all initial and boundary conditions have been defined, control is passed to the central integration routine.

To facilitate this work, documented utility libraries, examples, as well as a graphic user's interface (GUI) and miscellaneous post-processors have been developed and are supplied together with the code package. In the present paper, we shall describe the algorithms used in this code, with special emphasis on radiation transport (Sections 2 and 3) and Lagrangian hydrodynamic motion (Section 4). Thermal conduction and beam deposition are described in Section 5. The program structure and software organization is discussed in Section 6.

Finally, Section 7 is used to apply the code to an ICF target configuration of high current interest. It is similar to that designed at Livermore Lab [2,11] to demonstrate target ignition and gain at the National Ignition Facility (NIF). The simulation highlights both merits and present limitations of the code, by showing early phases of hohlraum heating and capsule implosion. Concerning the merits, the code describes the multi-dimensional aspects of radiative drive. This is crucial for the symmetry of capsule implosion. Concerning limitations, the code as described in this paper contains no fusion physics (ignition and burn not described), is restricted to one-frequency-group radiation transport, and is not designed for high-resolution studies of hydrodynamic instabilities. These limitations are not inherent to the new numerical methods presented here, but have been imposed for reasons of clear and focused presentation. It is straightforward to add the missing parts, and actually some corresponding subroutines are already included in the packages for download from the web [12].

2. Radiation transport

The central point of this paper is to give a correct description of directional radiative energy transfer with high angular resolution. This requires to go beyond the usual diffusive approximation, frequently used in 1D and 2D codes [7,13–16]. However, more sophisticated and accurate methods to solve the transport equation [17] demand extensive computer resources. Dealing with problems like ICF, in which drastically different optical thicknesses and thermodynamic properties occur simultaneously, robust, fast, and efficient algorithms are needed.

In the ICF context, matter velocity is typically smaller than the speed of light, so that the radiation field can be regarded as quasi-static at any instant of time. The spectral radiation intensity $I_\nu(\vec{r}, \vec{n})$ (energy flux per unit of solid angle and frequency) depends on position \vec{r} , propagation direction \vec{n} , and frequency ν . For conditions not too far away from local thermodynamical equilibrium, I_ν is determined (neglecting scattering and refraction) by the radiative transport equation [18,19]:

$$\vec{n} \cdot \nabla I_\nu = \frac{I_\nu^p - I_\nu}{\lambda'_\nu} \equiv -q_\nu, \quad (1)$$

where λ'_ν is the mean free path for radiation (taking into account stimulated emission), q_ν the power deposited in matter (per unit of volume, solid angle, and frequency), and I_ν^p is Planck's distribution function given by

$$I_\nu^p = \frac{2h\nu^3}{c^2(e^{h\nu/kT} - 1)}, \quad (2)$$

where h and k are Planck's and Boltzmann's constants, and c denotes velocity of light. The mean free path λ'_ν is assumed to depend only on frequency and position, but not on radiation direction \vec{n} .¹ It has to be computed by appropriate atomic physics models as a function of matter composition, density ρ , and temperature T (see, for example, Ch. 10 of [6]). The density of radiation energy U and the total flux of radiation \vec{S} , at a given position, are obtained by integrating I_ν/c and $I_\nu\vec{n}$ over all frequencies and directions

$$\frac{4\sigma T_R^4}{c} \equiv U = \frac{1}{c} \int_{\vec{n}} \int_{\nu} I_\nu d\nu d\vec{n}, \quad \vec{S} = \int_{\vec{n}} \int_{\nu} I_\nu \vec{n} d\nu d\vec{n}. \quad (3)$$

Here $\sigma = 2\pi^5 k^4 / 15h^3 c^2$ is the Stefan–Boltzmann constant, and T_R is called radiation temperature. We consider here the grey approximation, where one assumes a frequency-averaged value λ instead of the frequency-dependent λ'_ν . Integrating Eq. (1) over all frequencies and assuming that I_ν is not too far from the Planck distribution I_ν^P , one obtains

$$\vec{n} \cdot \nabla I = \frac{I^P - I}{\lambda} = -q, \quad (4)$$

where $I(\vec{r}, \vec{n})$, $q(\vec{r})$, and $I^P(\vec{r}) = \sigma T^4 / \pi$ represent integrals of $I_\nu(\vec{r}, \vec{n})$, $q_\nu(\vec{r})$, $I_\nu^P(\vec{r})$ over frequency, respectively, and the frequency-averaged mean free path λ is defined by

$$I^P / \lambda = \int_0^\infty (I_\nu^P / \lambda'_\nu) d\nu. \quad (5)$$

It is often referred to as the Planck mean free path (for basic approximations and frequency averages, see [6], Section 7.3). Eq. (4) can be integrated again over all directions to obtain the energy conservation equation

$$\nabla \cdot \vec{S} = \frac{1}{\lambda} \left(4\pi I^P - \int_{\vec{n}} I d\vec{n} \right) = \frac{4\sigma T^4 - cU}{\lambda} = -Q, \quad (6)$$

where Q is the power deposited into matter per unit of volume. If mean free path and temperature are known functions of position, Eq. (4) can be integrated over straight lines ($\vec{r} = \vec{r}_0 + l\vec{n}$, l being the length from point \vec{r}_0):

$$I(\vec{r}_0 + l\vec{n}, \vec{n}) = I(\vec{r}_0, \vec{n})e^{-\tau} + \int_0^\tau I^P(\vec{r}_0 + l'\vec{n})e^{-(\tau-l')} d\tau'. \quad (7)$$

Here τ is the optical depth along the distance l :

$$\tau(l) = \int_0^l \frac{dl'}{\lambda(\vec{r}_0 + l'\vec{n})}. \quad (8)$$

The first term on the right-hand side of Eq. (7) corresponds to the attenuation of the radiation flux at \vec{r}_0 ; the second term is due to radiation sources located between 0 and τ , each attenuated as function of the optical depth between source and point \vec{r} . A special case arises when λ is small compared to the characteristic length of the problem. The first term then vanishes very fast, and the second one can be approximated by expanding

$$I^P(\tau') = (I^P)_{\tau'=\tau} + \left(\frac{\partial I^P}{\partial \tau} \right)_{\tau'=\tau} (\tau' - \tau) + \dots, \quad (9)$$

to obtain the radiation field as

$$I \simeq I^P - \lambda \vec{n} \cdot \nabla I^P. \quad (10)$$

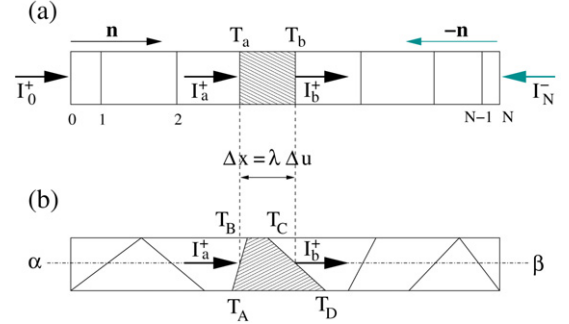


Fig. 1. Pure one-dimensional radiation transport.

Using this expression in Eq. (3), one obtains $T_R \simeq T$ and

$$\vec{S} \simeq -\lambda \left(\int_{\vec{n}} \vec{n} \vec{n} d\vec{n} \right) \cdot \nabla I^P = -\frac{4\pi}{3} \lambda \nabla I^P \simeq -\frac{16\lambda\sigma T^3}{3\pi} \nabla T \quad (11)$$

Here, integrating the tensor $\vec{n} \vec{n}$ over all space directions gives $(4\pi/3)\vec{u} \cdot \vec{u}$, \vec{u} being the unit tensor.

Although the formal solution of the transfer Eq. (4) is given by Eq. (7), this procedure (called the *long characteristics method* [17,18]) cannot be implemented directly into an ICF code because it would require, in order to avoid numerical noise, a separation between *characteristics* lines smaller than the minimum size of the mesh; typical Lagrangian grids are very distorted, with some of the cells having an extremely small thickness, resulting in a prohibitive number of lines. In order to clarify the algorithm used by MULTI2D, we will discuss below situations of increasing complexity before considering the 2D axial-symmetric geometry used by the code.

2.1. One-dimensional transport

We start describing the mathematical problem of pure one-dimensional transport (in euclidean space \mathbb{R}^1), where propagation can occur only in two opposite directions.² The frequency-integrated radiation field is described by two intensities in each space point: I^+ (in direction \vec{n}) and I^- (in direction $-\vec{n}$). Under this conditions, radiation density of energy and flux are given by $U = (I^+ + I^-)/c$ and $\vec{S} = (I^+ - I^-)\vec{n}$ instead of Eq. (3). For optically thick regions one has $\vec{S} \simeq -2\lambda dI^P/dx$ instead of Eq. (11). We consider a grid formed by N rectangles aligned along the direction of propagation, and assume that temperature is known at interfaces (see Fig. 1a). I^P , depending only on temperature T , is also known at the interfaces, and piecewise continuous linear variation with optical depth is assumed. Considering one of the cells between interfaces a and b , one has

$$I^P(\tau) = I_a^P + \frac{I_b^P - I_a^P}{\Delta\tau} \tau \equiv I_a^P + \frac{dI^P}{d\tau} \tau, \quad (12)$$

where τ is the optical depth measured from a , $\Delta\tau$ the cell optical depth, and I_a^P and I_b^P are the source function values at interfaces. Applying Eq. (7), one gets immediately the radiation intensity I^+ in direction \vec{n}

$$I^+(\tau) = \left(I_a^+ - I_a^P + \frac{dI^P}{d\tau} \right) e^{-\tau} + I_a^P + (\tau - 1) \frac{dI^P}{d\tau}, \quad (13)$$

where I_a^+ is the incoming intensity (see Fig. 1). This expression allows to compute the exiting intensity $I_b^+ \equiv I^+(\Delta\tau)$

$$I_b^+ = (I_a^+ - I_a^P) e^{-\Delta\tau} + I_b^P - (1 - e^{-\Delta\tau}) \frac{I_b^P - I_a^P}{\Delta\tau}. \quad (14)$$

¹ Dependence on \vec{n} would occur, for example, by Doppler shift, when fluid velocity direction breaks space isotropy.

² This is equivalent to the so-called “forward-reverse” approximation used to model some 1D problems [20].

Now, by sweeping from the leftmost cell (see Fig. 1), where the incoming intensity must be specified as a boundary condition, the intensity can be determined everywhere. The radiative power coupled to matter is, for radiation in direction \vec{n} , obtained from Eq. (4)

$$q^+ = -\vec{n} \cdot \nabla I^+ = -\frac{dI^+}{dx} = \frac{1}{\lambda} \left(I_a^+ - I_a^P + \frac{dI^P}{d\tau} \right) e^{-\tau} - \frac{1}{\lambda} \frac{dI^P}{d\tau}. \quad (15)$$

The absorbed power for the cell (per unit of transversal area) is

$$Q^+ = \int_a^b q^+ dx = I_a^+ - I_b^+. \quad (16)$$

On the other hand, the average value of I^+ in a cell is obtained from Eq. (13)

$$\bar{I}^+ = \frac{1}{\Delta\tau} \int_0^{\Delta\tau} I^+(\tau) d\tau = \frac{I_a^P + I_b^P}{2} - \frac{I_b^+ - I_a^+}{\Delta\tau}. \quad (17)$$

Combining the last two equations, we get

$$\frac{Q^+}{\Delta x} = \frac{\bar{I}^+ - \bar{I}^P}{\lambda_c}, \quad (18)$$

where $\bar{I}^P \equiv \frac{1}{2}(I_a^P + I_b^P)$, Δx is the cell thickness, and λ_c is the averaged opacity in the cell ($\lambda_c = \Delta x / \int_a^b \lambda^{-1}(x) dx$). Because we consider temperature defined at interfaces, it is natural to define also the deposition of energy at interfaces. For optically thin cells ($\Delta\tau < 1$), deposition is a smooth function, so that Q^+ can be divided into two equal parts. For $\Delta\tau > 1$, deposition is not smooth any more. Eq. (15) shows that, for large $\Delta\tau$, deposition takes place in a region of thickness of order unity, close to cell entrance (the second term in this equation ($\frac{dI^P}{d\tau}/\lambda$) will cancel exactly with a similar one originating by the transport in opposite direction $-\vec{n}$). We observe that we can write the deposition of energy in a cell also as

$$Q^+ = \left[I_a^+ - I_a^P + \frac{dI^P}{d\tau} \right] - \left[I_b^+ - I_b^P + \frac{dI^P}{d\tau} \right] - \Delta\tau \frac{dI^P}{d\tau}. \quad (19)$$

This expression suggests that, for $\Delta\tau > 1$, one can assign the first term to interface a , the second one to interface b , and ignore the third one (the integral of $\frac{dI^P}{d\tau}/\lambda$). Using this approach, for $\Delta\tau \rightarrow \infty$, the second term vanishes (see Eq. (14)) and all coupling goes to the cell entrance. In this situation, because I^+ and I^P are continuous functions, deposition at an internal interface ($1 < i < N$) is equal to the change of the slope of I^P in the adjacent cells. Considering both propagating directions, we have at interface i

$$Q_i = 2 \left(\frac{dI^P}{d\tau} \right)_{i+1/2} - 2 \left(\frac{dI^P}{d\tau} \right)_{i-1/2} = 2\lambda_{i+1/2} \frac{I_{i+1}^P - I_i^P}{\Delta_{i+1/2}x} - 2\lambda_{i-1/2} \frac{I_i^P - I_{i-1}^P}{\Delta_{i-1/2}x}. \quad (20)$$

This expression is the usual three-point discretization of the right-hand term of the diffusive transport equation $\partial E / \partial t = \nabla \cdot 2\lambda \nabla I^P$. The outlined algorithm has the following very desirable properties:

- (i) I^+ and I^- are easily obtained from interface temperatures and incident radiation at the boundaries, I_0^+ and I_N^- .
- (ii) I^+ and I^- are guaranteed to be positive everywhere.
- (iii) Energy is conserved. Adding Eq. (16) for all cells and for both propagation directions,

$$\sum_{i=0}^N Q_i = (I_0^+ + I_N^-) - (I_N^+ + I_0^-) = \text{input} - \text{output}. \quad (21)$$

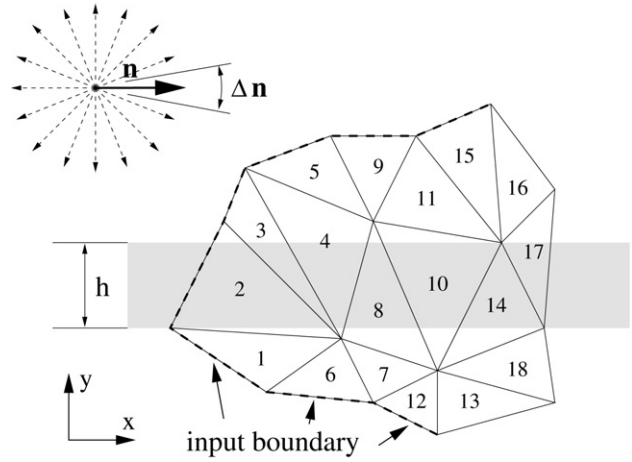


Fig. 2. Pure two-dimensional radiative transfer.

- (iv) Both for optically thin and optically thick regions, deposition is correctly determined by Eqs. (18) and (20), respectively.

The algorithm can be extended to one-dimensional propagation through a grid formed by trapezoidal elements filling the same region as before, but with temperatures defined at quadrangle vertices (see Fig. 1b). The radiation intensity is approximated by its value at the center of interfaces. I^P is computed first at vertices, the value of I^P at interface center is obtained by averaging the values at its ends; the algorithm is applied to obtain I^+ and I^- over the central line $\alpha\beta$. Finally the power coupling to matter, computed at each interface center, is distributed in equal parts to its ends.

2.2. Two-dimensional planar transport

The method outlined above can be extended to ideal planar radiation transport, where only propagation in directions parallel to a plane is allowed. In this situation, integrals in Eq. (3) should be extended over ordinary angles rather than solid angles, while Eq. (11) now becomes $\bar{S} \simeq -\pi\lambda \nabla I^P$. We consider a 2D planar grid composed of triangular cells with temperatures defined at the nodes (see Fig. 2), and radiative mean free path λ constant in each cell. The direction space is discretized by subdividing the unit circle by an even number N_n of equal parts of size $\Delta\vec{n} = 2\pi/N_n$. Transfer Eq. (4) for each direction is independent of the others, and can be solved separately. In principle, one could apply the 1D procedure of Section 2.1 for each direction by cutting the computational domain in parallel bands (shaded area in Fig. 2). The value of I^P could be defined, in each interface, as the average of values at its ends, while the deposition of radiation, computed at interfaces, would be distributed between the nodes at their ends. This approach is not convenient in distorted non-structured grids, because it will require as many bands as nodes. The computational effort, scaling with the number of triangles raised to the power 3/2, would be prohibitive. Instead, we take radiation intensity defined at cell interfaces. For a given propagation direction \vec{n} , we have two cell types: **a**, cells with radiation entering through one of the sides, and exiting through the other two (Fig. 3), **b**, cells with radiation entering through two sides, and exiting by the third (Fig. 4). We divide each cell into two parts by line AA' parallel to \vec{n} , and apply the concepts outlined above for each part. For cells of type **a**, the exit intensities are obtained by applying Eq. (14)

$$I_{AB} = (I_{BC} - I_{BC}^P) e^{-\Delta\tau} + I_{AB}^P - (1 - e^{-\Delta\tau}) \frac{I_{AB}^P - I_{BC}^P}{\Delta\tau}, \quad (22)$$

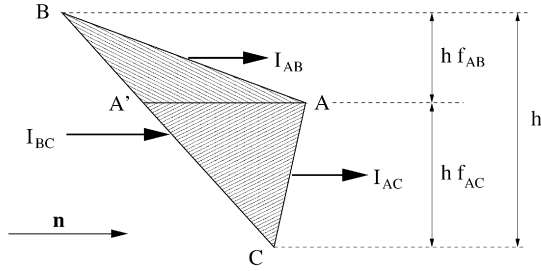


Fig. 3. Geometry of a cell of type a.

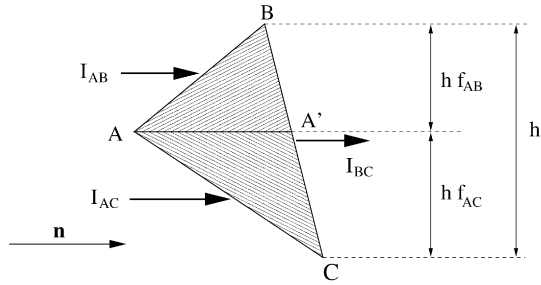


Fig. 4. Geometry of a cell of type b.

$$I_{AC} = (I_{BC} - I_{BC}^P)e^{-\Delta\tau} + I_{AC}^P - (1 - e^{-\Delta\tau}) \frac{I_{AC}^P - I_{BC}^P}{\Delta\tau}, \quad (23)$$

where I_{ij} is the radiation intensity over the side between points i and j , $I_{ij}^P = \frac{1}{2}(I_i^P + I_j^P)$ is the averaged Planck intensity on this side, and $\Delta\tau$ the average optical depth of the cell (area divided by the height h and by λ). For cells of type **b**, we would obtain different values on the two halves of the output interface

$$I_{A'B} = (I_{AB} - I_{AB}^P)e^{-\Delta\tau} + I_{BC}^P - (1 - e^{-\Delta\tau}) \frac{I_{BC}^P - I_{AB}^P}{\Delta\tau}, \quad (24)$$

$$I_{A'C} = (I_{AC} - I_{AC}^P)e^{-\Delta\tau} + I_{BC}^P - (1 - e^{-\Delta\tau}) \frac{I_{BC}^P - I_{AC}^P}{\Delta\tau}. \quad (25)$$

The averaged exit intensity on side BC is obtained using f_{AB} and f_{AC} (ratios between the heights of triangles $AA'B$ and $AA'C$ to the total cell height h) as weighting factors

$$I_{BC} = f_{AB}I_{A'B} + f_{AC}I_{A'C}. \quad (26)$$

Using Eqs. (22)–(26), the intensity of radiation in direction \vec{n} can be computed at all interfaces, starting at entrance boundary interfaces (dashed line in Fig. 2), where an external source (if any) can be specified, and proceeding in the appropriate order (i.e. following the numbers in Fig. 2). The power deposited in a cell is obtained adding the contributions of the two halves

$$Q = \pm(I_{BC} - f_{AB}I_{A'B} - f_{AC}I_{A'C})h\Delta\vec{n} \equiv - \sum_{k \text{ sides}} \vec{L}_k \cdot \vec{n} I_k \Delta\vec{n}, \quad (27)$$

where the sign is positive/negative for cells of type **a** and **b**, respectively, and \vec{L}_k is a vector perpendicular to triangle side k , with modulus equal to its length, and pointing outward. We generalize the procedure of Section 2.1, computing first the power at the interfaces (from Eq. (19))

$$Q_k = -\vec{L}_k \cdot \vec{n} (I_k - I_k^P + \lambda \vec{n} \cdot \nabla I^P) \Delta\vec{n}. \quad (28)$$

For optically thick cells ($\Delta\tau > 1$), the quantity Q_k is distributed in equal parts between both ends of side k ; for optically thin cells ($\Delta\tau < 1$), in equal parts between all cell vertices. Notice that adding depositions on the three sides we have

$$\sum_{k \text{ sides}} Q_k = Q + \vec{n} \cdot \sum_{k \text{ sides}} I_k^P \vec{L}_k \Delta\vec{n} \neq Q. \quad (29)$$

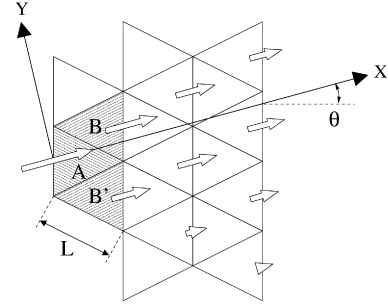


Fig. 5. Numerical lateral spread of radiation.

This apparent discrepancy in power deposition disappears when contributions of two opposite directions (\vec{n} and $-\vec{n}$) are added. The average value of I can be evaluated by applying Eq. (17) to cell halves and weighting them with factors f_{AB} and f_{AC} to get

$$\bar{I} = \frac{f_{AB}I_{AB}^P + f_{AC}I_{AC}^P + I_{BC}^P}{2} + \frac{Q}{h\Delta\tau\Delta\vec{n}}. \quad (30)$$

The total deposition of radiative transfer in one cell is obtained adding the contributions of all directions

$$Q_{\text{cell}} = \sum_{\vec{n}} Q \vec{n} = \frac{A}{\lambda} \left(\sum_{\vec{n}} \bar{I} \Delta\vec{n} - 2\pi \sum_{i=A,B,C} w_i I_i^P \right), \quad (31)$$

where the upper index \vec{n} indicates the direction, and the weighting factors w_i , satisfying $w_A + w_B + w_C = 1$, are given by

$$w_i = \sum_{\vec{n}} \frac{|\vec{r}_{ij} \times \vec{n}| + |\vec{r}_{ik} \times \vec{n}|}{\max(|\vec{r}_{ij} \times \vec{n}|, |\vec{r}_{ik} \times \vec{n}|, |\vec{r}_{jk} \times \vec{n}|)} \frac{\Delta\vec{n}}{8\pi}, \quad (32)$$

with $\vec{r}_{ij} \equiv \vec{r}_i - \vec{r}_j$, and j and k being the other two vertices of the triangle. In the limit $\Delta\vec{n} \rightarrow 0$, this expression becomes

$$w_i = \frac{1}{4\pi} \left(\pi + \frac{\beta \cot \alpha - \ln(\sin \gamma / \sin \alpha)}{\cot \gamma + \cot \alpha} + \frac{\gamma \cot \alpha - \ln(\sin \beta / \sin \alpha)}{\cot \beta + \cot \alpha} \right), \quad (33)$$

with $\alpha = \widehat{jik}$, $\beta = \widehat{ijk}$, and $\gamma = \widehat{ikj}$, the internal angles in each vertex. In the case of an equilateral triangle, $w_i = \frac{1}{3}$.

For the optically thin regime, the described algorithm produces a lateral spread of the radiation. This fact is illustrated in Fig. 5, where, over a regular grid of equilateral triangles of side L , absorption is considered negligible ($\Delta\tau \approx 0$). A ray with direction defined by angle $\theta \in [-\frac{\pi}{6}, \frac{\pi}{6}]$, impinges on triangle A of type **a**. The output intensities of this triangle, given by Eqs. (22)–(23), are equal, for $\Delta\tau \approx 0$, to entrance values. These intensities are coupled as input to triangles B and B' of type **b**, whose output intensities, given by Eq. (24)–(26), are $(1 \pm \sqrt{3} \tan \theta)/2$ (relative to entrance value). In this way, radiation flux is divided in $M + 1$ components after passing through M layers of cells. Asymptotically, it can be shown that, for $x, y \gg L$, the intensity distribution takes the form

$$I(x, y) \simeq \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right), \quad (34)$$

with

$$\sigma^2 = \frac{xL}{2\sqrt{3}} \cos^3 \theta (1 - 3 \tan^2 \theta). \quad (35)$$

Notice that for propagation along directions parallel to triangle sides (i.e. $\theta = \pm \frac{\pi}{6}$) lateral diffusion disappears. For a more realistic situation, characterized by an irregularly distorted grid, analysis is

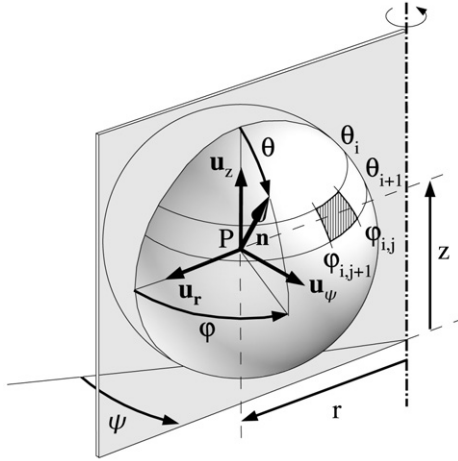


Fig. 6. Cylindrical coordinate system.

more complicated, but one can estimate roughly the lateral spread by averaging Eq. (35) over all angles

$$\langle \sigma^2 \rangle \simeq \frac{3}{\pi} \int_{-\pi/6}^{\pi/6} \sigma^2(\theta) d\theta = \frac{xL}{\pi\sqrt{3}}. \quad (36)$$

In any case, the exact solution of Eq. (1) is correctly obtained in the limit $L \rightarrow 0$, while Eq. (31) guarantees a consistent coupling of radiation with matter.

On the other hand, for the optically thick regime, in each internal interface, the two contributions coming from adjacent cells are given by Eq. (28) with opposite values of \vec{L}_k . The total power deposited at the interface can be written as

$$Q_k = \sum_{\vec{n}} \vec{L}_k \cdot \vec{n} (\lambda_+ \vec{n} \cdot (\nabla I^P)_+ - \lambda_- \vec{n} \cdot (\nabla I^P)_-) \Delta \vec{n} = \vec{L}_k \cdot (\vec{S}_- - \vec{S}_+), \quad \text{with } \vec{S} = -\pi \lambda \nabla I^P, \quad (37)$$

where \vec{L}_k is directed from cell “–” to cell “+”, and, for simplicity, N_n has been assumed to be a multiple of 4. Finally this value will be distributed to the nodes of the interface. This result is the one that would be obtained when diffusive transport (Eq. (11)) is discretized by a Galerkin-like method, such as the one described in Section 5 for the diffusive heat flux: \vec{S} is assumed constant in each cell and discontinuous at interfaces, the jump of energy flux being distributed between nodes. We can thus conclude that the optically thick regime is also correctly recovered and that the points listed at the end of Section 2.1 are satisfied for planar geometry.

2.3. Axial symmetric transport

We consider the cylindrical coordinates r , z , and ψ , shown in Fig. 6. For axially symmetric problems, quantities do not depend on ψ , so that radiation intensity can be written as $I(r, z, \theta, \varphi)$, where angles θ and φ define the direction of radiation propagation:

$$\vec{n} = \cos \theta \vec{u}_z + \sin \theta \cos \varphi \vec{u}_r + \sin \theta \sin \varphi \vec{u}_\psi. \quad (38)$$

In this geometry, transfer Eq. (4) takes the form:

$$\sin \theta \left(\cos \varphi \frac{\partial I}{\partial r} - \frac{\sin \varphi}{r} \frac{\partial I}{\partial \varphi} \right) + \cos \theta \frac{\partial I}{\partial z} = \frac{I^P - I}{\lambda}. \quad (39)$$

The radiation field will be discretized by dividing the 4π -sphere of directions into a finite number of domains of constant intensity (dashed area in Fig. 6). For clarity, the derivation will be done in two steps. First, we subdivide the range of variation of

θ into M intervals, assuming no dependence on θ in each interval: $I(r, z, \theta, \varphi) = I_i(r, z, \varphi)$ for $\theta_i < \theta < \theta_{i+1}$ (with $\theta_1 = 0$ and $\theta_{M+1} = \pi$). Multiplying Eq. (39) by $\sin \theta d\theta$ (lowest order moment, in order to conserve energy) and integrating over each interval, one arrives at a set of independent equations

$$\alpha_i \left(\cos \varphi \frac{\partial I_i}{\partial r} - \frac{\sin \varphi}{r} \frac{\partial I_i}{\partial \varphi} \right) + \beta_i \frac{\partial I_i}{\partial z} = \gamma_i \frac{I^P - I_i}{\lambda}, \quad (40)$$

with coefficients

$$\alpha_i \equiv \frac{\theta_{i+1} - \theta_i}{2} + \frac{\sin 2\theta_i - \sin 2\theta_{i+1}}{4},$$

$$\beta_i \equiv \frac{\cos 2\theta_i - \cos 2\theta_{i+1}}{4},$$

$$\gamma_i \equiv \cos \theta_i - \cos \theta_{i+1}. \quad (41)$$

Secondly, for each value of i , we subdivide the range of variation of φ into a number of intervals N_i (in general depending on i). Because the problem is symmetric with respect to a plane containing the z -axis ($I(r, z, \theta, \varphi) = I(r, z, \theta, -\varphi)$), we need only to consider the range $-\pi \leq \varphi \leq 0$. We integrate Eq. (40) over each interval, assuming a constant value $I_i(r, z, \varphi) = I_{ij}(r, z)$ for $\varphi_{ij} < \varphi < \varphi_{i,j+1}$ (with $\varphi_{i1} = -\pi$ and $\varphi_{i,N_i+1} = 0$) to obtain

$$A_{i1} \frac{\partial I_{i1}}{\partial r} + C_{i1} \frac{\partial I_{i1}}{\partial z} = D_{i1} \frac{I^P - I_{i1}}{\lambda}, \quad (42)$$

for $j = 1$, and

$$A_{ij} \frac{\partial I_{ij}}{\partial r} + B_{i,j-1} \frac{I_{ij} - I_{i,j-1}}{r} + C_{ij} \frac{\partial I_{ij}}{\partial z} = D_{ij} \frac{I^P - I_{ij}}{\lambda}, \quad (43)$$

for $j > 1$, where the coefficients

$$A_{ij} \equiv \alpha_i (\sin \varphi_{i,j+1} - \sin \varphi_{ij}),$$

$$B_{ij} \equiv -\alpha_i \sin \varphi_{i,j+1},$$

$$C_{ij} \equiv \beta_i (\varphi_{i,j+1} - \varphi_{ij}),$$

$$D_{ij} \equiv \gamma_i (\varphi_{i,j+1} - \varphi_{ij}), \quad (44)$$

satisfy

$$\sum_j A_{ij} = 0, \quad \sum_i \sum_j C_{ij} = \sum_i \pi \beta_i = 0,$$

$$\sum_i \sum_j D_{ij} = \sum_i \pi \gamma_i = 2\pi. \quad (45)$$

To obtain Eqs. (42)–(43) from Eq. (40), one has to integrate the term $\partial I_i / \partial \varphi$, singular at interval boundary φ_{ij} because I_i has been assumed discontinuous there. Although the integral of such term is $I_{ij} - I_{i,j-1}$, is not clear how to distribute this value between intervals j and $j - 1$. To solve this ambiguity, we use the following physical argument: this term takes into account that, for a photon moving along a straight line, φ is increasing (for $\varphi < 0$), until the boundary value φ_{ij} is reached and the photon passes from interval $j - 1$ to interval j . It is reasonable to assume that equation for interval j will contain the intensity of interval $j - 1$ (from where photons are coming) but not the intensity of interval $j + 1$ (to where photons are going). This assumption, together with the condition that $I = I^P = \text{constant}$ must be a valid solution, determines completely Eqs. (42)–(43). Current implementation in the code³ uses a uniform subdivision of θ and φ ranges, and requires M and N_i to be even and to verify the condition $N_i = N_{M-i+1}$. In order to get an equation in conservative form, suitable to be discretized in R - Z space, a new variable is defined:

³ The set of values of N_i are passed to the radiation routine as an integer array called **nphis**.

$\Gamma_{ij} \equiv 2\pi r I_{ij}$ for $j > 0$, and $\Gamma_{i0} \equiv 0$ for $j = 0$. Transfer equations take the form

$$A_{ij} \frac{\partial \Gamma_{ij}}{\partial r} + \frac{B_{ij} \Gamma_{ij}}{r} - \frac{B_{i,j-1} \Gamma_{i,j-1}}{r} + C_{ij} \frac{\partial \Gamma_{ij}}{\partial z} = D_{ij} \frac{\Gamma^P - \Gamma_{ij}}{\lambda}, \quad (46)$$

with $\Gamma^P \equiv 2\pi r I^P$. This equation has a structure similar to the discretization of Eq. (4), discussed for planar geometry in the previous section. The vector

$$\vec{n}_{ij} \equiv \frac{A_{ij} \vec{u}_r + C_{ij} \vec{u}_z}{E_{ij}} \quad (E_{ij} \equiv \sqrt{A_{ij}^2 + C_{ij}^2}) \quad (47)$$

plays the role of the unit vector in the propagation direction of radiation component Γ_{ij} . The two terms containing r can be interpreted as an attenuation term and a source term, originating geometrically from exit or entrance of photons in the direction domain. For $r \rightarrow \infty$, both terms vanish, and the planar 2D transport is recovered. This similarity can be emphasized by writing the transfer equations as

$$\vec{n}_{ij} \cdot \left(\frac{\partial \Gamma_{ij}}{\partial r} \vec{u}_r + \frac{\partial \Gamma_{ij}}{\partial z} \vec{u}_z \right) = \frac{\Gamma_{ij}^E - \Gamma_{ij}}{\lambda_{ij}^E}, \quad (48)$$

with

$$\lambda_{ij}^E \equiv \frac{E_{ij}}{D_{ij}/\lambda + B_{ij}/r}, \quad (49)$$

$$\Gamma_{ij}^E \equiv \frac{\lambda_{ij}^E}{E_{ij}} \left(\frac{D_{ij} \Gamma^P}{\lambda} + \frac{B_{i,j-1} \Gamma_{i,j-1}}{r} \right), \quad (50)$$

where λ_{ij}^E and Γ_{ij}^E are *effective* mean-free-path and source term. This set of equations can be solved by the method described in Section 2.2. A grid of triangular cells in the plane R – Z is considered, with temperatures defined at nodes and opacities assumed uniform inside cells. For each value of i , we have a coupled set of equations from $j = 1$ to $j = N_i$, that has to be solved in this order. For each direction \vec{n}_{ij} , the cells must be sorted before computing exit values of Γ_{ij} from entry values in each cell. λ_{ij}^E and the corresponding optical depths $\Delta \tau_{ij}^E$ at cells are computed assuming an average value of radius in Eq. (49)

$$\bar{r} = \frac{r_A + r_B + r_C}{3}. \quad (51)$$

The value of Γ_k^P at edge k is computed by averaging the values at edge ends A and B , for example, by taking

$$\Gamma_k^P = 2\pi \times \frac{1}{2}(r_A + r_B) \times \frac{1}{2} \left(\frac{\sigma T_A^4}{\pi} + \frac{\sigma T_B^4}{\pi} \right). \quad (52)$$

The values of $\Gamma_{ij,k}^E$ at cell edges are computed by Eq. (50), but using Γ_k^P in the place of Γ^P , and the cell average values \bar{r} and $\bar{\Gamma}_{i,j-1}$ in the place of r and $\Gamma_{i,j-1}$. The value $\bar{\Gamma}_{i,j-1}$ was made available when previous⁴ direction $\vec{n}_{i,j-1}$ was solved. With these values, Eqs. (22)–(26) allow to determine successively $\Gamma_{ij,k}$ at interfaces (using $\Delta \tau_{ij}^E$, $\Gamma_{ij,k}^E$ and $\Gamma_{ij,k}$ in the place of $\Delta \tau$, I_k^P and I_k). On the other hand, the average value $\bar{\Gamma}_{ij}$, needed for the next direction, is computed in a similar way by using Eqs. (27) and (30)

$$\bar{\Gamma}_{ij} = \frac{f_{AB} \Gamma_{ij,AB}^E + f_{AC} \Gamma_{ij,AC}^E + \Gamma_{ij,BC}^E}{2} \pm \frac{\Gamma_{ij,BC} - f_{AB} \Gamma_{ij,AB} - f_{AC} \Gamma_{ij,AC}}{\Delta \tau_{ij}^E}. \quad (53)$$

Deposition of energy in one cell can be obtained integrating Eq. (4) over solid angles and over cell volume V

$$Q_{\text{cell}} = \int_V \int_{4\pi} \frac{I - I^P}{\lambda} d\vec{n} dV. \quad (54)$$

Due to axial symmetry, we can do the integration over cell area A in the R – Z plane

$$Q_{\text{cell}} = \int_A \frac{1}{\lambda} \left(\int_{4\pi} \Gamma d\vec{n} - 4\pi \Gamma^P \right) dr dz. \quad (55)$$

Using $d\vec{n} = \sin \theta d\theta d\varphi$, Eq. (46), and the divergence theorem, we arrive at

$$\begin{aligned} Q_{\text{cell}} &= \int_A \sum_{ij} \frac{2D_{ij}(\Gamma_{ij} - \Gamma^P)}{\lambda} dr dz \\ &= \int_A \sum_{ij} \left(2A_{ij} \frac{\partial \Gamma_{ij}}{\partial r} + 2C_{ij} \frac{\partial \Gamma_{ij}}{\partial z} \right) dr dz \\ &= - \sum_{k \in \text{sides}} \sum_{ij} 2E_{ij} \Gamma_{ij,k} \vec{n}_{ij} \cdot \vec{L}_k, \end{aligned} \quad (56)$$

where $\Gamma_{ij,k}$ is the value of Γ_{ij} at cell interface k . As in previous sections, we start computing depositions at interfaces

$$Q_{ij,k} = -\vec{L}_k \cdot \vec{n}_{ij} (\Gamma_{ij,k} - \Gamma_{ij,k}^* + \lambda_{ij}^E \vec{n}_{ij} \cdot \nabla \Gamma_{ij}^E) 2E_{ij}, \quad (57)$$

where

$$\Gamma_{ij,k}^* = \Gamma_k^P + \frac{\lambda_{ij}^E B_{i,j-1} \bar{\Gamma}_{i,j-1}}{E_{ij} \bar{r}}. \quad (58)$$

Notice that we use $\Gamma_{ij,k}^*$ instead of $\Gamma_{ij,k}^E$, the quantity that would appear in Eq. (57) by straightforward application of Eq. (28). The difference between both quantities vanishes in the limit of optically thick cells

$$\Gamma_{ij,k}^* - \Gamma_{ij,k}^E = \left(\frac{B_{ij} \lambda_{ij}^E}{E_{ij} \bar{r}} \right) \Gamma_k^P \ll \Gamma_k^P, \quad \text{if } \lambda_{ij}^E \ll \bar{r}. \quad (59)$$

This modification is necessary in order to satisfy the conservation condition

$$Q_{\text{cell}} = \sum_{k \in \text{sides}} \sum_{ij} Q_{ij,k}. \quad (60)$$

Finally, the value of $Q_{ij,k}$ is either distributed in equal parts to the two end nodes of side k , or between the three nodes of the cell, for $\Delta \tau_{ij}^E$ larger or smaller than unity, respectively. Energy deposition in a cell can also be written as

$$Q_{\text{cell}} = \frac{A}{\lambda} \left(\sum_{ij} 2D_{ij} \bar{\Gamma}_{ij} - 4\pi \sum_{k \in \text{sides}} w_k \Gamma_k^P \right), \quad (61)$$

where $2D_{ij}$ is the solid angle for direction (i, j) , $\bar{\Gamma}_{ij}$ are given by Eq. (53), and w_k are coefficients, dependent on triangle shape and radiation discretization, that verify $\sum_{k \in \text{sides}} w_k = 1$. This equation expresses a consistent discretization of the integrand of Eq. (55). For optically thin configurations, due to the low order discretization of operator $\sin \varphi \partial I_i / \partial \varphi$ (in Eq. (40)) by $\sin \varphi_{i,j+1} (I_{i,j+1} - I_{ij})$ (in Eq. (43)), the diffusion of intensity perpendicularly to propagation direction, is mixed now with spreading of intensity to directions with different φ angle. A quantitative analysis of accuracy of the algorithm is given in Section 3.1.

In the optically thick regime, one has $\lambda \ll r$, and from Eqs. (49) and (50) one obtains

$$\lambda_{ij}^E \simeq \frac{\lambda E_{ij}}{D_{ij}}, \quad \Gamma_{ij}^E \simeq \Gamma^P. \quad (62)$$

⁴ For $j = 1$, $\Gamma_{i,j-1} = 0$.

Power deposition at an internal interface k is obtained by summing up Eq. (57) over all directions and over the two adjacent cells. For strong gradients with scale length smaller than the distance to the axis, we can neglect the second term in Eq. (58) and also assume $\nabla I^P \simeq 2\pi\bar{r}\nabla I^P$, obtaining

$$Q_k \simeq 2\pi\bar{r}\vec{L}_k \cdot \sum_{ij} \frac{2E_{ij}^2 \vec{u}_{ij} \vec{u}_{ij}}{D_{ij}} \cdot (\lambda_+(\nabla I^P)_+ - \lambda_-(\nabla I^P)_-), \quad (63)$$

with \vec{L}_k directed from cell “–” to cell “+”. The tensor quantity approaches $4\pi/3\vec{u}$ when the number of divisions of the direction space increases ($N, N_i \rightarrow \infty$). As discussed in Section 2.2, one can describe the power coupled to an interface as the jump of normal components of the radiative flux between both sides of the interface

$$Q_k \simeq 2\pi\bar{r}\vec{L}_k \cdot (\vec{S}_- - \vec{S}_+), \quad \text{with } \vec{S} \simeq -\frac{4}{3}\pi\lambda\nabla I^P. \quad (64)$$

For smoother gradients the situation is more complicated, and the algorithm has been validated through comparison with analytical problems as the one discussed in Section 3.2. Although the radiation discretization in R - Z geometry gives place to a set of equations (Eqs. (42)–(43) or (46)), each one corresponding to a propagation direction \vec{n}_{ij} in the R - Z plane, these equations include terms (the ones with r in the denominator) accounting for the fact that photons travelling along straight rays in physical space have curved trajectories in the R - Z plane. These terms appear in the discretization as an attenuation term (photons leaving direction \vec{n}_{ij}) and a source term (photons entering direction \vec{n}_{ij} from neighbor direction $\vec{n}_{i,j-1}$). In this way, a non-physical concentration of rays at the axis of symmetry is avoided. In particular, the radiation temperature is uniform inside a hohlraum with uniform temperature walls, except that, due to truncation errors when Eq. (46) is discretized in space, irregularities of radiation temperature or flux can occur in the 1–2 cells adjacent to the axis of symmetry. This localized anomaly affects only a fraction $\simeq N_r^{-2}$ of the cavity volume, where N_r is the number of cells in radial direction.

2.4. Implementation

The algorithm described above has been implemented in the routines of file **radiar**. The core of the algorithm is routine *radiarun*, that determines, for given values of i and j , the values of I_{ij} at all interfaces, the average value \bar{I}_{ij} at cells (needed to obtain the next value of j), and the power deposition at nodes. In addition, radiation temperature (used only for output diagnostics) is computed by

$$T_R = \left(\frac{1}{2\sigma} \sum_{ij} D_{ij} \bar{I}_{ij} \right)^{1/4}. \quad (65)$$

Before applying the algorithm, routine *radiasort* generates, for each propagation direction \vec{n}_{ij} , the appropriate sequence of cells (see Fig. 2). Triangle nodes must be given in counter-clockwise order, so that the property of a certain side AB , being an entry or an exit of radiation can be determined by the sign of $\vec{u}_\varphi \cdot ((\vec{r}_B - \vec{r}_A) \times \vec{n}_{ij})$. Triangles with all their entries belonging to the boundary are marked to be solved first in the solving sequence list, while triangles adjacent to these are pushed into a “circular buffer”. The buffer is continuously scanned for triangles with known entries, and, when one is found, it is put in the solving sequence, while its unsolved neighbors are pushed into the buffer. The process ends when the buffer is empty. In addition, before calling routine *radiarun*, the radiation intensity at entry boundary interfaces must be supplied by the user’s defined routine *external* as a function of

time, position, and propagating direction. If global option **xsymmetry** is set, the $z = 0$ plane is assumed to be a plane of symmetry. Radiation directions with $\cos\theta < 0$ are solved in the first place. The exit radiation flux at interfaces at $z = 0$ is stored temporally and used as incoming flux for the reflected directions (opposite values of $\cos\theta$ and same values of φ). Reflecting conditions are also assumed at the lower boundary (interfaces with $r = \min r$ at both ends); intensities for an incoming direction ($j_{in} < N_i/2$) are stored and used as input⁵ for the corresponding exit direction ($j_{out} = N_i + 1 - j_{in}$). This procedure avoids energy leakage at the axis of symmetry; notice that the algorithm does not guarantee $\Gamma_{ij} = 0$ at $r = 0$.

2.5. Time advance

With the scheme described so far, the power Q_i deposited by radiation at each node is computed as a function of geometry and node temperatures. The system evolution is determined by equations

$$\frac{dE_i}{dt} = Q_i(T_1, T_2, \dots, T_M), \quad (66)$$

where E_i is the energy assigned to node i . To solve this system, one needs to relate node temperatures and energies by equations-of-state of the form $T_i = T_i(E_i)$. With the discretization used in the code, the system is conservative because $\sum_i Q_i$ has been shown (see Eq. (56)) to be equal to the net radiation power entering the system; for a closed system, $\sum_i Q_i = 0$ and $\sum_i E_i$ remains constant. Eqs. (66) must be integrated in time; energies and temperatures at time level $t^{n+1} = t^n + \Delta t$ are obtained from values at time level t^n . The straightforward explicit method

$$\frac{E_i^{n+1} - E_i^n}{\Delta t} = Q_i(T_1^n, T_2^n, \dots, T_M^n), \quad T_i^{n+1} = T_i(E_i^{n+1}), \quad (67)$$

reduces to a diffusion scheme in optically thick regions, and it is well known that explicit schemes are not appropriate in such a case due to the prohibitively small time step needed to guarantee stability. On the other hand, a fully implicit scheme, where Q_i is evaluated with temperatures at time t^{n+1} , would require the solution of a very big system of coupled nonlinear equations. The linearized scheme

$$\begin{aligned} \frac{E_i^{n+1} - E_i^n}{\Delta t} &= \left(\frac{\partial E_i}{\partial T_i} \right)^n \frac{T_i^{n+1} - T_i^n}{\Delta t} \\ &= Q_i^n + \alpha \sum_{j=1}^M \left(\frac{\partial Q_i}{\partial T_j} \right)^n (T_j^{n+1} - T_j^n), \end{aligned} \quad (68)$$

where upper index n labels quantities evaluated from temperatures at time t^n and α is a parameter between 0 (full explicit) to 1 (full implicit), would require the solution of a system of M linear equations for T_i with full matrix due to radiative coupling between distant elements. Notice that Eqs. (68) satisfy energy conservation: if $\sum_i Q_i$ is zero for any set of temperatures, $\sum_i (\partial Q_i / \partial T_j)^n$ is zero, so that, when equations for all nodes are added, the terms in the right-hand side cancel, and one has $\sum_i E_i^{n+1} = \sum_i E_i^n$. The code MULTI2D uses the so-called “symmetric-semi-implicit” (SSI) method described in [21]. The idea is to neglect all elements outside the main diagonal of $(\partial Q_i / \partial T_j)^n$, so that one can determine T_i^{n+1} in a trivial way. If this is done directly, the energy is not conserved because $\sum_i (\partial Q_i / \partial T_i)^n (T_j^{n+1} - T_j^n)$ is in general different from zero. One can identify the total error as a sum of node errors of the form: $Q_i^n \Delta t + E_i^n - E_i^{n+1}$. This energy error must be corrected for

⁵ Optionally, multiplied by the factor **reflex**.

in subsequent time steps. At each node, a “ghost” energy ϵ_i is defined as the sum of previous and not yet corrected energy errors. At each time step, a fraction β of this value is added to the energy equation. The discretization of Eq. (66) becomes

$$E_i^{n+1} - E_i^n = \left(\frac{\partial E_i}{\partial T_i}\right)^n (T_i^{n+1} - T_i^n) = \left(Q_i^n + \alpha \left(\frac{\partial Q_i}{\partial T_i}\right)^n (T_i^{n+1} - T_i^n)\right) \Delta t + \beta \epsilon_i^n, \quad (69)$$

$$\epsilon_i^{n+1} = \epsilon_i^n + Q_i^n \Delta t - E_i^{n+1} + E_i^n.$$

Obviously the conservation of energy is then satisfied in the sense that $\sum_i E_i + \sum_i \epsilon_i$ is conserved. The method is consistent because, for small values of Δt , one has $|\epsilon_i| \rightarrow 0$. If, in addition, the method is stable, the numerical solution approaches to the true solution. In the context of complex and strongly nonlinear systems, these issues have to be assessed by numerical experiments. Nevertheless, the numerical analysis of simplified cases will allow to clarify the limitations of the method. As such an example, we consider the discretization of the equation of one-dimensional diffusion, $\partial T / \partial t = \partial^2 T / \partial x^2$, taking, for simplicity, a unlimited and uniform grid with $\Delta x = 1$:

$$\frac{dT_i}{dt} = T_{i-1} - 2T_i + T_{i+1}. \quad (70)$$

The solution of such a linear system can be expanded as

$$T_i(t) = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \tilde{T}(k, t) e^{iik} dk. \quad (71)$$

Using this expression in Eq. (70), one has

$$\tilde{T}(k, t) = \tilde{T}(k, 0) e^{\gamma_*(k)t}, \quad (72)$$

where

$$\gamma_*(k) \equiv -4 \sin^2 \frac{k}{2} \quad (73)$$

is the exact dispersion relation of Eq. (70). On other hand, by applying the SSI method to this equation (with $E_i \equiv T_i$ and $Q_i \equiv T_{i-1} - 2T_i + T_{i+1}$), one obtains a numeric dispersion relation

$$\gamma(k) = \frac{1}{\Delta t} \times \log \left(1 + \frac{\gamma_*(k) \Delta t - \beta \pm \sqrt{(\beta + \gamma_*(k) \Delta t)^2 + 8\alpha\beta\gamma_*(k)(\Delta t)^2}}{2 + 4\alpha\Delta t} \right). \quad (74)$$

It can be proven that, for $0 < \beta < 1$ and $\alpha > 1 - \beta/2$, the SSI method is unconditionally stable. Notice that, different from a straightforward explicit method (Eq. (67)), Δt values larger than one can be used (we assume $\Delta t \gg 1$ in the following), and that, due to the introduction of the additional variable ϵ_i , Eq. (74) provides two roots for each wavenumber k . For a given value of k , both roots are real for small enough Δt and can be developed in terms of the quantity $\zeta \equiv \gamma_*(k)(\Delta t)^2$, that we assume to be small. The slower root (positive sign in Eq. (74)) is given by

$$\gamma_1(k) = \gamma_*(k) \left(1 + \left(\frac{2\alpha}{\beta} - \frac{1}{2\Delta t} \right) \zeta + O(\zeta^2) \right), \quad (75)$$

while the faster root (negative sign in Eq. (74)) is

$$\gamma_2(k) \simeq \frac{1}{\Delta t} \ln \left(1 - \frac{\beta}{1 + 2\alpha\Delta t} \right) \simeq -\frac{\beta\gamma_*}{2\alpha\zeta}. \quad (76)$$

After a short transient time $\simeq |1/\gamma_2| \ll |1/\gamma_*|$, in which value of $\tilde{\epsilon}$ goes from 0 (if initially the “ghost” energy was set to be zero) to

$-2\alpha\zeta\tilde{T}/\beta \ll \tilde{T}$ (“ghost” energy is smaller than physical energy), the numerical solution approaches asymptotically the exact one ($\gamma_1 \simeq \gamma_*$). Long wavelength components extending over N intervals and having an attenuation time $t_a \simeq |1/\gamma_*| \simeq k^{-2} \simeq N^2$ are correctly reproduced when $\zeta \simeq k^2(\Delta t)^2 \simeq (\Delta t/N)^2 \ll 1$. This condition means that the number of time steps during the characteristic attenuation time t_a must be larger than N , so that numerical propagation of information (one interval per step) along distance N can take place. Shorter wavelength modes, although poorly reproduced, are strongly damped, as it is the case for the physical ones. MULTI2D uses the SSI method with $\alpha = 1$ and $\beta = 1/2$, a combination that has been found satisfactory for all the situations treated up to now. The accuracy limitations mentioned above are not restrictive in the context of strongly nonlinear problems, where anyway very small time steps are needed to obtain the correct evolution of nonlinear thermal waves; as in the SSI method, more than N time steps are required to solve structures extending over N cells. To apply Eq. (69), in addition to the radiation deposition Q_i (returned by routine *radia*), its temperature derivative ($\partial Q_i / \partial T_i$) and the value of the thermal capacity associated with nodes ($\partial E_i / \partial T_i$) are needed. Different from the implicit method (Eq. (67)) and due to energy conservation inherent to the SSI method, it is not critical that these quantities are extremely accurate. ($\partial Q_i / \partial T_i$) values are evaluated by routine *radiader* in an approximate way. For a node surrounded by optically thin cells, one has from Eq. (6)

$$\frac{\partial Q_i}{\partial T_i} \simeq -\frac{16\sigma T_i^3}{3} \sum_{k \in i} \frac{2\pi \bar{r}_k A_k}{\lambda_k}, \quad (77)$$

where the sum extends over cells that have i as a vertex. For optically thick situations, $\partial Q_i / \partial T_i$ is extracted from Eq. (122), derived in Section 6 for diffusive transport, using $\kappa = 16\lambda\sigma T^3/3\pi$ as thermal conductivity. For intermediate cases, smooth interpolation is used. Also provision is taken for the case where one side AB (of length l) of an optically thick cell belongs to the free boundary (or is in front of an optically thin cell). It will radiate a power given by $\simeq \sigma\pi l(r_A T_A^4 + r_B T_B^4)$, so that a $-4\pi\sigma l r_i T_i^3$ contribution is added to the value of $\partial Q_i / \partial T_i$ at its vertices. Finally, the evaluation of $\partial E_i / \partial T_i$, for mixtures of materials with tabulated thermodynamic properties will be described in detail in Section 6.4.

3. Numerical results

The concepts outlined above have been tested exhaustively in a variety of situations. Here we present two representative cases, which can be compared with analytical solutions. The first one describes an optically thin and the second one an optically thick configuration.

3.1. Optically thin example

The transfer of radiation between a spherical shell and a concentric inner sphere is considered; they are optically thick and have a transparent medium in between. On the inner surface of the outer shell at radius R_2 one specifies, at each point, an incoming isotropic radiative flux that depends only on the azimuthal angle $\Phi \equiv \tan^{-1} z/r$. The inner sphere (with radius R_1) receives on its surface a flux depending also only on Φ . This problem can be solved analytically (see [6], Section 9.4.3) and represents a very sensitive test for the directional radiative transfer algorithm. Both intensities are expressed in terms of spherical harmonics: $I_2(\Phi) = \sum_l a_l Y_{l0}(\Phi)$, $I_1(\Phi) = \sum_l b_l Y_{l0}(\Phi)$, choosing only modes with $m = 0$ due axial symmetry. In addition, we assume mirror symmetry with respect to the plane $z = 0$, so that odd- l terms are absent. It is found that each mode at R_2 maps into the same

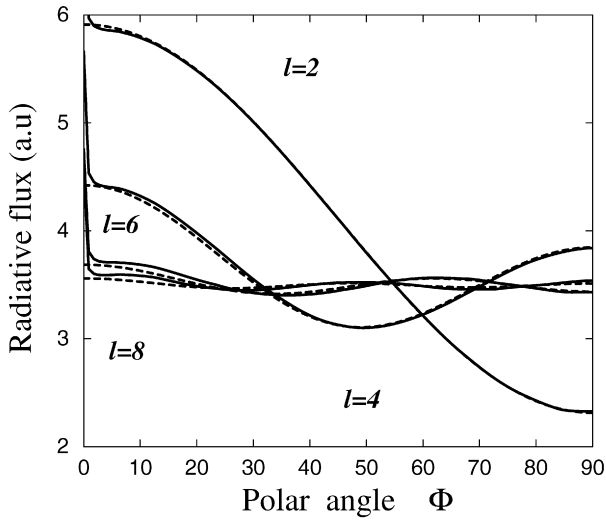


Fig. 7. Two spherical surfaces transfer problem. Radiative power on the internal sphere as a function of Φ , for $R_2/R_1 = 2$. Continuous curves are obtained from simulations, and dashed ones from analytical theory.

mode at R_1 . The ratio b_l/a_l depends only on l and R_2/R_1 , while $b_l/a_l = 0$ for $l \neq l'$. These values have been plotted in Fig. 8 as continuous lines. We simulate this problem numerically, using a regular orthogonal grid composed of quadrangles, with 100 intervals in azimuthal direction (only the range $0 \leq \Phi \leq 90^\circ$ is needed due to mirror symmetry) and $50 \times (R_2 - R_1)/R_1$ intervals in radial direction. Each quadrangle divides into two triangles. The transparent medium is modeled by using a large value of λ ($10^6 \times R_1$). The 2π half-sphere of directions (shown in Fig. 6) is discretized by 180 domains, and a uniform positive offset is added to I_1 to avoid negative intensities. Numerically the problem is quasi-static in the sense that I_2 is obtained after the first integration cycle, before matter energy and temperature are advanced. Given the source intensity $I_2(\Phi) \propto (1 + Y_{l0}(\Phi))$ for modes $l=2, 4, 6$, and 8 at the outer shell and $R_2 = 2R_1$, we plot in Fig. 7 the radiation intensity $I_1(\Phi)$ on the inner sphere (continuous lines). The analytical curves are also plotted in the same figure (dashed lines). There is an overall agreement except at the axis of symmetry where a peak occurs at $\Phi = 0$. When hydrodynamical motion is active, it is sometimes necessary to introduce some numerical filter to remove the singular behavior at this node. In addition one observes that modes with high l are strongly damped. This is quantitatively shown in Fig. 8, where flux is plotted as a function R_2/R_1 for different l -modes. Simulated values agree well with the analytical results.

3.2. Optically thick example

To illustrate the algorithm in an optically thick situation, we consider the problem of the radiative heat wave emerging from an instantaneous point source. The corresponding analytical solution is given in [20] (Ch. X, Sec. 6). A fixed amount of energy is instantaneously released in a point (in the simulation, the initial energy of the node at the origin) and launches a spherical self-similar thermal wave. The spherical front advances with time, while temperature decreases. For simulating this problem, we use a 50×50 rectangular grid, where each quadrangle is divided diagonally into two triangles. As in the previous case, only the half space $z \geq 0$ has to be simulated. A constant and uniform heat capacity is assumed, and the radiative mean free path is set to one tenth of the grid size, such that cells are optically thick. We use dimensionless variables, so that temperature and time are expressed in arbitrary

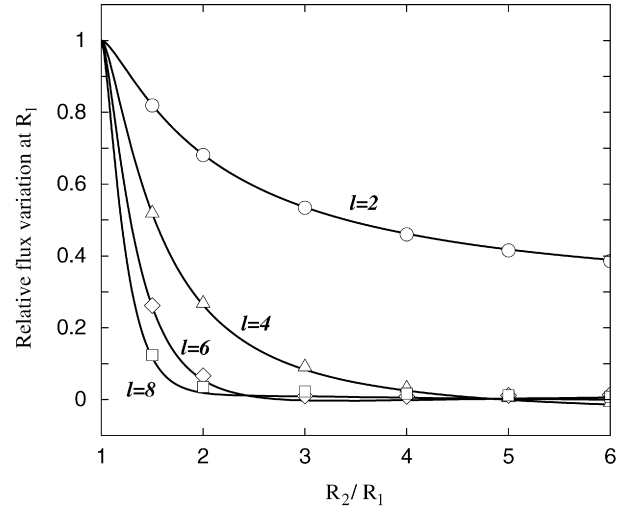


Fig. 8. Two spherical surfaces transfer problem. Reduction factor between emitted flux at R_2 and incident flux at R_1 , for different spherical modes, as a function of R_2/R_1 . Lines are analytical values, dots are from simulations.

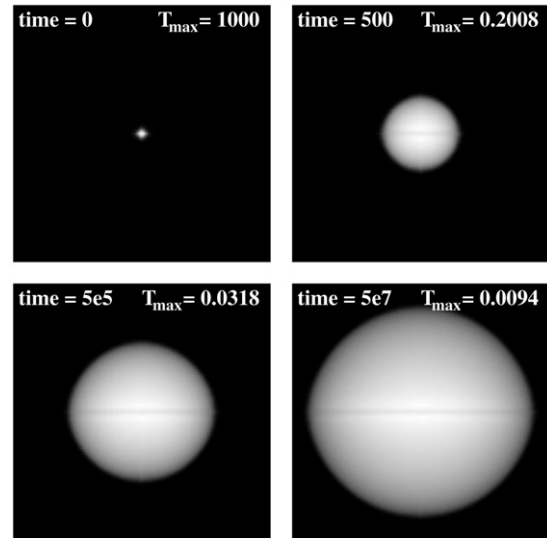


Fig. 9. Instantaneous point source problem. Radiation temperature plots at different times (both in arbitrary units).

units. Fig. 9 shows the radiation temperature distribution (almost identical to the matter temperature) at different times. In Fig. 10, radial temperature profiles obtained by the code (cuts at $z = 0$) are compared with the analytical solution. There is a good agreement: the position of the thermal front is correctly determined, and the profiles are similar. Nevertheless, small differences are visible in the region close to the axis of symmetry, where the used discretization does not describe diffusive transport accurately, because cell size is not small compared with distance to the axis. Also the shape of the numerical thermal wave front deviates slightly (about 8%) from a sphere.

4. Lagrangian hydrodynamics

4.1. General algorithm

We consider a Lagrangian description of a fluid. By definition, the positions of some fluid particles (*nodes*) are followed in time, each node i is described by its position vector $\vec{r}_i(t)$, and

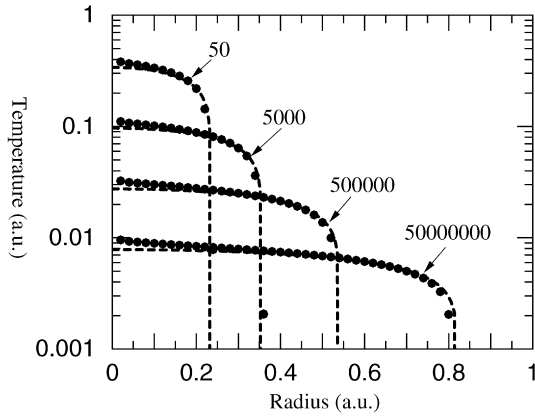


Fig. 10. Instantaneous point source problem. Temperature profiles as a function of r at $z = 0$ at different times. Dots are from simulation, dashed lines correspond to analytical solution.

$\vec{v}_i \equiv \dot{\vec{r}}_i \equiv d\vec{r}_i/dt$ is the fluid velocity at node position. For fluid particles inside tetrahedral domains (*cells*), velocity is assumed to vary linearly according to

$$\vec{v}(\vec{r}) = \sum_i \vec{v}_i w_i(\vec{r}), \quad (78)$$

where the functions $w_i(\vec{r})$ take the value 1 at node i , 0 in all other nodes, and vary linearly inside cells. The integrals over cell k of volume V_k are

$$\int_{V_k} w_i(\vec{r}) dV = \frac{1}{4} V_k, \quad \int_{V_k} w_i(\vec{r}) w_j(\vec{r}) dV = \frac{1 + \delta_{ij}}{20} V_k, \quad (79)$$

for vertices i and j of cell k , and zero otherwise. With this description, tetrahedra transform into tetrahedra as the fluid moves, mass is conserved because particles remain in the same cell, and, in absence of forces (pressure, gravity, ...), the solution is exact. On the other hand, $\nabla \vec{v}$ and $\nabla \cdot \vec{v}$ are uniform inside a cell, so that, if the density is assumed to be initially uniform inside a cell, it will remain uniform because⁶ of $D\rho/Dt = -\rho \nabla \cdot \vec{v}$. Cell mass, center of mass, and linear momentum of cell k take the form

$$m_k = \rho_k V_k, \quad (80)$$

$$\vec{r}_{Gk} = \frac{1}{m_k} \int_{V_k} \rho \vec{r} dV = \frac{1}{4} \sum_{i \in k} \vec{r}_i, \quad (81)$$

$$\vec{p}_k = \int_{V_k} \rho \vec{v} dV = \frac{m_k}{4} \sum_{i \in k} \vec{v}_i = m_k \dot{\vec{r}}_{Gk}, \quad (82)$$

where “ $i \in k$ ” indicates summation over the four vertices of the cell. The stress tensor $\vec{\sigma}$ in a fluid is commonly modeled as a function of thermodynamic variables and velocity derivatives. As density and velocity gradient have been assumed uniform inside cells, it is natural to assume that also $\vec{\sigma}$ is uniform there. With this assumption, the momentum equation

$$\rho \frac{D\vec{v}}{Dt} \equiv \rho \left(\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} \right) = \nabla \cdot \vec{\sigma} \quad (83)$$

is not satisfied everywhere, because its left-hand side is a piecewise discontinuous linear function and its right-hand side is zero inside the cells and singular at cell faces. Weighting the residue of Eq. (83) with functions $w_i(\vec{r})$, we obtain

$$\sum_{k \in i} m_k \sum_{j \in k} \frac{1 + \delta_{ij}}{20} \dot{\vec{v}}_j = \frac{1}{3} \sum_{l \in i} (\vec{\sigma}_{l+} - \vec{\sigma}_{l-}) \cdot \vec{A}_l \equiv \vec{F}_i, \quad (84)$$

where “ $k \in i$ ” indicates summation over all cells having node i as a vertex, j extends to all nodes of cell k , and subindex l extends to all faces having node i as a vertex. \vec{A}_l is the area vector, normal to the face, while $\vec{\sigma}_{l+}$ and $\vec{\sigma}_{l-}$ are the stress tensors in the two cells adjacent to the face (\vec{A}_l pointing from $l-$ to $l+$). At the boundary faces, an external stress has to be specified (for example, zero in the case of a free boundary). The right-hand side term of Eq. (84) has a simple intuitive mechanical interpretation: if an interface were a solid plate leaning on its three vertices, the total force exerted by the fluid on it would be $\sum_l (\vec{\sigma}_{l+} - \vec{\sigma}_{l-}) \cdot \vec{A}_l$, and one third of this value would be transmitted to each of its vertices. Adding the contribution of all faces that contain node i , one obtains the value \vec{F}_i . One can, in each face, separate the contributions of the two adjacent cells, and add separately the contribution of the stress of each cell k in its own nodes \vec{F}_{ik} . One has

$$\vec{F}_i = \sum_{k \in i} \vec{F}_{ik} + \vec{F}_i^{\text{ext}}, \quad (85)$$

where external forces \vec{F}_i^{ext} are due to boundary stresses. Notice that

$$\sum_{i \in k} \vec{F}_{ik} = \sum_{i \in k} \vec{r}_i \times \vec{F}_{ik} = 0; \quad (86)$$

the system of forces due to stress in cell k has null resultant and momentum. Eq. (84) is not appropriate for multi-dimensional numerical integration; if the forces are given, a linear system of equations involving all nodes has to be solved to obtain $\dot{\vec{v}}_i$. To avoid this inconvenience, we define an alternative set of functions $z_i(\vec{r})$ in the following way: each tetrahedron is subdivided in 4 octahedra, each one with a vertex at one of the tetrahedron vertices, the opposite one in the tetrahedron center, three vertices in the centers of edges, and three vertices in the centers of faces. $z_i = 1$, if point \vec{r} is inside one of the octahedra that have node i as a vertex, and 0, otherwise. These functions satisfy:

$$\int_{V_k} z_i(\vec{r}) dV = \frac{1}{4} V_k, \quad \int_{V_k} z_i(\vec{r}) w_j(\vec{r}) dV = \frac{23 + 52\delta_{ij}}{576} V_k. \quad (87)$$

Now, weighting Eq. (83) with $z_i(\vec{r})$ instead of $w_i(\vec{r})$, one obtains

$$\sum_{k \in i} m_k \sum_{j \in k} \frac{23 + 52\delta_{ij}}{576} \dot{\vec{v}}_j = \vec{F}_i. \quad (88)$$

Eqs. (84) and (88) have the same level of accuracy. They can be combined linearly (with weights $-\frac{115}{29}$ and $\frac{144}{29}$) to get:

$$\left(\sum_{k \in i} \frac{m_k}{4} \right) \dot{\vec{v}}_i \equiv m_i \dot{\vec{v}}_i = \vec{F}_i. \quad (89)$$

This equation has an obvious interpretation: the positions of nodes evolve as particles of mass m_i (the mass of each cell distributed in equal parts between its nodes) submitted to forces \vec{F}_i . The following conservation principles are satisfied:

- (i) **mass:** cell masses and node masses are constant

$$\frac{d}{dt} \sum_k m_k = \frac{d}{dt} \sum_i m_i = 0. \quad (90)$$

- (ii) **linear momentum:** adding Eq. (89) for nodes and taking into account Eq. (86) one has:

$$\frac{d}{dt} \sum_i m_i \vec{v}_i = \frac{d}{dt} \sum_k \vec{p}_k = \sum_i \vec{F}_i^{\text{ext}}. \quad (91)$$

- (iii) **center of mass velocity:**

$$\frac{d\vec{v}_G}{dt} = \sum_i m_i \dot{\vec{v}}_i / \sum_i m_i = \sum_i \vec{F}_i^{\text{ext}} / \sum_i m_i. \quad (92)$$

⁶ $D/Dt \equiv \partial/\partial t + \vec{v} \cdot \nabla$ is the convective time derivative.

(iv) **angular momentum:** adding the cross products of \vec{r}_i by Eq. (89), and taking into account Eq. (86) one has:

$$\frac{d}{dt}\vec{L} = \frac{d}{dt} \sum_i \vec{r}_i \times m_i \vec{v}_i = \sum_i \vec{r}_i \times \vec{F}_i^{\text{ext}}. \quad (93)$$

For an isolated system ($\vec{F}_i^{\text{ext}} = 0$), linear and angular momentum are conserved, and the center of mass of the system moves uniformly with time: $\vec{r}_G(t) = \vec{r}_G(0) + \vec{v}_G(0)t$. Notice that Eq. (93) requires to define angular momentum as the angular momentum of node particles of mass m_i instead of $\int \rho \vec{r} \times \vec{v} dV$ evaluated with the assumed form of ρ and \vec{v} . Finally, one can define kinetic energy as $T = \frac{1}{2} \sum_i m_i v_i^2$. As for angular momentum, this quantity is not identical to $\frac{1}{2} \int \rho v^2 dV$. The time derivative \dot{T} is equal to the power of forces acting on nodes, $\sum_i \vec{v}_i \cdot \vec{F}_i$. Because the force contributions \vec{F}_{ik} of cell k in its nodes are due to its own stress, the power exerted by this system should be extracted from the internal energy of the cell:

$$\dot{E}_k = - \sum_{i \in k} \vec{v}_i \cdot \vec{F}_{ik}, \quad (94)$$

so that

$$\frac{d}{dt} \left(\sum_k E_k + \sum_i \frac{m_i v_i^2}{2} \right) = \sum_i \vec{v}_i \cdot \vec{F}_i^{\text{ext}}, \quad (95)$$

also energy is conserved in an isolated system. The algorithm described above can be extended trivially to planar 2D geometry; tetrahedra become triangles, triangular interfaces become edges, and the coefficient $\frac{1}{3}$ in Eq. (84) and $\frac{1}{4}$ in Eq. (89) must be changed to $\frac{1}{2}$ and $\frac{1}{3}$, respectively.

4.2. Artificial viscosity

It is of common use in fluid dynamics to assume that the stress tensor is the sum of the thermodynamic equilibrium pressure P , depending on two thermodynamic variables (for example, ρ and e), plus a viscous contribution $\vec{\sigma}_v$, related to the gradient of velocity,

$$\vec{\sigma} \simeq -P\vec{u} + \vec{\sigma}_v. \quad (96)$$

The present code uses the so-called *artificial viscosity*; an appropriate value of viscous stress is added to the pressure to guarantee numerical stability, allowing shock waves to be treated as quasi-discontinuities with a thickness of few numerical cells [22]. For example, consider the motion of a cold gas ($P \simeq 0$). Fluid particles move with constant velocity, and the solution becomes multi-valued if one of the vertices of a cell crosses the opposite face. This will not occur in a real fluid; before becoming multi-valued, the velocity gradient ($\nabla \vec{v}$) becomes very large, and the viscous stress keeps the fluid elements apart. This effect can be modeled numerically by introducing an appropriate additional pressure Q , just to avoid vertex crossing. Consider an isolated, cold, and very flattened cell of base A and height h with $\dot{h} < 0$ (for simplicity assume a symmetrical base). The mass assigned to each node is $\frac{1}{4}m$, the vertical forces on the vertices are $\frac{1}{3}QA$ (top) and $\frac{1}{9}QA$ (base), so that $\dot{h} = -\frac{16}{9}AQ/m$. Choosing $Q = \frac{9}{32}m\dot{h}^2/Ah \equiv \frac{3}{32}\rho\dot{h}^2$, we obtain

$$h(t) = h_0 - \dot{h}_0 t + \frac{\dot{h}_0^2 t^2}{4h_0}. \quad (97)$$

The cell collapse is marginally stopped at time $t = 2\dot{h}_0/h_0$. In a more complex context (mass at the vertices include contributions from neighbor cell, thermal pressure peaks at cell collapse, ...), one can write:

$$Q = \begin{cases} a^2 \rho \dot{h}^2 & \text{for } \dot{h} < 0, \\ 0 & \text{for } \dot{h} \geq 0, \end{cases} \quad (98)$$

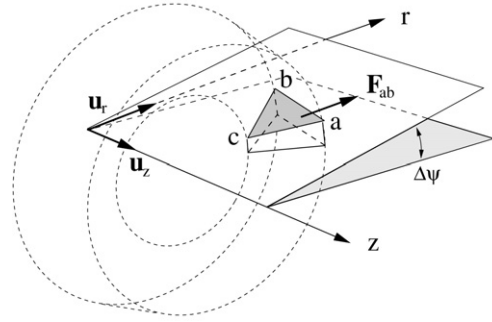


Fig. 11. Axial-symmetric element.

where a is a numerical factor, typically in the range of 1–2. To evaluate this expression in a multi-dimensional context, we use $h_k = \dot{V}_k h_k / V_k$, h_k being the minimum height of the cell, defined in Section 4.5.

4.3. Low-compressibility flows

The use of tetrahedral cells (triangles in 2D) becomes problematic, when either liquids or low-Mach-number incompressible flows are considered. In such situations, volumes of Lagrangian cells are approximately conserved. This fact can be viewed as a set of mechanical constraints that reduce the effective number of degrees of freedom of the system, jeopardizing drastically the accuracy of the calculation. For example, in a planar grid composed of triangles the number of degrees of freedom for an incompressible fluid is approximately equal to the number of boundary nodes. To overcome this difficulty, neighbor cells can be grouped together,⁷ using the composite volume to compute density and pressure, and reducing in this way the number of constraints. For example, in a planar grid formed by quadrangles, each composed by two triangles, the number of degrees of freedom is approximately equal to the number of quadrangles. Because adjoint cells are managed together, it is natural also to assign their total mass in equal parts to their vertices. When the grid becomes too distorted (the criterion implemented in 2D is that area ratios between quadrangle halves is larger than 2) the cell components are considered separately with individual pressures computed from their own volume, allowing the grid to react properly to avoid node crossing and multi-valued flows. Finally, it must be pointed out that all conservation principles discussed in the previous section are still satisfied after these changes.

4.4. Axial-symmetric hydrodynamics

Extension of the method described so far in this section to axial-symmetric configurations requires slight modifications. In cylindrical coordinates, triangular elements in the R – Z plane are toroidal figures in space. Conceptually, we consider sections of the toroidal elements between planes $\psi = \pm \Delta\psi/2$ ($\Delta\psi \ll 1$) as indicated in Fig. 11. The mass of these elements (either with triangular or quadrangular cross section) is distributed in equal parts to their vertices. On a given interface ab (a lateral face of the cell) the total force due to pressure is

$$\vec{F}_{ab} = \frac{r_a + r_b}{2} \Delta P \Delta\psi \left((z_a - z_b) \vec{u}_r + (r_b - r_a) \vec{u}_z \right), \quad (99)$$

where ΔP is the jump of pressures (in Fig. 11, it is the pressure inside the cell abc minus the external pressure). Notice that the

⁷ Only for hydrodynamics; for radiative transfer and heat conduction, cells are treated individually.

system of forces $\{\vec{F}_{ab,k}, \vec{F}_{bc,k}, \vec{F}_{ca,k}\}$, acting on interfaces of cell k due to its own pressure, is not a null system as in the planar case. Nevertheless, the resultant along \vec{u}_z is zero, guaranteeing conservation of linear momentum and center of mass position along Z direction. We consider the force \vec{F}_{ab} applied at the mean point $\vec{r}_{ab} = \frac{1}{2}(\vec{r}_a + \vec{r}_b)$, and we distribute it to nodes a and b , preserving resultant and momentum

$$\begin{aligned}\vec{F}_a + \vec{F}_b &= \vec{F}_{ab}, \\ \vec{r}_a \times \vec{F}_a + \vec{r}_b \times \vec{F}_b &= \vec{r}_{ab} \times \vec{F}_{ab}.\end{aligned}\quad (100)$$

Because these conditions are equivalent to three scalar equations, there is one degree of freedom to determine the four components of forces at nodes. In addition to the trivial choice

$$\vec{F}_a = \vec{F}_b = \frac{\vec{F}_{ab}}{2}, \quad (101)$$

an alternate scheme can be considered; first, \vec{F}_{ab} is divided into two parts: a radial one $(\vec{F}_{ab})_r$ along vector $\vec{r}_a + \vec{r}_b$, and $(\vec{F}_{ab})_\theta$ perpendicular to $\vec{r}_a + \vec{r}_b$. The first part is decomposed into two vectors of the same magnitude directed along \vec{r}_a and \vec{r}_b , and applied to nodes a and b ; the second part is divided into two equal contributions. We have

$$\begin{aligned}\vec{F}_a &= \frac{1}{2} \left(\vec{F}_{ab} + \frac{\vec{F}_{ab} \cdot (\vec{r}_a + \vec{r}_b)}{|\vec{r}_a + \vec{r}_b|^2} (\vec{r}_a - \vec{r}_b) \right), \\ \vec{F}_b &= \frac{1}{2} \left(\vec{F}_{ab} - \frac{\vec{F}_{ab} \cdot (\vec{r}_a + \vec{r}_b)}{|\vec{r}_a + \vec{r}_b|^2} (\vec{r}_a - \vec{r}_b) \right).\end{aligned}\quad (102)$$

This decomposition guarantees a spherically symmetric flow, as long as initial and boundary conditions are spherically symmetric and the grid is a regular one, composed of quadrangles between the nodes located at

$$\vec{r}_{ij} = r_i (\cos(j\Delta\theta)\vec{u}_z + \sin(j\Delta\theta)\vec{u}_r). \quad (103)$$

The input parameter **h.spheric** specifies the use of either Eq. (101) or Eq. (102) for interfaces with mean radius above or below a given value.

4.5. Time discretization

We assume that $\vec{r}_i(t)$ are continuous, piecewise linear functions of time; as a consequence, the first derivatives $\vec{v}_i(t)$ are discontinuous and piecewise constant, while the second derivatives are singular at these discontinuities. This means that forces are implicitly assumed to act impulsively. We assume that a time step begins at time t^n and ends at time $t^{n+1} = t^n + \Delta t$ and that velocities change instantaneously at time $t^{n+1/2} = t^n + \frac{1}{2}\Delta t$. The maximum allowable time step is limited by the Courant–Friedrich condition: $\Delta t \lesssim \min(h_k/c_k)$, where h_k is the characteristic grid size, and c_k the isentropic sound speed of cells. We define h_k^2 as the smaller eigenvalue of the tensor⁸

$$\vec{G}_k = \sum_{i \in k} \vec{r}_i^n \vec{r}_i^n - \frac{1}{N_k} \left(\sum_{i \in k} \vec{r}_i^n \right) \left(\sum_{i \in k} \vec{r}_i^n \right) \quad (104)$$

with $N_k = 3$ for triangles and $N_k = 4$ for quadrangles,

$$h_k^2 = \frac{G_{k,rr} + G_{k,zz}}{2} - \sqrt{\left(\frac{G_{k,rr} - G_{k,zz}}{2} \right)^2 + G_{k,rz}^2}. \quad (105)$$

⁸ The central planar tensor of inertia for a set of unit mass particles located at the vertices of the cell.

For rectangular cells, h_k results to be the smallest height. The isentropic sound speed is computed at time t^n from densities ρ_k^n , pressures P_k^n , and thermodynamic derivatives as

$$c_k = \sqrt{\left(\frac{\partial P}{\partial \rho} \right)_{e,k}^n + \frac{P_k^n}{(\rho_k^n)^2} \left(\frac{\partial P}{\partial e} \right)_{\rho,k}^n}. \quad (106)$$

Between t^n and $t^{n+1/2}$, nodes move ballistically with constant velocity:

$$\vec{r}_i^{n+1/2} = \vec{r}_i^n + \frac{1}{2} \vec{v}_i^n \Delta t. \quad (107)$$

Obviously, linear and angular momentum are preserved in this phase. At times t^n and $t^{n+1/2}$, cell volumes V_k are computed, and a compression factor is defined as

$$\beta_k = \frac{V_k^{n+1/2} - V_k^n}{V_k^n}. \quad (108)$$

If $\beta_k < 0$, the artificial pressure given by Eq. (98) is

$$Q_k = a^2 \rho_k^n \left(\frac{2\beta_k h_k}{\Delta t} \right)^2. \quad (109)$$

Cell pressures at *impulsion* time $t^{n+1/2}$ are extrapolated from their values at time t^n by adapting the 1D Lagrangian method described in [22]:

$$P_k^{n+1/2} = P_k^n - \beta_k \rho_k^n \left(\frac{\partial P}{\partial \rho} \right)_{e,k}^n + \left(\frac{\partial P}{\partial e} \right)_{\rho,k}^n \Delta e_k, \quad (110)$$

where the increment of specific internal energy Δe_k between t^n and $t^{n+1/2}$ is estimated from the energy equation to be

$$\Delta e_k = -\frac{\beta_k}{\rho_k^n} \left(Q_k + \frac{P_k^{n+1/2} + P_k^n}{2} \right). \quad (111)$$

Once the pressure in each cell $P_k^{n+1/2} + Q_k$ has been determined by Eqs. (109) to (111), the forces at nodes of cell k due to its own pressure $\vec{F}_{ik}^{n+1/2}$ are computed from Eqs. (99) to (102). The total forces at nodes $\vec{F}_i^{n+1/2}$ are obtained adding the contribution of each cell and, if requested, the external forces at nodes \vec{F}_i^e supplied by the user's routine *pext.f*. The new velocities are determined assuming that forces act impulsively and are proportional to Δt

$$\vec{v}_i^{n+1} = \vec{v}_i^n + \frac{\vec{F}_i^{n+1/2}}{m_i} \Delta t. \quad (112)$$

These velocities can be optionally modified by the user's routine *bound.f* to enforce boundary conditions like sliding boundaries or pistons. The increment of internal energy in the cell is computed according to Eq. (94) to be

$$\Delta E_k = -\sum_{i \in k} \vec{F}_{ik}^{n+1/2} \cdot (\vec{v}_i^{n+1} - \vec{v}_i^n) \Delta t. \quad (113)$$

This quantity is either distributed in equal parts to nodes (if $\Delta E_k > 0$) or subtracted from the nodes in proportion to their temperature (if $\Delta E_k < 0$). Finally, the position of nodes at time t^{n+1} is advanced according to

$$\vec{r}_i^{n+1} = \vec{r}_i^{n+1/2} + \frac{1}{2} \vec{v}_i^{n+1} \Delta t. \quad (114)$$

5. Heat transport and beam deposition

Diffusive heat transport, in absence of energy deposition and hydrodynamic motion, is determined by

$$\rho c \frac{\partial T}{\partial t} = -\nabla \cdot \vec{q} \equiv -\nabla \cdot (-\kappa \nabla T), \quad (115)$$

where \vec{q} is the thermal flux, κ the thermal conductivity, and $c \equiv \partial e_i / \partial T$ the specific heat capacity. A simple treatment of Eq. (115) on unstructured grids is to assume a continuous piecewise linear temperature distribution with values defined at nodes:

$$T(\vec{r}, t) = \sum_i T_i(t) w_i(\vec{r}), \quad (116)$$

where $w_i(\vec{r})$ are the functions defined in Section 4.1. Assuming κ and c to be evaluated at cells (i.e. from average values of ρ and T), \vec{q} is constant inside cells; its divergence is zero inside cells and singular at cell interfaces. Proceeding as in Section 4.1, Eq. (116) can be weighted with functions $w_i(\vec{r})$ or $z_i(\vec{r})$ to produce either

$$\sum_{k \in i} \rho_k c_k V_k \sum_{j \in k} \frac{1 + \delta_{ij}}{20} \dot{T}_j = \frac{1}{3} \sum_{l \in i} (\vec{q}_{l-} - \vec{q}_{l+}) \cdot \vec{A}_l \equiv Q_i, \quad (117)$$

or

$$\sum_{k \in i} \rho_k c_k V_k \sum_{j \in k} \frac{23 + 52\delta_{ij}}{576} \dot{T}_j = Q_i, \quad (118)$$

respectively. Both equations can be linearly combined to obtain

$$\left(\sum_{k \in i} \frac{\rho_k c_k V_k}{4} \right) \dot{T}_i = Q_i. \quad (119)$$

The left-hand side of this equation can be interpreted as the time derivative of the internal energy \dot{E}_i , the term in parentheses being a thermal capacity associated with node i . The right-hand side can be rearranged as

$$Q_i = \sum_{k \in i} \frac{1}{3} \vec{A}_{ki} \cdot (\kappa \nabla T)_k, \quad (120)$$

where subindex k extends to cells that have node i as a vertex, and \vec{A}_{ki} is a vector, normal to the face of cell k opposite to vertex i with modulus equal to face area and pointing outwards. The temperature gradient in cell k can be determined as a function of temperatures at vertices by

$$(\nabla T)_k = - \frac{\sum_{j \in k} \vec{A}_{kj} T_j}{V_k}. \quad (121)$$

Using this expression in Eq. (120), one obtains finally

$$\dot{E}_i = \sum_j M_{ij} T_j, \quad (122)$$

where the coefficients M_{ij} include contributions of all cells having simultaneous nodes i and j as vertices

$$M_{ij} = \sum_{k \in i, k \in j} - \frac{\kappa_k \vec{A}_{ki} \cdot \vec{A}_{kj}}{9V_k}. \quad (123)$$

The matrix of coefficients is symmetric ($M_{ij} = M_{ji}$), negative defined ($\sum_i \sum_j \alpha_i M_{ij} \alpha_j \leq 0$ for any set of real numbers $\{\alpha_i\}$), and satisfies $\sum_i M_{ij} = 0$. This last property guarantees that $\dot{E}_i = 0$, when temperatures of neighboring nodes are uniform, and that energy is conserved: $\sum_i \dot{E}_i = 0$. These results can be extended in a straightforward way to pure planar X - Y geometry; for a layer of thickness Δh composed of prisms with triangular base, the thermal capacity associated to a node (here, a vertical edge) is given by

$$\frac{\partial E_i}{\partial T_i} = \sum_{k \in i} \frac{\rho_k c_k V_k}{3}. \quad (124)$$

On the other hand, Q_i is determined by Eqs. (122)–(123) with

$$\vec{A}_{ki} = ((y_b - y_a)\vec{u}_x - (x_b - x_a)\vec{u}_y) \Delta h, \quad (125)$$

where a and b are the two vertices of the triangular base of prism k , distinct to i , given in counter-clockwise order. For axial-symmetric configurations, one can consider a thin slice contained between the planes $\psi = \pm \Delta \psi / 2$, as shown in Fig. 11. Now $\Delta h \simeq r \Delta \psi$ is not longer uniform over one cell. In this situation, we modify Eq. (125) using the value Δh at the center of the cell

$$\vec{A}_{ki} = ((r_b - r_a)\vec{u}_r - (z_b - z_a)\vec{u}_z) \frac{r_i + r_a + r_b}{3} \Delta \psi. \quad (126)$$

In this way, the desirable properties of M_{ij} are maintained, and consistency is satisfied ($\rho \partial e_i / \partial t \rightarrow \nabla \cdot \kappa \nabla T$, for evanescent grid size). Eq. (122) has the structure of Eq. (66), discussed in Section 2.4, and can be solved with the SSI method. In the current implementation, energy transfer by radiation and by heat conduction are solved together in a single application of Eq. (69). Sometimes heat flux under extreme gradients is modeled, imposing a maximum allowable value [23,24]. Typically, in ICF simulations flux limitation is applied according to

$$|\vec{q}| \leq q^{\max} \equiv f n_e k T \sqrt{\frac{kT}{m_e}}, \quad (127)$$

where k is the Boltzmann's constant, m_e and n_e , electron mass and number density, and f , a numerical coefficient to be adjusted empirically for each specific problem (see [6], Ch. 7.2). This requirement is satisfied using a nonlinear conductivity instead of κ

$$\kappa' = \frac{\kappa q^{\max}}{q^{\max} + \kappa |\nabla T|}. \quad (128)$$

For small temperature gradients, Fick's law is recovered. When this option is active, κ' is evaluated using the value of ∇T at the previous time step.

The 2D version of the code uses the ray-tracing approach to compute energy deposition into matter by laser light or ion beams. Beams are considered to be composed of a certain number of rays (beam-lets) that are assumed to follow straight trajectories. This is almost correct in the case of ion beams, but it is a rather crude approximation for laser light propagation. Each beam-let is followed in 3D space and deposits its energy into the toroidal elements drawn in Fig 11. From the point of view of the 2D grid, this is equivalent to an azimuthal average over ψ . User supplied routines must specify in each time step, the starting point, direction, and power of each beam-let as well as the appropriate attenuation coefficient.

6. Implementation

6.1. Programming style

The algorithms described in the previous sections have been coded as a set of C language routines. Data is passed and returned by double indirect reference; that is by pointers to structures (of type "D" defined in file **D.h**) that describe the data by specifying:

- **type**: integer, float, double, character ...
- **size**: number of data items.
- **address**: memory pointer to the beginning of the data array.
- **counter**: number of times this structure is currently referenced.

This information can be used to check data consistency and, if necessary, perform data conversion. In addition to numeric arrays, other types of data have been defined: **null** (no data at all), **function** (in this case, **address** is a pointer to a C-function, and **size** is the number of arguments), and **list** (the data is an array of pointers to "D-structures"). The "D-list" type allows to implement multi-level lists of arbitrary depth. Because data is managed by reference,

the same data can be concurrently used in several places. A simple strategy allows to control memory allocation: (i) each “D-structure” is created with **counter=1**, (ii) each time it is passed as a function argument or included in a “D-list”, the value of **counter** is incremented, (iii) before routine return, the value of **counter** of any locally created structure or function argument is decremented, and, if **counter = 0**, the structure is destroyed.

In addition to the radiation-hydrodynamics algorithms, a number of miscellaneous utility routines has been developed to perform creation, manipulation, and deletion of “D-structures”, basic arithmetic between numeric data, file access, operating system interface, grid generation, ... Top level components of MULTI2D and other related codes are basically composed of a large amount of calls to the appropriate routines. To facilitate their development, a special computer language (called **r94**), human readable and with a simple syntax, has been created. Files with **r94**-code are preprocessed to generate C code.

All this software has been organized in a modular way: all related source files are stored together in the same “tar” archive,⁹ identified by a unique name, typically a keyword followed by the version number. For example, **m2d-5.1** corresponds to the algorithms described in this paper. Each of these “modules” is documented and maintained individually, and it must be installed in its own subdirectory by converting source files to C, compiling the C code, and creating an object-library. Modules can also be used as containers of other types of data, like tables or documents.

The software package supplied through the **CPC** Program Library is composed of:

- (i) **r94_2005.tar.gz** - installation and control scripts, modules implementing general purpose libraries, the **r94** to C translator, a development toolkit (including a graphical user interface), and miscellaneous documentation. This subpackage has to be installed first.
- (ii) **multi_2008.tar.gz** - modules corresponding to the radiation-hydrodynamics (**m2d-5.1**), file access (**multi_data-2.0**), table interpolation (**matter-2.1**), grid generation (**malla-5.0**), a graphical user interface to control job submission, miscellaneous graphic post-processors, equation-of-state tables, opacity tables, and the example described in Section 7.
- (iii) **README** - note with detailed instructions for installation.

6.2. Centering of variables

When the algorithms described in Sections 2 and 5, using node-centered temperatures and energies, are coupled to the hydrodynamic methods described in Section 4, requiring cell-centered energies and densities, some conversion procedure is needed [16]. In MULTI2D the primary variables: internal energy (E), “ghost” energy (ϵ), node position (\vec{r}), and node velocity (\vec{v}) are node-centered, and only material masses (m) are associated to cells. From these quantities, several derived quantities (density, temperatures, etc) are obtained either at cells, or at nodes, or in both, in the course of an integration cycle, and are discarded after the time step is finished. Details are given in Section 6.4.

6.3. Specific problem code

To maximize the versatility, each separate physical problem (a “case”) is operated in a separate directory that contains specific **r94** source code, control scripts, and program output. A graphic user interface, shown in Fig. 12, allows to clone, delete, archive, install, edit, and submit “cases” to be run in the background. In addition, several graphic post-processors can be launched to visualize

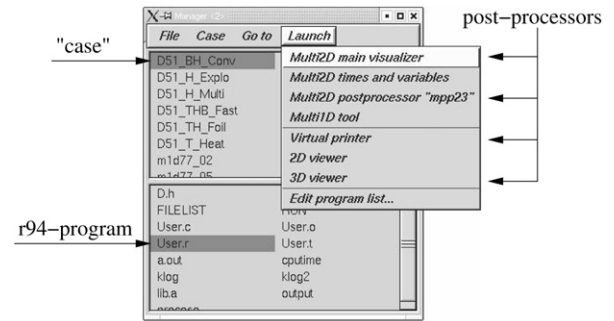


Fig. 12. Graphic user interface.

the results. For each “case”, a user defined code (file **User.r**) is used to initialize the problem (with the help of several special libraries) and pass control to the main integration routine. The shell procedure *RUN* translates this **r94** code to C, compiles the C code, links the object code with the libraries specified in file *DEPENDENCES* to generate an executable code, and launches it as a background job. **User.r** contains in fact a main program responsible to carry out the following tasks:

- (i) Define the geometry of the grid: initial node positions (arrays $\mathbf{x0}$ and $\mathbf{y0}$), and, for each triangular cell, the indices of its nodes (array \mathbf{tb}) and a flag (array \mathbf{ct}) indicating if this triangle is either an independent cell (value $\mathbf{0}$), or the first/second half of a quadrangular cell (values $\mathbf{1/2}$). This task is usually accomplished with the help of the grid generation package **malla-5.0**. The utility routine *NewTopo* uses this basic data to generate additional tables needed in different places of the code: edge numbering; edge–triangle, edge–node, edge–triangle, triangle–edge, and triangle–triangle connectivities; as well as a list of edges belonging to the boundary.
- (ii) Set the dynamical state variables: initial velocity \vec{v}_i , internal energy E_i , and “ghost” energy ϵ_i at each node i , and masses m_{lk} of material number l in cell k . This data (together with topological information of (i)) is grouped in a structure called **hydro**, created by routine *HydroNew*. This task is performed usually by the routine *AddToHydro* that allows to add a material with specified density and temperature to each cell.
- (iii) The structure **hydro** and all program parameters are stored in a structure called **multi** created by routine *MultiNew*. This structure stores also user’s functions to define boundary conditions (external pressure, kinematic restrictions, incoming radiation, incoming beams, ...), matter properties (opacities, thermal conductivities, beam attenuation, ...), timing control, and output control.
- (iv) Finally, the main integration routine *MultiLoop* is entered.

6.4. Integration loop code

Inside the routine *MultiLoop* the following operations are repeated cyclically:

- (i) Volume of cells V_k , cell masses ($m_k = \sum_i m_{ik}$), and node masses ($m_i = \sum_{k \in i} m_k / N_k$) are computed by routines *UpdateHydroVol* and *UpdateHydroMass*. In addition, a temporary version of array \mathbf{ct} is created, where all triangles that are part of a very distorted quadrangular cell are marked as independent.
- (ii) Cell density is computed from m_k and V_k . The specific internal energy is computed first at nodes, $e_i = E_i / m_i$, and then averaged in each cell, $e_k = \sum_{i \in k} e_i / N_k$.
- (iii) Cell temperatures T_k and pressures P_k , as well as their partial derivatives, $(\partial P / \partial \rho)_k$, $(\partial P / \partial e)_k$, $(\partial T / \partial \rho)_k$, and $(\partial T / \partial e)_k$,

⁹ Refer to UNIX literature for the **tar** command.

are computed by routine *HydroEOS*. For cells with only one material, these values are interpolated directly from tabulated values $T(\rho, e)$ and $P(\rho, e)$ generated externally, either from SESAME library [25] or by MPQEOS code [26]. For cells containing L different materials, we treat the materials as not mixed, assuming that mechanical and thermal equilibrium has been reached with uniform pressure and temperature inside the cell. We have to solve the system:

$$\left. \begin{aligned} T_k &= T_l(\rho_{lk}, e_{lk}) \\ P_k &= P_l(\rho_{lk}, e_{lk}) \end{aligned} \right\} \quad (l = 1, \dots, L),$$

$$m_k e_k = \sum_{l=1}^L m_{lk} e_{lk}, \quad \frac{m_k}{\rho_k} = \sum_{l=1}^L \frac{m_{lk}}{\rho_{lk}}. \quad (129)$$

For a mixture of ideal gases, this system can be solved analytically, but, for the general case, it must be solved iteratively to obtain P_k , T_k , and their derivatives with respect to ρ_k and e_k . In the current implementation, only the first iteration is performed. This approximation has been found satisfactory for most applications.

- (iv) Thermal capacities and temperatures at nodes, needed by radiation and heat transport algorithms, are evaluated. Firstly, the heat capacity of each node is computed adding contributions from cells that have the node as a vertex:

$$\left(\frac{\partial E}{\partial T} \right)_i = \sum_{k \in i} \frac{m_k}{N_k} \left(\frac{\partial e}{\partial T} \right)_k. \quad (130)$$

In a similar way, reference values of internal energy and temperature are obtained:

$$E_i^* = \sum_{k \in i} \frac{m_k e_k}{N_k}, \quad (131)$$

$$T_i^* = \left(\sum_{k \in i} \frac{m_k T_k}{N_k} \left(\frac{\partial e}{\partial T} \right)_k \right) / \left(\sum_{k \in i} \frac{m_k}{N_k} \left(\frac{\partial e}{\partial T} \right)_k \right). \quad (132)$$

In general, $E_i \neq E_i^*$, so that the node temperature is corrected by

$$T_i = T_i^* + (E_i - E_i^*) / \left(\frac{\partial E}{\partial T} \right)_i. \quad (133)$$

Notice that this algorithm does not diffuse numerically the energy (E_i at nodes is not changed when computing T_i), and for ideal gases (when $e = CT$, with constant and uniform C) the value obtained at nodes is exactly $T_i = E_i / C m_i$. Also for more complex situations, this has been found to work reasonably.

- (v) Hydrodynamic motion, computed by routine *HydroMotion*, takes place as described in Section 4, the values of \vec{r}_i , \vec{v}_i and E_i being advanced by this substep.
- (vi) Optionally, routine *HydroEuler* allows to project, using a conservative low-order scheme, the hydrodynamics variables into a new grid supplied by a user's routine. This grid must preserve the original topology and has to be close enough to the previous grid (volume fluxes through interfaces are required to be smaller than cell volumes). Using this option, "Arbitrary-Eulerian-Lagrangian" (ALE) hydrodynamics can be implemented.
- (vii) The generic ray-tracing routine *HydroHIB* computes the deposition of energy into matter. Rays (ion beams or laser light) are assumed to travel along straight trajectories (no refraction is taken into account). Only E_i is changed in this step.
- (viii) The transport of energy by both radiation and heat conduction is managed by routine *HydroTransport*. First, from the

current temperature distribution, the net radiative power deposited at nodes (absorption minus emission) Q_i and an estimate of the partial derivative $\partial Q_i / \partial T_i$ are computed by routine *radia*. Analogous values, corresponding to diffusive heat conduction, are computed by routine *heatcond* and added to the former ones. Finally, the SSI algorithm described in Section 2.5 is applied to obtain upgraded values of E_i and ϵ_i .

- (xi) While each of the above processes is carried out, a factor w_j , expressing a relative measure of the time step (current time-step divided by Courant–Friedrich limit, relative variation of density over a nominal variation, etc.) is computed. Ideally, the maximum of these values $\max_j w_j$ should be around 1. If it exceeds 1, the time step is repeated, starting from (v), halving the former time step.
- (x) Before starting a new cycle, a new value of the time step Δt_{new} is computed, as a function of the old one Δt_{old} , by a user defined function (by default, $\Delta t_{\text{new}} = \frac{1}{2} \Delta t_{\text{old}} / \max_j w_j$).
- (xi) At selected step numbers (fields **multi.out_steps** and **multi.out_times**) the relevant variables are written into binary files by routine *HydroOutput*.

7. Typical run

In order illustrate a typical application of the code, we present here the simulation of a laser-driven target configuration. It mimics to some extent a hohlraum target design (see Fig. 13) developed for first demonstration of ICF ignition and burn [2]. Here, early phases of hohlraum heating and radiatively driven capsule implosion are simulated, illustrating salient features of ICF target performance.

The target consists of a hollow cylindrical case (the so-called "hohlraum"), made of high-Z material (gold) and filled with gas (hydrogen). It encloses the capsule which is supposed to contain the fusion fuel in a real ICF target. Here we take it as a spherical shell made of low-Z material (beryllium) and filled also with low density gas. The target is irradiated by two sets of laser beams, arranged conically¹⁰ and delivering a total power of 10 TW of blue light, having a wavelength of 0.35 μm and an incidence angle of 40 degrees. The beams enter the case from the sides, pass through the gas creating a propagation channel, and finally heat the inner surface of the case. The high-Z walls re-emit a major portion of the absorbed energy to the cavity in form of soft X-ray radiation. This energy is partially absorbed by the capsule; other parts are lost by diffusion into the walls and through the entrance holes.

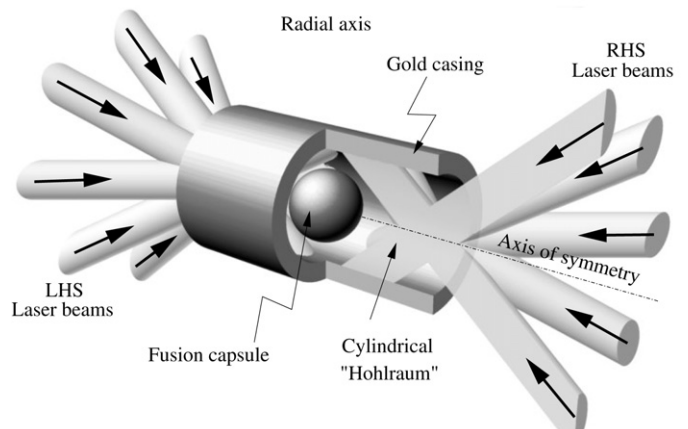


Fig. 13. Indirectly driven target.

¹⁰ We suppose that the number of laser beams distributed in azimuth is large enough to justify the cylindrical approximation.

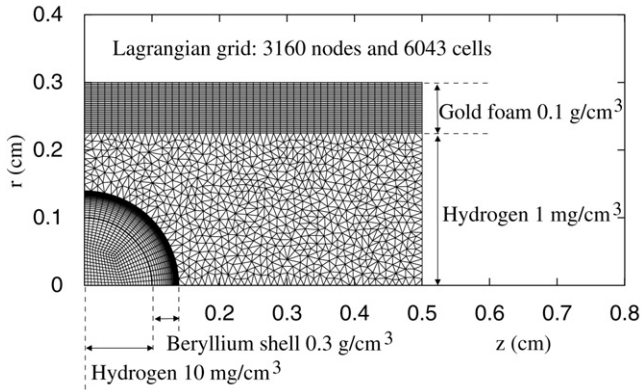


Fig. 14. Lagrangian grid at initial time.

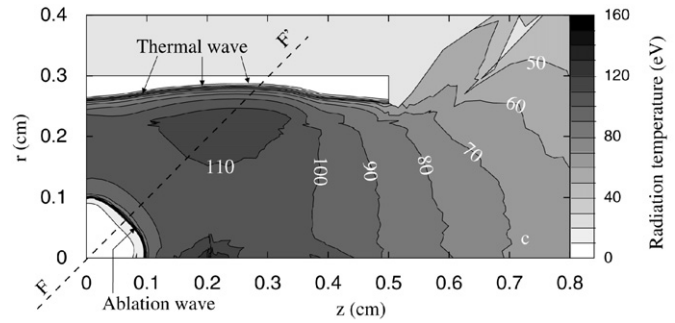


Fig. 17. Radiation temperature at 10 ns.

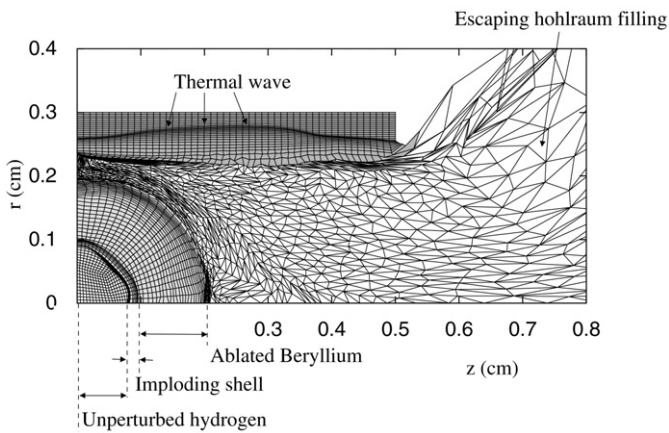


Fig. 15. Lagrangian grid at 10 ns.

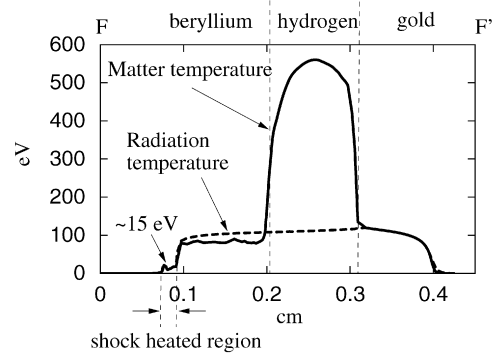


Fig. 18. Temperature along line FF' in Figs. 16 and 17.

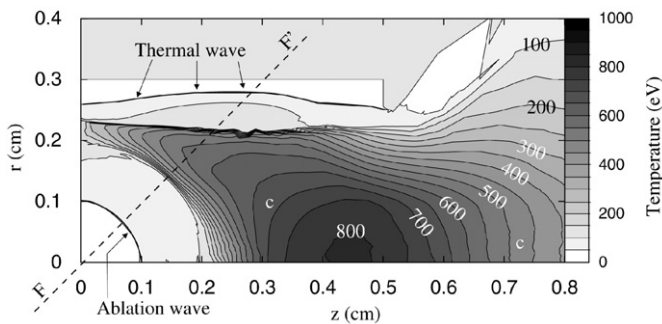


Fig. 16. Matter temperature at 10 ns.

Fig. 14 shows the Lagrangian grid at initial time. It is composed of regular regions with quadrangular cells and non-structured regions with triangular cells.¹¹ Also geometry and materials are specified; here we have chosen wall and capsule densities much smaller than in real targets (foams rather than solids); this is only for reasons of better visibility of the hydrodynamic evolution. Also material properties and laser absorption are modelled in a simplified way. Fig. 15 shows the grid after 10 ns of laser irradiation. One observes the expansion of the filling gas into vacuum and the ablation of the beryllium shell. The external part of the shell has been heated and expands into the cavity, while the inner part implodes. A thermal wave is seen to propagate in the gold wall.

Figs. 16 and 17 show snapshots of matter and radiation temperature at 10 ns. Matter temperature reaches a maximum value of 810 eV in the filling gas close to the Z-axis, where laser beams cross each other. Heating of the filling gas is due to light absorption, modelled here as inverse bremsstrahlung. In contrast, the temperature of the ablated beryllium and the gold walls is considerably lower. Although laser energy impinges directly on the gold wall, the gold temperature is limited due to its high opacity. It represents a balance between laser heating and cooling by radiation emission towards the optically thin cavity and radiation diffusion into the optically thick wall. These processes are described by the present code with accurate angular resolution. The resultant distribution of radiation inside the cavity appears to be quite uniform with temperatures ranging from 100 to 110 eV in this early phase of hohlraum heating. The radiation transport algorithm, despite the discrete number of directions used in the simulations ($\sum_{i=1}^M N_i = 32$) does not produce non-physical concentration of rays near the axis of the target (the structure visible at $z = 0.2$ cm is due to the collision between ablated material and hohlraum filling).

By design, the beryllium capsule is not irradiated directly by laser beams, but is driven indirectly by the highly isotropic radiation field. Nevertheless, one should notice that the shape of the imploding shell, considerably deviates from the spherical symmetry required for central fuel ignition. Apparently, the laser deposition in two rings is still not sufficiently symmetric and leads to a strong $l = 4$ asymmetry. In fact, the NIF design makes use of two cones of laser beams from each side with different opening angles. The present code may be used to investigate fine tuning of the laser irradiation geometry to improve implosion symmetry.

Fig. 18 shows the 2D temperature distributions along the line FF' indicated in Figs. 16 and 17. One can distinguish the optically thin region of the gas fill in which radiation and matter temperature decouple from each other, the optically thick region of the wall, where temperatures are in equilibrium as described at the end of Section 2 (Eqs. (9) to (11)), and the intermediate situation

¹¹ The unstructured region has been generated by the program **Triangle** [27].

occurring in the ablated beryllium, where photon mean free path is of the same order as layer thickness such that radiation and matter temperature become similar, but not exactly equal. The radiation temperature shows a slight gradient towards the center, indicating energy flux toward the capsule. This smooth behavior ends on both sides in sharp fronts of ablative heat waves. On the inner side, it drives a thin shell of beryllium (about 50% of the mass) which has been heated to 10–20 eV by a shock wave. Electronic heat transport is also included in the simulation, but is almost negligible except inside the fill gas.

In conclusion, this simulation demonstrates the various coexisting hydrodynamic and radiation regimes, for which this code has been developed.

Acknowledgements

This research has been supported by CICYT of Spain project ENE 2006-06339, and by EURATOM/CIEMAT as well as the EURATOM/Max-Planck-Institut für Plasmaphysik associations in the framework of the 'IFE Keep-in-Touch Activities'. The authors would like to express their thanks to S. Atzeni, M. Basko, J.J. Honrubia, J. Maruhn, T. Schlegel, and T. Yabe for many useful discussions during the development of this work. Also we are grateful to R.J. Shewchuk to allow us to use its code *Triangle* to generate the grid of the example in Section 7.

References

- [1] J.H. Nuckolls, L. Wood, A. Thiessen, G.B. Zimmerman, *Nature* 239 (1972) 129.
- [2] J. Lindl, *Phys. Plasmas* 2 (1995) 3933.
- [3] Th. Löwer, R. Sigel, K. Eidmann, I. Földes, S. Hüller, J. Massen, G. Tsakiris, S. Witkowski, W. Preuss, H. Nishimura, H. Shiraga, Y. Kato, S. Nakai, T. Endo, *Phys. Rev. Lett.* 72 (1994) 3186.
- [4] Th. Löwer, V.N. Kondrashov, M. Basko, A. Kendl, J. Meyer-ter-Vehn, R. Sigel, *Phys. Rev. Lett.* 80 (1998) 4000.
- [5] R. Sigel, Laser-generated intense thermal radiation, in: M.N. Rosenbluth, R.Z. Sagdeev, A.M. Rubenchik, S. Witkowski (Eds.), *Handbook of Plasma Physics*, in: *Physics of Laser Plasma*, vol. 3, North-Holland, Amsterdam, 1991.
- [6] S. Atzeni, J. Meyer-ter-Vehn, *The Physics of Inertial Fusion*, Oxford Science Publications, Oxford, 2004.
- [7] R. Ramis, R. Schmaltz, J. Meyer-ter-Vehn, *Comput. Phys. Comm.* 49 (1988) 475.
- [8] R. Ramis, J. Meyer-ter-Vehn, MULTI2D-a computer code for two-dimensional radiation hydrodynamics, Report MPQ174, Max-Planck-Institut für Quantenoptik, Garching, Germany, 1992.
- [9] R. Ramis, J. Ramírez, *Nucl. Fusion* 44 (2004) 720.
- [10] R. Ramis, J. Ramírez, G. Schurtz, *Laser and Part. Beams* 26 (2008) 113–126.
- [11] D.E. Hinkel, D.A. Callahan, A.B. Langdon, S.H. Langer, C.H. Still, E.A. Williams, *Phys. Plasmas* 15 (2008) 056314.
- [12] <http://server.faiia.upm.es/multi>.
- [13] M.M. Basko, DEIRA, A 1-D 3-T hydrodynamic code for simulating ICF targets driven by fast ion beams, Version 4, Institute for Theoretical and Experimental Physics, Moscow, 2001, p. 44.
- [14] S. Atzeni, *Plasma Phys. Control. Fusion* 27 (1987) 1535.
- [15] S. Atzeni, *Comput. Phys. Comm.* 43 (1986) 107.
- [16] A.I. Shestakov, J.L. Milovich, M.K. Prasad, *J. Comput. Phys.* 170 (2001) 81–111.
- [17] A. Dedner, P. Vollmöller, *J. Comput. Phys.* 178 (2002) 263–289.
- [18] D. Mihalas, B. Weibel Mihalas, *Foundations of Radiation Hydrodynamics*, Oxford University Press, Oxford, 1984.
- [19] L.I. Pomraning, *The Equations of Radiation Hydrodynamics*, Pergamon Press, Oxford, 1990.
- [20] Ya.B. Zel'dovich, Yu.P. Raizer, *Physics of Shock Waves and High-Temperature Phenomena*, Academic Press, New York, 1966.
- [21] E. Livne, A. Glasner, *J. Comp. Phys.* 58 (1985) 59.
- [22] R.D. Richtmyer, K.W. Morton, *Difference Methods for Initial Value Problems*, Interscience, New York, 1976.
- [23] R.C. Malone, R.L. McCrory, R.L. Morse, *Phys. Rev. Lett.* 34 (1975) 721.
- [24] A. Sunahara, J.A. Delettrez, C. Stoeckl, R.W. Short, S. Skupsky, *Phys. Rev. Lett.* 91 (2003) 095003.
- [25] SESAME, Report on the Los Alamos Equation-of-State Library, T-4 group, Report LALP-83-4, Los Alamos National Laboratory, Los Alamos, NM, 1983.
- [26] A. Kemp, J. Meyer-ter-Vehn, *Nucl. Instrum. Methods A* 415 (1998) 674–676.
- [27] J.R. Shewchuk, *Comput. Geom.* 22 (2002) 21–74.