NUMERICAL NEBULAE

NUMERICAL NEBULAE

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van de Rector Magnificus Dr. D.D. Breimer,
hoogleraar in de faculteit der Wiskunde en
Natuurwetenschappen en die der Geneeskunde,
volgens besluit van het College voor Promoties
te verdedigen op dinsdag 6 december 2005
te klokke 15.15 uur

door

Erik-Jan Rijkhorst

geboren te Assen in 1974

Promotiecommissie

Contents

Part II: Astrophysical Applications

CHAPTER 1

Numerical methods for the dynamical evolution of Planetary Nebulae

Numerical techniques for solving the Euler equations of hydrodynamics have been under continuous development for the past fifty years. These algorithms can accurately handle steep gradients and strong shocks, which are flow features frequently occurring in astrophysical objects. Radiative effects like ionization, heating, and cooling often play a decisive role in the outcome of the evolution of these objects as well. Therefore, numerical simulations of astrophysical flows should in general include both radiation and hydrodynamics. Moreover, if one wants to study realistic systems without imposing any symmetry, three-dimensional simulations are a requirement. Current supercomputer technology, with machines now consisting of thousands of processors interconnected by a fast network, makes such large-scale, three-dimensional, radiation hydrodynamical calculations feasible.

One of the many possible astrophysical applications of these techniques is the dynamical evolution of Planetary Nebulae (PNe). These objects represent the final phases in the life of low to intermediate mass stars. During these late stages of evolution, the star loses most of its mass in a series of interacting stellar winds, producing an extended shell of circumstellar material. The hot central core of the star ionizes this material, which radiates in the optical and becomes observable as a PN. The interplay between the successive outflows, in combination with the radiative processes taking place, determines the morphology of the PN.

Since it is thought that all stars of low to intermediate mass go through the PN phase, understanding their evolution is of great importance. In this thesis, numerical simulations are presented that address some of the outstanding issues in PN research. Results of a numerical model where a fast stellar wind collides with a warped disk may explain the observed multipolar morphologies of some (proto-)PNe. Calculations that follow the interaction of an ensemble of photoevaporating knots provide clues to the evolution of such small scale structures observed in evolved PNe like the Helix nebula.

In this introduction, aspects of the numerical techniques used to carry out the three-dimensional calculations are given, and some details of the evolutionary phases of PNe are discussed as well. It closes with a summary of the thesis.

1.1   Numerical method

To calculate the hydrodynamical evolution of astrophysical fluids numerically, a computational mesh is often employed. Such a mesh discretizes space into cells, which, for the case of regular structured meshes, lie along mesh lines that run parallel to the co-ordinate axes of the chosen geometry. In a finite volume method, the state of the gas in these cells represents the average state of the true distribution. The hydrodynamics solver calculates the fluxes through the interfaces between the cells, which are used to update the state. Using such a procedure ensures that quantities like mass, momentum, and energy are being conserved.

The hydrodynamics solver should represent shocks and contact discontinuities correctly, and several numerical methods that can do so exist. One of these is the method introduced by Godunov (1959), which considers the interaction between two adjacent computational cells as a Riemann problem, and is therefore able to capture discontinuities within a few cells. Second order extensions to Godunov's method (e.g. Van Leer 1979; Colella & Woodward 1984; Roe 1986; Eulderink & Mellema 1995) accurately represent smooth flow in regions away from the discontinuities as well.

Another way to attain high resolution is provided by the technique of Adaptive Mesh Refinement (AMR) (Berger & Colella 1989; Berger & LeVeque 1998). The principle behind this method is to increase the resolution of the computational domain in those regions where it is needed, while keeping all other parts at a lower resolution. Cells in a region of interest are flagged for refinement using some well chosen refinement criterion. These cells are clustered into logically rectangular patches from which a level of refinement with higher spatial resolution is constructed. By recursively repeating this process, refinement levels are added one by one, ensuring a proper nesting of the patches.

The technique of domain decomposition is used to parallelize the AMR method for distributed memory architectures. The computational domain is divided into sub-domains which are distributed over the available processors using a space-filling curve. This establishes a load-balanced decomposition of the computational domain that minimizes inter-processor communication (e.g. Deiterding 2003). A layer of so called 'ghost cells' is implemented around each patch to connect those parts of the computational domain that reside on different processor nodes. These ghost cells also turn each patch into a self-contained entity from the viewpoint of the hydrodynamics solver.

To include the effects of radiation into the calculations, the ionization state and temperature of the gas need to be determined (e.g. Frank & Mellema 1994a). If the optical depth from the source to each computational cell is known, the photoionization and heating rate can be calculated. The amount of energy lost through radiative cooling needs to be computed as well, for example by using a collisional equilibrium cooling curve. From this, the temperature and ionization fractions are found, resulting in an updated pressure for each cell, thereby coupling the radiative processes to the dynamics.

To calculate the optical depth, rays need to be traced from the source to each cell. For a parallelized AMR code this is non-trivial, since the computational domain is distributed over several processors. Moreover, it is sub-divided into a hierarchy of patches, requiring a radical rethink of the concept of ray tracing. This thesis presents the algorithmic details of such a method for ray tracing, which is implemented into the publicly available AMR

hydrodynamics package *Flash* (Fryxell et al. 2000).

1.2   The dynamical evolution of Planetary Nebulae

The evolution of a $1 - 8\,\mathrm{M_\odot}$ star consists of several phases. When it has finished burning hydrogen in its core, it will move away from the main sequence in the Hertzsprung-Russell (HR) diagram to become a red giant. Hydrogen burning proceeds in a shell around the contracting helium core, and in the HR diagram the star evolves along the Red Giant Branch.

After helium is ignited in the core and is converted into carbon, the core contracts again. The star is now in its Asymptotic Giant Branch (AGB) phase, during which it burns helium and hydrogen in shells around the degenerate carbon core. By this time the size of the star has increased dramatically, and the gravitational force at the surface of its convective envelope is so small that the outer layers can easily escape, providing ideal circumstances for the formation of a dense stellar wind.

Mass loss during the AGB phase is thought to be driven by radiation pressure on dust grains that form in the cool outer layers of the star. Pulsations in these outer layers increase the mechanical energy of the material and create conditions that accelerate the growth of dust grains. Collisions between the dust grains and the gas couple the two, so that the gas is dragged along by the dust, leading to mass loss rates of $\sim 10^{-5}\,\mathrm{M_\odot\,yr^{-1}}$, and terminal wind velocities of $\sim 10\,\mathrm{km\,s^{-1}}$. The AGB phase ends when the amount of material in the envelope has been reduced to approximately $10^{-3} - 10^{-2}\,\mathrm{M_\odot}$.

The star now enters the PN phase which lasts for about $10^3 - 10^4$ yr. The tenuous envelope contracts and the effective stellar temperature increases to $\sim 10^4$ K at constant luminosity. The stellar UV photons generate a line driven wind with a relatively low mass loss rate of $\sim 10^{-8}\,\mathrm{M_\odot\,yr^{-1}}$, but with velocities that can reach $\sim 1000\,\mathrm{km\,s^{-1}}$ and higher. When the mass in the envelope drops below $\sim 10^{-5}\,\mathrm{M_\odot}$, nuclear fusion stops, and the star cools down to become a white dwarf with a mass of $\sim 0.4 - 1.0\,\mathrm{M_\odot}$.

Extensive descriptions of stellar evolution and mass loss during the AGB phase and beyond can be found in reviews by Iben & Renzini (1983), Habing (1996), and Willson (2000), or in the textbook by Lamers & Cassinelli (1999).

The evolutionary picture sketched above naturally leads to the scenario of interacting stellar winds (ISW) (Kwok et al. 1978). In the PN phase, the fast, tenuous stellar wind runs into the dense, slow material left behind during the AGB. This leads to the formation of an outer, as well as an inner shock, separated by a contact discontinuity. The swept up gas can cool radiatively resulting in a thin, dense, outwards moving shell. During the early phases of the interaction, when the wind velocity is still relatively low, the wind blown bubble is momentum driven. Once the fast wind reaches higher velocities, a transition to energy driven flow occurs, and a hot bubble develops. High energy photons from the hot central star (CS) ionize and heat the environment, which cools by recombining and emitting emission line photons. This influences the dynamical evolution of the nebula and should be taken into account. See Kahn & Breitschwerdt (1990) and Breitschwerdt & Kahn (1990) for an analytical assessment of the early phases of the evolution of PNe.

Although observed properties of spherical PNe can be explained by the ISW scenario,
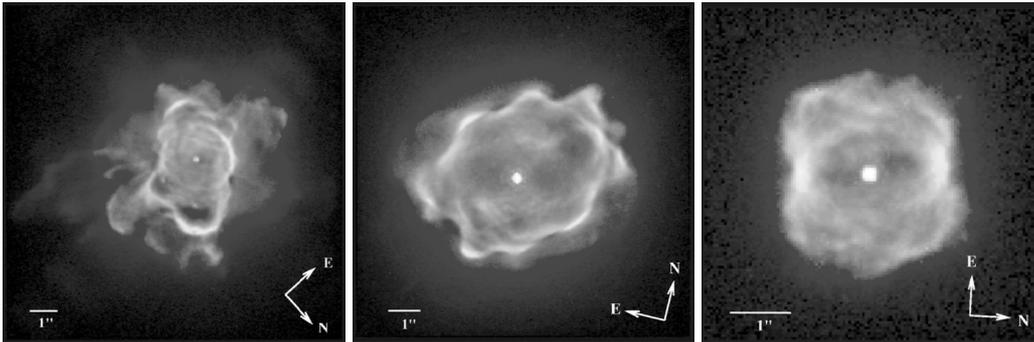
Figure 1.1: HST observations of proto-PNe taken from Sahai & Trauger (1998). Shown are (left to right): M1-26, He2-138, and M4-18.

many PNe have an elliptical or bipolar shape. This has led to the generalized interacting stellar winds (GISW) scenario, which assumes an aspherical density distribution for the slow AGB wind, with an increase in density from pole to equator (Kahn & West 1985; Balick 1987). When the fast wind runs into such a toroidal distribution, a bipolar shape emerges. Analytical (Icke 1988) and numerical work (Soker & Livio 1989; Icke 1991; Mellema et al. 1991; Frank & Mellema 1994a,b; Mellema 1995, 1997) has shown this to be a very effective mechanism, which may even lead to the formation of jets (Icke et al. 1992; Mellema & Frank 1997).

High resolution Hubble Space Telescope (HST) observations of (proto-)PNe have revealed morphologies that are difficult to explain in the context of the GISW scenario. Especially the often occurring multipolar, point-symmetric morphology, first mentioned by Schwarz et al. (1992), presents a great challenge to the GISW scenario, and some researchers even speak of 'a paradigm lost' (Frank 2000). Figure 1.1 shows some examples of observed multipolar (proto-)PNe. The nebulae presented in these images clearly show that some additional mechanism is needed to explain their shapes.

One solution that may resolve this issue is to invoke magnetic fields. Three-dimensional numerical calculations where the magnetic collimation axis is misaligned with the fast, bipolar wind result in point-symmetric morphologies (García-Segura 1997; García-Segura & López 2000). Alternatively, the occurrence of bipolar, rotating, episodic jets (BRETS) (Livio & Pringle 1996, 1997; Cliffe et al. 1995) may explain some observations. See Balick & Frank (2002) for a review of the various shaping mechanisms for (proto-)PNe.

Another mechanism that produces point-symmetric nebulae was proposed by Icke (2003), who presented two-dimensional calculations of the interaction of a fast, tenuous stellar wind with a *warped* circumstellar disk. Such warping can occur through a radiation driven instability, induced by the CS, which originates in differences in radiation pressure on slightly tilted annuli at different radii in the disk (Iping & Petterson 1990; Pringle 1996; Maloney et al. 1996, 1998). When the fast wind hits the inner rim of such a disk, a bow shock develops which, if radiative cooling is taken into account, leads to the formation of a multipolar bubble. Although the two-dimensional calculations show the underlying principles of the interaction, three-dimensional hydrodynamical simulations are required to fully capture the point-symmetry of the problem. Such simulations are presented in
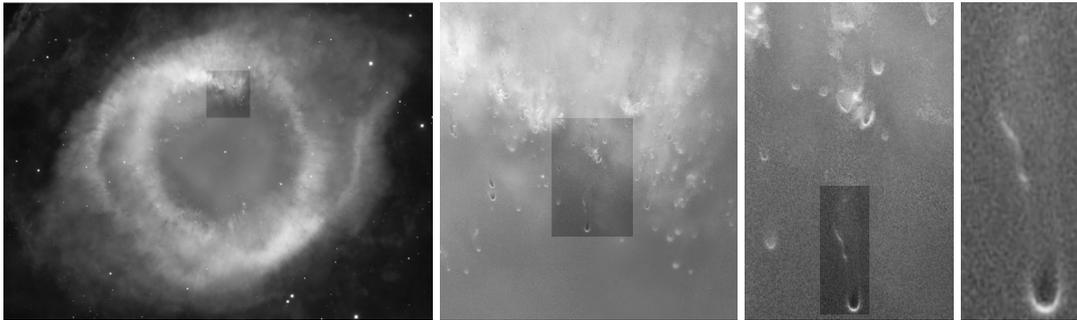
Figure 1.2: Zooming in on knots in the Helix. The knot on the right is designated 378-801 in the coordinate system of O'Dell & Burkert (1997).

this thesis.

Apart from their intriguing overall morphologies, PNe also show a large array of small scale structures. These range from fast low-ionization emission regions (FLIERS) (Balick et al. 1998), to high density neutral knots (O'Dell et al. 2002). Being the nearest PN, such knots are best observed in the Helix nebula (NGC 7293). Ground based imaging revealed that the Helix contains numerous high density knots (Meaburn et al. 1992, 1998), and observations with the HST provided a fully resolved view (O'Dell & Handron 1996; O'Dell et al. 2004, 2005). A sequence of successive zooms of knots in the Helix is shown in Figure 1.2.

The total number of knots is currently estimated to lie around $10,000$ (Meixner et al. 2005), which means they contain a significant fraction of the total nebular mass. Since similar knots have been observed in other nearby PNe as well (O'Dell et al. 2002), it has been suggested that they occur at some stage in *all* PNe. This would make them an important ingredient of the material that is fed back into the interstellar medium, and insight into the processes that create and shape the knots is required.

From Figure 1.2 one can see that the knots have limb brightened cusps of material pointing towards, and extended tails pointing away from the CS. This implies that the ionizing radiation from the CS is currently photoevaporating the knots. Analytical and numerical investigations into the hydrodynamical evolution of a single photoevaporating knot have been carried out (e.g Bertoldi 1989; Lefloch & Lazareff 1994; Mellema et al. 1998; O'Dell et al. 2005). However, since almost all knots in the Helix occur in groups, interaction among them is likely to take place, making it desirable to study the evolution of an ensemble of photoevaporating knots. Results of three-dimensional calculations involving the interaction between two knots were discussed by Lim & Mellema (2003). In this thesis, the extension to simulations that follow the evolution of a complete ensemble of photoevaporating knots is presented.

1.3   Thesis summary

This thesis consists of two parts. Part I: Numerical Methods (Chapters 2, 3, and 4) provides a description of the numerical techniques used to carry out three-dimensional radia-

tion hydrodynamical simulations. Part II: Astrophysical Applications (Chapters 5 and 6) presents two such simulations in the context of Planetary Nebulae.

In Chapter 2, algorithmic aspects of the numerical method used to carry out the simulations are presented. The method of AMR is discussed, together with its parallelization through the technique of domain decomposition. Details of the AMR package *Paramesh*, as implemented in the publicly available hydrodynamics code *Flash*, are given. *Flash* is used for all numerical AMR calculations presented in this thesis. We continue with a terse description of the PPM hydrodynamics solver and finish with the implementation details of the *Doric* code, used to calculate the contribution of radiative processes like ionization, heating, and cooling to the dynamical evolution of the gas.

In order to calculate this contribution, a value for the optical depth from the source up to each cell in the computational domain needs to be obtained. Chapter 3 presents a novel method for ray tracing, called *Hybrid Characteristics*, that can do so for a parallelized three-dimensional AMR hierarchy of patches. In this approach a ray is split into independent sections. Each processor calculates the local contribution to the optical depth for those ray sections that run through its part of the computational domain. These contributions are made available to all processors in a global communication operation, after which they are interpolated and accumulated to arrive at the desired total optical depth for each cell.

The interpolation routines are designed to give the exact value for the optical depth for the case of a homogeneous density distribution. Results of a test with a $1/r^2$ density distribution show that the method calculates the optical depth with errors $< 0.5\%$ in this case. A shadow casting test shows that the Hybrid Characteristics method is capable of producing shadows with very sharp boundaries, comparable to those obtained with a long characteristics method.

This chapter also discusses first results of an application of the method to the photo-evaporation of high density clumps. In these simulations, due to the effectiveness of the AMR, the computational domain can contain the source as well as the clumps. It is found that the interaction of the photoevaporation flows leads to a region of high density, having implications for objects like cometary knots in PNe and binary protoplanetary disks (proplyds) in HII regions.

Extensive weak and strong scaling tests of the Hybrid Characteristics method are presented in Chapter 4. The performance of the method as a function of source location, amount and location of refinement, number of sources, and machine architecture is investigated. The calculation part of the algorithm is found to scale almost perfectly, for weak as well as strong scaling. Due to the imbalance in the distribution of ray sections over the processors inherent in the approach to ray tracing taken by the Hybrid Characteristics method, almost half of the calculation time is spent in synchronization. The performance of the method is found to be independent of the (relative) location of the source and the refined region, as well as of the amount of refinement. The calculation time scales linearly with the number of sources, whereas the synchronization time averages out to a low value. Upon comparison of the calculation with the communication time, it is found that the method can be used efficiently for at least ~ 100 processors on the SGI Altix system, and ~ 1000 processors on the IBM BlueGene/L system. This concludes Part I of the

thesis.

In Part II, astrophysical applications of the developed numerical method in the context of the dynamical evolution of PNe are presented. Chapter 5 considers three-dimensional simulations of the formation of multipolar (proto-)PNe. In the spirit of the GISW scenario, the evolution of a fast, tenuous stellar outflow interacting with a warped circumstellar disk is calculated. The mechanism for inducing such a warped disk is briefly discussed, together with an equation for its precession time scale. This time scale is compared to the dynamical and cooling time scales of the wind blown bubble. Setting the physical parameters for the CS, wind, and disk to values that are appropriate for the early stages of PNe evolution, results in a parameter space applicable to this kind of objects.

Two-dimensional simulations involving two different approximations for the radiative cooling are presented. One involves the use of a collisional equilibrium cooling curve, whereas the other simply sets the adiabatic index γ to a low value. Both these approaches lead to similar point-symmetric morphologies, and for the three-dimensional calculations the computationally less expensive technique of using a low value for γ is applied.

Synthetic Hα images and position-velocity diagrams derived from the resulting three-dimensional data cube are presented. Upon comparison with observed (proto-)PNe, several good candidates that can be accommodated by this model are found.

In Chapter 6, the photoevaporation of multiple dense knots with physical parameters applicable to knots in the Helix nebula are studied. The high energy photons that emanate from the CS of this evolved PN induce an ionization front at the knots, resulting in their cusp-like appearance. A summary of observed properties of knots in the Helix is presented, and the assumptions and approximations that went into the numerical simulations are carefully motivated. The calculations use an approximation for the radiative cooling, and ignore stellar evolution, the presence of a stellar wind, and the diffuse radiation field.

Previous numerical studies of photoevaporating knots found a dynamical evolution consisting of two phases. First the collapse phase, where most of the knot's material is photoevaporated by the ionizing radiation. During this phase, the ionization front induces a shock, which gets focused towards the symmetry axis of the knot, resulting in a small clump of shocked knot material. The collapse phase is followed by the cometary phase, where the remains of the shocked material are accelerated through the Oort-Spitzer 'rocket effect'. These phases are in concordance with analytical results obtained by Bertoldi (1989) and Bertoldi & McKee (1990).

However, for the physical parameters appropriate for the knots in the Helix, the simulations show a rather different sequence of events. First the knot is compressed by the shock, and since, for these parameters, the shock velocity is higher than the velocity of the ionization front, a thick layer of shocked gas develops between the two fronts. This makes the shock focusing also less severe, so that the part of the shock close to the axis can outrun the part at the edge of the knot, leading to a 'ring' of increased density. Once the shock has left the back of the knot, an expansion wave is generated which greatly reduces the density inside the knot. When this wave hits the ionization front, a new shock is induced, accelerating the knot for a second time.

The condition for which a thick layer of gas between the shock and the ionization front can develop, so that an expansion of the knot instead of a collapse can take place, is

derived analytically. It is shown that this is the *same* condition found by Bertoldi (1989) to discern between two regimes in his parameter space of photoevaporating clouds. This means that the expansion phase found in our simulations is in fact quite common for this type of process, making this an important effect.

Simulations of a single photoevaporating knot are compared for two different methods: the Hybrid Characteristics, and the C^2-Ray method, which is a newly developed radiation hydrodynamics code that is explicitly photon conserving (Mellema et al. 2005). It is found that the conventional approach for incorporating radiative processes, as is implemented in for example the Hybrid Characteristics method, can lead to deviations in the outcome of values for global properties like the evaporation time of the knots. Results obtained with such non-conserving methods should therefore be interpreted with care.

A series of three-dimensional simulations of the different interactions that can take place among two photoevaporating knots is presented. A mechanism for the efficient dispersion of knot material into the tail region of another knot is found. An overview is given of the evolution of global flow properties like the neutral mass, kinetic energy, and internal energy of the knots. Synthetic Hα images are used to compare the simulation results to observations of knots in the Helix, explaining some of the observed features like wispy emission between the knots, and the existence of dust and molecules like CO and H$_2$ in the tail regions of the knots.

The chapter finishes with a calculation involving the photoevaporation of a whole ensemble of knots, proving the effectiveness of the Hybrid Characteristics method for carrying out large scale, three-dimensional, radiation hydrodynamical simulations.

Part I
Numerical Method

CHAPTER 2

3D radiation hydrodynamics

Numerical methods for high resolution multi-dimensional astrophysical fluid flow simulations are discussed. We give an outline of the technique of adaptive mesh refinement (AMR) and its parallelization, and present details of a particular AMR implementation used in the *Flash* code. We briefly recapitulate the piecewise-parabolic method for hydrodynamics, and describe the radiative processes we implemented using the *Doric* package. A list of currently available public AMR hydrodynamics codes is also given.

2.1 Introduction

The need for high resolution multi-dimensional astrophysical fluid flow simulations has led to the development of a variety of numerical techniques. Especially for the case of three-dimensional calculations, the requirements on memory and CPU speed are demanding, and often beyond current single processor machines. Therefore, in order to perform this type of calculations, one needs to resort to multi-processor supercomputers and techniques like domain decomposition and adaptive mesh refinement (AMR). Modern shock-capturing solvers for the hydrodynamics should be applied as well.

In Section 2.2 we discuss the specifics of a particularly efficient approach to AMR. We also describe some of the details of the AMR method implemented in the *Flash* code, which was used for all the simulations presented in this thesis. Section 2.3 gives a description of the piecewise-parabolic method, a finite volume hydrodynamics solver implemented in *Flash*. The evolution of Planetary Nebulae, which are the astrophysical applications of this thesis, also require a proper handling of radiative processes like ionization, heating, and cooling, which is described in Section 2.4. In Section 2.5 we outline some of the practical details on how extensions to the *Flash* code, like the radiative processes just mentioned, can be made. We finish this chapter with a list of publicly available AMR hydrodynamics codes in Section 2.6.

2.2 Adaptive Mesh Refinement

Numerical methods for fluid flow have in common that they somehow need to discretize space. To this end a computational mesh consisting of cells is often used. Such a mesh is defined by a collection of nodes and the connectivity between them. A mesh can either be structured or unstructured.

For unstructured meshes, the connectivity needs to be specified for each node. An example of an unstructured mesh is the Delaunay triangulation, where a cell is defined as the simplex (triangles, tetrahedra) of the dimensionality of the mesh. Although unstructured meshes have been successfully applied in fields like aerodynamics (e.g. Löhner 1987; Luo et al. 2000), they have not been very popular for calculating astrophysical fluid flows, notable exceptions being Whitehurst (1995) and Xu (1997). Instead, for astrophysical applications, the mesh-less, particle-based method of smooth particle hydrodynamics (SPH) has become widely used (e.g. Monaghan 1992; Springel et al. 2001).

For structured meshes, the connectivity is implicitly given by the order of the nodes. The geometry of the mesh categorizes the different types of structured mesh: a conformal mesh has an arbitrary geometry, whereas a regular mesh has mesh lines that run parallel to the coordinate axes of the chosen geometry (e.g. Cartesian, polar, spherical, cylindrical). When all cells in a regular structured mesh have the same shape, it is called isotropic. For their simplicity and ease of implementation, it is these isotropic regular structured meshes that are used most often in mesh-based methods for astrophysical fluid flows, and it is this type of mesh that we have used throughout this thesis.

2.2.1   High resolution meshes

Numerical methods for astrophysical fluid flow using structured meshes are limited in the resolution they can obtain. Depending on the application, limits are mainly set by memory capacity and CPU speed of the machine used to do the calculation. In order to improve on this, a number of different techniques has been developed. The principle behind these techniques is to increase the resolution only in those regions of the computational domain where it is needed, while keeping all other parts of the domain at a lower resolution. In astrophysical applications, the regions that should be resolved at high resolution are typically those regions where shocks and contact discontinuities occur in the flow. This implies that the method needs a mechanism to determine where these flow features are located in the computational domain. Below we will describe two approaches that are capable of attaining local high resolution on a computational mesh.

Moving meshes

One of the techniques that implements the idea of locally increasing the resolution is the so called 'moving mesh method' (e.g. Gnedin 1995; Pen 1998). For this method, cells are allowed to change shape and position in order to increase the resolution in certain parts of the domain. The change in shape of the cells is controlled by a condition that tries to keep the resolution in mass constant.

Since the total number of cells available is fixed, these methods are also known as 'penalty methods' (Quirk 1991), because an increase in resolution in one part of the domain must lead to a decrease in resolution in another part. The fact that cells can have arbitrary shapes complicates the method; limits on cell distortion have to be set to prevent mesh entanglement and preserve accuracy, and the flow solver needs to be more complex than ones that can be used with an isotropic mesh.

Adaptive refinement for regular structured meshes

Another approach to attaining high resolution in numerical fluid flow simulations is to adaptively refine certain parts of the computational domain. This means that cells in a region of interest are first flagged for refinement using certain criteria (for details, see Section 2.2.2). After this, cells of smaller size are added to the flagged regions of the mesh by creating a new level of refinement. By recursively repeating this process, levels are added one by one, in principle leading to an arbitrary high resolution. Since, during a simulation, the regions that need to be refined are in general moving through the mesh, the refinement should change accordingly, i.e. it should adapt to changes in the flow.

In order to store the newly created cells, two different techniques can be used. One approach, known as 'cell-based AMR', is to store each individual cell in a tree-like data structure (see for example Falle & Raga (1993); Raga et al. (2000b); Lim & Mellema (2003)).

An alternative and more sophisticated route to adaptive refinement is to first collect newly created mesh cells into a number of patches, and to store these patches in a data tree. This technique, referred to as 'block-structured AMR', or simply 'AMR', was in-
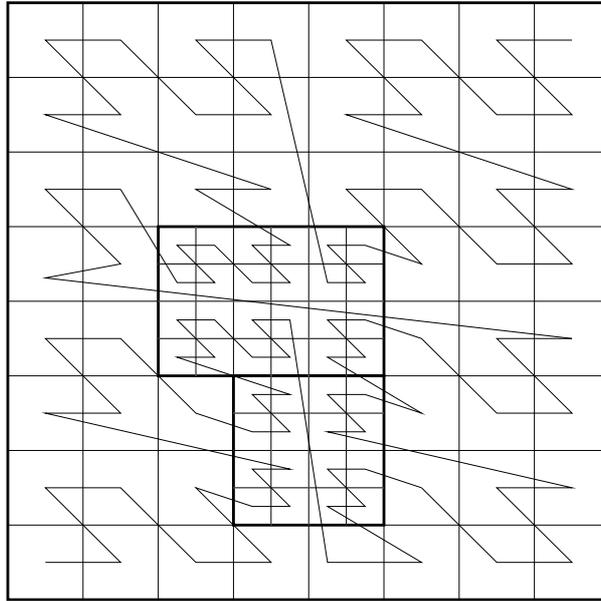
Figure 2.1: Example of a Morton (or Z) space filling curve for block-structured AMR. Three patches and two levels of refinement are shown. Using this curve and appropriate workload calculations for each patch, a load-balanced distribution of patches among processor nodes can be achieved.

troduced by Berger & Oliger (1984). Since its conception, numerous implementations of this method have followed, including versions that are parallelized for distributed memory machines (e.g. Berger & Colella 1989; Quirk 1991; Neeman 1996; Berger & LeVeque 1998; MacNeice et al. 2000; Plewa & Müller 2001; Keppens et al. 2002; Deiterding 2003; O'Shea et al. 2004).

In Section 2.2.2 we give a more detailed description of the different steps that make up a block-structured AMR algorithm, but first we briefly discuss the method of parallelization.

Domain decomposition for AMR

In order to parallelize an AMR algorithm for distributed memory machines, a technique known as 'domain decomposition' is used. Like its name suggests, the computational domain is divided into sub-domains which are distributed over the available processing nodes. To decide which part of the domain should be on which node, a so called space-filling curve is used. Such a curve maps the distribution of cells onto a one-dimensional array (see Figure 2.1). After identifying which part of the curve represents which patches, and calculating a measure for the amount of work to be done for each patch, a load-balanced decomposition of the computational domain that minimizes inter-processor communications can be established. See for example Deiterding (2003) for an implementation of such a method.

2.2.2 Block-Structured AMR

A number of Ph.D. theses has been written on the subject of AMR (Berger 1982; Quirk 1991; Neeman 1996; Deiterding 2003) and ample documentation is available in the literature. Therefore, in this section, we summarize the different aspects of AMR, and refer the interested reader for more details to these works.

The algorithm for block-structured AMR consists of a number of steps. First, cells in the computational domain that fulfill certain criteria are flagged for refinement. These cells are then clustered into logically rectangular patches using a pattern recognition algorithm. From these patches, a new level of refinement is created which contains cells that are smaller than the underlying coarser cells by a refinement factor r. The state vector of these newly created cells is filled through interpolation from the coarse level in a procedure called prolongation. By recursively adding refinement levels, a proper nesting of patches contained in different levels of refinement is ensured.

Around each patch, a layer of so called 'ghost' cells (also referred to as 'guard' or 'dummy' cells) is allocated. These cells are used to make each patch a self-contained entity that, from the viewpoint of the fluid flow solver, can be integrated as if it was a single mesh. Before the actual integration takes place, the state vectors of the ghost cells need to be filled, either by directly copying them from a neighboring patch, or through prolongation from cells at the coarser level below. When the ghost cells of a patch are located at a physical boundary of the computational domain, an appropriate boundary condition needs to be set. For a parallelized AMR algorithm, ghost cells are also used to connect parts of the decomposed computational domain that are residing on different processor nodes.

Besides refinement in space, the AMR method also facilitates refinement in time. Since cells at a coarser level are larger than those at more refined levels, the coarser level can be integrated with a larger time step. When the algorithm cycles through the refinement levels, it applies more and smaller time steps at finer levels, resulting in comparable Courant numbers for each level.

In order to ensure conservation, fluxes at boundaries of coarser cells that are abutting cells from a finer level need to be replaced in a flux-correction step. Furthermore, after a finer level has been updated, its state vectors are projected onto the underlying coarser cells in a procedure called restriction.

The fact that structured meshes with different geometry are topologically identical allows refinement of all of these mesh types through the block-structured AMR approach. Only those steps in the algorithm that involve the geometry of the mesh, like prolongation, restriction, and flux correction, need a different implementation for each geometry.

Figure 2.2 schematically shows the sequence of steps taken by the AMR algorithm. In the following Sections, more details on these individual steps are presented.

Flagging for refinement

Before actual refinement takes place, cells need to be flagged. To this end, appropriate refinement criteria need to be applied. Depending on the application, these criteria often need to be tuned so as to produce the required amount of refinement in the desired loca-
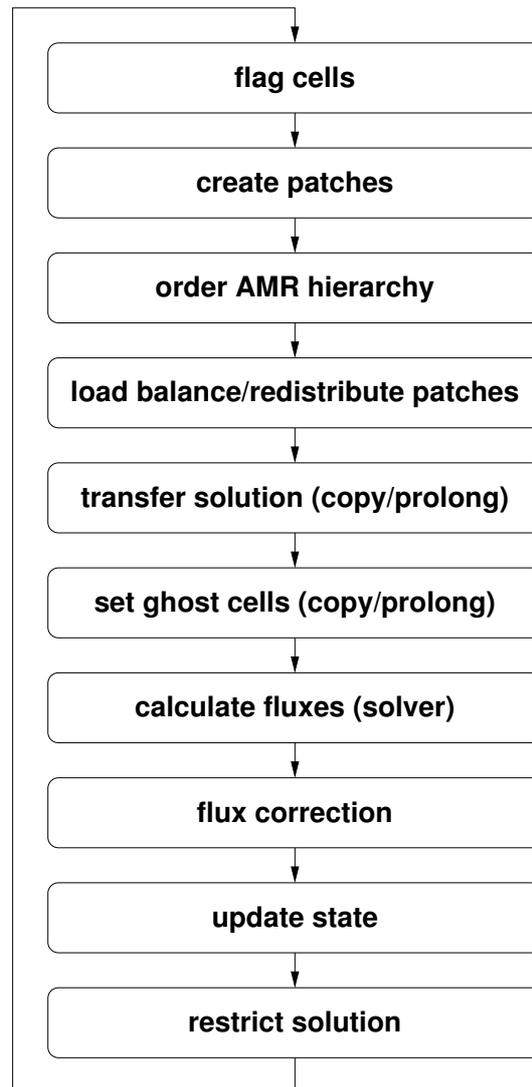
Figure 2.2: Individual steps taken by the AMR algorithm.

tions of the computational domain. The following refinement criteria are commonly used: refinement on threshold, derivative, truncation error, or a combination of these.

Refinement on threshold is used when it is desirable that regions where the state vector has a particular value stay refined throughout the simulation. Refinement using the (density) gradient works well for locating discontinuities in the flow. Also, the second derivative can be used as a refinement criterion to track these discontinuities (see Section 2.2.3).

Berger & Oliger (1984) promoted a refinement criterion that uses an estimate for the local truncation error derived using Richardson extrapolation. This criterion is defined by comparing the solution for a certain refinement level when integrated for two time steps Δt, with the one found when integrating a two times coarser mesh for a time step $2\Delta t$. Although this approach is strictly valid only in regions of smooth flow, it also gives large values for the error estimate in regions where discontinuities occur, and will therefore
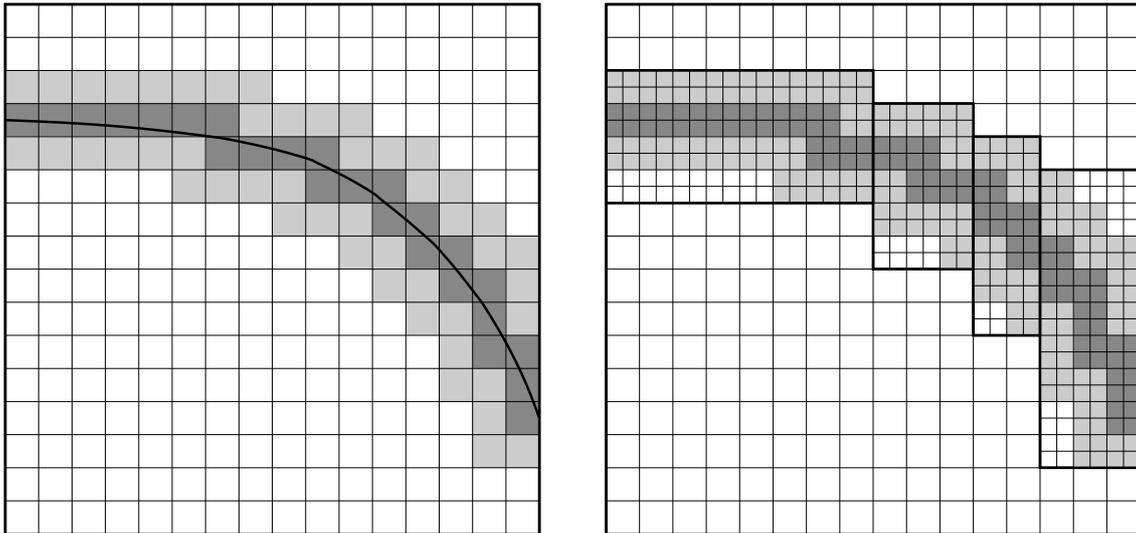
Figure 2.3: On the left, an example of cells flagged for refinement (dark grey) at a discontinuity in the flow and a corresponding single layer of buffer cells (light grey) are indicated. On the right, a possible outcome of the refinement procedure is shown.

refine these flow features as well (Berger & Oliger 1984).

In most implementations of AMR, after cells have been flagged using the refinement criteria, one or more extra layers of so called 'buffer' cells around the already flagged cells are also flagged (cf. Figure 2.3). This ensures that discontinuities in the flow are never derefined during a simulation, which is one of the secrets to the success of AMR (Quirk 1991).

Creating new patches

When cells from a certain level have been flagged for refinement, they need to be clustered into logically rectangular patches (Figure 2.3). One way to create these patches is to use a pattern recognition method (Berger & Rigoutsos 1992; Bell et al. 1994). This method calculates so called 'signature lists' along each coordinate direction. At places where the second derivative of these lists changes sign, boundaries between patches are located.

Another method is to start with the minimal bounding box of the flagged cells and to recursively bisect it, where subdivision takes place along the longest patch side (Quirk 1991). Subdivision is halted when the ratio of the number of flagged cells to the number of total cells in a patch is larger than some chosen level of tolerance, or when a minimum allowed patch size is reached.

New patches need to be merged with the existing AMR hierarchy and distributed among the available processing nodes. Both methods just described result in non-overlapping patches, which is desirable since this simplifies the transfer of solution procedure, described in the next section.
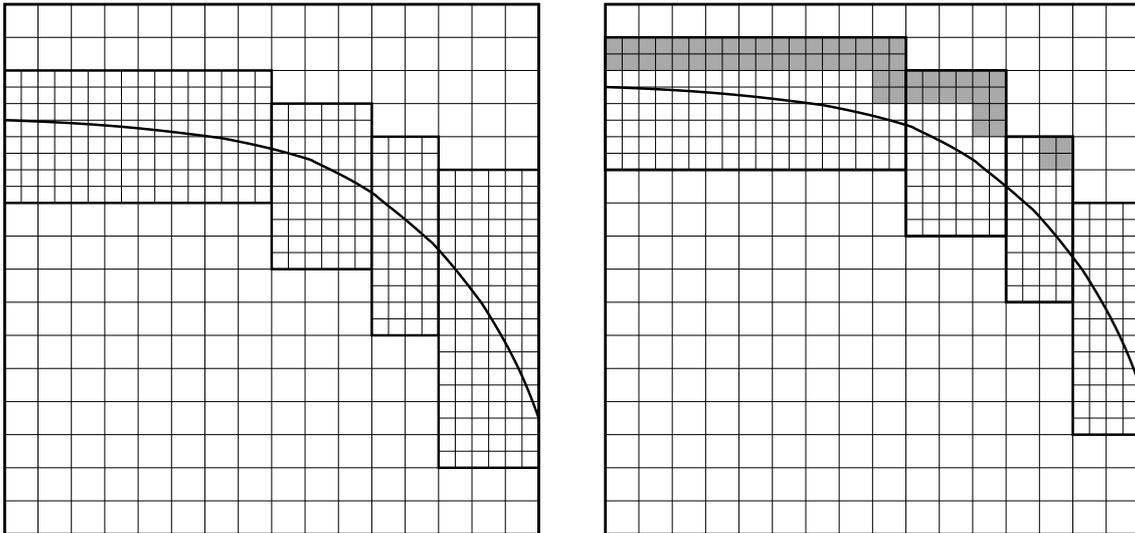
Figure 2.4: Patches created during two consecutive iterations. The state vector of cells in overlapping regions can be copied directly, whereas values for cells that lie outside this region of overlap (shown in grey) need to be obtained through prolongation from cells at the coarser level below.

Transfer of solution

When new patches have been created, their state vectors are set in a procedure called 'transfer of solution'. This procedure should be conservative and monotonicity preserving.

In general, the newly created patches overlap with the patches from the previous iteration, so values of state vectors for cells located in these regions of overlap can be copied directly (see Figure 2.4). Values for state vectors of cells that lie outside the region of overlap need to be obtained from the underlying coarser grid through the process of prolongation.

Updating ghost cells

Each patch has one or more layers of ghost cells allocated around it. The number of layers depends on the size of the stencil required by the fluid flow solver (see Section 2.3). This layer is implemented for a number of reasons: it facilitates a straightforward implementation of the solver, it provides a means to set external boundary conditions, and it connects sub-domains that are located on different processor nodes.

After refinement has completed, and before the fluid flow solver is called, the state vectors of the ghost cells of all patches in existence in the AMR hierarchy should be set to their appropriate values. Depending on the location of a patch, these values stem from different sources (see Figure 2.5). Values are copied directly from those neighboring patches that are on the same refinement level as the patch under consideration, and are prolonged from those patches that are on a coarser level of refinement.

When a ghost cell lies at a boundary of the domain, the appropriate conditions are
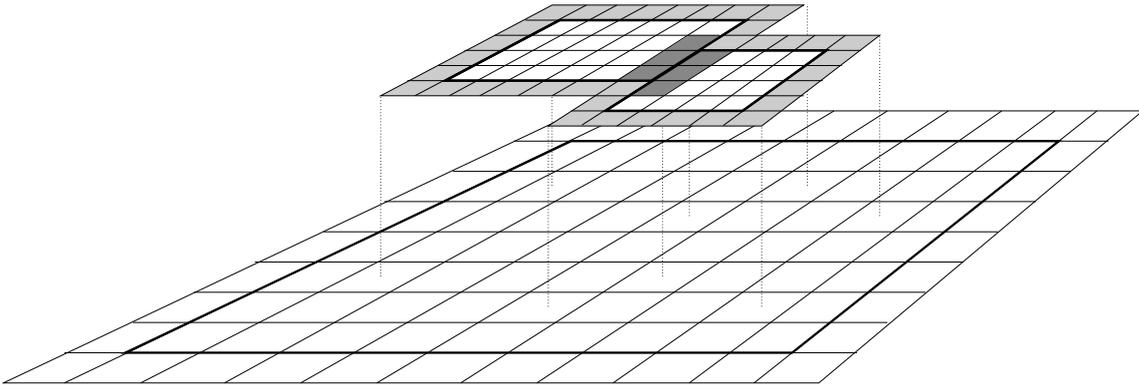
Figure 2.5: Example of the procedure of ghost cell update. A single layer of ghost cells is shown. Some ghost cells can be copied directly (dark grey) from a neighboring patch at the same refinement level, whereas others need to be prolonged from the coarser level below (light grey). Ghost cells of the patch at the coarse level are set through external boundary conditions.
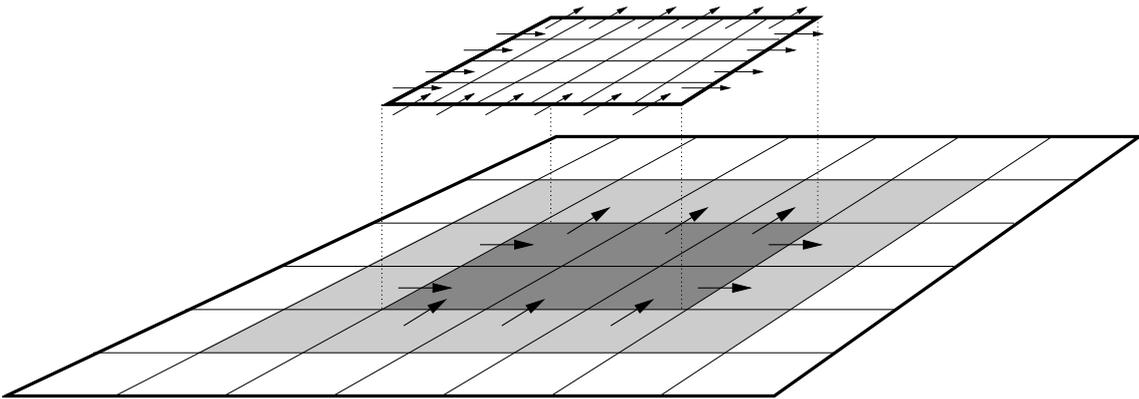
Figure 2.6: Illustration of the flux correction procedure. At coarse-fine boundaries, the fluxes for coarse cells (light grey) need to be replaced by ones combined from the fine cells. After the correction, state vector values from the fine level patch are used to replace the coarse cell values that lie directly underneath (dark grey) in a process called restriction.

applied. Examples of boundary conditions are: inflow, outflow, reflective, periodic, or hydrostatic conditions. Ghost cells that lie at a boundary between sub-domains that are on different processors are filled through inter-processor communication procedures.

Flux correction

After all ghost cells are filled, the fluid flow solver is called for each patch. For a finite-volume scheme, fluxes across cell faces are calculated and used to update the state vector in each cell (Section 2.3). Such a scheme is conservative by construction: what is added to one cell must be taken from another, and only at fine-coarse boundaries a correction needs to be applied (see Figure 2.6).

At these boundaries, the flux at the coarse level is replaced by the sum of the fluxes
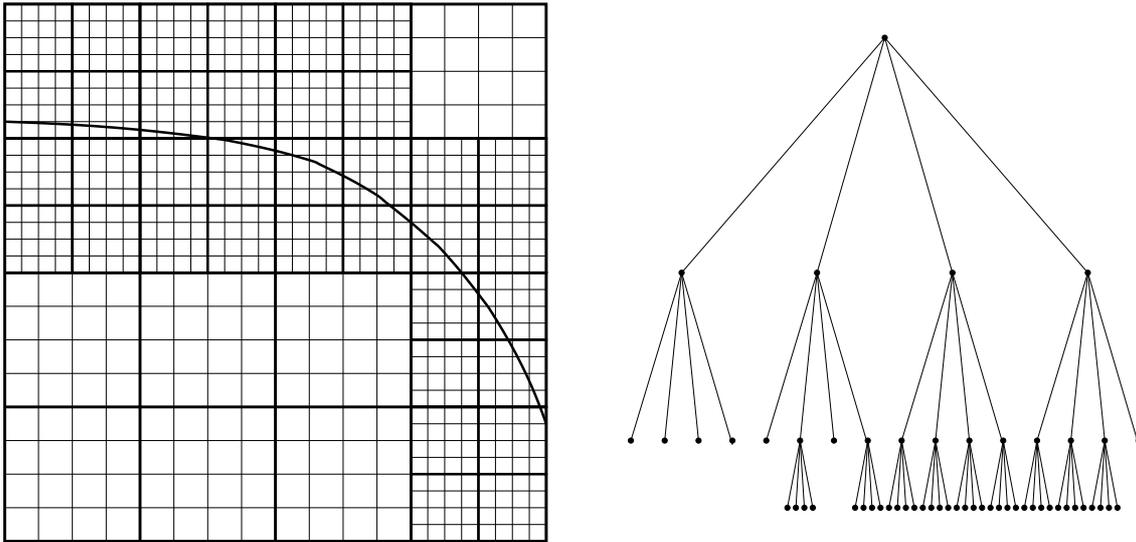
Figure 2.7: 2D example of refinement as employed by *Paramesh* (left). Each patch consists of 4x4 cells. The corresponding quad-tree data structure is shown on the right, where each dot indicates a single patch.

from the fine level. When the flux correction step has finished, all state vectors can be updated. To retain a consistent solution throughout the complete AMR hierarchy, all levels need to restrict their solutions to the coarser level below (Figure 2.6).

2.2.3   AMR in *Flash*: *Paramesh*

In this section we give some details of the AMR package *Paramesh* (MacNeice et al. 2000), as implemented in the *Flash* code. We describe the differences of approach to AMR with the one by Berger & Oliger (1984).

Instead of using rectangular patches that may have arbitrary sizes, *Paramesh* employs patches that all have the same dimensions. Refinement is implemented by subdividing a patch (cf. Figure 2.7), where the connectivity between patches is stored in a tree-type data structure (quad-tree in 2D, oct-tree in 3D).

There are a number of advantages to this approach. First, there is no need for a pattern recognition step, since cells are clustered into patches by default. Second, since patches from different refinement levels fully overlap, the procedures of prolongation and restriction are particularly easy to implement. Furthermore, domain decomposition and load balancing becomes a straightforward process.

There are also a few drawbacks. Since each patch needs a layer of ghost cells, and since, for a typical simulation, a large number of patches is needed, *Paramesh* has a larger memory overhead than the approach of Berger & Oliger (1984). This large number of ghost cells also results in more time spent in communicating their values between processor nodes.

Besides refinement in space, the approach of Berger & Oliger (1984) also implements refinement in time. In *Flash* however, no refinement in time is applied. Instead, each

level is integrated with the same small time step, dictated by the Courant condition on the finest level. Naively, one may expect this to impair the performance of the method since more time steps are taken on the coarser levels than are strictly needed. However, since the bulk of the cells is located at the finest level, this is where most of the computing time is spent, so using a single small time step at all levels does not have a serious impact on the overall performance of the calculation (see Dursi & Zingale 2004, for an assessment of this issue).

As a refinement criterion, *Flash* uses a modified second derivative (see Löhner 1987; *Flash* User's Guide V2.5). This criterion employs a small constant to be set by the user, which prevents refinement when small ripples occur in the flow.

2.3   Hydrodynamics

Many numerical methods for solving Euler's equations of compressible gasdynamics exist. In astrophysical applications steep gradients, strong shocks, and contact discontinuities often occur, so algorithms that can accurately deal with these phenomena are of great interest. For an extensive overview of the different methods, we refer the reader to the textbooks by Laney (1998) and Leveque (2002).

One particularly successful class of solvers, introduced by Godunov (1959), considers the interaction between two adjacent computational cells as a Riemann problem. This approach has the advantage that shocks and contact discontinuities are captured automatically. The constant average state in the cells serves as initial conditions, and fluxes at the interface are obtained using an (approximate) solution to the Riemann problem.

When the value of the state in a cell is taken to be the average of the true distribution, and when fluxes across cell boundaries are used to update the cells, one speaks of a finite volume method. Such an approach has the important property that it explicitly conserves mass, momentum, and energy.

Second order extensions to Godunov's first order method were developed by several authors (e.g. Van Leer 1979; Colella & Woodward 1984; Roe 1986), and the following section gives a terse description of one of them.

2.3.1   The piecewise-parabolic method

For the purpose of completeness, this section briefly recapitulates the direct Eulerian version of the piecewise-parabolic method (PPM) as implemented in *Flash*. PPM was used for all 3D AMR simulations presented in this thesis. For mathematical details one may consult the original papers by Colella & Woodward (1984) and Colella & Glaz (1985), and the descriptions given in Fryxell et al. (2000) or the *Flash* User's Guide (V2.5).

PPM solves Euler's equations in two steps. First, the value of the average state in a cell is approximated by a properly constrained parabolic function. In a second step, these parabolae are used to construct the initial states for a Riemann problem, which is solved using an iterative method.

In the first step, values for the primitive variables at the interface between computational cells are obtained by higher order interpolation of the average state in the cells. To

this end, a cubical polynomial is chosen such that, when integrated over space, it reproduces the average state in the cells. Using a slope limiter, a resulting interface value is restricted to lie between the average values of its neighboring cells. The interface values thus obtained, together with the average state in the cell, are used to construct a parabola which is also constrained to have the average value when integrated over the cell. A special algorithm to detect contact discontinuities is used to steepen the mass density profile by applying the same slope limiter just mentioned, whereas a flattening procedure is used in regions near shocks to prevent oscillations from developing in the flow. Also, a monotonicity constraint is applied to ensure that the parabola only acquires values that lie between the corresponding interface states. It is this constraint that introduces jumps between interface values of the different parabolae, giving the method its name.

Once the parabolic distribution of the state is known, it is used to derive the initial conditions for the Riemann problem. Characteristics are traced back in time from the cell interface so that the proper contribution by each wave family to the state can be determined. The solution to the Riemann problem, for which the rarefaction fan is approximated by a linear interpolation between the head and the tail of this wave, is obtained using a secant iteration scheme.

In spite of the flattening procedure mentioned above, a small amount of additional, multidimensional dissipation needs to be applied in shocks. Apart from reducing postshock oscillations, this has the additional advantage of providing a weak coupling between cells in different coordinate directions.

2.4 Radiative processes

To properly model the dynamical evolution of Planetary Nebulae, the influence of radiative processes should be taken into account. This means that the effects of photo- and collisional ionization, photoheating, and radiative cooling need to be calculated. By tracing rays from the stellar source to each cell in the computational domain, the column density $N(\boldsymbol{r})$ at position $\boldsymbol{r}$ is obtained. The optical depth is then given by

$$\tau_\nu(\boldsymbol{r}) = a_\nu(\mathrm{HI})N(\boldsymbol{r}) , \tag{2.1}$$

with $a_\nu(\mathrm{HI})$ the frequency dependent hydrogen absorption cross section which has a functional form

$$a_\nu = a_{\nu_0} \left(\beta(\nu/\nu_0)^{-s} + (1-\beta)(\nu/\nu_0)^{-s-1} \right) , \; \nu > \nu_0 , \tag{2.2}$$

where $a_{\nu_0} = 6.3 \times 10^{-18}$ cm^2, $\beta = 1.34$, and $s = 2.99$ (Osterbrock 1989). A method for tracing rays through a domain decomposed AMR hierarchy of patches is presented in Chapter 3.

Once the optical depth up to each cell face is known, the ionization fractions and temperature can be computed. For this we use a simplified version of the *Doric* routines (see Mellema & Lundqvist 2002; Frank & Mellema 1994a). In the following sections, we summarize the way in which these routines calculate the ionization, heating, and cooling rates (for more details refer to Frank & Mellema 1994a). Although the *Doric* package is

capable of handling a large number of species (H, He, C, N, O, and Ne), we use hydrogen as the only gas component in order to keep the complexity of our simulations to a minimum, and we will therefore describe just this case.

2.4.1 Ionization

The ionization fractions of hydrogen are given by

$$x(\mathrm{HI}) = \frac{n(\mathrm{HI})}{n(\mathrm{H})}, \ x(\mathrm{HII}) = \frac{n(\mathrm{HII})}{n(\mathrm{H})}, \tag{2.3}$$

with

$$n(\mathrm{H}) = n(\mathrm{HI}) + n(\mathrm{HII}) \tag{2.4}$$

the total hydrogen number density. The electron number density follows from

$$n_\mathrm{e} = n(\mathrm{HII}) + n(\mathrm{C}), \tag{2.5}$$

where the number density of carbon is included to prevent the possibility of $n_\mathrm{e} = 0$ by assuming that carbon is always at least singly ionized due to a background FUV field.

For hydrogen, the number of photoionizations per second is given by (Osterbrock 1989)

$$A_\mathrm{p} = \int_{\nu_0}^{\infty} \frac{4\pi J_\nu}{h\nu} a_\nu(\mathrm{HI})\, d\nu, \tag{2.6}$$

with J_ν the local mean intensity of the radiation field and ν_0 the hydrogen ionization threshold frequency.

The number of collisional ionizations per second is calculated using

$$A_\mathrm{c} = A_\mathrm{c}(\mathrm{HI}) n_\mathrm{e} \sqrt{T} \exp(-I(\mathrm{HI})/kT) \tag{2.7}$$

with $A_\mathrm{c}(\mathrm{HI}) = 5.84 \times 10^{-11}\,\mathrm{cm^3 K^{-1/2}}$, and $I(\mathrm{HI})$ the hydrogen ionization potential (Cox 1970).

For the on-the-spot approximation, the radiative recombination rate is given by (cf. Osterbrock 1989)

$$\alpha_\mathrm{B} = \alpha_\mathrm{B}(10^4\mathrm{K}) \left(\frac{T}{10^4} \right)^{-0.7}, \tag{2.8}$$

with $\alpha_\mathrm{B}(10^4\mathrm{K}) = 2.59 \times 10^{-13}\,\mathrm{cm^3 s^{-1}}$. The temperature is determined from the pressure using

$$p = (n(\mathrm{H}) + n_\mathrm{e}) kT. \tag{2.9}$$

The rate equation for the hydrogen ionization fraction is given by

$$\frac{dx(\mathrm{HII})}{dt} = x(\mathrm{HI})A - x(\mathrm{HII}) n_\mathrm{e} \alpha_\mathrm{B}, \tag{2.10}$$

with $A = A_\mathrm{p} + A_\mathrm{c}$ the total number of photo- and collisional ionizations per second. When one assumes that the electron density is constant, an analytic solution for $x(\mathrm{HII})$ can be found, and iterating for n_e gives the time dependent solution (Schmidt-Voigt & Köppen 1987). Since A_c and α_B are both temperature dependent, the change in temperature due to heating and cooling needs to be recalculated for each iteration step as well.

2.4.2   Heating and cooling

The photoionization heating rate is given by

$$\Gamma_{\mathrm{p}} = n(\mathrm{HI}) \int_{\nu_0}^{\infty} \frac{4\pi J_\nu}{h\nu} a_\nu(\mathrm{HI}) h(\nu - \nu_0)\, d\nu, \qquad (2.11)$$

and for the cooling rate we use a collisional equilibrium cooling curve from Dalgarno & McCray (1972) (more general composition-dependent non-equilibrium cooling is available in the *Doric* package). For calculating the local mean intensity of the radiation field we use a blackbody spectrum, so we have

$$4\pi J_\nu(\boldsymbol{r}) = \left(\frac{R_{\mathrm{S}}}{|\boldsymbol{r}|}\right)^2 \frac{2\pi}{c^2} \frac{h\nu^3}{\exp(\frac{h\nu}{kT_{\mathrm{S}}}) - 1} \exp(-\tau_\nu(\boldsymbol{r})). \qquad (2.12)$$

Here, R_{S} is the radius, and T_{S} is the effective temperature of the source. The optical depth τ_ν at position $\boldsymbol{r}$ follows from equation (2.1). Since evaluating the integrals for the photoionization and heating rate [equations (2.6) and (2.11)] is too time consuming to perform for every value of τ, they are stored in look-up tables for a range of optical depths and interpolated when needed.

2.4.3   Coupling radiation and hydrodynamics

The hydrodynamics and radiation calculations are coupled through operator splitting. To avoid having to take time steps that are the minimum of the hydrodynamics, ionization, and heating/cooling time scales, we use the fact that the equations for the ionization and heating/cooling can be iterated to convergence. Since these are so called 'stiff' equations (e.g. Press et al. 1992), we use a special iteration scheme (Frank & Mellema 1994a). This means that the only restriction on the time step comes from the hydrodynamics (i.e. the Courant condition). See Frank & Mellema (1994a) for an assessment of the validity of this approach.

2.5   Extending *Flash*

Using a publicly available code requires a specific approach, especially when extensions are added to it. Thorough tests should be performed, since the initial conditions one uses are often different from the ones the code was validated for. As with any code, one should compare the numerical results to known analytical solutions. If this is not possible one needs to compare the results to ones obtained with another, similar numerical code, examples of which are given in Sections 5.6 and 6.6.3.

The potential benefits of using a publicly available code will only pay off if extensions can be implemented in a simple fashion. In the case of *Flash* this is particularly straightforward since it is well documented and has a highly modular design (see the *Flash* User's Guide V2.5). It implements an interface that facilitates the communication between the different modules by accessing global variables through special purpose database calls. It

also presents the user with a simple mechanism for introducing new global variables and runtime parameters at pre-compile time, which are made available in the code by special source code generating scripts. These setup scripts also collect for compilation only those modules that are required by the user for a specific run.

To compile *Flash* two additional libraries need to be installed: one that implements the input/output of data files, and one that handles the communication between processors. For the (parallel) i/o, one can chose between either HDF5[1] or netCDF[2]. Inter-processor communication is implemented using the Message Passing Interface[3] (MPI).

2.6   Public codes

This section lists some AMR hydrodynamics codes that are currently publicly available. All these codes are three-dimensional, parallelized, and implement block-structured AMR:

- *AMROC*[4] is a C++ AMR implementation based on *DAGH*[5]. *AMROC* uses the F77 package *CLAWPACK*[6] to solve the fluid equations.

- *Chombo*[7] is a C++ package that provides an infrastructure for AMR calculations.

- *Enzo*[8] combines AMR with an N-body algorithm. It is written in F90 and aimed at cosmological applications.

- *Flash*[9] is a F90 general purpose AMR code developed for simulating thermonuclear explosions in compact stars. *Flash* uses *Paramesh*[10] as its AMR library.

- The *Versatile Advection Code*[11] (*VAC*) is a F90 package targeted at multi-dimensional magnetofluid simulations, of which an AMR version for Cartesian grids is available.

---

[1]http://hdf.ncsa.uiuc.edu/HDF5/
[2]http://www.unidata.ucar.edu/software/netcdf/
[3]http://www-unix.mcs.anl.gov/mpi/
[4]http://amroc.sourceforge.net/
[5]http://www.caip.rutgers.edu/~parashar/DAGH/
[6]http://www.amath.washington.edu/~claw/
[7]http://seesar.lbl.gov/anag/chombo/
[8]http://cosmos.ucsd.edu/enzo/
[9]http://flash.uchicago.edu/
[10]http://ct.gsfc.nasa.gov/paramesh/Users_manual/amr.html
[11]http://www.phys.uu.nl/~toth/

CHAPTER 3

Hybrid Characteristics

We have developed a three-dimensional radiative transfer method designed specifically for use with parallel adaptive mesh refinement hydrodynamics codes. This new algorithm, which we call hybrid characteristics, introduces a novel form of ray tracing that can neither be classified as long, nor as short characteristics, but which applies the underlying principles, i.e. efficient execution through interpolation and parallelizability, of both. We present a detailed description of the different components constituting the new algorithm, with a focus on ray tracing, data interpolation, and the amount and pattern of communication. We also provide a brief comparison between our method and two other, recently proposed, algorithms, that share some features with ours.

Primary applications of the hybrid characteristics method are radiation hydrodynamics problems that take into account the effects of photoionization and heating due to point sources of radiation. The method is implemented into the hydrodynamics package *Flash*. The ionization, heating, and cooling processes are modeled using the *Doric* ionization package. Upon comparison with the long characteristics method, we find that our method calculates the column density with a similarly high accuracy and produces sharp and well defined shadows. We demonstrate the quality of the new algorithm in an application to the photoevaporation of multiple over-dense clumps. We show that the interaction of the photoevaporation flows produced by these clumps results in the creation of an optically thick region between them. This structure may explain the appearance of binary proplyds observed in HII regions like NGC 3603 and the Orion Nebula, and the excess emission observed in between some cometary knots in for example the Helix nebula. This additional structure is also responsible for the emergence of extra shadows which become apparent when the high density interaction zone cools, recombines, and becomes optically thick. These shadows may influence the evolution and survival time of clumps that lie farther away from the source.

The test problems demonstrate the feasibility of our method for performing high resolution three-dimensional radiation hydrodynamics calculations that span a large range of scales. Initial performance tests show that the ray tracing part of our method takes less time to execute than other parts of the calculation (e.g. hydrodynamics and adaptive mesh refinement), and that a high degree of efficiency is obtained in parallel execution. Although the hybrid characteristics method is developed for problems involving photoionization due to point sources, the algorithm can be easily adapted to the case of more general radiation fields.

E.-J. Rijkhorst, T. Plewa, A. Dubey, and G. Mellema
Submitted to Astronomy & Astrophysics

3.1 Introduction

Current multi-dimensional parallel adaptive mesh refinement (AMR, see Berger & Oliger (1984); Berger & Colella (1989)) hydrodynamics codes, include more and more physical processes like (self-) gravity, nuclear burning, and composition dependent equations of state. Furthermore, a wealth of different solvers for relativistic or magneto-hydrodynamics, have become available. These codes are in general implemented as a modular framework, facilitating a rather straightforward inclusion of new physics modules, and are often distributed freely for scientific use (Fryxell et al. 2000; O'Shea et al. 2004; Norman 2000).

Since astrophysical applications are many times dominated by radiative processes, it is highly desirable that radiative transfer in some form is included into these codes. Efforts to solve the full equations of radiative transfer (using the Eddington tensor formalism in combination with short characteristics, see Stone et al. 1992), or in the flux-limited diffusion approximation (Turner & Stone 2001; Whitehouse & Bate 2004), together with the hydrodynamics have been made, but it remains a complex task to create a parallel algorithm which combines radiative transfer and hydrodynamics for multi-dimensional calculations that runs efficiently on todays multi-processor supercomputers (e.g. Hayes & Norman 2003).

For many astrophysical applications however, it is not necessary to solve the full set of radiative transfer equations; for these specific cases it is sufficient to just determine the optical depth due to absorption along a line of sight from the source to a certain location in the computational domain. For the purpose of our application of ionization calculations, the optical depth is used to determine the photoionization and heating rates. When this is combined with detailed calculations of radiative cooling, many applications come within reach, such as the evolution of planetary nebulae (Frank & Mellema 1994a), photoevaporation of cosmological mini-haloes (Shapiro et al. 2004), photoevaporation of cometary knots (Lim & Mellema 2003), the evolution of proplyds (e.g. Richling & Yorke 2000), or even simplified scenarios of explosions of massive stars (Janka & Mueller 1996), to name just a few.

In creating a method that combines radiative transfer and hydrodynamics, one in general starts with an existing hydrodynamics code and adds the necessary radiation processes to it (e.g. Mellema et al. 1998; Turner & Stone 2001; Whitehouse & Bate 2004; Heinemann et al. 2005; Liebendörfer et al. 2005). In this chapter we describe the addition of a new radiative transfer algorithm, which we call *hybrid characteristics*, to the parallel 3D AMR hydrodynamics package *Flash* (Fryxell et al. 2000).

Most of the radiative transfer methods that were successfully combined with extant hydrodynamics codes apply some form of ray tracing to find the optical depth at each location in the computational domain. Apart from ray tracing one could also use statistical methods to find the solution to the radiative transfer equations (e.g. Maselli et al. 2003). Yet another approach could be the use of Fourier transforms (Cen 2002), or unstructured grids (Ritzerveld et al. 2004), to determine the radiation field.

For a one-dimensional, non-AMR, serial code, ray tracing becomes a rather straightforward procedure which requires little second thought. Equivalently, the case of a plane parallel radiation field on a Cartesian grid, or a single point-source at the centre of a

spherically symmetric grid, for which all rays run parallel to a coordinate axis, can be handled quite easily. Although this kind of implementation can readily be used to study a number of interesting astrophysical phenomena, it is still highly desirable to have a code that can treat the more general case of a point source of ionizing radiation on a 3D Cartesian domain. Such more general methods were for example implemented by Raga et al. (2000a); Richling & Yorke (2000); Lim & Mellema (2003), but none of these methods was explicitly parallelized for distributed memory machines though.

The aim of this work is to create a characteristics-based radiative transfer method that can handle multiple sources of ionizing radiation in AMR enabled simulations to be run on distributed memory parallel machines. For this, a radical rethink of the concept of ray tracing is necessary, since, for this type of parallel AMR codes, the computational domain is not only sub-divided into a hierarchy of patches, but is also distributed over a number of processors. The first choice one therefore has to make is which flavour of ray tracing one wants to apply: either long or short characteristics. Since these two methods have rather different properties when it comes to efficiency and parallelizability, this choice will determine the success of the final algorithm.

We are aware of a number of other methods that use some form of adaptivity to solve the radiative transfer equations: Abel & Wandelt (2002) designed a method where the ray itself is adaptively split into sub-rays, but the underlying grid is still regular. Steinacker et al. (2002) employed second order finite differencing of the full radiative transfer equations on an oct-tree AMR grid, and, more recently, Juvela & Padoan (2005) implemented a ray tracing method for cell-based AMR. Jessee et al. (1998) presented a radiative transfer method for patch-based AMR that uses the discrete ordinates approach. However, none of these methods resulted in parallel algorithms used in applications in which radiation is coupled to hydrodynamics.

Efforts to create a parallel, two-dimensional, radiation hydrodynamics code were presented by Hayes & Norman (2003), and, more recently, a three-dimensional method by Heinemann et al. (2005), who developed a ray tracing algorithm for decomposed domains. However, none of these two methods uses AMR.

Our presentation begins with Section 3.2 in which we describe our new method. This method can *not* be classified as either short or long characteristics, but does have the desired properties, namely high parallel and computational efficiency, of its two predecessors. We also compare our method to two recent ones which share similar features with ours. Supplemental physics components required by our primary target application (gas ionization, heating, and cooling) are presented in Section 3.3, where we give a brief description of the *Doric* routines (Mellema & Lundqvist 2002, and references therein). In Section 3.4 we first compare the accuracy with which our method calculates column densities to results obtained with a standard long characteristics approach. Then we present a pure radiation transport problem aimed at testing the accuracy of the ionization state calculations and shadow casting. This is followed by a coupled radiation hydrodynamics calculation of photoevaporation flow. Section 3.5 presents some initial performance results. Discussion of possible extensions and future applications for our method together with the conclusions are given in Section 3.6.

3.2 Characteristics based radiative transfer

When calculating the effects of ionizing radiation due to a point source, the radiation field is often dominated by this source, and one can safely ignore contributions to the radiation field due to the ambient gas. This means that the radiative transfer equations assume a particularly simple form, since we can take the total emission coefficient (and thereby the source function) to be equal to zero. Furthermore, when we also ignore the effects of scattering, the solution to the radiative transfer equations for the specific intensity I at location r is given by

$$I(r) = I(0) \exp(-\tau(r)) , \qquad (3.1)$$

and only depends on the optical depth τ, which is defined by

$$\tau(r) = a_0 N(r) , \qquad (3.2)$$

with a_0 the absorption cross section, and N the column density at r.

Once the optical depth is known at every location in the computational domain, one can use it to find the ionization, heating, and cooling rates, and calculate the ionization state and temperature of the gas. Since, for finite-volume hydrodynamics codes, the computational domain is discretized into cells, the optical depth, or, equivalently, the column density for a certain cell, is found by adding the contributions from all cells that lie between the source and the destination cell under consideration. This can be achieved by casting a ray, or *long characteristic*, from the source to the cell, accumulating contributions to the total column density along the way. In case of an AMR hierarchy, the algorithm first needs to identify the patches and cells contained within the patches that are traversed by the ray, and then calculate their local contributions to the total column density.

Although the method of long characteristics is very accurate, it is also rather inefficient, since, the closer a cell is to the source, the more rays cut through (approximately) the same part of the cell, introducing a lot of redundant calculations (see Figure 3.1a). A way to eliminate this redundancy is to use the so called method of *short characteristics*. Here, the total column density for a certain cell is calculated by interpolating upwind values of column density calculated in a previous step, thereby creating some diffusion, but removing the redundant calculations inherent in the long characteristics method (Figure 3.1b). For this to work, the appropriate information from upwind cells needs to be available at all times, which means one needs to sweep the numerical grid outwards from the source. This necessity of having to traverse the grid in a certain order makes this method intrinsically serial, since values of column density in cells now depend on one another. The long characteristics method does not suffer from this restriction, because here contributions to the total column density from cells cut by a ray do not depend on column densities in other cells. Therefore, the long characteristics method is fully parallelizable, since calculations of contributions to the column density along each ray can be performed independently. For our method we combine the desirable qualities of both these approaches; the idea of interpolation is adopted from the short characteristics method, while parallelism is obtained following principles of the long characteristics method.

a   **b**
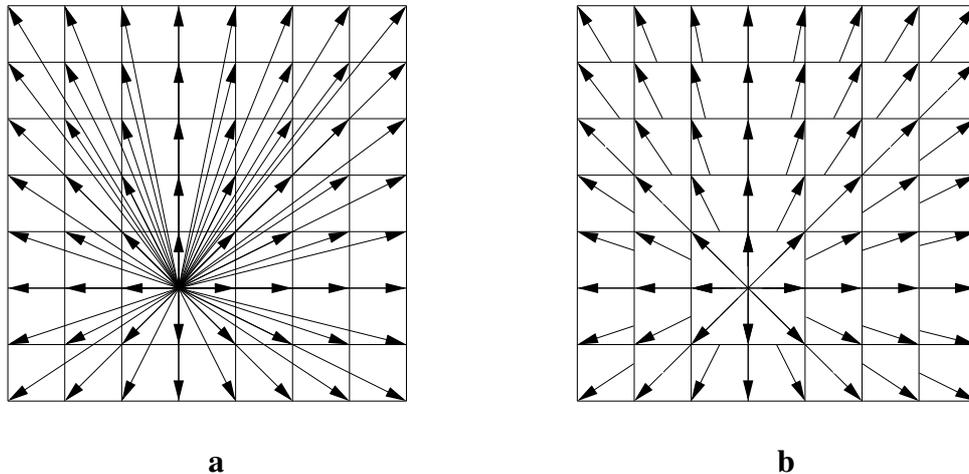
Figure 3.1: Comparing the long (a) and short (b) characteristics method. For the long characteristics method, the closer one gets to the source, the more rays pass through (approximately) the same part of a cell, resulting in a large number of redundant calculations. The short characteristics method does not suffer from this, since here column densities are interpolated from cells that have been dealt with previously, so only the contributions to the column density of the short ray sections that pass from cell to cell need to be computed.

In what follows, we start with a general description of the algorithm used to trace rays on AMR hierarchies. We explain how the long characteristics method is exploited to make this a parallel algorithm, and where the interpolation comes in to increase the efficiency of the calculation.

Although our algorithm is designed for three dimensions, many features of its implementation can be explained using two-dimensional analogues. Whenever the generalization from two to three dimension is non-trivial, we will supply the full, three-dimensional, description. Since the algorithm is naturally subdivided into a number of steps, we will expand on these separately.

3.2.1 The distributed computational domain

Consider a computational domain that is distributed over N_p processors (for a two-dimensional example, see Figure 3.2). Rays are traced over these different sub-domains and must therefore be split up into independent *ray sections*. Naturally, these sections are in the first place defined by the boundaries of each processor's sub-domain, and in the second place by the boundaries of the patches contained within that sub-domain.

So first each processor calculates for all the patches it owns the *local* column densities ΔN. These local contributions are found by tracing ray sections that originate at the patch faces that are located *closest* to the source, and that terminate at the centres of the cells (see

processor 0   **processor 1**
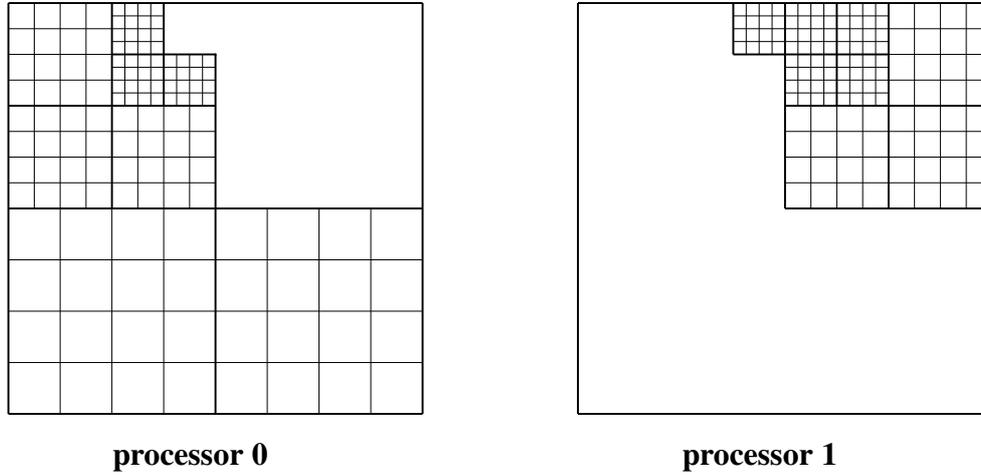
Figure 3.2: Two-dimensional example of an AMR hierarchy distributed over two different processors. Here, each patch contains 4×4 cells.

Figure 3.3). Since finding these contributions is a local process, this part of the algorithm is fully parallel, and can be implemented using either the short or long characteristics method. Details on how the ray tracing for individual patches is implemented are given in Section 3.2.2. Note however that, before each processor can calculate its ΔN, it needs to know the physical location of the source, so this information is made available first.

Since in general rays traverse more than one processor domain, exchange of information has to take place at some point in the algorithm. After this communication step has finished, each processor should have available all contributions of column density to the rays that terminate in its domain. By interpolating and accumulating all these contributions for all rays, one ultimately obtains the total column density for each cell (see Section 3.2.3 for a more elaborate description of the communication patterns involved). Details on the procedure applied to find the patches cut by a ray, and the way in which their contributions to the total column density are subsequently calculated, are given in Section 3.2.3 and Section 3.2.3, respectively.

3.2.2 Ray tracing a single patch

In this section we will explain how the contribution to the total column density along a local ray section in a single patch can be calculated (see Figure 3.3 and Figure 3.7). As explained above, each patch can be dealt with independently, which makes this part of the calculation fully parallel.

The local column density contributions are calculated from

$$\Delta N = \sum_{\text{cells}} x(\text{HI})n(\text{H})\Delta s \,, \tag{3.3}$$

a   **b**
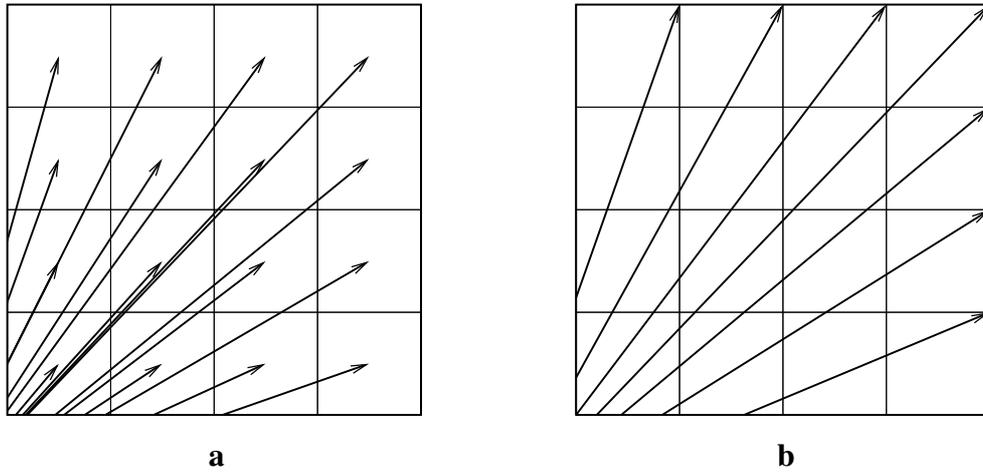
Figure 3.3: Two-dimensional example of ray sections for a single patch. Local contributions to the column density are indicated by ray sections that terminate at cell centres (a), whereas contributions that are to be communicated between processors, and are subsequently used in an interpolation step, terminate at cell corners (b). The source lies outside of the patch in the direction of the lower left corner.

with $x(\mathrm{HI})$ the ionization fraction of neutral hydrogen, $n(\mathrm{H})$ the hydrogen number density, and Δs the physical path length through the cell.

These contributions are found by casting a ray section from the faces of the patch that are located closest to the source towards each cell centre (see Figure 3.3). Column density contributions by the cells that lie inside the patch along each section are calculated using the 'fast voxel traversal algorithm' from Amanatides & Woo (1987) (for more details on this traversal method, see App. 3.A). Besides ray sections that terminate at cell centres, we also need to calculate the column density contribution for ray sections that lead to cell *corners* located at those patch faces that are farthest away from the source (see Figure 3.3). These are the contributions to the column density that need to be communicated (Section 3.2.3), and interpolated (Section 3.2.3) in subsequent steps of the algorithm.

Calculating these ray sections is similar to the method of long characteristics, but since the number of cells per patch is low relative to the effective resolution of the full computational domain, this actually does not impair the performance of the method too much (see Section 3.5 for an analytical comparison of our method with the short and long characteristics one for a regular grid).

We also considered using short instead of long characteristics to ray trace a single patch (see App. 3.B for a description of a possible implementation). However, although the short characteristics method executes presumably more efficient than the long characteristics one, the first requires interpolation, whereas the latter simply adds up column density contributions by individual cells. When the number of cells that need to be tra-

versed is relatively small, as is the case when ray tracing the single patches, these extra calculations may render the short characteristics method even less efficient than the long characteristics one. Furthermore, the interpolation introduces undesirable diffusion. We therefore decided to implement the more accurate and straightforward ray tracing approach of Amanatides & Woo (1987).

3.2.3 Hybrid Characteristics

As was mentioned above, in AMR hydrodynamics codes, each processor owns a sub-domain of the computational volume which is covered by a collection of patches. In order to obtain the total column density for a certain ray that traverses these sub-domains, individual local contributions by the patches need to be accumulated. This can be interpreted as applying the method of long characteristics, in this case *not* to add up contributions from individual cells, but instead to add up contributions from individual patches. So here our algorithm does again make use of long characteristics but now at the level of patches.

Since each processor knows the direction of its rays and the co-ordinates where they terminate, it can find the patches cut by these rays and perform the required calculations. For certain flavours of AMR, patches from different refinement levels may partially overlap. In such cases, one would have to make sure that only parts of the patches that contain valid data (i.e., the data from regions resolved to the highest resolution) are considered in the calculation of the column density. One way to eliminate the overlap is to apply a procedure called 'grid homogenization', as described by Kreylos et al. (2002).

For the oct-tree type of AMR implemented in *Flash*, patches from different refinement levels do not overlap. Patches are either fully covered by still more refined patches or otherwise contain valid data (the latter are the so-called 'leaf patches' in terminology of *Flash*). Therefore, a simple check to see if a patch is a 'leaf patch' is sufficient to determine whether or not it should contribute to the total column density along the ray.

Once the list of patches traversed by a ray is known, we loop through it, and determine the local column density contributed by each patch to the total column density for the ray. Unless the ray terminates in the patch under consideration, it will in general not exit a patch exactly at a cell corner. This means that we need to interpolate the values of column density contribution ΔN, obtained earlier (using either the short or long characteristics method as described in Section 3.2.2) at that face of the patch where the ray leaves it.

We would like to emphasize that, although our method makes use of some form of long characteristics, *nowhere* in the algorithm is a ray traced on a cell-by-cell basis over the *full* computational domain. To the contrary, ray sections are traced through the cells of each patch and it is these local contributions which are combined through interpolation by performing another ray trace, this time not over cells but over patches, as described in Section 3.2.3. This is why we call our algorithm *hybrid characteristics*.

Below, we first explain how the local column density contributions ΔN, obtained with one of the methods from Section 3.2.2, are communicated between processors. Then we describe how the list of patches traversed by a ray is constructed, after which we show the way in which this list is used to calculate the contributions to the total column density N.

Communicating local column density contributions

Since, for a parallel AMR hydrodynamics code, the patches are distributed over a number of processors, communication between processors is inevitable at certain points in the algorithm. In particular, as soon as the local contributions to the column density have been calculated (Section 3.2.2), values of these ΔN located at patch faces that are farthest away from the source are communicated between processors. In this way, each processor has the information regarding the face values of local column density from *all* patches in existence (i.e. the so called 'gather' operation is used). Apart from these face values, all processors also need information about the location and size of each patch and its refinement level in order to determine if a particular ray cuts a patch. This information is communicated using the 'gather' operation as well.

The size of the messages to be communicated and the memory needed for storage of this information is given by

$$P_{\text{tot}}\, p_{\max}\, S \,, \tag{3.4}$$

where P_{tot} is the total number of processors, $p_{\max}$ is the maximum number of patches in existence on any processor, and S is the required storage space per patch. In three dimensions, S should contain the values of ΔN from the three patch faces located farthest away from the source, as well as the location, size, and refinement level information of each patch.

For an initial test of the performance of the algorithm as a whole, and of its communication patterns in particular, see Section 3.5.

Constructing the list of patches cut by a ray

A straightforward approach to constructing the list of patches traversed by a ray would be to simply check for all patches whether or not they are cut by the ray under consideration. Since this would have to be done for all rays, and since there are as many rays as there are cells, this approach quickly becomes prohibitively slow. We therefore developed a new, more elaborate, but much faster method to find the list of patches cut by a ray.

First, each processor creates a so called 'patch-mapping' which consists of an integer array representing the full computational domain that stores the id (i.e. a unique integer identifier) of all patches containing valid data. In Figure 3.4 we show an example of such a mapping. These local patch-mapping arrays then need to be communicated and merged (using a so called 'reduce' communication operation) after which each processor has the same global patch-mapping corresponding to the full computational domain.

In order to discern patches that are on different processors we use the following coding for the global patch id:

$$p_G = p_L + P\, p_{\max} \,, \tag{3.5}$$

with p_G the global patch id, p_L the local patch id, and P the processor id.

We then trace the ray, again using the 'fast voxel traversal algorithm' (Amanatides & Woo (1987), see App. 3.A), but now to trace through the *global patch-mapping array*. This results in the list of patches cut by the ray, which is used to accumulate their local contributions, which were already communicated earlier, to arrive at the total column density (as described in Section 3.2.3).

PSfrag replacements
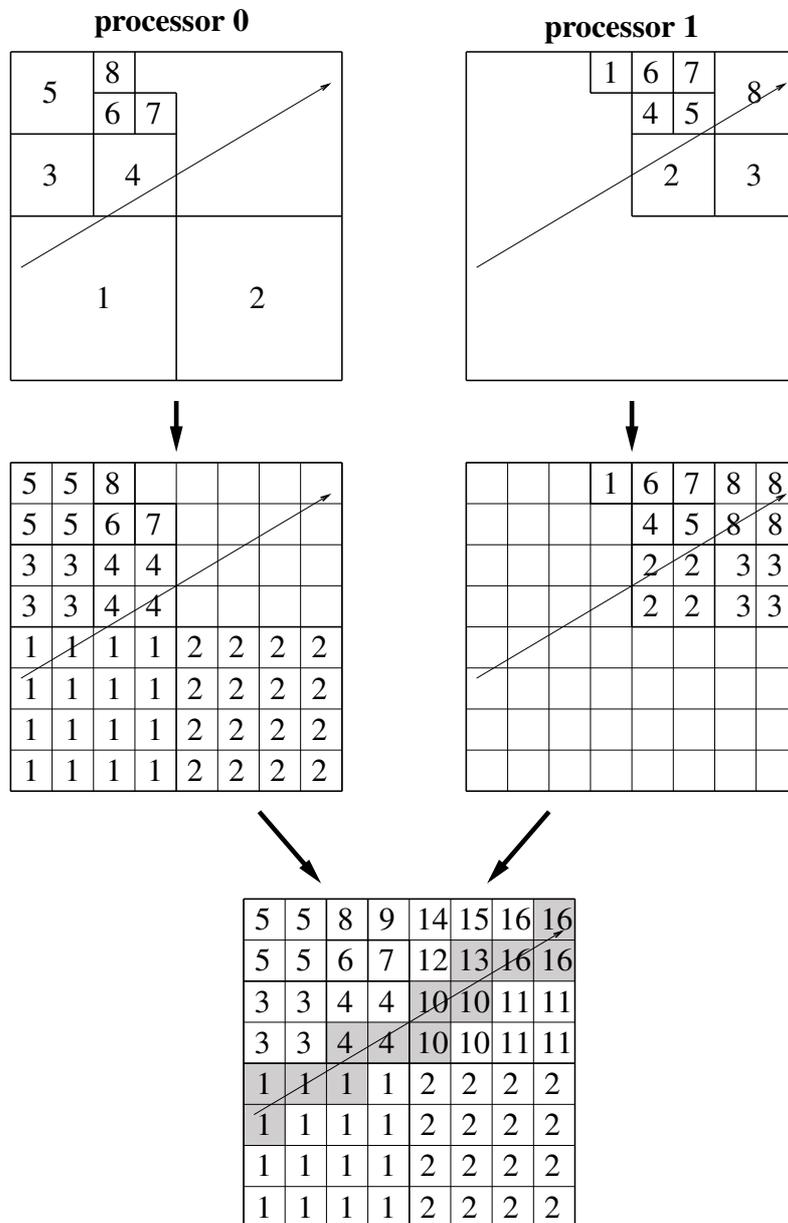
Figure 3.4: Two-dimensional example of a 'patch-mapping' for a computational domain that is split over two different processors. In the top row the local ids of the patches on the different processors are shown. The mapping of these patch ids onto the patch-mapping array is shown in the middle row. The bottom row shows the global patch-mapping after the local patch-mappings have been communicated. Tracing the depicted ray results in the patch list $\{1, 4, 10, 13, 16\}$. The patch-mapping entries visited during the ray tracing are shown in grey.
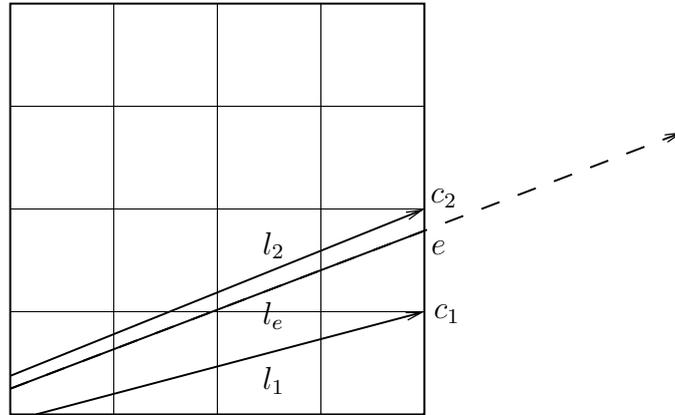
Figure 3.5: Two-dimensional illustration of the linear interpolation scheme used to accumulate local column density contributions. Shown are the ray sections used in the interpolation (see text for further details).

Although this approach to ray tracing can be a potential bottle-neck in the algorithm, one needs to keep in mind that the maximum number of patch-mapping entries along a ray is given by $\sqrt{3}C/c$, with C^3 the total number of cells if the computational domain would be fully refined, and c^3 the number of cells per patch.

For a typical three-dimensional oct-tree type AMR simulation with $C = 512$ and $c = 16$, we find a maximum amount of ~ 55 patch-mapping entries that are cut by a ray. Note however that this is an upper limit. The number of entries is drastically smaller when the source and destination of the ray are not located at opposite sides of the domain (which will be the case for most rays). Note also that, although we have to trace through the patch-mapping entries, the actual number of patches that ends up in the list is strongly reduced due to the adaptive nature of the discretization. In the example given in Figure 3.4, the number of patch-mapping entries visited by the ray is thirteen, but the number of patches that end up in the list is only five. It is this latter number which determines how many interpolations are needed when accumulating the local column density contributions.

Accumulating local column density contributions

Now that we have the list of patches traversed by a ray (Section 3.2.3) and the values of local column density at the patch faces located farthest away from the source have been made available to all processors (Section 3.2.3), we can proceed and calculate the local contributions to the total column density through interpolation.

The calculations that need to be performed for a ray r traversing a patch p can be broken up into the following steps (two-dimensional case, see Figure 3.5):

1 Find the location e where r exits p.

2 Use this to find the two cell corners c_1 and c_2 that are closest to e and store their corresponding local column density contributions ΔN_1 and ΔN_2.

3 Calculate the geometrical path lengths of the ray sections that terminate in c_1, c_2, and e, and denote these by l_1, l_2, and l_e, respectively.

4 Use these path lengths to calculate the following normalized interpolation weights:

$$w_1 = |l_2 - l_e|/(l_1 + l_2), \; w_2 = |l_1 - l_e|/(l_1 + l_2) \,. \tag{3.6}$$

5 Calculate the desired value of local column density at e through linear interpolation:

$$\Delta N_e = w_1 \, \Delta N_1 + w_2 \, \Delta N_2 \,. \tag{3.7}$$

After all ΔN_e for each patch in the list of patches cut by r are calculated, we simply need to sum them to arrive at the total column density for r:

$$N(r) = \sum_p \Delta N_e(p) \;\; [p \in \text{list}(r)] \,. \tag{3.8}$$

The interpolation weights given above were constructed using the conditions

$$w_1 \, l_1 + w_2 \, l_2 = l_e, \;\; \text{and} \;\; w_1 + w_2 = 1 \,, \tag{3.9}$$

which, for the case of a homogeneous density distribution, results in the *exact* solution for the column density (i.e., apart from a constant factor, the path length itself). Other weights, like ones derived from the distances between the exit locations e, c_1, and c_2, can also be used, but this leads to $\sim 10\%$ errors for rays that enter a patch close to a patch corner (as is depicted by the example ray section of Figure 3.5).

In three dimensions (see Figure 3.6) it is not straightforward to derive weights that are a generalization of the two-dimensional ones described above. We therefore give a more intuitive derivation of these weights, using a procedure where we apply the weights for the two-dimensional case twice in succession:

1 Find the location e where r exits p.

2 Use this to find the four cell corners c_1, c_2, c_3, and c_4 that are closest to e and store their corresponding local column density contributions ΔN_1, ΔN_2, ΔN_3, and ΔN_4.

3 Calculate the geometrical path lengths of the ray sections that terminate in c_1, c_2, c_3, c_4, and e and denote these by l_1, l_2, l_3, l_4, and l_e, respectively. Also calculate the path lengths l_5 and l_6 of the ray sections that terminate in c_5 and c_6 respectively (see Figure 3.6).

4 Use these path lengths to calculate the following normalized interpolation weights:

$$\begin{aligned}
w_1 &= |l_2 - l_5|/(l_1 + l_2), & w_2 &= |l_1 - l_5|/(l_1 + l_2) \,, \\
w_3 &= |l_4 - l_6|/(l_3 + l_4), & w_4 &= |l_3 - l_6|/(l_3 + l_4) \,, \\
w_5 &= |l_6 - l_e|/(l_5 + l_6), & w_6 &= |l_5 - l_e|/(l_5 + l_6) \,.
\end{aligned} \tag{3.10}$$
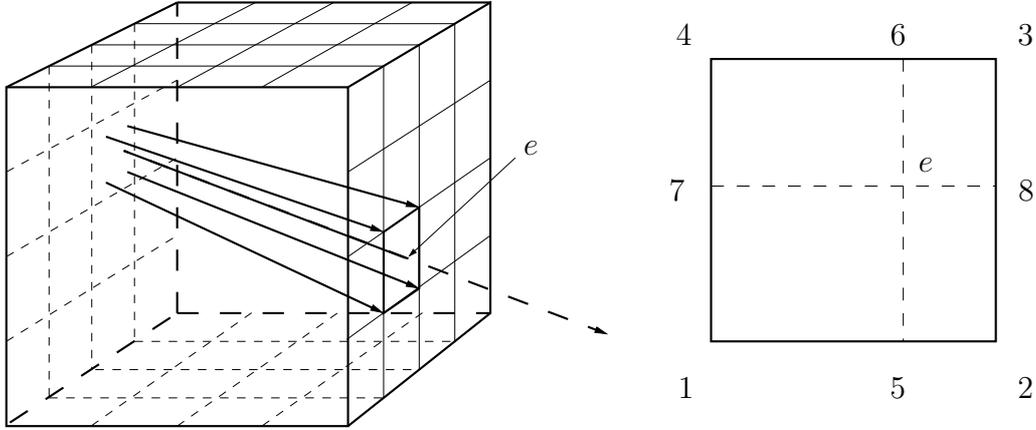
Figure 3.6: Illustration of the interpolation scheme in three dimensions. For clarity we show outlines of cells on patch faces only. In the left image we show a ray r that exits the patch at location e through a cell face, together with the ray sections used in the interpolation that terminate at the corners of this cell face. The image on the right shows the cell face in more detail, where we indicated the cell corners by 1, 2, 3, and 4. In addition to these cell corners, ray sections used in the interpolation that terminate at 5, 6, 7, and 8 are also indicated (see text for further details).

5 Calculate the values of local column density ΔN_5 and ΔN_6 at c_5 and c_6 through linear interpolation:

$$\begin{aligned}
\Delta N_5 &= w_1 \, \Delta N_1 + w_2 \, \Delta N_2 \, , \\
\Delta N_6 &= w_3 \, \Delta N_3 + w_4 \, \Delta N_4 \, .
\end{aligned} \tag{3.11}$$

6 Calculate the desired value of local column density at e through linear interpolation of ΔN_5 and ΔN_6:

$$\Delta N_e = w_5 \, \Delta N_5 + w_6 \, \Delta N_6 \, . \tag{3.12}$$

Our choice of using the values of local column density at c_5 and c_6 to arrive at ΔN_e is arbitrary. Instead, one may also use the ones from c_7 and c_8 (cf. Figure 3.6) in the steps described above.

The main difficulty in finding an interpolation scheme for the three-dimensional case lies in the fact that we need to weigh with the *lengths* of the ray sections to avoid the errors which will otherwise occur when the ray under consideration enters the patch close to a patch corner. Since in general all these path lengths are different from one another, this introduces quite a number of independent variables into the equations. So, although the two-step procedure just described is not unique, it is simple and fast, and it gives good results in practice.
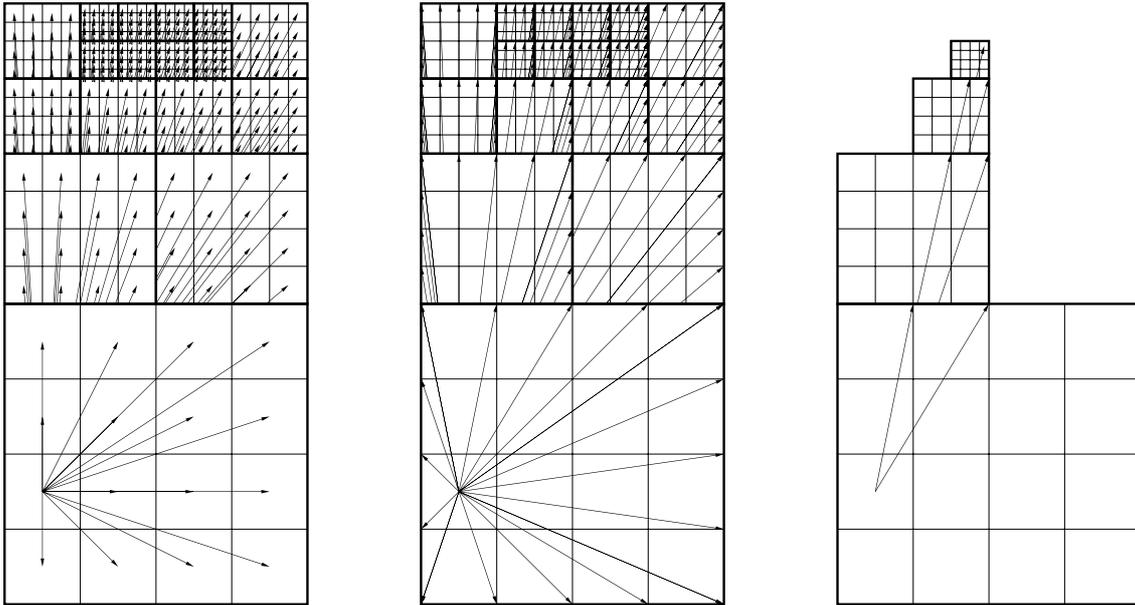
Figure 3.7: Summary of steps taken in the hybrid characteristics method. On the left we show ray sections that represent local contributions to the column density (summary step 3), whereas ray sections that represent values of column density that need to be communicated are shown at the centre image (summary step 4). Note that only those values on patch faces located farthest away from the source need to be communicated. On the right we show an example of the interpolation of these local values for a particular destination cell (summary step 6). Note that there is no need to interpolate the value for the final ray section in the destination patch since its value was already calculated previously (summary step 3).

3.2.4   Summary of the algorithm

The steps taken in the algorithm can be summarized as follows (see Figure 3.7):

1. Each processor checks if its sub-domain contains the source. The processor that owns the source stores its patch and processor id and makes it available to all other processors (broadcast). Note that this id may change during a simulation due to changes in refinement and the consequent redistribution of patches among processors.

2. On each processor, create the local patch-mapping and communicate (reduce) it so that each processor ends up with the global patch-mapping (Section 3.2.3).

3. On each processor, calculate local column density contributions ΔN for each patch using the 'fast voxel traversal algorithm' (Section 3.2.2 and Figure 3.7 left).

4. Communicate (gather) all ΔN values at patch faces located farthest away from the source (Figure 3.7 centre). Also, the coordinates and refinement levels of all patches need to be gathered. This communication is done most efficiently when this information is combined in a single data type of size S (Section 3.2.3).

5 On each processor, construct for each ray the list of patches that are traversed by that ray (Section 3.2.3).

6 On each processor, interpolate and accumulate the local contributions ΔN from the patches that are in the list to arrive at the total column density N (cf. Section 3.2.3 and Figure 3.7 right) .

3.2.5 Comparison to other methods

To conclude this section, we compare our method to two more recent ones that either use some form of adaptivity to trace rays (Juvela & Padoan 2005), or that are parallelized for distributed memory architectures (Heinemann et al. 2005). Unlike ours, these methods are intended to solve for the full radiation field, and therefore need to employ multiple sets of rays to sample the angular parameter space. Depending on the adopted form of the source function, (lambda-)iteration is to be performed as well in order to obtain a converging solution.

Juvela & Padoan (2005) proposed a ray tracing method for cell based AMR intended to be used in calculations of line emission. Their method uses sets of parallel (in the geometrical sense) long characteristics to find the intensity at cell faces, which are then interpolated to get the intensity at the cell centre using a short characteristic. This is repeated for a number of directions after which angle averaged quantities are obtained. This process is then lambda-iterated to get converging line intensities.

Since their method refines on a cell-by-cell basis, and ours employs patches structured in an oct-tree hierarchy, there is a one-to-one correspondence between the procedures of ray tracing used in the two methods: their long characteristics correspond to our ray tracing of the patch-mapping, whereas their short characteristics correspond to our ray tracing of a single patch.

More recently, Heinemann et al. (2005) developed a method for tracing rays through a decomposed computational domain (i.e. sub-domains that are distributed over a number of processors). To sample the radiation field, rays are traced that are either parallel or diagonal to a regular patch. As they mention, this means that there is no need for them to interpolate local values. Furthermore, since their source function acts only locally, their is no need to iterate the solution.

As in our approach, Heinemann et al. (2005) first obtain all local contributions (which they call 'intrinsic') and add these up to arrive at the total solution. However, in contrast to our method, the communication pattern of Heinemann et al. (2005) is intrinsically serial (i.e. processors have to wait for one another, see their Figure 1). In their case of a decomposed regular domain, the performance penalty due to the serial nature of their algorithm is small, but in case of an AMR type of grid, the performance would be severely degraded. Heinemann et al. (2005) also consider the special case of periodic boundary conditions with rays running only parallel along a coordinate axis. In such a situation, the boundary values are broad casted and the inter-processor communication is more efficient than in the serial case.

Although our method is designed to study the effects of ionization due to point sources of radiation, it can be easily adapted to trace sets of parallel rays instead. Depending

on the application, a prescription for the source function and (lambda-)iteration would need to be implemented. This would make our method suitable for solving the radiative transfer equation in a more general way, similar to the methods just discussed. The added advantage of such an approach is that our method is highly parallel *and* coupled to an AMR hydrodynamics code.

3.3   Ionization, heating, cooling

When the column density from the source up to each cell face is known, the ionization fractions and temperature can be computed. For this we use a simplified version of the *Doric* routines (see Mellema & Lundqvist 2002; Frank & Mellema 1994a), as discussed in Section 2.4.

These routines calculate the photo- and collisional ionization, the photoheating, and the radiative cooling rate. Using an analytical solution to the rate equation for the hydrogen ionization fractions, the temperature and ionization fractions are found through an iterative process. Since evaluating the integrals for the photoionization and heating rate is too time consuming to perform for every value of the optical depth, they are stored in look-up tables and are interpolated when needed.

The hydrodynamics and ionization calculations are coupled through operator splitting. To avoid having to take time steps that are the minimum of the hydrodynamics, ionization, and heating/cooling time scales, we use the fact that the equations for the ionization and heating/cooling can be iterated to convergence. This means that the only restriction on the time step comes from the hydrodynamics (i.e. the Courant condition).

3.4   Tests

In this section we present a number of tests for our new algorithm. First, we discuss the accuracy with which column densities and ionization fractions are calculated. We compare the results obtained with the hybrid characteristics method to those calculated using the long characteristics one. Since the interpolation scheme of our method is designed to give the exact result for the column density in case of a homogeneous density distribution (Section 3.2.3), we also consider its performance in case of a more general density field.

We conclude this section by testing the shadow casting capabilities of our method and apply it to a 'real-world' application of photoevaporating flows. This last test demonstrates the performance of out method when used in combination with hydrodynamics.

3.4.1   Column density

We performed two-dimensional calculations where we placed a single point source at the centre of a $1/r^2$ density distribution, the result of which is shown in Figure 3.8. In order to prevent an under-resolved singularity at the location of the source, we used a constant density sphere with a radius of 5×10^{14} at the source location.
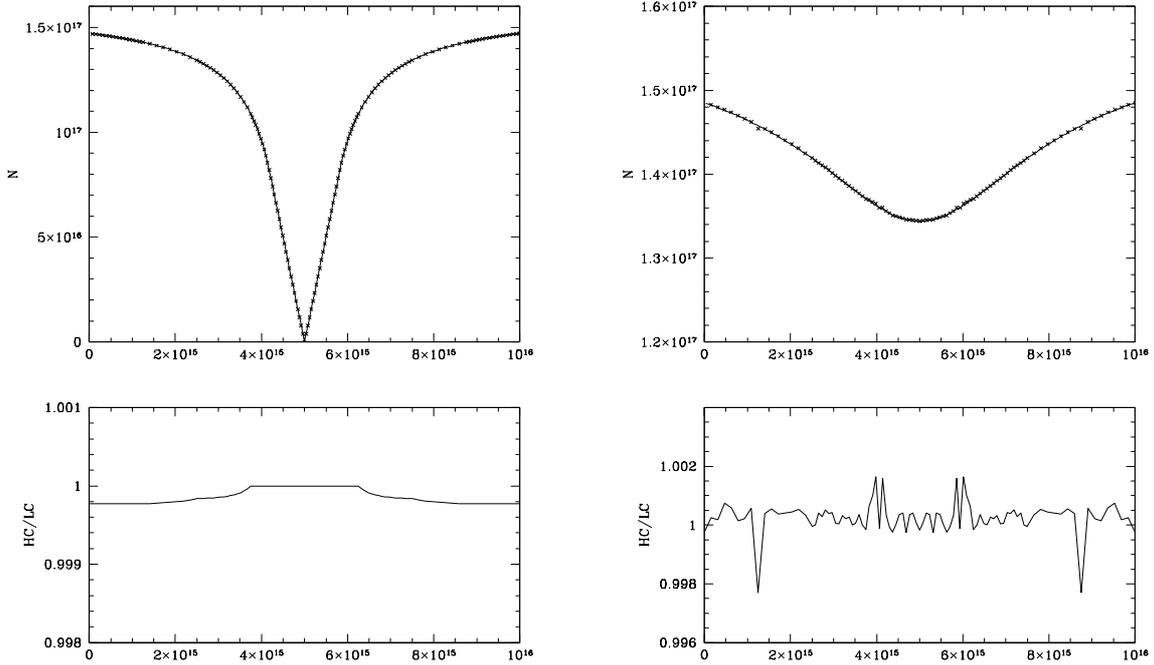
Figure 3.8: Values of column density for the case of a single point source in a two-dimensional domain with a $1/r^2$ density distribution. Shown are one-dimensional cuts along the y-direction through the source located at the centre of the domain (left panels) and at $3/4$ of the domain (right panels). In the top panels, the solid line indicates the result for the long, whereas the crosses indicate the result for the hybrid characteristics method. The bottom panels show the ratio (hybrid/long) of column density values.

The left panel of Figure 3.8 shows the column density distribution along a line $y = \mathrm{const}$ cutting through the exact location of the source. Since for this special case no interpolation is necessary, only very small differences between the two methods are found. These differences are due to uneven sampling of the $1/r^2$ density distribution on the adaptive mesh. The errors increase only slightly ($< 0.5\%$) when interpolation is used, as indicated by the results obtained along the $y = \mathrm{const}$ line located at $3/4$ of the horizontal extent of the domain (the right panel in Figure 3.8).

3.4.2 Shadow casting

To test the shadow casting capabilities of our algorithm, we calculate the column density and ionization fractions for a homogeneous environment containing higher density clumps, which are taken to be spherical and neutral. The ionization state is found by iterating the ionization fractions over a period equal to a few recombination time scales, while keeping the temperature fixed.

The computational domain spans the region $(2.0, 1.0, 1.0) \times 10^{18}$ cm. The ambient medium has a number density $n_{\mathrm{env}} = 10^2$ cm^{-3} and a temperature $T_{\mathrm{env}} = 5000$ K. The source of ionizing radiation is located at $(x, y, z) = (0.0, 0.5, 0.5) \times 10^{18}$ cm. It has a luminosity $L_{\mathrm{S}} = 7000\ L_{\odot}$ and an effective temperature $T_{\mathrm{eff}} = 50000$ K. The resulting
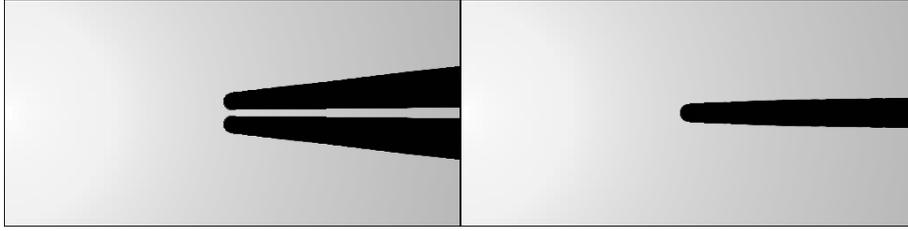
Figure 3.9: Values of $\log_{10}$ of the HII ionization fraction for the case of a single point source in an environment with a homogeneous density distribution containing neutral clumps with higher density. Shown are plots of a xy-cut through the centre of the domain (left) and a xz-cut through the centre of the bottom clump (right).

Strömgren sphere has a radius that is larger ($\sim 3 \times 10^{18}$ cm) than the physical size of the computational domain. Two identical clumps are placed at a distance of $\sim 10^{18}$ cm from the source. Each clump has a density $n_{\mathrm{clump}} = 10^4$ cm^{-3}, a temperature $T_{\mathrm{clump}} = 100$ K, and a radius $r_{\mathrm{clump}} = 4 \times 10^{16}$ cm. We used 6 levels of refinement with patches of 16^3 cells. The effective resolution in this test was 1024×512^2 cells.

The results of the shadow casting test are shown in Figure 3.9. As one can see, our hybrid characteristics method is capable of casting shadows with very sharp boundaries, indicating a low numerical diffusivity of the scheme. We note that since the initial conditions do not contain any density gradient, column densities calculated in this test are identical to the ones one would obtain using a long characteristics method.

3.4.3 Application: photoevaporating clumps

To illustrate that our hybrid radiative transfer algorithm can be used efficiently in combination with hydrodynamics, we present a first 3D application of the evolution of over-dense clumps being photoevaporated. We use the parameters of the simulation setup described in Section 3.4.2 as initial conditions and follow the dynamical evolution for ~ 4000 yr. This simulation is similar to the ones presented by Lim & Mellema (2003), with this difference that in our simulation both the source and the clumps are inside the computational domain, and that our radiation field is not approximated by parallel rays.

These computations are relevant to the shaping and evolution of cometary knots which are observed in objects like the Helix (NGC 7293), Eskimo (NGC 2392), and Dumbbell (M27) nebula. Another application is the interaction zone that is observed between binary proplyds in HII regions like NGC 3603 (Brandner et al. 2000) and the Orion Nebula (Graham et al. 2002).

Figure 3.10 shows a sequence of snapshots of the density and neutral hydrogen fraction at different times during the simulation. One sees that the interaction of the photoevaporation flows coming from the clumps results in a zone of higher density between the clumps, which, as was already found by Lim & Mellema (2003), can explain the excess emission observed between some cometary knots and binary proplyds. This interaction zone recombines, becomes optically thick, and casts a shadow. It is interesting to see that the zone, and the shadow region behind it, persist even after the two clumps have been
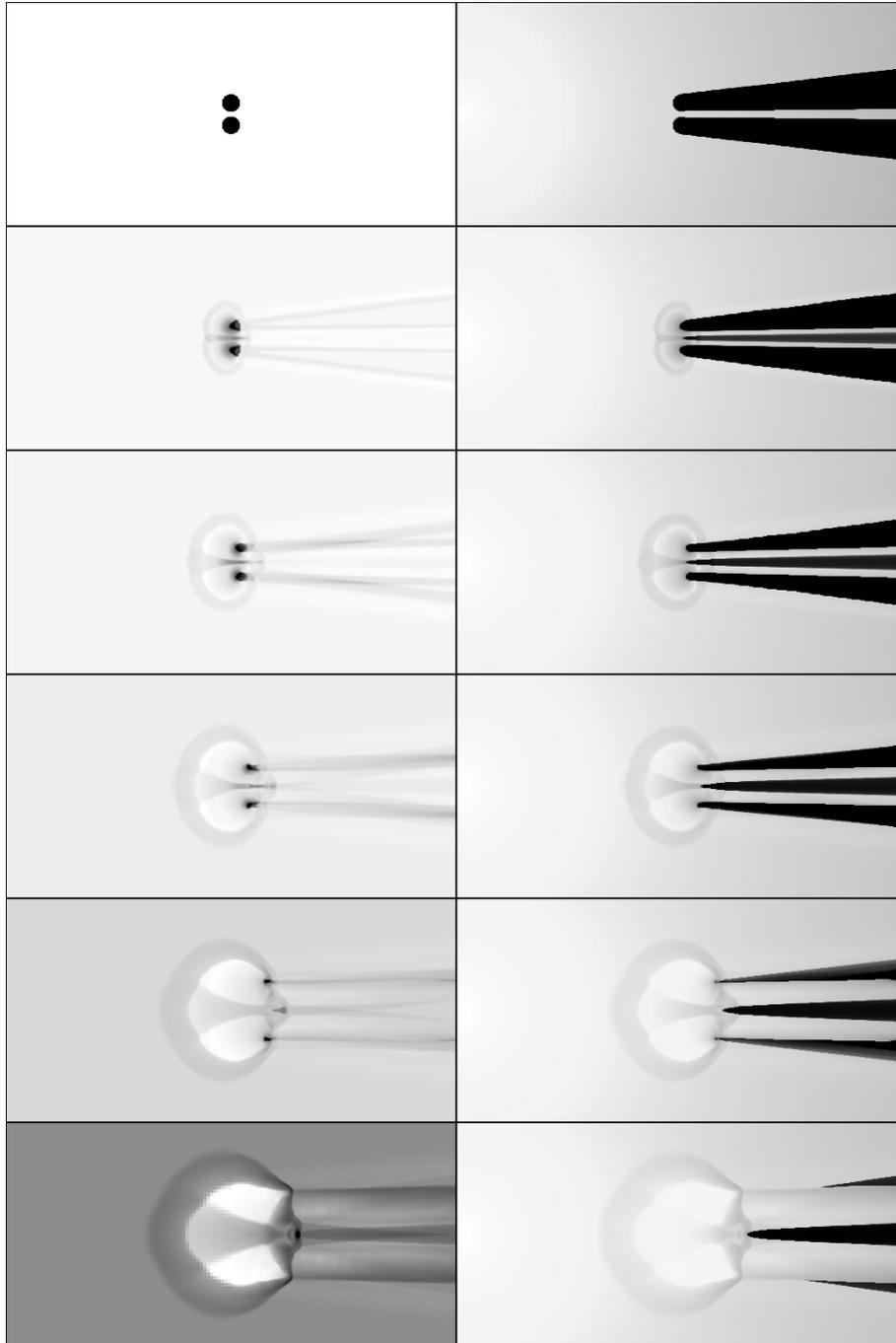
Figure 3.10: Snapshots of the evolution of the $\log_{10}$ of the mass density (left) and the $\log_{10}$ of the HII ionization fraction (right) for the case of a single point source in an environment with a homogeneous density distribution containing two neutral clumps with higher density. The source is located at $(0.0, 0.5, 0.5) \times 10^{18}$ cm. Shown are plots of xy-cuts through the centre of the domain at $t = 0$ yr (first row), $t = 792$ yr (second row), $t = 1584$ yr (third row), $t = 2377$ yr (fourth row), $t = 3169$ yr (fifth row), and $t = 3961$ yr (sixth row).

fully evaporated. This mechanism for creating extra shadows may influence the evolution and survival time of clumps that lie farther away from the star, an effect not taken into account in previous numerical studies. We are currently investigating further into this kind of flows and will present our findings in a forthcoming publication.

3.5 Performance analysis

We start this section by comparing our hybrid characteristics method to the more traditional long and short characteristics methods for regular grids. In order to do this, we discern two types of computations: first we determine how many calculations are needed to arrive at the local contribution each cell makes to the column density along a ray, and second we look at the number of interpolations the different methods have to perform to compute the total column density up to each cell.

Consider a computational domain with a resolution of C^3 cells and a source located at one of the corners of the domain. For the case of a regular grid, the maximum number of cells a long characteristic would encounter is $\sqrt{3}C$, and, since we assume that a ray is cast to all cells, the number of calculations needed to provide the total column density is therefore $\lesssim C^4$.

For our hybrid characteristics method, which employs an oct-tree type of AMR grid, the maximum number of cells a local ray section encounters is $\sqrt{3}c$, where c^3 is the number of cells in a single patch (cf. Section 3.2.2). So in this case, for a fully refined grid, the total number of calculations would amount to $\lesssim c\,C^3$. But, since in general the domain would be refined by a factor r, with $0 \leq r \leq 1$, this number reduces to $\lesssim r\,c\,C^3$. This means that when $r\,c \simeq 1$, our method needs $\sim C^3$ calculations to arrive at the local contributions to the column density for each cell, which is of the same order a short characteristics method would need on a regular grid. However, the local values of column density still need to be communicated and interpolated to arrive at the total column density for each cell. On the other hand, a short characteristics method also needs to interpolate local values when it sweeps through the grid, whereas a long characteristics method, although it executes a factor C more calculations, does not need to perform any interpolations at all.

The number of interpolations to be performed by our method is determined by the number of patches that are encountered when ray tracing through the patch-mapping (cf. Section 3.2.3 and 3.2.3). This number is at most $\sqrt{3}C/c$, since, for a fully refined grid, there are C/c patches along a coordinate axis. For a grid that is not fully refined this number is again reduced by a factor r. A ray trace through the patch-mapping is to be performed for every cell, which brings the total number of interpolations to $\lesssim r^2(C/c)C^3$, where one factor of r comes from the number of patches cut by a ray, whereas the other factor comes from the total number of cells that exist in the computational domain.

A short characteristics method needs to do an interpolation for every cell, so, for a regular grid, the total number of interpolations is C^3. This implies that when $r^2C/c \simeq 1$, our method needs to compute a similar number of interpolations as a short characteristics one. Note that we assume that the calculations needed to do the interpolations are comparable in execution speed for the short and hybrid characteristics methods, which may
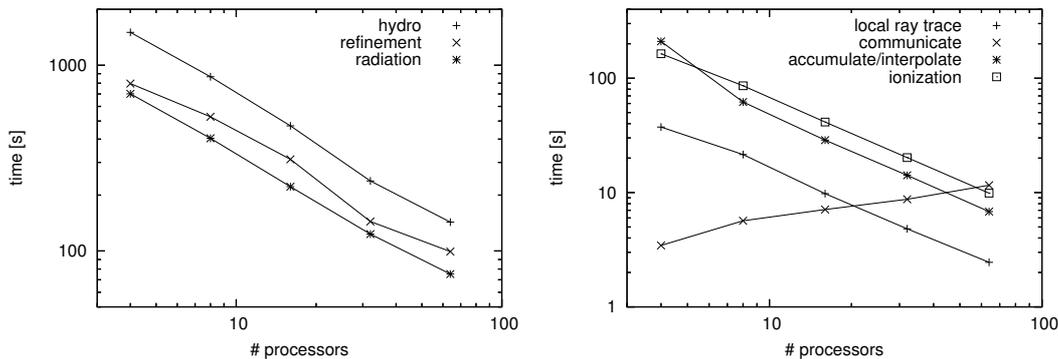
Figure 3.11: Performance of the main components of a radiation hydrodynamics calculation (left), and performance of the different steps in our hybrid characteristics method (right). For this specific test, the communication takes as much time as the rest of the calculation when using 64 processors. As is explained in the text, for patches with a larger number of cells, this constraint may become less severe.

actually not be the case.

As an example, a typical AMR calculation has $C = 512$, $c = 16$ (i.e. 6 levels of refinement), and $r = 0.25$, which results in $rc = 4$ and $r^2 C/c = 2$. This shows that, for a single processor calculation with a proper choice of the ratio C/c and a reasonable amount of refinement, our hybrid characteristics method is expected to perform equally well as a short characteristics method on a regular grid. It also means that, when our method is used in parallel, a better performance will be obtained when increasing the number of processors.

To investigate this aspect in some more detail we have conducted a preliminary performance analysis using the photoevaporating clumps test case described in Section 3.4.3 as the underlying physics problem. We used 5 levels of refinement irrespectively of the number of processor used in the test run (i.e., the problem had a fixed total work). Calculations have been terminated after reaching 10% of the nominal simulation time. Otherwise the simulation parameters were identical to those used in the calculations presented in Section 3.4.3.

The results of our performance study are shown in Figure 3.11, where we compare the overall performance of the hydrodynamics, AMR, and radiation calculations. Detailed results for the radiation part are also presented in Figure 3.11, where the timings for the individual components of our hybrid characteristics method (local ray trace, communication, accumulation/interpolation, and ionization) are shown.

Performance data obtained for our realistic test problem indicates that the ray tracing part of the calculation takes less time than either the hydrodynamics or grid adaptation. Furthermore, it shows that most of the computational time during ray tracing is spent in interpolating and accumulating the local contributions to the column density (i.e. step 6, Section 3.2.4). Following the analytical assessment made above, we conclude that in order to reduce the number of interpolations required during calculation one should try to minimize the value of $r^2 C/c$ rather than rc when setting up a simulation. This suggest that one should use patches that contain a relatively large number of cells compared to the

effective resolution of the computational domain, and, of course, keep the filling factor of the finest AMR level at a minimum.

Figure 3.11 shows performance results for the radiation module including the ionization package for our fixed size problem. As one can see, the time required to calculate the column densities is about the same as the time needed to calculate the ionization state of the gas. Furthermore, both these calculations are local and therefore perform very well. On the other hand, we notice that the communication part of our algorithm does not perform perfectly. This is somewhat expected since, with the increasing number of processors, the efficiency of our algorithm becomes limited by the efficiency of the global gather operation (used to collect column densities from patch faces). The results for this specific test indicate that communication is likely to dominate the runtime when more than ~ 64 processors participate in the computations. We expect that this limitation becomes less severe when larger patches are used in the simulation. In this case the cost of communication may still be lower than, for example, the time needed to accumulate and interpolate the column densities. To determine whether this is indeed the case, more elaborate performance tests involving a larger number of processors (of the order of ~ 1000) are required, and such tests are currently underway.

3.6   Conclusions

We described a new radiative transfer algorithm for parallel AMR hydrodynamics codes, called hybrid characteristics. We presented details of several aspects of the algorithm: ray tracing, communication, and interpolation.

The ray tracing is performed in two steps. First, local long characteristics are used to calculate column density contributions for each patch. A second ray trace is then performed where a so called patch-mapping is used to find the patches cut by each ray. When the list of patches cut by a ray is known, interpolation of local column density values is required to find the total column density up to each cell. For this, one needs the values of local column density contribution at patch faces, which are communicated to all processors. The coefficients used in the interpolation are chosen such that the exact solution for the column density is retrieved when there are no gradients in the density distribution.

For the case where the distribution is not constant but has a $1/r^2$ profile we find deviations of the order of $\sim 0.5\%$ when comparing our method with a long characteristics one. This high accuracy with which column densities values are calculated results in well defined and sharp shadows.

We showed that our method can be used efficiently for parallel radiation hydrodynamics calculations in three dimensions on AMR grids. We presented preliminary results for our new method in application to the problem of the photoevaporation of two over-dense clumps due to the ionization by a single source of radiation. The results of this simulation offer a possible explanation for the excess emission observed in between cometary knots seen in for example the Helix Nebula, and the interaction zone observed in binary proplyds found in HII regions like NGC 3603 and the Orion Nebula. These simulations also suggest a possible mechanism for the creation of extra shadows by the high density

interaction zone forming between the clumps. This additional shadowing may influence the evolution and survival time of clumps that lie farther away from the source. We are currently investigating further into this kind of interactions between photoevaporating flows, and their consequences for the dynamics, and will report our findings in a future publication.

An initial performance test showed that our method works very well when used for calculations on a parallel machine. For this specific test, the communication part of our algorithm starts to dominate the calculation when more than ~ 64 processors are used. However, we showed analytically that a careful choice of the ratio of the number of cells per patch to the total number of cells in the computational domain controls the amount of communication used in the calculation. This analysis can be used to optimize the design of our method. More in-depth performance and scaling studies are currently underway, using large (~ 1000) number of processors, and these will also be used to further optimize the current implementation.

Because of the modular nature of the *Flash* code and the *Doric* routines, additional elements like more sophisticated cooling or multiple species can easily be added. Also, multiple point sources can be handled by our method, and in principle moving sources could be implemented. Furthermore, it should be straightforward to extend the hybrid characteristics method so that it can be used to solve for a more general radiation field, with the added advantage that our method is already parallelized and coupled to an AMR hydrodynamics code.

Another possible application for our method is the calculation of the propagation of ionization fronts in a cosmological context. For these calculations photon conservation is an important issue. Recently, Mellema et al. (2005) developed a method for following R-type ionization fronts that may move more than one cell per time step, where a special formulation of the equations ensures photon conservation. Although the parallel nature of our algorithm may complicate the implementation of such an approach, we may still benefit from the ideas presented by Mellema et al. (2005).

We intend to make our method publicly available in a future *Flash* release.

Acknowledgements

Figure 3.12: Explanation of different quantities used in the fast voxel traversal method. Shown is a single patch with a local long characteristic ray section.

3.A   Fast voxel traversal

Here we briefly discuss the 'fast voxel traversal algorithm' from Amanatides & Woo (1987). We have used this algorithm twice in our method, once to ray trace through the cells ('voxels') of a single patch (local long characteristics, Section 3.2.2), and once to ray trace the patch-mapping (hybrid characteristics, Section 3.2.3). The idea behind the algorithm is to keep track of three different ray parameters t_x, t_y, and t_z, one for each co-ordinate direction, and to use these to determine how to step from cell to cell through the patch, ensuring that all cells cut by the ray are visited (see Figure 3.12). First, values for the increments in ray parameter t needed to step from cell to cell in the x-, y-, and z-direction, indicated by δt_x, δt_y, and δt_z, respectively, are determined:

$$\delta t_x = t_{\max}/\Delta x \;, \delta t_y = t_{\max}/\Delta y \;, \delta t_z = t_{\max}/\Delta z \;, \tag{3.13}$$

where $t_{\max} = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ is the final ray parameter (i.e. the total path length of the ray). Next, the ray parameters t_x, t_y, and t_z are set to their respective initial values, indicated by t_x^{i}, t_y^{i}, and t_z^{i}, after which a loop is entered where the *minimum* of these three values is determined. This gives the co-ordinate direction in which the cell lies that is to be visited next by the ray. For example, if $\min(t_x, t_y, t_z) = t_x$, the next cell the ray will enter lies in the x-direction, and we have to increment the ray parameter for the x-direction accordingly, i.e. $t_x = t_x + \delta t_x$. We loop as long as all ray parameters are smaller than
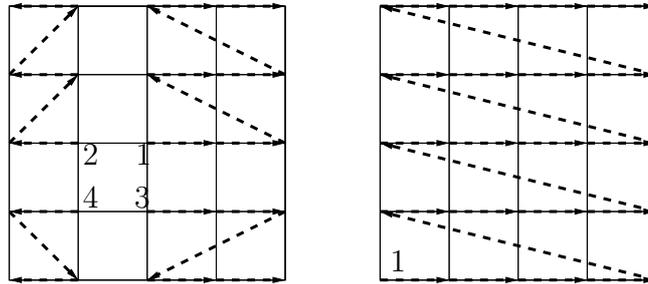
Figure 3.13: Two examples of short characteristics sweeping sequences for a single patch that may occur in practice. For the illustration on the left, the source is located inside the patch at the starting point of curve 1. In this case the four space filling curves shown should be swept in the order indicated. For the patch on the right, the source is external to the patch, and lies in the direction of the lower left corner, so there is only one curve that needs to be swept.

the final ray parameter $t_{\max}$. As a by-product, the algorithm produces the path length of the ray section for each cell that is crossed, which is obtained by subtracting the previous from the current ray parameter.

3.B   Ray tracing a single patch: short characteristics

As an alternative to the 'fast voxel traversal algorithm' for ray tracing a single patch as presented in Section 3.2.2, we here briefly describe the short characteristics method, which could be used for the same purpose. Since the method of short characteristics uses interpolation from neighbouring cells, upwind values need to be available at all times, so cells need to be swept in a certain order. This sweeping sequence is determined by the physical location of the patch relative to the source position (see Figure 3.13). Using the known physical location of the source, the geometrical path length of the ray section that crosses a cell is calculated for every cell contained in the patch. The short characteristics method then sweeps the patch in a direction away from the source, interpolating upwind column density contributions for each cell along the way.

For the two-dimensional case, Figure 3.14 illustrates which two cells, indicated by c_1 and c_2, are used in this interpolation. Simple linear interpolation weights

$$w_1 = 1 - d; \; w_2 = d \qquad (3.14)$$

could be used to arrive at the column density contribution at cell c, using

$$\Delta N_c = \sum_i w_i \Delta N_i + \Delta r \, n \,, \qquad (3.15)$$

with ΔN_i the upwind values of column density that need to be interpolated, d the normalized distance from c_1 to the location where the ray pierces the line connecting c_1 and
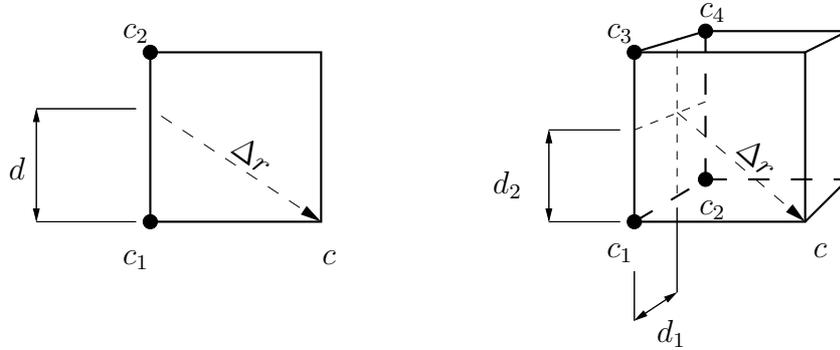
Figure 3.14: Illustration of the interpolation scheme for a single cell used in the short characteristics method for the 2D (left) and 3D (right) case.

c_2, Δr the physical path length of the short characteristic ray section, and n the number density inside the cell crossed by this short characteristic.

For the three-dimensional case the ray pierces a cell face, so four instead of two quantities need to be interpolated. The normalized weights are chosen to correspond to the partial areas of the cell face defined by the corners of this face and the location at which the ray leaves the cell (cf. Figure 3.14):

$$
\begin{aligned}
w_1 &= (1 - d_1)(1 - d_2); & w_2 &= d_1\,(1 - d_2); \\
w_3 &= (1 - d_1)\,d_2; & w_4 &= d_1\,d_2\,.
\end{aligned}
\tag{3.16}
$$

CHAPTER 4

Scaling tests

We present weak and strong scaling tests for the Hybrid Characteristics method as presented in Chapter 3. This method introduces a novel approach to ray tracing for 3D, massively parallel, adaptive mesh refinement hydrodynamics codes. The effects of the location of the source, location and amount of refinement, number of sources, and machine architecture on the performance are investigated. Timing results for the calculation, communication, and synchronization parts of the algorithm are obtained. For the calculation part we find the method to scale almost perfectly, for the case of weak, as well as for strong scaling. However, due to the imbalance in the calculation time inherent in the approach to ray tracing of the Hybrid Characteristics method, for the case of a single source, almost half the time is spent in synchronization. The distribution of this time over the different processors strongly depends on the location of the source in the computational domain. We also find that, when the amount of work per processor is chosen large enough, the communication time stays below the calculation time. Furthermore, the performance of the method is found to be to a great extent independent of the relative location of the source and refined region, as well as of the amount of refinement. As expected, we find that the calculation part of the method scales linearly with the number of sources, but that the synchronization time averages out to a relatively low value in this case. Tests for different architectures show that the method can be used effectively on at least ~ 100 processors on a SGI Altix, and ~ 1000 processors on an IBM BlueGene/L system, bringing massively parallel radiation hydrodynamical simulations within reach.

4.1   Introduction

An assessment of the parallel performance of the Hybrid Characteristics method presented in Chapter 3 is made. To this end, we perform weak scaling tests, where the amount of work per processor is kept constant when increasing the number of processors, as well as strong scaling tests, where the total amount of work is kept constant when increasing the number of processors.

Since the Hybrid Characteristics algorithm consists of two main parts, i.e. calculation and communication, these are looked into separately. The preliminary performance test presented in Chapter 3 indicate that for a certain number of processors the time spent in communication becomes similar to the time spent in the calculation part of the algorithm. We show however that this can to a large extent be avoided by carefully choosing the amount of work per processor.

It is expected that the performance of the method changes with the location of the source in the computational domain. Processors that own patches located close to the source have less contributions to add up than processors that own patches located further away from the source. Furthermore, when the source is located at the edge of the domain, this imbalance may be greater than for a source located at the centre, although this depends on how the patches are distributed among processors. Therefore a measure for the synchronization time during which processors are idle is obtained for different source locations.

It is also expected that the method scales linearly with the number of sources present. We test this by increasing the number of sources, which are randomly distributed throughout the computational domain, while keeping all other parameters constant.

The amount by which the domain is refined may influence the performance as well. This is tested by comparing two cases for which a small and a large amount of the computational domain is refined.

Another aspect that is of importance for the performance is the type of architecture used in the tests. To determine how this choice influences the results, we ran part of the scaling tests on two different architectures: a SGI Altix, and an IBM BlueGene/L system.

As was already mentioned, the communication part of the algorithm may become a bottleneck at a certain number of processors. The dependence of this aspect on the number of cells per patch, and the choice of architecture is examined.

We finish this chapter with recommendations on how to optimize the ray tracing and communication algorithms of the Hybrid Characteristics method.

4.2   Weak and strong scaling tests

In this section we present results from weak and strong scaling tests carried out for the Hybrid Characteristics method. All tests in this section were performed on a SGI Altix 3700 system, consisting of Intel Itanium 2 1.3 GHz CPUs with 2GB of memory per node.

As was mentioned above, for a weak scaling test, the amount of work per processor should be kept constant for an increasing number of processors. For example, for a hydrodynamics algorithm, where the amount of work per cell is constant, this would mean that

Table 4.1: Parameters used in the scaling tests on SGI Altix.

N_{pe}	2	4	8	16	32	64
P ($c = 8$)	32	40	50	64	80	102
c ($P = 32$)	8	10	12	16	20	26

one should increase the number of cells by the same amount with which one increases the number of processors N_{pe}.

However, for a ray tracing algorithm like the Hybrid Characteristics method, the work per ray is not constant since it also depends on the extent of the ray over the computational domain. In the next section we estimate the impact of this aspect, although for the actual tests we simply scale the number of cells linearly with the number of processors.

4.2.1 Amount of work estimate

In Chapter 3 it was found that accumulating and interpolating the local column density contributions dominates the calculation part of the method. It was also found that the amount of work does not only depend on the number of rays, but on the number of patches *along* each ray as well, with the total number of interpolations for a fully refined domain being proportional to $(C/c)C^3$. Here, C is the number of cells along a coordinate axis of the domain, and c is the number of cells along a coordinate axis contained in a single patch. A weak scaling test for the calculation of column densities should therefore formally satisfy the condition $(C/c)C^3 \propto N_{\mathrm{pe}}$, instead of $C^3 \propto N_{\mathrm{pe}}$, which is normally used in this type of test.

In practice this means that if one increases N_{pe} by a factor of f, one should also increase $(C/c)C^3$ by that same factor. Since $(C/c)C^3 = P(Pc)^3 = P^4 c^3$, with P the number of patches along a coordinate axis, one way of doing this is to increase P^4 by a factor of f while keeping c constant. Another option is to increase c^3 by a factor f while keeping P constant. Also, one could combine the two: change P^4 by a factor f_P and c^3 by a factor f_c, with the constraint that $f = f_P f_c$.

For the case of a partially refined domain, the condition for weak scaling changes to $r^2(C/c)C^3 \propto N_{\mathrm{pe}}$, where r is the ratio between the number of cells in leaf patches divided by the maximum number of cells possible at the highest refinement level, i.e. $0 < r \leq 1$ (see also Section 3.5). For the tests presented in this chapter we keep the amount of refinement fixed by forcing refinement in a certain region of the computational domain, while varying c or P as a function of N_{pe}.

4.2.2 Test description

Two types of weak scaling tests are set up, details of which are listed in Table 4.1. In one series, c is being varied as $N_{\mathrm{pe}}^{1/3}$ and P is kept fixed when increasing the number of processors. We also present a test where P is set proportional to $N_{\mathrm{pe}}^{1/3}$ and c is kept constant. Note that, to stay in line with the procedure normally followed for this type of tests, we chose the values for c and P such that the number of cells per processor stays
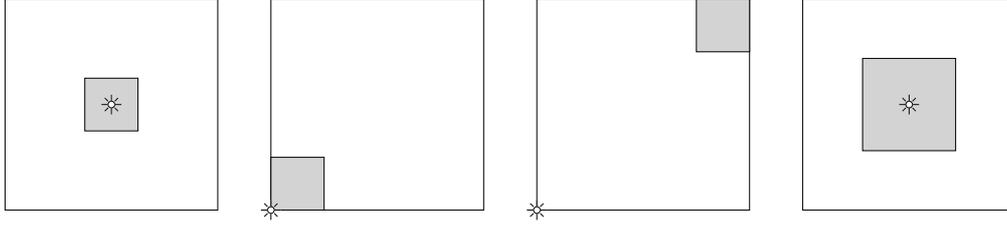
Figure 4.1: Schematic depiction of the four different setups used for the scaling tests. Shown are (left to right): source and refined region (grey) at the centre of the domain, source and the refined region at a corner of the domain, source and refined region at opposite sides of the domain, and a setup with twice the amount of refinement.

(approximately) constant, although the estimate presented in the previous section suggest that when P is varied it should actually be done proportional to $N_{\text{pe}}^{1/4}$.

Two different source locations are used in the tests: one with the source at the centre of the domain, and one with the source at a corner of the domain, where the domain is a cube with sides of unit size. In both these cases, the refined region is centred on the source. Furthermore, a test where the source and the refined region are located at opposite corners of the domain is also carried out. The different test setups are schematically depicted in Figure 4.1.

All calculations have a fixed number of six levels of refinement, including the base level. Two different values for the amount of refinement are used: $r = 1/64$ and $r = 1/8$, corresponding to a cubical region of the computational domain with sides of size $1/4$ and $1/2$, respectively.

4.2.3 Weak scaling results

Timing results for the weak scaling tests where $c \propto N_{\text{pe}}^{1/3}$ and P is kept constant are shown in Figure 4.2. We first discuss the case where the source as well as the refinement are located at the centre of the computational domain (Figure 4.2, top row), in terms of timing results obtained for the different parts of the algorithm:

- accumulate-interpolate (solid line): approximately flat curve. This is expected since the number of patches P along a ray is kept constant for an increasing number of processors and the total number of rays is kept approximately constant by increasing c. The error bars get larger with increasing number of processors due to an increasing imbalance in the number of rays that need to be dealt with per processor.

- local ray trace (dashed line): slight increase. This is expected since the number of operations for local ray tracing of the patches scales as c^4, whereas the number of processors only increases as $N_{\text{pe}} \propto c^3$.

- synchronization time (dashed-dotted line): increases when the imbalance in the accumulate-interpolate part sets in at around 16 processors.
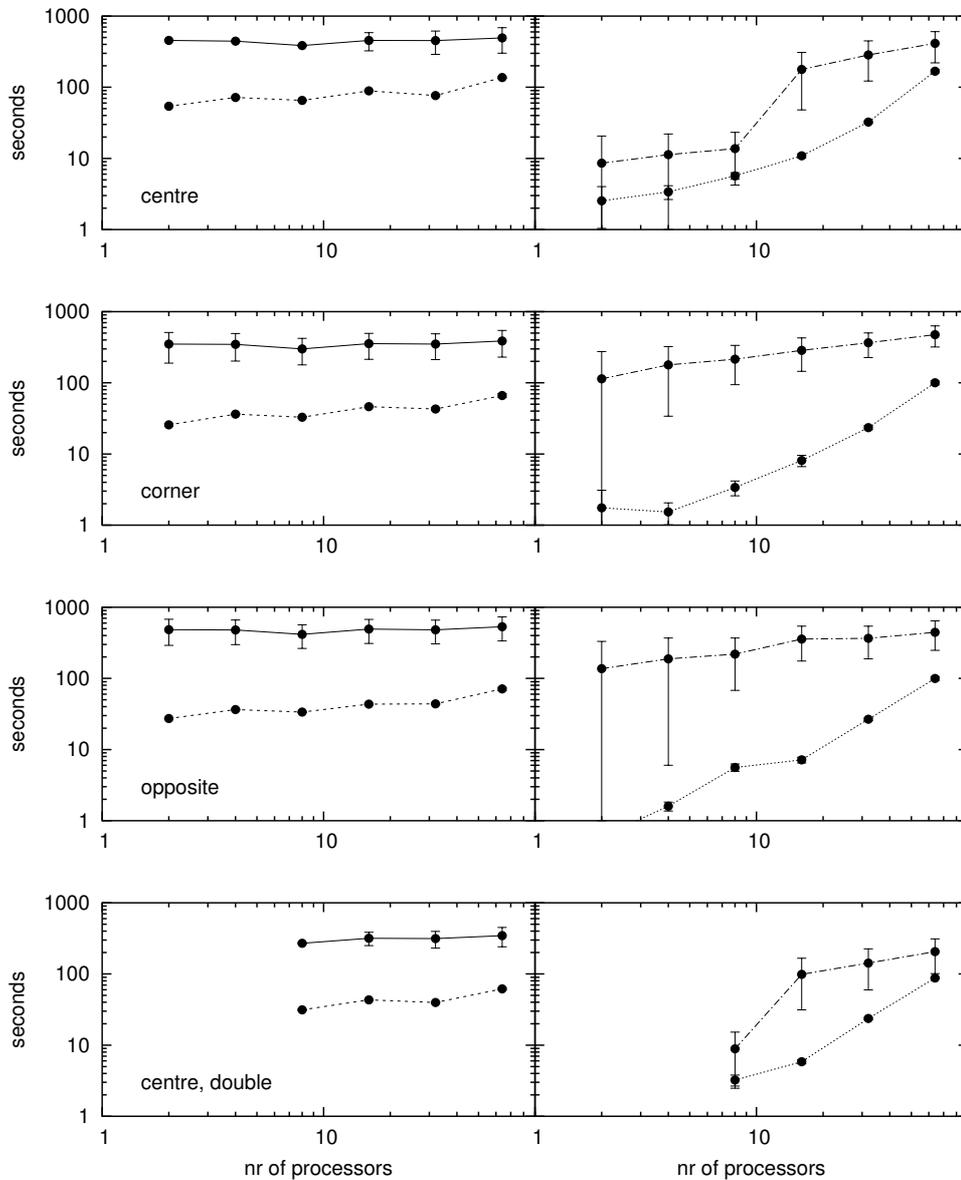
Figure 4.2: Weak scaling results for tests where $c \propto N_{\mathrm{pe}}^{1/3}$ and P constant. Results for a single source at the centre (first row) and corner (second row) of the computational domain are shown, as well as results where the source and refined region are located at opposite corners (third row), and a test with the source at the centre but with twice the amount of refinement (fourth row). The line types represent different parts of the algorithm: accumulate-interpolate (solid), local ray trace (dashed), synchronization (dashed-dotted), and communication (dotted), respectively. The size of the errorbars is twice the standard deviation determined from the different timing results obtained across processors. The values for the test where the amount of refinement is doubled (r=1/8, bottom row) have been scaled to the one with a lower amount of refinement (r=1/64, top row).
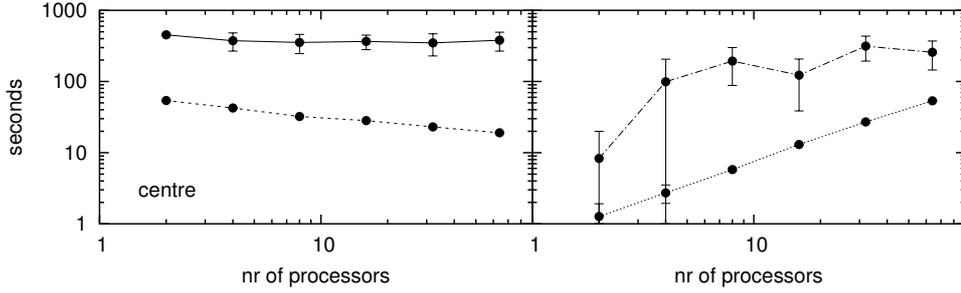
Figure 4.3: Weak scaling results for tests where $P \propto N_{\mathrm{pe}}^{1/3}$ and c constant. Results for a test with a single source at the centre of the computational domain are shown. The line types represent different parts of the algorithm: accumulate-interpolate (solid), local ray trace (dashed), synchronization (dashed-dotted), and communication (dotted), respectively. The size of the errorbars is twice the standard deviation determined from the different timing results obtained across processors.
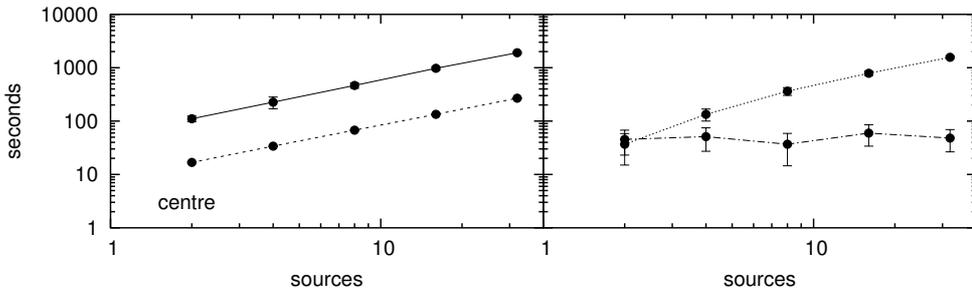

Figure 4.4: Weak scaling results for an increasing number of sources for $c = 16$ and $P = 32$. All tests use 16 processors with sources distributed randomly throughout the computational domain. The line types represent different parts of the algorithm: accumulate-interpolate (solid), local ray trace (dashed), synchronization (dashed-dotted), and communication (dotted), respectively. The size of the errorbars is twice the standard deviation determined from the different timing results obtained across processors.

- communication (dotted line): increases substantially for an increasing number of processors, but for 64 processors is still a factor ~ 5 smaller than the time needed by accumulate-interpolate.

We compare these results to ones obtained with the source and refinement in the same (Figure 4.2, second row), and in opposite (Figure 4.2, third row) corners of the domain, as well as to a test where the amount of refinement is doubled (Figure 4.2, fourth row). When the source is located in a corner of the computational domain, the imbalance in the number of rays per processors sets in immediately, which is evident from the size of the errorbars of the accumulate-interpolate part, and from the larger values for the synchronization part of the algorithm over the whole range of processors. Since the adaptive mesh needs a slightly lower number of patches to create the refined region in the corner of the domain as compared to the number of patches needed when the refined region lies at the centre of the domain, the values for local ray trace and communication are slightly lower in the
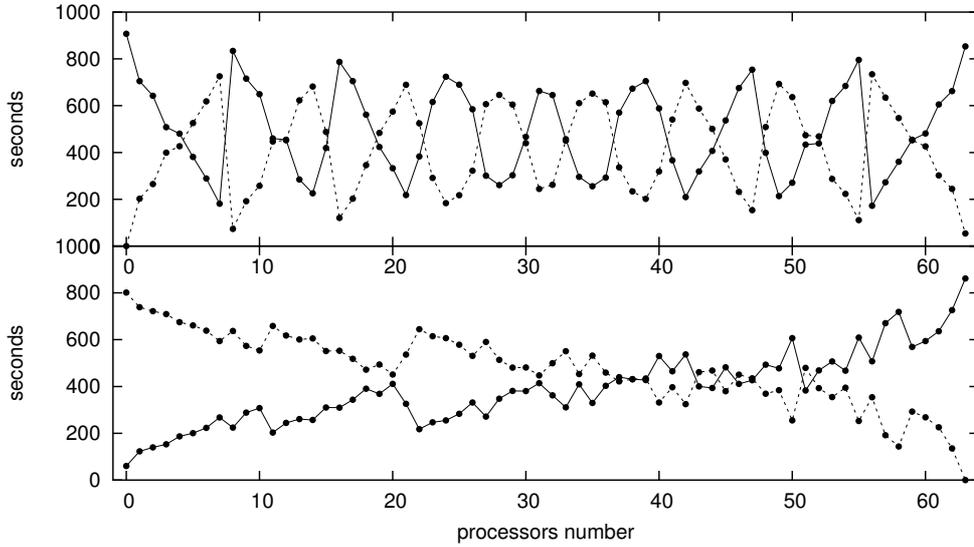
Figure 4.5: Timing results for individual processors for the accumulate-interpolate (solid) and synchronization (dashed) parts of the algorithm. Shown are calculations involving 64 processors with the source at the centre (top panel) and corner (bottom panel) of the computational domain.

former case.

Comparing the results of the test with the source at the centre of the domain with $r = 1/64$ (Figure 4.2, top row), to a test where $r = 1/8$ (Figure 4.2, bottom row, scaled values), we find similar behaviour in all aspects.

The results of a weak scaling test with the source at the centre for which $P \propto N_{\mathrm{pe}}^{1/3}$ and c is kept constant are shown in Figure 4.3. Since the number of operations for the accumulate-interpolate part of the algorithm increases as P^4, but the number of processors only increases as $N_{\mathrm{pe}} \propto P^3$, we expect a slight incline in the plot of this quantity. We also expect the curve for the local ray trace part of the algorithm to be flat, since c is kept constant for the whole range of processors in this test. Instead, the results show a flat curve for the accumulate-interpolate, and a slight decline for the local ray trace part of the algorithm. To understand this, we need to take a closer look at the actual number of patches the AMR algorithm generates at levels below the highest level of refinement to accommodate the central refined region. It turns out that this number is considerable for low values of P ($\sim 64\%$ of the total number of patches for $P = 32$), and becomes less important as P increases ($\sim 7\%$ for $P = 102$). Correcting for this would change the timings to the expected values.

We also test the performance of the algorithm for an increasing number of sources, distributed randomly throughout the domain. For this we use the test for which $c = 16$ and $P = 32$, see Figure 4.4. As expected, we find that the algorithm scales linearly with the number of sources, with the exception of the synchronization time, which stays at an approximately constant and rather low value. This is due to the random distribution of sources which averages out extremes in the synchronization time.
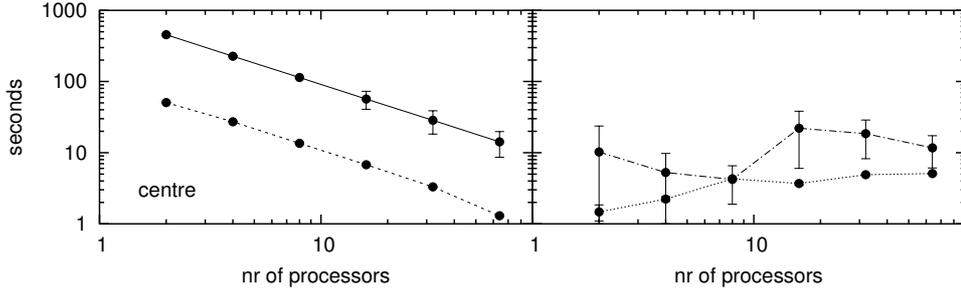
Figure 4.6: Strong scaling test results with $c = 8$, $P = 32$, and $r = 1/64$. Results where source and refined region are located at the centre of the computational domain are shown. The line types represent different parts of the algorithm: accumulate-interpolate (solid), local ray trace (dashed), synchronization (dashed-dotted), and communication (dotted), respectively. The size of the errorbars is twice the standard deviation determined from the different timing results obtained across processors.

Table 4.2: Parameters used for the scaling test on IBM BlueGene/L.

N_{pe}	64	128	256	512	1024
$c\,(P = 32)$	8	10	12	16	20

We conclude this section with timing results for individual processors for the test with $c = 26$ and $P = 32$ for a total of 64 processors (see Figure 4.5). We plot the results for the accumulate-interpolate and the synchronization parts of the algorithm, for the case with the source at the centre, and at the corner of the computational domain. Figure 4.5 clearly shows that the amount of work to be done in accumulating and interpolating, and the time spent waiting in synchronization, varies considerably between individual processors, and that the distribution of work strongly depends on the source location.

4.2.4   Strong scaling results

Strong scaling test results for $c = 8$ and $P = 32$ are shown in Figure 4.6. We find a linear decrease for the calculation parts of the algorithm, and a fluctuating synchronization time. The latter is caused by the different ways in which the patches get distributed when increasing the number of processors. The communication time gets larger when increasing the number of processors, still being a factor ~ 2 lower than the time spent in accumulate-interpolate for 64 processors.

4.3   Performance for different architectures

In this section we compare the results described above, which were obtained on an SGI Altix system, to results from a preliminary test performed on an IBM BlueGene/L system. This system consists of PowerPC 440 700 MHz CPUs with 512MB of memory per node.

We repeat the weak scaling test where c is being varied as $N_{\mathrm{pe}}^{1/3}$ (for details, see Table 4.2). The source and refinement are located at the centre of the computational domain,
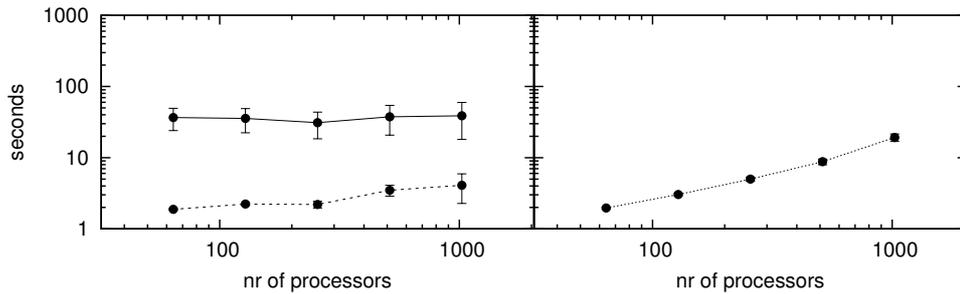
Figure 4.7: Weak scaling results for a test where $c \propto N_{\mathrm{pe}}^{1/3}$ and P is constant on an IBM BlueGene/L system. Results for a single source at the centre of the computational domain are shown. The line types represent different parts of the algorithm: accumulate-interpolate (solid), local ray trace (dashed), and communication (dotted), respectively. Timings results for the synchronization part of the algorithm were not obtained. The size of the errorbars is twice the standard deviation determined from the different timing results obtained across processors.

with the amount of refinement set to $r = 1/64$. When comparing Table 4.1 to Table 4.2, one can see that we use the same sequence for c, while the number of processors was increased by a factor of 32.

Timing results for this scaling test are shown in Figure 4.7. We find an approximately flat curve for the accumulate-interpolate part, and a slight increase for the local ray trace part of the algorithm. The communication part is seen to increase substantially for an increasing number of processors, but for 1024 processors it is still a factor ~ 2 smaller than the time needed by accumulate-interpolate.

A direct comparison between results obtained on SGI Altix and IBM BlueGene/L can be made by taking into account the increase of the number of processors by a factor of 32. When multiplying the timing results of BlueGene/L by this factor, we find that the accumulate-interpolate part of the algorithm is 2.7 times faster on SGI Altix than on IBM BlueGene/L. We can not compare the communication part of the algorithm in this way since it is expected to scale non-linearly with the number of processors.

4.4 Discussion and conclusions

We have performed weak and strong scaling tests for the Hybrid Characteristics method for a number of different setups with varying location of the source and refined region, amount of refinement, and number of sources. These tests were performed on a SGI Altix and an IBM BlueGene/L system.

The results of the weak scaling tests show that the calculation part of the algorithm is dominated by the accumulate-interpolate part, which displays a flat curve for an increasing number of processors. The amount of time spent in this calculation part of the algorithm is to a great extent independent of the relative location of the source and refined region. It also does not depend on the amount of refinement.

Because the amount of work to be done varies from ray to ray, time is spent in syn-

chronization between processors. For 64 processors on SGI Altix, this amount of time is almost equal to the time spent in accumulate-interpolate. A more detailed analysis of this aspect shows that the distribution of work across processors, and therefore the time spent waiting, varies considerably, and that this distribution strongly depends on the location of the source.

Tests involving randomly distributed sources show that the algorithm scales linearly with the number of sources. The synchronization time however averages out to a lower value than in the case of a single source.

For the weak scaling tests the communication time increases with an increasing number of processors, but is still a factor ~ 5 below the time spent in accumulate-interpolate for 64 processors on SGI Altix. For the strong scaling test this factor is ~ 2. This shows that, when there is a large enough amount of work to be done per processor, the communication time can be kept below the time spent in calculation.

A weak scaling test was also performed on an IBM BlueGene/L system. Results show a similar behaviour for the calculation part of the algorithm as on the SGI Altix system. For the accumulate-interpolate part of the algorithm we find the IBM BlueGene/L system to be a factor ~ 2.7 slower than the SGI Altix system. However, for this test, the time spent in communication is still a factor ~ 2 lower than the time spent in calculation for 1024 processors due to the much faster network topology of BlueGene/L. We also note that if one wants to run the same application with the Hybrid Characteristics method, more processing nodes are needed on BlueGene/L than on SGI Altix due to the limited amount of memory per node available on the former system.

We now discuss some possible ways to improving the Hybrid Characteristics algorithm. A way to reduce the synchronization time is to optimize the distribution of patches for the ray tracing. This however will inevitably impair other parts of the code, like the hydrodynamics or ionization calculations. Furthermore, when using more than one source, the patches will have to be redistributed for each source, which is not an option considering the communication overhead this will bring.

In order to reduce the time spent in communication, the amount of data that is communicated between processors should be minimized. In the current implementation, values from all patch faces are communicated to all processors, while in general only a subset of these are needed by the interpolation routines. One complication is that the number of values to be communicated varies from patch to patch since it depends on the way in which a ray cuts through the patches on different processors. We expect an optimized algorithm to reduce the amount of information that needs to be communicated by at least a factor of two.

To conclude, the tests presented for different architectures show that the Hybrid Characteristics algorithm scales well for at least ~ 100 processors on SGI Altix, and ~ 1000 processors on BlueGene/L, bringing massively parallel radiation hydrodynamical simulations within reach.

Acknowledgements

Part II
Astrophysical Applications

CHAPTER 5

Blowing up warped disks in 3D

The Generalized Interacting Stellar Winds model has been very successful in explaining observed cylindrical and bipolar shapes of planetary nebulae. However, many nebulae have a multipolar or point-symmetric shape. Previous two-dimensional calculations showed that these seemingly enigmatic forms can be reproduced by a two-wind model in which the confining disk is warped, as is expected to occur in irradiated disks. In this chapter we present the extension to fully three-dimensional Adaptive Mesh Refinement simulations using the publicly available hydrodynamics package *Flash*. We briefly describe the mechanism leading to a radiation driven warped disk, and give an equation for its shape. We derive time scales related to the disk and compare them to the radiative cooling time scale of the gas, thereby determining the relevant part of parameter space. By comparing two-dimensional calculations including realistic radiative cooling through a cooling curve, with ones employing a low value for the adiabatic index γ, we show that the latter, computationally less expensive approach, is a valid approximation for treating cooling in our nebulae. The results of the fully three-dimensional wind-disk simulations show our mechanism to be capable of producing a plethora of multipolar (and quadrupolar) morphologies, which can explain the observed shape of a number of (proto-)planetary nebulae.

5.1   Introduction

In the final phases of stellar evolution, low mass stars, such as our Sun, first swell up and shed a dense, cool wind in the asymptotic giant branch (AGB) phase. This episode is followed by a fast, tenuous wind that is driven by the exposed stellar core, which will later become a white dwarf. Surprisingly, the planetary nebulae (PNe) resulting from this expulsion phase are rarely spherical. More often they show a pronounced bipolar shape. Balick (1987) proposed that such forms arise due to an interaction between a slow disk-shaped inner AGB nebula and the fast 'last gasp' of the star. Analytical (Icke 1988; Icke et al. 1989) and numerical (Soker & Livio 1989; Icke 1991; Mellema et al. 1991) work showed that this Generalized Interacting Stellar Winds mechanism (GISW) works very well (see Section 1.2 for more details on this mechanism). For an up-to-date review, see Balick & Frank (2002).

However, many circumstellar nebulae have a multipolar or point-symmetric (i.e. antisymmetric) shape (Schwarz 1993; Sahai & Trauger 1998). Icke (2003) demonstrated that these seemingly enigmatic forms can be reproduced by a two-wind model in which the confining disk is *warped*. Such a warp is possible even around a single star, due to the combined effects of irradiation and cooling (Petterson 1977; Iping & Petterson 1990; Pringle 1996; Maloney et al. 1996).

Icke's computations were restricted to a two-dimensional proof-of-principle. Here we present a first series of fully three-dimensional hydrodynamic computations of such a wind-disk interaction. For this we use the publicly available hydrodynamics package *Flash* (Fryxell et al. 2000). We extended this massively parallel, Adaptive Mesh Refinement (AMR) code with a radiative cooling module and the proper initial conditions for our problem.

Even with the current supercomputers and AMR techniques we can follow the interaction between the stellar wind and the warped disk at a sufficiently high resolution only for the first couple of years of its evolution. This implies that our models are strictly speaking only valid for the first stages of proto-PN formation. However, since it is believed that PNe expand in a self-similar fashion (Icke 1988), our models may, although tentatively, be applied to later stages of PN evolution as well.

After an introduction to point-symmetric nebulae, we proceed by describing the mechanism behind radiatively warped disks, and give an equation for its shape. We then derive several time scales related to the disk, and compare these to a typical radiative cooling time scale of the gas, thus providing us with the limits of our parameter space.

In order to check the applicability of using a low value for the adiabatic index as an approximation to radiative cooling in our three-dimensional runs, we compare two-dimensional simulations ran with such a low γ, to ones that applied the cooling curve module.

We then digress briefly to explain the mechanism behind the formation of multipolar shells in our simulations, emphasizing the importance of radiative cooling and the intrinsic three-dimensional nature of the problem.

Thereafter, we present synthesized $H\alpha$ images and position velocity diagrams of our data, and compare these with observations of (proto-)PNe.

5.2   Point-symmetric nebulae

Work on cylindrically symmetric nebulae showed (Icke 1988, 1991; Icke et al. 1992) that sharply collimated bipolar flows are a frequent and natural by-product of the GISW. However, many (proto-)PNe have a multipolar or point-symmetric shape (Schwarz 1993; Sahai & Trauger 1998; Guerrero et al. 1999; Sahai 1999; Su et al. 2003; Harman et al. 2004). The nebulae that are formed in the wind-disk interaction would naturally acquire the observed antisymmetry if the disk that confines the fast wind is warped, instead of symmetric under reflection about the equatorial plane. Several mechanisms have been proposed for warping an accretion disk. The most interesting one for our purposes invokes radiative instability (Petterson 1977; Iping & Petterson 1990; Pringle 1996; Maloney et al. 1996).

Livio & Pringle (1996, 1997) already proposed that the precession of warped disks might be responsible for point-symmetric nebulae. They proved conclusively that the various physical scales for mass, accretion, luminosity and precession match the observations. The production of the nebulae proper they attributed to an unspecified 'jet' mechanism.

Another mechanism capable of producing point-symmetric PNe was presented by García-Segura (1997) and García-Segura & López (2000). In their three-dimensional magnetohydrodynamical models, the magnetic collimation axis is misaligned with respect to the bipolar wind, resulting in point-symmetric morphologies.

Observations of many bipolar nebulae with 'ansae' (e.g. NGC 3242, NGC 7009) and 'FLIERS' (e.g. NGC 6751, NGC 7662) leave little doubt that jets are occasionally formed during the evolution of some aspherical PNe, probably in the late post-AGB phase, before the fast wind has switched on. But the nebulae presented by Sahai & Trauger (1998) do not seem to resemble such shapes very much. The lobes of point-symmetric nebulae (Schwarz 1993; Sahai & Trauger 1998) look as if they were produced almost simultaneously. This is difficult in the case of a precessing jet, which would make a corkscrew-like nebula of a type not readily apparent in post-AGB shells, although some objects do show features that are likely to be due to precession (Schwarz 1993; Velázquez et al. 2004). While leaving open the possibility that jets may be responsible for additional structures, as in the case of the 'ansae', we show that the interaction between a warped disk and a spherically symmetric wind suffices.

Our numerical models of such a wind-disk interaction are, due to resolution restrictions, aimed at the first stages of proto-PN evolution (see Section 5.6 and Section 5.8). The following parameters, used in the simulations, are appropriate for this kind of objects (e.g. Reyes-Ruiz & López 1999): central star luminosity $L_* = 5 \times 10^3 L_\odot$, environment number density $n_e = 10^5 \, \text{cm}^{-3}$ and temperature $T_e = 500 \, \text{K}$, disk number density $n_d = 10^7 \, \text{cm}^{-3}$ and temperature $T_d = 5 \, \text{K}$, and stellar wind mass loss rate $\dot{M}_w = 1.7 \times 10^{-9} \, M_\odot \, \text{yr}^{-1}$ and velocity $v_w = 200 \, \text{km} \, \text{s}^{-1}$.

5.3   Radiation driven warping

When an accretion disk is subject to external torques it may become unstable to warping (Bardeen & Petterson 1975; Petterson 1977; Papaloizou & Pringle 1983) and when

irradiated by a sufficiently luminous central star even an initially flat disk will warp (Iping & Petterson 1990; Pringle 1996; Maloney et al. 1996, 1998). This warp originates in differences in radiation pressure on slightly tilted annuli at different radii in the disk.

For warping to be induced, it is essential for the disk to be optically thick for both the stellar radiation *and* for its own cooling flux. The latter condition is the most restrictive, because a disk that is optically thin in the infrared dust continuum will not suffice. While strictly speaking all we need is a warped disk, however this may be produced, we do believe that the Petterson-Iping-Pringle disks are the most plausible. This restricts the antisymmetric nebulae to a specific subclass of post-AGB stars with high density and low temperature disks around them. Another aspect that determines whether the disk gets warped is the luminosity of the central star, which should be sufficiently high.

But, even if the above conditions for warping are valid, it is still possible that the interaction of the wind with the disk does not lead to a point-symmetric nebula, since this further requires that the cooling time of the gas in the swept up shell is shorter than the other, dynamical, time scales of the problem (see Section 5.4 and Section 5.7). So, only when both the requirements for warping and time scales are met, a point-symmetric nebula may emerge.

Analytical considerations lead to expressions for growth and precession rates and morphologies of the warp whereas numerical calculations including the effects of self-shadowing show that the non-linear evolution of the warp can produce highly distorted shapes with, in extreme cases, an inverted, counter rotating inner disk region (Pringle 1997; Wijers & Pringle 1999). Applications of warped disk theory range from active galactic nuclei (e.g. Pringle 1997; Maloney et al. 1998) to X-ray binaries (Maloney & Begelman 1997; Wijers & Pringle 1999), protostellar disks (Armitage & Pringle 1997), and PNe (Livio & Pringle 1996, 1997). Other mechanisms that produce warped disks besides the radiatively driven one are the wind driven (Quillen 2001) and the magnetically driven instability (Lai 1999, 2003).

5.3.1 Origin of the disk

As we will show below, the required disks are quite small ($\sim 10 - 100$ AU) and are in Keplerian rotation around the star. This means that they are conceptually somewhat different from the confining structures in 'classic' GISW models, where a slowly expanding AGB wind is responsible for the aspherical shape of the emerging PN. Here we require a dense accretion disk, whose presence during the post-AGB phase will imprint itself on the shapes of young PNe.

There are several ways to form such a disk. In a binary system, a disk can be formed due to a main-sequence companion being out of equilibrium when emerging from a common envelope (CE) phase with a primary AGB star. This companion loses most of the mass it accreted during the CE phase which subsequently forms a disk around the primary (Soker & Livio 1994) and which, in the post-AGB phase, can get radiatively warped when illuminated by the central star. The accretion disk may also form around the companion star (Soker & Rappaport 2000), or possibly be circumbinary.

Reyes-Ruiz & López (1999) also investigated a number of processes for the formation
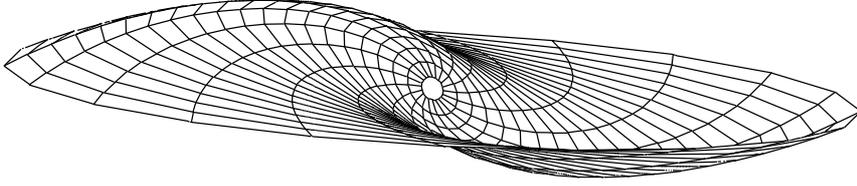
Figure 5.1: Example of a warped disk surface, equation (5.2).

of accretion disks in proto-PNe through binary interactions. For the case where a low mass secondary gets disrupted during a dynamically unstable mass transfer process, they found that an accretion disk forms within ~ 100 yr that has a radius of ~ 10 AU and a mass of $\sim 2 \times 10^{-3}\ M_\odot$.

A mechanism which works for single stars is a partial back flow from the stellar mass loss. Soker (2001) showed that some of the material lost by the (post-)AGB star may flow back to the star, forming an accretion disk.

Whatever the mechanism for their formation, circumstellar disks around proto-PNe (e.g. Miranda 1999; Kwok et al. 2000) and post-AGB stars have been observed. *ISO* spectra reveal the presence of crystalline silicates, which require a dense environment to form (Molster et al. 1999, 2002). Anomalous atmospheric abundance patterns found in a number of post-AGB stars, are most easily understood if gas and dust were separated while orbiting in an accretion disk (Van Winckel et al. 1998, 1999; Maas et al. 2003). CO line shapes also indicate the presence of reservoirs of gas in Keplerian orbits (Jura & Kahane 1999; Bujarrabal et al. 2003). So, although the mechanism by which they form is not fully understood, there is mounting evidence for the presence of accretion disks around proto-PNe and post-AGB stars.

5.3.2 Shape of the disk

Following Pringle (1997), an expression for the radius R_{crit} beyond which the disk is unstable to radiation driven warping is found from comparing the time scales of the viscous and radiation torques, leading to

$$R_{\mathrm{crit}} \equiv (2\pi/A)^2 = 16\pi^2 c^2 G M_* L_*^{-2} \eta^2 \dot{M}_{\mathrm{acc}}^2 \ , \tag{5.1}$$

where we assumed a surface density $\Sigma_{\mathrm{d}} = \dot{M}_{\mathrm{acc}}/(3\pi\nu_1)$ (e.g. Pringle 1981). Here $\eta \equiv \nu_2/\nu_1$ is the ratio of the azimuthal to the radial viscosity, M_* is the mass and L_* the luminosity of the central star, and $\dot{M}_{\mathrm{acc}}$ is the disk's accretion rate. For a typical PN with a central white dwarf with mass of $\sim 0.6\ M_\odot$, the luminosity is $\sim 5 \times 10^3\ L_\odot$, which is sufficiently high to induce a radiation driven warp.

In a Cartesian coordinate system, the warped disk surface is given by (Pringle 1996)

$$\boldsymbol{x}(R, \phi) = R \left(\begin{array}{ccc} \cos\phi\sin\gamma & + & \sin\phi\cos\gamma\cos\beta \\ -\cos\phi\cos\gamma & + & \sin\phi\sin\gamma\cos\beta \\ & - & \sin\phi\sin\beta \end{array} \right) \, , \tag{5.2}$$

with local disk tilt angle $\beta(R, \phi)$, and orientation angle of the line of nodes $\gamma(R, \phi)$. An example of such a surface is shown in Figure 5.1. R and ϕ are the non-orthogonal radial and azimuthal coordinates respectively, pointing to the surface of the disk (cf. Pringle 1996). In our model calculations we adopt the case of a steady precessing disk with no growth and zero torque at the origin for which we have in the precessing frame that $\gamma = A\sqrt{R}$ and $\beta = \sin\gamma/\gamma$, with the constant A defined by equation (5.1) (Maloney et al. 1996). The radius at which the warped disk returns to the plane is calculated from setting $x_3 = 0$ in equation (5.2). This leads to $R = (k\pi/A)^2$ $(k = 0, 1, 2, ...)$ and since the disk is unstable to warping for radii $R > R_{\rm crit}$, we take the outer disk radius to be $r_{\rm d} = 4R_{\rm crit}$ (i.e. $k = 4$).

5.4 Time scales

As shown in Icke (2003), the mechanism for the formation of multipolar nebulae only works if the gas is close to isothermal, or in other words, strongly cooling. In order to establish the part of parameter space where the mechanism applies, we need to compare the cooling time scale to three other time scales related to the disk: the precession and growth time scales of the warp, and the shock passing time, where the latter is defined as the time the expanding swept up shell takes to travel to the outer disk radius.

In our simulations we implemented the radiative cooling by interpolation from a tabulated cooling curve taken from Dalgarno & McCray (1972) (see Section 5.5). In order to calculate an estimate for the cooling time scale, we represent this curve by the following, piecewise linear, approximation:

$$\Lambda = \Lambda_i T_s^{\alpha_i} \, , \tag{5.3}$$

with Λ in $\rm erg\,g^{-1}s^{-1}$. We define two temperature ranges: $10^2 - 5 \times 10^4$ K, and $5 \times 10^4 - 10^7$ K, which we refer to as temperature range A and B, respectively. The parameters Λ_i and α_i for these ranges are taken as

$$\Lambda_{\rm A} = 2.5 \times 10^{-27}, \ \Lambda_{\rm B} = 1.33 \times 10^{-19} \, , \tag{5.4}$$

and

$$\alpha_{\rm A} = 0.3, \ \alpha_{\rm B} = -0.5 \, . \tag{5.5}$$

From a similar derivation as the one given by Kahn (1976) [see also Koo & McKee (1992)], for which the details are given in Appendix 5.A, we find a cooling time

$$t_{{\rm c},i} = \frac{3}{2(1 - \alpha_i)} \frac{x_t k_{\rm B} T_s^{1-\alpha_i}}{n_{\rm H}\Lambda_i} = C_i \frac{v_s^{2(1-\alpha_i)}}{\rho_{\rm e}} \tag{5.6}$$
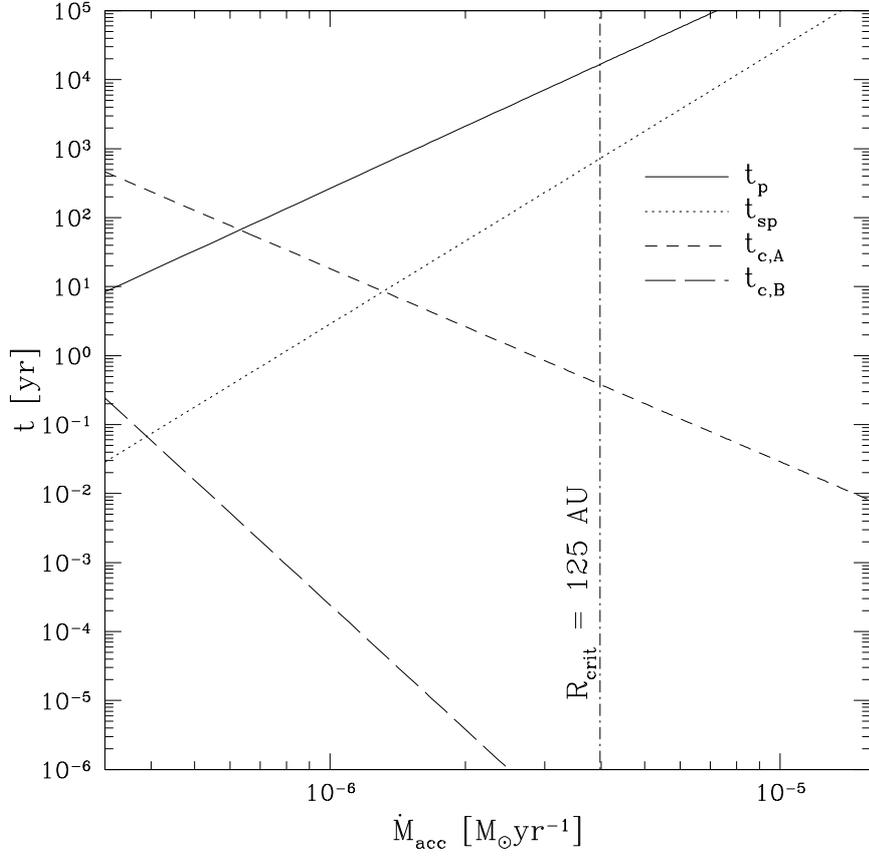
Figure 5.2: Plot of t_p, t_sp, $t_\mathrm{c,A}$, and $t_\mathrm{c,B}$ as a function of $\dot{M}_\mathrm{acc}$. We used $r_\mathrm{d} = R_\mathrm{crit}$ to calculate t_p, and $r_\mathrm{d} = 4 \times R_\mathrm{crit}$ to calculate t_sp, $t_\mathrm{c,A}$, and $t_\mathrm{c,B}$. The vertical line indicates the value of $\dot{M}_\mathrm{acc}$ for which $R_\mathrm{crit} = 125$ AU. The environment density was set to $n_\mathrm{e} = 10^5$ cm^{-3}. See Section 5.6 for the values of other parameters.

where

$$C_i = \frac{1}{\Lambda_i} \left[\frac{3}{2(1 - \alpha_i)} \mu_\mathrm{H}^{2-\alpha_i} x_t^{\alpha_i} k_\mathrm{B}^{\alpha_i} 2^{1-\alpha_i} (\gamma - 1)^{2-\alpha_i} (\gamma + 1)^{-(3-2\alpha_i)} \right] , \qquad (5.7)$$

and with x_t the number of particles per hydrogen nucleus, k_B Boltzmann's constant, T_s the temperature of the shocked gas, n_H the number density of hydrogen, v_s the shock speed, and ρ_e the pre-shock environment density. For a fully ionized gas of cosmic abundances we have

$$C_A = 1.0 \times 10^{-19}; \; C_B = 6.0 \times 10^{-35} . \qquad (5.8)$$

We further assume that the wind blown bubble is in the 'momentum driven snowplow phase' (e.g. Lamers & Cassinelli 1999), so the swept up shell is thin, and the outer shock is approximately at the radius

$$r(t) = \left(\frac{3\dot{M}_\mathrm{w} v_\mathrm{w}}{2\pi \rho_\mathrm{e}} \right)^{1/4} t^{1/2} , \qquad (5.9)$$

with time t, mass loss rate of the wind $\dot{M}_\mathrm{w}$, and terminal wind velocity v_w. The shock velocity readily follows by taking the derivative, which leads to an expression for the cooling time of the swept up shell given by

$$t_{\mathrm{c},i} = C_i \left(\frac{3\dot{M}_\mathrm{w} v_\mathrm{w}}{32\pi} \right)^{\frac{1}{2}(1-\alpha_i)} \rho_\mathrm{e}^{-\frac{1}{2}(3-\alpha_i)} t^{\alpha_i - 1} \ . \tag{5.10}$$

As mentioned above, three time scales related to the disk are of importance. First the precession time scale

$$t_\mathrm{p} = 48\pi^2 c G^{1/2} M_*^{1/2} L_*^{-1} r_\mathrm{d}^{3/2} \Sigma_\mathrm{d} \tag{5.11}$$

(Maloney et al. 1996) where we assumed Keplerian rotation. Second the time scale for the initial growth of the warp which is of the same order as t_p. The third time scale is the shock passing time t_sp which follows from setting $r(t_\mathrm{sp}) = r_\mathrm{d}$ in equation (5.9) as

$$t_\mathrm{sp} = \left(\frac{2}{3} \pi \rho_\mathrm{e} \right)^{1/2} \dot{M}_\mathrm{w}^{-1/2} v_\mathrm{w}^{-1/2} r_\mathrm{d}^2 \ . \tag{5.12}$$

When we use equation (5.1) for the critical radius as a typical disk radius, we find that the different times scale as

$$
\begin{aligned}
t_\mathrm{p} &\propto M_*^2 L_*^{-4} \dot{M}_\mathrm{acc}^3 \eta^3 \Sigma_\mathrm{d} \\
t_\mathrm{sp} &\propto \rho_\mathrm{e}^{1/2} \dot{M}_\mathrm{w}^{-1/2} v_\mathrm{w}^{-1/2} M_*^2 L_*^{-4} \dot{M}_\mathrm{acc}^4 \eta^4 \\
t_{\mathrm{c},i} &\propto \rho_\mathrm{e}^{\alpha_i - 2} \dot{M}_\mathrm{w}^{\frac{1}{2}(1-\alpha_i)} v_\mathrm{w}^{\frac{1}{2}(1-\alpha_i)} M_*^{2(\alpha_i-1)} L_*^{-4(\alpha_i-1)} \dot{M}_\mathrm{acc}^{4(\alpha_i-1)} \eta^{4(\alpha_i-1)} \ ,
\end{aligned}
$$

where we set $t = t_\mathrm{sp}$ in equation (5.10) for the cooling time scale.

We are quite limited in our choice of M_*, L_*, v_w, and $\dot{M}_\mathrm{w}$ since values for these parameters are constrained by stellar evolution and wind models (e.g. Pauldrach et al. 1988; Blöcker 1995) but since the dependence of t_p, t_sp, and t_c on $\dot{M}_\mathrm{acc}$ is so strong, a proper choice of this latter parameter leads to the desired relation between the different time scales. However, since the disk's radius depends on $\dot{M}_\mathrm{acc}$ as well [equation (5.1)], we can not take it too large, since that would lead to an unrealistic size for the disk. We have indicated this restriction in Figure 5.2 which shows a plot of t_p, t_sp and t_c as a function of $\dot{M}_\mathrm{acc}$. For all simulations we chose our parameters such that at all times $t_\mathrm{c} < t_\mathrm{sp} < t_\mathrm{p}$, so that cooling will indeed be important and that we can safely ignore the disk's precession. See Section 5.6 and Section 5.8 for actual values of the parameters we used as initial conditions for our simulations.

5.5 Numerical implementation

We used the three-dimensional hydrodynamics package *Flash* (Fryxell et al. 2000) to model the interaction between a spherical wind and a warped disk. This parallelized code implements block-structured AMR initially developed by Berger & Oliger (1984) (see also Berger & Colella 1989) and a PPM type hydro-solver (Colella & Woodward 1984).

Besides implementing the proper initial and boundary conditions we also added our own cooling module to the code in order to model radiative cooling using a cooling curve (Dalgarno & McCray 1972; Mellema et al. 2002). This curve gives the energy loss rate as a function of temperature for a low density gas in collisional ionization equilibrium. The radiative losses were implemented through operator splitting and if the time step required by the hydrodynamics was larger than the cooling time, the former was subdivided into smaller steps when calculating the cooling.

The temperature T_e of the environment was kept at its initial equilibrium value by introducing a heating term with a linear dependency on the density, as photo-electric heating would have. The temperature T_d of the disk was chosen such that it was in pressure equilibrium with the environment. Furthermore, we made sure no grid cell could obtain a temperature below a minimum value of T_d.

To construct the warped disk, equation (5.2) was combined with a constant 'flare angle' ϕ_d, and a proper value for A. The outer disk radius r_d was taken to be larger than R_{crit} (cf. equation (5.1)) and the disk was given a constant number density n_d. The environment was given a constant number density n_e. The spherical wind was implemented as an interior boundary condition and given a $1/r^2$ density profile, outer radius r_w, and a constant wind velocity v_w. The pressure was calculated from an equation of state with a constant adiabatic index γ.

Both the module to construct the initial conditions as well as the cooling curve module are available from the authors upon request.

5.6   Two-dimensional wind-disk simulations

To check the implementation of the wind-disk interaction model into the AMR code described above, we repeated a number of the two-dimensional calculations previously done by Icke (2003). Since in his simulations a different solver (FCT/LCD) for the hydrodynamic equations was used, and the shape of small scale structures in the outcome of the calculations depend on turbulent processes in the gas, the two-dimensional comparison simulations did not agree in every single detail, but the overall point-symmetric morphologies found, were similar in both simulations.

To see the effects of more realistic cooling, we also ran some simulations with the cooling curve module and an adiabatic index $\gamma = 5/3$. At the start of each simulation, the cooling time is already smaller than the hydrodynamical time scale, so the bubble is in the momentum driven stage from the outset. Since the cooling scales with the density squared and the density in the swept up shell can only increase with time, the cooling gets more efficient as the simulation progresses, and the bubble remains momentum driven throughout the entire simulation. To sufficiently resolve the developing bow shock and instabilities in these calculations required a fairly high (effective) resolution of 2048^2 cells. All two-dimensional simulations used either a small value for the adiabatic index or the cooling curve module, both resulting in momentum driven shells. See Table 5.1 for information on the initial parameters. For all two-dimensional simulations with cooling applied through the cooling curve module we used the following parameters: $\dot{M}_{acc} = 1.5 \times 10^{-6}\,M_\odot\,\mathrm{yr}^{-1}$, $v_w = 200\,\mathrm{km\,s}^{-1}$, $M_* = 0.6 M_\odot$, $L_* = 5 \times 10^3 L_\odot$, $\Sigma_d = 1\,\mathrm{g\,cm}^{-2}$, $r_d = 100\,\mathrm{AU}$, $\phi_d = 5°$,
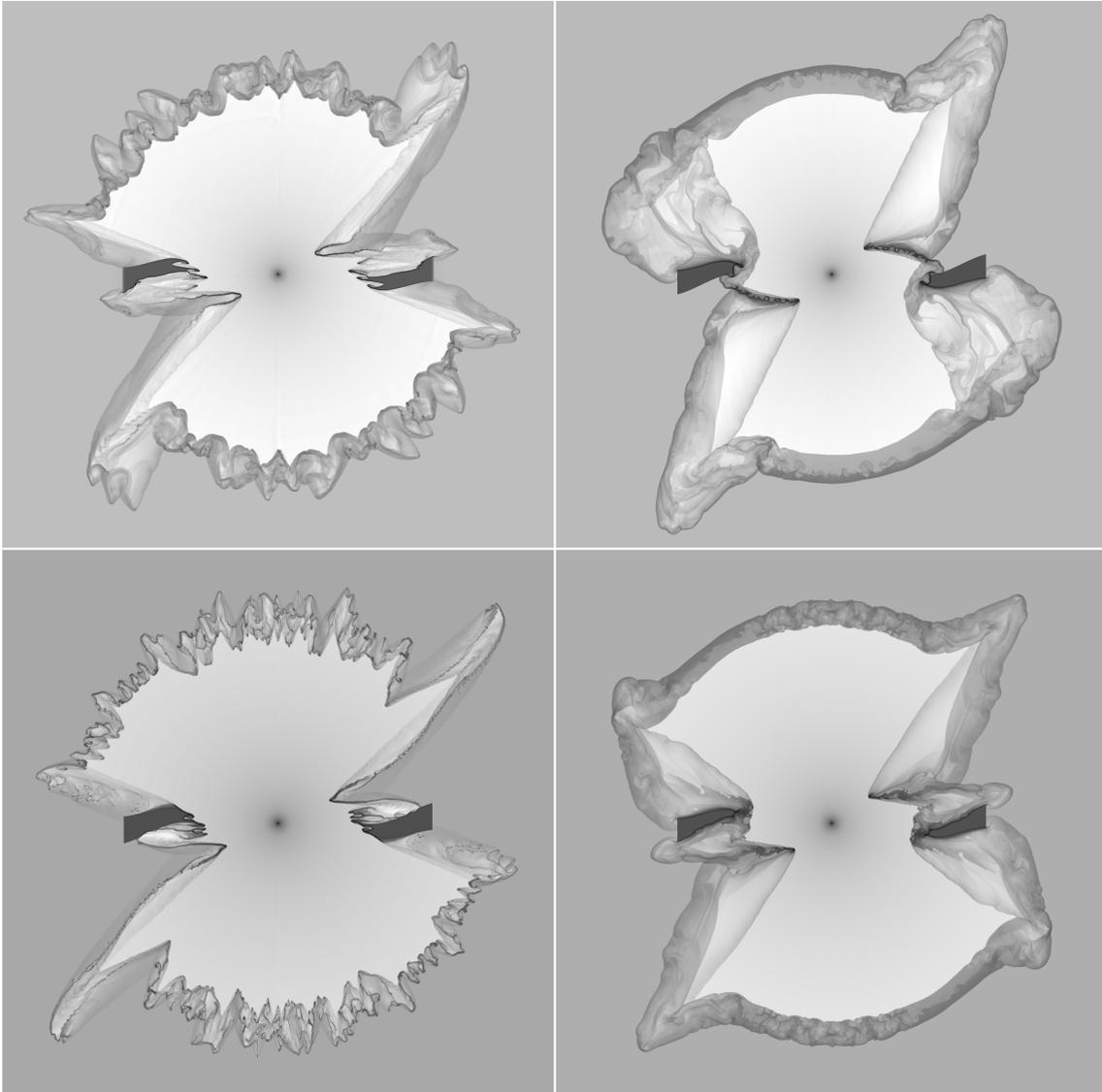
Figure 5.3: Grey scale plot of $\log_{10}$ of the density $(\mathrm{g\,cm^{-3}})$ at $t = 9.0$ yr for several two-dimensional wind-disk simulations using either the cooling curve module (left hand side) or a low value of gamma ($\gamma = 1.1$, top right; $\gamma = 1.05$, bottom right) to simulate radiative cooling. Parameters for initial densities and temperatures can be found in Table 5.1. Actual values of the density range from 4.0×10^{-20} to 1.25×10^{-15} (top row) and from 3.2×10^{-20} to $7.9 \times 10^{-15} \mathrm{\,g\,cm^{-2}}$ (bottom row). The size of the domain is $334 \mathrm{\,AU^2}$, and a total of eight levels of refinement were used resulting in an effective resolution of 2048^2 cells.

Table 5.1: Parameters for the two- and three-dimensional runs.

Figure #	5.3 (top-row)	5.3 (bottom-row)	5.5
$n_{\rm e}$ (cm^{-3})	10^5	10^6	10^5
$T_{\rm e}$ (K)	500	5000	500
$n_{\rm d}$ (cm^{-3})	10^7	10^8	10^7
$T_{\rm d}$ (K)	5	50	5
$\dot{M}_{\rm w}$ ($M_\odot$ yr^{-1})	1.7×10^{-9}	1.7×10^{-8}	1.7×10^{-9}
$R_{\rm crit}$ (AU)	18	18	18
$t_{\rm p}$ (yr)	> 890	> 890	> 890
$t_{\rm sp}$ (yr)	15	15	15
$t_{\rm c,A}$ (yr)	5.8	0.58	5.8
$t_{\rm c,B}$ (yr)	2.0×10^{-5}	2.0×10^{-6}	2.0×10^{-5}

and $\eta = 1$. For the calculation of $t_{\rm sp}$ we used $r_{\rm d} = 4 \times R_{\rm crit} \simeq 72$ AU which is of the same order as the value of 100 AU we used in the simulations. Note that, for this parameter choice, the Shakura-Sunyaev parameter $\alpha \simeq \nu_1/(4R_{\rm crit} c_{\rm s})$, obeys $\alpha \le 1$, as is required (see e.g. Pringle 1981). Also note that, in the simulations, the mass loss rate of the wind $\dot{M}_{\rm w}$ is a function of $\rho_{\rm e}$, such that at $r = r_{\rm w}$ the wind density is set equal to the initial environment density, i.e. $\dot{M}_{\rm w} = \pi r_{\rm w}^2 \rho_{\rm e} v_{\rm w}$, which renders equation (5.12) independent of $\rho_{\rm e}$ (as is reflected in the constant $t_{\rm sp}$ in Table 5.1 for different values of $n_{\rm e}$). The disk mass corresponding to these numbers is quite modest, only $3.5 \times 10^{-3} \, M_\odot$. This shows that the mechanism does not require excessive conditions to operate.

For the runs that were done with the cooling curve module we find that, apart from the formation of the, by now familiar, point-symmetric lobes, the outer shell of swept up gas is much thinner and unstable due to the effective cooling at higher densities. It develops into a number of smaller lobes merging with one another as the shell expands (Figure 5.3). Furthermore, the more effective the cooling, the more elongated the lobes created by the bow shock become.

5.7 Mechanism

The mechanism behind the formation of the multipolar lobes seen in the simulations is as follows (see also Icke 2003). As the central wind impinges on the inner rim of the disk, a three-dimensional bow shock develops around it. The opening angle of the shock depends inversely on the Mach number of the wind. A two-dimensional cut perpendicular to the plane of the disk (see Figure 5.3 and Figure 5.5) shows the developing bow shock, of which one branch flies off into space creating a lobe jutting out from the nebula, whereas the other slams into the concave side of the disk.

Due to the cooling of the gas, the swept up shell is highly compressed and therefore thin, and the ram pressure from the wind directly drives the shell outwards, which are necessary ingredients for the bow shock to produce the lobes. When the cooling is very effective, the shell gets so thin that it becomes unstable, and develops into a number of smaller lobes. Ultimately, when the density of the disk is not too high, the wind breaks
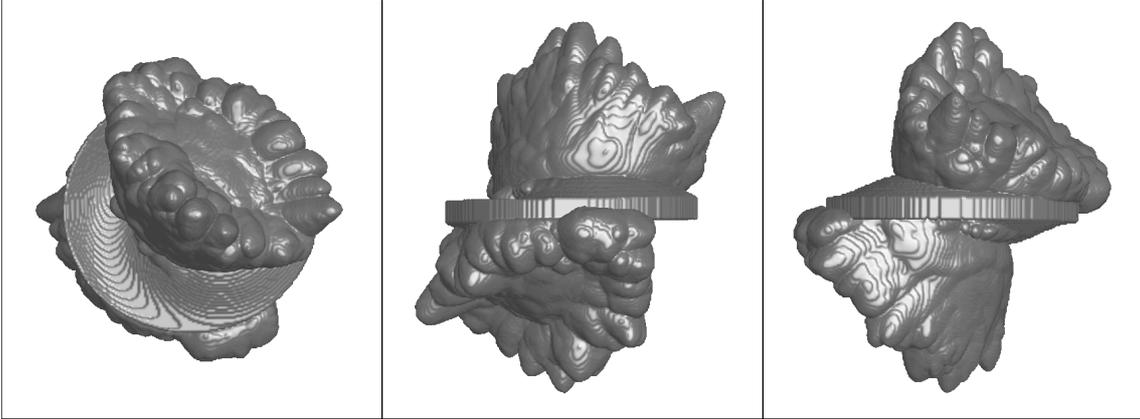
Figure 5.4: Isosurfaces for a density value of $10^{-17} - 10^{-16}\,\mathrm{g\,cm^{-3}}$ corresponding to the cuts shown in the bottom row of Figure 5.5 for the three-dimensional calculation at $t = 12\,\mathrm{yr}$. The size of the domain is $334\,\mathrm{AU}$.

through the concave part of the disk, producing another pair of lobes (Figure 5.3).

Since the true bow shock is a three-dimensional structure and the disk is warped, the shape of the disk seen in a two-dimensional vertical cut at different azimuthal angles ϕ changes such that the concave side of the disk turns into the convex side and vice versa for increasing ϕ.

As was pointed out above, the interaction between a spherical wind and a warped disk is point-symmetric by nature, and thus intrinsically three-dimensional. The two-dimensional calculations should therefore be regarded as trial calculations that show the underlying principles of the interaction but, due to the slab symmetry of these two-dimensional simulations, which does not correctly capture the point-symmetry of the problem, can not be used to derive any observable morphologies or position-velocity diagrams. So, to truly understand the emerging point-symmetric structure of this interaction, it is necessary to carry out fully three-dimensional simulations.

5.8 Three-dimensional wind-disk simulations

We ran wind-disk simulations in three dimensions on a Cartesian grid with an effective resolution of 512^3 using six levels of refinement. For our largest calculations we needed approximately 24 hours on 64 nodes of an SGI Origin 3800 system, where each node consists of a 500 MHz R14000 CPU with 1 GB of memory. At the end of each simulation, $\sim 30\%$ of the domain is refined to the highest level, taking all the memory available. This means that if one would like to follow the simulation for a longer evolutionary time by using a larger domain, and keeping the same (effective) resolution, one should increase the number of nodes accordingly.

Since we found in our two-dimensional calculations that simulations with cooling applied through a cooling curve did not result in a qualitatively different morphological outcome compared to calculations with a low value for the adiabatic index, we opted not to use the cooling curve module during our three-dimensional simulations to save
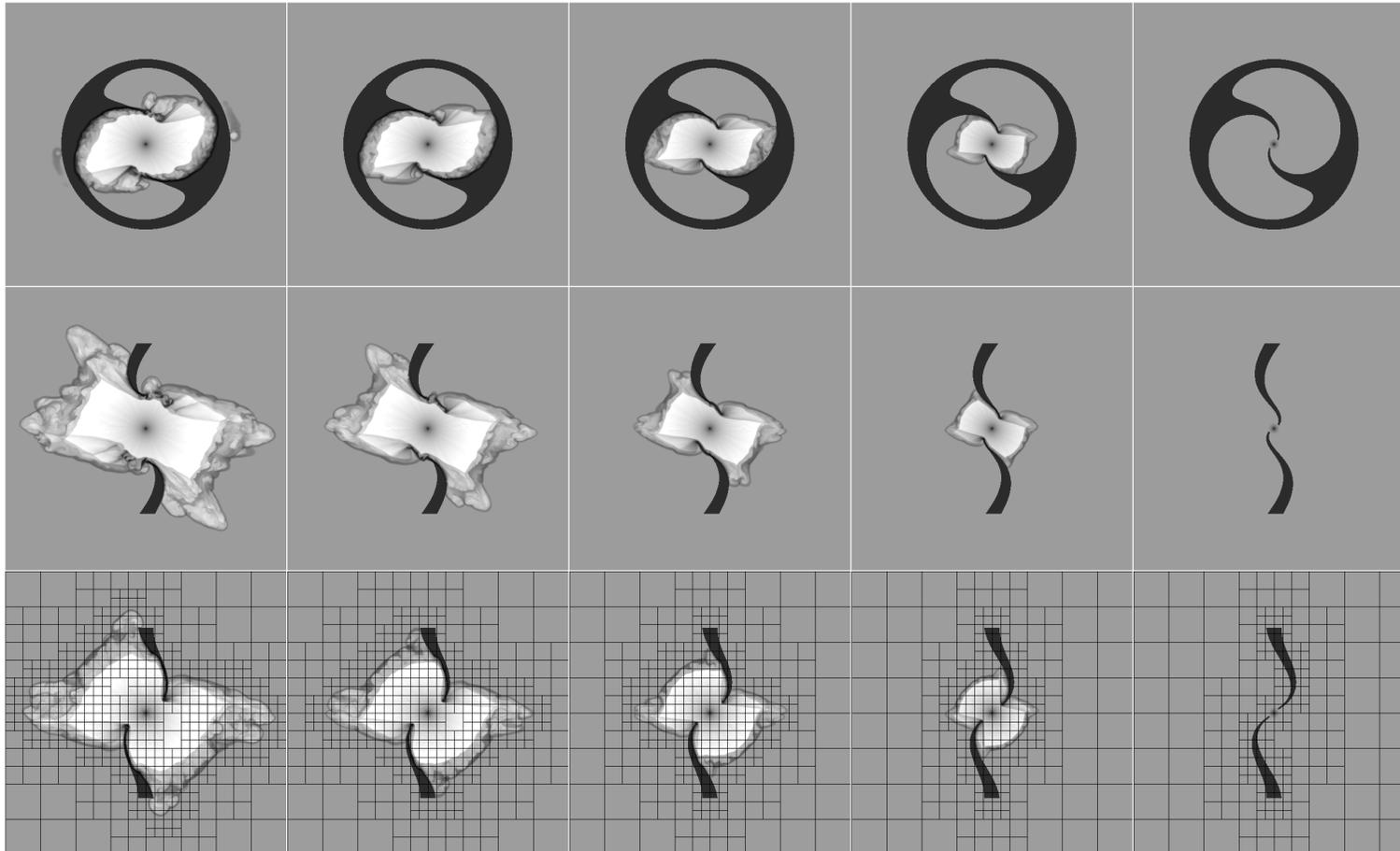
Figure 5.5: Grey scale plots of $\log_{10}$ of the density ($\mathrm{g\,cm^{-3}}$) at $t = 0$ yr, $t = 3$ yr, $t = 6$ yr, $t = 9$ yr, and $t = 12$ yr (right to left). Actual values of the density range from 3.0×10^{-21} to $1.0 \times 10^{-16}\,\mathrm{g\,cm^{-3}}$. The size of the domain is $334\,\mathrm{AU}^3$. Shown are cuts through the centre of the data cube in the xy-plane (top), xz-plane (middle), and yz-plane (bottom). The plots for the yz-plane show the grid structure superimposed where only the three finest levels of refinement are visible (coarser levels are fully refined). The corresponding isosurfaces are shown in Figure 5.4.

on computing time. Like in the two-dimensional simulations, we used a disk radius of $100\,\mathrm{AU}$, a wind velocity of $200\,\mathrm{km\,s}^{-1}$ and a computational domain of $334\,\mathrm{AU}^3$. See Table 5.1 for a summary of the simulation parameters and resulting time scales.

5.8.1   Morphology and kinematics

In Figure 5.5 we show cuts through the centre of the data cube at different times during the simulation. One sees that the point-symmetric structure again emerges, and that the bow shock produces multiple lobes, as in the two-dimensional simulations. To visualize the three-dimensional shape of the swept up shell, corresponding isosurfaces are also shown (Figure 5.4).

Since it is the bow shock that drives the shaping of the bubble, any extra small scale instabilities resulting from the extra degree of freedom of our three-dimensional simulations are of relatively little importance. However, one difference between the two-dimensional and the three-dimensional calculations is the way in which the small scale lobes can merge along the surface of the swept up shell, but this does not alter the global shape of the bubble. Also, the extra degree of freedom prevents the wind from breaking through the concave part of the disk, in contrast to what was found from the two-dimensional simulations.

We derived synthesized $H\alpha$ images by projecting the three-dimensional data cube onto the plane of the sky. We integrated the density squared along the line of sight and used this as an estimate for the emission. In Figure 5.6 we present an overview of the point-symmetric shapes we found. Varying the projection angle produces a plethora of morphologies ranging from clearly quadrupolar (e.g. Figure 5.6 $(\theta, \phi) = (000, 040)$) to multipolar (e.g. Figure 5.6 $(120, 020)$). We can also chose our projection angle such that a more bipolar-like shape emerges, see e.g. Figure 5.6 $(160, 080)$, or even an almost round shape (Figure 5.6 $(140, 120)$).

In Figure 5.7 the kinematic data in the form of position-velocity diagrams is presented. These diagrams correspond to the $H\alpha$ images of Figure 5.6, where, for each diagram, the 'slit' was positioned at the centre of the image and ran in the vertical direction. To be representative for current observations, we convolved the position-velocity diagrams with two-dimensional Gaussians with a FWHM of $10\,\mathrm{km\,s}^{-1}$ along the velocity axis, and a FWHM of $0.''05$ along the position axis (assuming a distance of $1\,\mathrm{kpc}$ to the nebula). One sees that the point-symmetry is also retained in these diagrams. We present larger versions of three of our $H\alpha$ images and their corresponding position-velocity diagrams in Figure 5.8.

5.8.2   Comparison with observations

Since, even with the current state-of-the-art supercomputer and AMR techniques, we can follow the evolution of the expanding bubble only up to about 1% (10%) of the physical size of a typical PN (proto-PN), it is difficult to make a quantitative comparison between our models and the observations. However, although our models are strictly speaking only valid during this early development, we expect the shape imprinted on the wind blown
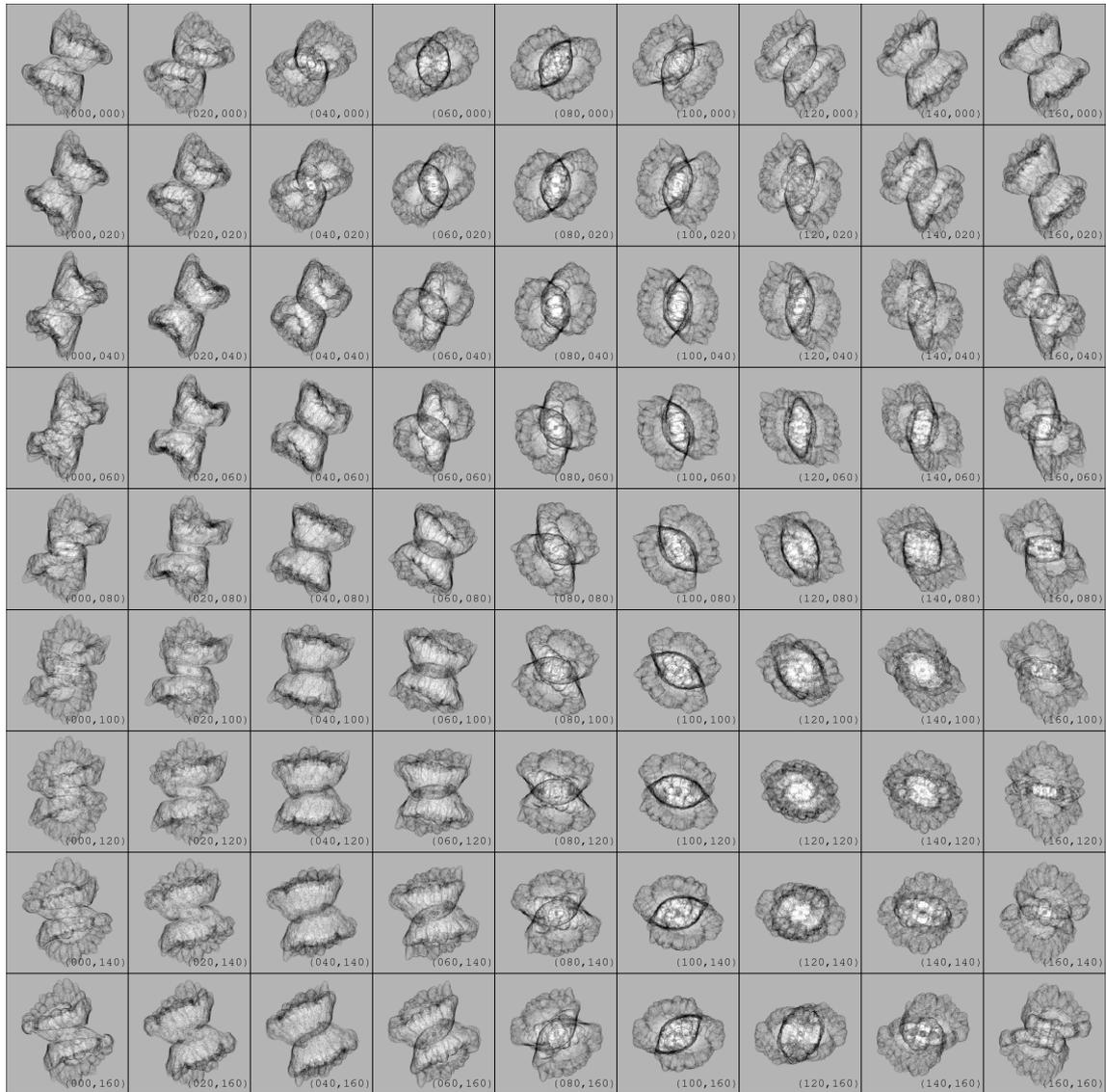
Figure 5.6: Grey scale plots of the synthesized Hα images (inverted; darker shades represent higher emission rates). Different projections of the three-dimensional data cube at $t = 12\,\mathrm{yr}$ are shown. The size of each box is $334\,\mathrm{AU}^2$ and the $\log_{10}$ of the density squared integrated along the line of sight is plotted. The angles of rotation (θ, ϕ) for each image are shown in the lower right corner, with θ and ϕ the usual spherical coordinate angles. Those parts of the disk that were not yet swept up by the wind are filtered out.
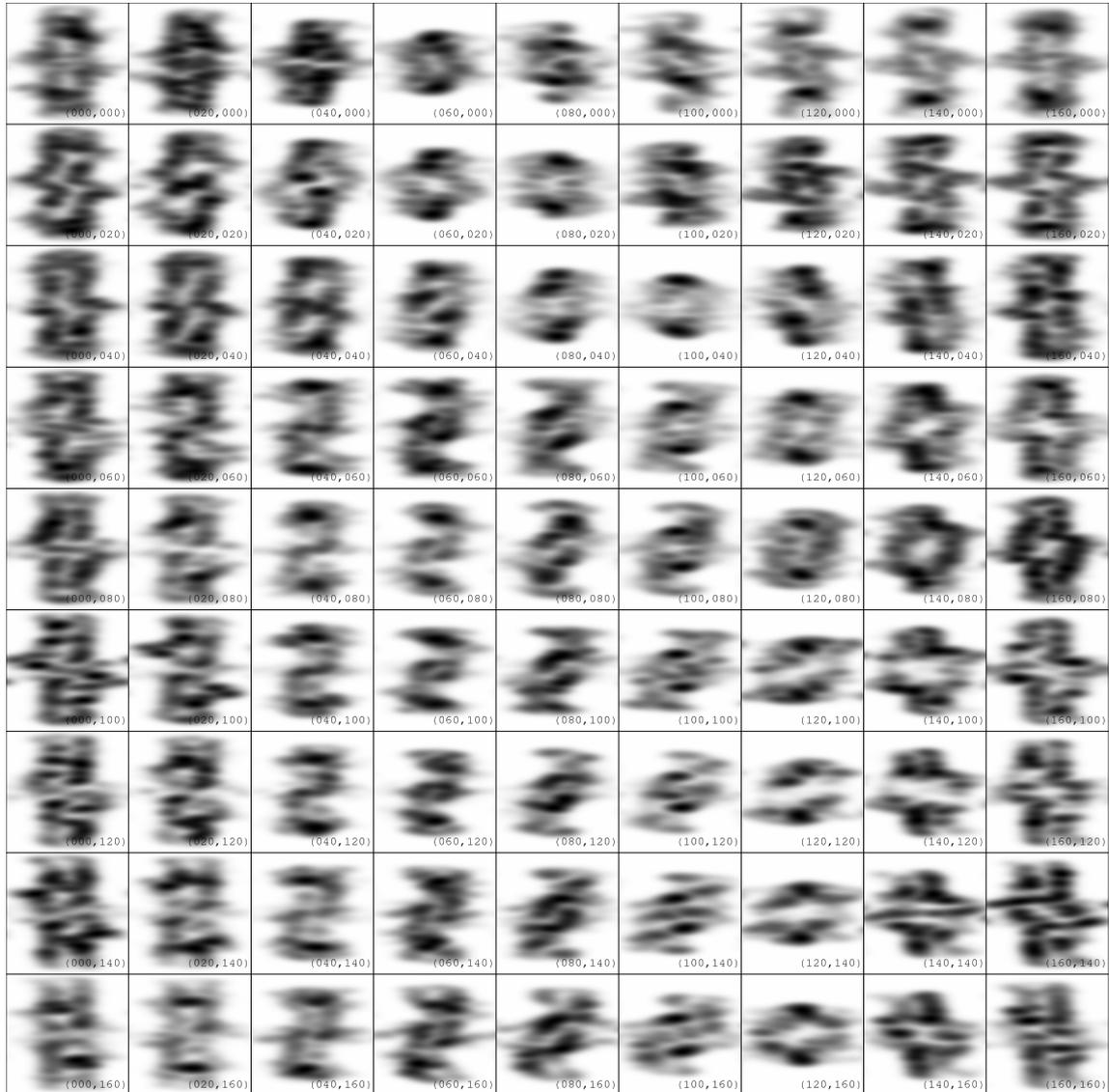
Figure 5.7: Grey scale plots of the synthesized position-velocity diagrams corresponding to the Hα images of Figure 5.6. Velocity is plotted along the horizontal axis and ranges from -33 to $+33\,\mathrm{km\,s^{-1}}$. Position is plotted along the vertical axis and ranges from 0 to $334\,\mathrm{AU}$. The grey scale represents the $\log_{10}$ of the flux, (inverted; darker shades represent higher emission rates). All diagrams were convolved with a two-dimensional Gaussian with a FWHM of $10\,\mathrm{km\,s^{-1}}$ along the velocity axis, and a FWHM of $0''.05$ along the position axis.
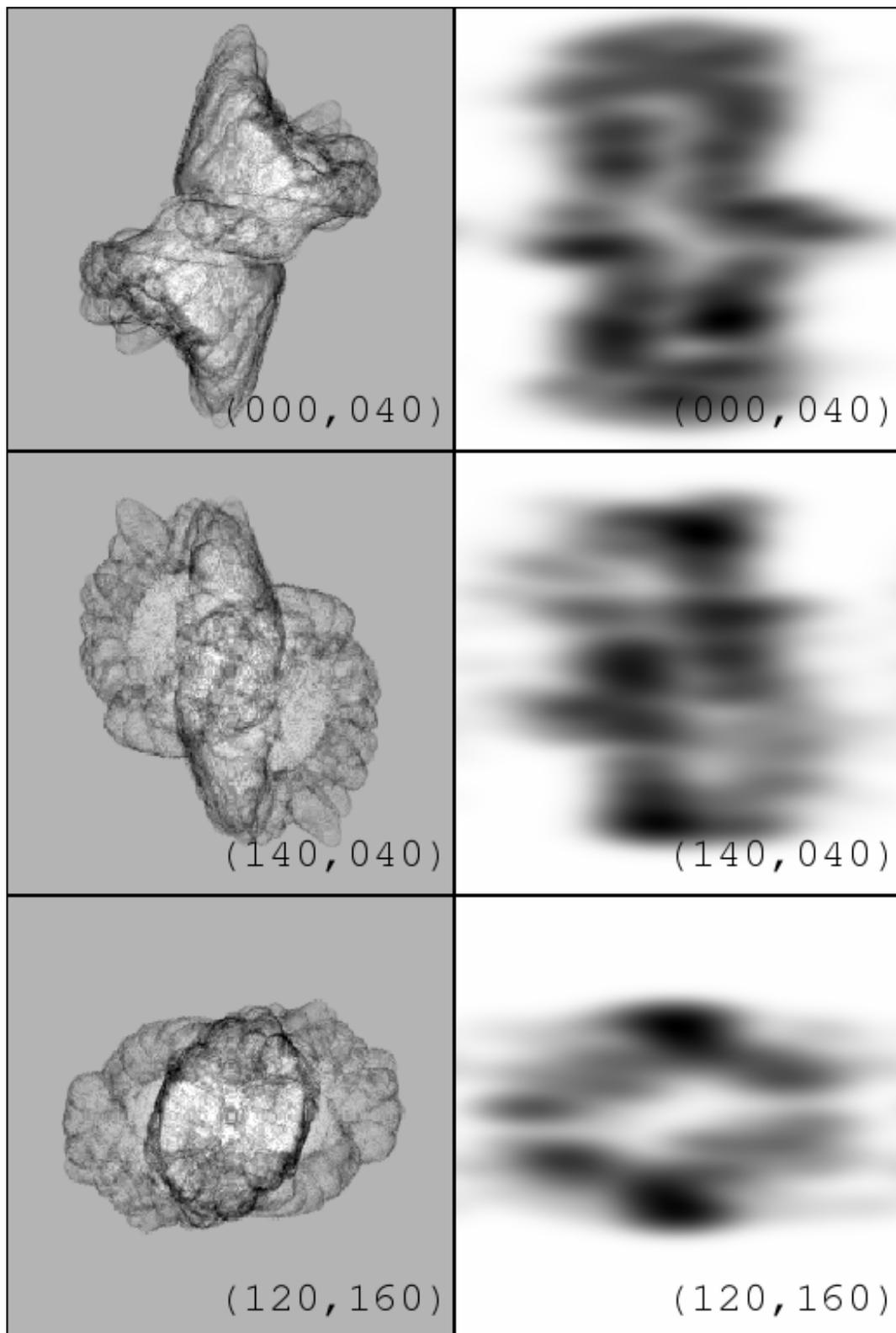
Figure 5.8: Larger versions of three synthesized Hα images and their corresponding position-velocity diagrams (consult Figure 5.6 and Figure 5.7 for more details on these results).

bubble to persist into later phases, since, due to the $1/r^2$ decrease in density (created by an earlier AGB mass loss phase) of the environment at larger scales (10^4 AU), the bubble will expand in a self-similar fashion (Icke 1988).

Whether the shell remains momentum driven in the long term when it expands into a $1/r^2$ environment depends on parameters like the velocity and mass loss rate of the evolving fast wind, and the actual density distribution in the surrounding medium left behind by the earlier AGB phase. For example, detailed radiation hydrodynamical calculations performed by Mellema (1993), taking an evolving fast wind and ionization effects into account, showed that there is a transition from momentum to energy driven flow at ~ 1000 yr. We therefore expect the multipolar shape of the bubble acquired in the early stages through our mechanism to persist for at least this long a period. Thus we may also, although tentatively, since the fast wind velocity is expected to increase in time which may alter the morphology of the bubble, compare our results to more evolved PNe. So, if one assumes that the bubble will expand self-similarly, it will have a size of $\sim 10^4$ AU (for a PN of $\sim 10''$ at a distance of 1 kpc) at $t \simeq 500$ yr, which are reasonable values for observed PNe.

When comparing our Hα images to observed objects, good candidates that can be accommodated by our model are the quadrupolar PNe K3-24 (Manchado et al. 1996) and NGC 7026 (Cuesta et al. 1996; Solf & Weinberger 1984), and the proto-PNe M1-26 and M4-18 (Sahai & Trauger 1998; de Marco & Crowther 1999). Although nebulae as compact as those in our models have not yet been observed in sufficient detail, the morphological similarity with the more evolved objects just mentioned suggests that they represent early stages of these.

Observed position-velocity diagrams for NGC 7026 are presented in Cuesta et al. (1996), together with a rather ad-hoc kinematic model. Upon comparison of our position-velocity diagrams with the observed ones we find qualitatively similar structures, strengthening the validity of our model. For K3-24 and the two proto-PNe mentioned, kinematic data are unfortunately not (yet) available. Another interesting property of observed point-symmetric PNe is that not only the imagery, but also the position-velocity diagrams, show evidence of point-symmetry (e.g. Guerrero et al. 1999; Harman et al. 2004), like we find in our synthetic position-velocity diagrams.

5.9   Conclusions

We have presented three-dimensional AMR simulations for the formation of multipolar nebulae from the interaction of a fast spherical stellar wind with a warped circumstellar disk. Although two-dimensional simulations demonstrate the mechanism behind the creation of these multipolar bubbles, the point-symmetry of the problem requires a fully three-dimensional treatment.

From a calculation of the different time scales of the problem, we determined the boundaries of our parameters space. We found that, although stellar evolution models limit the freedom of choice for a number of parameters, the scaling of the time scales is such that one can always find a proper set of physical parameters.

The influence of more realistic radiative cooling using a cooling curve was investi-

gated, and it was found that, in order to resolve the bow shock that generates the protruding point-symmetric lobes, high resolution simulations were required. Furthermore, the resulting morphologies of the wind blown bubbles found in these two-dimensional simulations using a cooling curve did not differ qualitatively from the ones found in the simulations where simplified cooling was applied through the use of a low value of the adiabatic index. Therefore, we used the latter, computationally less expensive cooling strategy, for our three-dimensional runs.

When analyzing the results from the three-dimensional calculations, we found that our model produces a wide variety of morphologies, ranging from multipolar to quadrupolar and bipolar. All these shapes were extracted from one single time frame of one single calculation by systematically choosing a number of different projection angles for the data cube. Besides these synthetic Hα images, we made the corresponding position-velocity diagrams, which also clearly show the point-symmetry of the model.

By comparing a number of observed (proto-)PNe with these morphological and kinematic results, four compelling cases were found. These are the quadrupolar PNe K3-24 and NGC 7026, and the multipolar proto-PNe M1-26 and M4-18. Since for K3-24, M1-26, and M4-18 kinematic data is not available, we could not compare our position-velocity diagrams with ones for these objects, so, to further test and constrain our model, it would be very interesting if this data could be obtained.

The method to arrive at the synthetic Hα images presented in this chapter does not include the effects that absorption may have. Absorption effects may change the resulting image for those lines of sight where the swept up shell becomes optically thick. In order to facilitate such a more detailed comparison, projection models of radiative solutions taking absorption into account are currently being calculated using a newly developed ray trace algorithm especially suited for the AMR data structures of the Flash code (see Chapter 3). Furthermore, since this chapter presents one three-dimensional result only, it will be of interest to run more models with different wind-disk parameters to provide insight into the variation in resulting morphologies.

Acknowledgements

5.A   Derivation of the cooling time

Starting with a slightly more general version of equation (13) in a paper by Kahn (1976) for the rate of loss of energy per unit mass

$$\mathsf{L} = \Lambda_i \mu_{\mathrm{H}}^{-2} \rho_s T_s^{\alpha_i} \quad (\mathrm{erg\ g^{-1}\ s^{-1}}) \ , \tag{5.13}$$

and following a similar derivation as the one given by Kahn, we find a cooling time

$$t_{\mathrm{c,i}} = \frac{3}{2(1-\alpha_i)} \frac{x_t k_{\mathrm{B}} T_s^{1-\alpha_i}}{n_{\mathrm{H}} \Lambda_i} \ . \tag{5.14}$$

Expressing the post-shock temperature T_s as a function of the shock speed v_s using the Rankine-Hugoniot jump conditions (e.g. Ostriker & McKee 1988), we find

$$T_s = \frac{v_s^2 \mu}{k_{\mathrm{B}}} \frac{2(\gamma-1)}{(\gamma+1)^2} \ , \tag{5.15}$$

and a cooling time

$$t_{\mathrm{c,i}} = C_i \frac{v_s^{2(1-\alpha_i)}}{\rho_{\mathrm{e}}} \ , \tag{5.16}$$

where

$$C_i = \frac{1}{\Lambda_i} \left[\frac{3}{2(1-\alpha_i)} \mu_{\mathrm{H}}^{2-\alpha_i} x_t^{\alpha_i} k_{\mathrm{B}}^{\alpha_i} 2^{1-\alpha_i} (\gamma-1)^{2-\alpha_i} (\gamma+1)^{-(3-2\alpha_i)} \right] \ . \tag{5.17}$$

Using the derivative of equation (5.9) for the shock velocity, the cooling time scale becomes

$$t_{\mathrm{c,i}} = C_i \left(\frac{3\dot{M}_{\mathrm{w}} v_{\mathrm{w}}}{32\pi} \right)^{\frac{1}{2}(1-\alpha_i)} \rho_{\mathrm{e}}^{-\frac{1}{2}(3-\alpha_i)} t^{\alpha_i-1} \ . \tag{5.18}$$

Using equation (5.1) as a typical disk radius, and setting $t = t_{\mathrm{sp}}$, we find that the cooling time scales as

$$t_{\mathrm{c,i}} \propto \rho_{\mathrm{e}}^{\alpha_i-2} \dot{M}_{\mathrm{w}}^{\frac{1}{2}(1-\alpha_i)} v_{\mathrm{w}}^{\frac{1}{2}(1-\alpha_i)} M_*^{2(\alpha_i-1)} L_*^{-4(\alpha_i-1)} \dot{M}_{\mathrm{acc}}^{4(\alpha_i-1)} \eta^{4(\alpha_i-1)} \ . \tag{5.19}$$

CHAPTER 6

3D simulations of knots in the Helix

We study the photoevaporation of multiple dense neutral knots with physical parameters applicable to observed knots in the Helix planetary nebula (NGC 7293). The ionizing photons from the central star of this nebula induce an ionization front at the knot. This results in a shock running through the knot, which is followed by the ionization front. We show that for the Helix knots conditions are such that the ionization front velocity is much lower than the shock velocity. This leads to a thick layer of shocked material between the two fronts, resulting in an expansion phase once the shock has left the back of the knot. We analytically derive a condition for which this expansion occurs and show that this is the same condition as used by Bertoldi (1989) to discern between two regimes in the evolution of photoevaporating clouds. This shows that the expansion phase we find in our numerical simulations is in fact quite common for this type of process. We proceed by presenting 3D radiation hydrodynamical calculations using both the parallelized Hybrid Characteristics as well as the explicitly photon conserving C^2-Ray method. After comparing these two methods for a simulation involving a single photoevaporating knot, we describe the different interactions that may occur between two of these knots. We quantify the evolution of global flow properties like the neutral mass, kinetic, and internal energy of the knots, and finish with a simulation of the photoevaporation of a whole ensemble of knots. We compare our results to observations of knots in the Helix using synthetic Hα images, explaining some of the observed features in the tail region of these knots.

<div align="right">

E.-J. Rijkhorst and G. Mellema
to be submitted to Astronomy & Astrophysics
(Section 6.5 will be submitted as a separate Letter)

</div>

6.1 Introduction

High resolution observations of the Helix (NGC 7293) planetary nebula (PN) show tens of thousands of resolved knots (O'Dell & Handron 1996; Meaburn et al. 1998; O'Dell et al. 2004; Meixner et al. 2005). These knots are known to be of high density and have extended tails pointing away from the central star (CS). Their number rapidly increases when moving radially outwards, towards the ionization front of the main nebular component, which is thought to be a thick disk (e.g. O'Dell 1998). The knots are optically thick to the incident ionizing radiation coming from the CS, which therefore induces a photo-evaporation flow emerging from the head of the knot. The associated D-type ionization front drives a shock into the dense material while slowly eating its way through the knot, and the back-reaction of the knot material to this flow accelerates the knot away from the source (Oort & Spitzer 1955).

Because of limitations in computational power and algorithmic constraints, numerical radiation hydrodynamical simulations of such photoevaporating knots traditionally focused on calculations involving a single knot only (e.g. Lefloch & Lazareff 1994; Mellema et al. 1998; García-Arredondo et al. 2001; Raga et al. 2005; O'Dell et al. 2005). These calculations quantify properties like, for example, the emission structure, velocity, and evaporation time of single knots. However, in PNe like the Helix, most knots occur in groups and show clear signs of interaction among them.

To systematically study the evolution of such a group of knots, three-dimensional high-resolution radiation hydrodynamical calculations are a requirement. First results of such simulations were presented by Lim & Mellema (2003), who studied the interaction among two photoevaporating knots. Here, we take these calculations one step further by performing numerical simulations involving an ensemble of knots.

We start with a summary of the observed properties of the Helix knots, which give us the physical parameters for our simulations, and briefly discuss the existing scenarios for the origin of the knots (Section 6.2).

Section 6.3 summarizes the numerical methods used to run the simulations. We use both the newly developed photon conserving C^2-Ray method (Mellema et al. 2005), and the Hybrid Characteristics method which was presented in Chapter 3. Since these methods contain different implementations for the hydrodynamics, ray tracing and computational mesh, we run some test problems and compare the results obtained. This section also addresses the numerical issues of spatial and temporal conservation of photons, important for the outcome of the simulations.

Section 6.4 describes the assumptions and approximations that went into the simulations.

In Section 6.5 we analyze the evolution of a single photoevaporating knot. We show that the condition that determines the thickness of the layer of material between the ionization front and the preceding shock is important for the way in which the knot evolves. We place this condition in the context of the parameter space for photoevaporating knots as introduced by Bertoldi (1989), and show that this is a common and therefore important effect.

We continue with a description of a baseline calculation involving a single knot in Section 6.6. The values of the physical parameters specifying the Helix knots are different

from the ones normally used in this kind of simulations. This significantly alters the way in which a knot evolves, which is discussed in detail.

Section 6.7 presents simulations involving two knots, showing the different types of interaction that can occur among them, and the impact these have on the evolution of the knots.

In Section 6.8 the results for the photoevaporation of an ensemble of knots are given, which we discuss and compare to observations of the Helix knots in Section 6.9.

Conclusions are given in Section 6.10.

6.2 Observed properties of the Helix knots

To set the stage for the radiation hydrodynamical simulations presented in this chapter, this section discusses observed properties of the knots in the Helix nebula, and some of the scenarios that may explain their origin.

At a distance of 213 pc (Harris et al. 1997), the Helix is the nearest PN. Its CS has an extremely high temperature of $123,000 \, \mathrm{K}$ (Bohlin et al. 1982), and a luminosity of $120 \, \mathrm{L_\odot}$. The mass of the progenitor of the Helix nebula is thought to be $\sim 6.5 \, \mathrm{M_\odot}$ (Henry et al. 1999). The main nebular component is a geometrically thick disk, which is ionized up to a radius of $210''$, and has a symmetry axis that is tilted away from the line of sight by $\sim 23°$ (O'Dell et al. 2004). Embedded in this disk, starting at a radius where the transition from a highly ionized, HeII emitting central region, to a lower ionization zone emitting HeI takes place, tens of thousands of dense knots of $\sim 1''$ in diameter are observed (O'Dell et al. 2004; Meixner et al. 2005). These knots rapidly increase in number when moving outwards from the source, towards the ionization front.

The side of a knot that faces the star is ionized and, due to limb brightening, is observed as a cusp-like shape in Hα and [NII] images. The core region of the knots on the other hand is observed in absorption against the bright nebular background due to dust extinction (Meaburn et al. 1992). This absorption increases when the knot lies relatively close to the surface of the disk. The amount of extinction observed in [OIII] thus gives a lower limit for the mass of the knots, which is approximately $\sim 10^{-5} \, \mathrm{M_\odot}$, and a density of $\sim 10^5 \, \mathrm{cm^{-3}}$ (O'Dell & Handron 1996; Meaburn et al. 1998). This assumes a dust to gas ratio in the knot of $\sim 1/150$, similar to that of the general interstellar medium. The Helix is an oxygen rich PN, and any dust emission is therefore assumed to arise primarily from silicates (Speck et al. 2002). The density of the ionized environment is of the order of $\sim 50 \, \mathrm{cm^{-3}}$ (O'Dell & Handron 1996).

CO observations by Huggins et al. (2002) of one of the more prominent knots in the Helix [designated 378-801 in the coordinate system introduced by O'Dell & Burkert (1997)], show a main peak that is associated with the core of the knot, and a second peak located at the edge of the tail region behind the core, where the tail itself shows extended (but weaker) CO emission as well. The observed line width of this CO emission is slightly broader ($\sim 0.8 \, \mathrm{km \, s^{-1}}$) in the tail region than for the knot itself ($\sim 0.5 \, \mathrm{km \, s^{-1}}$) (Huggins et al. 2002). The CO excitation temperature is ~ 25 K, and assuming a CO abundance of 3×10^{-4}, observations of the CO flux give a molecular mass of at least $10^{-5} \, \mathrm{M_\odot}$ (Huggins et al. 2002), consistent with the value found from dust extinction. Velocity measurements

obtained from CO observations of the knots give a kinematic age of the main nebular component of $5,000 - 10,000$ yr depending on the adopted model (e.g. Young et al. 1999; O'Dell et al. 2004).

The H_2 emission of the knot 378-801 is observed as a limb-brightened cusp to lie just within the cusp observed in $H\alpha$+[NII], and is interpreted to be a thin photo-dissociation region (Huggins et al. 2002). The tail of this particularly knot also shows enhanced H_2 emission near two peaks in $H\alpha$+[NII] (Huggins et al. 2002), which is thought to be due to the remains of a second knot that is partially shadowed from direct stellar photons by 378-801 (O'Dell et al. 2005).

A number of scenarios have been proposed to explain the origin of the Helix knots (for a summary, see O'Dell & Handron 1996). One of the options is an instability at the ionization front, for example as a Rayleigh-Taylor instability (Capriotti 1973), or a shadowing instability due to small inhomogeneities in the interstellar medium (Williams 1999). Another option is instabilities due to successive stellar winds as described by García-Segura et al. (1996). The knots may also originate from SiO maser spots in the atmosphere of the red giant progenitor. Dyson et al. (1989) considered this possibility by studying their survival during the successive mass loss phases of the CS. Finally, the knots may also have a primordial origin (see O'Dell & Handron 1996, and references therein). Since it is far from clear which of these scenarios is responsible for the origin of the knots, we chose the initial shape of the knots in our simulations to be spherical, although in reality this will probably not be the case.

This chapter focuses on the knots in the Helix since this nebula has been observed extensively, and many physical properties of the knots and the nebula are known. However, knots similar to the ones in the Helix have been observed in the nearby PNe NGC 2392 (Eskimo), NGC 6720 (Ring), NGC 6853 (Dumbbell), and IC 4406 as well (O'Dell et al. 2002), showing this type of knots to be a common feature in PNe. This makes them an important ingredient of the material that is fed back into the interstellar medium, and insight into the processes that create and shape the knots is required.

6.3   Numerical methods

In this section we briefly discuss the two numerical methods we use for the photoevaporation simulations presented in this chapter. These are the Hybrid Characteristics method (Chapter 3) as implemented in the *Flash* code (Fryxell et al. 2000), and the recently developed C^2-Ray method (Mellema et al. 2005).

The main differences between these codes are the hydrodynamics solver, the method of ray tracing, and the mesh utilized. The *Flash* code uses a PPM type hydrodynamics solver (Colella & Woodward 1984) on an adaptive mesh, while the C^2-Ray method employs a Roe type solver (Roe 1986) on a regular mesh (although this method is capable of handling AMR as well). However, the main difference between the methods lies in the implementation of ray tracing and ionization routines.

The Hybrid Characteristics method uses distributed ray tracing as described in Chapter 3, making it a fully parallel code, whereas the C^2-Ray method traces rays using short

characteristics which is an intrinsically serial method. This has implications for the implementation of photon conservation, which is discussed next.

The C^2-Ray method by Mellema et al. (2005) represents a novel approach for numerically calculating the photo-ionization and heating rate on a discretized computational domain that is explicitly photon conserving. It implements spatial conservation by rewriting the hydrogen photo-ionization and heating rate in terms of the optical depth τ_ν to the cell and the optical depth $\Delta\tau_\nu$ over the cell.

Since during a time step Δt the ionization fraction in a cell actually changes, one should correct for this as well. The usual solution is to take a larger number of smaller time steps to approximate photon conservation (Abel et al. 1999; Shapiro et al. 2004). However, Mellema et al. (2005) propose a different solution to this problem of temporal conservation of photons by introducing a time averaged ionization fraction. This is used to find a time averaged optical depth for the cell, thereby lifting the restriction on the time step with which the calculation has to evolve. This time averaged optical depth is also used to calculate the ionization and heating rates.

Another crucial aspect of the C^2-Ray method is that the order of the cells through which the rays are traced is important. Since the ionization and heating rates in cells that lie further away from the source depend on the ionization fraction of cells that lie closer to the source, one should start at the source and trace outwards to stay photon conserving. As already mentioned, this interdependency makes their, and any other photon conserving method that employs such a causal form of ray tracing, intrinsically serial.

As explained in Chapter 3, the Hybrid Characteristics method takes a different approach to ray tracing. Instead of starting at the source and tracing outwards, the ray is split up into sections according to the distribution of the AMR patches over the different processors. Partial column densities along these sections are determined by each processor, after which they are communicated and accumulated to arrive at the total column density for each cell. Through this process, parallelization of ray tracing for distributed, decomposed computational domains becomes possible.

Unfortunately, this way of dealing with column densities makes it difficult to combine the method of Mellema et al. (2005) with the Hybrid Characteristics method in a consistent way. Since, for the Hybrid Characteristics method, column densities are *first* determined for *all* cells, and ionization fractions are updated *afterwards* (again for all cells), this means that for this method photons can only be conserved spatially but not temporally.

In Section 6.6.3 we compare the two methods to quantify the error made when using the Hybrid Characteristics for the specific case of photoevaporation simulations of the Helix knots.

6.4 Assumptions and approximations

This section discusses the assumptions and approximations used in our simulations of photoevaporating knots. We start with a description of the implementation of the radiative cooling, followed by a discussion of the effects stellar evolution may have. Although the composition of the gas in our simulations is hydrogen only, we do account for the

presence of helium in an approximate way. We also briefly discuss the influence the diffuse radiation field may have on the outcome of the simulations.

6.4.1   Radiative cooling

In PNe the main cooling processes are energy loss through recombination, collisionally excited line radiation, and bremsstrahlung. In our simulations, radiative cooling is approximated by using the collisional equilibrium cooling curve Λ^* from Dalgarno & McCray (1972). From this, the cooling rate as a function of temperature follows as

$$\Lambda(T) = n_e n_H \Lambda^*(T) \qquad [\mathrm{erg\,cm^{-3}\,s^{-1}}] , \qquad (6.1)$$

The cooling curve was calculated by assuming that all ionizations are collisional, which is certainly not the case in the photoionized regions in our numerical models. The way in which we approximate radiative cooling in our simulations is quite crude in comparison with simulations performed by for example Mellema et al. (1998), who employ non-equilibrium cooling for a large number of ions. Furthermore, inside the knots, molecular cooling may be of importance as well, which we neglect in our current models.

6.4.2   Stellar evolution and stellar wind

The mass of the central star of the Helix is estimated to be ~ 0.93 M$_\odot$ (Gorny et al. 1997). Together with a kinematic age of the main nebular component of $\sim 10,000$ yr (e.g. Young et al. 1999; O'Dell et al. 2004), shows that the central star is in a late phase of its evolution for which the luminosity and effective temperature do not change much (Blöcker 1995) during the time span of our simulations of ~ 1500 yr. Therefore, in our models, we do not take the stellar evolution into account, but instead keep the ionizing stellar flux constant throughout the simulation. Furthermore, although in earlier stages of the evolution of the Helix a stellar wind must have been present and probably played an important role in shaping the knots (see Section 6.2), the current evolved state of the Helix shows no observational evidence for such a wind (Cerruti-Sola & Perinotto 1985), so we do not include it here.

6.4.3   Influence of helium

The knots in the Helix are known to lie just outside the central nebular region that shows pronounced HeII radiation (O'Dell et al. 2004). This means that the spectrum of the radiation that arrives at the knots is softer than the stellar spectrum, since all photons with energies above 54.5 eV were used to ionize He+ to He++ (e.g. Osterbrock 1989).

   Although photons resulting from recombinations from He++ to He+ have enough energy to ionize HI and should therefore be taken into account when calculating the ionization fractions in the He+ region surrounding the central He++ core, we neglect this effect here. Instead, we simply integrate the stellar spectrum up to a frequency corresponding to 54.4 eV, and our results for the evaporation time given below therefore represent an upper limit. The number of ionizing photons per second in this case is $S_* = 7.1 \times 10^{45}$ s^{-1}.

Upon comparison of simulations run with and without high energy photons, we find that the knots takes about 10% longer to evaporate in the latter case.

6.4.4 Diffuse field

Our calculations ignore the diffuse field, which arises mainly from the recombinations of hydrogen, with a smaller contribution due to helium and the scattering of stellar photons by dust grains. The diffuse field may be of importance since it is expected to (partly) ionize the shadow region behind the knots. An assessment of this effect was made by Canto et al. (1998), who, using the 'on the spot' approximation, found that the diffuse flux is $\sim 15\%$ of the direct stellar flux.

Recently Ritzerveld (2005) showed that, when one drops the 'on the spot' approximation, the direct stellar field actually decreases in strength in favour of the diffuse field when moving outwards from the source. Effectively, more and more direct stellar photons get 'converted' into diffuse photons when moving outwards, so that at some radius r_{eq} the diffuse field starts to dominate the direct stellar field. Ritzerveld (2005) also showed that this effect only slightly diminishes for a hard direct stellar spectrum. Furthermore, the fact that, in the Helix' case, the knots are embedded in a thick disk instead of a sphere, will slightly increase the diffuse flux, but only near the surface of the disk (Ritzerveld, priv. comm.).

For a homogeneous density distribution, which we assume the disk in which the knots are embedded to have, the radius r_{eq} at which the diffuse radiation field becomes equal to the direct stellar field is given by Ritzerveld (2005) as

$$r_{\mathrm{eq}} = R_{\mathrm{S}} \left(1 - 2^{\alpha_{\mathrm{B}}/\alpha_1}\right)^{1/3} \simeq 0.88 R_{\mathrm{S}} \,, \tag{6.2}$$

with R_{S} the Strömgren radius, α_{B} the hydrogen recombination coefficient to all but the ground state, and α_1 the hydrogen recombination coefficient to the ground state only.

Although the inclusion of ionization effects due to helium may slightly change this result, we use this value for r_{eq} as an estimate for the radius at which the diffuse field starts to become important. So for the knots in our simulations, which lie at $\sim 35\%$ of the Strömgren radius, the diffuse field can be safely ignored, since at this radius, the diffuse flux is $\lesssim 10\%$ of the direct stellar flux when we assume a homogeneous density distribution.

6.5 The expansion phase of photoevaporating clouds

In this section we describe the evolution of the photoevaporation of a single cloud[1]. Analytical (Bertoldi 1989; Bertoldi & McKee 1990) and numerical (Lefloch & Lazareff 1994; Mellema et al. 1998) work has shown that this evolution consist of two distinct phases: the collapse phase where an ionization fronts eats its way into the cloud, driving a shock

---

[1]This section uses the term 'cloud' instead of 'knot', since its results apply to more general cases than just to the Helix knots.
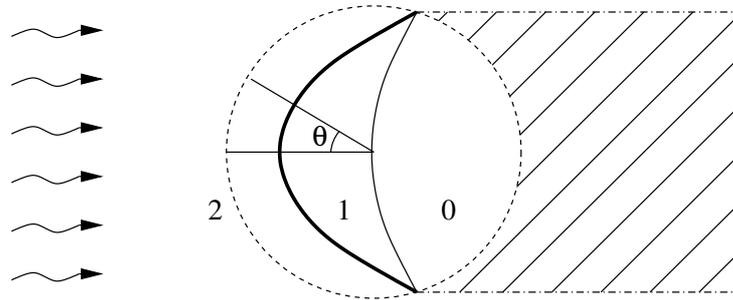
Figure 6.1: Schematic depiction of the cloud. Ionizing radiation is coming from the left, creating an ionization front (thick solid line) and a shock front (thin solid line). The original extent of the cloud is indicated by the dashed circle. Three different regions can be discerned: region 0 contains unshocked, neutral cloud material. Region 1 contains shocked, neutral cloud material. Region 2 is ionized and contains the photoevaporation flow. A shadow region in which no direct stellar photons can penetrate is indicated by the hashed region to the right of the cloud.

which compresses it, followed by the cometary phase, during which the cloud gets accelerated through the 'rocket effect' (Oort & Spitzer 1955) while being photoevaporated.

Although elaborate descriptions of these phases have been given before (Bertoldi 1989; Bertoldi & McKee 1990; Lefloch & Lazareff 1994; Mellema et al. 1998), the parameters we use for our simulation are tailored to the Helix nebula, which significantly alters the way in which the cloud evolves.

6.5.1   Compression phase

At the start of the evolution of the cloud it is assumed that the source has just switched on. Its ionizing radiation heats a thin layer of gas at the hemisphere of the cloud that faces the source, which strongly raises the thermal pressure in that layer, and a spherical shock is driven into the cloud. This shock is followed by the ionization front which eats its way into the cloud, ionizing it along the way (see Figure 6.1).

Previous calculations assumed that the layer of shocked gas between the shock and the ionization front is *thin* (Bertoldi 1989), or, in other words, the ionization front's velocity is only slightly smaller than the shock velocity. This means that, during the period of time the shock needs to travel all the way through the cloud, the ionization front moves through a significant part of the cloud as well, shaping the cloud into its distinctive parabolic (or cometary) form.

The change of shape of the ionization front from spherical to parabolical causes the direction of flow to get focused towards the axis of symmetry, to a region somewhere beyond the initial centre position of the cloud (cf. Bertoldi 1989; Lefloch & Lazareff 1994). By the time the shock has traveled all the way through the cloud, most of the initial mass has been photoevaporated, and an over-pressured, dense core, containing typically

$\sim 10\%$ of the original mass, is left.

In the case of more dense clouds and a lower ionization flux from the source, like is the case for the knots in the Helix nebula, the evolution during the initial compression phase proceeds differently. Because of the relatively high density of the cloud, in combination with the relatively low ionizing flux of the source, the velocity of the ionization front is significantly lower than the shock velocity. This means that the shocked layer between the ionization front and the shock is *not* thin. On the contrary; by the time the shock has traveled all the way through the clump, only a small fraction of $\sim 20\%$ of its original mass has been photoevaporated.

The fact that the ionization front does not closely follow the shock also means that the latter does not get driven towards the axis of symmetry, but instead moves along it. In these cases it is therefore better to speak of a compression, rather than of a collapse phase.

6.5.2   Shocked mass after compression

An analytical estimate of the amount of shocked mass M_1 that is left at the end of the compression phase relative to the initial mass M_0 can be calculated (see Mellema et al. 1998)

$$\frac{M_1}{M_0} = 1 - \frac{F}{n_0 v_s} \,, \tag{6.3}$$

where F is the flux of ionizing photons reaching the ionization front at the symmetry axis, n_0 is the initial number density of the cloud, and v_s is the shock velocity. In the case of a strong isothermal shock (e.g. Zel'Dovich & Raizer 1967),

$$v_s = \left(\frac{P_1}{\rho_0}\right)^{1/2} \,, \tag{6.4}$$

with P_1 the post-shock pressure, and ρ_0 the initial mass density of the cloud.

Assuming hydrostatic equilibrium, and using the fact that the ionization front is of D-critical type (Kahn 1954), the pressure driving the shock is given by

$$P_1 = 2Fmc_2 \,, \tag{6.5}$$

with $m \simeq 1.3m_H$ the average mass per atom. Using equations (6.4) and (6.5) in (6.3), one finds

$$\frac{M_1}{M_0} = 1 - \chi \,, \tag{6.6}$$

with the parameter χ defined by

$$\chi \equiv \left(\frac{F}{2c_2 n_0}\right)^{1/2} \,. \tag{6.7}$$

The photon flux F depends on the number of recombinations taking place in the region between the source and the cloud, including the recombinations taking place in the photoevaporation flow, and can be approximated by (e.g. López-Martín et al. 2001)

$$F(\theta) \simeq \frac{\xi}{2}\left[\left(\frac{4F_* \cos(\theta)}{\xi} + 1\right)^{1/2} - 1\right] \,, \tag{6.8}$$
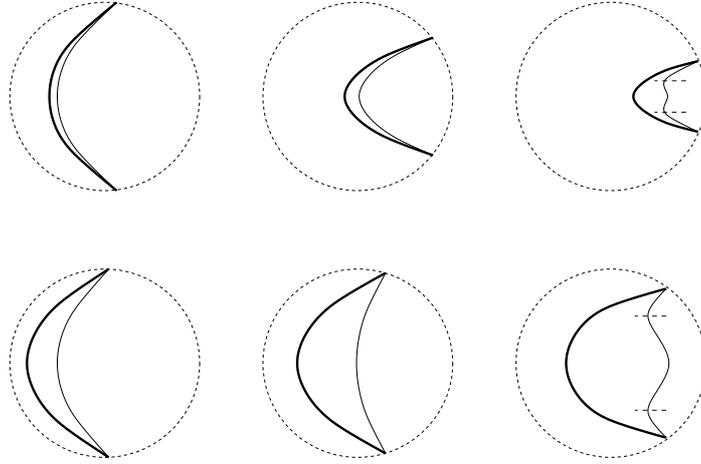
Figure 6.2: Schematic depiction of the evolution of the shape of the shock and ionization front for the case of a thin (top row) and a thick (bottom row) layer of shocked material between the shock and the ionization front. Time increases from left to right. The dashed circle shows the initial extent of the cloud, the thick line the ionization front, and the thin line the shock. The dashed line segments shown in the images on the right represent the ring-like structure of enhanced density where the flow converges (see also Figure 6.3).

with $\xi \equiv c_2^2/(r_0 \omega \alpha_B)$ and θ as defined in Figure 6.1. Here, c_2 is the sound speed of the ionized material, r_0 the initial radius of the cloud, $\omega \simeq 0.1$ a constant which depends on the assumed density profile of the photoevaporation flow (e.g. Henney & Arthur 1998), and α_B the hydrogen recombination coefficient for recombinations to all but the ground level.

6.5.3   Shape of the shock

As mentioned above, the shape of the shock during the compression phase is determined by the evolving shape of the ionization front. When the layer of shocked material between the ionization front and the shock is thin, the shock gets focused towards the symmetry axis, whereas for a thick layer this focusing effect is less severe. Figure 6.2 schematically shows the typical evolution for the two cases.

The shock velocity of equation (6.4) depends on the photon flux of equation (6.8) at the ionization front through equation (6.5) for the pressure. Since the flux is a function of the angle θ, the shock velocity also depends on this angle. The shock velocity decreases with increasing θ, which means that the section of the shock close to the symmetry axis outruns the section that is close to the edge of the cloud. This leads to a gradual distortion of the shape of the shock as shown in Figure 6.2, which at some point becomes so severe that it produces a kink. Cloud material passing through the shock acquires a velocity in the direction of shock propagation, and therefore accumulates in a 'ring' behind the kink. This effect is illustrated in more detail in Figure 6.3. From the numerical simulations presented below we find that the density in the 'ring' is a factor ~ 2 higher than the
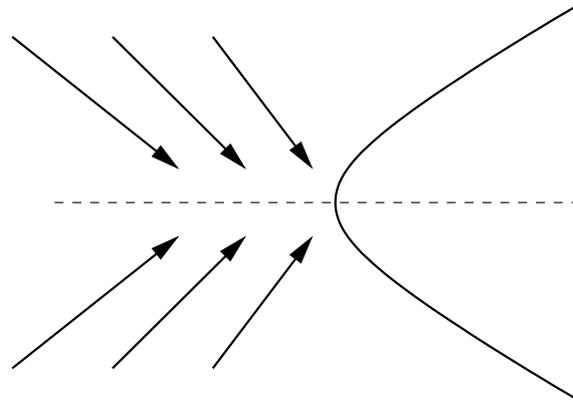
Figure 6.3: Detail of the distorted shape of the shock front (solid curved line). After passage of the bent shock, the cloud material acquires a velocity perpendicular to it (arrows), which, behind the kink, leads to the accumulation of material in a ring-like structure (dashed line, corresponds to the dashed line segments of Figure 6.2).

density at the symmetry axis.

6.5.4   Expansion and cometary phase

The compression phase ends when the shock front leaves the back side of the cloud. While the shock was traveling through the cloud, the high pressure layer was confined on one side by the ionization front and on the other by the shock. Now that the shock has left the cloud, this pressure gets released and a rarefaction wave travels from the rear of the cloud towards the ionization front, where the ring of high density produced in the compression phase acts as a 'channel' through which this wave travels. This greatly reduces the density of the cloud within the region contained by the ring, which expands in a direction along the symmetry axis, away from the source. The thicker the layer of high pressure material left at the end of the compression phase, the more important the expansion phase becomes for the final stages of the evolution of the cloud.

When the rarefaction wave arrives at the ionization front, the drop in density changes the conditions at the front from D-critical to M-type, and a second shock is formed which quickly travels through the remains of the cloud, accelerating it a second time. This effect was predicted by Kahn (1954), and noticed before in numerical simulations by Lefloch & Lazareff (1994), who identified it as radial oscillations.

When the oscillations have been damped, the cloud enters what is often referred to as the cometary phase. During this phase, the cloud is accelerated in its entirety due to the 'rocket effect' (Oort & Spitzer 1955), and can reach a considerable velocity ($\sim 10\,\mathrm{km\,s^{-1}}$) before it is completely photoevaporated.

The thickness of the pressurized layer between the ionization front and the shock, as parametrized by the factor χ of equation (6.7), determines the shape the cloud acquires during the expansion phase. For a thin layer (i.e. $\chi > 1/2$), most of the cloud material is evaporated during the compression phase, and a compact core of shocked material is left. When the ionization front moves fast enough, shocks traveling inwards from the sides of the cloud converge on the symmetry axis before the part of the shock traveling along this axis can leave the rear side of the cloud. Also, the ring structure hardly gets a chance to develop, since the converging ionization front quickly destroys it.

For clouds that develop a thick layer of shocked gas (i.e. $\chi < 1/2$), the ring structure has enough time to grow. Also, in this case, the shock does not get focused onto the axis and can travel all the way through the cloud, leading to a significant expansion once it has left it. Apart from expanding in a direction away from the source, the cloud also expands laterally, and swells up to a size comparable to its initial dimensions. This increases the surface area exposed to the ionizing radiation, which accelerates the destruction of the cloud.

So, in summary, the evolution of the cloud is dominated by the evolving shape of the ionization front when $\chi > 1/2$, and by the shape of the shock when $\chi < 1/2$.

6.5.5   Cloud parameter space

Bertoldi (1989) defines a parameter space for photoevaporating clouds spanned by two dimensionless parameters η and ν, which, in our notation, are given by

$$\eta \equiv \frac{\alpha_{\mathrm{B}} r_0 n_0}{c_2} \; , \tag{6.9}$$

and

$$\nu \equiv \frac{v_{\mathrm{s}0}}{c_2} = \left(\frac{2 F_*}{n_0 c_2} \right)^{1/2} \; , \tag{6.10}$$

where $v_{\mathrm{s}0} \equiv v_{\mathrm{s}}(F = F_*)$ is the initial shock velocity. The parameter η is a measure for the initial column density of the cloud, whereas ν represents the dimensionless initial shock velocity.

In Figure 6.4 we reproduce the parameter space of Figure 1 from Bertoldi (1989). The regions where photoevaporation effects are important are region II and III, for which conditions in the cloud are initially M-type. As explained by Bertoldi (1989), for clouds in region II, a shock precedes the D-critical ionization front, and the shock propagates faster through the cloud than the ionization front. The transition from region II to III is determined by the condition

$$v_{\mathrm{s}} = c_2 \; , \tag{6.11}$$

so for clouds in region III, the shock velocity would exceed c_2. Since this is physically not possible (cf. Bertoldi 1989), the conditions in the recombining photoevaporation flow will be such that the shock velocity is limited to c_2.

Conditions for clouds in region I are of weak D-type and they are photoevaporated only slowly, without the occurrence of a shock. Conditions for clouds in region IV are
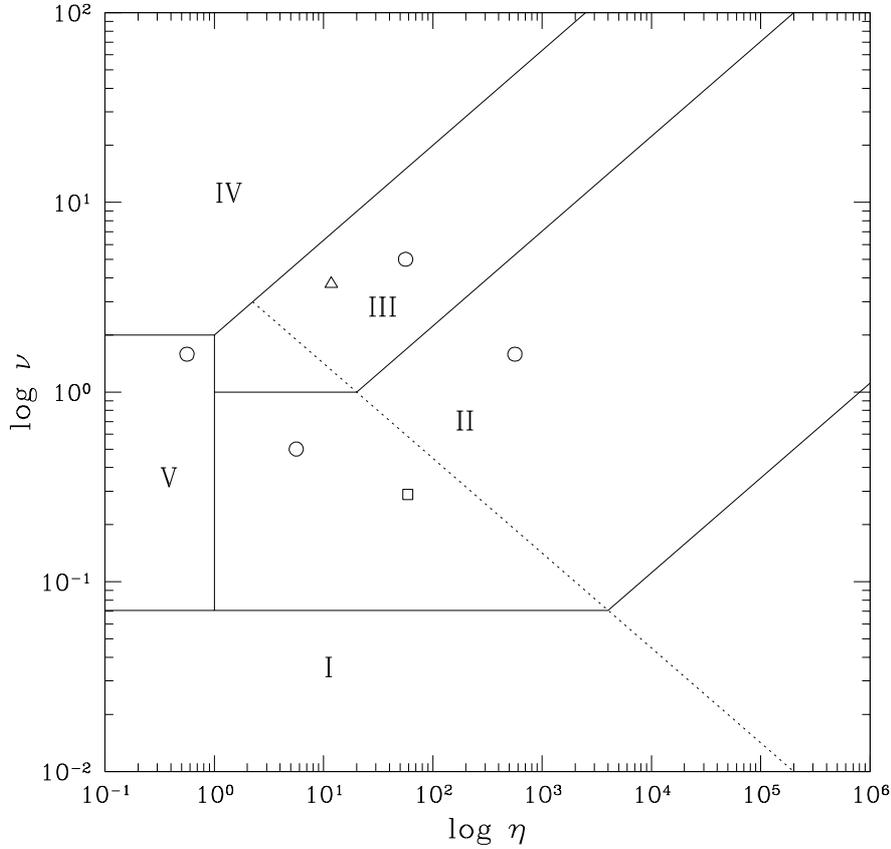
Figure 6.4: Reproduction of Figure 1 of Bertoldi (1989) showing the parameter space for photoevaporating clouds in terms of the dimensionless parameters η and ν. For regions II and III, the ionization front drives a shock into the cloud. The dashed line represent $\psi = \omega^{-1}$. Circles indicate the different models discussed by Bertoldi (1989), the triangle is the model by Mellema et al. (1998), and the square denotes model 1K of this chapter.

R-type, and they are 'zapped' instantaneously. Clouds in region V have such a low initial column density that a steady state photoevaporation flow can not develop.

Bertoldi (1989) also introduced a photoevaporation parameter ψ, defined by

$$\psi \equiv \frac{\alpha_{\mathrm{B}} F_* r_0}{c_2^2} = \nu^2 \eta / 2 \;, \tag{6.12}$$

and a parameter q, defined by

$$q \equiv F_*/F \; (q \geq 1) \;. \tag{6.13}$$

Through these parameters, two regimes are identified. When recombinations in the photoevaporation flow are not important one has

$$q = 1 \qquad \text{for all } \psi < \omega^{-1}, \tag{6.14}$$

whereas when recombinations in the photoevaporation flow are important it follows that

$$q(q - 1) \simeq \omega\psi \qquad \text{for all } \psi > \omega^{-1}, \tag{6.15}$$

where $\omega \simeq 0.1$ (cf. Bertoldi 1989, Section IIIc). Following Bertoldi (1989), we approximate (6.15) by $q \simeq (\omega\psi)^{1/2}$, and using equation (6.12) in equation (6.15) we find

$$q \simeq \nu(\omega\eta/2)^{1/2} \qquad \text{for all } \psi > \omega^{-1}. \tag{6.16}$$

Using equation (6.13) one can rewrite (6.10) as [see also equation (6.4) and (6.5)]

$$\nu = \frac{v_s}{c_2} q^{1/2} \, , \tag{6.17}$$

and using condition (6.11), we find

$$\nu = 1 \qquad \text{for all } \psi < \omega^{-1}, \tag{6.18}$$

and

$$\nu \simeq (\omega\eta/2)^{1/2} \qquad \text{for all } \psi > \omega^{-1}. \tag{6.19}$$

Equation (6.18) together with equation (6.19) represent the line dividing region II from region III in Figure 6.4.

We now return to the parameter χ as defined by equation (6.7). Comparing this equation to equation (6.10), we find that

$$\nu = 2\chi q^{1/2} \, . \tag{6.20}$$

So condition (6.11) together with equation (6.17) gives

$$\chi = 1/2 \, , \tag{6.21}$$

which was shown above to represent the transition from a thick to a thin layer of shocked gas between the ionization front and the shock.

In other words, the condition that divides the parameter space of Figure 6.4 into region II and III, is the *same* condition that determines whether a thick or a thin layer of shocked gas develops between the ionization front and the shock. This means that *all photoevaporating clouds in region II develop a thick layer of shocked material*, and therefore are expected to show a strong expansion when the shock leaves the back of the cloud.

6.6   Evolution of a single knot: baseline calculations

In this section we describe a numerical model for the photoevaporation of a single knot that we use as a baseline calculation to which we compare our photoevaporation simulations of multiple knots presented in subsequent sections. All parameters were chosen such as to match the physical conditions observed in the Helix nebula as closely as possible.

Table 6.1: Parameters for the single knot (1K) baseline model.

$n_{\rm e}$ [cm^{-3}]	$T_{\rm e}$ [K]	$n_{\rm k}$ [cm^{-3}]	$T_{\rm k}$ [K]
50	10,000	5×10^4	20

6.6.1 Initial conditions

Here, we describe the initial conditions for the baseline calculation presented in Section 6.6.2 below, which is performed with the Hybrid Characteristics method. A comparison of this method with the photon conserving C^2-Ray method is presented in Section 6.6.3.

Since the mechanism from which the knots originate is unknown (cf. Section 6.2), we take the initial shape of the knots to be spherical, although it is likely that in reality the knots have a more complex initial shape. For the same reason we use a homogeneous distribution for the initial mass density of the knot. The knot has a radius of 5×10^{15} cm and is located at a distance of 5×10^{17} cm from the source.

In case of the Helix, the source has a luminosity of $L_* = 120\ L_\odot$ and an effective temperature of $T_* = 120,000$ K, with a black-body spectrum that is cut off at 54.4 eV to account for the effects of helium (see Section 6.4). This results in a rate of $S_* = 7.1 \times 10^{45}$ ionizing photons per second.

Simulations performed with the Hybrid Characteristics method use a computational domain of size $(56, 4, 4) \times 10^{16}$ cm, so that it contains the source as well as the knot. The base level of the AMR hierarchy of patches consists of a total of $(224, 16, 16)$ cells, and three additional levels of refinement give it an effective resolution of $(1792, 128, 128)$ cells. For the simulations done with the C^2-Ray method we employ a regular grid containing $(256, 128, 128)$ cells and a physical size of $(8, 4, 4) \times 10^{16}$ cm, where the left boundary is located at a distance 48×10^{16} cm from the source. These choices for the resolution result in a knot that is resolved by 32 cells.

Initial values for the density and temperature of the knot and the environment are given in Table 6.1. The density and temperature are chosen such that the knot and the environment are initially in pressure equilibrium. To prevent numerical complications, the outer 5% of the knot is given a linear slope to ensure a more gradual transition between the environment and the knot.

6.6.2 Single knot simulations

In Figure 6.5 we illustrate the evolution of a single photoevaporating knot for the case of model 1K obtained with the Hybrid Characteristics method.

The compression phase of the knot, shown on the left in Figure 6.5, ends at around ~ 500 yr when the shock leaves the back of the knot. By this time, about 20% of the knot's material has evaporated, and the shock has compressed the material by a factor of ~ 3, leading to a density of $\sim 1.5 \times 10^5$ cm^{-3} and a temperature of ~ 650 K along the symmetry axis, and a density of $\sim 3 \times 10^5$ cm^{-3} and a temperature of ~ 250 K at the outer edge of the knot. At the end of the compression phase, the knot has a velocity of ~ 3 kms^{-1}. Note that the values for the density and temperature are only indicative of actual values because of the approximate cooling and the hydrogen only composition of
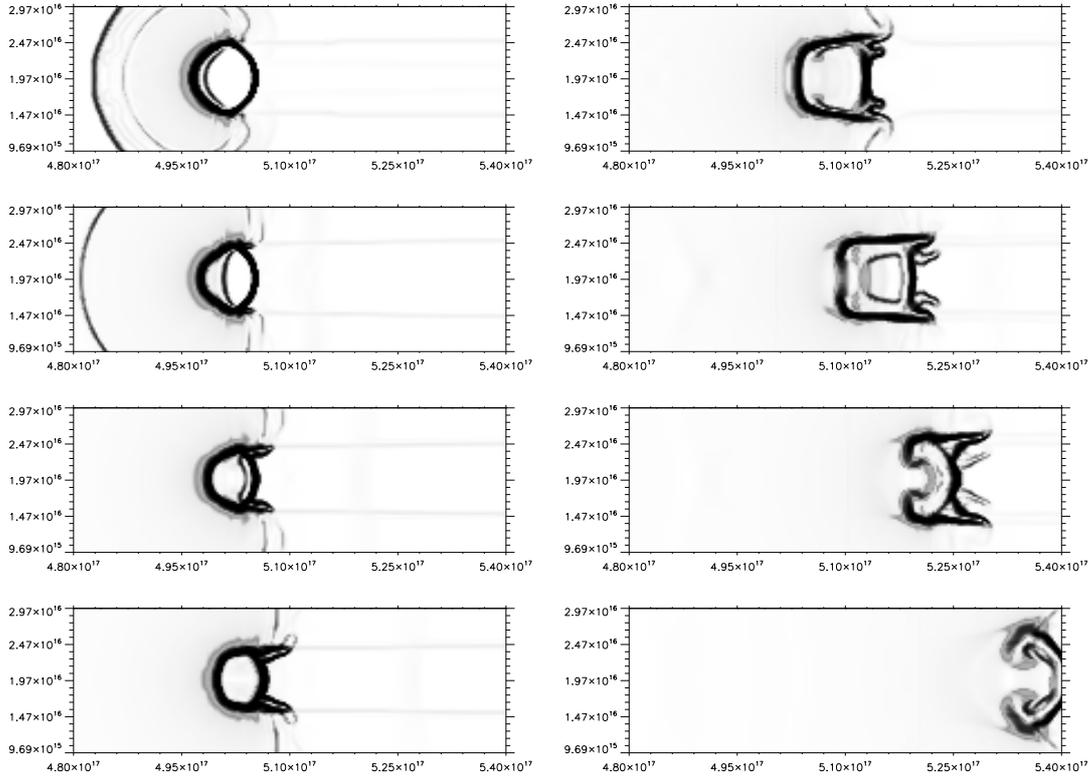
Figure 6.5: Evolution sequence of the compression phase (left) and the expansion phase (right) for a single knot. Shown are xy-cuts through the centre of the computational domain of Schlieren plots of the $\log_{10}$ of the density. In a Schlieren plot, the first derivative of a quantity is plotted in order to bring out steep gradients in the flow. In the left column, time increases in equal intervals from $t = 125$ yr (top) to $t = 500$ yr (bottom). In the right column, time increases in equal intervals from $t = 750$ yr (top) to $t = 1500$ yr (bottom). (See text for more details).

the gas. Also note the evolution of the shape of the shock, which is in accordance with the description of Section 6.5.

The expansion phase is shown on the right of Figure 6.5. The expansion lowers the density of the knot along the symmetry axis to $\sim 5 \times 10^3$ cm^{-3} at ~ 1000 yr. At this late stage in the evolution, approximately 40% of the initial knot material has evaporated.

6.6.3   Global flow properties and photon conservation

As mentioned above, the evolution of a photoevaporating knot consists of several phases. This section discusses these phases in terms of some global properties of the flow, and the importance of photon conservation for the evolution of these quantities.

In Figure 6.6, we show the results of a comparison between the C^2-Ray method by Mellema et al. (2005), which conserves photons spatially as well as temporally, and the Hybrid Characteristics method with spatial photon conservation only. The top row of
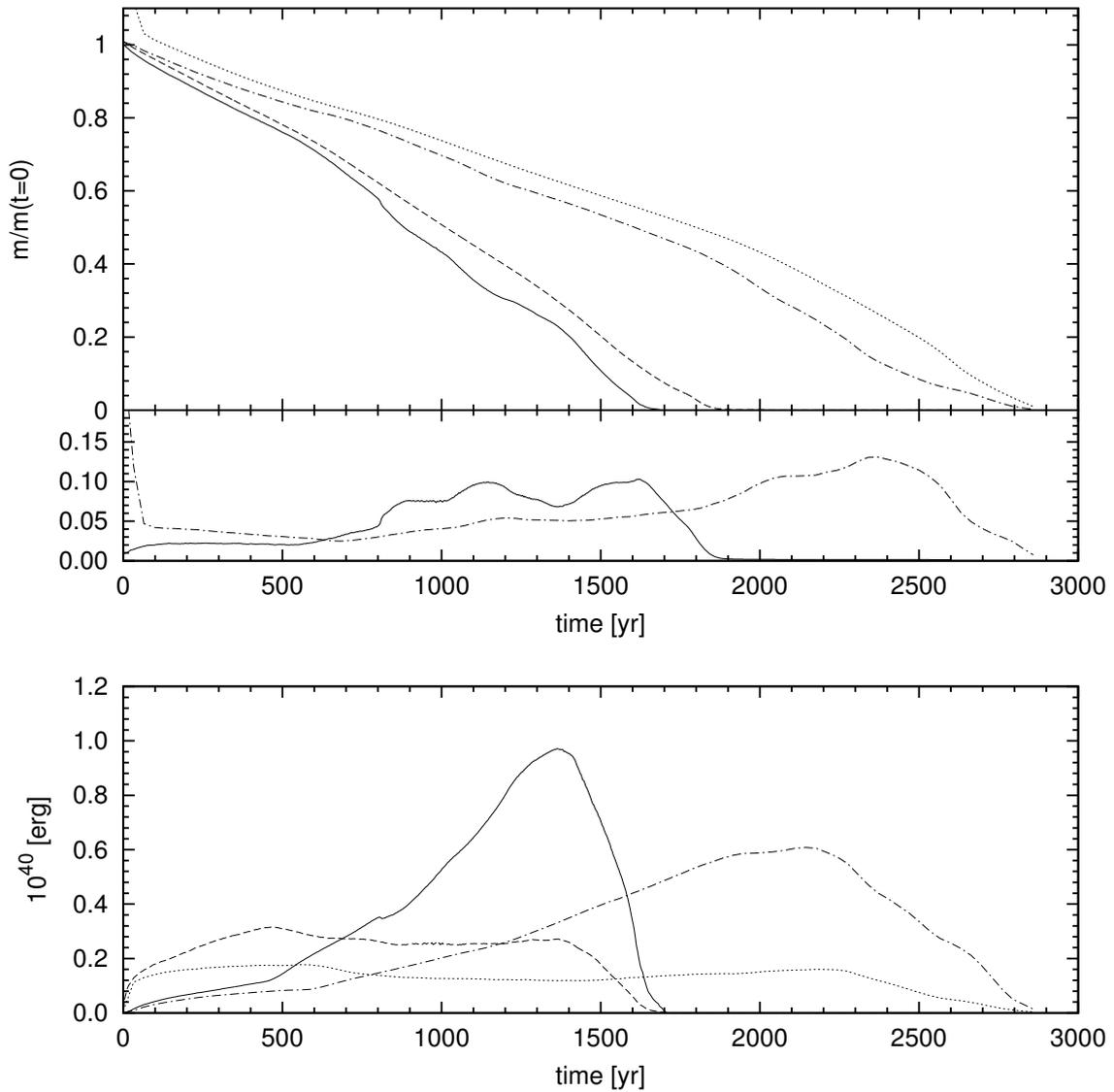
Figure 6.6: Global flow properties of the 1K baseline calculation as a function of time for both the Hybrid Characteristics method (solid and dashed lines), and the C^2-Ray method (dashed-dotted and dotted lines). The top panel shows the neutral knot mass (solid and dashed-dotted lines), and the total neutral mass (dashed and dotted lines), with the difference between these two quantities plotted beneath. The bottom panel shows the kinetic (solid and dashed-dotted lines) and internal (dashed and dotted lines) energy of the knots.

Figure 6.6 shows the neutral knot mass m_{k} and the total neutral mass m_{tot} as a function of time, beneath which we show the difference between these two quantities, $m_{\text{tot}} - m_{\text{k}}$. The latter quantity is a measure for the amount of neutral material in the photoevaporation flow and the tail region. The bottom row shows the kinetic and internal energy of the neutral knot material as a function of time. We used a threshold of 10% of the initial knot density to determine whether or not a cell contains knot material. All times mentioned below refer to the results obtained with the C^2-Ray method.

As can be seen from these plots, the two methods show the same overall evolutionary behaviour, although the Hybrid Characteristics method clearly overestimates the amount of photo-heating. This results in a faster evolution of the knot, and a higher internal and kinetic energy. We note that, besides the differences in ray tracing, the two methods also employ different solvers for the hydrodynamics (see Section 6.3 for details), which may influence the outcome of the simulations as well. These results show that quantitative results (e.g. the evaporation time) obtained from calculations that do not conserve photons should be interpreted with care.

During the compression phase, which lasts until $t \sim 700$ yr, there is a steady decrease in m_{k}, while $m_{\text{tot}} - m_{\text{k}} \simeq 0.03 m_{\text{k}}(t = 0)$ stays approximately constant. This implies that the decrease in neutral knot material is mainly due to the photoevaporation process, and that the increase of neutral material through accretion in the tail region is negligible. The kinetic and internal energy of the neutral clump material both show a steady increase during the compression phase.

The transition from the compression to the expansion phase at $t \sim 700$ yr is characterized by a slight increase in the mass loss rate for m_{k}, and thus a steady increase of $m_{\text{tot}} - m_{\text{k}}$. This is caused by the expansion wave that runs from the back of the knot towards the source once the shock has left the knot. This transition also forms the onset of the 'rocket effect' where the knot starts to accelerate in its entirety. This shows up in a steeper increase of the kinetic energy, and a slight decrease of the internal energy.

When the expansion wave arrives at the head of the knot at $t \sim 1200$ yr, the lowering of the density at the ionization front produces another shock (see Section 6.5), which accelerates the knot a second time. This is reflected in a further steepening of the slope of the kinetic energy plot. By this time, the knot has developed a central region of decreased density.

At $t \sim 2200$ yr, only $\sim 20\%$ of the original neutral material is left, and the kinetic and internal energy of the neutral knot material start to decrease as the knot gets completely evaporated.

6.7   Two knots: special cases

Before we present photoevaporation simulations of an ensemble of multiple knots, we first concentrate on some specific interactions that may occur when only two knots are present. Details of the setup of these models can be found in Table 6.2. The results of these calculations will help us in identifying similar interactions in simulations where a whole ensemble of knots is being photoevaporated simultaneously.

We start with a calculation where the photoevaporation flows of two knots that lie

Table 6.2: Summary of initial knot positions (in units of 10^{15} cm) used for the different simulations involving two knots. The size of the computational domain is $(64, 4, 4) \times 10^{16}$ cm. All knots are located at the centre of the domain in the z-direction. Parameters for density and temperature are the same as the ones used for model 1K.

pos	2KIF	2KIF'	2KIK	2KSK
(x_1, y_1)	(50.0,1.25)	(50.0,1.25)	(50.0,1.75)	(50.0,1.75)
(x_2, y_2)	(50.0,2.75)	(50.5,2.75)	(51.0,2.25)	(56.0,2.25)

at a similar distance from the source interact. We then present a simulation where one knot partially obscures another that lies further away from the source, where the distance between the knots is chosen such that the two will interact hydrodynamically during the expansion phase. We finish with a similar calculation but now the knot that is partially shadowed lies at a greater distance from the other knot, to ensure that no direct hydrodynamical interaction takes place among the two knots. Apart from the latter setup, similar calculations have been done before by Lim & Mellema (2003). Like in the case for a single knot, we discuss the results by identifying signatures in the global properties of the flow.

6.7.1 2KIF: interacting flows

Here, we study the interaction of photoevaporation flows originating from two knots located at a similar distance from the source. The distance between the knots is chosen such that their photoevaporation flows will interact, leading to a region of enhanced density between the two knots where these flows collide. When the knots are well aligned, this interaction zone can become dense enough to recombine and, eventually, may even cast a shadow. However, for the physical conditions in the Helix, the recombination time is $\lesssim 2500$ yr, which is too long to make this a dynamically important effect.

In Figure 6.7 and Figure 6.8 we show the evolution of this interaction for two cases: for the first model (2KIF), the knots are at an equal distance from the source, and for the second model (2KIF'), one knot is located slightly further away in order to break the symmetry of the problem. From these images one sees that, apart from the formation of an interaction region between the knots, the evolution proceeds roughly the same as in baseline model 1K. This is also reflected in the almost identical curves for the global properties of the flow as shown in Figure 6.9.

6.7.2 2KIK: interacting knots

This section describes the interaction of two photoevaporating knots (model 2KIK). For this setup, the distance between the knots was chosen such that, once the knot that lies closest to the source starts to expand and accelerate, it will interact with the second knot.

The results of this interaction are shown in Figure 6.10. During the compression phase, the knots are still fairly separated, and the knot that is partially shadowed is driven into the shadow region of the knot that lies closer to the source. When the compression
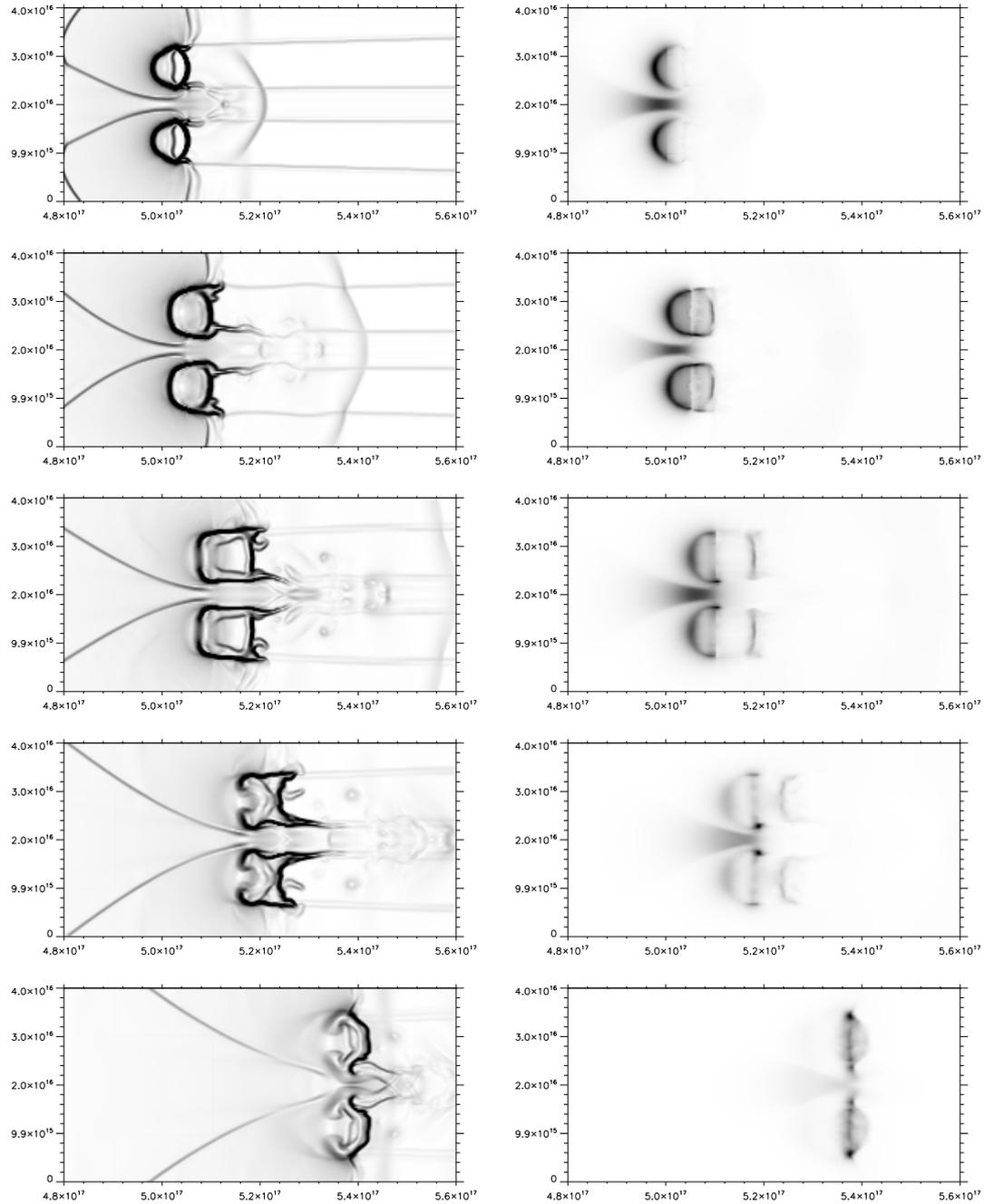
Figure 6.7: Model 2KIF: 3D calculations of the evolution of two knots that lie at the same distance from the source. Shown are xy-cuts through the centre of the computational domain of Schlieren plots of the $\log_{10}$ of the density (left) and synthesized Hα images (right). Time increases in equal intervals from $t = 316$ yr (top) to $t = 1580$ yr (bottom).
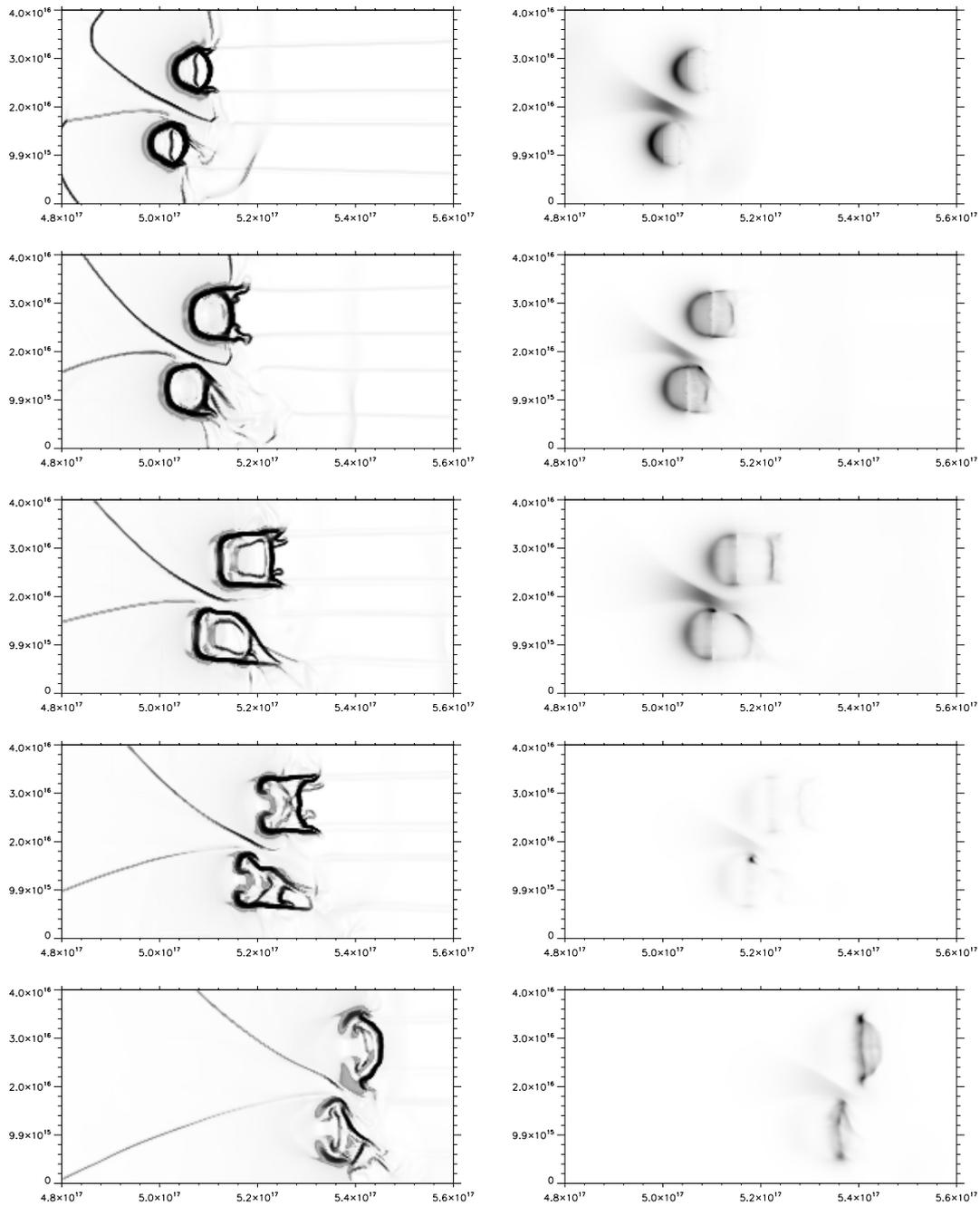
Figure 6.8: Model 2KIF': 3D calculations of the evolution of two knots that lie at approximately the same distance from the source. Shown are xy-cuts through the centre of the computational domain of Schlieren plots of the $\log_{10}$ of the density (left) and synthesized Hα images (right). Time increases in equal intervals from $t = 316$ yr (top) to $t = 1580$ yr (bottom).
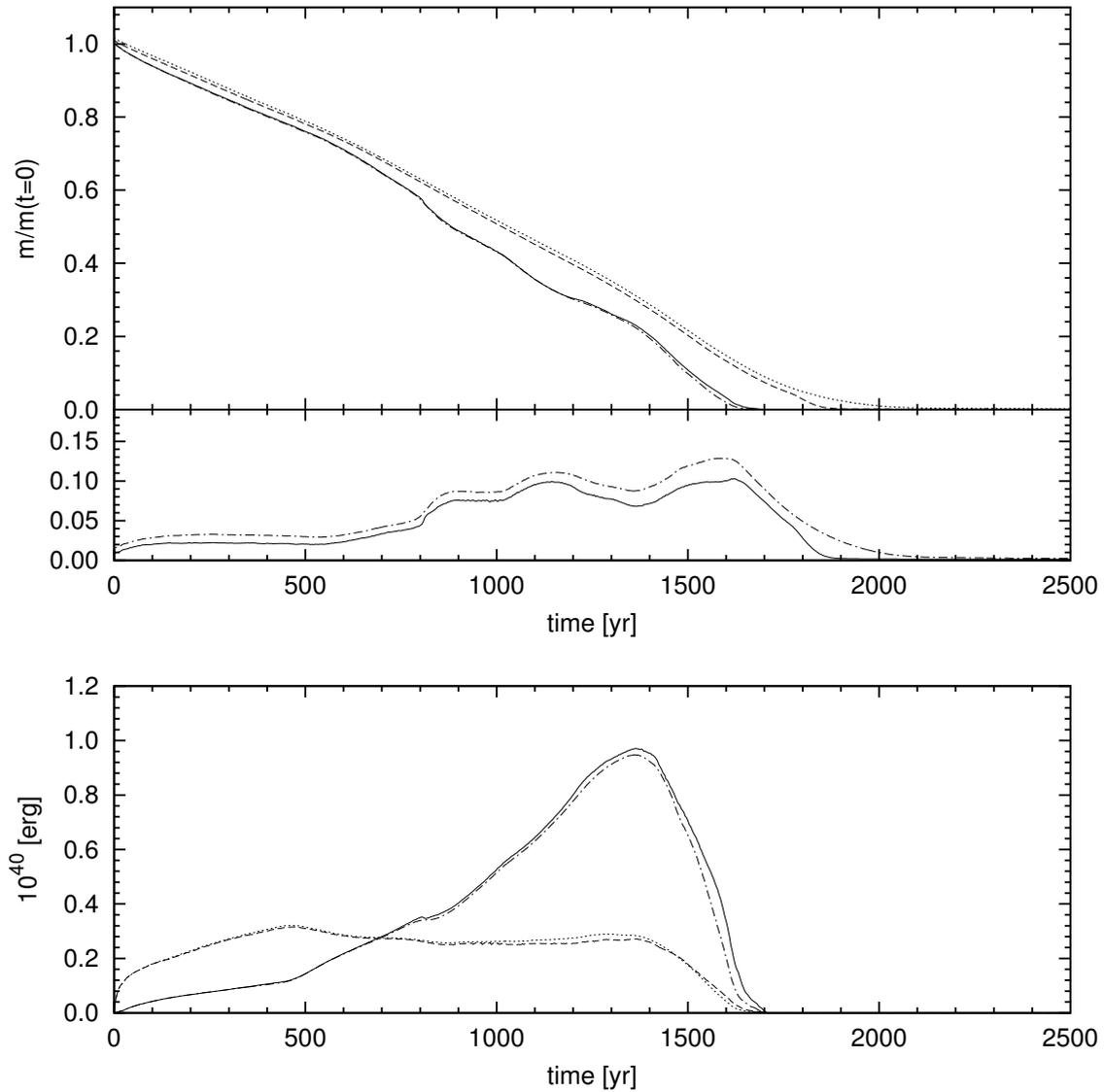
Figure 6.9: Global flow properties as a function of time comparing the 1K (solid and dashed lines) and 2KIF' (dashed-dotted and dotted lines) models. The top panel shows the neutral knot mass (solid and dashed-dotted lines), and the total neutral mass (dashed and dotted lines), with the difference between these two quantities plotted beneath. The bottom panel shows the kinetic (solid and dashed-dotted lines) and internal (dashed and dotted lines) energy of the knots, where for model 2KIF' values have been divided by two.

phase has finished, the knots merge and are being accelerated as a single entity. As can be seen from the evolution of the global properties of the flow shown in Figure 6.11, this results in a slightly longer evaporation time and a lower average kinetic energy for the knots. The interaction also leads to a slightly larger amount of neutral material in the tail region, as is evident from the top panel of Figure 6.11.

6.7.3   2KSK: shadowing knots

The evolution of a knot that is far enough away from any other knot to not interact hydrodynamically during its lifetime may still be influenced by the shadowing effect of knots that lie closer to the source. This effect is investigated in model 2KSK, for which two knots are positioned such that one partially obscures the other from the direct stellar photons, while the distance between the two is chosen large enough to prevent any direct hydrodynamical interaction among them. The results of this simulation are shown in Figure 6.12 and 6.13.

This sequence of images shows that only the side of the knot that is partially illuminated by the source starts to photoevaporate. The sideways outflow from the knot makes that it is driven further into the shadow region, and the knot material acquires a transverse velocity. After the shock has passed all the way through the second knot the increase in pressure makes that the material expands. Since, in the tail region behind the first knot, there is no ionizing radiation that can induce pressure to counter this expansion, this material spreads throughout a large part of this region. Any material that flows out of the shadow region gets photoevaporated immediately, and the back reaction keeps most of the material confined. After the first knot is fully evaporated, the material from the second knot is exposed to the direct radiation of the source, and starts to photoevaporate as well.

The mechanism just described provides an effective means for dispersing knot material throughout a large portion of space. Since in this model the knots are chosen to be so far apart that no hydrodynamical interaction takes place among them, the material of the second knot that is driven into the shadow region is not being swept up by the remains of the first knot as was the case for model 2KIK. To the contrary: the material of the second knot spreads extensively throughout the shadow region.

This is also reflected in the evolution of global flow properties shown in Figure 6.14. This figure shows that the amount of neutral material that gets confined in the shadow region increases drastically, and reaches a maximum value of $\sim 30\%$ of the initial mass of the knot. The subsequent photoevaporation of this material during the later phases of the evolution shows up as small peaks in the kinetic and internal energy at $t \simeq 1700$ yr.

6.8   Multiple knots

In this section we present a calculation involving the simultaneous photoevaporation of an ensemble of 25 knots (model 25K). These knots are distributed randomly throughout the region $([50, 56], [1, 7], [1, 7]) \times 10^{16}$ cm of a computational domain of size $(64, 8, 8) \times 10^{16}$ cm. This domain has an effective resolution of $(2048,256,256)$ cells. The density
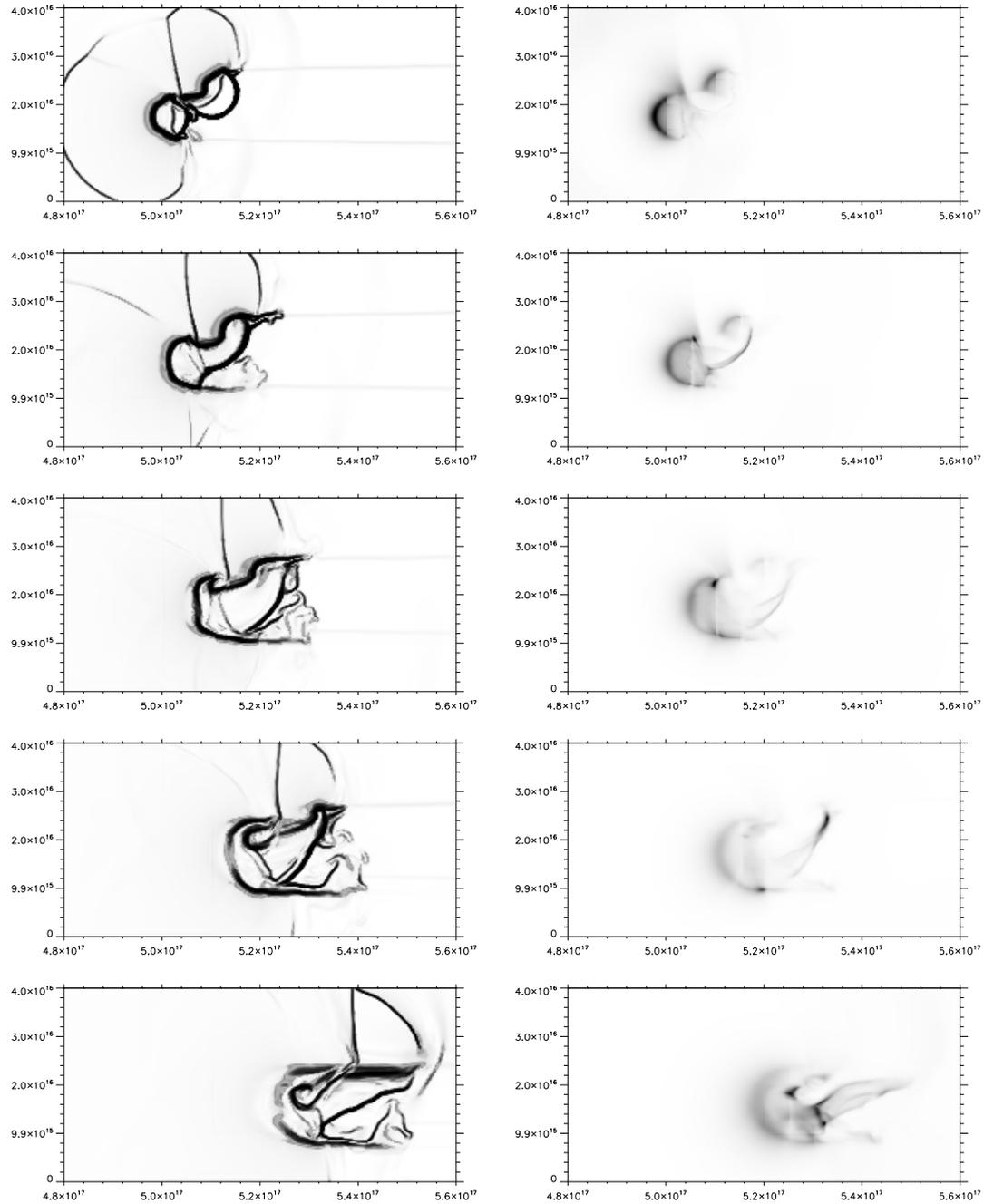
Figure 6.10: Model 2KIK: 3D calculation of the evolution of two interacting knots. Shown are xy-cuts through the centre of the computational domain of Schlieren plots of the $\log_{10}$ of the density (left) and synthesized Hα images (right). Time increases in equal intervals from $t = 316$ yr (top) to $t = 1580$ yr (bottom).
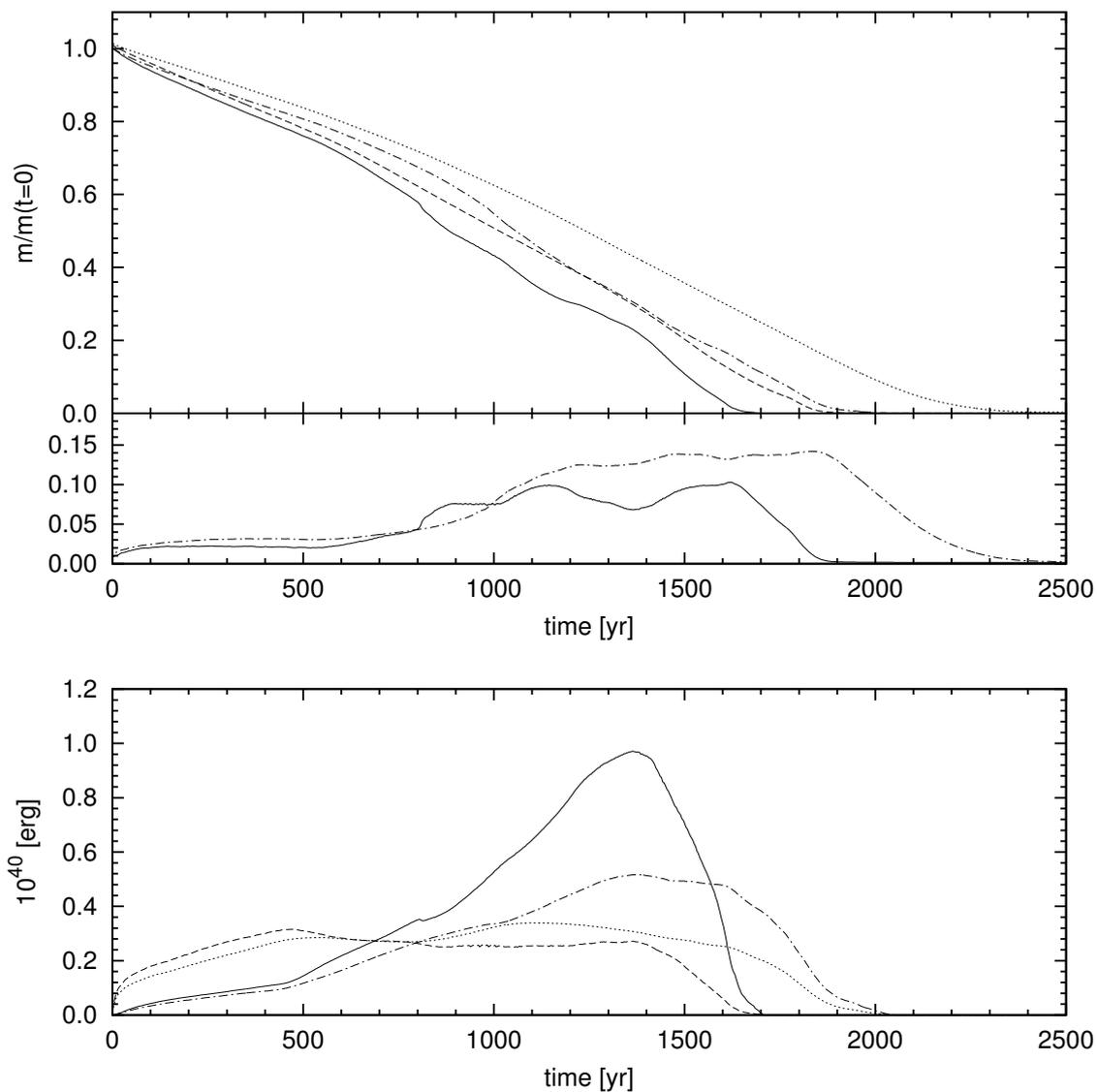
Figure 6.11: Global flow properties as a function of time comparing the 1K (solid and dashed lines) and 2KIK (dashed-dotted and dotted lines) models. The top panel shows the neutral knot mass (solid and dashed-dotted lines), and the total neutral mass (dashed and dotted lines), with the difference between these two quantities plotted beneath. The bottom panel shows the kinetic (solid and dashed-dotted lines) and internal (dashed and dotted lines) energy of the knots, where for model 2KIK values have been divided by two.
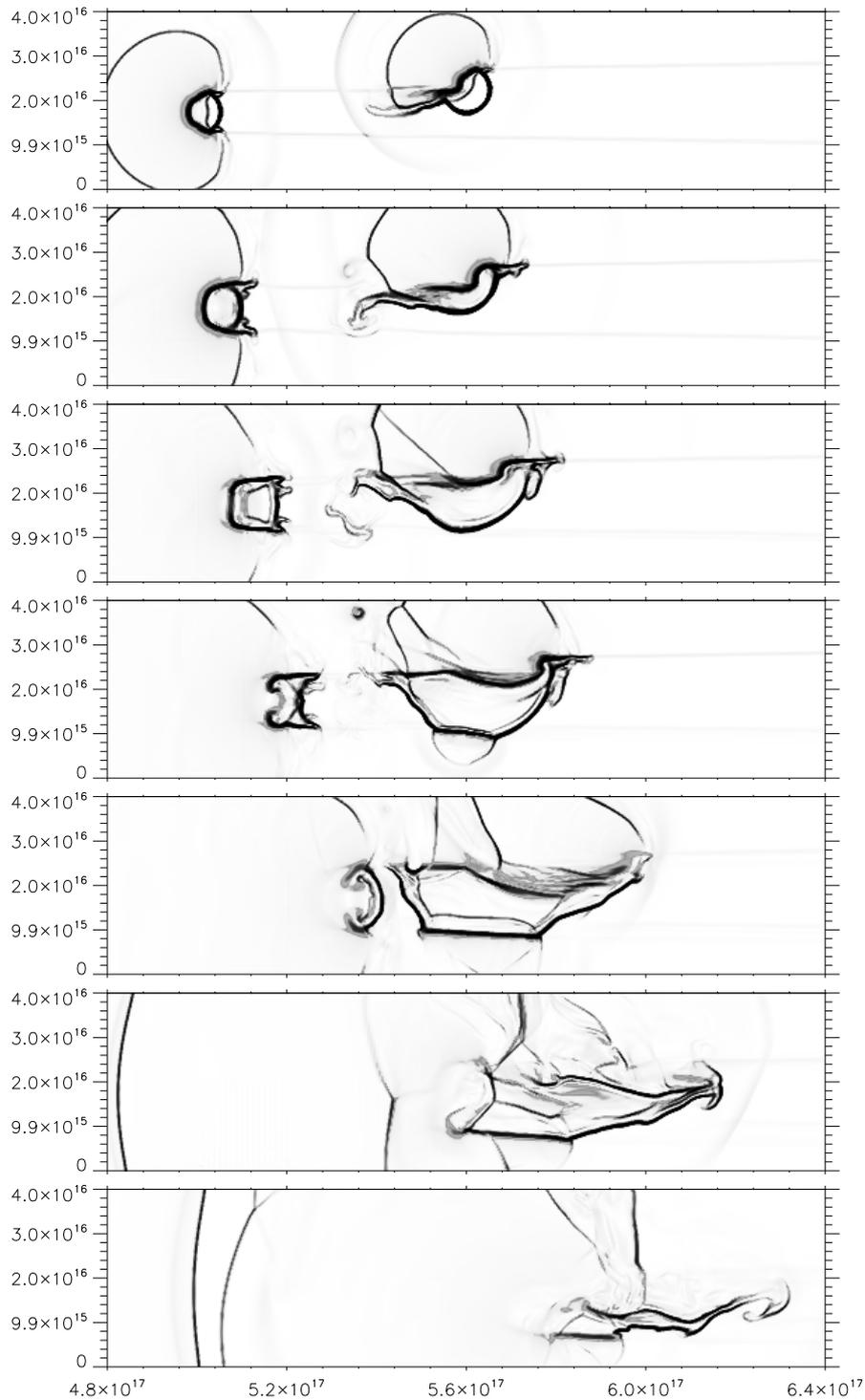
Figure 6.12: Model 2KSK: 3D calculation of the evolution of two knots, where one is partially obscured from direct stellar radiation by another. Shown are xy-cuts through the centre of the computational domain of Schlieren plots of the $\log_{10}$ of the density. Time increases in equal intervals from $t = 316$ yr (top) to $t = 2220$ yr (bottom).
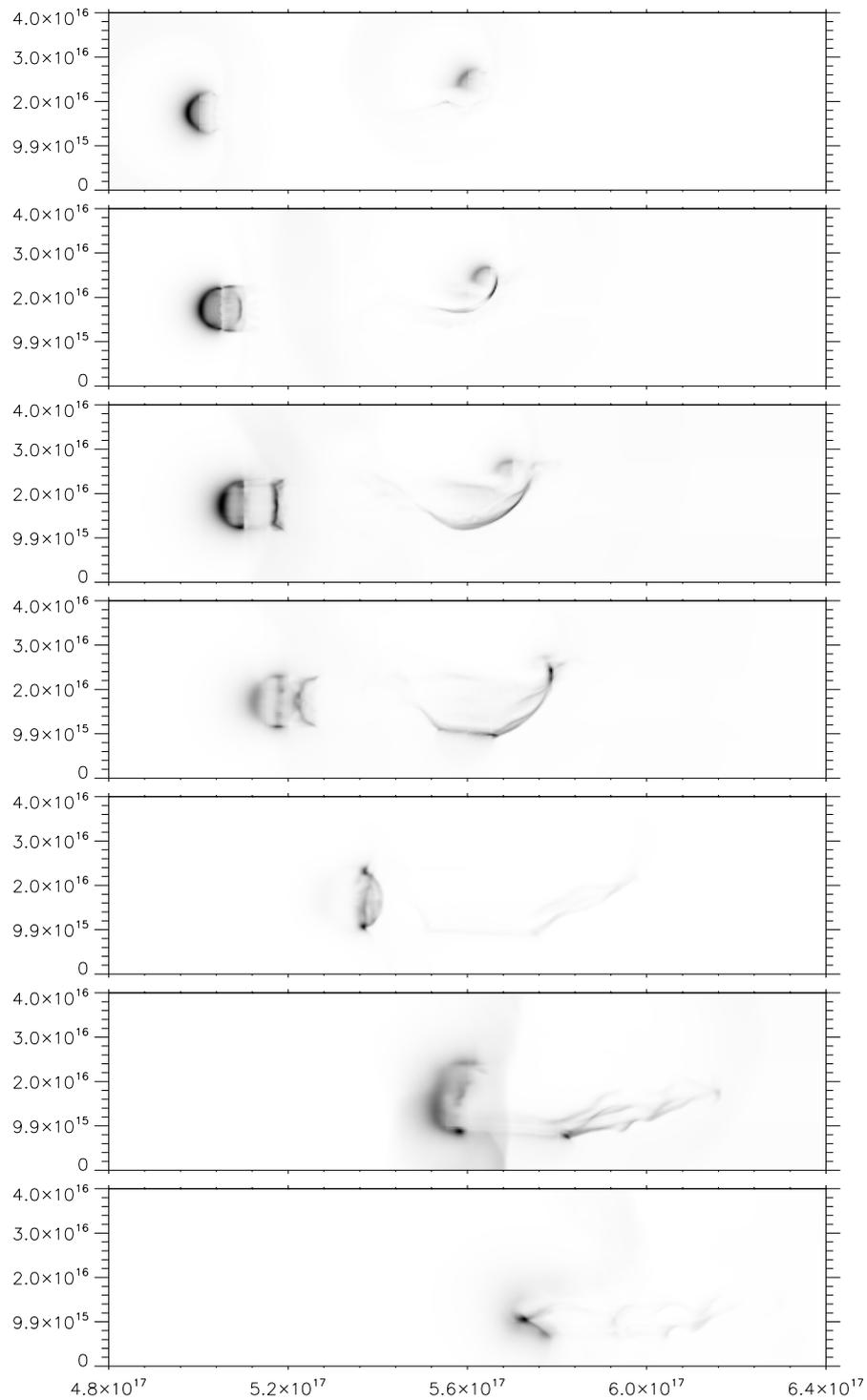
Figure 6.13: Model 2KSK: 3D calculation of the evolution of two knots, where one is partially obscured from direct stellar radiation by another. Shown are synthesized Hα images. Time increases in equal intervals from $t = 316$ yr (top) to $t = 2220$ yr (bottom).
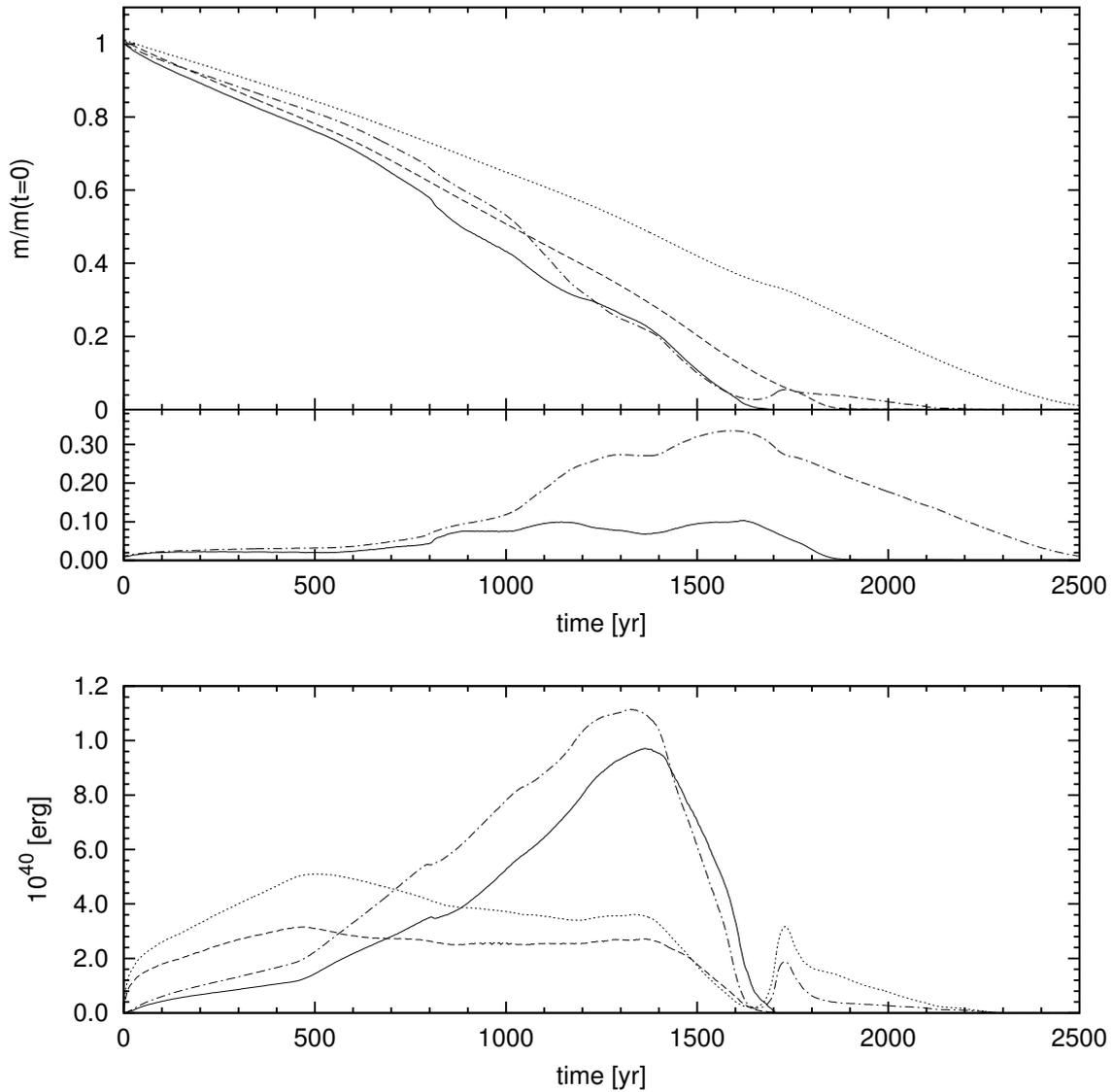
Figure 6.14: Global flow properties as a function of time comparing the 1K (solid and dashed lines) and 2KSK (dashed-dotted and dotted lines) models. The top panel shows the neutral knot mass (solid and dashed-dotted lines), and the total neutral mass (dashed and dotted lines), with the difference between these two quantities plotted beneath. The bottom panel shows the kinetic (solid and dashed-dotted lines) and internal (dashed and dotted lines) energy of the knots, where for model 2KSK values have been divided by two.

of the knots is selected randomly from the range $[2.5, 7.5] \times 10^4$ cm^{-3}, and the radius is selected from the range $[3.75, 6.25] \times 10^{15}$ cm. This gives a space filling factor of approximately $25(4\pi/3)(5/60)^3 \simeq 0.06$.

The results of this simulation are shown in Figure 6.15 and 6.16. For this particular setup, all three interactions that may occur between two knots as described in Section 6.7 actually do take place. Wisps of emission seen in between knots resulting from the collision of their photoevaporation flows (as in model 2KIF') are clearly seen in the synthesized Hα images, whereas the Schlieren plots show the effects of obscuration from direct stellar photons leading to knot material expanding into the tail regions (as in model 2KSK), and direct hydrodynamical interaction among the knots (as in model 2KIK).

The influence of all these effects together on the global properties of the flow are shown in Figure 6.17. The average photoevaporation time is increased by a factor ~ 1.5 as compared to model 1K, and the maximum amount of neutral material in the tail regions is approximately $\sim 20\%$ of the initial mass of the knots. Due to merging of knots and the expansion of knot material into the tail regions, the average total kinetic energy of the neutral knot material is a factor ~ 0.4 lower than in the case of model 1K. The average internal energy of the neutral knot material on the other hand is slightly higher than in the case of model 1K, and is comparable to the average total kinetic energy.

6.9 Discussion and comparison to observations

In this section, we compare the results of the 3D simulations to observations of knots in the Helix nebula. From the Schlieren plots, shown in for example Figure 6.5 of model 1K, one can see that, as soon as the expansion phase has started, the internal structure of the knot starts to deviate severely from the initial homogeneous state. However, the appearance of the knot in Hα still shows a limb-brightened cusp like shape (see for example Figure 6.7). Note that the structure in the Hα emission in the synthesized images as seen in the core region of the knot will in reality be invisible due to dust extinction.

The difference in appearance between internal structure and observed shape can be explained by the thickness of the layer of photoevaporating material streaming of the knot, which keeps the ionization front away from the knot's distorted shape. This means that the knots in the Helix may be in an evolved state, close to, or even beyond, the transition from the compression to the expansion phase. For example, a cut through the [OIII] image of knot 378-801 (see O'Dell et al. 2005, their Figure 11) shows a gradual decrease in extinction when moving from the core of the knot into the tail region, which could be interpreted as evidence for expansion of the knot.

Such an evolved state for the knots may also be required by the kinematic age of the nebula. Whether the knots are in an evolved stage also depends on which mechanism responsible for the origin of the knots one adheres to. However, the observed high density in the core region of the knots suggest they can not have expanded too much yet, unless they were of even higher density in the past. These considerations, together with the evaporation times obtained from our numerical simulations, suggest an age of at most ~ 1000 yr for the knots.

The 'wispy' emission observed in between many of the knots in the Helix can be
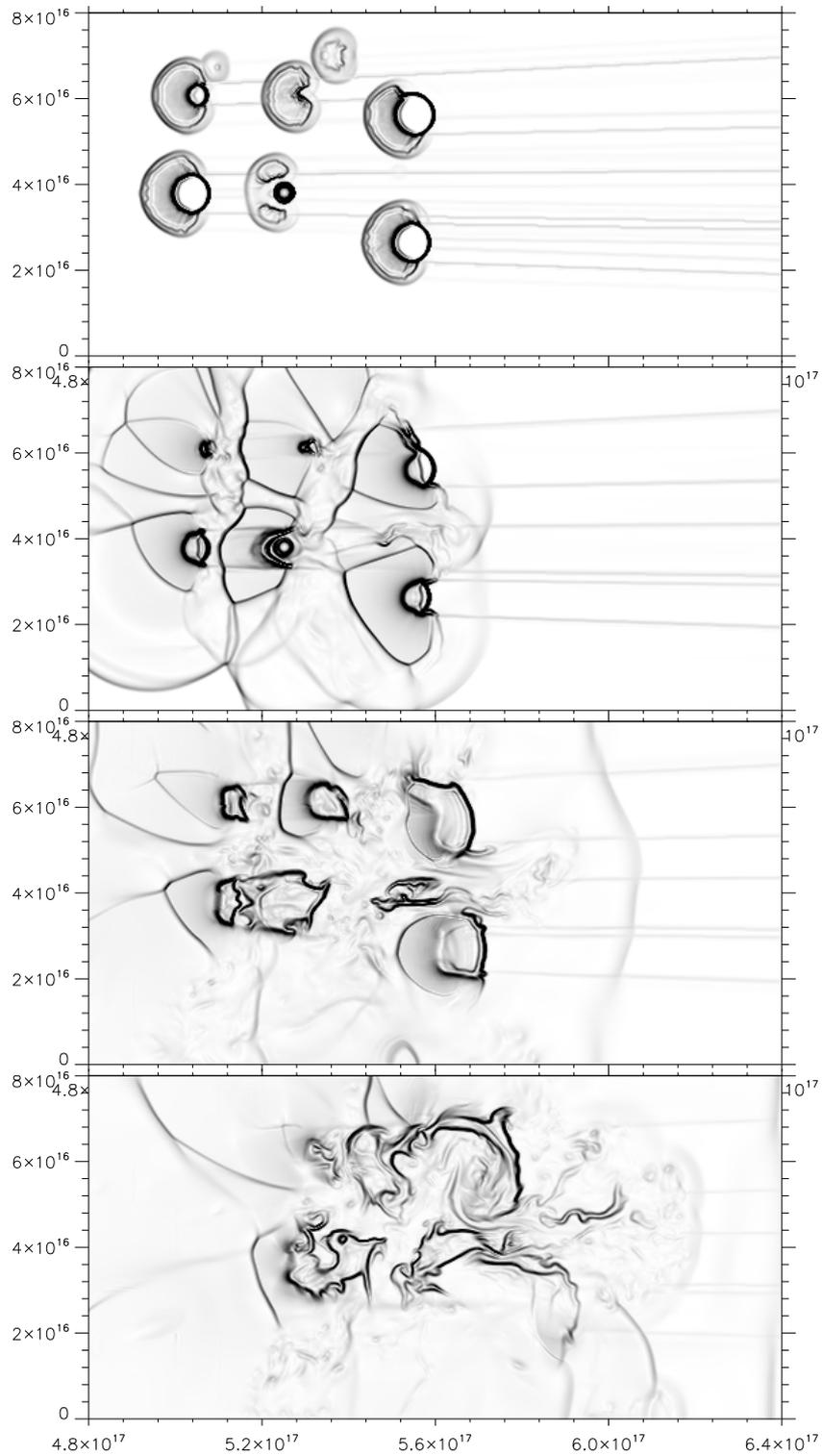
Figure 6.15: Model 25K: 3D calculation of the evolution of 25 photoevaporating knots. Shown are xy-cuts through the centre of the computational domain of Schlieren plots of the $\log_{10}$ of the density for times $t = 64$, $t = 316$, $t = 950$, and $t = 1580$ yr (top to bottom).
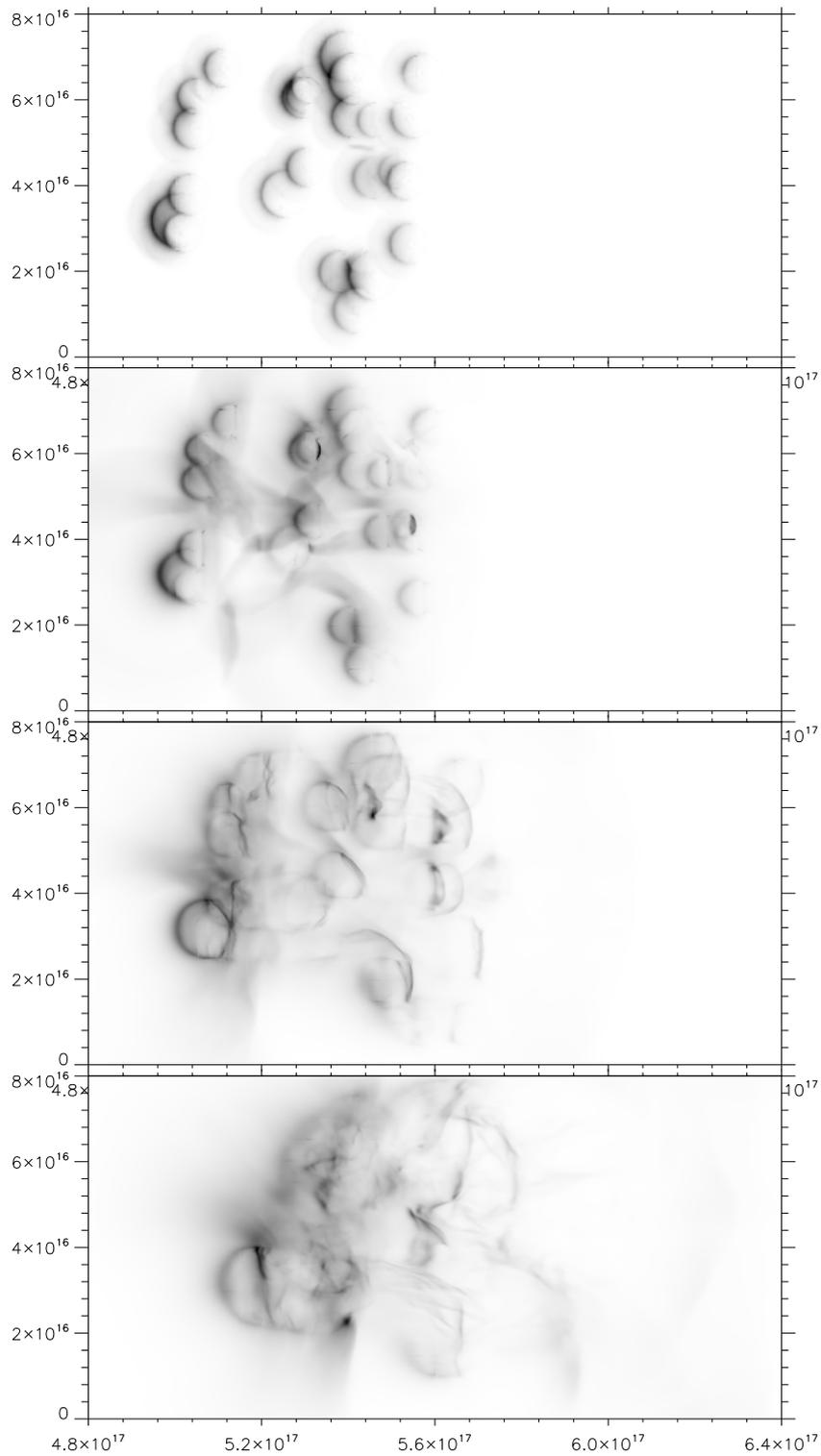
Figure 6.16: Model 25K: 3D calculation of the evolution of 25 photoevaporating knots. Shown are synthesized Hα images for times $t = 64$, $t = 316$, $t = 950$, and $t = 1580$ yr (top to bottom).
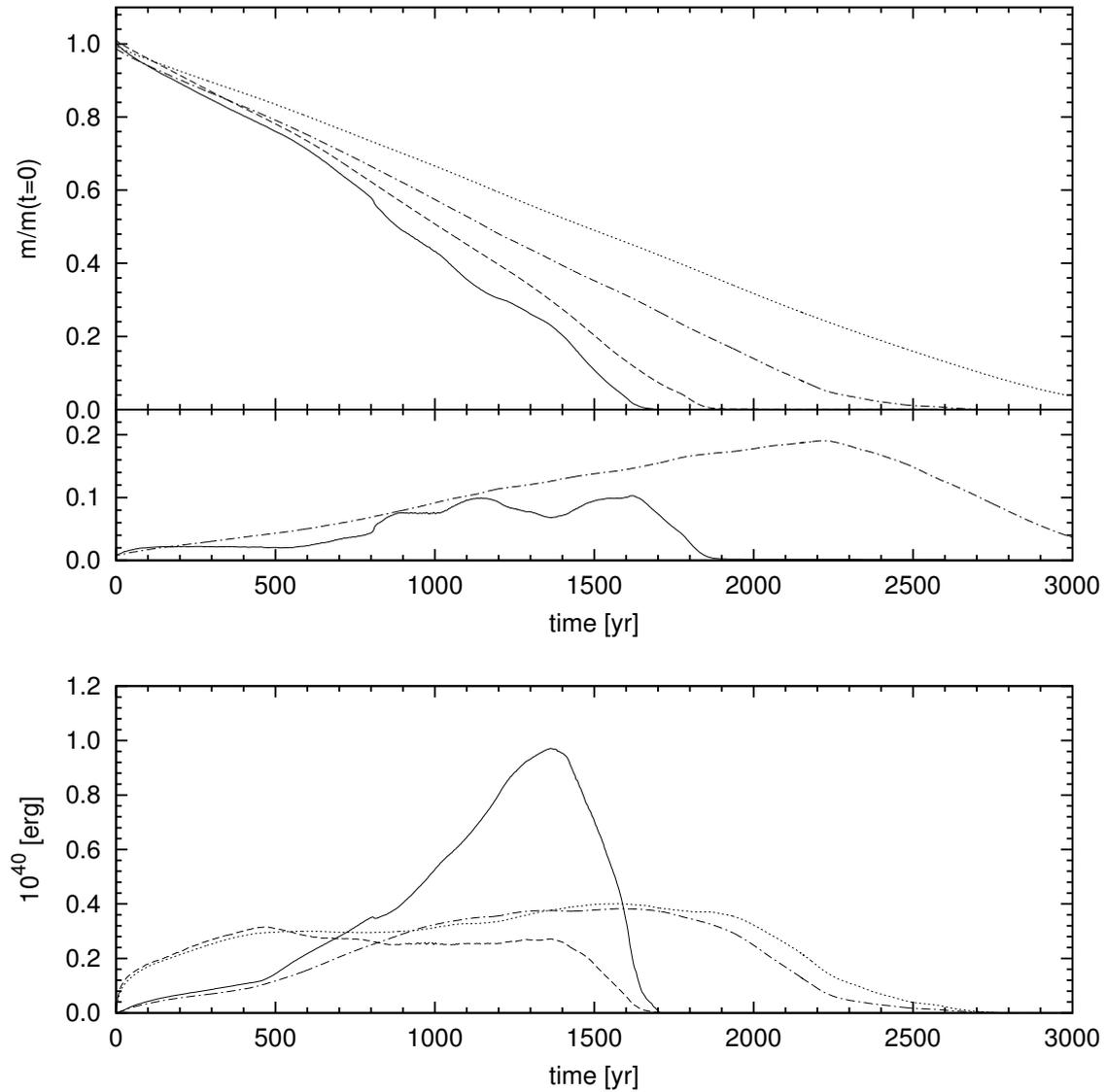
Figure 6.17: Global flow properties as a function of time comparing the 1K (solid and dashed lines) and 25K (dashed-dotted and dotted lines) models. The top panel shows the neutral knot mass (solid and dashed-dotted lines), and the total neutral mass (dashed and dotted lines), with the difference between these two quantities plotted beneath. The bottom panel shows the kinetic (solid and dashed-dotted lines) and internal (dashed and dotted lines) energy of the knots, where for model 25K values have been divided by a factor of 25.

explained by Model 2KIF('). This emission is then interpreted as resulting from the colliding photoevaporation flows. As mentioned by Lim & Mellema (e.g. 2003), this same mechanism can explain the interaction zone observed in between two proplyds (e.g. Graham et al. 2002; Brandner et al. 2000). For such objects, this effect is expected to be even more prominent because of the recombination time scale, which is short compared to the evolutionary time scale of proplyds.

The results of model 2KSK supports the assertion made by O'Dell et al. (2005) that the peaks observed at the edge of the tail region of knot 378-801 are the result of (a) partially shadowed knot(s), and can also explain the extended dust (Meaburn et al. 1992) and molecular emission (Huggins et al. 2002) observed in the tail region of this knot. The appearance of long tails observed behind proplyds in for example the Carina Nebula (Smith et al. 2003), may be the result of this mechanism as well.

Model 25K of multiple photoevaporating knots shows that the mechanism of model 2KIK can significantly prolong the survival of the knots. Also, the mechanism of model 2KSK drastically increases the amount of neutral material in the tail regions associated with the knots. Although the most prominent knots in the Helix are single, the bulk of the knots occur in close groups (see for example the Hα images in O'Dell et al. 2004, 2005; Meixner et al. 2005), so these two mechanisms will be of importance for the amount and density of the neutral material that eventually survives the PN stage and gets fed back into the interstellar medium.

6.10   Conclusions

We have shown that the physical conditions in the Helix knots are such that the ionization front, induced by the ionizing radiation from the central star, has a lower velocity than the shock that precedes it. This leads to a thick layer of shocked material between these two fronts, which, once the shock has left the back of the knot, starts to expand. We analytically derived a condition for which such an expansion phase occurs, and showed that this is the *same* condition introduced by Bertoldi (1989) to discern between two regimes for photoevaporating clouds. This implies that this expansion phase is in fact a common and therefore important phase in the evolution of photoevaporating clouds.

The 3D radiation hydrodynamical simulations presented in this chapter involving multiple knots illustrate the different interactions that take place among this type of objects. The interaction of the evaporation flows leads to enhanced emission in Hα, the collision of one knot with another, partially shadowed, knot, results in a longer evaporation time and lower velocity of the knots, and the partially shadowing of a knot that lies far enough away from any other knot to prevent hydrodynamical interaction leads to the dispersion of a large amount of the knot's material into the shadow region. These simulations thus show that calculations involving just one knot are insufficient in describing the evolution of most of the knots present in the Helix nebula.

Acknowledgements

Bibliography

Abel, T., Norman, M. L., & Madau, P. 1999, ApJ, 523, 66

Abel, T. & Wandelt, B. D. 2002, MNRAS, 330, L53

Amanatides, J. & Woo, A. 1987, in Proc. Eurographics, 1

Armitage, P. J. & Pringle, J. E. 1997, ApJ, 488, L47

Balick, B. 1987, AJ, 94, 671

Balick, B., Alexander, J., Hajian, A. R., et al. 1998, AJ, 116, 360

Balick, B. & Frank, A. 2002, ARA&A, 40, 439

Bardeen, J. M. & Petterson, J. A. 1975, ApJ, 195, L65

Bell, J., Berger, M., Saltzman, J., & Welcome, M. 1994, SIAM J. Sci. Comp., 15, 127

Berger, M. 1982, Ph.D. Thesis

Berger, M. & LeVeque, R. J. 1998, SIAM J. Numer. Anal., 35, 2298

Berger, M. & Oliger, J. 1984, J. Comp. Phys., 53, 484

Berger, M. & Rigoutsos, I. 1992, IEEE Trans. on Systems Man and Cybernetics, 21, 1278

Berger, M. J. & Colella, P. 1989, J. Comp. Phys., 82, 64

Bertoldi, F. 1989, ApJ, 346, 735

Bertoldi, F. & McKee, C. F. 1990, ApJ, 354, 529

Blöcker, T. 1995, A&A, 299, 755

Bohlin, R. C., Harrington, J. P., & Stecher, T. P. 1982, ApJ, 252, 635

Brandner, W., Grebel, E. K., Chu, Y., et al. 2000, AJ, 119, 292

Breitschwerdt, D. & Kahn, F. D. 1990, MNRAS, 244, 521

Bujarrabal, V., Neri, R., Alcolea, J., & Kahane, C. 2003, A&A, 409, 573

Canto, J., Raga, A., Steffen, W., & Shapiro, P. 1998, ApJ, 502, 695

Capriotti, E. R. 1973, ApJ, 179, 495

Cen, R. 2002, ApJS, 141, 211

Cerruti-Sola, M. & Perinotto, M. 1985, ApJ, 291, 237

Cliffe, J. A., Frank, A., Livio, M., & Jones, T. W. 1995, ApJ, 447, L49

Colella, P. & Glaz, H. M. 1985, J. Comp. Phys., 59, 264

Colella, P. & Woodward, P. R. 1984, J. Comp. Phys., 54, 174

Cox, D. P. 1970, Ph.D. Thesis

Cuesta, L., Phillips, J. P., & Mampaso, A. 1996, A&A, 313, 243

Dalgarno, A. & McCray, R. A. 1972, ARA&A, 10, 375

de Marco, O. & Crowther, P. A. 1999, MNRAS, 306, 931

Deiterding, R. 2003, Ph.D. Thesis

Dursi, L. J. & Zingale, M. 2004, in Adaptive Mesh Refinement - Theory and Applications, Eds. T. Plewa, T. Linde & V. G. Weirs

Dyson, J. E., Hartquist, T. W., Pettini, M., & Smith, L. J. 1989, MNRAS, 241, 625

Flash User's Guide. V2.5

Eulderink, F. & Mellema, G. 1995, A&AS, 110, 587

Falle, S. A. E. G. & Raga, A. C. 1993, MNRAS, 261, 573

Frank, A. 2000, in ASP Conf. Ser. 199: Asymmetrical Planetary Nebulae II: From Origins to Microstructures, 225

Frank, A. & Mellema, G. 1994a, A&A, 289, 937

—. 1994b, ApJ, 430, 800

Fryxell, B., Olson, K., Ricker, P., et al. 2000, ApJS, 131, 273

García-Arredondo, F., Henney, W. J., & Arthur, S. J. 2001, in Revista Mexicana de Astronomia y Astrofisica Conference Series, 65–68

García-Segura, G. 1997, ApJ, 489, L189

García-Segura, G. & López, J. A. 2000, ApJ, 544, 336

García-Segura, G., Mac Low, M.-M., & Langer, N. 1996, A&A, 305, 229

Gnedin, N. Y. 1995, ApJS, 97, 231

Godunov, S. K. 1959, Mat.Sb., 47, 271

Gorny, S. K., Stasinska, G., & Tylenda, R. 1997, A&A, 318, 256

Graham, M. F., Meaburn, J., Garrington, S. T., et al. 2002, ApJ, 570, 222

Guerrero, M. A., Vázquez, R., & López, J. A. 1999, AJ, 117, 967

Habing, H. J. 1996, A&A Rev., 7, 97

Harman, D. J., Bryce, M., López, J. A., Meaburn, J., & Holloway, A. J. 2004, MNRAS, 348, 1047

Harris, H. C., Dahn, C. C., Monet, D. G., & Pier, J. R. 1997, in IAU Symp. 180: Planetary Nebulae, 40

Hayes, J. C. & Norman, M. L. 2003, ApJS, 147, 197

Heinemann, T., Dobler, W., Nordlund, A., & Brandenburg, A. 2005, astro-ph/0503510

Henney, W. J. & Arthur, S. J. 1998, AJ, 116, 322

Henry, R. B. C., Kwitter, K. B., & Dufour, R. J. 1999, ApJ, 517, 782

Huggins, P. J., Forveille, T., Bachiller, R., et al. 2002, ApJ, 573, L55

Iben, I. & Renzini, A. 1983, ARA&A, 21, 271

Icke, V. 1988, A&A, 202, 177

—. 1991, A&A, 251, 369

—. 2003, A&A, 405, L11

Icke, V., Mellema, G., Balick, B., Eulderink, F., & Frank, A. 1992, Nature, 355, 524

Icke, V., Preston, H. L., & Balick, B. 1989, AJ, 97, 462

Iping, R. C. & Petterson, J. A. 1990, A&A, 239, 221

Janka, H.-T. & Mueller, E. 1996, A&A, 306, 167

Jessee, J. P., Fiveland, W. A., Howell, L. H., Colella, P., & Pember, R. B. 1998, J. Comp. Phys., 139, 380

Jura, M. & Kahane, C. 1999, ApJ, 521, 302

Juvela, M. & Padoan, P. 2005, ApJ, 618, 744

Kahn, F. D. 1954, Bull. Astron. Inst. Netherlands, 12, 187

—. 1976, A&A, 50, 145

Kahn, F. D. & Breitschwerdt, D. 1990, MNRAS, 242, 505

Kahn, F. D. & West, K. A. 1985, MNRAS, 212, 837

Keppens, R., Nool, M., Tóth, G., & Goedbloed, J. P. 2002, Comp. Phys. Comm., 153, 317

Koo, B. & McKee, C. F. 1992, ApJ, 388, 103

Kreylos, O., Weber, G. H., Bethel, W., et al. 2002, Remote Interactive Direct Volume Rendering of AMR Data, Tech. Rep. LBNL 49954, Lawrence Berkeley National Laboratory

Kwok, S., Hrivnak, B. J., & Su, K. Y. L. 2000, ApJ, 544, L149

Kwok, S., Purton, C. R., & Fitzgerald, P. M. 1978, ApJ, 219, L125

López-Martín, L., Raga, A. C., Mellema, G., Henney, W. J., & Cantó, J. 2001, ApJ, 548, 288

Lai, D. 1999, ApJ, 524, 1030

—. 2003, ApJ, 591, L119

Lamers, H. J. G. L. M. & Cassinelli, J. P. 1999, Introduction to stellar winds (Cambridge University Press)

Laney, C. B. 1998, Computational gasdynamics (Cambridge University Press)

Lefloch, B. & Lazareff, B. 1994, A&A, 289, 559

Leveque, R. J. 2002, Finite Volume Methods for Hyberbolic Problems (Cambridge University Press)

Liebendörfer, M., Rampp, M., Janka, H.-T., & Mezzacappa, A. 2005, ApJ, 620, 840

Lim, A. J. & Mellema, G. 2003, A&A, 405, 189

Livio, M. & Pringle, J. E. 1996, ApJ, 465, L55

—. 1997, ApJ, 486, 835

Löhner, R. 1987, Comp. Meth. App. Mech. Eng., 61, 323

Luo, H., Baum, J., & Löhner, R. 2000, Computers & Fluids, 30, 137

Maas, T., Van Winckel, H., Lloyd Evans, T., et al. 2003, A&A, 405, 271

MacNeice, P., Olson, K. M., Mobarry, C., deFainchtein, R., & Packer, C. 2000, Comp. Phys. Comm., 126, 330

Maloney, P. R. & Begelman, M. C. 1997, ApJ, 491, L43

Maloney, P. R., Begelman, M. C., & Nowak, M. A. 1998, ApJ, 504, 77

Maloney, P. R., Begelman, M. C., & Pringle, J. E. 1996, ApJ, 472, 582

Manchado, A., Stanghellini, L., & Guerrero, M. A. 1996, ApJ, 466, L95

Maselli, A., Ferrara, A., & Ciardi, B. 2003, MNRAS, 345, 379

Meaburn, J., Clayton, C. A., Bryce, M., et al. 1998, MNRAS, 294, 201

Meaburn, J., Walsh, J. R., Clegg, R. E. S., et al. 1992, MNRAS, 255, 177

Meixner, M., McCullough, P., Hartman, J., Son, M., & Speck, A. 2005, AJ, 130, 1784

Mellema, G. 1993, Ph.D. Thesis

—. 1995, MNRAS, 277, 173

—. 1997, A&A, 321, L29

Mellema, G., Eulderink, F., & Icke, V. 1991, A&A, 252, 718

Mellema, G. & Frank, A. 1997, MNRAS, 292, 795

Mellema, G., Iliev, I., Alvarez, M., & Shapiro, P. 2005, astro-ph/0508416

Mellema, G., Kurk, J. D., & Röttgering, H. J. A. 2002, A&A, 395, L13

Mellema, G. & Lundqvist, P. 2002, A&A, 394, 901

Mellema, G., Raga, A. C., Canto, J., et al. 1998, A&A, 331, 335

Miranda, L. F. 1999, in ASP Conf. Ser. 188: Optical and Infrared Spectroscopy of Circumstellar Matter, 257

Molster, F. J., Waters, L. B. F. M., Tielens, A. G. G. M., & Barlow, M. J. 2002, A&A, 382, 184

Molster, F. J., Yamamura, I., Waters, L. B. F. M., et al. 1999, Nature, 401, 563

Monaghan, J. J. 1992, ARA&A, 30, 543

Neeman, H. J. 1996, Ph.D. Thesis

Norman, M. L. 2000, in Revista Mexicana de Astronomia y Astrofisica Conference Series, 66–71

O'Dell, C. R. 1998, AJ, 116, 1346

O'Dell, C. R., Balick, B., Hajian, A. R., Henney, W. J., & Burkert, A. 2002, AJ, 123, 3329

O'Dell, C. R. & Burkert, A. 1997, in IAU Symp. 180: Planetary Nebulae, 332

O'Dell, C. R. & Handron, K. D. 1996, AJ, 111, 1630

O'Dell, C. R., Henney, W. J., & Ferland, G. J. 2005, AJ, 130, 172

O'Dell, C. R., McCullough, P. R., & Meixner, M. 2004, AJ, 128, 2339

Oort, J. H. & Spitzer, L. J. 1955, ApJ, 121, 6

O'Shea, B., Bryan, G., Bordner, J., et al. 2004, in Adaptive Mesh Refinement - Theory and Applications, Eds. T. Plewa, T. Linde & V. G. Weirs

Osterbrock, D. E. 1989, Astrophysics of gaseous nebulae and active galactic nuclei (University Science Books)

Ostriker, J. P. & McKee, C. F. 1988, Reviews of Modern Physics, 60, 1

Papaloizou, J. C. B. & Pringle, J. E. 1983, MNRAS, 202, 1181

Pauldrach, A., Puls, J., Kudritzki, R. P., Mendez, R. H., & Heap, S. R. 1988, A&A, 207, 123

Pen, U. 1998, ApJS, 115, 19

Petterson, J. A. 1977, ApJ, 214, 550

Plewa, T. & Müller, E. 2001, Comp. Phys. Comm., 138, 101

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical recipes in FORTRAN (Cambridge University Press)

Pringle, J. E. 1981, ARA&A, 19, 137

—. 1996, MNRAS, 281, 357

—. 1997, MNRAS, 292, 136

Quillen, A. C. 2001, ApJ, 563, 313

Quirk, J. J. 1991, Ph.D. Thesis

Raga, A., López-Martín, L., Binette, L., et al. 2000a, MNRAS, 314, 681

Raga, A. C., Navarro-González, R., & Villagrán-Muniz, M. 2000b, Revista Mexicana de Astronomia y Astrofisica, 36, 67

Raga, A. C., Steffen, W., & González, R. F. 2005, Revista Mexicana de Astronomia y Astrofisica, 41, 45

Reyes-Ruiz, M. & López, J. A. 1999, ApJ, 524, 952

Richling, S. & Yorke, H. W. 2000, ApJ, 539, 258

Ritzerveld, J. 2005, A&A, 439, L23

Ritzerveld, J., Icke, V., & Rijkhorst, E.-J. 2004, astro-ph/0312301

Roe, P. L. 1986, Annual Review of Fluid Mechanics, 18, 337

Sahai, R. 1999, ApJ, 524, L125

Sahai, R. & Trauger, J. T. 1998, AJ, 116, 1357

Schmidt-Voigt, M. & Köppen, J. 1987, A&A, 174, 211

Schwarz, H. E., ed. 1993, Mass loss on the AGB and beyond

Schwarz, H. E., Corradi, R. L. M., & Melnick, J. 1992, A&AS, 96, 23

Shapiro, P. R., Iliev, I. T., & Raga, A. C. 2004, MNRAS, 348, 753

Smith, N., Bally, J., & Morse, J. A. 2003, ApJ, 587, L105

Soker, N. 2001, MNRAS, 328, 1081

Soker, N. & Livio, M. 1989, ApJ, 339, 268

—. 1994, ApJ, 421, 219

Soker, N. & Rappaport, S. 2000, ApJ, 538, 241

Solf, J. & Weinberger, R. 1984, A&A, 130, 269

Speck, A. K., Meixner, M., Fong, D., et al. 2002, AJ, 123, 346

Springel, V., Yoshida, N., & White, S. D. M. 2001, New Astronomy, 6, 79

Steinacker, J., Bacmann, A., & Henning, T. 2002, Journal of Quantitative Spectroscopy and Radiative Transfer, 75, 765

Stone, J. M., Mihalas, D., & Norman, M. L. 1992, ApJS, 80, 819

Su, K. Y. L., Hrivnak, B. J., Kwok, S., & Sahai, R. 2003, AJ, 126, 848

Turner, N. J. & Stone, J. M. 2001, ApJS, 135, 95

Van Leer, B. 1979, J. Comp. Phys., 32, 101

Van Winckel, H., Waelkens, C., Fernie, J. D., & Waters, L. B. F. M. 1999, A&A, 343, 202

Van Winckel, H., Waelkens, C., Waters, L. B. F. M., et al. 1998, A&A, 336, L17

Velázquez, P. F., Riera, A., & Raga, A. C. 2004, A&A, 419, 991

Whitehouse, S. C. & Bate, M. R. 2004, MNRAS, 353, 1078

Whitehurst, R. 1995, MNRAS, 277, 655

Wijers, R. A. M. J. & Pringle, J. E. 1999, MNRAS, 308, 207

Williams, R. J. R. 1999, MNRAS, 310, 789

Willson, L. A. 2000, ARA&A, 38, 573

Xu, G. 1997, MNRAS, 288, 903

Young, K., Cox, P., Huggins, P. J., Forveille, T., & Bachiller, R. 1999, ApJ, 522, 387

Zel'Dovich, Y. B. & Raizer, Y. P. 1967, Physics of shock waves and high-temperature hydrody-
    namic phenomena (New York: Academic Press)

Numerieke Nevels

Numerieke technieken voor het oplossen van de vergelijkingen van de gasdynamica zijn gedurende de afgelopen vijftig jaar steeds verder ontwikkeld. Met behulp van deze technieken kunnen de effecten van in astrofysische objecten veel voorkomende verschijnselen als schokgolven en andere discontinuïteiten accuraat berekend worden. Andere processen die een belangrijke rol spelen in de dynamische evolutie van deze objecten zijn de ionisatie, verhitting en koeling van het gas. In numerieke modellen is het dan ook van belang het effect van deze stralingsprocessen op de energiebalans van het gas mee te nemen. Om het model zo realistisch mogelijk te maken is het verder noodzakelijk dat de simulaties in drie dimensies worden uitgevoerd. Zulke drie-dimensionale stralingshydrodynamische berekeningen zijn met de huidige stand van zaken in supercomputer technologie binnen bereik gekomen.

Eén van de vele mogelijke (astrofysische) toepassingen van deze numerieke technieken is het modelleren van de evolutie van Planetaire Nevels. Deze gasvormige objecten vertegenwoordigen de laatste evolutiefase in het leven van sterren met een massa van $1-8$ zonsmassa's. Gedurende deze fase verliest de ster het grootste deel van haar massa in een aantal, elkaar opeenvolgende, sterrenwinden, waarbij een uitgebreide schil van circumstellair materiaal gevormd wordt. De hete kern van de ster ioniseert en verhit dit materiaal wat vervolgens koelt en daardoor oplicht. Het waargenomen object wordt een Planetaire Nevel genoemd. De wisselwerking tussen de gasdynamische- en stralingsprocessen heeft een grote variëteit aan verschijningsvormen tot gevolg.

Omdat wordt aangenomen dat alle sterren met een massa van $1-8$ zonsmassa's de Planetaire Nevel fase doorlopen, is het van belang hun dynamische evolutie te begrijpen. In dit proefschrift wordt een aantal aspecten van deze evolutie met behulp van numerieke modellen nader onderzocht.

Numerieke technieken

Voor het numeriek berekenen van de evolutie van astrofysische gasstromen wordt vaak gebruik gemaakt van een rekenrooster. Zo'n rooster discretiseert de ruimte in cellen, waarbij elke cel een gemiddelde van de werkelijke fysische toestand van het gas representeert. Om de evolutie van het gas in de tijd te kunnen volgen, wordt een speciaal hiervoor ontwikkelde techniek gebruikt, waarbij de materieflux van cel tot cel wordt bepaald. De flux wordt gebruikt om de nieuwe toestand van het gas in de cel uit te rekenen. Deze zogenaamde 'finite volume' benadering zorgt ervoor dat globale grootheden als massa, impuls en energie behouden blijven. Een ander belangrijk aspect is dat de gebruikte tech-

niek schokken en andere discontinuïteiten in het gas correct volgt. Een aantal methoden om dit te bewerkstelligen is beschikbaar. Vooral de methode van Godunov (1959) is hiervoor zeer geschikt gebleken, omdat deze de discontinuïteiten door middel van slechts een paar cellen kan representeren. Hogere orde uitbreidingen op Godunov's methode maken het mogelijk dat ook de stromingspatronen buiten de gebieden waar de discontinuïteiten optreden met grote nauwkeurigheid zijn te volgen.

De zojuist beschreven methode berekent de stroming van het gas over het gehele computationele domein met eenzelfde, hoge nauwkeurigheid. Echter, voor veel toepassingen is deze nauwkeurigheid slechts nodig in een beperkt deel van het domein, bijvoorbeeld daar waar schokken optreden. Om rekentijd en geheugen te besparen wordt dan ook vaak gebruik gemaakt van de techniek van 'Adaptive Mesh Refinement' (AMR). Het principe achter deze methode is dat de resolutie alleen in die gedeelten verhoogd wordt waar het nodig is, waarbij de rest van het domein op een lagere resolutie wordt uitgerekend. Met behulp van een speciaal criterium worden de cellen gevonden die voor verfijning in aanmerking komen. Deze worden verzameld in rechthoekig gevormde deelgebieden (grids) door gebruik te maken van een patroonherkenningsalgoritme. Door dit proces recursief te herhaling worden de verfijningsniveaus één voor één opgebouwd.

Om de AMR methode vervolgens op een parallelle supercomputer te kunnen gebruiken wordt zogenaamde 'domain decomposition' toegepast. Iedere processor beschikt slechts over een deel van het totale computationele domein, waarbij de subdomeinen met elkaar verbonden worden door middel van zogenaamde 'ghost cells'. Deze 'ghost cells' zorgen er ook voor dat elk grid als een losstaande entiteit door de gasdynamicamethode kan worden benaderd. Details van de hierboven beschreven technieken voor het numeriek berekenen van de gasdynamica met behulp van een geparallelliseerde AMR code staan beschreven in hoofdstuk 2.

Stel nu dat er zich op een bepaalde locatie in het computationele domein een stralingsbron (dat wil zeggen een ster) bevindt. Om de invloed van de straling van deze bron op de dynamica mee te nemen, moeten de ionisatiegraad en temperatuur van het gas bepaald worden. Hiertoe is het noodzakelijk lichtstralen vanuit de bron naar elke cel te volgen om zodoende de invloed van de straling op het gas te kunnen berekenen. Dit wordt echter bemoeilijkt doordat het domein niet alleen verdeeld is over verschillende processoren, maar ook is opgesplitst in grids met verschillende afmetingen, wat het volgen van de lichtstralen bemoeilijkt. In het algemeen zal een lichtstraal over meerdere subdomeinen en grids lopen, zodat het noodzakelijk is de berekening op te splitsen. In hoofdstuk 3 wordt een nieuw ontwikkelde methode beschreven, genaamd *Hybrid Characteristics*, die dit mogelijk maakt. Deze methode is geïmplementeerd in de publiekelijk beschikbare AMR gasdynamica code *Flash*[1].

Een voorbeeld van het opsplitsen van lichtstralen ten gevolge van de AMR structuur van het computationele domein is te zien in Figuur 7.1. In een eerste stap worden de bijdragen ten gevolge van de locale secties van de verschillende lichtstralen uitgerekend. Vervolgens worden deze gecommuniceerd naar alle processoren, waarna elke processor beschikt over alle informatie van alle locale secties. Door de locale secties te interpoleren en vervolgens op te tellen, wordt de totale bijdrage van elke lichtstraal berekend. Het

---

[1] http://flash.uchicago.edu/
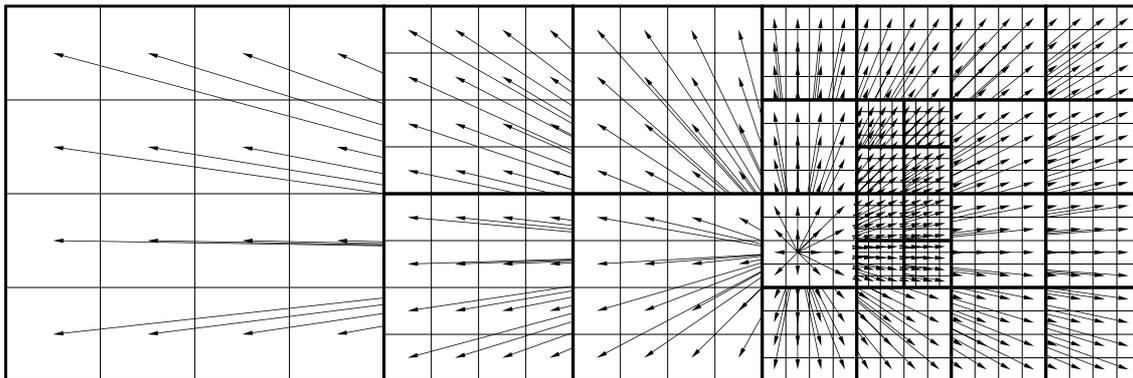
Figuur 7.1: Voorbeeld van het opdelen van lichtstralen ten gevolge van AMR zoals toegepast in de *Hybrid Characteristics* methode.

interpolatieschema is zodanig gekozen dat, in het geval van een homogene omgevingsdichtheid, de exacte oplossing gevonden wordt.

Uitgebreide schalingstesten van de *Hybrid Characteristics* methode worden gepresenteerd in hoofdstuk 4. Hierbij wordt gekeken naar de prestatie van de methode bij een toenemend aantal processoren. Omdat de hoeveelheid gegevens die gecommuniceerd moet worden tussen de processoren toeneemt als functie van hun aantal, is er een limiet aan het aantal processoren waarvoor de methode nog efficiënt te gebruiken is. De testresultaten laten zien dat de methode, afhankelijk van het type supercomputer, praktisch inzetbaar is voor tenminste $100 - 1000$ processoren.

De dynamische evolutie van Planetaire Nevels

De late fasen in de evolutie van een ster met een massa van $1 - 8$ zonsmassa's worden gekenmerkt door massaverlies. Tijdens het stadium van Rode Reus is de ster zodanig in afmeting gegroeid dat de zwaartekracht in de buitenste lagen zeer zwak is. Stofdeeltjes die zich in deze relatief koele lagen kunnen vormen worden versneld door stralingsdruk geleverd door de ster. De stofdeeltjes slepen het gas mee. Dit resulteert in een stellaire wind met een massaverlies van ongeveer 10^{-5} zonsmassa's per jaar en een snelheid van ongeveer 10 kilometer per seconde. Hierdoor verliest de ster in een periode van ongeveer 100 000 jaar het grootste deel van haar massa.

Zodra er nog maar een kleine hoeveelheid materie in de nabijheid van de kern van de ster aanwezig is, begint de Planetaire Nevel fase. De effectieve temperatuur van de kern loopt op tot ongeveer 10 000 Kelvin, wat een tweede fase van massaverlies inzet. Deze nieuwe stellaire wind heeft, ten opzichte van de voorgaande, een veel lager massaverlies van ongeveer 10^{-8} zonsmassa's per jaar en een veel hogere uitstroomsnelheid van meer dan 1000 kilometer per seconde.

Het hierboven geschetste beeld leidt op een vanzelfsprekende manier naar het scenario van de 'Interacting Stellar Winds' (ISW); gedurende de Planetaire Nevel fase botst de snelle wind van lage dichtheid met de langzame wind van hoge dichtheid, die is achtergebleven uit de Rode Reus fase van de ster. Deze interactie resulteert in de formatie van een
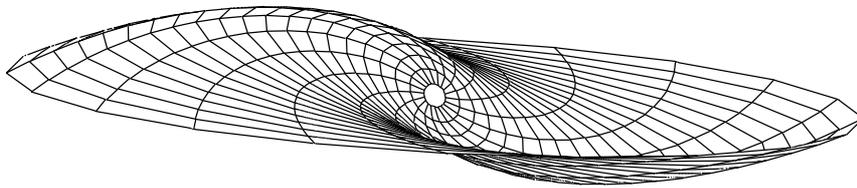
Figuur 7.2: Voorbeeld van een door stralingseffecten vervormde circumstellaire disk.

hete bel van gas met daar omheen een dunne schil van hoge dichtheid. De hoge-energie fotonen van de ster ioniseren en verhitten dit gas, wat de dynamische evolutie beïnvloedt. Het gas in de resulterende nevel koelt en wordt zichtbaar in het optische deel van het spectrum als een Planetaire Nevel.

Alhoewel de waargenomen eigenschappen van sferische Planetaire Nevels met behulp van het ISW-scenario verklaard kunnen worden, zijn de meeste Planetaire Nevels niet exact rond, maar elliptisch of bipolair van vorm. Dit heeft geleid tot een generalisatie van het ISW-scenario. Hier wordt aangenomen dat het massaverlies tijdens de Rode Reus fase niet sferisch, maar asymmetrisch verloopt, waarbij de dichtheid in de langzame wind toeneemt van pool naar equator. Als de snelle wind vervolgens botst met zo'n toroïdale massaverdeling ontstaat een bipolaire nevel. Dit model is zeer succesvol gebleken in het verklaren van de waargenomen eigenschappen van veel Planetaire Nevels.

Hoge resolutie Hubble Space Telescope waarnemingen van jonge Planetaire Nevels laten echter morfologieën zien die moeilijk te verklaren zijn in de context van het gegeneraliseerde ISW-scenario. Met name de veel voorkomende punt-symmetrische of multipolaire nevels vormen een uitdaging.

Eén manier waarop punt-symmetrie kan worden verkregen, voorgesteld door Icke (2003), is te veronderstellen dat de verdeling van massa in de omgeving van de ster puntsymmetrisch van aard is. Zo'n verdeling kan ontstaan als een circumstellaire schijf van gas wordt blootgesteld aan de straling van de ster. Het verschil in stralingsdruk aan het oppervlak van de schijf resulteert in een instabiliteit die er voor zorgt dat de schijf vervormt (zie Figuur 7.2).

In hoofdstuk 5 worden numerieke modellen van de interactie van een snelle wind met zo'n vervormde schijf gepresenteerd. Twee-dimensionale simulaties laten zien dat een boegschok zich rond de binnenrand van de schijf vormt, wat een multipolaire nevel tot gevolg heeft. Door de koelingstijdschaal met de dynamische tijdschaal van het model te vergelijken, kan een bepaald gebied in de parameterruimte aangewezen worden waarvoor dit model geldig is. De resultaten van deze berekeningen laten zien dat het meenemen van de effecten van energieverlies door koeling van essentieel belang zijn om dit model te laten werken. Door de punt-symmetrie van het probleem zijn twee-dimensionale simulaties echter niet voldoende en is het noodzakelijk om drie-dimensionale simulaties uit te voeren. De resultaten van deze berekeningen worden omgezet in synthetische observaties. Door deze te vergelijken met waargenomen Planetaire Nevels, zijn een aantal goede
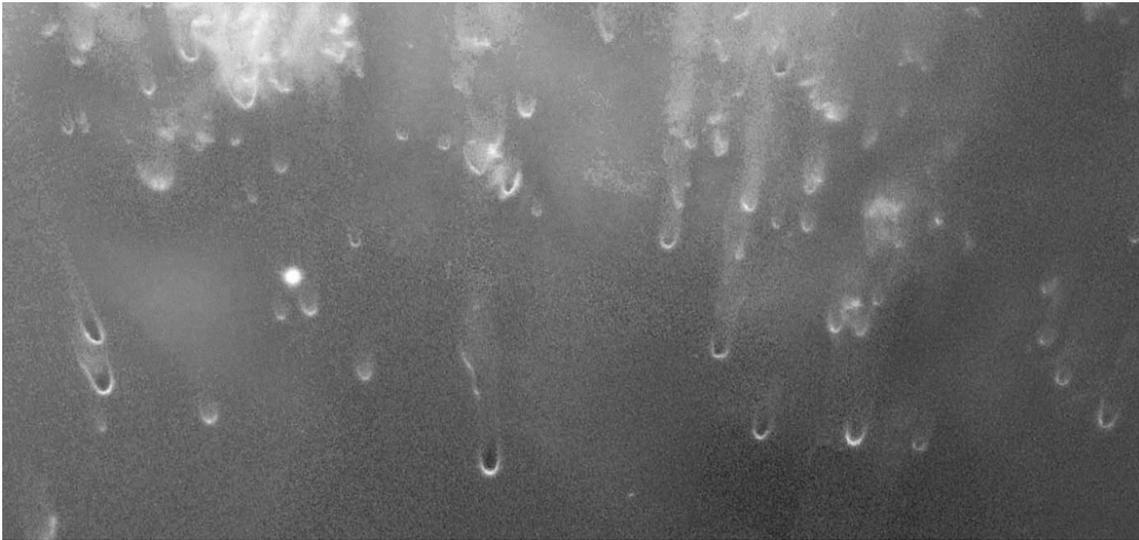
Figuur 7.3: Detail van een Hubble Space Telescope waarneming van kleine schaal structuren in de Helix nevel. De centrale ster, verantwoordelijk voor deze structuren, bevindt zich (op schaal) ongeveer midden onderaan deze pagina.

kandidaten gevonden waarvan de morfologie met dit model verklaard kan worden.

Naast de intrigerende globale verschijningsvormen van Planetaire Nevels, laten waarnemingen ook structuur op kleinere schaal zien. Het beste voorbeeld hiervan is te vinden in de Helix nevel. In deze meest nabij gelegen Planetaire Nevel worden tienduizenden van zulke kleine schaal structuren waargenomen (zie Figuur 7.3). Deze structuren bestaan uit een kern van hoge dichtheid, met 'staarten' die van de centrale ster vandaan gericht zijn. Het gedeelte van de kern dat blootgesteld wordt aan de ioniserende straling van de ster wordt waargenomen als een oplichtende rand van materiaal. Dit impliceert dat het materiaal in de kernen door de straling wordt geëvaporeerd, waarbij een ionisatiefront zich langzaam een weg door de kern baant. Aangezien waarnemingen van ander Planetaire Nevels soortgelijke kleine schaal structuren laten zien, wordt aangenomen dat ze op een bepaald moment in de evolutie van alle Planetaire Nevels voorkomen. Als dit zo is, dan zal een aanzienlijke fractie van de materie die wordt teruggevoerd in het interstellaire medium bestaan uit deze kernen van hoge dichtheid. Omdat de meeste van deze structuren in groepen voorkomen, zodat onderlinge interactie tussen de evaporatie stromen en de hoge dichtheids kernen plaatsvindt, is het wenselijk om de evolutie van een complete groep van deze structuren te volgen.

In hoofdstuk 6 wordt de dynamische evolutie van zo'n groep met behulp van driedimensionale modellen bestudeerd. Eerdere numerieke en analytische berekeningen laten zien dat deze evolutie bestaat uit twee fasen: een compressiefase, waarbij het ionisatiefront een schok induceert, gevolgd door een 'komeet-fase', waar de kern versneld wordt door het zogenaamde 'rocket effect' (Oort & Spitzer 1955). De fysische parameters voor de kleine schaal structuren in de Helix zijn echter enigszins verschillend van wat tot nu toe werd aangenomen voor dit soort simulaties. Dit heeft tot gevolg dat de evolutie op een andere manier verloopt. Doordat voor deze parameters de snelheid van het schok-

front hoger is dan de snelheid van het ionisatiefront, kan zich een relatief dikke laag van geschokt materiaal tussen deze twee fronten vormen. Het gevolg is dat het gas, aan het einde van de compressiefase, begint te expanderen, waardoor ook de 'komeet-fase' op een andere manier verloopt. Analytische berekeningen laten zien dat de voorwaarde voor het optreden van deze expansiefase, identiek is aan de voorwaarde die eerder door Bertoldi (1989) afgeleid is voor het onderscheiden van twee regimes in de foto-evaporatie van dit soort kernen met hoge dichtheid. Dit laat zien dat de expansiefase, die in de numerieke simulaties wordt gevonden, in werkelijkheid een veel voorkomend verschijnsel is. De simulaties tonen verder aan dat, door de interactie tussen de kernen, een groot gedeelte van het geëvaporeerde materiaal in de staartgebieden terecht komt, wat een langere evaporatietijd tot gevolg heeft.

Curriculum Vitae

PERSONALIA

Naam	Erik-Jan Rijkhorst
Geboortedatum	4 februari 1974
Geboorteplaats	Assen

OPLEIDING

1986-1992	VWO, Doctor Nassau College, Assen
1992-1995	Werktuigbouwkunde, Technische Universiteit Delft
	Technische Natuurkunde (propedeuse), Technische Universiteit Delft
1995-1999	Algemene Sterrenkunde (doctoraal), Universiteit van Amsterdam
	- Waarneemstage, Instituto de Astrofísica de Canarias
	- Afstudeeronderzoek *Thick Gravitational Lenses*, onder begeleiding van prof. dr. V. Icke

WERKERVARING

1998-1999	Programmeur, OGD, Amsterdam
2000	Bijlesdocent, Instituut de Leeuw, Amsterdam
2000	Programmeur, Netlinq, Amsterdam
2000-2001	Wetenschappelijk medewerker, TNO TPD, Delft
2001-2005	Assistent in opleiding (promotor: prof. dr. V. Icke), Universiteit Leiden
	- NOVA Herfstschool
	- Workshops/conferenties: Canada, Duitsland, Engeland, México Nederland, USA
	- Werkbezoek Flash Center, University of Chicago, USA

Credits

Promoveren doe je niet alleen. Ik heb dan ook veel te danken aan alle (ex)leden van de theoriegroep: Franco, Huib, Inti, Jelle, Paul, Peter, Rowin, Siard en Sijme-Jan, het was een genoegen om met jullie (samen) te werken. Ook bedank ik graag mijn kamergenoten Inga, Davor, Remco en Stefan, voor de perfecte werksfeer gedurende die dagen dat ik *wel* in Leiden was te vinden. Cyril et Edouard, merci de m'avoir montré le style de vie breton! It's my pleasure to thank everyone at the Flash Center for making my working visits to Chicago such enjoyable ones.

Bartele, Gilbert, Hessel, Nanne, en Nikolai, ik ben blij dat ik zulke fantastische vrienden heb! Maria, it's been great to go through the last bits together! Ten slotte bedank ik mijn zus en ouders voor de continue steun die ze mij de afgelopen 31 jaar hebben gegeven.