

A NEW NUMERICAL CODE FOR HYDRODYNAMICAL 3D SIMULATIONS OF SUPERNOVA REMNANTS

P. Kostić

Astronomical Observatory, Volgina 7, 11060 Belgrade, Serbia

E-mail: pkostic@aob.rs

(Received: October 16, 2019; Accepted: December 9, 2016)

SUMMARY: We develop a 3D hydrodynamical code written in C programming language to study the expansion of supernova remnants (SNRs) in the surrounding medium. It is based on the MUSCL-Hancock finite volume scheme with the HLLC Riemann solver. The code initiates the supernova remnant already in the Sedov phase and simulates hydrodynamics of the subsequent remnant expansion. The simulation is optimized for studies of large scale interaction of a supernova remnant with the interstellar medium (ISM). After a detailed description of the code, and three tests of hydrodynamics, we present the results for a single remnant expanding into a uniform and fractally structured ISM, as the first application of the code. The simulation of SNR expanding in a uniform medium is compared with the Sedov law of expansion and Sedov self-similar solution to density, velocity and pressure profiles. The results indicate that the simulation presented here reproduces well the hydrodynamics of the supernova remnant expansion and is very practical due to its simplicity and speed. The SNR evolution in fractal ISM shows that clumps disturb the blast wave and produce interference of bow shocks, resulting in turbulent motions and inhomogeneities inside the remnant.

Key words. Methods: numerical – **ISM:** supernova remnants

1. INTRODUCTION

As a result of the supernova explosion a collisionless shock wave forms and expands into the surrounding interstellar medium (ISM) with speeds of up to $\approx 10^4$ km s⁻¹ (Raymond 1984). The ISM swept up by the shock is heated and, together with the supernova ejecta, fills the interior of this shock wave surrounded structure called supernova remnant (SNR). SNRs emit radiation over almost the entire span of the electromagnetic spectrum by synchrotron and thermal radiation mechanisms.

1.1. SNRs and ISM

Simplistic view of hydrodynamics of SNRs typically discerns four evolution phases for these objects (Woltjer 1972):

- (i) Free expansion – the ejecta can be approximated to expand freely into the surrounding ISM, forming a shock behind which the swept-up matter starts accumulating.
- (ii) Adiabatic (Sedov) phase – the swept-up matter is massive enough to take domination in the SNR dynamics, while the irradiated energy is still small compared to the explosion energy. Due to accumulation of the surrounding matter, the shock decelerates at rate $R \propto t^{2/5}$, conserving the energy.
- (iii) Radiative cooling – at this point the radiative energy losses become significant and the post-shock matter cools efficiently. As its thermal pressure decreases, the shell expansion decelerates preserving the radial momentum.

- (iv) Dissipation – eventually, the expansion slows down to the velocities of random motions of ISM (~ 10 km/s) and the remnant dissipates into ISM.

The total evolution lasts on the scales of $\sim 10^5 - 10^6$ yr. It is evident from this simple classification of evolutionary phases that hydrodynamics of SNR is heavily dependent on the ISM density (besides the energy of explosion), especially for the Sedov phase and later phases of evolution. This also reflects in a significant manner to the SNR properties and is evident in the empirical data since the density of ISM generally correlates with radiation fluxes due to the physics of radiation mechanisms involved (Pavlović et al. 2018, Arbutina and Urošević 2005, Berkhuijsen 1986).

The matter overrun by the SNR shock wave is more energized and clearly has a higher profile of observational signatures than the undisturbed surrounding ISM. A better understanding of the SNR–ISM interaction will enable the use of SNRs as proxies for the ISM structure. For recent studies in this direction see Orlando et al. (2019), Zhang and Chevalier (2019), Yu et al. (2015) and Slavin et al. (2017).

1.2. Hydrodynamical simulations of SNRs

Simulating different conditions of SNRs can be a good opportunity to investigate the impact of ISM environments on the SNR morphology or radiation. This way, the peculiar features of many observed supernova remnants can be better understood.

The astrophysical simulations are broadly divided to particle-based and grid-based, which is in essence a Lagrangian or Eulerian approach, respectively. In the former case, the fluids of interest are simulated as a conglomerate of individual particles that are subjected to gravitational and hydrodynamical acceleration while the latter approach calculates the fluid parameters at specific locations in space. For the purpose of a Sedov blast phenomena, the grid-based approach turned out much more successful than the particle-based approach (see Tasker et al. 2008). The present state-of-the-art grid-based simulation software implements the adaptive grid (mesh), where the grid resolution changes during the simulation, which enables greater dynamical range at a cost of a reasonably slower computation rate. At present, the well-known such codes, that are extensively used for simulating supernova remnants, are FLASH (Fryxell et al. 2000), PLUTO (Mignone et al. 2007, 2012), RAMSES (Teyssier 2002), ENZO (Bryan et al. 2014), etc. These codes offer different computational modules with parallel computing possibility and excellent accuracy. They can perform multitude of physical processes in various astrophysical problems. Consequently, some of them require installation of external libraries or some other necessary software.

The aim of this paper is to present the very basic hydrodynamical code that can be used for simulating expanding SNRs. It should be easily applied to generate large simulated samples of SNRs in or-

der to further advance the studies of the SNR–ISM interaction. The code is written in the C programming language and has a simple run settings and results output that are standard for each C-code. It requires no installation procedure and is compiled with a standard C compilers such as GCC. To achieve maximum comprehension, the code is described and explained in detail. It is not meant to be competitive with the state-of-the-art codes described above, but rather an easy to implement, simple and efficient approach to simulate the SNR–ISM interaction that gives similar results like more complex codes. The code is available to all interested practitioners in the field by just sending a demand to the author of this work.

We use a grid-based approach – the finite volume method of computational fluid dynamics (CFD). The next section is a short introduction to equations of fluid dynamics. In Section 3, we explain the basic theory of numerical methods for CFD and present the MUSCL-Hancock method. This will be presented in more detail in order to make a transparent guideline through the methodology of the code, making it suitable for future developments and upgrades by the community. In Section 4, we perform and discuss the test runs from the literature. The next two sections give description of the first application of the code: an adopted model of the interstellar environment will be presented in Section 5, and in Section 6 we will give details of the simulation setup. Finally, the conclusions of this work are given in the last section.

2. EQUATIONS OF FLUID DYNAMICS

Here we present the time-dependent three dimensional Euler equations that govern the dynamics of a compressible fluid, such as interstellar gas, given in a form of a system of non-linear hyperbolic conservation laws. These equations can be written using the two types of variables: primitive and conservative. The primitive variables are density $\rho(x, y, z, t)$, x -, y - and z -components of velocity $u(x, y, z, t)$, $v(x, y, z, t)$ and $w(x, y, z, t)$, and pressure $p(x, y, z, t)$. The conservative variables are density $\rho(x, y, z, t)$, x -, y - and z -components of momentum $\rho u(x, y, z, t)$, $\rho v(x, y, z, t)$ and $\rho w(x, y, z, t)$, and total energy per unit volume $E(x, y, z, t)$. The system of five conservation laws in differential form reads:

$$\begin{aligned}
 \rho_t + (\rho u)_x + (\rho v)_y + (\rho w)_z &= 0, \\
 (\rho u)_t + (\rho u^2 + p)_x + (\rho uv)_y + (\rho uw)_z &= 0, \\
 (\rho v)_t + (\rho uv)_x + (\rho v^2 + p)_y + (\rho vw)_z &= 0, \\
 (\rho w)_t + (\rho uw)_x + (\rho vw)_y + (\rho w^2 + p)_z &= 0, \\
 E_t + [u(E + p)]_x + [v(E + p)]_y + [w(E + p)]_z &= 0,
 \end{aligned} \tag{1}$$

where indices t , x , y and z represent partial derivatives (e.g. $\rho_t \equiv \partial\rho/\partial t$). The energy E is the sum of

internal and kinetic energy per unit volume:

$$E = \rho e + \frac{\rho(u^2 + v^2 + w^2)}{2}. \quad (2)$$

Here, e is the specific internal energy. This system of equations can be written in a compact form by introducing the column vector $\mathbf{U}(x, y, z, t)$ of conserved variables and flux vectors $\mathbf{F}(\mathbf{U})$, $\mathbf{G}(\mathbf{U})$ and $\mathbf{H}(\mathbf{U})$ in the x , y and z direction, respectively. Now, the equations read:

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y + \mathbf{H}_z = 0, \quad (3)$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E + p) \end{bmatrix}, \quad (4)$$

$$\mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E + p) \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \rho w \\ \rho vw \\ \rho w^2 + p \\ w(E + p) \end{bmatrix}.$$

Notice that in the system Eq. (1), including Eq. (2), there are more unknowns than equations, so a closure condition is needed to completely describe the physical process. In other words, there must be another expression that relates the internal energy e to other given quantities. That expression will be the equation of state which describes the nature of the medium. The interstellar medium is assumed to be calorically ideal gas, with caloric equation of state:

$$e = \frac{p}{\rho(\gamma - 1)}, \quad (5)$$

where γ is the so called *adiabatic index*, or *ratio of specific heats*, with $\gamma = 5/3$ describing the monoatomic gas. Although the adiabatic index in general is non-constant due to thermal and chemical processes (see Vaidya et al. 2015), it is considered constant in the presented simulation.

3. NUMERICAL METHODS

All finite volume (or finite difference) methods for non-linear systems of hyperbolic laws rely on the first such method presented by Godunov (Godunov 1959). For the reason of understanding and simplicity, we will shortly explain the basic concept of Godunov's method on time-dependent one dimensional Euler equations. For the exhaustive reading one should see the book by Toro (2009).

Lets say we have a spatial domain $x = [0, L]$ discretized in M computational cells $I_i =$

$[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ of regular size $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} = L/M$ with $i = 1, \dots, M$, and a time domain $t = [0, T]$ discretized in variable Δt time steps. At a certain time t^n , where the upper index n is the number of time step, all the data is written in a form $\mathbf{U}_i^n = \mathbf{U}(x_i, t^n)$. The Godunov's method for evolving this solution to $\mathbf{U}_i^{n+1} = \mathbf{U}(x_i, t^n + \Delta t)$ is given by this formula:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x} [\mathbf{F}_{i-\frac{1}{2}} - \mathbf{F}_{i+\frac{1}{2}}]. \quad (6)$$

The time step size Δt is calculated in every time step separately, because it depends on a maximum speed of a signal inside the domain at a given time step.

One can immediately see that the essential ingredient of Godunov's method is obtaining the fluxes on boundaries between the cells. This can be done using the state vectors on the left and right sides of the boundary and it is referred to as the *Riemann problem*. The flux $\mathbf{F}_{i+\frac{1}{2}}$ is calculated by solving the Riemann problem $RP(\mathbf{U}_i^n, \mathbf{U}_{i+1}^n)$. The methods for solving the Riemann problems are called Riemann solvers. Because the state values \mathbf{U}_i^n inside a cell domain are *piece-wise constants*, the method of Godunov is of first order accuracy.

3.1. MUSCL-Hancock method

In our simulation, we use MUSCL-Hancock scheme, an unsplit finite volume method for computational fluid dynamics. MUSCL stands for Monotonic Upstream-Centred Scheme for Conservation Laws. It is a second order extension of the Godunov first order upwind method in which the solution is updated in the same way as in the Godunov method, but in three dimensions it looks like this:

$$\begin{aligned} \mathbf{U}_{i,j,k}^{n+1} = & \mathbf{U}_{i,j,k}^n + \frac{\Delta t}{\Delta x} [\mathbf{F}_{i-\frac{1}{2},j,k} - \mathbf{F}_{i+\frac{1}{2},j,k}] \\ & + \frac{\Delta t}{\Delta y} [\mathbf{G}_{i,j-\frac{1}{2},k} - \mathbf{G}_{i,j+\frac{1}{2},k}] \\ & + \frac{\Delta t}{\Delta z} [\mathbf{H}_{i,j,k-\frac{1}{2}} - \mathbf{H}_{i,j,k+\frac{1}{2}}]. \end{aligned} \quad (7)$$

Given that a uniform grid is used, $\Delta x = \Delta y = \Delta z$ stands in our simulation.

The whole point of this scheme is to determine the intercell fluxes \mathbf{F} , \mathbf{G} and \mathbf{H} more precisely than the classical Godunov method. In order to do so, there are three main steps which lead to their calculation:

- (i) Data reconstruction.
- (ii) Evolution.
- (iii) Riemann problem.

In the first and third step there are different choices of methods one can use, so, by describing these steps, we will also present the methods we adopted.

3.1.1. Data reconstruction

The main task of this step is to replace the constant distribution of state values \mathbf{U}_i^n inside every cell with a convenient linear function so that the integral value stays the same (i.e. to replace piece-wise constant states with *piece-wise linear* functions, see Fig. 1). By “convenient” we mean to calculate the slope of this linear function using the state values of neighboring cells and the considered cell (in all three directions, independently). The point is to reconstruct the distribution of the conserved variables among cells as closely as possible to the *real* distribution. This is of key importance for a more accurate calculation of intercell fluxes. Actually, we can say that this type of reconstruction ensures the second order spatial accuracy of the scheme.

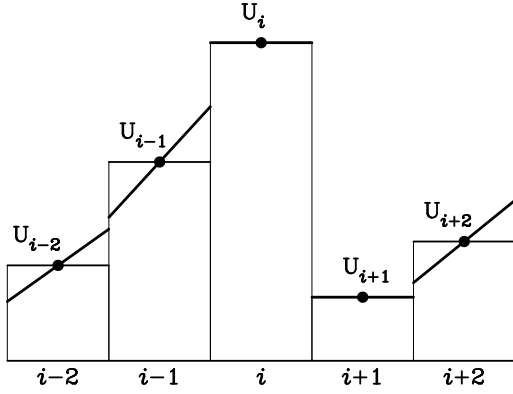


Fig. 1. The illustration of piece-wise linear data reconstruction. The (bold) slopes represent the reconstructed states \mathbf{U}_i using the van Leer slope limiter.

The reconstruction of these slopes is not a trivial task, because it must satisfy the *total variation diminishing* (TVD) condition which demands that no new local extrema is introduced after data reconstruction to avoid spurious oscillations (see Toro 2009). Thus, in certain parts of the domain, the slopes must be limited so that the left or right value cannot become a new local extremum among the cells. There are several methods to evaluate such limited slopes and the main few of them are:

(i) minbee (or minmod)

$$\phi_{\text{mb}}(r) = \begin{cases} 0, & r < 0, \\ r, & 0 \leq r < 1, \\ 1, & r \geq 1. \end{cases} \quad (8)$$

(ii) superbee

$$\phi_{\text{sb}}(r) = \begin{cases} 0, & r < 0, \\ 2r, & 0 \leq r < \frac{1}{2}, \\ 1, & \frac{1}{2} \leq r < 1, \\ r, & 1 \leq r < 2, \\ 2, & r \geq 2. \end{cases} \quad (9)$$

(iii) van Leer

$$\phi_{\text{vl}}(r) = \begin{cases} 0, & r \leq 0, \\ \frac{2r}{1+r}, & r > 0. \end{cases} \quad (10)$$

where:

$$r = r(i) = \frac{\mathbf{U}_i^n - \mathbf{U}_{i-1}^n}{\mathbf{U}_{i+1}^n - \mathbf{U}_i^n} \quad (11)$$

and $\phi(r)$ is the slope limiter that multiplies the slope $(\mathbf{U}_{i+1}^n - \mathbf{U}_i^n)/\Delta x$ in order to obtain the limited slope. Here, index i represents the considered coordinate. Finally, after obtaining slopes, the values at the left and right boundary of the cell can be extrapolated:

$$\begin{aligned} \mathbf{U}_{i,j,k}^{x-} &= \mathbf{U}_{i,j,k}^n - \frac{1}{2}\Delta_i, & \mathbf{U}_{i,j,k}^{x+} &= \mathbf{U}_{i,j,k}^n + \frac{1}{2}\Delta_i, \\ \mathbf{U}_{i,j,k}^{y-} &= \mathbf{U}_{i,j,k}^n - \frac{1}{2}\Delta_j, & \mathbf{U}_{i,j,k}^{y+} &= \mathbf{U}_{i,j,k}^n + \frac{1}{2}\Delta_j, \\ \mathbf{U}_{i,j,k}^{z-} &= \mathbf{U}_{i,j,k}^n - \frac{1}{2}\Delta_k, & \mathbf{U}_{i,j,k}^{z+} &= \mathbf{U}_{i,j,k}^n + \frac{1}{2}\Delta_k, \end{aligned} \quad (12)$$

where:

$$\Delta_l = \phi_{\text{vl}}(\mathbf{U}_{l+1}^n - \mathbf{U}_l^n), \quad l = i, j, k, \quad (13)$$

represents the extrapolation term in the respective direction. Note that Δ_l is a vector with five components. The signs “-” and “+” in the exponents represent the left and right extrapolated values within a cell. These values are referred to as *boundary extrapolated values*.

3.1.2. Evolution

In this step the boundary extrapolated values are evolved for a time step $\frac{1}{2}\Delta t$ to achieve the second order time accuracy of the scheme:

$$\begin{aligned} \hat{\mathbf{U}}_{i,j,k}^{l\pm} &= \mathbf{U}_{i,j,k}^{l\pm} + \frac{\Delta t}{2\Delta x} [\mathbf{F}(\mathbf{U}_{i,j,k}^{x-}) - \mathbf{F}(\mathbf{U}_{i,j,k}^{x+})] \\ &\quad + \frac{\Delta t}{2\Delta x} [\mathbf{G}(\mathbf{U}_{i,j,k}^{y-}) - \mathbf{G}(\mathbf{U}_{i,j,k}^{y+})] \\ &\quad + \frac{\Delta t}{2\Delta x} [\mathbf{H}(\mathbf{U}_{i,j,k}^{z-}) - \mathbf{H}(\mathbf{U}_{i,j,k}^{z+})], \end{aligned} \quad (14)$$

for $l = x, y, z$. Note that the last three terms for any l are the same, because the fluxes are obtained from the boundary extrapolated values within the same cell.

3.1.3. Riemann problem

Now we have a conventional Riemann problem to solve in order to obtain the intercell flux $\mathbf{F}_{i+\frac{1}{2}}$. The left and right state of the Riemann problem at the interface $x_{i+\frac{1}{2}}$ are $\mathbf{U}_L \equiv \hat{\mathbf{U}}_i^{x+}$ and $\mathbf{U}_R \equiv \hat{\mathbf{U}}_{i+1}^{x-}$.

Solving a Riemann problem is a procedure to determine the intermediate state(s) between two initial piecewise constant states, with a single discontinuity at the contact, after some time. It is explained very well by the Sod shock tube experiment and also theoretically (Sod 1978, Toro 2009). The Riemann problem can be solved exactly (the Riemann exact solver), but as it is computationally expensive, many approximate Riemann solvers have been developed. Some of them are HLLE, HLLC, PVRs, Roe solver, Osher solver, etc. Our simulation uses the HLLC Riemann solver (short of Harten-Lax-van Leer-Contact) so we will shortly present its procedure for obtaining the intercell flux.

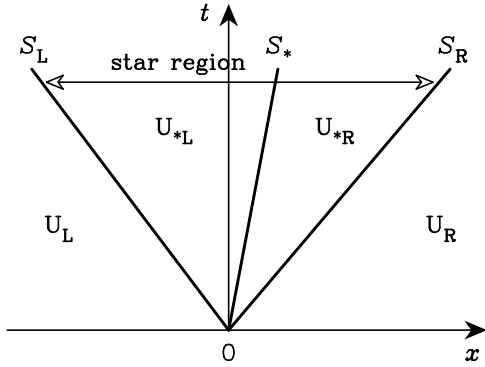


Fig. 2. The three-wave solution of the Riemann problem. The region between S_L and S_R , the star region (named after the asterisk in index), is divided in two regions by the middle wave (contact discontinuity) of speed S_* .

The HLLC solver assumes, for the solution, a wave configuration of three waves separating four state vectors, namely \mathbf{U}_L , \mathbf{U}_{*L} , \mathbf{U}_{*R} and \mathbf{U}_R (see Fig. 2). Depending on the wave structure, one of these states will be the intermediate state with its evaluated flux $\mathbf{F}_{i+\frac{1}{2}}^{\text{hllc}}$. To find the right flux one has to perform the following three steps:

- (i) Pressure estimate.

The pressure has no jump at the middle wave ($p_{*L} = p_{*R}$) so there is just one pressure p_* in the star region, which can be estimated in different ways, one of which is (Shen et al. 2016):

$$p_*^{\text{hllc}} = \frac{\alpha_R p_L - \alpha_L p_R - \alpha_L \alpha_R (u_L - u_R)}{\alpha_R - \alpha_L}, \quad (15)$$

where:

$$\begin{aligned} \alpha_L &= \rho_L (S_L - u_L), \\ \alpha_R &= \rho_R (S_R - u_R). \end{aligned} \quad (16)$$

For S_L and S_R see the next paragraph.

- (ii) Wave speed estimate.

In this step the speeds of three waves S_L , S_R and S_* are estimated. These speeds can be

computed as (Shen et al. 2016):

$$\begin{aligned} S_L &= \min(u_L - a_L, u_R - a_R), \\ S_R &= \max(u_L + a_L, u_R + a_R), \\ S_*^{\text{hllc}} &= \frac{\alpha_R u_R - \alpha_L u_L + p_L - p_R}{\alpha_R - \alpha_L}, \end{aligned} \quad (17)$$

where a_L and a_R are the left and right sound speed.

- (iii) HLLC flux.

Finally, the flux is computed as:

$$\mathbf{F}_{i+\frac{1}{2}}^{\text{hllc}} = \begin{cases} \mathbf{F}_L, & 0 \leq S_L \\ \mathbf{F}_{*L}, & S_L \leq 0 \leq S_* \\ \mathbf{F}_{*R}, & S_* \leq 0 \leq S_R \\ \mathbf{F}_R, & 0 \geq S_R \end{cases}, \quad (18)$$

where:

$$\mathbf{F}_L = \mathbf{F}(\mathbf{U}_L), \quad \mathbf{F}_R = \mathbf{F}(\mathbf{U}_R), \quad (19)$$

and:

$$\begin{aligned} \mathbf{F}_{*L} &= \mathbf{F}_L + S_L (\mathbf{U}_{*L} - \mathbf{U}_L), \\ \mathbf{F}_{*R} &= \mathbf{F}_R + S_R (\mathbf{U}_{*R} - \mathbf{U}_R). \end{aligned} \quad (20)$$

There are different methods for pressure and wave speed estimation. For more details on the HLL and HLLC solver see Shen et al. (2016) and Toro (2009). Once the intercell fluxes \mathbf{F} , \mathbf{G} and \mathbf{H} are obtained, the solution can be updated to the next time step according to Eq. (7).

3.1.4. Time step size

For a one dimensional problem, the time step size Δt must satisfy the condition

$$\Delta t \leq \frac{\Delta x}{S_{\max}^n}, \quad (21)$$

where S_{\max}^n is the largest wave speed throughout the domain at time level n , calculated as:

$$S_{\max}^n = \max_{i,j,k} \{ |u_{i,j,k}^n| + a_{i,j,k}^n, |v_{i,j,k}^n| + a_{i,j,k}^n, |w_{i,j,k}^n| + a_{i,j,k}^n \}. \quad (22)$$

Here, $a_{i,j,k}^n$ is the sound speed in the cell $I_{i,j,k}$ at time level n .

It is important not to violate the condition Eq. (21) to avoid spurious oscillations, so there is a number called the *CFL coefficient* (from Courant-Friedrichs-Lewy condition), $C_{\text{cfl}} \leq 1$, usually around 0.9 so that

$$\Delta t = C_{\text{cfl}} \frac{\Delta x}{S_{\max}^n}. \quad (23)$$

However, in three dimensions there is more conservative constraint on the CFL coefficient due to scheme stability reasons, $C_{\text{cfl}} \leq \frac{1}{3}$, so we use the value of 0.25 (in all presented simulations).

3.1.5. Pressure at cell boundary and positivity preservation

The method described above works perfectly when the internal energy is not a very small fraction of kinetic energy. However, in cells where the kinetic energy greatly dominates over the internal energy, the slopes may be high enough to produce negative pressure at a cell boundary. In this case, the simulation might collapse (e.g. the sound speed at the boundary cannot be computed) or leads to unphysical solution, so it is necessary to additionally limit the slopes to bring the value of pressure at least to zero. This is done by finding a slope limiting coefficient k at the given boundary, such that the local internal energy equals zero:

$$E \pm k\Delta E - \frac{m_{\text{tot}}^2}{2(\rho \pm k\Delta\rho)} = 0, \quad (24)$$

with:

$$m_{\text{tot}}^2 = (\rho u \pm k\Delta(\rho u))^2 + (\rho v \pm k\Delta(\rho v))^2 + (\rho w \pm k\Delta(\rho w))^2, \quad (25)$$

where ρ , ρu , ρv , ρw and E are the average states ($\mathbf{U}_{i,j,k}^n$) of the cell in question and $\Delta\rho$, $\Delta(\rho u)$, $\Delta(\rho v)$, $\Delta(\rho w)$ and ΔE are the limited increments obtained in the Section 3.1.1 (see Eq. (12)). The coefficient k is obtained as solution to the quadratic Eq. (24),

$$k = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}, \quad (26)$$

where:

$$\begin{aligned} A &= 2\Delta E\Delta\rho - [\Delta(\rho u)]^2 - [\Delta(\rho v)]^2 - [\Delta(\rho w)]^2, \\ B &= \pm 2[\rho\Delta E + E\Delta\rho - \rho u\Delta(\rho u) - \rho v\Delta(\rho v) \\ &\quad - \rho w\Delta(\rho w)], \\ C &= 2E\rho - (\rho u)^2 - (\rho v)^2 - (\rho w)^2. \end{aligned} \quad (27)$$

In Eqs. (24), (25) and (27), “ \pm ” covers the left ($-$) and right ($+$) cell boundary.

From the two solutions for k , one should choose $0 < k < 1$. Then, the slope limiters ϕ (see Section 3.1.1) for all five states in the cell should be constrained to $k\phi$, but just for the direction in which the negative pressure occurs.

3.1.6. Boundary conditions

Besides the initial values, for updating the solution by Godunov method, we need *boundary conditions* in order to calculate fluxes at edges of the simulation box. These conditions, which are set at every time step, are problem specific and they usually belong to one of the three types:

(i) Reflective.

The states in boundary cells of this type are like mirrors of the first neighboring cells inside

the domain. Let \mathbf{U}_0^n and \mathbf{U}_{M+1}^n be the left and right boundary states of an one-dimensional domain \mathbf{U}_i^n , $i = 1, \dots, M$. Then, they have values

$$\mathbf{U}_0^n = \begin{bmatrix} \rho_1^n \\ -(\rho u)_1^n \\ E_1^n \end{bmatrix}, \quad \mathbf{U}_{M+1}^n = \begin{bmatrix} \rho_M^n \\ -(\rho u)_M^n \\ E_M^n \end{bmatrix}. \quad (28)$$

In a 3D case everything is the same except that the components of momentum that are parallel to the boundary surface keep the same sign.

(ii) Transmissive.

This boundary condition allows the density, momentum and energy to flow out of the domain with the velocity from the last cell. This case is trivial, i.e. the boundary states are a copy of the neighboring cell states.

$$\mathbf{U}_0^n = \begin{bmatrix} \rho_1^n \\ (\rho u)_1^n \\ E_1^n \end{bmatrix}, \quad \mathbf{U}_{M+1}^n = \begin{bmatrix} \rho_M^n \\ (\rho u)_M^n \\ E_M^n \end{bmatrix}. \quad (29)$$

(iii) Inflow.

In case of a constant inflow from one side of the box, one should set the custom values for the boundary cells.

The analogous method is used when obtaining boundary extrapolated values at boundary cells. For example, for a reflective boundary $(\rho u)_{0,R}^n = -(\rho u)_{1,L}^n$ (this is done after the evolution step). All three types of boundary conditions will be utilized in a test in Section 4.2.

4. VALIDATION TESTS

In order to validate the code, we perform three tests that utilize the described in previous section. The first one, called the Sod shock tube problem, is a common test for validation of the HD/MHD codes. The second test is crushing of a spherical cloud by a flat shock wave and the third test shows an expansion of the clumpy ejecta of the supernova explosion.

4.1. Test I - Sod shock tube problem

The Sod problem (Sod 1978) is a common and basic test for the accuracy of HD codes, because it shows the code’s ability to resolve the shock wave, contact discontinuity (CD) and rarefaction wave. For an accurate code it is of crucial importance to capture the shocks and CDs within a small number of cells (or zones) and to produce the correct profile for rarefaction. Higher the number of cells across the shock or CD – higher the code’s diffusivity, which means that the real effects like the Kelvin-Helmholtz (KH) and Rayleigh-Taylor (RT) instabilities are going to be harder (or even impossible) to produce. The test will show that the diffusivity much depends on the choice of the type of data reconstruction.

The Sod problem is the one-dimensional flow discontinuity problem and here it will be performed exactly as in Section 7.2 of Fryxell et al. (2000). The 1D domain of the unit length is divided in 256 cells. The fluid, initially at rest, has a density and pressure jump in the middle of the domain. The left and right initial states are:

$$\begin{bmatrix} \rho_L \\ u_L \\ p_L \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \rho_R \\ u_R \\ p_R \end{bmatrix} = \begin{bmatrix} 0.125 \\ 0 \\ 0.1 \end{bmatrix}, \quad (30)$$

and the ratio of specific heats is $\gamma = 1.4$.

The results at $t = 0.2$ are shown on Fig. 3 for the van Leer (left) and superbee (right) reconstructions. Fig. 3 shows the density, specific internal energy and velocity profiles. In both left and right panels it can be seen that the shock wave is captured within $\sim 2 - 3$ cells, but for the contact discontinuity the difference between two reconstructions is much bigger. For the van Leer limiter CD is resolved with ~ 8 cells while using the superbee limiter reduced this region to ~ 4 cells (for comparison, in the FLASH code test from Fryxell et al. (2000), CD also covers just $\sim 2-3$ cells). The consequence is that the van Leer limiter produces much more diffuse solutions, which will be more clearly seen in Test II of this section. However, the superbee limiter is known to often produce spurious oscillations near discontinuities and tends to produce steps in smooth regions

(it's too compressive). In Test II it will be used in comparison with van Leer to illustrate the difference in diffusivity.

4.2. Test II - Crushing of spherical cloud by shock wave

This test is based on a simulation by Orlando et al. (2005). The homogeneous spherical cloud of radius 1 pc, density 1 H cm^{-3} and temperature 10^3 K is in pressure balance with an ambient medium of density $\rho_0 = 0.1 \text{ H cm}^{-3}$ and temperature of 10^4 K . The flat shock wave of Mach number $M = 50$ is approaching and crushing the cloud. The density, velocity and pressure in the upstream are:

$$\begin{aligned} \rho &= \frac{\gamma + 1}{\gamma - 1} \rho_0, \\ u &= \frac{2}{\gamma + 1} v_{\text{sh}} \quad (v_{\text{sh}} = Ma), \\ p &= \frac{2}{\gamma + 1} \rho_0 v_{\text{sh}}^2. \end{aligned} \quad (31)$$

The simulation box covers just one quadrant of the cloud, whose center is in $(x, y, z) = (0, 0, 0)$. The boundaries at $x = 0$ and $y = 0$ are reflective,

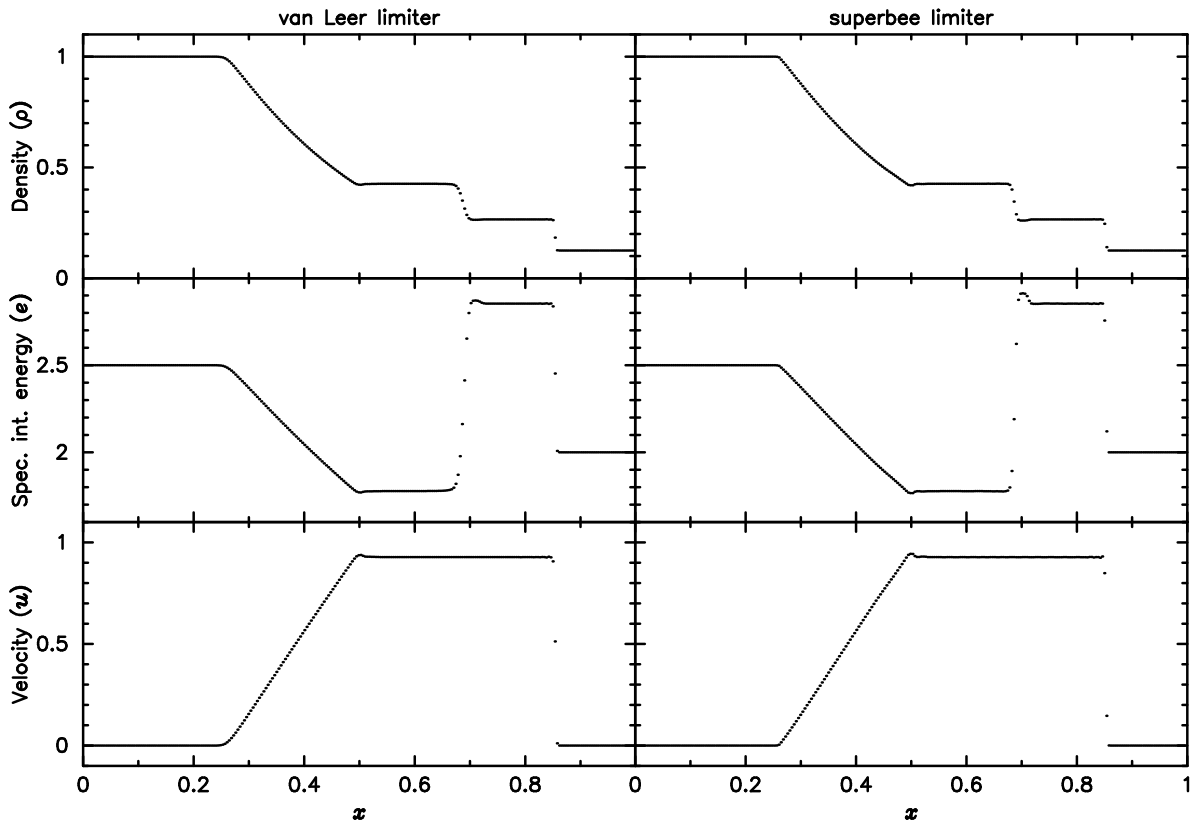


Fig. 3. The results of the 1D Sod shock tube test at $t = 0.2$ with the van Leer (left) and superbee (right) slope limiters. Figure shows profiles of density, specific internal energy and velocity.

$x = x_{\max}$, $y = y_{\max}$ and $z = z_{\max}$ are transmissive, and $z = z_{\min}$ is the constant inflow of the shock wave. The box size is $2 \times 2 \times 5.5$ pc with resolution of $200 \times 200 \times 550$ cells. In order to avoid a stairs-like density distribution at the edge of the cloud, these densities are calculated as the averages of $10 \times 10 \times 10$ subcells for every grid-cell placed at the cloud border.

The results are showed in Fig. 4, with left and right half panels representing van Leer and superbee reconstructions, respectively. At first, it is seen that interaction of the blast wave with the cloud has produced a transmitted and a reflected (bow) shock. Soon, the shock wave encompasses the cloud, compressing it, followed by a conical self-reflection of the

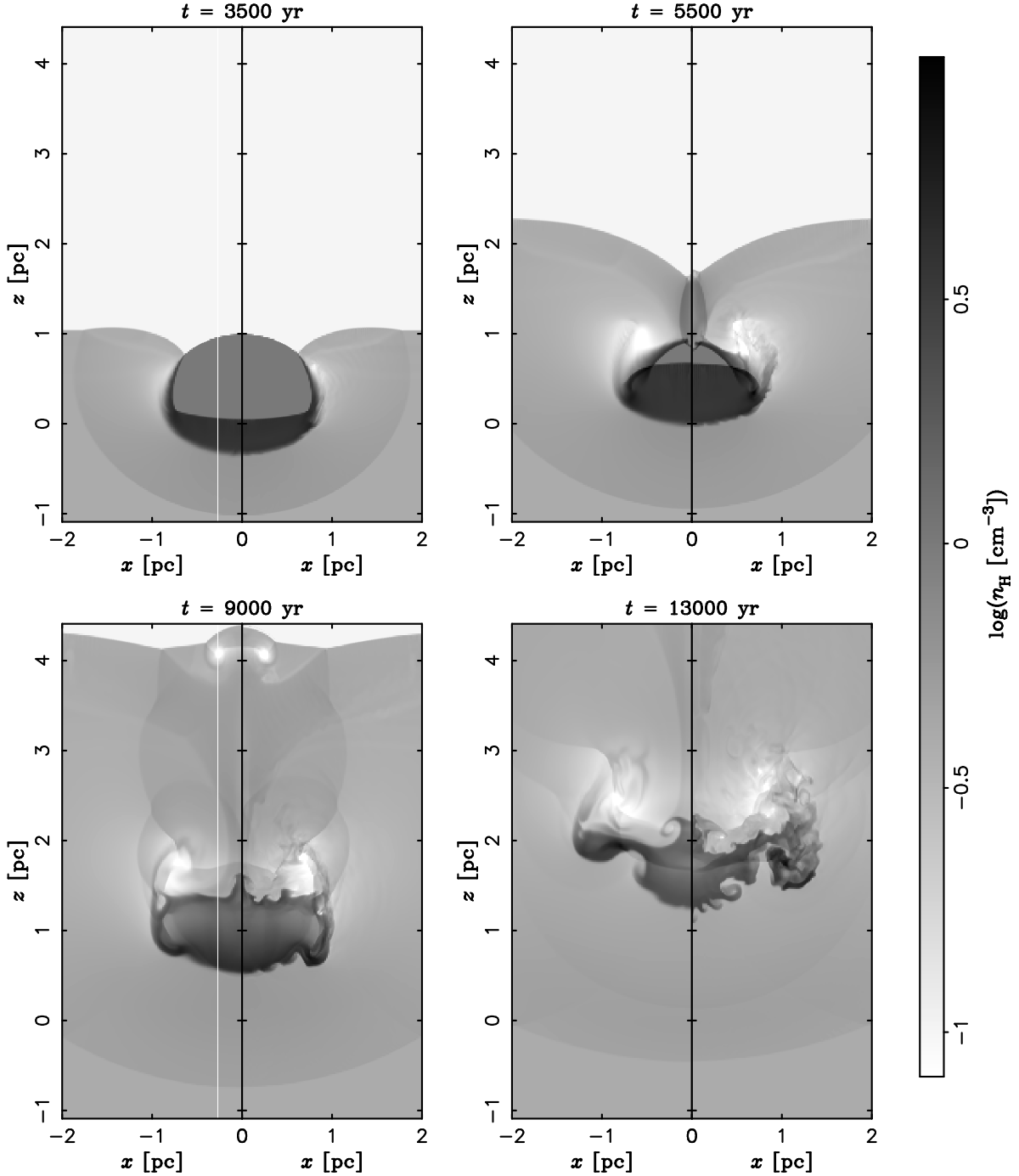


Fig. 4. Density slices from the cloud crushing test. The frames are caught at 3500, 5500, 9000 and 13000 yr after the beginning of simulation. If we adopt the characteristic cloud-crushing time from Orlando et al. (2005), $\tau_{\text{cc}} = 5400$ yr, these times are $0.62\tau_{\text{cc}}$, $0.99\tau_{\text{cc}}$, $1.64\tau_{\text{cc}}$ and $2.38\tau_{\text{cc}}$ (from the shock-cloud collision). Left and right half panels represent van Leer and superbee reconstructions, respectively.

primary shock. Then the cloud starts expanding and developing hydrodynamic instabilities due to vorticity deposited by the primary shock. Finally, the deformation of the cloud eventually leads to its fragmentation and destruction. These images match quite good with the Fig. 3 in Orlando et al. (2005), considering the resolution of 100 cells per cloud radius (they had 132), which is the minimum adequate resolution predicted by Klein et al. (1994) and Mac Low et al. (1994). Apparently, the main distinction between the van Leer and superbee reconstructions is the amount of produced turbulence and instabilities, origins of which are differences in diffusivity (on the left half panels, the instabilities at the borders of the cloud are suppressed).

4.3. Test III - Expansion of the clumpy ejecta

This test is based on a simulation of Orlando et al. (2012) (their model EX-C1.5-D1). Dwarkadas and Chevalier (1998) represented the ejecta density structure of Type Ia SNe by exponential profile. For deflagration models it is a power-law profile with an index $n = 7$ (Chevalier 1983, Nomoto et al. 1984). Our initial setup consists of spherically symmetric ejecta with an exponential density profile, expanding with the velocity that is proportional to the radius (pseudo-Hubble flow, Hole et al. 2010):

$$\begin{aligned}\rho(r) &= Ae^{-r/R}, \\ v(r) &= \frac{r}{R}v_e,\end{aligned}\quad (32)$$

where R is the initial radius of the sphere and v_e is the ejecta velocity at R . If the ejecta mass $M_e (= 1.4M_\odot)$ and kinetic energy E_k are set, then

$$\begin{aligned}A &= \frac{M_e}{4\pi R^3(2 - 5/e)}, \\ v_e &= \left(\frac{4e - 10}{24e - 65} \frac{E_k}{M_e} \right)^{\frac{1}{2}}.\end{aligned}\quad (33)$$

The kinetic energy is set to 99% of the total energy ($E_0 = 10^{51}$ erg) in every cell. Further, the density of the ejecta is perturbed from cell to cell by a random factor between 0.625 and 1.5, with power-law probability distribution with index $n = -1$ (after the perturbation, the integrated ejecta mass stays M_e). The perturbation of density also affects the momentum and total energy in the cells, but the pressure stays smooth. This is how the clumpy structure of the ejecta, which triggers Rayleigh-Taylor instabilities at the contact discontinuity, is modeled (Orlando et al. 2012, Pavlović 2017). The ambient medium is homogeneous gas with density of 0.05 H cm^{-3} and temperature of 10^3 K .

The remnant is initialized as one octant of a sphere in order to achieve higher resolution for the same execution time. This requires that $x = 0$, $y = 0$ and $z = 0$ boundaries are reflective, while $x = x_{\max}$, $y = y_{\max}$ and $z = z_{\max}$ boundaries are transmissive.

4.3.1 Remapping

Unlike the Test II case, in Test III the spatial domain of the expanding SNR constantly grows, so due to the uniform grid, after some time it would be an exhausting task (in terms of time consumption) to simulate the SNR with the starting cell size. That is the reason why the adaptive mesh refinement was invented. Here, we present the approach known as remapping technique in which the mesh is adapting periodically to the remnant's size, so it keeps the number of cells constant throughout the simulation run. For example, it is used in Ono et al. (2013) for expanding the domain of the simulation from small central region of the star to its whole volume. The idea is to gradually extend the computational domain, keeping the total number of cells by remapping the physical quantities. In such a way, for the same computational cost, one can continue simulating phenomena in constant expansion. Ono et al. (2013) use the expansion factor of 1.2, suggesting that larger values (e.g. 2, used in Kifonidis et al. 2006) can introduce hydrodynamical diffusivity in some parts of domain. We used the factor of 2 and after checking the conservation of total mass, momentum and energy we found that one remapping does not introduce errors larger than factor of $\sim 10^{-5}$.

At the beginning of the run, the initial ejecta (0.5 pc in radius) are placed in a $2.5 \times 2.5 \times 2.5$ pc box with resolution of $250 \times 250 \times 250$ cells (so the initial radius covers 50 cells). When the remnant reaches a boundary of the box, the code performs rescaling of the whole box by turning $2 \times 2 \times 2$ cubes of cells into single cells (of double edge size) and adding the new cells so the grid keeps $250 \times 250 \times 250$ cells while the edge of the box doubles. Now, the run continues until the box boundary is reached again, and so on. We can refer to this procedure also as *mesh merging*. The merging is done by averaging the state values from eight cells. The added new cells are initialized with the states of the ambient medium. In the case of the inhomogeneous medium, which will be presented in Section 5, the new ambient medium, added in this way, is taken from the prepared data files.

This method does not affect the conservative nature of the scheme, it just adapts the resolution so the remnant at any scale can be efficiently simulated. Since the MUSCL-Hancock scheme is of second order accuracy, it can quickly converge to the satisfying level of accuracy with increasing resolution.

In Fig. 5, after nearly 1000 yr from the simulation start, we see the developed Rayleigh-Taylor instabilities at the contact discontinuity. Before reaching this size (8 pc in radius), there were two remapping steps when the remnant reached the radii of 2.5 and 5 pc. Without remapping, the number of cells per dimension would have to be doubled for the remnant to reach the double size, so it would require 2^3 times longer execution time. In addition, remapping also doubles the time step size, making the simulation 16 times faster than without remapping. Thus, this simulation is able to be accomplished on a standard performance laptop in less than 4 hours.

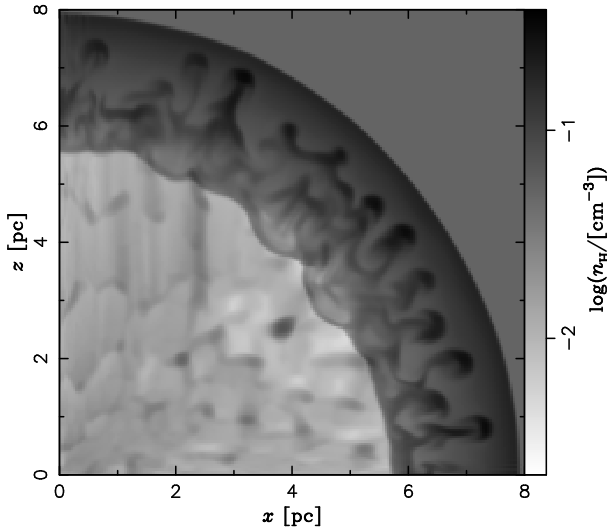


Fig. 5. Density slice from the clumpy ejecta expansion test. The SNR is ≈ 920 yr old. The Rayleigh-Taylor instabilities (mushroom-shaped structures) developed in the intershock region.

5. INTERSTELLAR MEDIUM

The simulations of SNRs are carried out in both uniform and inhomogeneous medium. In the case of uniform medium, the density is set to $\rho_0 = 0.25 \text{ H cm}^{-3}$ with a temperature $T_0 = 10^4 \text{ K}$. These numbers are typical values for warm ISM and are also used in Slavin et al. (2017) who simulated supernova remnants in a cloudy environment. Since the medium is set to have no momentum and kinetic energy, all energy is the internal (pressure energy). Unlike Slavin et al. (2017), who created a cloudy medium by randomly distributing the homogeneous spherical clumps of the same higher density (25 H cm^{-3}), but of different sizes, we model the molecular cloud by distributing the density as a hierarchical structure, creating the smooth fractal distribution of very different densities. Another way of modeling the inhomogeneous medium is given in Yu et al. (2015) in an MHD study of SNR expanding in a turbulent plasma with a relatively small amplitude of turbulence.

The modeling of inhomogeneous medium will be presented in the next two sections.

5.1. Model of molecular cloud

Apart from the uniform medium, the SNR simulations will be run in inhomogeneous medium that resembles the molecular clouds. Here, we present a model of such medium which is based on the work of Elmegreen (1997) and developed for application on SNRs in Kostić et al. (2016). As the researches from past few decades indicate, the interstellar medium, especially molecular clouds, show some kind of *fractal structure* (Elmegreen and Falgarone 1996, Stutzki et al. 1998, Blitz and Williams

1999). This has been confirmed by observations and simulations of ISM, and it is probably result of turbulence driven motions within the medium (Federrath et al. 2009).

By definition, the fractal structure is independent of the length scale. That's also the basic idea of making the fractal cloud in Elmegreen (1997). The method consists of placing N random points in a box, then placing the next N random points in a smaller boxes around the every of original points, and doing it repeatedly H times. Every box from a certain level of such hierarchy is L times smaller in edge length than the box from the previous level. At the end, just the points from the last level (their number is N^H) will be used as seeds for making the density distribution in the box.

The numbers N and H are chosen according to the simulation requirements, $N = 15$ and $H = 4$. More on these choices in Results and Discussion section. The number L is related to N as

$$N = L^{D_f}, \quad (34)$$

where D_f is called the *fractal dimension*. This quantity, and its possible influence on hydrodynamical evolution of supernova remnants, is discussed in more details in Kostić et al. (2016) and references therein. Shortly, it is a measure of fractal object's space-filling ability and characterizes the scaling ratio between the structure and its substructures. Based on the results of Federrath et al. (2009), we adopt the value $D_f = 2.3$.

5.2. Clump profile

The density distribution in the box is created in such a way that instead of every seed point (see the previous section) we place a predefined spherical density field of finite radius, a *clump*, and if two fields overlap they add to each other. After summation of all fields, the rest of the space in the box is filled with the so called *interclump* density which is set to 0.25 H cm^{-3} . Also, if there is a place within the clump domain with density below the interclump, then it is also assigned with interclump density.

For this clump density distribution which will produce the cloud-like structure, we adopted two variants, Gaussian and Lorentzian distribution, based on appearance in literature. For example, Williams et al. (1994) state that clump profiles are not greatly different from triaxial Gaussians. Also, Stutzki and Güsten (1990) assumed Gaussian shapes of clumps because of their Gaussian projections. On the other hand, Williams et al. (1995) found that clumps in Rosette Molecular Cloud have density profiles $\rho(r) \propto r^{-2}$ which is similar to the Lorentz profile, taken that real clumps must have cores of finite density. Parmentier (2011) states that various studies put forward the power-law density profile for molecular clumps $\rho(r) \propto r^{-p}$, with index p being in the range $1.5 \leq p \leq 2.5$. So, we run the simulations with both profiles, with the same spatial distribution of clumps, to see the difference. Because the Lorentz profile is wide, we make a profile cut-off at radius $r_{\text{clump}} = 3 \text{ pc}$.

To avoid placing the clumps too close to an initial SNR sphere, all the seed points within radius $r \leq 8$ pc from the SNR center are rejected. This secures the remnant from encountering the clumps until its radius is 5 pc, which is needed due to stabilization of initialized SNR profiles (see Section 6.1).

Regardless of the clump profile, the entire ISM is placed in pressure equilibrium by assigning the zero momentum and constant pressure to its cells. Considering that the speed of the SNR expansion is larger than that of motions in ISM, this assumption doesn't affect the SNR evolution. The value of the outside pressure (p_{out}) is the same as for the uniform medium case:

$$\frac{p_{\text{out}}}{k_B} = \rho_0 T_0 = 2500 \text{ K cm}^{-3}, \quad (35)$$

where k_B is the Boltzmann constant, ρ_0 is the concentration of atomic hydrogen in the interclump medium, and T_0 is its temperature (note that interclump medium is practically identical to the uniform medium case).

6. SIMULATION SETUP

This section reveals all details and settings of the simulations. All relevant values and parameters are given in Table 1.

Table 1. Simulation parameters.

Parameter	Description	Value
Model of molecular cloud		
H	model parameter	4
N	model parameter	15
D_f	fractal dimension	2.3
σ	Gaussian width	0.3
Γ	Lorentzian width	0.1
r_{clump}	cut-off clump radius	3 pc
m_{cloud}	total mass in clumps	$8.5 \cdot 10^4 M_{\odot}$
SNR and ISM		
E_0	SN energy	10^{51} erg
r_{sh}	initial SNR radius	1.5 pc
ρ_0	interclump density	0.25 H cm^{-3}
T_0	interclump temp.	10^4 K
γ	adiabatic index	$5/3$
Grid and scheme		
resolution	number of cells	300^3
box	final box edge	80 pc
initial box	initial box edge	20 pc
C_{cfl}	CFL coefficient	0.25
limiter	reconstruction type	van Leer

6.1. SNR initialization

The simulation starts with SNR of total energy $E_0 = 10^{51}$ erg and radius $r_{\text{sh}} = 1.5$ pc (radius at the shock). The initial states of the remnant are set by inputting analytical Sedov profiles for density, velocity and pressure into the grid (see the next section). However, from the start there is a small velocity and pressure perturbation that moves inwards and fades away when SNR reaches ≈ 5 pc. Then, the ‘‘true’’ numerical Sedov profiles are formed that are a bit smeared compared to the analytical ones. This smearing is normal numerical effect due to limited accuracy and finite cell size. The full spherical remnant is placed at the center of the $20 \times 20 \times 20$ pc box with resolution of $300 \times 300 \times 300$ cells. There are two remapping steps, so the final size of the box is 80 pc.

Before the remapping, there is a check whether the remnant can be moved in the x , y or z direction for a certain number of cells (with all the surrounding medium, of course) to stay in the finer mesh as long as possible. This ensures that the remnant will maximally develop before the remapping, so we avoid the situation of remapping the remnants that are undeveloped on one side, which can happen in a certain configurations of dense clumps.

In the next few paragraphs we introduce the Sedov self-similar solutions.

6.2. Sedov solution

The following solution applies for the point explosion in a uniform ambient medium. For a blast wave propagating through a density gradient $\rho = \rho_0 r^{-\omega}$ see Kamm and Timmes (2007). The Sedov solution, given in Sedov (1959), is a set of parametric equations for radius, density, velocity and pressure. Sedov gave them in a general form as a function of dimensionality ν , ratio of specific heats γ , and a parameter V which is related to the position between the center of the explosion and the shock wave. The range of variation of V inside this domain is:

$$\frac{2}{(\nu + 2)\gamma} \leq V \leq \frac{4}{(\nu + 2)(\gamma + 1)}. \quad (36)$$

The next four formulas (simplified after inserting $\nu = 3$ and $\gamma = 5/3$) give the radius r , density ρ , velocity v and pressure p , relative to their values immediately behind the shock, as a function of V :

$$\begin{aligned} \frac{r}{r_2} &= \left(\frac{10}{3}V\right)^{-\frac{2}{5}} \left(\frac{50}{3}V - 4\right)^{-\alpha_2} (5 - 10V)^{-\alpha_1}, \\ \frac{\rho}{\rho_2} &= \left(\frac{50}{3}V - 4\right)^{\alpha_3} (4 - 10V)^{\alpha_5} (5 - 10V)^{\alpha_4}, \\ \frac{v}{v_2} &= \frac{10}{3}V \frac{r}{r_2}, \\ \frac{p}{p_2} &= \left(\frac{10}{3}V\right)^{\frac{6}{5}} (4 - 10V)^{\alpha_5+1} (5 - 10V)^{\alpha_4-2\alpha_1}, \end{aligned} \quad (37)$$

where:

$$\alpha_1 = \frac{1}{10} - \frac{25}{12}\alpha_2, \quad \alpha_2 = -\frac{2}{13}, \quad \alpha_3 = \frac{9}{13}, \quad (38)$$

$$\alpha_4 = 15\alpha_1, \quad \alpha_5 = -6.$$

If we set energy of explosion E_0 , ambient density ρ_0 and radius of a remnant r_{sh} (or time after the explosion t), we can obtain the Sedov profiles $\rho(r)$, $v(r)$ and $p(r)$ numerically from Eq. (37) using the following relations:

$$r_2 = r_{\text{sh}} = 1.15 \left(\frac{E_0 t^2}{\rho_0} \right)^{\frac{1}{5}},$$

$$v_{\text{sh}} = \frac{2}{5} \frac{r_{\text{sh}}}{t},$$

$$\rho_2 = \frac{\gamma + 1}{\gamma - 1} \rho_0, \quad (39)$$

$$v_2 = \frac{2}{\gamma + 1} v_{\text{sh}},$$

$$p_2 = \frac{2}{\gamma + 1} \rho_0 v_{\text{sh}}^2,$$

where v_{sh} is speed of the shockwave. The first two equations in Eq. (39) are the Sedov law of expansion, and the expressions for ρ_2 , v_2 and p_2 are the Rankine-Hugoniot shock jump relations for strong shocks.

It is important to note that some corrections have to be made to initial Sedov densities in the vicinity of the remnant's center because in the Sedov solution the internal energy (and also temperature) at the center goes to infinity. These small corrections to density ensure a finite temperature as well as needfully lower sound speeds which are of immense importance at calculation of the time step size (as the sound speed actually dictates its size). This is done by introducing the maximal speed of sound allowed near the center of explosion:

$$a_{\text{max}} = \sqrt{\frac{p}{\gamma\rho}}, \quad (40)$$

and then calculating ρ from this equation. The limit a_{max} is chosen not to affect the stability of the Sedov-like profiles as the remnant evolves.

7. RESULTS AND DISCUSSION

First, we run the SNR in uniform medium to see if the numerical profiles resemble theoretical Sedov profiles. The Fig. 6 shows the numerical profiles, at radius $r \approx 5.5$ pc, which are in excellent agreement with analytical solutions. The peak of density at the shock is $\approx 20\%$ lower than theoretical value which is normal for this resolution of the grid. The time evolution of the SNR radius compared to the Sedov law of expansion, given by the first equation of Eq. (39), is shown in Fig. 7.

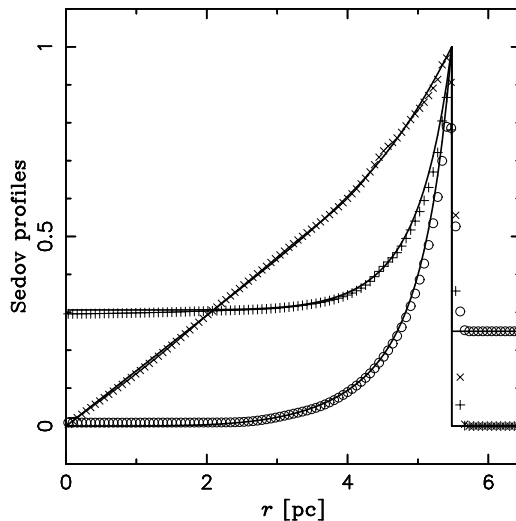


Fig. 6. Comparison of analytical Sedov profiles (lines) and numerical data. The density (\circ), velocity (\times) and pressure ($+$) profile.

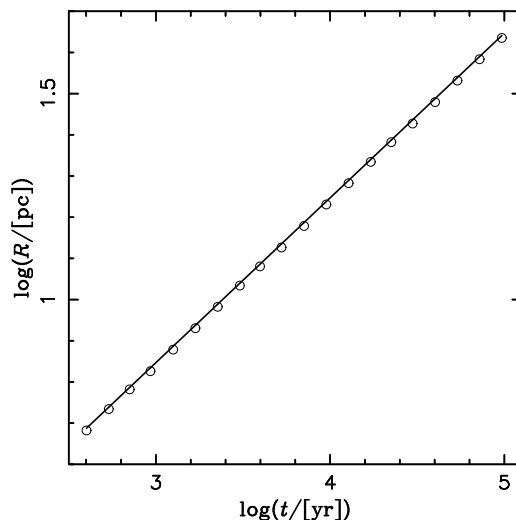


Fig. 7. Comparison of analytical Sedov law of expansion (line) and numerical data (circles).

The simulation of SNR evolution through the fractal cloud was carried out for two cases of the clump profile: Gaussian and Lorentzian. The Figs. 8, 9 and 10 show slices of density (ρ), velocity (v), and pressure (p) fields for these two cases. The figures show the slices of evolved remnants before the remappings (see the figure scales).

The first noticeable feature of the SNR evolving in the MC environment is that the shock front always tends to go towards the region of lower density. The shock velocity decreases significantly in contact with clumps, while the pressure increases. Inside the remnant we can see many elongated and convoluted shapes of structure as a result of interaction of the blast wave with clumps. Many features in the v and p field, that look like tenuous shock fronts crossing each other inside the remnant, are the reflected perturbations (bow shocks, like in Test II, Section 4.2)

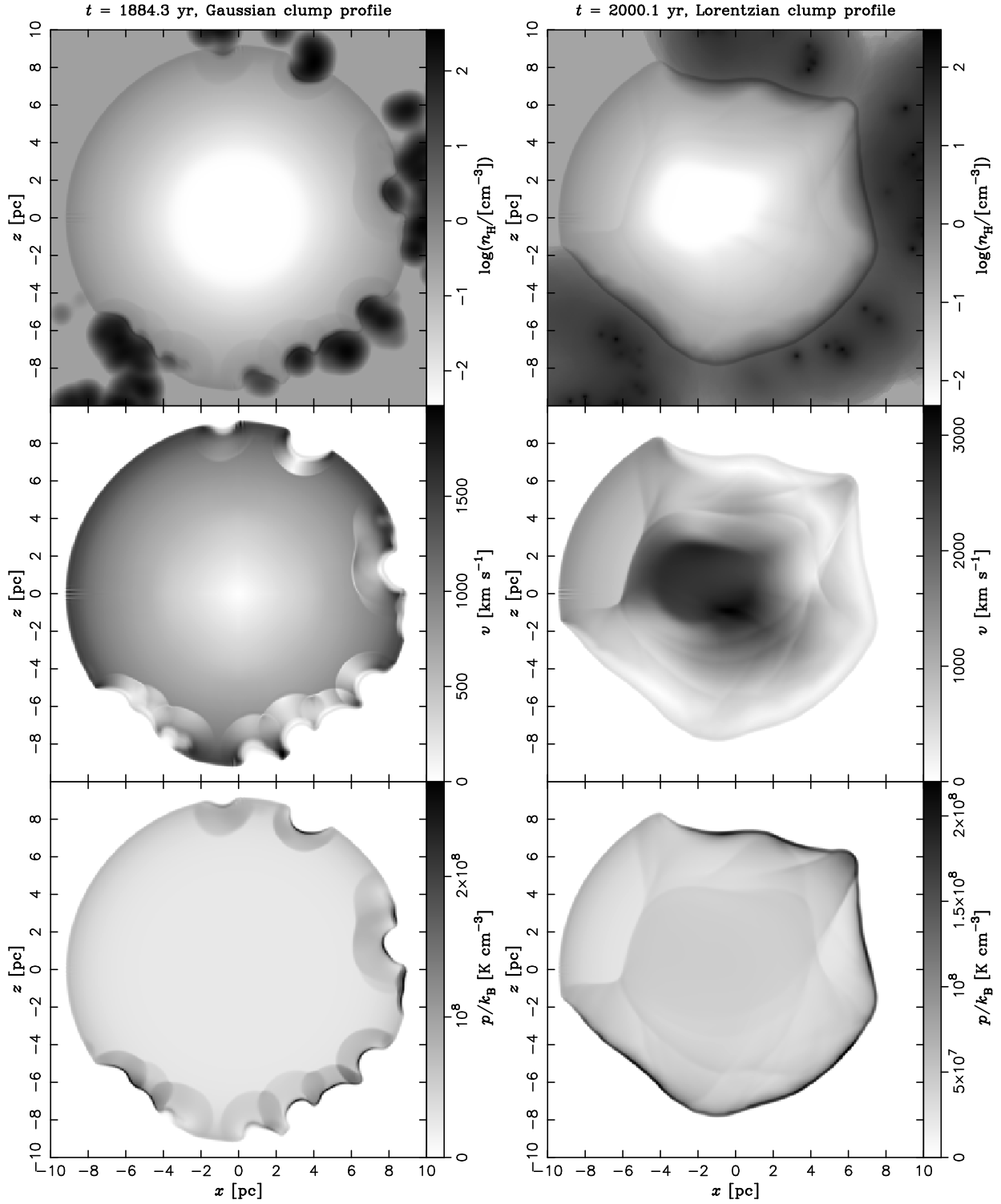


Fig. 8. Density, velocity and pressure fields for Gaussian (left) and Lorentzian (right) clump profile. The grayscale palette shows range of values between minimum and maximum cell value throughout the slice. The time t above the images is the simulation time from the beginning of SNR expansion.

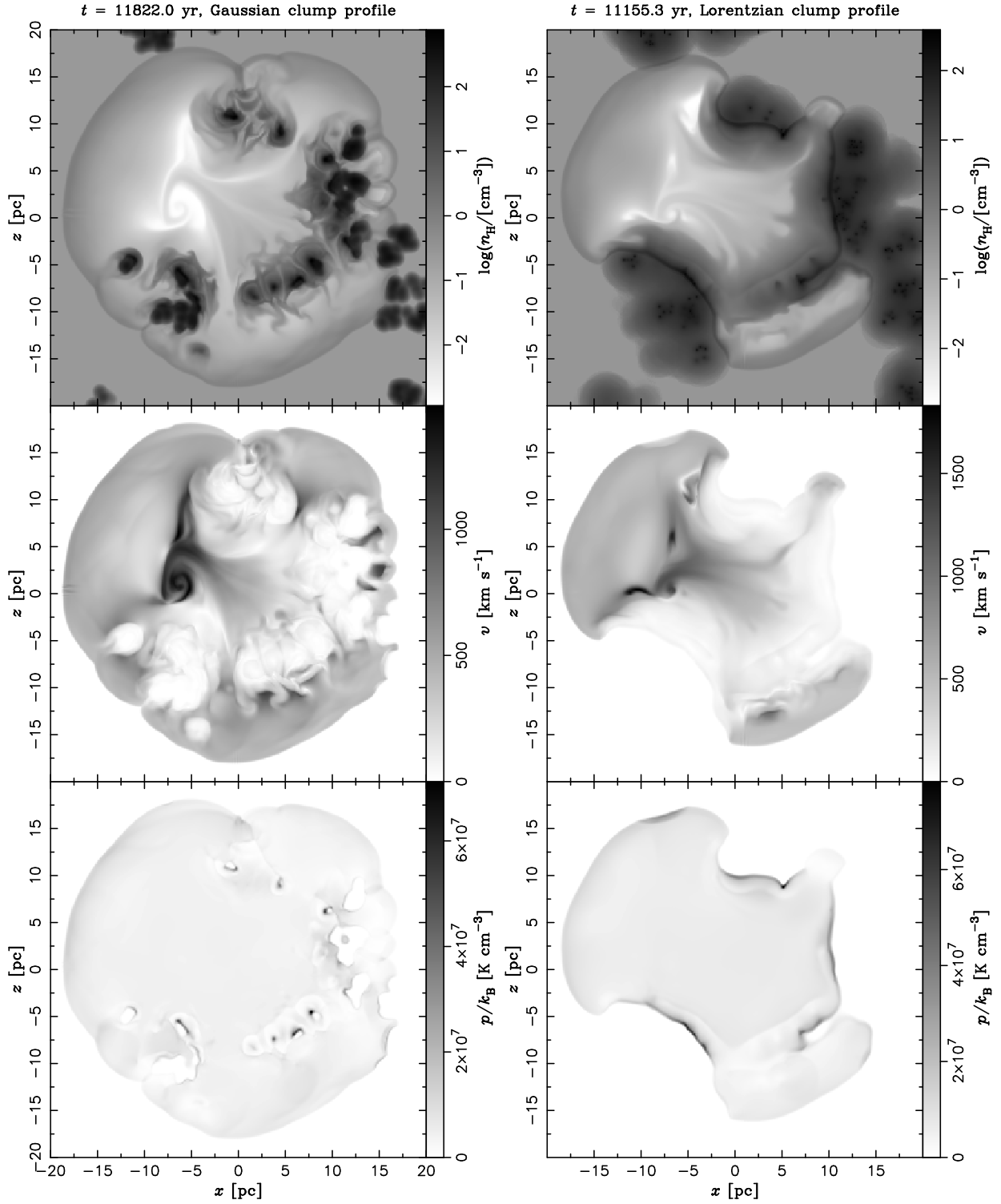


Fig. 9. Density, velocity and pressure fields for Gaussian (left) and Lorentzian (right) clump profile. The grayscale palette shows range of values between minimum and maximum cell value throughout the slice. The time t above the images is the simulation time from the beginning of SNR expansion.

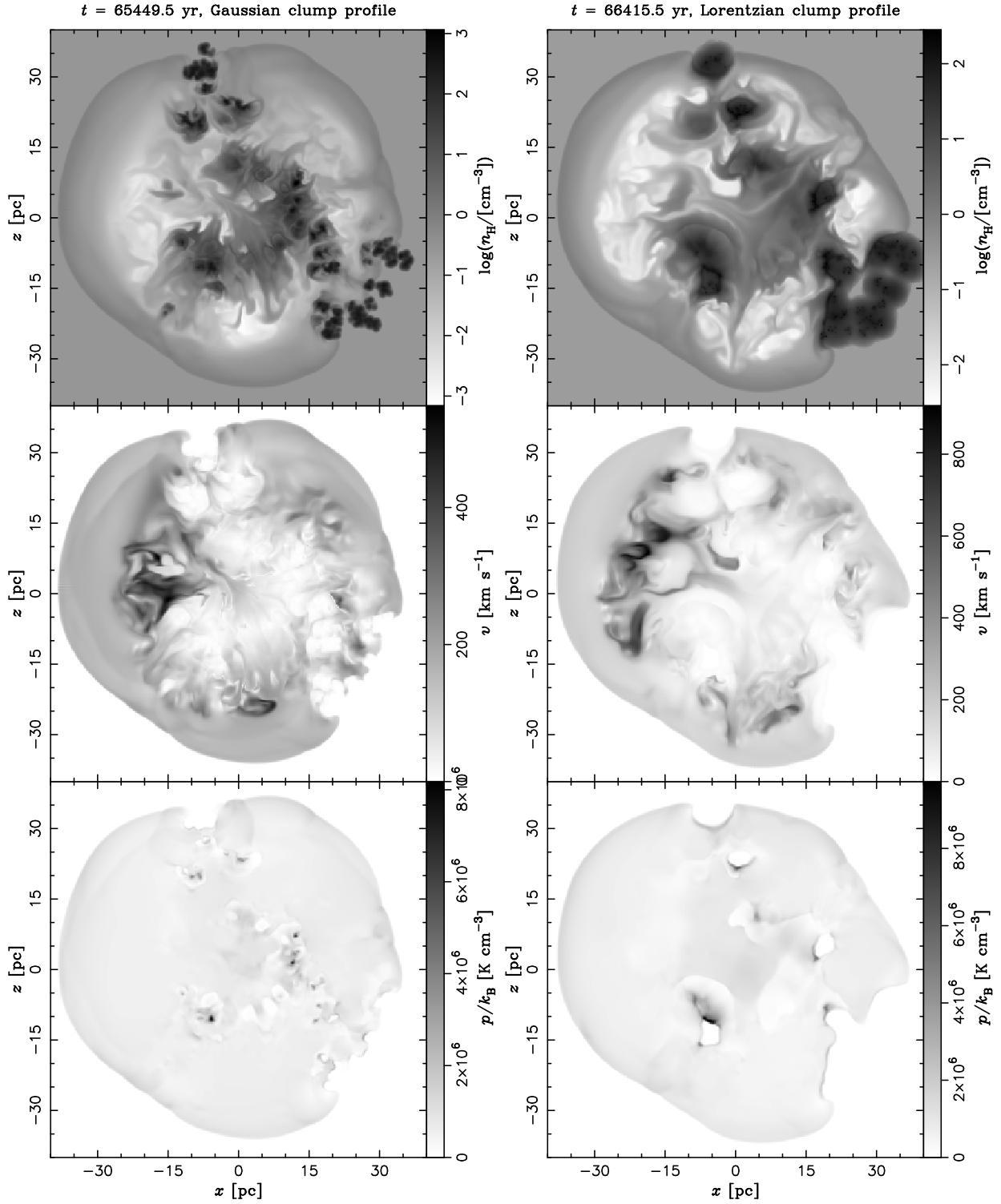


Fig. 10. Density, velocity and pressure fields for the Gaussian (left) and Lorentzian (right) clump profile. The grayscale palette shows range of values between the minimum and maximum cell value throughout the slice. The time t above the images is the simulation time from the beginning of the SNR expansion.

from the clumps which to a certain extent appear as shock reflecting walls. The bow shocks, reflected from the clumps, are clearly seen on the left slices of Fig. 8. These shock waves are coming back to the central region and interfere with each other producing non-symmetric features inside the remnant. From Figs. 8, 9 and 10 it is clear that even a relatively small increase in density (~ 1 order of magnitude at the edge of Lorentzian clumps ensemble) drastically decrease velocity of shock, so the wide Lorentzian profiles slow down the shock much further from the center of the clumps, while steep and narrow Gaussian profiles allow the SNR shock to find its way between the clumps. However, at the end (see Fig. 10), the SNR encircles the cloudlets and two cases become quite similar to each other. The emerging large scale shape of the SNR is weakly dependent on the selected clump profile as long as the clump size is small compared to the SNR. Notice that in both cases the total mass in clumps is the same.

The number of seed points in the box is N^H . Elmegreen (1997) chose $H = 4$, according to the density contrast in ISM, and $N = 12$, $L = 3$ because it results in $D_f = 2.3$, see Eq. (34). However, we set $N = 15$ for purely practical reasons. Due to the random number based algorithm for making the fractal, many clouds are not suitable because they are highly anisotropic relative to the center of the box (e.g. almost all points are placed in one half of the box). The solution to this is an isotropy condition that checks the spatial balance of the points at the first level of hierarchy (see Section 5.1), requiring these points to be evenly distributed over the box volume (to some acceptable extent). With 15 random points it is much easier to meet the isotropy condition than with 12. Finally, this does not make much difference except that in the case of lower N , there are fewer clumps (N^H) that are more massive.

For this first application, the SNR initial conditions were analytic Sedov profiles, but alternatively, we could initialize supernova ejecta similarly as in Test III (Section 4.3). It is known that the ejecta play a significant role in shaping the morphology of SNR because, even at later stages, it keeps memory on the early evolution of structures developed in the ejecta dominated phase. It is showed by simulations that the clumps of ejecta can overtake the forward shock and produce protrusions seen in some remnants (Wang and Chevalier 2001, Orlando et al. 2012). However, for the protrusions to occur, the density contrast in the ejecta must be ≥ 5 , or even ~ 100 as demonstrated by Wang and Chevalier (2001) for the case of Tycho SNR. The nickel (^{56}Ni) bubble scenario taking place in first ~ 15 days after the supernova event can be responsible for such clumping of the ejecta (for the 3D simulations of the early supernova evolution and development of nickel bubbles, see Wongwathanarat et al. 2015). Also, the interaction of the blast wave with the ISM clumps may have an impact on ejecta structures behind, via triggered bow shocks, and thus affect the final morphology of an SNR. Considering these notes, it might be that Sedov profiles used in our simulations are a somewhat weak approximation. However, our re-

sults show the behavior of the clumpy material in the downstream region of the analytic Sedov remnant, in terms of hydrodynamic flows. In the future use of this code for the purpose of comparison with SNR observations, the initial conditions should include the ejecta.

An important note must be given regarding the thermal conduction inside an SNR. McKee and Ostriker (1977) found that the evaporation of the dense clumps embedded in hot SNR interior plays significant role in shaping their *three-phase* model of ISM. The clump evaporation, which results from thermal conduction, is of great importance in the early adiabatic phase of the SNR evolution. It causes the density of hot gas to increase by the mass loss from clumps, as the thermal conduction effectively smooths out the temperature and density inhomogeneities. Slavin et al. (2017) included thermal conduction in their simulations and conclude that (a) the evaporation does not affect the Sedov expansion law $R \propto t^{2/5}$ and (b) the evaporation timescale increases proportionally to the remnant age. This means that the clumps far away from the SNR center will not be efficiently evaporated, as pointed out also by McKee and Ostriker (1977). The more detailed picture is given in Orlando et al. (2005) with inclusion of radiative losses apart from thermal conduction. In the cloud crushing experiment they showed that, in the test with a 50 Mach shock, the thermal conduction plays dominant role over the radiative losses, smoothing the temperature and density gradients which leads to cloud expansion and gradual evaporation without fragmenting into cloudlets. Instead, in the 30 Mach shock case, the radiative losses are dominant over the thermal conduction, and the cloud collapses under the effect of radiative cooling, and eventually fragments into dense and cold filaments. The thermal conduction exceeds the radiative losses only in the outer parts where the hot diluted corona develops, expands and evaporates. Our code does not include neither thermal conduction nor radiative losses. It can be seen by the long-lived clumps inside the remnant (e.g. on Fig. 9 and 10, Gaussian clumps) which diffuse to some extent, but certainly not as fast as they would with the thermal conduction included. However, the code is intended to be developed further, so these options are planned to be integrated at one point.

8. CONCLUSIONS

We have presented the 3D hydrodynamical code for simulating evolution of a supernova remnant in the ambient of a fractal molecular cloud. The features of the code and underlying simulation concepts are described in detail making a standalone manual for hydrodynamical simulations of SNRs. With the described remapping procedure of grid coarsening, the code can simulate the SNR evolution over a range of spatial scales of few orders of magnitude, even on a standard personal computers. The performed tests of spherical cloud crushing and clumpy ejecta simulation successfully produced the expected hydrody-

namical phenomena (such as the transmitted and reflected shock and Rayleigh-Taylor instabilities). The comparison of the numerically obtained Sedov profiles with the analytical ones also proved the validity of the code.

The standard contemporary AMR codes surely have higher performance, because they can achieve much higher resolution compared to this code. Besides that, they are optimized to run on several core processors, computing multiple parts of the simulation at the same time. However, the simplicity of the code presented in this work makes it very practical and easy to use while achieving a high fidelity of hydrodynamical features. The starting resolution should be selected carefully so it still satisfies the requirements after all remapping steps. Also, the selection of the initial box size should be such that the final box size sensibly compares to the size of SNR at the end of the simulation. The Figs. 8, 9 and 10 show that the expansion of SNR in inhomogeneous medium leads to asymmetric shapes with highly turbulent flows in its interior. Future work will extensively use the code presented in this paper to simulate large scale interactions of SNRs with the ISM.

Acknowledgements – The author acknowledges the financial support by the Ministry of Education, Science and Technological Development of the Republic of Serbia through the project #176005 “Emission nebulae: structure and evolution”. The author greatly appreciate the help of Marko Pavlović during the initial stages of this project. In particular, author wishes to thank his mentor Branislav Vukotić for the unselfish and continuous assistance in work. Also, author is grateful to the referee for the comments that improved the paper in many parts.

REFERENCES

- Arbutina, B., and Urošević, D.: 2005, *Mon. Not. R. Astron. Soc.*, **360**, 76.
- Berkhuijsen, E. M.: 1986, *Astron. Astrophys.*, **166**, 257.
- Bryan, G. L., Norman, M. L., O’Shea, B. W., Abel, T., Wise, J. H., Turk, M. J., Reynolds, D. R., Collins, D. C., Wang, P., Skillman, S. W., Smith, B., Harkness, R. P., Bordner, J., Kim, J., Kuhlen, M., Xu, H., Goldbaum, N., Hummels, C., Kritsuk, A. G., Tasker, E., Skory, S., Simpson, C. M., Hahn, O., Oishi, J. S., So, G. C., Zhao, F., Cen, R., Li, Y., and Enzo Collaboration: 2014, *Astrophys. J. Suppl. Ser.*, **211**, 19.
- Blitz, L., and Williams, J. P.: 1999, arXiv:astro-ph/9903382.
- Chevalier, R. A.: 1983, *Astrophys. J.*, **272**, 765.
- Dwarkadas, V. V., and Chevalier, R. A.: 1998, *Astrophys. J.*, **497**, 807.
- Elmegreen, B. G.: 1997, *Astrophys. J.*, **477**, 196.
- Elmegreen, B. G., and Falgarone, E.: 1996, *Astrophys. J.*, **471**, 816.
- Federrath, C., Klessen, R. S., and Schmidt, W.: 2009, *Astrophys. J.*, **692**, 364.
- Fryxell, B., Olson, K., Ricker, P., Timmes, F. X., Zingale, M., Lamb, D. Q., MacNeice, P., Rosner, R., Truran, J. W., and Tufo, H.: 2000, *Astrophys. J. Suppl. Ser.*, **131**, 273.
- Godunov, S. K.: 1959: *Matematicheskii Sbornik*, **89**, 271.
- Hole, K. T., Kasen, D., and Nordsieck, K. H.: 2010, *Astrophys. J.*, **720**, 1500.
- Kamm, J. R., and Timmes, F. X.: 2007, On Efficient Generation of Numerically Robust Sedov Solutions, <http://cococubed.asu.edu/papers/la-ur-07-2849.pdf>
- Kifonidis, K., Plewa, T., Scheck, L., Janka, H.-Th., and Müller, E.: 2006, *Astron. Astrophys.*, **453**, 661.
- Klein, R. I., McKee, C. F., and Colella, P.: 1994, *Astrophys. J.*, **420**, 213.
- Kostić, P., Vukotić, B., Urošević, D., Arbutina, B., and Prodanović, T.: 2016, *Mon. Not. R. Astron. Soc.*, **461**, 1421.
- Mac Low, M., McKee, C. F., Klein, R. I., Stone, J. M., and Norman, M. L.: 1994, *Astrophys. J.*, **433**, 757.
- McKee, C. F., and Ostriker, J. P.: 1977, *Astrophys. J.*, **218**, 148.
- Mignone, A., Bodo, G., Massaglia, S., Matsakos, T., Tesileanu, O., Zanni, C., and Ferrari, A.: 2007, *Astrophys. J. Suppl. Ser.*, **170**, 228.
- Mignone, A., Zanni, C., Tzeferacos, P., van Straalen, B., Colella, P., and Bodo, G.: 2012, *Astrophys. J. Suppl. Ser.*, **198**, 7.
- Nomoto, K., Thielemann, F.-K., and Yokoi, K.: 1984, *Astrophys. J.*, **286**, 644.
- Orlando, S., Bocchino, F., Miceli, M., Petruk, O., and Pumo, M. L.: 2012, *Astrophys. J.*, **749**, 156.
- Orlando, S., Miceli, M., Petruk, O., Ono, M., Nagataki, S., Aloy, M. A., Mimica, P., Lee, S.-H., Bocchino, F., Peres, G., and Guarrasi, M.: 2019, *Astron. Astrophys.*, **622**, A73.
- Orlando, S., Peres, G., Reale, F., Bocchino, F., Rosner, R., Plewa, T., and Siegel, A.: 2005, *Astron. Astrophys.*, **444**, 505.
- Ono, M., Nagataki, S., Ito, H., Lee, S.-H., Mao, J., Hashimoto, M., and Tolstov, A.: 2013, *Astrophys. J.*, **773**, 161.
- Parmentier, G.: 2011, *Mon. Not. R. Astron. Soc.*, **413**, 1899.
- Pavlović, M. Z.: 2017, *Mon. Not. R. Astron. Soc.*, **468**, 1616.
- Pavlović, M. Z., Urošević, D., Arbutina, B., Orlando, S., Maxted, N., and Filipović, M. D.: 2018, *Astrophys. J.*, **852**, 84.
- Raymond, J. C.: 1984, *Annu. Rev. Astron. Astrophys.*, **22**, 75.
- Sedov, L. I.: 1959, Similarity and Dimensional Methods in Mechanics, Academic Press, New York.
- Slavin, J. D., Smith, R. K., Foster, A., Winter, H. D., Raymond, J. C., Slane, P. O., and Yamaguchi, H.: 2017, *Astrophys. J.*, **846**, 77.
- Sod, G. A.: 1978, *Journal of Computational Physics*, **27**, 1.
- Stutzki, J., Bensch, F., Heithausen, A., Osenkopf, V., and Zielinsky, M.: 1998, *Astron. Astrophys.*, **336**, 697.
- Stutzki, J., and Guesten, R.: 1990, *Astrophys. J.*, **356**, 513.

- Tasker, E. J., Brunino, R., Mitchell, N. L., Michielsen, D., Hopton, S., Pearce, F. R., Bryan, G. L., and Theuns, T.: 2008, *Mon. Not. R. Astron. Soc.*, **390**, 1267.
- Teyssier, R.: 2002, *Astron. Astrophys.*, **385**, 337
- Toro, E. F.: 2009, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer-Verlag Berlin Heidelberg.
- Vaidya, B., Mignone, A., Bodo, G., and Massaglia, S.: 2015, *Astron. Astrophys.*, **580**, A110.
- Wang, C.-Y., and Chevalier, R. A.: 2001, *Astrophys. J.*, **549**, 1119.
- Williams, J. P., Blitz, L., and Stark, A. A.: 1995, *Astrophys. J.*, **451**, 252.
- Williams, J. P., de Geus, E. J., and Blitz, L.: 1994, *Astrophys. J.*, **428**, 693.
- Woltjer, L.: 1972, *Annu. Rev. Astron. Astrophys.*, **10**, 129.
- Wongwathanarat, A., Müller, E., and Janka, H.-Th.: 2015, *Astron. Astrophys.*, **577**, A48.
- Yu, H., Fang, J., Zhang, P. F., and Zhang, L.: 2015, *Astron. Astrophys.*, **579**, A35.
- Zhang, D., and Chevalier, R. A.: 2019, *Mon. Not. R. Astron. Soc.*, **482**, 1602.

НОВИ НУМЕРИЧКИ ПРОГРАМ ЗА 3D ХИДРОДИНАМИЧКЕ СИМУЛАЦИЈЕ ОСТАКА СУПЕРНОВИХ

P. Kostić

Astronomical Observatory, Volgina 7, 11060 Belgrade, Serbia

E-mail: *pkostic@aob.rs*

УДК 533.951 : 524.354

Претходно саопштење

Развијен је 3D хидродинамички код, написан у програмском језику C, ради проучавања ширења остатака супернових у околном медијуму. Заснован је на MUSCL-Hancock шеми са HLLC Римановим решавачем. Код иницијализује остатак супернове већ у Седовљевој фази и симулира хидродинамику његовог ширења. Симулација је оптимизирана за проучавање интеракције остатка са међузвезданом средином, на великој скали. После детаљног описа кода и три теста хидродинамике, представљени су резултати симулације остатка у униформној средини и фрактално структурираном пољу густине

МЗМ, као прва примена кода. Симулација ОСН у униформној средини упоређена је са Седовљевим законом ширења и Седовљевим самосличним решењима за профиле густине, брзине и притиска. Резултати показују да симулација добро репродукује хидродинамику ширења остатка супернове, као и да је врло практична због своје једноставности и брзине. Еволуција ОСН у фракталној међузвезданој средини показује да грудве ометају ударни талас и производе интерференцију рефлектованих таласа, резултирајући турбулентним кретањима и нехомогеностима унутар остатка.