

High resolution simulations of the long-term
evolution of jets from young stellar objects using
parallel algorithms

by

Ronan Michael Keegan

for the degree of
Doctor of Philosophy

Dublin City University
Advisor: Dr. Turlough Downes
School of Mathematical Sciences



May 2005

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of PhD. is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Jana Keegan

ID No.: 51174073

Date: 15/7/05

A long time ago, in a GMC far, far away

Table of Contents

Abstract	viii
Acknowledgments	ix
List of Figures	xv
1 Introduction	2
1.1 Jets from Young Stellar Objects: Observations	3
1.1.1 History	3
1.1.2 Formation & Morphology	4
1.1.3 Observational Properties	8
1.1.4 Mass-Velocity Relationship	11
1.2 Jets from YSOs: Theory and Numerical Simulations	12
1.2.1 Theory of formation: different models	12
1.2.2 Mass-velocity relationship: previous results	14
1.3 Parallel Programming	17

1 3 1	Motivation for Parallelising	17
1 3 2	Different Methods of Parallelising	17
1 3 3	MPI - Message-Passing Interface	18
1 4	Structure of this Work	19
2	Numerical Method Hydrodynamics	21
2 1	Computational Fluid Dynamics	21
2 1 1	Astrophysical CFD	22
2 2	Euler equations of hydrodynamics	22
2 2 1	Finite volume methods for solving the equations of hydrodynamics	23
2 2 2	High-resolution Schemes	26
2 2 3	Alternative methods	29
2 3	The Riemann problem	30
2 4	Scheme Used	31
2 4 1	Basic Equations	32
2 4 2	Numerical scheme	33
2 4 3	Riemann Solver	38
2 5	Numerical Setup	41
2 5 1	Computational domain	41
2 5 2	Boundary Conditions	43
2 6	Radiative Cooling & Emissions	44

2 6 1	Cooling	44
2 6 2	Line Emission	46
2 7	Methods for dealing with numerical problems	47
2 7 1	Artificial Viscosity	47
2 7 2	Shear Viscosity	48
2 7 3	Dealing with Negative Pressures	49
3	Numerical Method Code Parallelisation	51
3 1	Simple Splitting	52
3 1 1	Splitting of the computational grid	52
3 1 2	Splitting a 2-dimensional grid	53
3 1 3	Communications Local & Global	57
3 2	Redistributive Splitting & Load Balancing	61
3 2 1	Method for Load-balancing	61
3 2 2	Resizing the grid and redistributing cells	62
3 2 3	Boundary conditions	65
3 2 4	Test Simulation	67
4	Parallel Code Testing	70
4 1	Amdahl's Law	70
4 2	Parallel computing environment	73
4 3	1-D Testing	74

4 4	2-D Testing	74
4 4 1	Simple domain-decomposition method testing	77
4 4 2	Redistributive domain-decomposition method testing	77
4 5	Testing different versions of MPI	86
4 5 1	Features of each Implementation	87
4 5 2	Results and Comparisons	88
5	Long Duration Simulations of jets from YSOs	89
5 1	Initial Conditions	89
5 2	The Pulsed Jet	90
5 2 1	Morphology	90
5 2 2	Mass-velocity Relationship	93
5 2 3	Intensity-velocity Relationship	96
5 3	The Episodic Jet	98
5 3 1	Morphology	101
5 3 2	Mass-velocity Relationship	102
5 3 3	Intensity-velocity Relationship	104
5 4	Behaviour of γ with angle to the plane of the sky	106
6	An examination of the effects of jet episodicity on molecular outflows	112
6 1	Short-period episodically varying jets	114

6 1 1	Morphology	114
6 1 2	The mass-velocity and intensity-velocity relations	118
6 1 3	Comparison with a sinusoidally varying jet	120
6 1 4	Momentum transfer to ambient molecules	123
6 1 5	Pulsed jet and input velocity variation	127
6 2	Long duration simulations	129
6 2 1	Shell Stripping	131
6 2 2	Mass-velocity relation	133
6 2 3	Bipolarity	135
7	Conclusions	138
7 1	Code	138
7 2	Code Parallelisation	139
7 3	Long duration simulations	140
7 4	Input jet velocity variation and its effect on the resulting outflow	141
7 5	Further Work	143
7 5 1	Improvements to the code	143
7 5 2	Further developments for the parallelisation method	144
7 5 3	Further investigation of molecular outflows	145
	Bibliography	147

High resolution simulations of the long-term evolution of jets from young stellar objects using parallel algorithms

Ronan Michael Keegan

ABSTRACT

Outflows and jets are an integral part of the formation of young stars and are found to be commonplace in all regions where star formation is known to occur. There has been much work done in the development of computational fluid dynamic methods for the simulation of these outflows in an attempt to gain a greater insight into the processes taking place in their formation. Observational data presents key characteristics of such outflows that can be used to determine the validity of any computational model.

Here, we have developed a sophisticated parallelisation method for the splitting-up of a jet simulation across a Beowulf type computer cluster using a message-passing method. The parallelised code allows us to run simulations for much longer and on larger domains than was possible with the original serial code. This allows us to investigate the development of some important characteristics of the computational model over large time-scales with a suitably high resolution. In particular we investigate the behaviour of the mass-velocity and intensity-velocity relationships for molecular outflows driven by a prompt-entrainment type jet model. Up to now simulations have indicated good agreement between these characteristics for this model and the observed behaviour of these relationships. However, the short time-scales used did not allow for an evolutionary study of the relationships and as a result long-term simulations are deemed necessary.

ACKNOWLEDGMENTS

There are many people that I am indebted to for aiding me in the work presented in this thesis. I apologise to anyone I may have left out. First and foremost I thank Dr. Turlough Downes for his guidance and helpful contributions to this work, also for his patience. I would also like to thank the people of the School of Mathematical Sciences in Dublin City University for helping me to complete the work and those at the School of Cosmic Physics in the Dublin Institute for Advanced Studies for their contributions and the use of their computational cluster without which this work would not have been possible. In particular I would like to thank Dr. Tom Ray, Dr. Eric van der Swaluw, and also John Walsh for setting up and managing the cluster. A further debt of gratitude is owed to Dr. Stephen O' Sullivan, for explaining the details of the Riemann solver to me.

I also wish to thank my parents Deirdre and Michael Keegan and my brothers Niall, Brian and Ciaran for their support throughout the years. Thanks is also due to Luca Sartori, Thomas Waters, Daire Byrne and Cronan McNamara for making sure that I had plenty of distractions to keep me sane along the way. I should also mention Cornelia Amend for giving me a metaphorical kick up the backside when I needed it.

I would like to thank Dr. Colin Nave and Dr. Martyn Winn of CLRC Daresbury Laboratory for granting me the time and facilities to continue my studies in the final couple of years. I would also like to thank Dave Meredith for his advice on how to actually finish a PhD and John Kewley for helping me to debug my code and teaching me the art of shell scripting.

Finally, I would like to acknowledge Enterprise Ireland, Cosmogrid and the School of Mathematical Sciences in Dublin City University for funding this research without which things would have been a lot more difficult

List of Figures

1 1	Outflow diagram	6
1 2	Outflow collimation in HH 211	7
1 3	Intensity-velocity relations for some observed outflows	10
2 1	Solution to the linearized Riemann problem	31
2 2	The Riemann problem at a cell interface	39
2 3	Boundary conditions set-up for a simulation	43
3 1	Domain splitting for 1-D simulation	52
3 2	Splitting strategies for a 2-D grid	54
3 3	Example of an evolved outflow	54
3 4	Simple splitting of a 2-D grid	56
3 5	2-D stencil for calculating fluxes	58
3 6	Shear velocity calculations	60
3 7	Redistributive parallelisation method splitting of the grid	62
3 8	Resizing the active grid	63

3 9	Flow chart for parallel code	66
3 10	Example simulation using redistributive parallel method	69
4 1	Amdahl's Law	72
4 2	Timing for 1-D parallel code	75
4 3	2-D boundary communications	76
4 4	Timing results for simple parallelisation method	78
4 5	Timing results for optimum time-steps between resizing of active grid	80
4 6	Test grid for communication overhead tests	81
4 7	Percentage of total execution time for communication of cells	82
4 8	Percentage of total execution time for global communications	83
4 9	Percentage of total execution time for redistribution of cells	84
4 10	Speedup of redistribution parallel code	85
4 11	Comparison with speedup of simple parallel method	85
4 12	Percentage of total execution time to do the flux calculations	86
4 13	Speedup tests for LAM and MPICH	87
4 14	Comparison of communication times for LAM and MPICH	88
5 1	Long-term simulation of an outflow driven by a pulsed jet	92
5 2	The $m(v)$ relation at different points in time for the pulsed jet	93
5 3	The $m(v)$ relation for the low velocity component of the outflow	95
5 4	Time-averaged plots for $m(v)$ and $m_{H_2}(v)$ relations	96
5 5	Time-averaged $I_{CO}(v)$ for the pulsed jet	97

5 6	Time-averaged $I_{\text{H}_2}(v)$ for the pulsed jet	97
5 7	Long-term simulation of an outflow driven by an episodic jet	100
5 8	Collimation factor as a function of time for the two outflows	102
5 9	The exponent of the $m(v)$ relation for the two outflows as a function of time	103
5 10	Time-averaged $m(v)$ relation for both simulations	104
5 11	The time-averaged $I_{\text{H}_2}(v)$ relation for the pulsed and episodic jet-driven outflows	105
5 12	The time-averaged $I_{\text{CO}}(v)$ relation for the pulsed and episodic jet-driven outflows	105
5 13	The $m(v)$ relation for the two outflows at different viewing angles	107
5 14	The $I_{\text{CO}}(v)$ for the pulsed jet-driven outflow at different viewing angles	108
5 15	The $I_{\text{H}_2}(v)$ relation for the episodic jet-driven outflow at different viewing angles	109
5 16	The $I_{\text{CO}}(v)$ relation for the pulsed jet at different times	110
5 17	The $I_{\text{CO}}(v)$ relation for the episodic jet at different times	111
6 1	Input jet velocity for the 5 episodic jet tests	115
6 2	Log-scale plots of the five episodic jet simulations	117
6 3	Collimation factors for the five simulations	118
6 4	Comparison of the $m(v)$ relation for the five simulations	120
6 5	Comparison of the $I_{\text{CO}}(v)$ relation for the five simulations	121

6 6	Comparison of the $I_{H_2}(v)$ relation for the five simulations	121
6 7	Input velocity profile for the sine-wave jet	122
6 8	Comparison of the sine-wave jet and episodic jet-driven outflow in terms of density	123
6 9	Velocity profile along the jet axes	124
6 10	Fraction of jet momentum transferred to ambient molecules for the five episodic and the sine-wave outflows	126
6 11	Azimuthal velocity for velocity variation tests	127
6 12	The $m(v)$ relation for velocity variation tests	128
6 13	Log-scale number density plots for the four simulations at 2300 years	130
6 14	Collimation ratio as a function of time for each of the four simulations	131
6 15	Fraction of jet momentum in ambient molecules over the course of each simulation	132
6 16	An example of “shell-stripping” on the inside of the shell	133
6 17	Exponent in the $m(v)$ relation as a function of time for the outflow driven by the episodic jet with a period of 50 years	134
6 18	Comparison of mass-velocity relation for the episodic outflow with the two simulations of Chapter 5	135
6 19	Log-scale number density plots and plots of the jet material tracer for the three outflows	136

6 20 Fraction of outflow material with various velocity vector directions relative to the jet-flow axis for each of the four simulations

137

High resolution simulations of the long-term evolution of jets
from young stellar objects using parallel algorithms

Chapter 1

Introduction

With the advent of infra-red and radio astronomy it was possible to peer into the inner workings of molecular clouds and view the dramatic processes involved in star formation. It is believed that star formation occurs in dense regions or cores within Giant Molecular Clouds as most young stars are first seen in the vicinity of these cores. Most observations have indicated large mass fluxes surrounding the young stars. What surprised observers is the fact that a large proportion of the mass flux (~ 0.1 to $100M_{\odot}$ Lada, 1985, Moriarty-Schieven & Snell, 1988) happens in an outward direction, away from the forming star.

Understanding the nature of these outflows is a highly important part in understanding the overall nature of star formation. Molecular outflows are thought to occur soon after the initiation of the star forming process and are therefore a useful record of the early stages in this process.

This chapter will give an overview of the field of molecular outflows, from an observational and theoretical point of view. We will review the most important physical characteristics of molecular outflows and, in particular, a summary of what is known about the mass-velocity relationship of these outflows is given. An overview of the computational side of this work will also be given.

1.1 Jets from Young Stellar Objects: Observations

1.1.1 History

Outflows driven by young stellar objects (YSOs) are found to be observable over a wide range of wavelengths, from ultra-violet to radio. The first observations of outflows driven by young stars were observed in the visible spectrum and were made in the early 1950's by Herbig (1951) and Haro (1952) who observed small nebulosities with emission line spectra. Soon afterwards, these so called Herbig-Haro objects (hereafter referred to as HH objects) were associated with stellar winds (Osterbrock, 1958) and later recognised as being the result of the interaction of highly supersonic stellar winds with ambient material by Schwartz (1975). A study of the large-scale proper-motion of several HH objects by Cudworth & Herbig (1979) and the detection of highly collimated HH jets by Mundt & Fried (1983), confirmed that wind ejection takes place from young stars.

The first observations of broad CO lines indicative of high velocity molecular gas were in the Orion A molecular cloud (Zuckerman et al , 1976, Kwan & Scoville, 1976). The detailed mapping of the supersonic CO emission in different objects (Snell et al 1980, Rodriguez et al , 1980) showed that outflows occur in two lobes of gas, one blue shifted and one red shifted, lying on either side of the central driving source. Searches since have shown that outflows are common to all star forming regions and occur in low and high mass stars (Bally & Lada, 1983, Edwards & Snell 1982, 1983, 1984). The number of known outflows has increased rapidly in the last 25 years. A recent survey by Wu et al (2004) lists more than 391 known outflows.

New observations using millimetre and sub-millimetre radio telescopes and interferometers, in the 90's, have detected highly collimated outflows with very high velocities. Bachiller & Tafalla (2000) suggest that these new collimated flows are from low mass stars in an earlier stage of evolution and that there is an evolutionary link between them and the "Classical" outflows observed in the 1980's.

1 1.2 Formation & Morphology

The formation of a star is thought to result from molecular clouds collapsing in on themselves under the force of gravity which overwhelms the gas pressure within the cloud core. The pressure can be released via infra-red radiation while the cloud remains transparent. For low mass stars, it takes between 1 and 10 million years for a core to collapse. Gravity is strongest at the centre of the core and as a result the centre collapses more rapidly forming what can be called a “Protostar”. Eventually the protostar becomes opaque as the material in the cloud becomes more concentrated. Infra-red radiation can no longer escape and the pressure builds until it can balance the force of gravity. At this early stage, a low mass protostar can typically have a mass of about 1 solar mass (M_{\odot}). This mass will grow as more material falls onto the surface from the surrounding cloud. A wind forms, similar to the solar wind except much stronger, which can blow some of the in-falling material away. It is not known what causes this wind but it is believed that the initiation of the nuclear process, the fusion of H and He within the protostar’s core, causes it to take place. This fusion process leads to convection within the protostar. The convection then causes rapid rotation of the protostar which gives rise to strong magnetic fields which can carry off material from the protostar’s surface via a wind. Another feature of the protostar at this stage of development is the formation of a disk of material in the equatorial plane surrounding the protostar. As the core collapses to form the protostar, the rotation of the core gets faster because angular momentum must be conserved. In the equatorial plane of the core and protostar, the inertial effect of rotation counteracts the inward pull of gravity and slows the collapse of the cloud. As the collapse continues, rotation becomes more rapid and in-falling material accumulates in a rotating disk.

Despite the wealth of observational data that has been collected, it is not known how these outflows evolve over time or how the mass of the central driving source influences the behaviour of the outflow. There is strong evidence to suggest that the bulk of the molecular material in an outflow is swept up from the ambient medium,

since the masses of the outflows can greatly exceed the masses of the associated stars (Moriarty-Schieven & Snell, 1988) It has also been suggested that the more massive the central star, the less well collimated the outflow (Richer et al , 2000, Churchwell, 2000) However, recent observations by Beuther et al (2002) show that this may just be an observational artifact due to the greater distances to the target sources (on average a few kpc) For low to intermediate mass stars, one of the more popular theories is that molecular outflows are driven by stellar jets with speeds of 100-300 km s^{-1} and densities of about 10^3cm^{-3} (Richer et al , 1992, Padman & Richer, 1994, Bachiller et al , 1995, Davis et al , 2000, Arce & Goodman, 2002b)

In the above model, outflows driven by jets are usually characterised by three main features, the jet, a cocoon surrounding the jet and a bow-shock containing swept-up ambient material surrounding the cocoon Figure 1 1 illustrates this model The commonly-held theory is that jets and their associated outflows form at a very early stage in the protostar's development It is thought that, initially, the outflows are well-collimated with the highest velocity material ($\sim 50\text{-}100 \text{ km s}^{-1}$) forming a well collimated jet-like structure at the centre of a broader cocoon of slower moving material ($\sim 5\text{-}50 \text{ km s}^{-1}$) HH211 (figure 1 2) is a typical example of a highly collimated outflow Older outflows can extend up to several parsecs from the driving source and quite often beyond the confines of the parent cloud (HH92, Bally et al 2002, HH34, Devine et al , 1997) Some, such as HH315, show evidence of precession occurring in the outflow over time, possibly driven by a precession in the driving source caused by the gravitational affects of a binary companion to the protostar (Arcé & Goodman, 2002a, 2002b) Older flows also exhibit signs of the driving source possessing an episodic nature by the presence of high density clumps of molecular material along the outflow axis (Bence et al , 1998) However, a recent study of the CB3 and L1157 outflows by Viti et al (2004) suggests that these clumps may have existed in the ambient medium before the star's formation began

When discussing the collimation of an outflow it is common to refer to a collima-

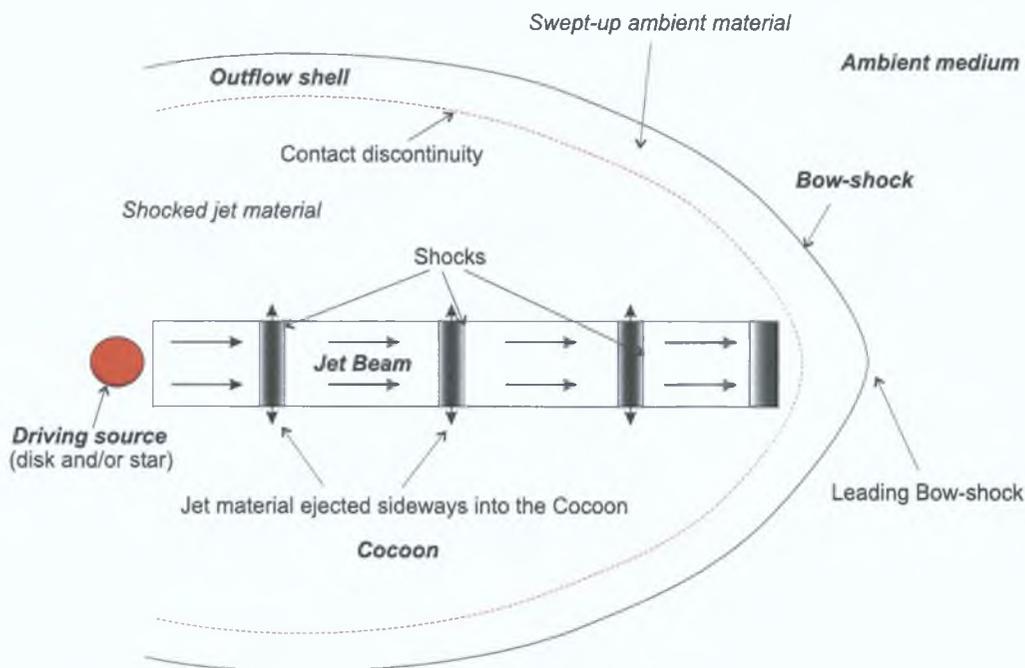


Figure 1.1: An overview of the jet model for creating molecular outflows. This is the model we have adopted in this work and we will refer to the definitions in this figure throughout the text.

tion ratio or factor, q . this is the ratio of the length of the outflow to its maximum width at a given velocity. For example, in the highly collimated outflow HH211 (figure 1.2) this ratio at velocities of $20\text{-}30 \text{ km s}^{-1}$ is found to be 30:1 and for the low velocity components ($\sim 4 \text{ km s}^{-1}$) in the broader shell, the ratio is only 4:1.

It is not known whether the driving jet is composed of molecular or purely atomic material. Also, the mechanism for accelerating the molecular material seen in the outflows remains unclear. The possible mechanisms for accelerating molecular material are entrainment of ambient molecular gas by the bow-shock or sides of the jet, or entrainment by a wide angle wind.

There are many mysteries in the field of molecular outflows. Solving these mysteries requires a combination of observational studies, theory and, more increasingly, numerical simulations.

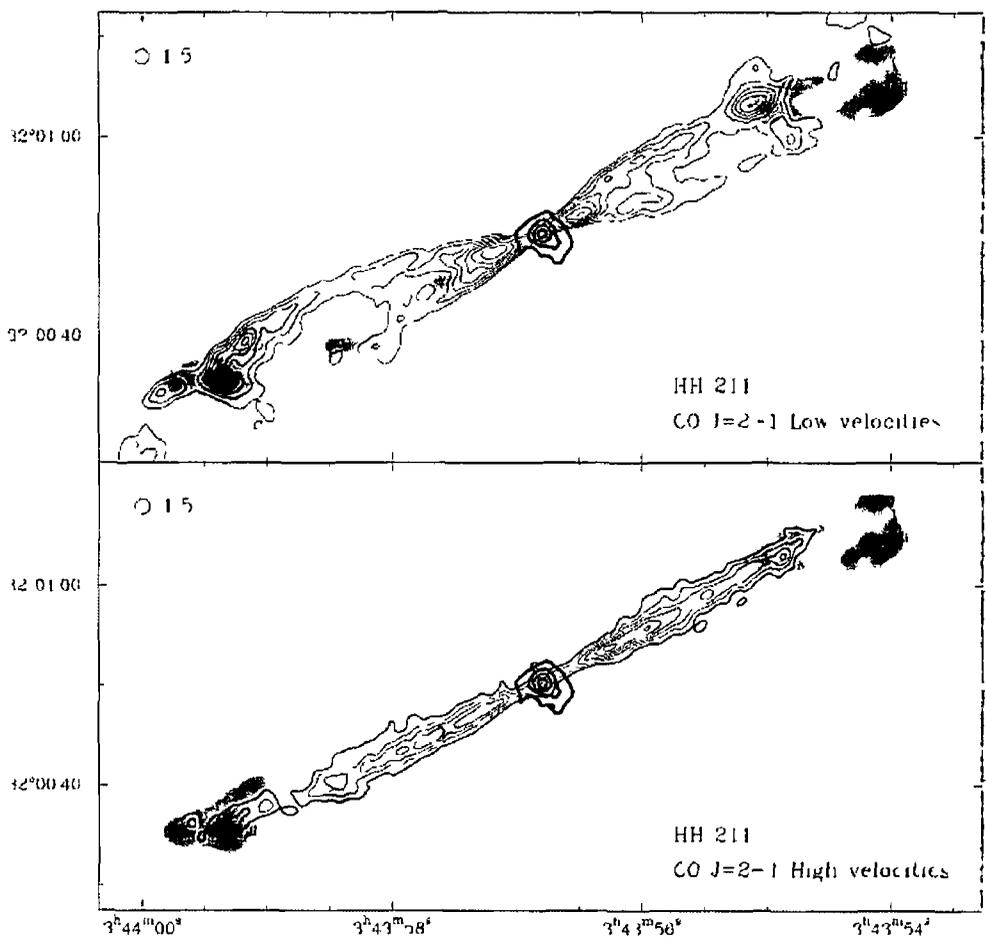


Figure 1 2 This figure is taken from Gueth & Guilloteau (1999) It shows contour plots for the low and high velocity material in the HH 211 outflow in IC348 The top plot shows the slower, less well collimated, material The lower plot shows the high velocity material The velocities were calculated from the CO J=2-1 emission line The grey-scale shows the intensity of emission from the H₂ S1 (1-0) line

1 1 3 Observational Properties

Molecular outflows can be traced at many different wavelengths such as optical and infra-red. However, from the point of view of studying the evolution and morphology of the outflows the most interesting data stems from the low excitation CO emission lines. The CO emission is given off by the cool outflow cocoon region and thus represents a fossil record of the mass-loss from the protostar (Richer et al , 2000). Most observations are conducted in the CO¹² J=2-1 and CO¹² J=1-0 rotational emission lines which are the emissions given off by transitions in the Carbon-monoxide molecule due to collisional and radiative excitations. These transitions results in the cooling of the gas.

Some authors have outlined the typical observational properties of molecular outflows from low-mass stars. Lada & Fich (1996) studied the basic physical parameters of the outflow NGC 2264G and these are seen as typical parameters for most low-mass outflows. Others have described similar outflow properties (Masson & Chernin, 1993, Bachiller, 1996, Cabrit, Raga & Gueth, 1997, Richer et al , 2000). Their main findings are

- The behaviour of the velocity of the material in an outflow can be described by a single “Hubble Law” over the entire extent of the outflow. The velocity of the material making up the outflow appears to increase with increasing distance from the source.
- Outflows are, in general, found to occur in a pair of red and blue-shifted lobes on either side of a central driving source and perpendicular to the plane of the disk of material from which the parent star is forming (Snell et al , 1980). In many cases the outflows are found to exhibit a high degree of bipolarity. In other words, the flows are found to be strongly forward directed with very little velocity components moving in directions other than the jet direction. The ratio between blue and red shifted material in either lobe can be as high as 20 : 1.

- Outflows exhibit a well-behaved broken power-law variation of mass with velocity. A plot of the log-scale velocity against the log-scale mass shows a broken power-law relation with a slope of -1.5 to -3.5 at low velocities ($< 20 - 30 \text{ km s}^{-1}$) and between -4 and -8 at high velocities ($> 30 \text{ km s}^{-1}$). Figure 1.3 shows the mass-velocity relationship for a selection of observed outflows.
- The collimation of the flow increases systematically with velocity and distance from the driving source. At the highest velocities outflows can have very narrow opening angles ($\leq 5^\circ$) and appear jet-like.
- There is often found to be a similarity between the lobes on either side of the central star, particularly in the high velocity components suggesting that the underlying driving wind from the central driving source is characterised by a high degree of bipolar symmetry. However, this is not always the case, as many observations have shown that differing conditions in the ambient molecular cloud, such as a variation in density on either side of the forming star, can result in very different morphologies for the outflow lobes (e.g. HL Tauri, Cabrit et al., 1996). It has also been shown by Woitas et al. (2002) that asymmetries in the “central engine” of the RW Aurigae outflow cause the asymmetries observed in this outflow’s lobes.
- There is increasing evidence that the mechanism driving some outflows exhibits an episodic nature. This is particularly evident in older outflows such as HH 315 (Arce & Goodman, 2002a, 2002b) and in HH 34 (Raga & Noriega-Crespo, 1998).
- For most low-mass YSOs, masses of molecular outflows range from $0.1 M_\odot$ to $100 M_\odot$. This is a lot more than the typical mass of a class 0 low-mass protostar and we conclude that outflows must consist of swept-up ambient material rather than material ejected directly from the central star.

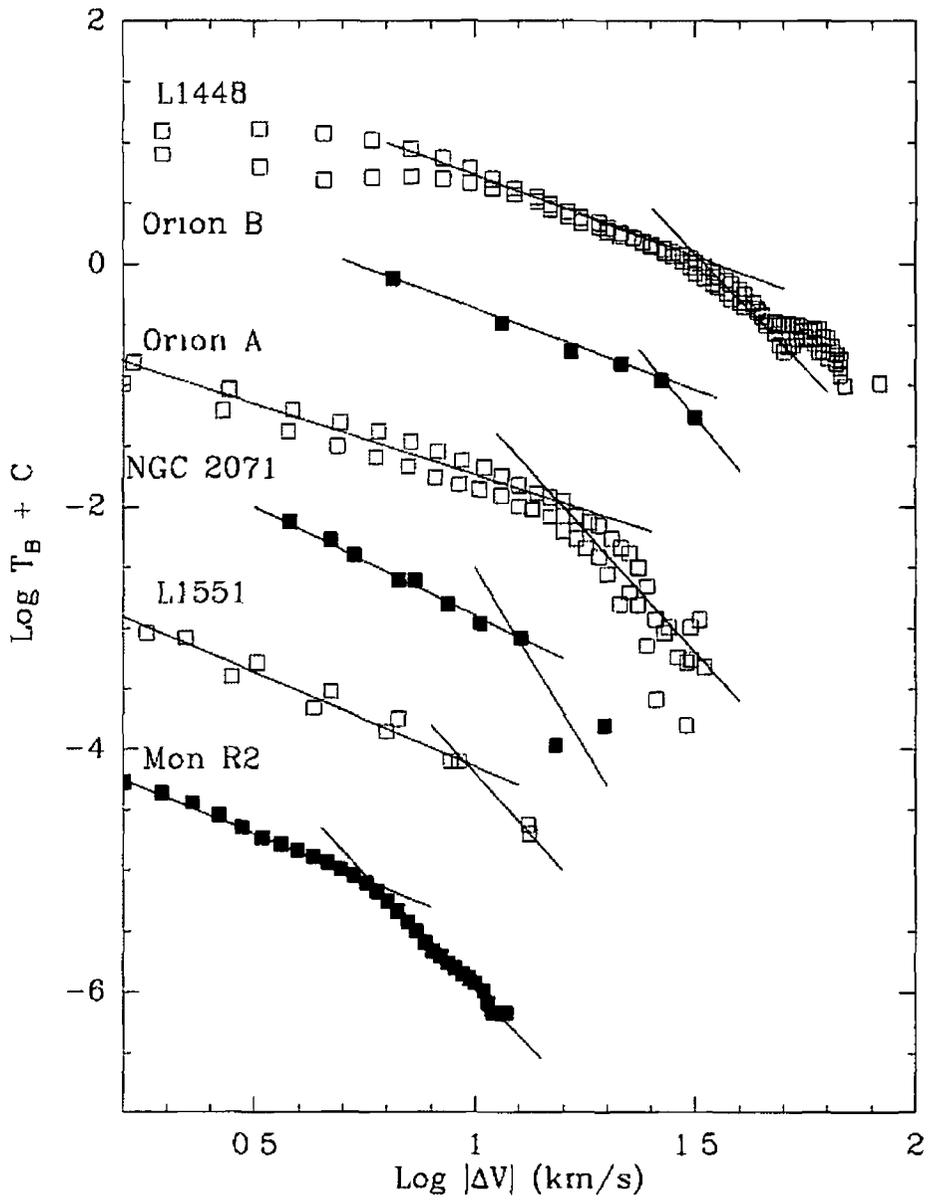


Figure 13 Log-scale brightness distribution with velocity for a selection of different outflows of different ages. It is noticeable that the “break velocity”, where the slope steepens, moves to lower velocities as we go from top to bottom. This behaviour is thought to be related to the outflows age (Taken from Bachiller & Tafalla, 2000)

1.1.4 Mass–Velocity Relationship

One of the key properties of any molecular outflow is the behaviour of the mass in the flow in terms of its velocity. Although not directly measured in observations, the mass can be inferred from the intensity of low-J CO emission lines. Most observational outflow studies estimate outflow mass by assuming that the CO emission is optically thin (emissions from all parts of the outflow are not obscured by material residing between the observer and the point of emission) and the temperature of the gas is roughly constant. The intensity-velocity relationship for this line is usually found to exhibit a broken power-law (Rodríguez et al., 1982, Masson & Chernin, 1992, Stahler, 1994, Lada & Fich, 1996, Bachiller & Tafalla, 2000, Davis et al., 2000, Arce & Goodman, 2002a). It is important for any theoretical model to be able to exhibit the same behaviour. The intensity-velocity relation for the low-J CO lines can be defined as follows

$$I_{CO}(v) \propto v^{-\gamma} \quad (1.1)$$

where, from observation, $\gamma \sim 1.5 - 2$ up to $\sim 20 - 30 \text{ km s}^{-1}$ and $\gamma \sim 3 - 7$ at higher velocities. The profile is constructed by creating a set of velocity bins with a velocity range for each bin. All of the mass moving at a particular velocity range is added into the corresponding bin and the resulting data is graphed on a log-scale plot. v is the flow velocity relative to the systemic rest velocity v_0 . It is worth pointing out at this stage that in many of the simulations of molecular outflows that have been performed (see section 1.2.2 and references therein) there is found to be no break in the mass-velocity relationship even though a break in the corresponding intensity-velocity relationship is predicted. We shall discuss possible reasons for this in Chapter 5.

It is speculated that the position of the break in the slope of the intensity-velocity relation is related to the age of the outflow. Whether the break-point moves to lower or higher velocities with increasing age remains unclear. A sample set of outflows at different stages of evolution compiled by Bachiller & Tafalla (2000) indicated that

the older the outflow, the higher the velocity at which the break occurs (figure 1.3). However, a similar survey by Salas & Cruz-Gonzalez (2002) found that the break-point velocity gets lower with increasing age of the outflow. With the long duration simulations presented in this work, we shall investigate this property.

Arce & Goodman (2001) claim that many outflows have an episodic mass ejection from the driving source. They found that this may cause outflows to have steeper slopes for their mass-velocity relations than has been inferred from most observations. To account for this, they suggest that a lot of observers fail to account for the true optical thickness of CO emission and that the optical thickness is dependent on the velocity. They suggest that with adequate information one could reconstruct the past ejection history of an outflow from the “fossil record” embedded in a molecular outflow’s mass-velocity and position-velocity relations. They also propose that each mass ejection episode can result in the sweeping-up of more ambient material. Using our model, we shall run simulations with episodic velocity inputs and investigate whether or not this occurs.

1.2 Jets from YSOs: Theory and Numerical Simulations

1.2.1 Theory of formation: different models

In this section we shall outline the three main models put forward for the formation of molecular outflows.

Wide-angle wind model

This model consists of a wind generated from the surface of the young star or its circumstellar disk (Shu et al., 1991). This wind can sweep up ambient material and the resulting outflow can have the velocity, shape and extent as seen in observations.

The main problem with this model is that there is no necessity for a jet component. It is accepted now that outflows contain a well collimated jet component and that there is a strong link between the jet and the behaviour of the molecular outflow. Masson & Chernin (1992) found that wind driven models did not reproduce the observed mass properties such as the “Hubble Law” profile for the velocity structure. They found that an entraining jet model gave better results. In an attempt to address this problem, it has been suggested by some that if the wind has an axial density gradient, the core of the wind can take on the appearance of a collimated jet (Shu et al, 1995, Ostriker, 1997, Shang, Shu & Glassgold, 1998). A study by Lee et al (2000) of several young stellar systems, showed that some molecular outflows exhibit features consistent with the wide angle wind model that cannot adequately be explained by the simple jet model. Examples include L43 (Bence et al, 1998), L1551 (Moriarty-Schieven et al, 1988) and B5 (Velusamy & Langer, 1998). Lee et al (2000) concluded that more detailed calculations encompassing aspects of both models may be necessary. Some other authors (Bence, 1998, Arce & Goodman, 2002a) have suggested that the wind emerging from the disk around the protostar becomes more significant in the evolution of the molecular outflow as the star ages.

Turbulent Jet model

In this model molecular material is entrained on the sides of a fast moving jet by some process of turbulent mixing such as the Kelvin-Helmholtz instability at the jet-ambient gas interface (Raga et al, 1993, Stahler, 1994, Lizano & Giovanardi, 1995, Downes & Ray, 1998). The advantage of this model is that it is able to produce the forward directed motion for the outflowing gas as is seen in observations. However, the model by Raga et al (1993) produced flows that were too narrow and did not produce the Hubble law effect showing an increase in the velocity of the gas with distance from the source.

Bow-shock model

This is the model we have adopted in this work (figure 1.1). With this model, a supersonic wind or jet propagates into the ambient medium forming a bow-shock surface at its head (Masson & Chernin, 1993, Raga & Cabrit, 1993, Stone & Norman, 1993, Smith et al., 1997, Downes & Ray, 1999, Lee et al., 2001). This bow-shock carves into the ambient cloud, accelerating the ambient material as it hits it, thereby producing a molecular outflow propagating in the direction of the jet. A low-density cocoon forms around the jet between it and the sides of the bow-shock. Variations in the velocity of the jet or an episodic nature in the output from the jet source cause shocks to occur along the body of the jet. These shocks are also referred to as internal working surfaces (Raga et al., 1990, Canto, Raga & D'Alessio, 2000) and are thought to result in the ejection of jet material from the jet into the surrounding cocoon.

One of the main problems with this model is the difficulty it has in reproducing the forward motion for the swept-up outflow material as discussed by Lada & Fich (1996).

Other models

As outlined above, both the jet model and the wind model have problems when it comes to explaining the formation of molecular outflows (Cabrit, Raga & Gueth, 1997). Lery (2002) proposed an alternative model for outflows in which the molecular outflow is powered by in-falling matter as well as the accretion-ejection engine at the driving source.

1.2.2 Mass-velocity relationship. previous results

Several papers have been published about attempts made to reproduce the observed mass-velocity relation using the jet entrainment model. Here, we shall review some of the more recent publications with particular emphasis on those involving jet-driven

outflow simulations. In all cases the relation is found to be a negative power-law with the form $m \propto v^{-\gamma}$ and no break at higher velocities.

Smith et al. (1997) used a bow-shock model with post-shock mixing. They predicted a value for γ that increased with time. This was due to gradual changes in the bow-shock shape. It became more aerodynamic as time progressed and this resulted in less material being entrained at the bow-shock, hence, reducing the high velocity mass component of the outflow. They constructed a simple analytic model for the bow-shock that predicted that γ is related to a bow shape parameter “ s ” by $\gamma = s/2$. Here the bow-shock shape is approximated to a simple parabola and is described by the relationship $z \propto r^s$ in cylindrical coordinates. The value for γ increases from 1.0 to 1.5 as the value for s goes from 2 to 3.

Downes & Ray (1999) used a bow-shock model but without post-shock mixing and also found a power-law relationship between the mass of molecular gas and velocity. They determined that γ lies between 1.58 and 3.75 and tends to increase with time, in agreement with Smith et al. (1997) and also consistent with observations (e.g. Davis et al., 1998). The key features they investigated suggested that γ increases with decreasing abundance of molecular material in the jet, the decrease in the molecular abundance causes a reduction in the strength of the high velocity component. They compared a steady jet (input velocity $\sim 216 \text{ km s}^{-1}$) with one that had a variation in the input velocity and found that it did not have any effect on the value of γ . Also, it was determined that a wide shear layer dramatically reduces the value of γ .

A follow-up paper to this by Downes & Cabrit (2003) used similar numerical simulations to examine the intensity-velocity relations in the CO J=2-1 and H₂ S(1)1-0 lines for jet-driven molecular outflows. It was shown that the mass-velocity relation for the swept-up gas is a single power-law, with a shallow slope of $\simeq 1.5$ with no break to a steeper slope at higher velocities (contrary to predictions of Zheng and Zhang (1997)). Here too, the $m(v)$ relation could be explained by a bow-shock model with no post-shock mixing. They showed that the simulated CO J=2-1 line compared well

with that of observations. This line had a break in the slope at $20\text{-}30\text{km s}^{-1}$ caused by molecular dissociation near the bow-shock and a $1/T$ dependence for emission at temperatures exceeding the energy of the upper level of the line. The model also predicts a shallower slope at high velocity in the higher excitation lines $\text{H}_2\text{ S}(1)(1\text{-}0)$ and high J CO lines.

They concluded that the CO $J=2\text{-}1$ line only reflects mass at lower velocities ($\leq 20\text{ km s}^{-1}$) and that the use of higher temperature tracers is needed to probe the mass distribution at higher velocities. One weak point of the bow-shock model with no post-shock mixing is that the emitting gas expands perpendicular to the bow surface and this contradicts the observations of Lada & Fich (1996) who observed that outflows exhibit a strongly forward directed motion. Downes & Cabrit (2003) constructed an analytical model that gave the value for γ in terms of the bow shape parameter, s , as being $s/(s - 1)$. This gives a lower value for γ than that predicted by Smith et al (1997) and possibly more in agreement with observations.

Lee et al (2001) conducted simulations of jet and wind driven outflows using the ZEUS-2D hydrodynamic code with simplified cooling and no external heating. This, they said, was a reasonable simplification to make as they were looking at the macroscopic properties of the outflow. However, this also meant that they were unable to resolve the cooling length and hence the temperature of the shell material in the outflow could not be used to calculate the line emission. They used the mass rather than line emission to examine the kinematics. For the jet-driven simulations they found a value for γ in the range 1.5 to 3.5 which was highly dependent on viewing angle. The wind driven simulations yielded a value of 1.3 - 1.8 for γ . They concluded that both models correlated well with different sets of observations and that a more complex model is obviously needed to describe the behaviour of all molecular outflows.

1.3 Parallel Programming

1.3.1 Motivation for Parallelising

Downes & Ray (1999) conducted several simulations of outflows from YSOs based on the same numerical scheme as is used in this work. Their simulations were limited to 400 years in length, due to computational resource limitations. They found that the exponent in the mass-velocity relationship, $m(v)$, γ , agreed well with that of observed values in real outflows. However, they found that the value for γ was also increasing with the age of the outflow and concluded that longer simulations were required to determine whether this trend would continue or if γ would tend to a constant value at longer time-scales. This is the main motivation for the parallelisation of the code. By parallelising the code for operation on a distributed memory cluster, we can gain both a speed-up of the simulations and the ability to run larger simulations by utilising the combined memory of all the machines we employ. However, the adaptation of the code to run to longer time-scales also gives us the ability to look at other interesting characteristics of the simulated outflows and how they behave over time.

1.3.2 Different Methods of Parallelising

Parallel programming has become one of the most important computational tools for tackling large, computationally intensive, programming tasks. It is used for many different applications, from molecular dynamics to computational fluid dynamics. There are two main designs for parallel computers, shared memory and distributed memory. In a shared memory machine all of the processors have access to the same memory space. This negates the need for any sort of communication between the processors and computer code does not need much modification to work on such a machine. However, this type of machine can be expensive to buy and does not scale very well if large numbers of processors are required. Distributed memory machines usually consist of a cluster of standard PC-type machines linked together via a high-

speed network. The advantage of this type of system is that these machines can be built relatively cheaply and the performance of them scales well. The disadvantage is that computer code has to be modified to allow for communications between the individual machines in the cluster.

In this work we have utilised a “Beowulf” type cluster to conduct the astrophysical fluid dynamic simulations. There are two main types of parallelism that can be implemented on a machine of this type

- The simplest is the master-worker approach where the computational task can be broken into independent tasks. The master processor coordinates the solution of these independent tasks carried out by the worker processors.
- The second type consists of tasks that cannot easily be broken into independent tasks. The most common model used for this type of programming is ‘message-passing’. The two most common message-passing systems are MPI (Message Passing Interface) and PVM (Parallel Virtual Machine).

The coding in this work falls into the second category as communicating between processors will be required to maintain the overall coherency of the simulation across the cluster. We have chosen to use the MPI system to implement the parallelisation of the code.

1 3 3 MPI - Message-Passing Interface

Message-passing is one of the simplest ways to get a group of machines to coordinate themselves when executing a complex computation. MPI addresses the message-passing model of parallel computation, in which processors synchronize with one another and move data from one processor to another by sending and receiving messages. It is a library of subroutines written in the most commonly used scientific programming languages such as Fortran and C. Codes written in these languages can call upon these subroutines to perform the message-passing. MPI is a specification

(not an implementation) created by the MPI forum, a group of computer vendors, computer scientists and users who came together to define a standard. The first standard was released 1994 and is referred to as MPI-1 (1994). Since this release extra functionality and improvements have been made and a second release has been made, MPI-2 (1998).

1.4 Structure of this Work

In this thesis we study the evolutionary behaviour of several properties of molecular outflows driven by the jet-driven bow-shock model. To achieve the long time-scales required for this we have developed a load-balanced, parallel version of a temporally and spatially second order, non-adiabatic, cylindrically symmetric, hydrodynamic code for running on a distributed memory cluster. We have also examined in detail the properties of molecular outflows driven by episodic jets.

Chapter 2 details the numerical method used to integrate the hydrodynamic, ionisation and dissociation equations. Chapter 3 describes the method developed to efficiently parallelise the code. In Chapter 4 the tests carried out to examine the efficiency of the parallel code are described.

In Chapter 5 we discuss the long-term evolutionary behaviour of the mass-velocity and intensity-velocity relations for molecular outflows driven by a pulsed jet and an episodic jet with a long-period (800 years) episodicity. It was found by Smith et al. (1997) and Downes & Ray (1999), for outflow simulations up to 400 years, that the slope in these relations agreed with observations but increased with time. We examine whether or not this upward trend continues for longer term simulations. The simulations presented here are up to 2400 years in age, a significant fraction of the dynamical age of at least some molecular outflows. We also look at the behaviour of the break-point velocity in the intensity-velocity relations as a function of outflow age.

In Chapter 6 we look in detail at the effects of varying the input jet velocity on the resulting outflow. In particular, we look at episodic jet-driven outflows. It has been suggested (Arce & Goodman, 2001a) that many molecular outflows are driven by jets with an episodic nature. We look at how different periods for the episodicity of the jet effects the mass-velocity and intensity-velocity relations, how it effects outflow collimation and what impact it has on the amount of ambient material swept-up by the outflow.

Finally, in Chapter 7 we summarise the results of this work and discuss possibilities for further studies.

Chapter 2

Numerical Method:

Hydrodynamics

In this chapter we present the methods used to integrate the hydrodynamic equations. We shall also discuss the numerical set-up. The method used for the simulation of the outflows is based on that used by Downes & Ray (1999) and consequently the code used is an adaptation of the code used by these authors. The numerical scheme is a Godunov type scheme that is second order in time and space. A brief overview of the different numerical schemes used in astrophysical computational fluid dynamic problems is given before we outline the details of the scheme we have employed. We shall discuss the method used to parallelise the code in Chapter 3.

2.1 Computational Fluid Dynamics

The physical aspects of any fluid flow are governed by three fundamental principles (1) mass is conserved, (2) $F = ma$ (Newton's Second Law), (3) energy is conserved. These fundamental principles can be expressed in terms of mathematical equations, which in their most general form are usually partial differential equations. Computational fluid dynamics is the art of replacing the governing partial differential equations with algebraic equations, and advancing these numbers through space and time to obtain a final numerical description of the complete flow field.

2.1.1 Astrophysical CFD

There are many phenomena in astrophysics which can be modelled using computational fluid dynamics. They can range from the very slow progression of the stars in a galaxy to the relativistic speeds of an exploding supernova. Jet's from YSOs usually involve speeds on a supersonic scale. To ensure correct treatment of all the physical conditions that may arise in the simulation of these jets, such as shocks and rarefactions, we must choose an appropriate numerical scheme. The computational time required to solve the scheme must also be considered and a balance needs to be made between the sophistication of the scheme and time it takes to compute.

2.2 Euler equations of hydrodynamics

In gas dynamics the conservation equations of mass, momentum and energy must be solved. For the sake of simplicity we shall only consider the 1-dimensional case. The equations are given by

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} = 0 \quad (2.1)$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} = 0 \quad (2.2)$$

$$\frac{\partial e}{\partial t} + \frac{\partial[u(e + p)]}{\partial x} = 0 \quad (2.3)$$

where ρ is density, u is velocity, p is pressure and e is the total energy density. t and x indicate time and space respectively. These three conservation laws give the Euler Equations of gas dynamics. The value of p is a function of ρ , ρu and e and is given by the equation of state for a polytropic gas

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2 \quad (2.4)$$

where γ is the ratio of specific heats in the gas and is given by $\gamma = (\alpha + 2)/\alpha$, where α is the number of internal degrees of freedom of the gas molecules. In astrophysics, a monatomic gas is usually assumed, giving values of $\alpha = 3$ and $\gamma = 5/3$.

We can consider these three state variables in vector form as

$$\mathbf{q}(x, t) = \begin{bmatrix} \rho(x, t) \\ \rho(x, t)u(x, t) \\ e(x, t) \end{bmatrix} \equiv \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

so that we can express the system of equations in the simplified form as

$$q_t + f(q)_x = 0 \quad (2.5)$$

where

$$f(q) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e + p) \end{bmatrix} = \begin{bmatrix} q_2 \\ q_2^2/q_1 + p(q) \\ q_2(q_3 + p(q))/q_1 \end{bmatrix}$$

More generally, the integral form for a system of m equations says that

$$\int_{x_1}^{x_2} q(x, t_2) dx = \int_{x_1}^{x_2} q(x, t_1) dx + \int_{t_1}^{t_2} f(q(x_1, t)) dt - \int_{t_1}^{t_2} f(q(x_2, t)) dt \quad (2.6)$$

for all x_1, x_2, t_1 and t_2 . These integral forms of the conservation law are the fundamental conservation laws and hold even if the function $q(x, t)$ is discontinuous. The differential form, equation 2.5, holds only if q is smooth.

2.2.1 Finite volume methods for solving the equations of hydrodynamics

There are three main methods used for the discretisation of the equations for hydrodynamics when solving them on a computer. Discretisation of the partial differential equations is called finite difference, discretisation of the integral form of the equations is called finite volume, and the third method is finite element in which one discretises the solution of the differential equations. The discrete solution is made up of polynomials defined piecewise on the elements. In this work, the discretisation is done using a two-dimensional, cylindrically symmetrical, uniform grid of cells and we use a finite volume method.

For a finite volume method, the change in the state of a variable, Q_i^n , within a given volume, V_i , from time n to $n + 1$, can be written in the flux-differencing form as

$$Q_i^{n+1} = Q_i^n - \frac{k}{h}(F_{i+1/2}^n - F_{i-1/2}^n) \quad (2.7)$$

where $F_{i\pm 1/2}^n$ are some approximations to the average fluxes along the interfaces $x = x_{i\pm 1/2}$ between any two adjacent cells and are given by

$$F_{i\pm 1/2}^n \approx \frac{1}{k} \int_{t_n}^{t_{n+1}} f(q(x_{i\pm 1/2}, t)) dt \quad (2.8)$$

with $k = \Delta t$, the time-step value (we shall refer to $h = \Delta x = \Delta y = \Delta z$ as the spatial step-size, assuming a finite volume mesh with equal spacing in all directions) As information propagates at a finite speed we can obtain a value for $F_{i-1/2}^n$, for example, based on Q_i^n and Q_{i-1}^{n-1} , the cell averages on either side of the interface. We can then express $F_{i-1/2}^n$ as

$$F_{i-1/2}^n = F(Q_{i-1}^n, Q_i^n) \quad (2.9)$$

where F is some numerical flux function. Equation 2.7 then becomes

$$Q_i^{n+1} = Q_i^n - \frac{k}{h}(F(Q_i^n, Q_{i+1}^n) - F(Q_{i-1}^n, Q_i^n)) \quad (2.10)$$

The specific method obtained depends on how we choose the formula F . Equation 2.10 is also said to be in conservation form since it mimics the exact solution as given in equation 2.6

Lax-Friedrichs

The simplest way to define the average flux, $F(q_l, q_r)$, at a point based on data to the left and right of that point, q_l and q_r , would be to take an average

$$F(q_l, q_r) = \frac{1}{2}(f(q_l) + f(q_r)) \quad (2.11)$$

which would give

$$Q_i^{n+1} = Q_i^n - \frac{k}{2h}(f(Q_{i+1}^n) - f(Q_{i-1}^n)) \quad (2.12)$$

However, this turns out to be unstable. Instead, we can introduce a diffusive flux or artificial viscosity term based on an approximation to $\frac{h^2}{2k}q_x$. The flux then becomes

$$F(q_l, q_r) = \frac{1}{2}(f(q_l) + f(q_r)) - \frac{h}{2k}(q_r - q_l) \quad (2.13)$$

The conserved form of an equation solved on a grid should be given by the equation

$$Q_i^{n+1} = \frac{1}{2}(Q_{i-1}^n + Q_{i+1}^n) + \frac{k}{2h}(f(Q_{i+1}^n) - f(Q_{i-1}^n)) \quad (2.14)$$

This is the Lax-Friedrichs Equation. Although it is robust, it is inaccurate when used for modelling supersonic flow due to the fact that it gives solutions that are formally first order accurate and excessively diffusive.

Lax-Wendroff

The Lax-Wendroff method is second order accurate and is derived from the Taylor series expansion for the solution at $q(x, t+k)$. The expansion is truncated to the first three terms to give second order accuracy. The solution for the state variable in cell i at time $n+1$ is given as

$$Q_i^{n+1} = Q_i^n - \frac{k}{2h}A(Q_{i+1}^n - Q_{i-1}^n) + \frac{k^2}{2h^2}A^2(Q_{i+1}^n - 2Q_i^n + Q_{i-1}^n) \quad (2.15)$$

This is the generalised form for a system of m hyperbolic equations of the form $q_t + Aq_x = 0$ where A is an $m \times m$ matrix. This too is a robust method, but in regions of the flow where data is not smooth it suffers from excessive oscillations.

Godunov

Godunov's method is based on using a Riemann solver to solve for the fluxes at cell interfaces. It assumes a piecewise-constant distribution of a quantity Q_i^n in a grid cell.

The most important condition, when solving for the fluxes, is to ensure that a wave from the Riemann problem at one interface does not reach an adjacent interface within the time-step. If this is violated, then there is the potential for numerical instabilities to occur. If we take s_{max} as being the maximum wave speed, the condition requires that

$$\frac{ks_{max}}{h} \leq 1 \quad (2.16)$$

This is the CFL or Courant condition that is also required for stability. The CFL condition states that a numerical method can be convergent only if its numerical domain of dependence contains the analytical domain of dependence of the equations being solved.

In the simplest case, for scalar advection, solving the Riemann problem gives

$$q^*(q_l, q_r) = \begin{cases} q_l & \text{if } u > 0 \\ q_r & \text{if } u < 0 \end{cases} \quad (2.17)$$

where $q^*(q_l, q_r)$ denotes the exact solution to the Riemann problem at $x = 0$. The case where $u = 0$ is not relevant given that we are only interested in the flux values and if $u = 0$ then $f(q^*) = 0$. Applying this to the conservative method (equation 2.10), gives Godunov's Upwind Method. The solution for Q_i^{n+1} is given by

$$Q_i^{n+1} = \begin{cases} Q_i^n - \frac{k}{h}u(Q_i^n - Q_{i-1}^n) & \text{if } u > 0 \\ Q_i^n - \frac{k}{h}u(Q_{i+1}^n - Q_i^n) & \text{if } u < 0 \end{cases} \quad (2.18)$$

The method is easily generalised to a non-linear system if we can solve the non-linear Riemann problem at each cell interface.

2.2.2 High-resolution Schemes

Godunov's method, at best, is first order accurate in smooth regions and generally gives very smeared approximations to shock waves or other discontinuities. Many extensions to Godunov's method have been developed. The desired features of these methods are to give second-order accuracy on smooth flow and to avoid non-physical

oscillations near discontinuities. To do this we need to limit the slope of the second-order function in the region of discontinuities so as to avoid the oscillations. Methods based on this approach are known as “slope limiter” methods.

Van Leer

The slope-limiting method was introduced by van Leer when he developed the MUSCL scheme (Monotonic Upstream-centred Scheme for Conservation Laws (van Leer, 1977)). The idea behind incorporating slope-limiting is to prevent oscillations in the presence of discontinuities while maintaining second order accuracy. This is achieved by incorporating a second order accurate method in smooth regions, such as Lax-Wendroff’s with a piecewise linear set-up, and limiting the slope of the piecewise linear approximations in the presence of discontinuities. A similar procedure is to use flux-limiting, reducing the magnitude of the numerical flux to avoid oscillations. This method was first used by Boris & Book (1973) in their flux-corrected transport (FCT) algorithms.

One method of measuring the effectiveness of this type of scheme for eliminating oscillations is to test whether it satisfies the total variation diminishing (TVD) condition. This was first introduced by Harten (1983) for analysing numerical schemes. The total variation of a function Q across a grid can be given by

$$TV(Q) = \sum_{i=-\infty}^{\infty} |Q_i - Q_{i-1}| \quad (2.19)$$

If a method introduces oscillations we would expect the total variation of Q to increase with time. Hence, for a scheme to satisfy the TVD condition it must satisfy the following

$$TV(Q^{n+1}) \leq TV(Q^n) \quad (2.20)$$

One method of limiting the slope is called the minmod slope. It is defined as

$$\sigma_i^n = \text{minmod}\left(\frac{Q_i^n - Q_{i-1}^n}{h}, \frac{Q_{i+1}^n - Q_i^n}{h}\right) \quad (2.21)$$

where σ_i^n is the slope in cell i at time n and the minmod function of two arguments is defined by

$$\text{minmod}(a, b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases} \quad (2.22)$$

This also satisfies the TVD condition

Flux-Vector Splitting

The Godunov method can be interpreted as a flux-difference splitting method as the difference $f(Q_i) - f(Q_{i-1})$ is split into a left going portion which modifies Q_{i-1} and a right going portion which modifies Q_i . An alternative to this, introduced by Sanders & Prendergast (1974) and later by Steger & Warming (1981), is to split each flux $f(Q_i)$, evaluated at the cell average, into a left going part f_i^- and a right-going f_i^+ , so we have

$$f(Q_i) = f_i^- + f_i^+ \quad (2.23)$$

The flux at an interface between two cells can then be defined as

$$F_{i-\frac{1}{2}} = f_{i-1}^+ + f_i^- \quad (2.24)$$

A method of this form is called a flux-vector splitting method. This method is found to be robust for strong shocks and expansion waves and is used heavily in astrophysics, however, it can be excessively dissipative at contact discontinuities (boundary and shear layers). Several improvements have been made to it recently to make the method more robust. This has been done by producing a hybrid scheme where it is combined with a flux-difference splitting scheme, that handles contact discontinuities better (Coquel & Liou, 1995).

Piecewise parabolic method

The Piecewise Parabolic Method, developed by Woodward & Collela (1984), is a higher-order extension of Godunov's method. Quadratic curves are used to interpolate

the dependent variables on the edges of the piecewise cells. Parabolic interpolation is used to obtain the left and right states q_l and q_r in the Riemann problem. The domain of dependence for the cell interface is determined and the piecewise monotone parabolic function within the domain of dependence is integrated. The left and right states at a cell interface are simply the averages of the integrated quantities over the domain of dependence. If the cells are equally spaced, then the scheme is fourth-order accurate, otherwise it is third-order in smooth regions of the flow.

Zeus Model

Stone & Norman (1993) implement their code using an adaptation of the piecewise parabolic method (PPM) of Woodward & Colella (1984). They do not use a Riemann solver to solve for the fluxes but rather a simple advection method. This was the basis for the method used by Downes (1997) for his thesis. Falle (2002) tested Zeus and found that it was only acceptable for use in gas dynamics if the linear artificial viscosity is multiplied by the smallest local Courant number since the shock errors are small in this case.

2.2.3 Alternative methods

Smooth particle Hydrodynamics (SPH)

SPH is a mesh-less Lagrangian method that uses a pseudo-particle interpolation method to compute smooth field variables. Each pseudo-particle has a mass, Lagrangian position, Lagrangian velocity and internal energy. Other quantities can be derived from interpolation or from constitutive relations. The method was originally developed independently by Lucy (1977) and by Gingold & Monaghan (1977) for astrophysics purposes. SPH is well established in many areas of astrophysics, from cosmic structure formation and galaxy formation to the dynamics of accretion disks.

The advantage of the mesh-less approach is that it does not suffer from the mesh distortion problems that limit Lagrangian approaches based on structured a mesh when simulating large deformations. It also allows for individual time-steps for each particle. Computationally, this can be give a 3-fold speedup over mesh based methods.

The main problem with SPH is that it gives poor results in regions of low density. Other limitations of SPH include the difficulties in modelling systems with extremely different characteristic lengths and the fact that boundary conditions usually require a more involved treatment than in finite volume schemes.

2.3 The Riemann problem

A conservation law together with piecewise constant data having a single discontinuity is known as the Riemann problem. In general, the Riemann problem consists of the conservation law $q_t + f(q)_x = 0$ together with the special initial data

$$q(x, 0) = \begin{cases} q_l & \text{if } x < 0 \\ q_r & \text{if } x > 0 \end{cases} \quad (2.25)$$

The form of the solution depends on the relationship between q_l and q_r .

Solving the Riemann problem can be viewed as finding a way of splitting up the jump from q_l to q_r into a series of jumps, each of which can propagate at an appropriate speed λ_i . Each λ_i is an eigenvalue of the $m \times m$ matrix A in the general equation $q_t + Aq_x = 0$.

The solution consists of elementary waves as illustrated in figure 2.1. An interesting property of the solution is that the flow variables are constant along straight lines in $x - t$ space, in other words, the solution is self-similar. In particular it is constant in time at the interface. As long as the solutions at each interface of each interval do not interact it is possible to compute the exact solution at the new time level from the initial piecewise constant data. It is then possible to compute the new cell values and restart the process.

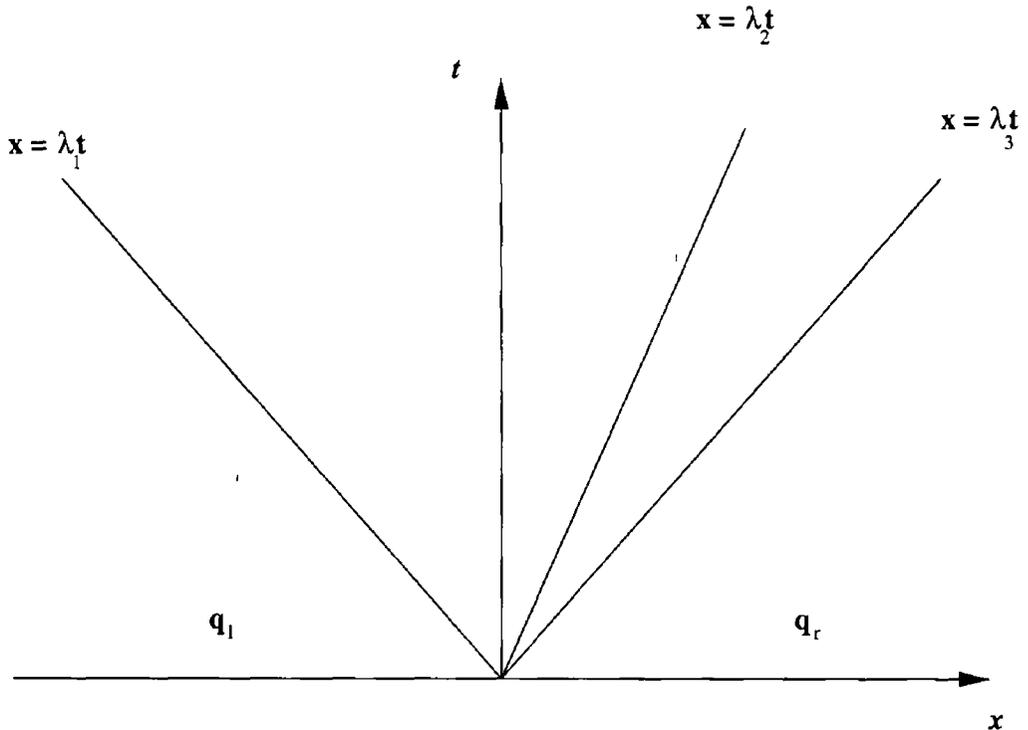


Figure 2.1 Solution to the linearized Riemann problem

2.4 Scheme Used

In this work we have used a method based on Van Leer's MUSCL scheme. It is a Godunov type method that is spatially and temporally second order accurate. We use the minmod method (equation 2.21) to limit the slopes of the piecewise linear approximations to prevent oscillations occurring in the presence of strong shocks. To resolve the fluxes at the cell boundaries a linear Riemann solver is used except where the resolved pressure differs from either the left or right state at the cell interface by more than 10%. In this case a non-linear Riemann solver is used. This strategy helps to optimise the code's efficiency while allowing for proper treatment of shocks and rarefactions in non-smooth regions of the flow.

An explicit technique is used to integrate the scheme on a 2-D cylindrically symmetric grid. We use an explicit scheme because the time-step must be kept small to track the changes in the physical variables. For such short time-steps an explicit

scheme is less computationally expensive than an implicit scheme. As a result the scheme must obey the Courant-Friedrichs-Lewy (CFL) condition to maintain numerical stability. That is, no signal can travel a distance of more than one cell width in one time-step.

2.4.1 Basic Equations

The equations to be solved for the hydrodynamics, ionisation and dissociation are

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{u}) \quad (2.26)$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} = -\nabla \cdot [\rho \mathbf{u} \mathbf{u} + p \mathbf{I}] \quad (2.27)$$

$$\frac{\partial e}{\partial t} = -\nabla \cdot [(e + p) \mathbf{u}] - L \quad (2.28)$$

$$\frac{\partial n_{\text{H}x}}{\partial t} = -\nabla \cdot [n_{\text{H}x} \mathbf{u}] + J(x, n_{\text{H}}, T) \quad (2.29)$$

$$\frac{\partial n_{\text{H}_2}}{\partial t} = -\nabla \cdot [n_{\text{H}_2} \mathbf{u}] - n_{\text{H}_2} n_{\text{H}} k(T) \quad (2.30)$$

$$\frac{\partial n_{\text{H}}}{\partial t} = -\nabla \cdot [n_{\text{H}} \mathbf{u}] + 2n_{\text{H}_2} n_{\text{H}} k(T) \quad (2.31)$$

$$\frac{\partial \rho \tau}{\partial t} = -\nabla \cdot (\rho \tau \mathbf{u}) \quad (2.32)$$

where ρ , \mathbf{u} , p , \mathbf{I} and e are the mass density, velocity, pressure, identity matrix and total energy density respectively. n_{H} is the number density of atomic hydrogen and n_{H_2} is the number density of molecular hydrogen. x is the ionisation fraction of atomic hydrogen, T is the temperature, $k(T)$ is the dissociation coefficient of molecular hydrogen, $J(x, n_{\text{H}}, T)$, is the rate of ionisation/recombination of atomic hydrogen and τ is a scalar that is used to track the jet gas. We also have the following definitions

$$p = nk_{\text{B}}T \quad (2.33)$$

$$e = \frac{1}{2} \rho \mathbf{u} \cdot \mathbf{u} + \frac{c_v}{k_{\text{B}}} p \quad (2.34)$$

$$L = L_{rad} + E_I J(x, n_H, T) + E_D k(T) \quad (2.35)$$

where c_v is the specific heat at constant volume, k_B is Boltzmann's constant, E_I is the ionisation energy of hydrogen and E_D is the dissociation energy of H_2 . L represents the energy loss and gain due to radiative and chemical processes. L_{rad} is the loss due to radiative transitions (see section 2.6 for more details). The second and third term in equation 2.35 represent the energy dumped into the ionisation of H and the energy dumped into dissociation of H_2 respectively. The dissociation coefficient $k(T)$ comes from Dove & Mandy (1986) and the ionisation rate, J , is that used by Falle & Raga (1995).

2.4.2 Numerical scheme

The conserved Euler equations are solved numerically in the following manner. The code has been designed to work both in slab symmetry and in cylindrical symmetry. As the results presented in this work relate to simulations run in cylindrical symmetry we shall outline the numerical scheme in relation to this geometry. The cylindrically symmetric simulations should give a better approximation to a full three-dimensional approach as slab symmetry does not stretch the flow as it moves away from the axis of the jet. The set-up for the solution to the equations in cylindrical symmetry is derived from the method in Falle (1991).

We have adopted a two-step approach to achieving second order accuracy in time. The first step is first order accurate in space and is used to allow for some variation in the state of a cell over the course of the time-step. The values in the cells after the first step are used to determine the fluxes for the second, spatially second-order accurate step.

In cylindrical symmetry, strict conservation is impossible since we have to work with the radial momentum and this is not a conserved quantity. However, by introducing a source term into the conservation equations to account for this, we can maintain conservation to accuracy of truncation error. We can write the Euler equa-

tions in the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial z} + \frac{1}{r} \frac{\partial r \mathbf{G}}{\partial r} = \mathbf{S} \quad (2.36)$$

where

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}$$

is the vector of conserved quantities u and v are the velocity components in the axial and radial directions respectively

$$\mathbf{F} = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ u(e + p) \end{pmatrix}$$

$$\mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ v(e + p) \end{pmatrix}$$

are the fluxes in the z and r directions e , the total energy per unit volume is given by

$$e = \frac{1}{2} \rho (u^2 + v^2) + \frac{c_v}{k_B} p \quad (2.37)$$

where c_v is the specific heat \mathbf{S} is the source term and is given by

$$\mathbf{S} = \begin{pmatrix} 0 \\ 0 \\ p/r \\ 0 \end{pmatrix}$$

In cylindrical symmetry, the volume of a cell (i, j) with sides from $(i - 1)\Delta h \leq z \leq i\Delta h$ and $(j - 1)\Delta h \leq r \leq j\Delta h$, where Δh is the spatial discretisation, is given by

$$\begin{aligned} V_{i,j} &= [\pi(j\Delta h)^2 - \pi((j - 1)\Delta h)^2] \Delta h \\ &= \pi(2j - 1)\Delta h^3 \end{aligned} \quad (2.38)$$

Given the solution at time n we can derive the solution at time $n + 1$ using the discretisation

$$\begin{aligned} Q_{i,j}^{n+1} &= Q_{i,j}^n - \frac{\Delta t}{\Delta h} \left[\left(F_{i+1/2,j}^{n+1/2} - F_{i-1/2,j}^{n+1/2} \right) \right. \\ &\quad \left. + \frac{2}{2j-1} \left(j G_{i,j+1/2}^{n+1/2} - (j-1) G_{i,j-1/2}^{n+1/2} \right) \right] + \Delta t S_{i,j}^{n+1/2} \end{aligned} \quad (2.39)$$

The mean value of Q in cell (i, j) at time n is derived from the integral

$$Q_{i,j}^n = \frac{2\pi}{V_{i,j}} \int_{(j-1)\Delta h}^{j\Delta h} \int_{(i-1)\Delta h}^{i\Delta h} Q(z, r, t_n) dz dr \quad (2.40)$$

F and G are the axial and radial fluxes respectively and are given by

$$F_{i+1/2,j}^{n+1/2} = \frac{2}{\Delta t (2j-1) \Delta h} \int_n^{n+1} \int_{(j-1)\Delta h}^{j\Delta h} F(i\Delta h, r, t) dr dt \quad (2.41)$$

$$G_{i,j+1/2}^{n+1/2} = \frac{1}{\Delta t \Delta h} \int_n^{n+1} \int_{(i-1)\Delta h}^{i\Delta h} G(z, j\Delta h, t) dz dt \quad (2.42)$$

where Δt represents the time discretisation and these are the fluxes averaged over time and the cell interfaces. The source term, averaged over time and the volume of the cell, is given by

$$S_{i,j}^{n+1/2} = \frac{2\pi}{\Delta t V_{i,j}} \int_n^{n+1} \int_{(j-1)\Delta h}^{j\Delta h} \int_{(i-1)\Delta h}^{i\Delta h} S(z, r, t) dz dr dt \quad (2.43)$$

The source term in a cell varies depending on the radial coordinate of the cell with respect to the axis of symmetry. This is due to the fact that the upper face of the cell, at $j + 1$, has a slightly larger surface than the lower face (j) this gives cells further from the axis a larger volume than those closer to the axis. We evaluate S by assuming a constant pressure and then integrating over the cell. This gives us

$$\begin{aligned} \left(\frac{p}{r} \right)_{i,j}^{n+1/2} &= \frac{2\pi}{\Delta t V_{i,j}} p_{i,j}^n \int_n^{n+1} \int_{(j-1)\Delta h}^{j\Delta h} \int_{(i-1)\Delta h}^{i\Delta h} dz dr dt \\ &= \frac{2p_{i,j}^n}{(2j-1)\Delta h} \end{aligned} \quad (2.44)$$

for the non-zero component of S

We can achieve second order accuracy in time by using the first order scheme to obtain an intermediate solution $Q_{i,j}^{n+1/2}$ at the half time $n + 1/2$. $Q_{i,j}^{n+1/2}$ is then used to determine the second order fluxes. To achieve second order accuracy in space we require knowledge of the gradients within each cell. Care has to be taken to ensure that we do not cause oscillations in the region of shocks so we apply a non-linear averaging function (equation 2.22) to the gradients which has the effect of taking the smallest of the two gradients on either side of the cell interface to limit the slopes (van Leer, 1977). Here, we use the primitive variables as they are more convenient to work with

$$P = \begin{pmatrix} \rho \\ u \\ v \\ p \\ n_H \\ n_{H_2} \\ x \\ \tau \end{pmatrix}$$

where ρ is the density, u and v are the axial and radial velocities respectively, p is the pressure, n_H and n_{H_2} are the number densities of atomic and molecular hydrogen, x is the ionisation fraction and τ is the tag variable for tracking the jet material. The gradients for these variables in a cell (i, j) are calculated from

$$\left(\frac{\partial P}{\partial z}\right)_{i,j}^{n+1/2} = av \left(\frac{P_{i,j}^{n+1/2} - P_{i-1,j}^{n+1/2}}{\Delta h}, \frac{P_{i+1,j}^{n+1/2} - P_{i,j}^{n+1/2}}{\Delta h} \right) \quad (2.45)$$

and for the fluxes in the radial direction the gradients are calculated as follows

$$\left(\frac{\partial P}{\partial r}\right)_{i,j}^{n+1/2} = av \left(\frac{P_{i,j}^{n+1/2} - P_{i,j-1}^{n+1/2}}{r_g(j) - r_g(j-1)}, \frac{P_{i,j+1}^{n+1/2} - P_{i,j}^{n+1/2}}{r_g(j+1) - r_g(j)} \right) \quad (2.46)$$

where $av(a, b)$ is the averaging function and $r_g(j)$ is the radial coordinate of the centre of gravity for a cell with radial index j . We determine this value using the following

equation

$$\begin{aligned}
 r_g(j) &= \frac{2}{3} \left[\frac{(j\Delta h)^3 - ((j-1)\Delta h)^3}{(j\Delta h)^2 - ((j-1)\Delta h)^2} \right] \\
 &= \frac{2}{3} \left[\frac{3j^2 - 3j + 1}{2j - 1} \right] \Delta h
 \end{aligned} \tag{2 47}$$

The *av* function to determine the gradients is the minmod method described in section 2 4 1 and given by equation 2 21 The value returned by the function is defined equation 2 22

Once we have determined the gradients in the cells either side of the interface we are calculating the fluxes at, we can set up the left and right states for the Riemann problem If we let $\mathbf{Q}^* = (\mathbf{Q}_l, \mathbf{Q}_r)$ be the resolved state, where \mathbf{Q}_l and \mathbf{Q}_r are the values of the state variables to the left and right of the cell interface, then the fluxes across the interface $(i + 1/2, j)$, for example, at time $n + 1/2$ are given by

$$\mathbf{F}_{i+1/2,j}^{n+1/2} = \mathbf{F} [\mathbf{Q}^*(\mathbf{Q}(\mathbf{P}_l), \mathbf{Q}(\mathbf{P}_r))] \tag{2 48}$$

where the left and right states are defined as

$$\mathbf{P}_l = \mathbf{P}_{i,j}^{n+1/2} + \left(\frac{1}{2} \right) \left(\frac{\partial \mathbf{P}}{\partial z} \right)_{i,j}^{n+1/2} \tag{2 49}$$

$$\mathbf{P}_r = \mathbf{P}_{i+1,j}^{n+1/2} - \left(\frac{1}{2} \right) \left(\frac{\partial \mathbf{P}}{\partial z} \right)_{i+1,j}^{n+1/2} \tag{2 50}$$

Similarly, when calculating the radial fluxes they are defined as

$$\mathbf{P}_l = \mathbf{P}_{i,j}^{n+1/2} + [j\Delta h - r_g(j)] \left(\frac{\partial \mathbf{P}}{\partial r} \right)_{i,j}^{n+1/2} \tag{2 51}$$

$$\mathbf{P}_r = \mathbf{P}_{i,j+1}^{n+1/2} + [j\Delta h - r_g(j + 1)] \left(\frac{\partial \mathbf{P}}{\partial r} \right)_{i,j+1}^{n+1/2} \tag{2 52}$$

We need to evaluate our source term for the second order time-step To do this we use the linear pressure distribution to evaluate the integral in equation 2 43 This

gives us

$$\left(\frac{p}{r}\right)_{i,j}^{n+1/2} = \frac{2}{(2j-1)\Delta h} \left(p_{i,j}^{n+1/2} + \left[\Delta h(j - \frac{1}{2}) - r_g(j) \right] \left(\frac{\partial p}{\partial r}\right)_{i,j}^{n+1/2} \right) \quad (2.53)$$

Once the second order fluxes have been calculated, we can update the cells, using equation 2.39, to have the values at time $n + 1$. Before we can restart the process, we must recalculate the time-step which must satisfy the Courant stability condition. This is done by evaluating the maximum wave speed in each cell. This is calculated from

$$C = \frac{\Delta t}{\Delta h} \left[\max \left(\left(\sqrt{\frac{\gamma p}{\rho}} + |u| \right), \left(\sqrt{\frac{\gamma p}{\rho}} + |v| \right) \right) \right] \quad (2.54)$$

where C is the Courant number and γ is the ratio of specific heats. If the new value for C is greater than the previous one then we shall use this new value in the following time-step. However, if this value exceeds 0.3 we shorten the time-step by 10% and the time-step is repeated. This is to prevent any information propagating more than 0.3 of a cell in one time-step. Similarly, if the value for C is found to be less than a certain fraction of 0.3 then we increase the length of the time-step for the purpose of computational efficiency.

2.4.3 Riemann Solver

The method we use to solve the Riemann problem at the cell interfaces is a hybrid scheme involving a linear solver to solve the fluxes in smooth regions and a non-linear solver in the presence of shocks and rarefactions. We obtain the solution to the Riemann problem using the primitive variables

$$P = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix}$$

The non-linear solver can be broken down into three cases, the case where there is a double rarefaction occurring at the interface, where there is a double shock

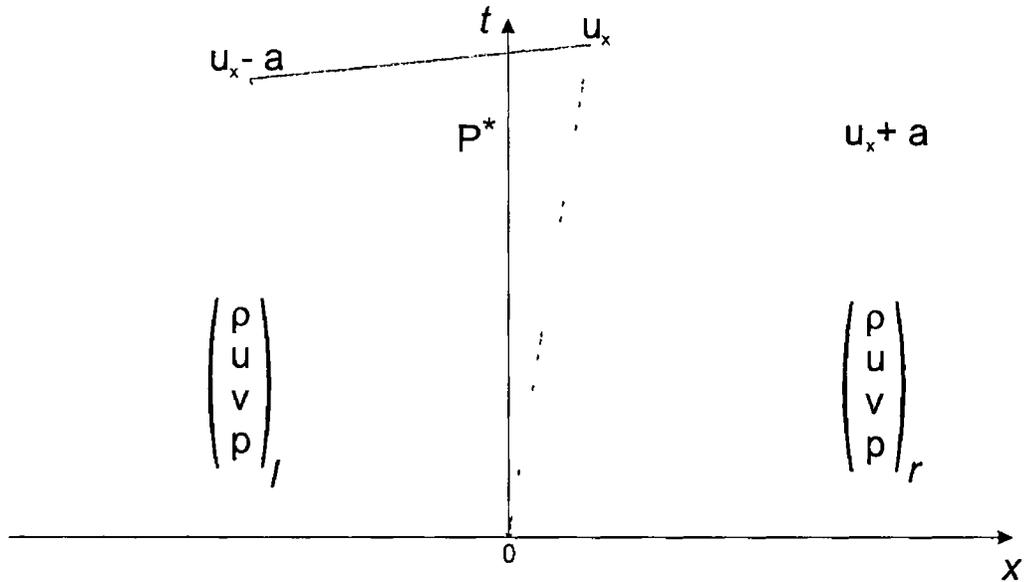


Figure 2 2 The Riemann problem at a cell interface. A linear or non-linear (used if the pressure in the left and right states differs by more than 10%) solver is used to determine the resolved state P^* at the cell interface.

and also where there is a shock/rarefaction occurring. This scheme affords us the computational efficiency of the linear solver while giving us robustness through the use of the non-linear solver. We use the non-linear solver if the pressure in the left and right states differs by more than 10%. The switching between the two non-linear cases is motivated by the behaviour of the resolved pressure from the linear solution. If the resolved pressure is less than that of both the left and right states, then we assume we have a double rarefaction, otherwise we have a shock/rarefaction or a double shock.

In all cases, we first determine the resolved pressure for the linear solver. This is given by

$$p^* = \frac{1}{C_l + C_r} [C_r p_l + C_l p_r + C_l C_r (u_l - u_r)] \quad (2.55)$$

$u_{l/r}$ are the velocities in the left and right states and C_l and C_r are given by

$$C_l = \sqrt{\gamma_l p_l \rho_l} \quad (2.56)$$

$$C_r = \sqrt{\gamma_r p_r \rho_r} \quad (2.57)$$

Here, $\gamma_{l/r}$ are the ratios of specific heat capacity in the cells on either side of the interface, $p_{l/r}$ are the pressures on either side of the interface and $\rho_{l/r}$ are the left and right densities. With this value for the resolved pressure we can determine whether or not we need to use the non-linear solver to handle shocks or rarefactions as outlined above.

In the event of a double rarefaction, we determine the new resolved pressure depending on whether the ratio of specific heats on either side of the interface is equal or not. In the case where the ratio is the same we determine the new resolved pressure from

$$p^* = \left[\frac{\frac{C_r}{\rho_r} + \frac{C_l}{\rho_l} - \frac{\gamma-1}{2}(u_r - u_l)}{\frac{C_l}{\rho_l p_l^z} + \frac{C_r}{\rho_r p_r^z}} \right] \quad (2.58)$$

where $z = \frac{\gamma-1}{2\gamma}$. In the alternative case, non-equal specific heat capacities, we use an iterative Newton-Raphson method to determine the value for the resolved pressure. Should this fail, we resort to a simple bisection method.

If there is a shock/rarefaction occurring at the cell interface an iterative method is used to determine the resolved pressure. p^* is given by

$$p^* = \frac{1}{W_l + W_r} [W_r p_l + W_l p_r + W_l W_r (u_l - u_r)] \quad (2.59)$$

where

$$W_l = -C_l \times a_l \quad (2.60)$$

$$W_r = C_r \times a_r \quad (2.61)$$

Here a_l and a_r are the wave speeds for connecting the states with pressures p^* and p_l , and p^* and p_r , respectively. We iterate, calculating refined values for the wave speeds with an updated value for the resolved pressure, until the value for the resolved pressure converges.

Once the pressure has been determined we calculate the resolved density and velocities. For the linear solver, the resolved density (ρ^*) and velocities (u^* and v^*) will be equal to values on either the left or right states depending on the direction of

the flow. We determine the direction from an initial estimate to the resolved velocity, u^* , given by

$$u^* = \frac{1}{W_r - W_l} (W_r u_r - W_l u_l - p_r + p_l) \quad (2.62)$$

For the non-linear case we determine whether the contact discontinuity is to the right or to the left of the cell interface by looking at the sign of the velocity. When we have the resolved pressure, density and velocities, we can determine the fluxes of the conserved variables, density, total energy, parallel and perpendicular momentum, and then update them accordingly.

2.5 Numerical Setup

The simulations presented in this work were run on a 32 processor, distributed memory machine at the Dublin Institute for Advanced Studies (DIAS). To facilitate this, the code was parallelised using MPI (Message Passing Interface) as the means for communication between the processors. For a detailed outline of the parallelisation method see Chapter 3.

2.5.1 Computational domain

The computational domain is initialised at the start of the simulation and with the aid of the parallel code we could prepare a grid large enough to take jets advanced to long-term stages of evolution. A typical computational domain consists of a 1.0×10^4 by 1000 grid divided among and stored in the memory of the 32 available processes. In general the ratio of the dimensions of the grid in the r and z directions is about 10:1. The computational domain is initialised to a set of ambient conditions with the jet input conditions initialised in a section of the left-hand boundary.

All of the simulations were of jets which are density matched with the ambient medium at a value of 100 cm^{-3} . The ambient temperature is set to 10^2 K and the jet temperature 10^3 K . The energy loss/gain function L is set to zero below 10^3 K as the

data used in the cooling functions is unreliable below this temperature and cooling is not dynamically significant here either. The ratio of the number density of molecular hydrogen to that of atomic hydrogen is set to be $n_{H_2}/n_H = 9$ within the jet and in the ambient medium. The gas is assumed to be one of solar abundances. The grid spacing was set to be 10^{14} cm. This value is used as it was found by Downes & Ray (1999) that the efficiency of momentum transfer is sensitive to grid spacing. They found that a value of 10^{14} cm was the minimum necessary to get reliable results.

For the purpose of determining the effect of different types of input velocity conditions we have experimented with three different jet velocity profiles. They are (1) “steady”, (2) “periodic” and (3) “pulsed/episodic”. We set the input jet velocity to have a time-averaged velocity of $v_0 = 215 \text{ kms}^{-1}$. The periodic jet has a sinusoidal variation superimposed upon this value with periods of 5, 10, 20 and 50 years with a total amplitude of v_1 . We have set the ratio $v_1/v_0 = 0.6$ for the periodic jet. For the steady jet the ratio of the $v_1/v_0 = 0$.

For the pulsed/episodic jet we have set $v_1/v_0 = 0$ but we have imposed a long-term pulsing of the jet with it switching on for several hundred years and then switching off for several hundred years. We have set the period for one complete cycle of this pulsing to be 800 years. Rather than switching the jet on and off instantaneously, we have used a ramped start-up and termination of each pulse. The value for the velocity varies linearly from 30 kms^{-1} to v_0 over the course of 50 years at the beginning of the pulse and inversely at the end of the pulse. This has been suggested (Lim et al., 2002), to give rise to increased molecular abundances at the head of the bow-shock, which in turn, affects the intensity-velocity relation. Some authors (Arce & Goodman, 2001) have suggested that observed outflows can have a very random, episodic, nature, giving rise to an outflow that varies in shape, mass-loss rate, direction and velocity in an unpredictable way. However, for the purpose of comparison with previous work, we use a more simplistic set-up of the episodicity.

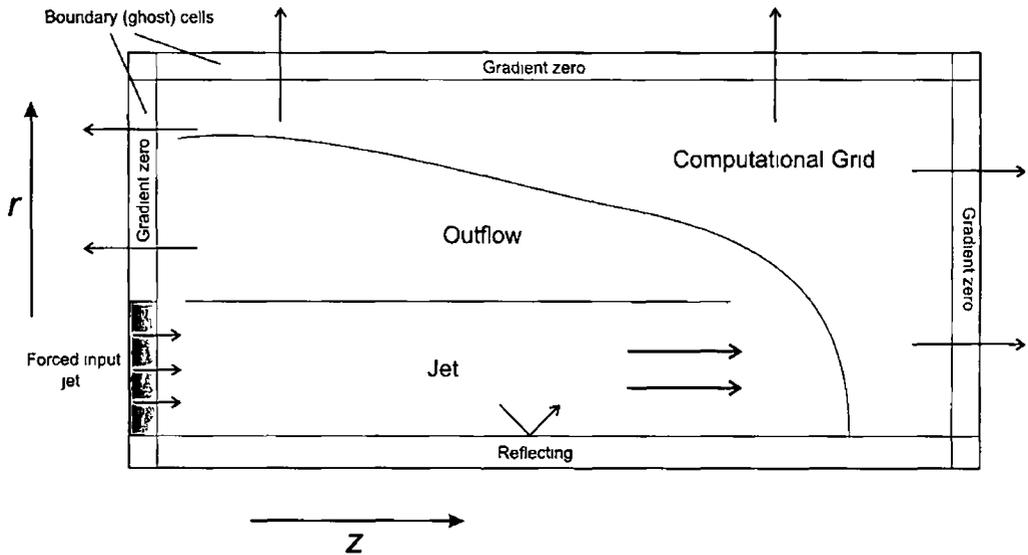


Figure 2 3 Set-up for the computational grid The computational grid is bounded by a double layer of cells The input jet is initialised in the bottom of the left-hand boundary This part of the boundary is set to be reflecting and so too is the bottom boundary All other boundaries are set to have gradient zero

2 5 2 Boundary Conditions

The boundary conditions are applied to a double layer of “ghost cells” set-up around the boundary of the computational grid We need a double layer of these cells due to the fact that we are using a 13-point stencil meaning that calculations in any cell on the computational grid involves the cell values at the two neighbouring cells in both the horizontal and vertical directions, along with the cell values from each of the cells diagonally offset from the current one Figure 3 5 in Chapter 3 illustrates the stencil The boundary conditions are set at the start of the half-step and the start of the full-step

Along the symmetry axis, $r = 0$, the boundary cells are set to be reflecting of the conditions in the two rows of cells along this axis within the computational grid The values at $\text{cell}(i,-1)$ and $\text{cell}(i,-2)$ will be set to the values in $\text{cell}(i,0)$ and $\text{cell}(i,1)$ respectively In addition, the sign of the radial velocities is changed in the cells along $j=-1$ and $j=-2$

For the axes $r = n$ and $z = m$ (considering a grid of size $m \times n$ cells), we set the values of the boundary cells to be equal to the values of the cells immediately beside them within the computational grid. This results in having a gradient of zero at these boundaries and allows material to flow freely on or off the grid. As an example, the right-hand boundary cells $\text{cell}(n, j)$ and $\text{cell}(n + 1, j)$ are set to match the cells $\text{cell}(n - 1, j)$.

The left-hand boundary is split into two parts. A section between $r = 0$ and $r = r_j$, where r_j is the jet radius, within which the input jet conditions are set-up in the boundary cells. These cells remain fixed at these values throughout the course of the simulation. Typically, a jet radius of 50 cells is used, giving a physical radius of 5×10^{15} cm. Between $r = r_j$ and $r = m$ we set the boundary cells to zero gradient in the two columns of cells at the left-hand edge of the computational grid. The values in $\text{cell}(-1, j)$ and $\text{cell}(-2, j)$ are set to $\text{cell}(0, j)$ and $\text{cell}(1, j)$ respectively, with the sign of the longitudinal velocities changed as is done with the radial velocities of the boundary cells along the axis of symmetry.

To prevent numerical problems occurring at the edge of the jet as it enters the computational domain we have inserted a shear layer of about 5 cells. The value for the density, pressure and velocity in these cells is tapered to meet the ambient conditions in the first cell of the ambient medium above them. We have experimented with what the effects of varying the width of the shear layer has on the resulting outflows.

2.6 Radiative Cooling & Emissions

2.6.1 Cooling

Due to the fact that we are dealing with a radiatively cooled jet, we must consider a number of timescales. These include the jet crossing time, the cooling time, the ionisation time and the dissociation time. The jet crossing time is long and is au-

tomatically resolved so long as the jet is wider than one grid cell and the courant number is less than 1. We ensure that the energy losses due to cooling are consistent with the dynamics occurring by choosing a time-step that ensures that the energy lost in a cell due to cooling is less than 30% of the internal energy of the cell. The ionisation and dissociation time-scales are quite short and the time-step is also restricted so that the dissociated molecular hydrogen in any one cell is limited to less than 90% of the total molecular hydrogen in that cell.

To ensure that the cooling and dissociation functions are calculated properly, the code performs sub-steps after the flux calculations have been completed in each dynamical time-step. Doing this reduces the error involved in allowing large amounts of energy loss and dissociation in any dynamical time-step. The lengths of these sub-steps are chosen such that the change in the internal energy in any one of the sub-steps is less than 5%, and the change in the molecular hydrogen number density is less than 10%. We assume that the fluxes across cell interfaces is zero for these sub-steps, which can only be valid if the courant number for the dynamical time-step is quite low. The value used here is $C \leq 0.3$.

The molecular cooling function used is from Lepp & Schull (1983). The radiative cooling rate, Λ_r , is fitted to a sum of terms representing vibrational and rotational cooling from both $\text{H} - \text{H}_2$ and $\text{H}_2 - \text{H}_2$ collisions. It is given by

$$\Lambda_r = n_{\text{H}_2} \left[\frac{L_{vh}}{1 + \frac{L_{vh}}{L_{vl}}} + \frac{L_{rh}}{1 + \frac{L_{rh}}{L_{rl}}} \right] \quad (2.63)$$

where the vibrational cooling coefficients at high and low density are

$$L_{vh} = (1.10 \times 10^{-13} \text{ ergs s}^{-1}) \exp(-6744K/T) \quad (2.64)$$

$$L_{vl} = [n_{\text{H}} k_{\text{H}}(0, 1) + n_{\text{H}_2} k_{\text{H}_2}(0, 1)] \times (8.18 \times 10^{-13} \text{ ergs}) \quad (2.65)$$

where the terms $k_{\text{H}}(0, 1)$ and $k_{\text{H}_2}(0, 1)$ are the $v = 0 \rightarrow 1$ collisional excitation rate for transitions involving H and H_2 respectively. Since cooling due to $\text{H}_2 - \text{H}_2$ interactions

are only important for low temperatures (Lepp & Shull, 1983), we ignore the $k_{\text{H}_2}(0, 1)$ term $k_{\text{H}}(0, 1)$ is defined as

$$k_{\text{H}}(0, 1) = \begin{cases} (1.0 \times 10^{-12})T^{1/2} \exp\left(\frac{-1000K}{T}\right) & \text{if } T > 1635K \\ (1.4 \times 10^{-13}) \exp\left[\frac{T}{125K} - \left(\frac{T}{577K}\right)^2\right] & \text{if } T \leq 1635K \end{cases} \quad (2.66)$$

The rotational cooling rate coefficients at high and low density are

$$L_{rh} = 3.90 \times 10^{-19} \exp\left(\frac{-6118}{T}\right) \quad (2.67)$$

$$L_{rl} = Q(n) 1.38 \times 10^{-22} \exp\left(\frac{-9243}{T}\right) \quad (2.68)$$

where

$$Q(n) = n(\text{H}_2)^{0.77} + 1.2n(\text{H})^{0.77} \quad (2.69)$$

is a function that bridges the gap between the low density ($\Lambda \propto n^2$) and high density ($\Lambda \propto n$) limits of the cooling. This function is applicable in the temperature range $100K < T < 1 \times 10^5 K$. If $T > 10^5 K$, we set the molecular cooling to zero as the number densities of H_2 will be very low at these high temperatures due to collisional dissociation.

2.6.2 Line Emission

The intensity of emission from transitions of observational interest were calculated in the same way as by Downes (1997). These include forbidden (SII) and molecular lines (H_2), and permitted and semi-forbidden lines (several lines of hydrogen, helium, nitrogen and sulphur). Forbidden line emissions were calculated assuming a Boltzmann distribution of populations of the electronic levels. The equation for the energy emitted through each forbidden atomic line is

$$L = n \Delta E_{ij} \exp\left(\frac{-E_i}{kT}\right) \frac{A_{ij}}{\Omega_n} \quad (2.70)$$

where ΔE_{ij} is the energy of the transition and E_i is the energy of the upper level in the transition. n is the number density of the particular ion. The number densities

of all ions, except for H^+ , were calculated assuming ionisation equilibrium and that each ion has the same flow velocity as the rest of the fluid. T is the temperature, Ω_n is the partition function for the ion, which relates the microscopic properties of the atoms and molecules to the bulk thermodynamic properties of the gas, and A_{ij} is the probability per second of a transition from level i to level j . This equation was also used to calculate the emissions for molecular hydrogen.

The emissions from permitted and semi-forbidden lines were calculated assuming transitions occurred exclusively by collisional excitation followed immediately by radiative de-excitation (Dopita (1977)). The rate of the process is given by

$$R_{ij} = 8.63 \times 10^{-6} n_e \frac{\Theta_{ij}}{\omega_i} T^{-1/2} \exp\left(\frac{-(E_j - E_i)}{kT}\right) \quad (2.71)$$

where the transition is from level i to level j and Θ_{ij} is the collisional strength of the transition, ω_i is the statistical weight of the level i , and n_e is the electron number density.

2.7 Methods for dealing with numerical problems

There are several numerical problems that can occur in the simulation of supersonic flows that involve cooling effects, particularly in the presence of shocks and rarefactions. Dealing with them can only be done through trial and error and the use of ad hoc methods. Here, we outline the ones we are most concerned with and the way we have dealt with them.

2.7.1 Artificial Viscosity

To prevent the undesirable effects of oscillations appearing in the solution in the presence of a strong gradient, we have explicitly included some artificial viscosity in the flux calculations. The viscosity term is given by

$$\nu_i = \frac{l^2 \rho \Delta u^2}{\Delta x^2} \quad (2.72)$$

where Δu is the change in the velocity across the cell interface, ρ is the resolved density, Δx is the cell spacing and l is some coefficient. The use of artificial viscosity is an inexact science in computational fluid dynamics but there is general agreement that the judicious use of this quantity can lead to reasonable and accurate solutions. For l , we have found that a value of between 0.15 and 0.2 is the minimum needed to give a smooth and stable solution.

The value for ν_i is then added to the momentum flux perpendicular to the cell interface. The energy flux equation is also updated accordingly to ensure that we do not affect its conservation.

2.7.2 Shear Viscosity

In the simulation of supersonic jets there are many regions where strong shear layers can occur. The most notable is that of the boundary between the jet itself and the inside of the outflow cocoon. Here, the high velocity, high density jet is surrounded by a slow moving, low density region. Over the course of a long simulation, the lack of any sort of momentum transfer between these two regions can give rise to problems. This is a numerical problem as the jet material and the cocoon material usually lie in cells adjacent to each other without any tapering between them. In reality, slow moving material residing in the shear layer between the two regions is accelerated by the jet and results in a net transfer of momentum across the layer.

To cope with this problem we have introduced a shear viscosity term to the flux equations to try and aid the transfer of momentum across the shear layer. To calculate the shear viscosity we need to determine the values for $\partial u/\partial y$, the rate of change of the velocity perpendicular to the cell interface with respect to the parallel direction, and $\partial v/\partial x$, the rate of change of the velocity parallel to the cell interface in the perpendicular direction. Figure 3.6 in Chapter 3 illustrates what is required. If we

consider the interface at $(i - 1/2, j)$, we can write $\partial u/\partial y$ as

$$\frac{\partial u}{\partial y} = \frac{(u_{i,j+1} + u_{i-1,j+1})/2 - (u_{i,j-1} + u_{i-1,j-1})/2}{2} \quad (2.73)$$

where the $u_{i,j\pm 1}$ and $u_{i-1,j\pm 1}$ are the velocities perpendicular to the cell interfaces above and below the interface $(i - 1/2, j)$. The value for $\partial v/\partial x$ for any interface is simply given by the difference between the v_r and v_l , the velocities parallel to the interface in the left and right states. That is

$$\frac{\partial v}{\partial x} = v_r - v_l \quad (2.74)$$

We then need to incorporate these quantities into our flux calculations. For the momentum flux parallel to the interface we add in a shear velocity term that will allow for the transfer of some momentum across the interface even in the case of a strong shear layer. The new flux becomes

$$\mathbf{F}_\perp = \rho uv - \rho \nu_s \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad (2.75)$$

where ν_s is some coefficient that we have determined by trial and error to give enough transfer of momentum to give a more physical behaviour at shear layers without adversely affecting the overall solution. We found that a value of 0.01 is enough to give an appropriate transfer.

2.7.3 Dealing with Negative Pressures

It is a common problem of conservative schemes, that negative pressures can be computed in the regions of the flow where the internal energy is very small compared to the kinetic energy. The internal energy and pressure are calculated from the total energy after the kinetic energy has been subtracted out. Typical regions where this can occur are low density, high speed flows. Small inaccuracies in the conserved quantities can lead to the value of the kinetic energy exceeding that of the total energy which results in a negative pressure being calculated.

In the event of a negative pressure being calculated in a cell, one possible solution to this problem is to simply set the value of the pressure in that cell back to the value it had at the beginning of the half-step. However, we have found that using this method, in most cases, merely results in the negative pressure being calculated again in the following time-step and over the course of long simulation can result in a significant amount of extra energy being added. A less costly method is to assign the value for the pressure in the cell with the negative pressure to the maximum value for the pressure at the beginning of the half-step in each of its four neighbouring cells. For example, if a negative pressure is encountered in cell (i, j) then we reassign the value for the pressure in that cell according to

$$p_{i,j}^n = \max\left(p_{i\pm 1,j}^{n-1/2}, p_{i,j\pm 1}^{n-1/2}\right) \quad (2.76)$$

where $p_{i,j}$ is the pressure in cell (i, j) . We have found that using this method helps prevent the re-occurrence of the negative pressure in that cell in subsequent time-steps.

Chapter 3

Numerical Method: Code

Parallelisation

Enhanced computer application performance is the only practical purpose of parallel processing. In the field of Computational Fluid Dynamics (CFD), harnessing the aggregate capabilities of multiple processors can allow an application to run over much larger domains, for much longer time-scales, and to greater resolutions. The key to successfully adapting a CFD code to run in a parallel environment is to ensure that all of the processors are used efficiently by balancing the computational load among them.

In this chapter we will give a detailed account of how the parallelisation of the code described in Chapter 2 was implemented. The parallelisation strategy is based on a domain decomposition approach and is implemented using the Message Passing Interface (MPI) software which is the industry standard for distributed memory parallel programming. We will go through the steps that were taken in evolving the parallelisation scheme to an effective load-balanced method.

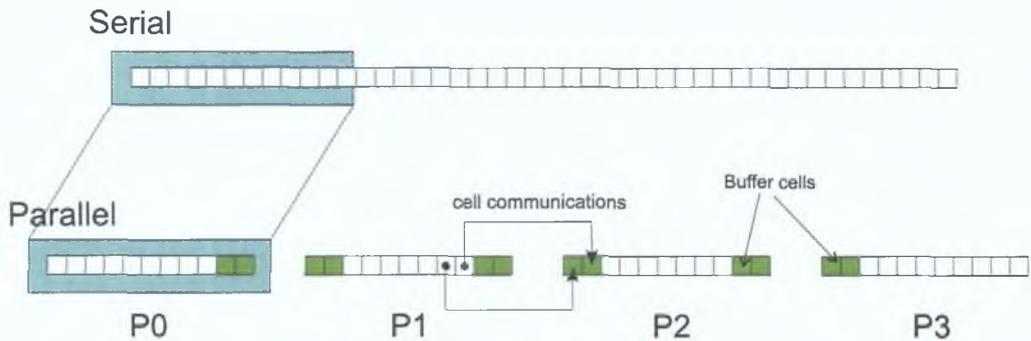


Figure 3.1: Splitting of a one dimensional grid over several processors in a one dimensional simulation. The two edge cells on the boundaries of each processor's domain need to be passed to the neighbouring processor. The values from these cells are stored in buffer cells on the edge of the neighbouring processor's domain.

3.1 Simple Splitting

3.1.1 Splitting of the computational grid

For the sake of simplicity and experimentation, initially, it was chosen to develop the parallelisation method on a 1-dimensional code. The 1-D code consisted of a single line of cells across which a shock wave would propagate. The cells were divided among the available processors as shown in figure 3.1. Each processor takes an equal share of the total number of cells and communications take place between neighbouring processors in order to maintain the continuity of the outflow simulation as it crosses the boundaries between processor's sections of the computational grid. Some global communications are also necessary to maintain synchronisation in such variables as the Courant number, for example.

For the flux calculations in any particular cell, information from the four neighbouring cells is required (two in each direction). When a processor is calculating the fluxes in a cell at the edge of its domain it requires information from cells on the neighbouring processor to complete the calculations. For example, when processor P_n , with m cells in its domain, reaches the point where it needs to calculate the fluxes across the cell interfaces of cells $m - 2$ and $m - 1$ in time-step t , it will require the

cell values in the first two cells of processor P_{n+1} . To achieve this, these cell values are packed into a message at the beginning of time-step t by processor P_{n+1} and sent to processor P_n . Processor P_n receives the message and unpacks the two sets of cell values into its buffer cells which are denoted by m and $m+1$. When the unpacking is complete, it is possible to proceed with time-step t and the fluxes in cells $m-2$ and $m-1$ can be calculated correctly. This allows the simulation to maintain its continuity across processor boundaries. Figure 3.1 illustrates how such a communication would take place.

3.1.2 Splitting a 2-dimensional grid

With a working model for the 1-dimensional code it was then possible to develop a method for the 2-dimensional code. In this case the domain is initialised as a rectangular grid of cells, of the correct dimensions and size sufficient to contain the fully-formed jet and outflow at the end of the simulation. This can vary depending on the type of input conditions used (adiabatic jet, cooled jet, wide or narrow opening angle, etc.). It is also constrained by the total amount of memory available, i.e. the sum of the memory in all of the machines to be used in the simulations. Usually the grid will have its greatest dimension in the direction of the jet input velocity as, ultimately, the outflow will be collimated in this direction. A typical grid size will be of the order 12000×1000 cells for a simulation that will last for a few thousand years. Three copies are required for the updating of the grid cells that occurs during the time-steps. Each cell requires 16 “doubles” in memory. Each double is 8 bytes in size so that gives a total memory requirement of ~ 4.5 Gigabytes in total. At the time of this work, the only way to fulfil this memory requirement is to combine the memory of several machines in a cluster. Figure 3.3 illustrates how the outflow might look when it reaches an evolved state.

The computational grid is split into blocks of cells, each processor holding one of these blocks in its local memory. To split the grid for parallelisation, there are

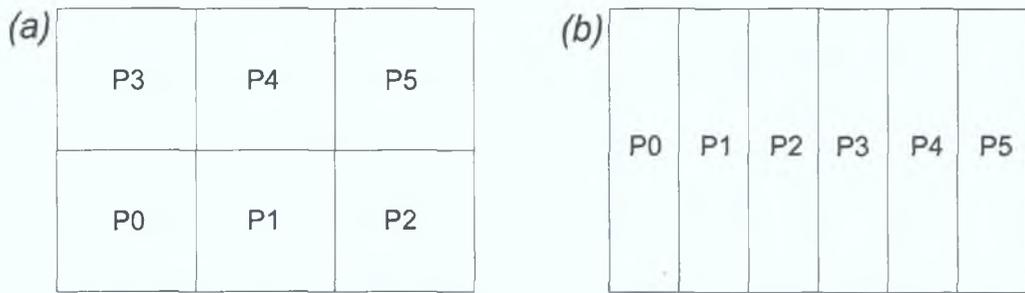


Figure 3.2: Splitting strategies for a 2-dimensional grid. The closer to squares the sub-domains are, the less significant is the time taken to do the communications between processors. The left-hand figure is the best choice in this respect.

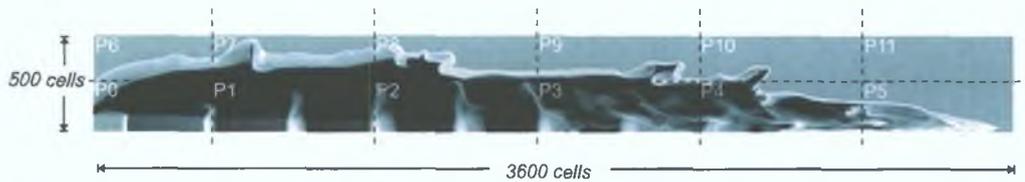


Figure 3.3: An example of an outflow simulation in an evolved state. This is a log-scale plot of the density of a radiatively cooled jet. The dashed lines represent inter-processor boundaries and how the computational grid would be typically split among the processors.

two possible approaches: 1-dimensional and 2-dimensional. We seek to minimise the amount of communicating necessary between the processors as this will reduce the length of time it takes to run a simulation. The communication overhead is minimised where the area to perimeter ratio of a block is maximized, i.e. the larger the volume a block of cells has relative to the total length of its boundaries the better. This essentially means that the closer the data blocks are to perfect squares the smaller the communication overhead will be. Figure 3.2 shows two possible splitting strategies. If we take the case where we have a grid of size 1000×400 cells with 10 available processors then the optimum split will be 5×2 processors. In this case each processor will have 200×200 cells.

Decomposition Routine

At the start of a simulation, enough memory for the maximum desired size of the outflow must be allocated. This is dependent mainly on how much memory is available

in total across all the processors. The desire is to run the simulation for as long as is feasibly possible. The overall grid is then split among the available processors. In the parallelised code there is a function that performs the domain decomposition at the beginning of the simulation. It tries to ensure that the dimensions of each processor's local grid are as close as possible to being square. However, in the simulation of jet-driven molecular outflows, due to their collimated nature, it is more suitable to split the domain into sections that are more akin to long rectangles. How we split up the computational grid also depends on the number of processors that are to be used for the simulation. Figure 3.3 shows how a fully evolved outflow might look and how its computational grid might be split among 12 processors.

The basic algorithm for decomposing the grid is as follows. The goal is to get each sub-domain to be as close as possible to a square. The algorithm will work for all even numbers of processors as well as for a single processor. It will not work for odd numbers of processors other than 1. We first calculate the 'split factor' (Cahill, 2001) w as follows

$$w = \frac{N_x}{N_x + N_y} \quad (3.1)$$

where N_x is the total number of cells allocated in the x-direction and N_y is the total number of cells allocated in the y-direction. Then n_x and n_y , the number of processors to be allocated in the x and y directions respectively, is calculated in the following manner

$$n_x = \lfloor (nproc^w) + 0.5 \rfloor \quad (3.2)$$

$$n_y = \left\lfloor \frac{nproc}{n_x} \right\rfloor \quad (3.3)$$

Here, $nproc$ is the total number of processors allocated for the simulation. If the number of processors in the x-direction multiplied by the number in the y-direction doesn't equate to the total number of processors available, then processors are added

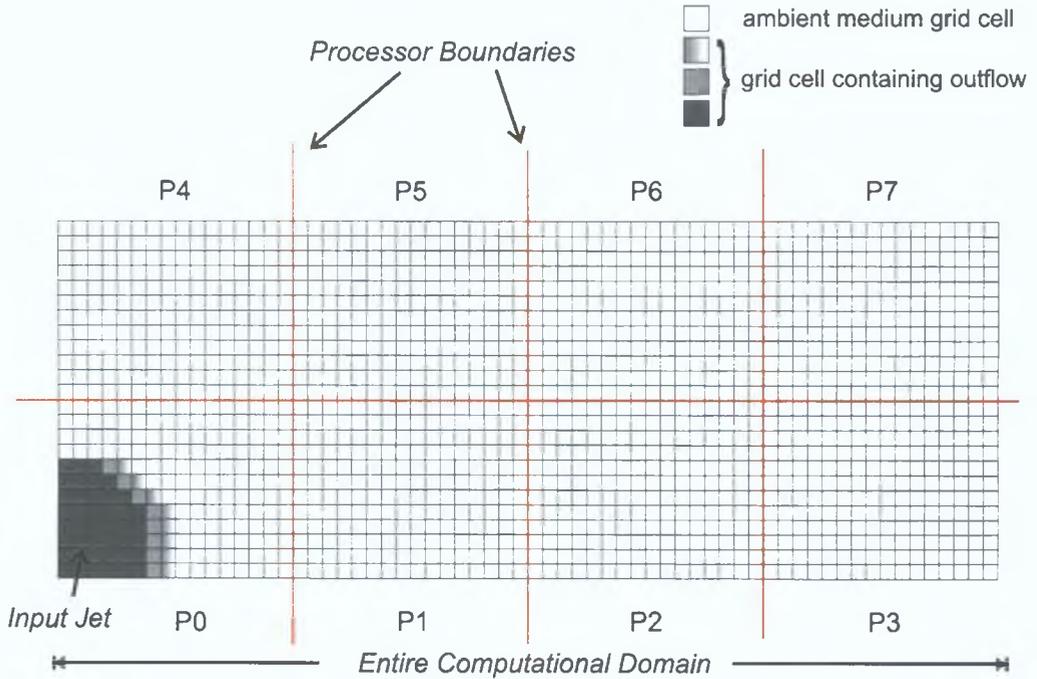


Figure 3.4: Simple splitting of the computational grid. The jet enters the grid from the left-hand boundary and propagates across the decomposed grid crossing processor boundaries seamlessly. Inter-processor communications take place at the processor boundaries during each time-step to ensure that each processor has all the information it needs to perform the flux calculations in all of the cells in its section of the computational grid.

onto or subtracted from the number in the x-direction according to the following criteria:

$$n_x = \begin{cases} n_x + 1 & \text{if } n_x < n_y \\ n_x - 1 & \text{otherwise} \end{cases}$$

and n_y is recalculated. Both n_x and n_y are set to 1 in the case where only one processor is being used.

An example of how a simple splitting of the grid might look is shown in figure 3.4. The initialised jet is fed into the computational domain of the first processor on the left and it then propagates in the direction of the input velocity vector, crossing processor boundaries seamlessly.

3.1.3 Communications Local & Global

To maintain the continuity of the simulation across the boundaries between each processor's section of the computational domain we need to perform communications between the processors. The code we are using is a time-stepping code and the calculations of the fluxes across the boundaries of each grid cell at each time-step require information from several of the neighbouring cells. As a result, we must communicate cell values between processors during each time-step. At the processor boundaries there are a set of buffer cells into which are copied the cell values for the cells at the edge of the neighbouring processor. Figure 3.5 illustrates this. These communications are "local" communications as they only involve communications between two adjacent processors. There is also a requirement for "global" communications between all of the processors at the end of each time-step in order to synchronise certain parameters such as the Courant number.

Local Communications

Recalling from section 2.4.2 in Chapter 2, the state vector, \mathbf{Q} , in each cell is updated during each half-step according to equation 3.4

$$\mathbf{Q}_{i,j}^{n+1} = \mathbf{Q}_{i,j}^n - \frac{\Delta t}{\Delta h} \left[\left(\mathbf{F}_{i+1/2,j}^{n+1/2} - \mathbf{F}_{i-1/2,j}^{n+1/2} \right) + \frac{2}{2j-1} \left(j \mathbf{G}_{i,j+1/2}^{n+1/2} - (j-1) \mathbf{G}_{i,j-1/2}^{n+1/2} \right) \right] + \Delta t \mathbf{S}_{i,j}^{n+1/2} \quad (3.4)$$

where \mathbf{F} and \mathbf{G} denote the fluxes in the axial and radial directions respectively and \mathbf{S} is the source term. The resulting state depends on the fluxes of the various parameters across each of the cell faces. For the second order scheme we have used, we need the information from the two neighbouring cells in each direction (up, down, left and right) to calculate the gradients of the primitive variables. For example, if we want to calculate the flux of a variable U across the right-hand face of a cell with index i, j

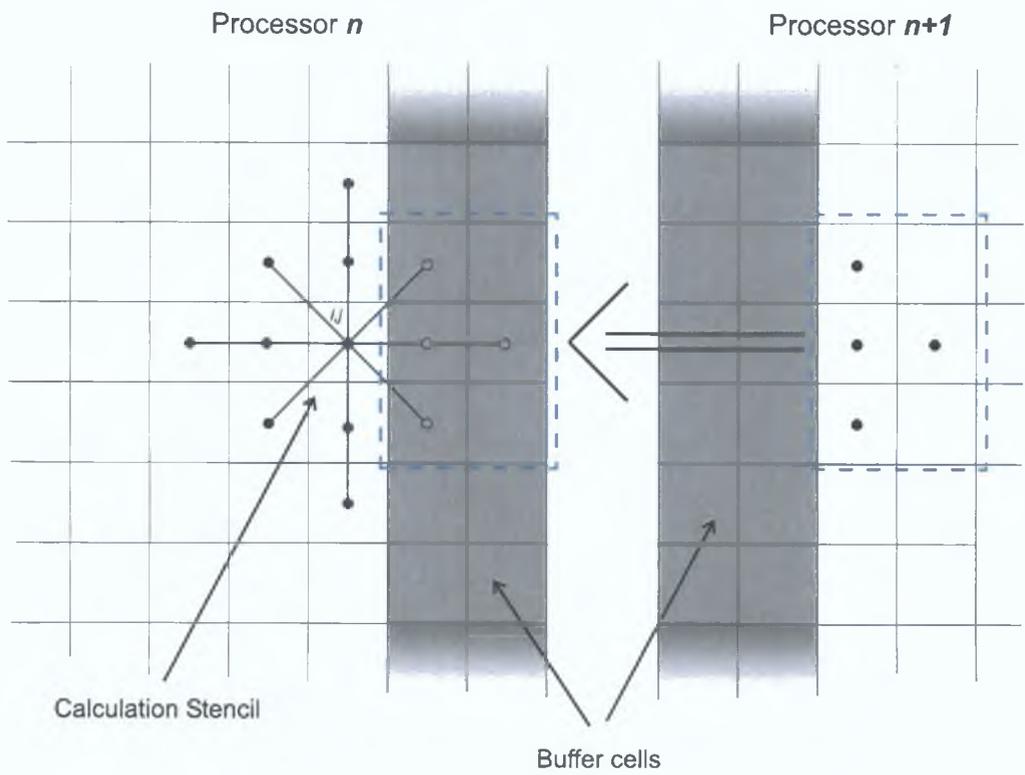


Figure 3.5: Calculations at each cell involve 12 neighbouring cells. This can result in a calculation at a particular cell close to the boundary of a processes domain requiring cell information from a neighbouring process.

we would use the following equations to calculate the gradient of U to the left and right of the interface

$$g_l = av(U_{i+1,j} - U_{i,j}, U_{i,j} - U_{i-1,j}) \quad (3.5)$$

for the gradient of U on the left of the interface, and

$$g_r = av(U_{i+2,j} - U_{i+1,j}, U_{i+1,j} - U_{i,j}) \quad (3.6)$$

for the right of the interface (note that for the sake of simplicity we have ignored any terms related to cylindrical symmetry). The av function is defined as in equation 2.22 in Chapter 2.

To calculate equations 3.5 and 3.6 for all sides of a cell, it is clear that we require information from the two adjacent cells in each direction. Additionally, to work out the shear viscosity (section 3.6 in Chapter 2), information from the four cells diagonally offset from the cell i, j is also required. This is to work out the velocity gradient parallel to the cell interfaces (see figure 3.6). In all, information from twelve neighbouring cells is required. When one or more of these cells resides on one or more neighbouring processors there is a need for a communication of cells between the processors (figure 3.5).

Global Communications

At the end of every time-step we need to recalculate the value for the time-step to account for any changing conditions arising on the computational grid. To do this we first calculate three variables on each processor's grid: the Courant number, which is the maximum wave speed on the grid in terms of cell spacing, Δ , a measure of the maximum cooling taking place in any one cell and 'discheck', a measure of the maximum amount of dissociation taking place in any one cell. If any of these values falls outside a range of allowed values, we then increase or decrease the time-step value accordingly. Before we can recalculate the time-step we figure out the maximum

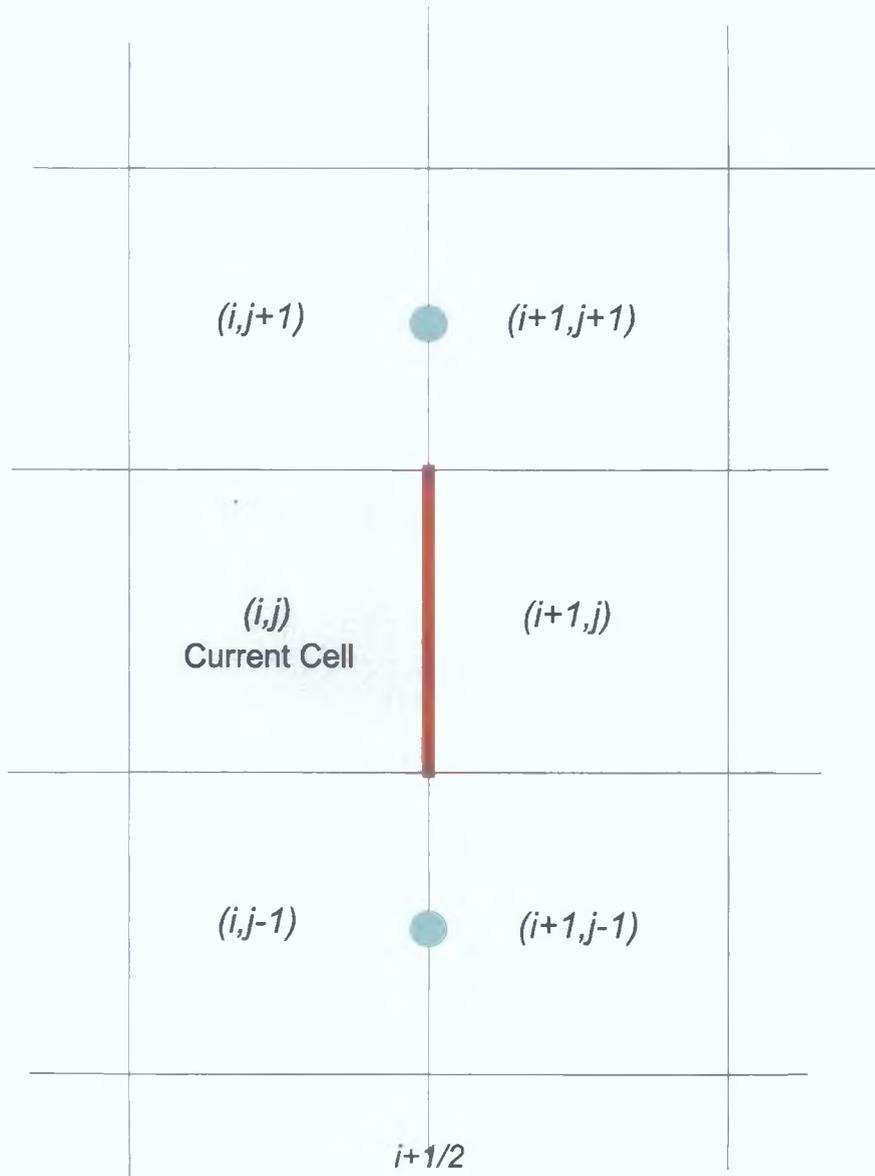


Figure 3.6: Calculation of the shear viscosity requires information from the cells diagonally offset from the current cell. Here, we wish to calculate the shear viscosity at the cell interface $i + 1/2, j$ (highlighted in red). To do this we need to know the velocity gradient parallel to this interface. This, in turn, requires that we know the velocities at the points $i + 1/2, j \pm 1/2$ (blue dots). The velocities at these points are taken as the average of the velocities in the cells either side of the interface, in this case, cells $i, j \pm 1$ and $i + 1, j \pm 1$.

values for each of these variables across the entire computational grid. We do this by using the MPI function *MPI_Allreduce* which takes a mathematical operation as one of its arguments. In this case we tell it to get the maximum value for the three variables and relay this value to all of the processors.

3.2 Redistributive Splitting & Load Balancing

3.2.1 Method for Load-balancing

With the simple splitting of the grid among the available processors it was possible to achieve a speed-up over the serial code by virtue of the fact that the computational grid and the work-load was being shared between several processors. It also meant that larger simulations could be run using the combined memory capacity of several computers. However, due to the nature of the simulations, the jet starts off as a small input to the left-hand boundary and steadily increases in size, it is obvious that there is a lot of wastage of the computational resources on areas of the grid within which there is nothing happening. Also, because of the more complex physical conditions arising within the part of the computational grid occupied by the jet and outflow, processors holding this part of the grid work harder than those calculating over the ambient medium. This results in an imbalance in the computational load across all of the processors and reduces the efficiency of the simulation. The desired method of parallelisation should concentrate all the available processing power in the area of the grid that contains the outflow. As the outflow gets bigger, this processing area enlarges to accommodate it. To achieve this, the portion of the overall grid containing the outflow must be split among the processors. We will call this portion of the grid the “active grid”. Figure 3.7 illustrates this. As the outflow increases in size, so too does the active grid. This results in each processor’s section of the active grid increasing in size. As each processor adds new cells to its grid, it must ensure that these new cells are filled with the correct parts of the outflow from the neighbouring

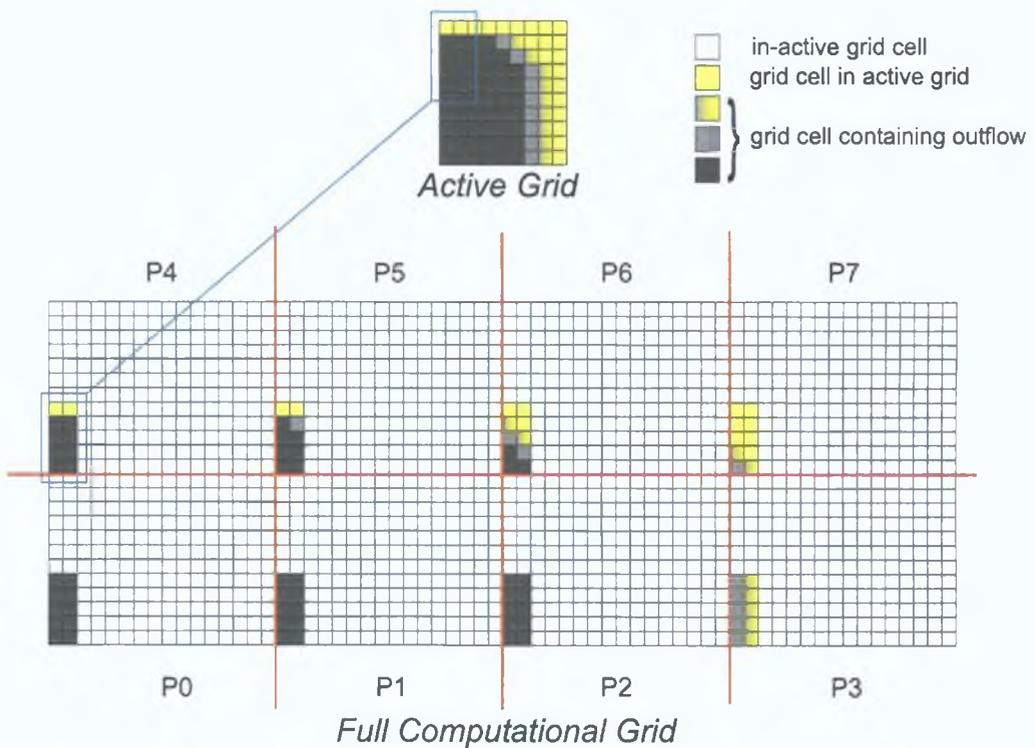


Figure 3.7: The redistributive splitting parallelisation method. In this method each processor takes a section of the “active” grid, i.e. that portion of the full computational grid in which the outflow resides. This concentrates the computational power only in areas where it is needed and allows us to run simulations in shorter lengths of time.

processors. For example, the left-most processor in figure 3.7, ‘P0’, must copy the values from the cells on the left-hand edge of P1 into its new cells when it expands its active grid in the x-direction. Each time it becomes necessary to increase the size of the active grid a redistribution of cells occurs. We call this method for parallelising the code the “redistributive splitting” method.

3.2.2 Resizing the grid and redistributing cells

At set intervals, the code checks whether resizing of the active grid to accommodate the increased size of the outflow was necessary. This interval is usually about 10 to 50 time-steps. At these set intervals, an iterative loop passes over the active grid of all of the processors holding parts of the overall active area along the top and right-hand

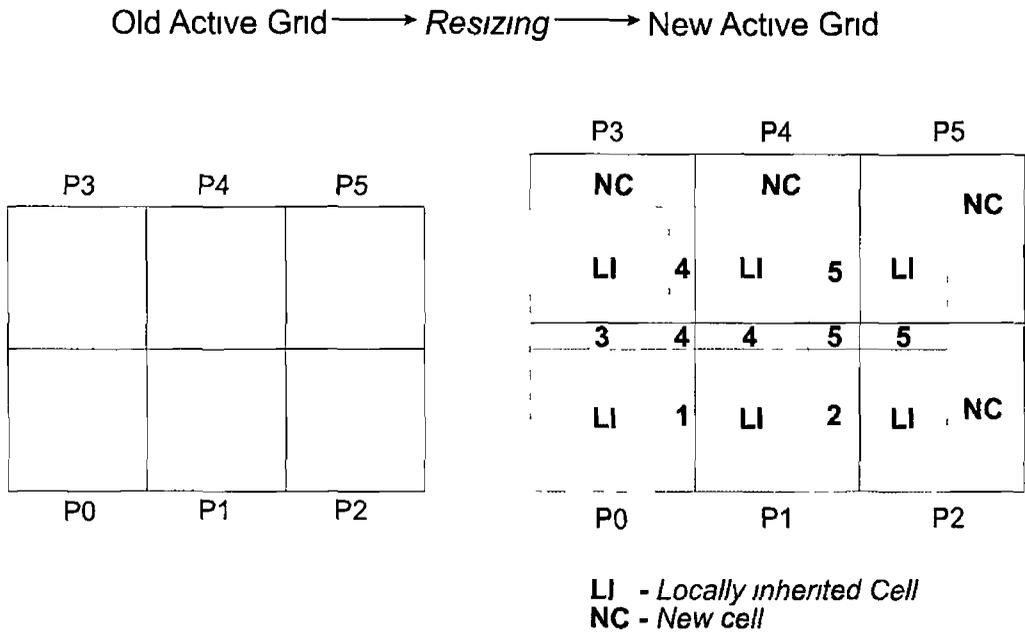


Figure 3 8 When a redistribution occurs all the cells in the old active mesh/grid are copied into the new active mesh/grid. Cells are assigned one of three types of values depending on their origin/destination.

boundaries to determine the distance of the edge of the outflow from the current edge of the overall active grid. If the edge of the outflow is within a certain number of cells of either of these boundaries, the processors will add a new buffer of cells to that boundary. In parallel terms this means that each processor's partition will increase by a set amount. When this occurs, the code performs the communications necessary for redistributing the cells.

There are several steps to be done each time there is a resizing of the active grid. They involve the basic redistribution of the grid cells and also checking the boundaries to make sure the new cells in the active grid containing the ambient medium, have the same ambient physical conditions as the ambient cells already in the active grid.

Redistribution Routine

The main aim of the redistribution algorithm is to identify where each cell is in the current active grid and then to determine where each cell must go in the new, larger,

active grid. It then performs the necessary communications between processors.

When a redistribution call is made, each processor must first scan over all of its cells in the old active grid to determine where each cell will be placed in the new active grid. Secondly, each process must then scan over the cells in its portion of the new active grid and determine where, and from which processor, it will receive new cells. Figure 3.8 illustrates this process. For both of these steps, the scan over the old active grid will designate each of the cells to be one of the following three options depending on where it will go in the new active grid:

- *NEW CELL* if the cell lies outside the current overall active grid
- *LOCAL INHERIT* if the cell is going to remain in the same processor domain
- *P* if the cell will go or come from another processor, '*P*' being the rank of that processor

Once these steps have been taken each processor has all the information it needs for its communications, i.e. where to receive cells from and where to send cells to, and the sizes in terms of cell numbers of each of these communications. Each block of cells to be communicated is packed into a contiguous block of memory. This is necessary because the MPI communication functions require that their "payload" be read from a contiguous piece of memory. The communications are then set up and when all processors have completed unpacking the new cells into their new active grids the simulation of the outflow can continue within the overall new active grid.

Figure 3.9 shows a flow chart for the code utilising the redistributive parallel method. Any boxes containing text highlighted in red italics, represents steps that have been added to the underlying serial code in order to parallelise it. The blue box encompasses the program core where the cell interface fluxes are calculated and is the most time-consuming part of the code. As such, this is the part of the code where most of the parallelisation effort is concentrated. When a resizing of the active grid is required, the main code will hand over the current active grid along with the

new active grid dimensions to the redistribution routine (red box in figure 3.9), which pieces together the new active grid from communicated cells, any locally inherited cells and any new cells that are to be brought into the active grid. It then passes the new active grid back to the main code which carries on with the simulation process.

Memory Efficiency

An important feature of the redistribution method is that it does not require any more memory than is used in the simple parallelisation method. When a redistribution takes place it is necessary to transfer the cells from the old active grid to a new active grid. This involves copying all the cells into a new array in memory. The original serial code upon which the parallel code is based, used three arrays to store three copies of the overall grid. This was to accommodate the updating of the cells during the time-steps. To avoid having to allocate a new block of memory for the new grid for the purposes of the redistribution routine, one of the three existing blocks of memory is recycled. This means that the outflows we can simulate with the redistributive method can be as big as those simulated with the simple parallelisation method.

3.2.3 Boundary conditions

As the active grid is resizing throughout the course of the simulation, we have constantly moving boundaries. It is important to ensure that at each instance of the creation of a new boundary, the new boundary conditions are set correctly. In the case of our jet-driven outflow simulations the bottom and left-hand boundaries are fixed for the course of the simulation and the top and right-hand boundaries move with the increasing size of the outflow. Each time either the top or right boundary moves, we re-initialise the boundary conditions on these boundaries.

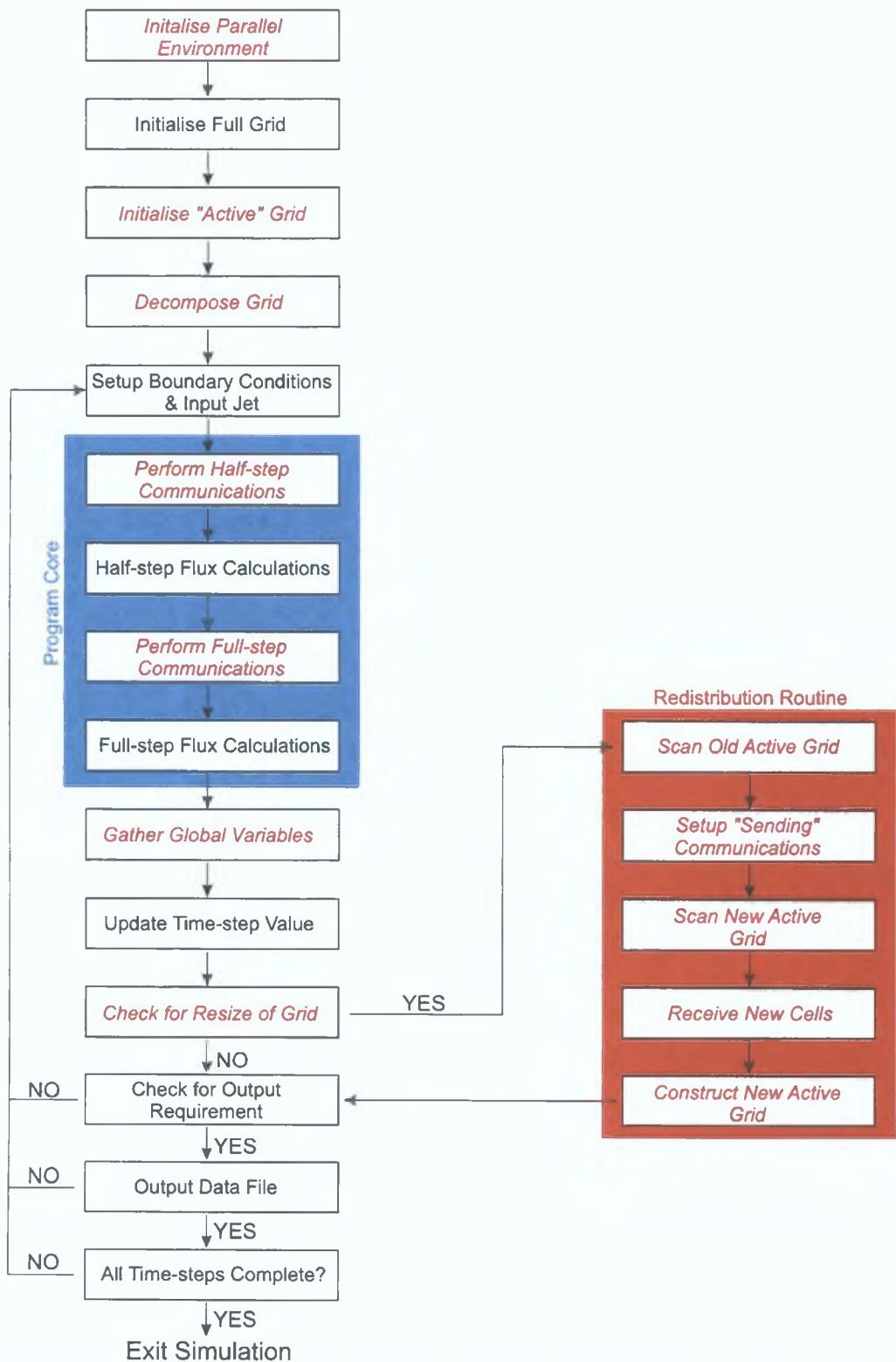


Figure 3.9: Flow chart for the redistributive parallel implementation of the code. Terms in red and in italics signify the parallel components that have been added to the underlying serial code (black text) to allow it to be executed on a cluster. The blue box represents the core component of the hydrodynamic code in which the cell interface fluxes are calculated. This is the most time-consuming part of the code and the parallelisation scheme is designed to distribute the work-load of this section among the processors.

Source term error in cylindrical symmetry

In the case where the simulation is done using cylindrical symmetry, care has to be taken to ensure that the small source-term error in the energy equation due to slight pressure differences at the top and bottom of each cell was accounted for in the new boundaries. Even though this is a negligible error in the general running of the simulation ($\sim 10^{-16}$), it gets compounded if not corrected each time there is a resize and can ultimately effect the way the head of the jet behaves. To stop this from happening, each time a resizing of the active grid takes place, the new cells in the processors on the right-hand boundary have their values set to the cell values immediately inside the original or old active grid.

As this source error causes lines of differing velocity, throughout the grid, running parallel to the jet axis, it is not of significance in terms of the moving of the top boundary. However, as the boundary moves outward, the new cells incorporated into the active grid have their transverse velocity set to that of the cells immediately below them on the old active grid. This means that there will be no velocity gradient at the old-new active grid interface.

Other boundary considerations

When a resizing of the active grid takes place, the communication of the non-moving boundary cells between processors is not necessary. However, the new cells in these boundaries need to be reset to have gradient zero with the cells adjacent to them immediately inside the grid after a resizing occurs.

3 2 4 Test Simulation

In figure 3 10 we can see an example of a simulation run with the redistributive parallel method. It shows an adiabatic jet simulation at three different stages in its evolution. We used a 1600×400 cells grid subdivided across 8 processors. The figure shows the

entire computational grid and the active grid contained on each processor. We can clearly see how the active grid expands across all of the processors to accommodate the expanding outflow. Using this method allows us to balance the load of the simulation more evenly across the processors than in the simple parallelisation method. The additional advantage is the method grants us the ability to run simulations in less time. In Chapter 4 we will examine the speed-up factor for the redistribution method relative to the simple method.

Additional test simulations were run at each stage in the development of the parallel code to ensure that they gave the same results as the original serial code. The most rigorous way of doing this was to run relatively long simulations (200-300 years) with the parallel code and serial code and compare the resulting output files.

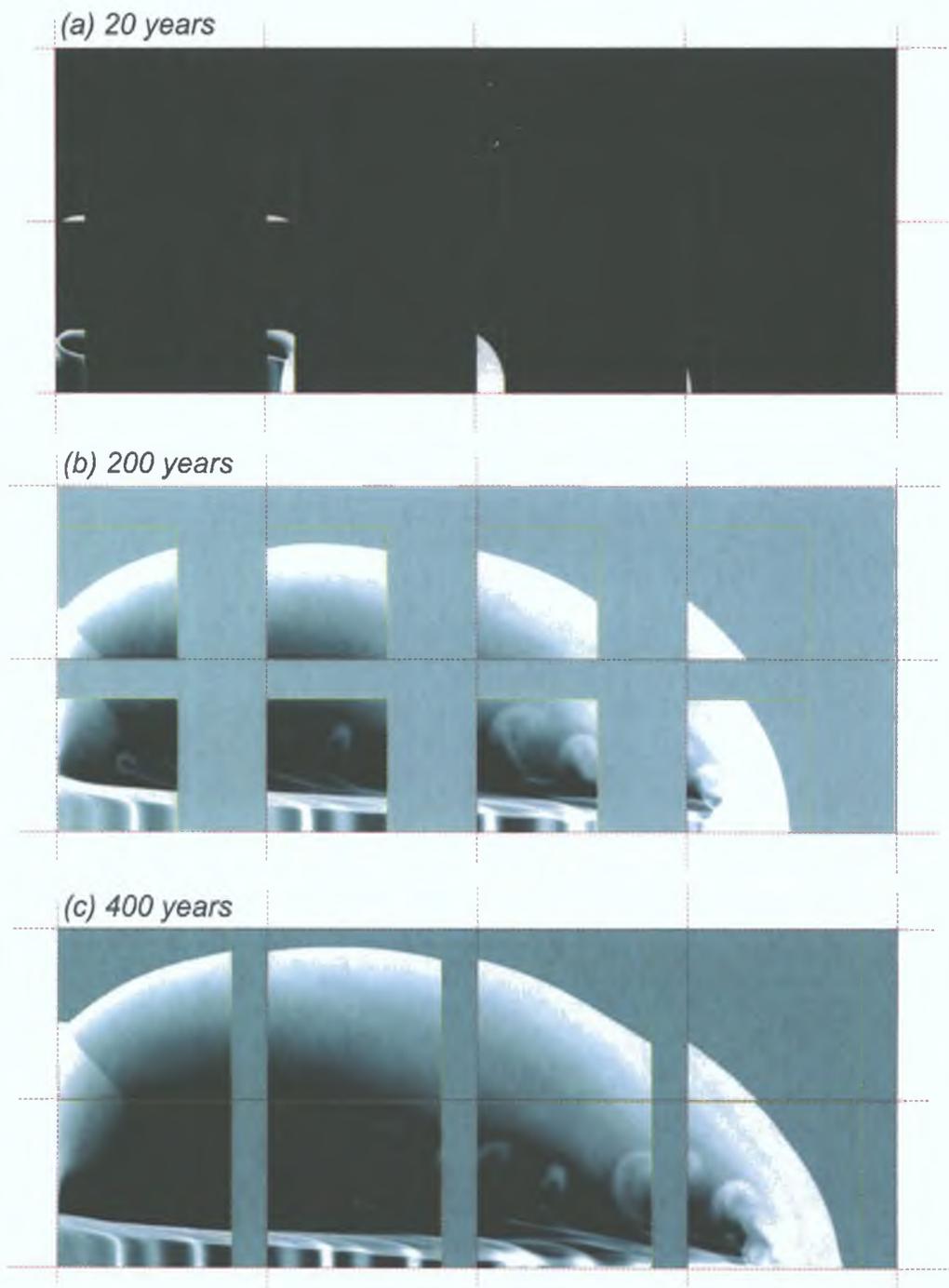


Figure 3.10: An example of an adiabatic jet simulation run over eight processors. Each image is taken at different stages in the evolution of the flow. As the jet propagates further into the computational grid, each processor computes over a larger and larger area.

Chapter 4

Parallel Code Testing

This chapter is concerned with the testing of the code parallelisation implemented in Chapter 3. To determine what kind of computational advantages can be gained from the parallelisation of the code, several tests were carried out. We conducted timing tests to see how the performance of the code scales with the addition of extra processors and we also conducted tests to see how efficient the parallelisation method is in balancing the computational load across the processors. The tests were carried out, firstly, with the 1-D code and then with the 2-D code. In addition, different implementations of the MPI standard were tested to see which was the best version to use. The MPICH version, from Argonne National Laboratory, and the LAM version, from Indiana University, were compared.

4.1 Amdahl's Law

Amdahl's Law (1967) is a law governing the speed-up gained when a parallelised code is used on a problem, compared with using a serial version of the code. It is based on the observation that each computation contains a "serial portion" that is not amenable to parallelisation. The speed of a program is the time it takes the program to execute. Speedup is defined as the time it takes a program to execute in serial (with one processor) divided by the time it takes to execute in parallel (with

many processors) The formula for speed-up is given in equation 4.1

$$\text{Speedup} = \frac{T_1}{T_N} \quad (4.1)$$

where T_N is the time taken to execute the program when using N processors and T_1 is the time for the serial execution on a single processor. Efficiency is the *Speedup* divided by the number of processors used.

$$\text{Efficiency}_N = \frac{\text{Speedup}}{N} \quad (4.2)$$

If we assume the strictly serial part of the program is performed in time $B \times T_1$, where B is the fraction of the code that is serial, then the strictly parallel part is performed in time $\frac{(1-B) \times T_1}{N}$. With some substitution and number manipulation, we get the formula for *Speedup* as

$$\text{Speedup} = \frac{N}{(B \times N) + (1 - B)} \quad (4.3)$$

The best speed we could hope for, $\text{Speedup} = N$ or $B = 0$, would yield a curve of $y = x$ if plotted on a graph with an x-axis of the number of processors compared against a y-axis of the speed-up. In practice, with a code suited to parallelisation, Amdahl's Law yields a speed-up curve similar to that shown in figure 4.1. For low numbers of processors $\text{Speedup} \sim N$, but for increasing values of N the curve falls away from ideal speed-up. Adding even more processors eventually leads to the curve flattening out and the efficiency is slowly reduced to the point where more processors make no improvement or actually reduce the efficiency. The main contributor to this flattening of the speed-up curve is the time overhead caused by the inter-processor communications.

There are two factors which can affect the length of time it takes to do a communication: latency and bandwidth. Latency is basically the length of time it takes to transmit a message from one location to another. This is most important in what

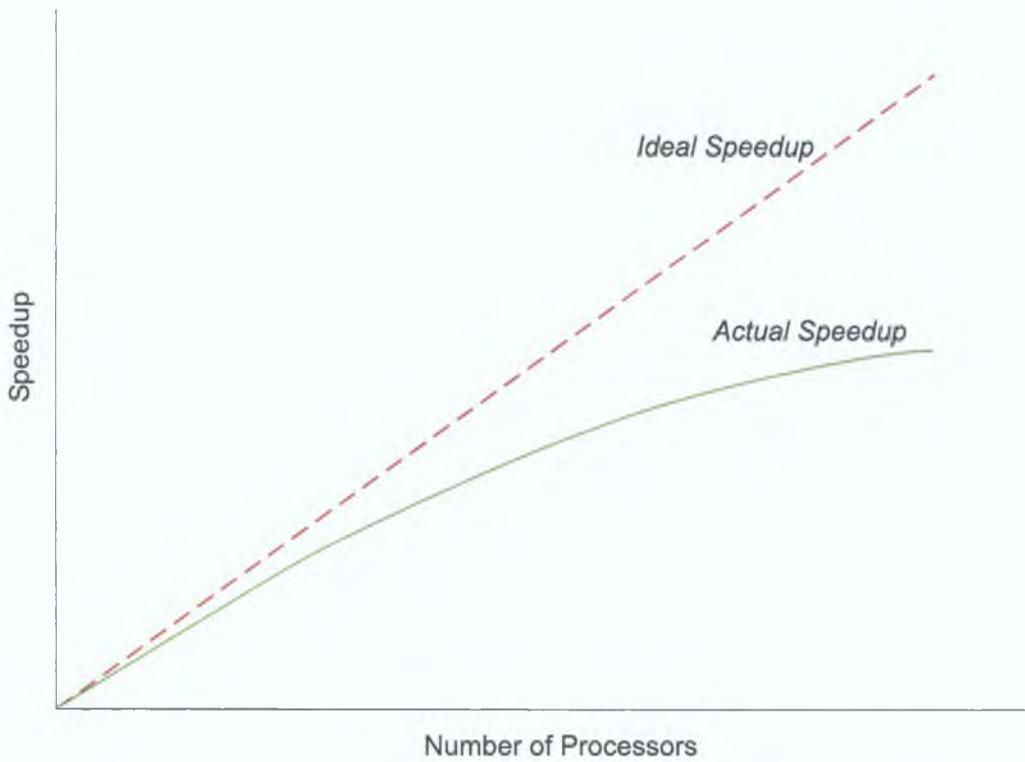


Figure 4.1: Amdahl's Law. As more processors are used the efficiency decreases resulting in a fall-off in the speed-up factor. This is due to the fact that a parallelised code will almost always contain a serial component that is not amenable to parallelisation and the communication overhead involved in using large numbers of processors.

is referred to as the “hand-shaking” process that goes on between processors at the beginning and end of a communication. The second influencing factor, bandwidth, is the capacity of the communication link between any two processors, i.e. the amount of bytes that can be transferred down the link per second. If the amount of data being communicated for any single communication is large, then bandwidth is usually a much greater contributing factor to the communication overhead than latency. In the case of our jet simulations, latency is much more of a problem, as there are many small-size communications taking place for each time-step.

4.2 Parallel computing environment

All testing and simulations in the present work were performed on a 32-processor Linux-based Beowulf cluster dedicated to astrophysical and geophysical simulations based at the Dublin Institute for Advanced Studies. A cluster is a type of parallel or distributed system that consists of a collection of interconnected whole computers and is used as a single unified computing resource. The cluster we have used has the following specifications:

- Central Master Node
 - 1GHz Pentium4 machine,
 - 400MB SDRAM,
 - Running Linux Operating System,
 - Used for analysing data generated on main cluster

- 16 Slave Nodes
 - 16 Dual Pentium3 600Mhz machines,
 - 512MB SDRAM per machine,
 - 2GigaByte local hardisk in each machine,

- 3Com 100MB network cards,
- Dual switch,
- Linux-based operating system,
- MPICH and LAM versions of the MPI message passing library

The naming of the cluster comes from the sixteen stars in the Ursa Major Cluster which is part of the Ursa Major or Great Bear constellation, commonly referred to as “The Plough”. Each of the nodes in the cluster is named after one of the stars in the constellation.

For most of the timing results presented in this chapter, adiabatic jet simulations using up to 28 processors were run for 1.0×10^4 time-steps. For the testing of various communication time overheads, a simulation on 28 processors with 1.5×10^4 time-steps was used. All timing of the code was done using the MPI function *MPI.Wtime()*.

4.3 1-D Testing

In the 1-D code communications only take place in one dimension and only involves two cells being sent in either direction. Figure 4.2 illustrates a simple timing test done over 30 processors. The performance of the parallelised 1-D code is relatively poor compared to the 2-D parallel code but this is due to the low ratio between communication time and the time spent performing flux calculations between the communications.

4.4 2-D Testing

In the 2-dimensional simulations, there are a lot more communications necessary. In the worst case scenario (figure 4.3), a processor may have potentially four neighbours to communicate their entire borders with. Each processor has to send and receive two columns or rows of cells with each of its neighbours. Before communication can take

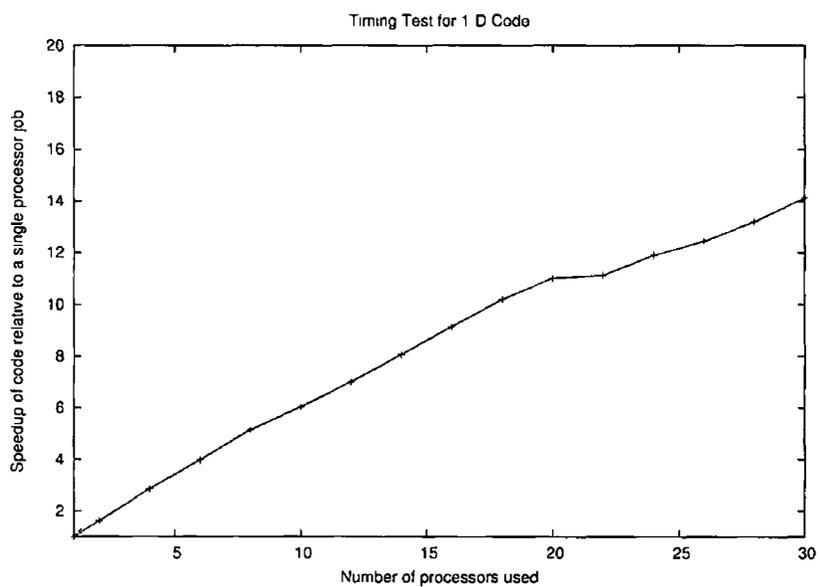


Figure 4.2 Timing results for the 1-dimensional code using a redistribution algorithm. The speed-up of the code relative to using a single process averages at about 0.61 of the perfect speed-up which is quite poor. The reason for this is due to the fact that the communication time relative to the calculation time between communications is quite large for a one-dimensional code. Using a larger amount of cells in the simulation would result in improved parallel performance.

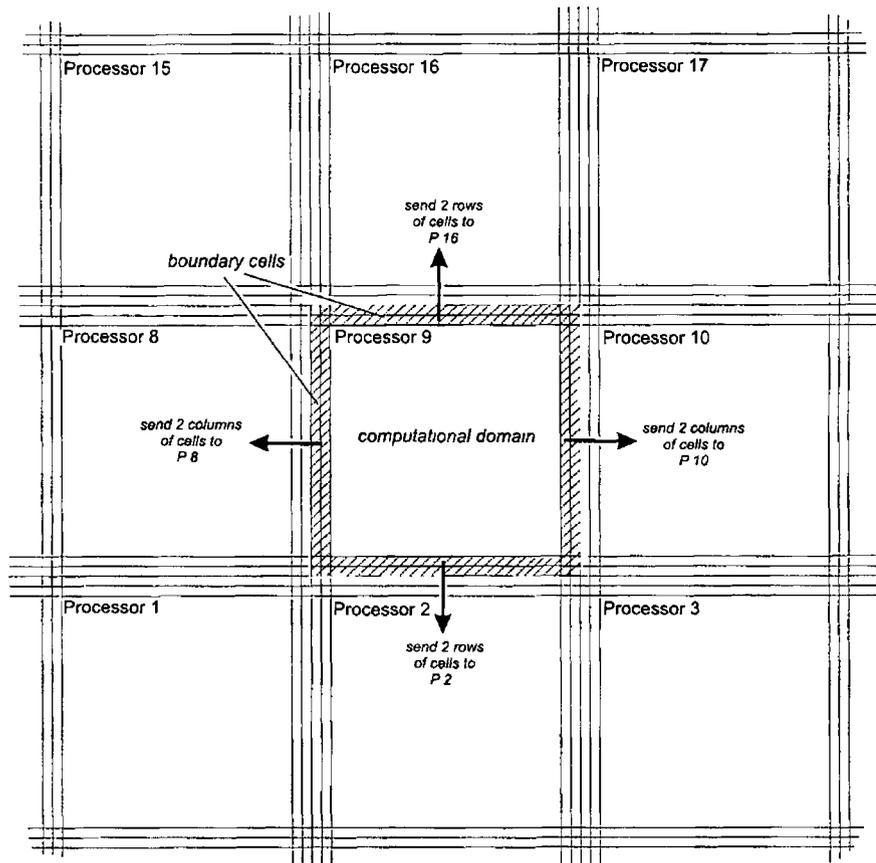


Figure 4 3 In the worst case scenario, a processor will have to communicate its boundary cells with four of its neighbouring processors. In this case processor 9 is sending either two rows or columns of cells to processors 2, 8, 10 and 16. This scenario results in this processor having the highest possible communication overhead.

place, these cells must be buffered into arrays which are contiguous in memory. These blocks of arrays are communicated using the MPI library functions for sending and receiving messages and then unpacked into their correct positions on the neighbouring processor. This all adds to the length of time it takes to do a complete communication step.

With the increasing size of the jet, the memory domain on each processor gets larger and the amount of time it takes to do a complete cycle of the main loop increases. This results in the communication time between processors becoming less significant. If we consider a processor with a grid of size $n \times n$ cells, that is commu-

communicating in all four directions, then the largest the total size of the communication buffers will be is $4 \times 2 \times n$ cells (two columns or rows on each boundary). Clearly, then, the larger n is, the greater the ratio between the size of a processor's domain and the size of its boundary buffers. Hence, the communication time becomes less, relative to the time it takes to do the flux calculations for all the cells on the processors domain.

4.4.1 Simple domain-decomposition method testing

Here, the timing results for the code are presented for the case where the full computational domain is split across all the available processors. All processors then perform their calculations over their entire section of the grid for the total number of allocated time-steps. The test is performed using a grid of size 2000×500 cells. A simple simulation is performed for incremental numbers of processors, initially using a single processor and culminating in the use of twenty-six processors. As even numbers of processors are required for the code to work, apart from in the serial case where a single processor is used, results are only presented for tests using even numbers of processors. A value of 4.0×10^4 was used for the number of time-steps in each test to allow the jet to evolve to a mature state similar to what is needed in the full simulation runs. Figure 4.4 shows the resulting speed-up. In these tests we have maximised the domain size to communication buffer size ratio and hence, minimized the communication overhead. As a result of this, the performance of the code scales well with increasing numbers of processors.

4.4.2 Redistributive domain-decomposition method testing

In this section we present the results for the timing tests conducted with the code utilising the redistribution method for load-balancing the simulation across all the processors. In this case, there are three places where inter-processor communications are significant in the overall time to execute the code. Firstly, as with the simple parallelisation, there are the inter-processor boundary communications taking place

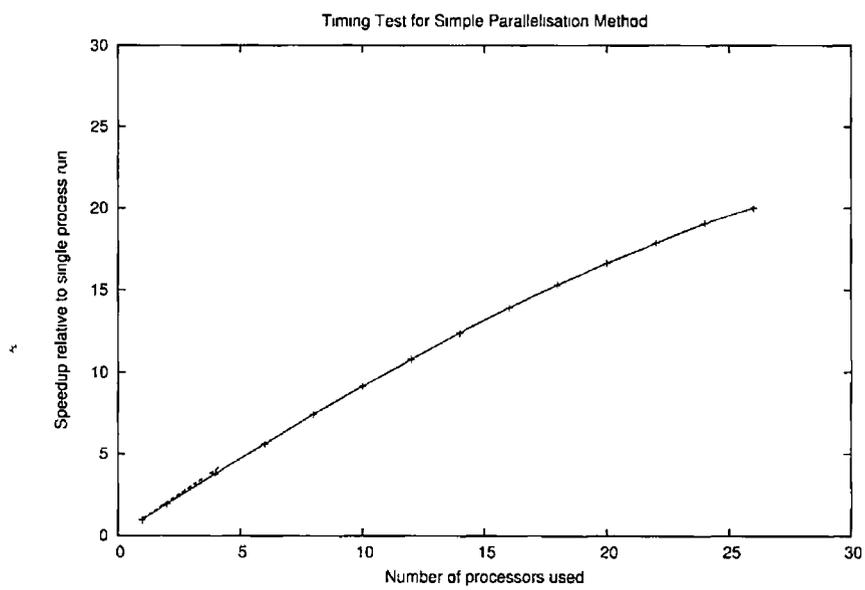


Figure 4.4 This is the timing results for the testing of the simple parallelisation method, where the total computational grid is divided and stored according to the number of processors available. The dashed line indicates the perfect speed-up situation i.e. if the speed-up scaled exactly with the number of processors used. The continuous line is the actual speed-up gained. The results are shown for tests on one to twenty-six processors. When twenty-six processors are used the speed-up is 0.78 of the perfect speed-up.

twice every time-step. Secondly, there are the global communications occurring at the end of each time-step, and finally, there are the communications taking place between processors whenever a redistribution of cells occurs. The rate at which redistributions take place is dependent on the number of extra cells incorporated into the grid each time there is a resizing of the grid. To determine the optimum size for this volume of extra cells some timing tests were done.

Testing for the optimum redistribution volume

A large increase in the number of cells on the grid at each redistribution step translates into greater numbers of time-steps between redistributions. The volume of extra cells incorporated into the grid at each redistribution is directly proportional to the number of time-steps between redistributions. A larger volume of cells means more time-steps between redistributions and a smaller volume means fewer time-steps. Although using large volumes translates into the need for fewer communications, the drawback is that the processors are wasting more time doing flux calculations between cells representing the ambient medium. To establish the optimum number of time-steps between redistributions, we conducted a series of timing tests. In each test we varied the number of cells added to the computational grid during each redistribution. Figure 4.5 is a plot of the times taken to execute these simulations in terms of the number of time-steps between redistributions. This plot shows that the larger the volume of cells added, the longer the overall execution time for the simulation. This means that the communication time during a redistribution is negligible and for full jet simulations a small value should be chosen for the numbers of rows and columns of cells added on to the grid each time it needs to be resized.

Communication time compared with overall run time

A test was carried out to see what the overall percentage of total time the communications would consume for a long-term simulation. A worst case scenario was used

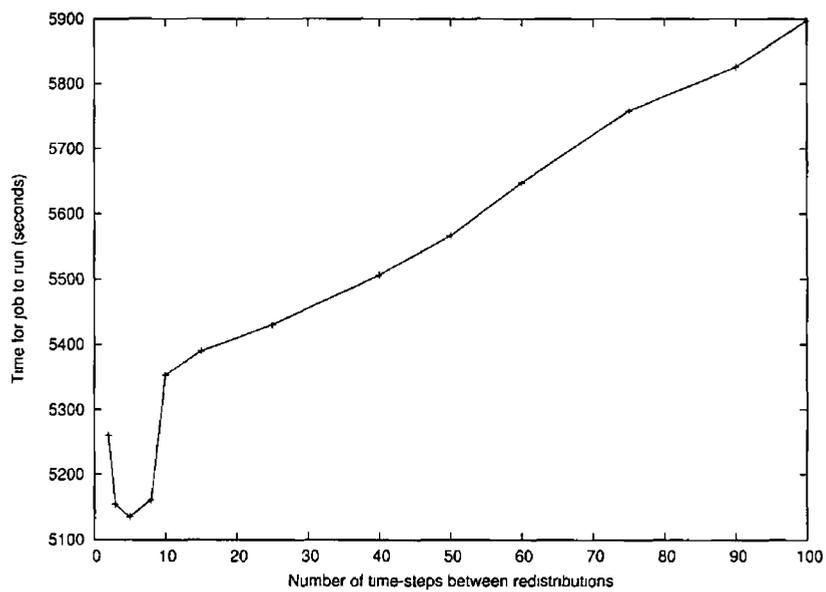


Figure 4 5 Timing results for tests on varying the number of time-steps between redistributions. The smaller the time-step gap, the greater the efficiency of the code. The breakpoint in the slope of the plot at 10 time-steps and the subsequent increase in the slope of the plot below this point, is as a result of the fact that the grid is only being increased m size m in one direction for each redistribution. This results in a smaller communication overhead for each redistribution.

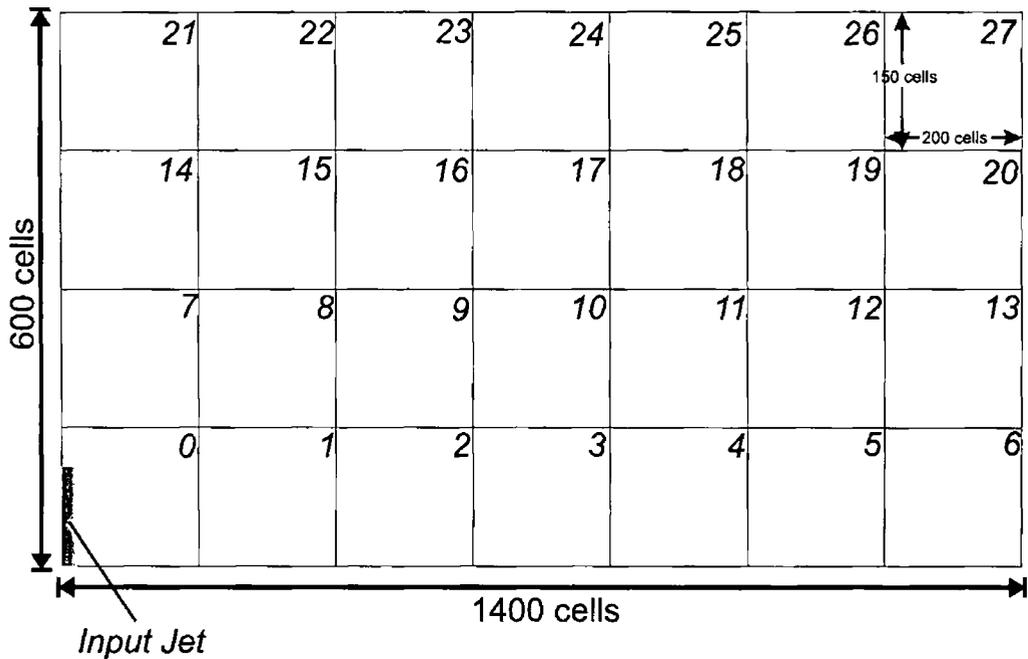


Figure 4 6 To test the communications overhead in a full scale simulation we set up a test grid of 1400×600 cells subdivided into 7×4 blocks. We used 28 processors for the tests and each processor took one of these blocks. We choose this domain decomposition to maximise the communications overhead.

to do this test. A simulation of an adiabatic jet on a grid of 1400×600 cells was split among 28 processors (figure 4 6). The domain was subdivided into 7×4 processors to maximise the communication overhead. Most of the processors would have to communicate in at least three directions and those not on a boundary would be communicating in all four directions.

From figure 4 7 we can see that the amount of time spent communicating per time-step falls as the grid expands in size. This is in agreement with the argument presented in section 4 4. After 1.5×10^4 time-steps, the communication overhead is below 10% of the total execution time. The communication time varies quite widely from processor to processor. Processor 27, which lies in the top right of the grid, starts out with a relatively low communication overhead as it and its neighbouring processors do not yet contain any part of the jet. When the jet fills the total initial grid size, including the domain of processor 27, the communication time for this

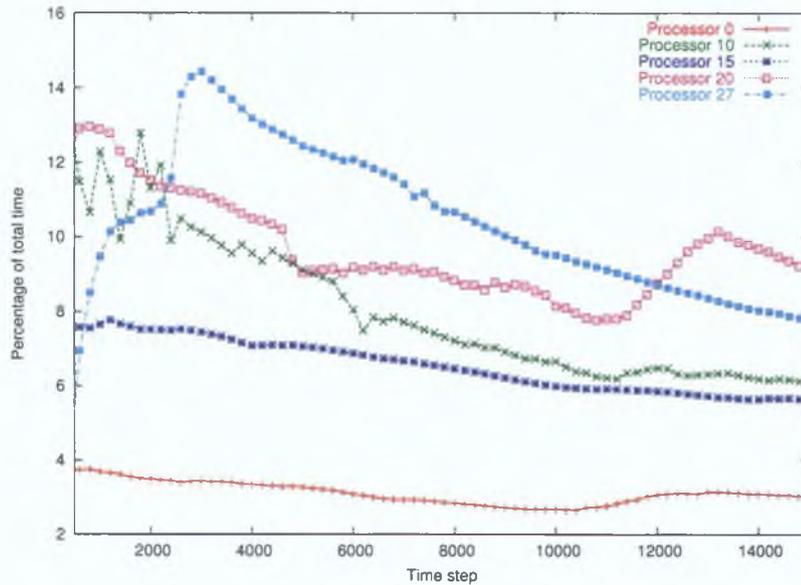


Figure 4.7: Percentage of total execution time taken up by communication of cells over 1.5×10^4 time-steps. Results are taken from a selection of processors covering different parts of the overall grid. As expected, the proportion of time spent communicating cells falls off as the grid increases in size. (Note that this test was done using the redistribution code)

processor increases rapidly to about 15% of the overall time and then declines as the grid size continues to expand.

Global communications

At the end of every time-step, a set of global communications are carried out for synchronising the value of certain variables, such as the Courant number, across all of the processors. Global communication functions in MPI, such as the *MPI_Allreduce* function, are known to have a large time overhead as they involve all processors communicating with all other processors to achieve a global value. Times were measured for this part of the code across all the processors. Results for several processors on different parts of the grid are presented in figure 4.8. The overhead for these communications is quite large initially, between 7% and 11% of total execution time, but as the grid expands, the relative time spent on this step decreases to an acceptable level. In this test we used 1.5×10^4 time-steps and by the end of the simulation the

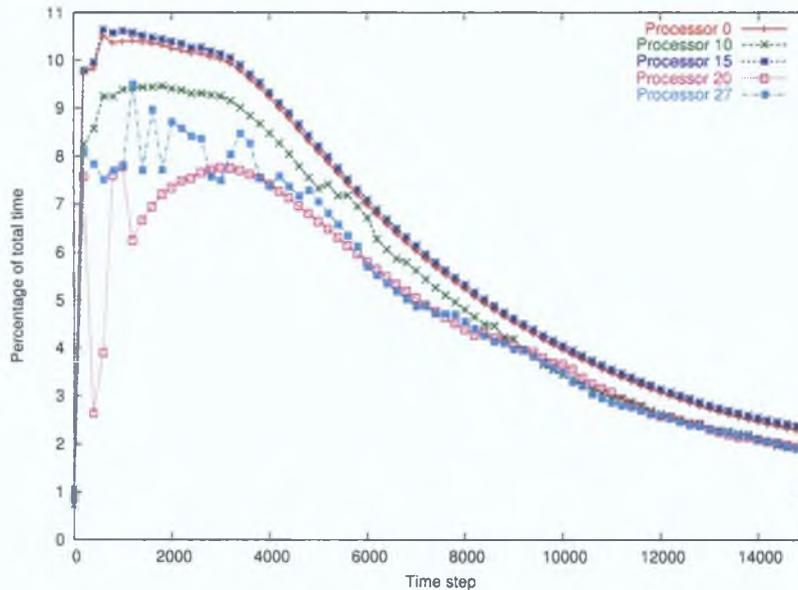


Figure 4.8: Percentage of total execution time taken up by global communications at the end of each time-step. The results are for a 28 processor simulation running over 1.5×10^4 time-steps. Results are taken from a selection of processors covering different parts of the overall grid. (Note that this test was done using the redistribution code)

global communications were occupying 2% of the overall execution time.

Redistribution time compared with overall run time

As with the cell communications and global communications, a test was carried out to see what the overall percentage of total time the redistribution algorithm would consume for a long-term simulation. For this test the same set-up was used as in section 4.4.2. The results in figure 4.9 show that the redistribution algorithm consumes a negligible fraction of the overall execution time. Initially, the redistribution time is zero until the initial grid is filled by the jet and grid resizing begins, accommodating the expanding flow. At this point the redistribution routine is called and the percentage time consumed by it increases rapidly with grid expansion until it plateaus at between 0.5 and 0.6 percent of the total execution time. Beyond this point, the proportion of overall time spent redistributing cells decreases linearly with increasing time. This is due to the greater amount of time spent performing the

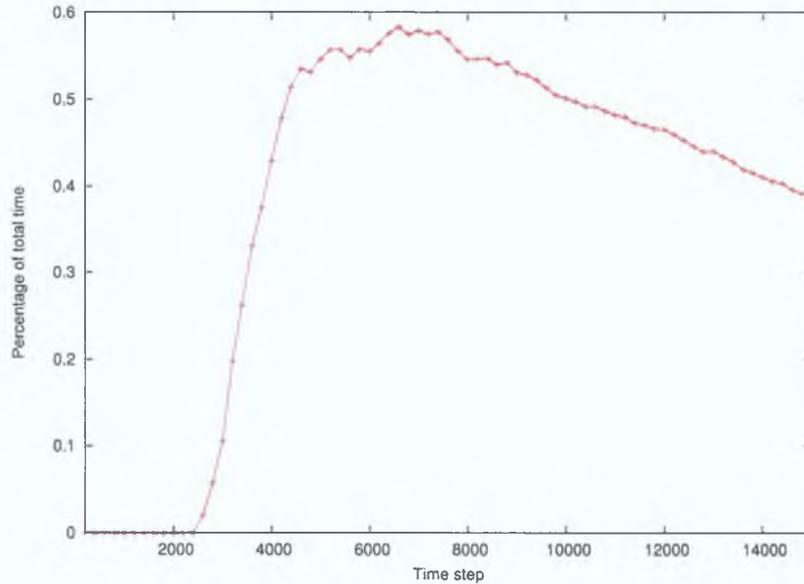


Figure 4.9: Percentage of total execution time for redistribution of cells over 1.5×10^4 time-steps. This is results is taken from processor ‘0’ in a 28 processor simulation. Processor ‘0’ lies in the bottom left-hand corner of the grid and has the highest redistribution time as it only receives cells and has to wait for all sending processors to initiate their communications before it can copy its new cells into place and exit the redistribution routine.

flux calculations by the processors as the grid gets bigger.

Overall timings with redistribution

Figure 4.10 shows the resulting speed-up of the code utilising the redistribution method. The speed-up using 28 processors is about a factor of 0.5 of the ideal speed-up. Figure 4.11 compares this result with the speed-up gained from the simple parallel code. The concentration of the processing power in the jet flow region achieves a speed-up about 3.5 times faster than the simple domain decomposition method.

To achieve a speed-up closer to ideal speed-up would require a more sophisticated algorithm for balancing the computational load among the processors. The head of the jet involves the most work and processors whose domain contains this region spend more time calculating the fluxes between cells and the cooling than most of the other processors. Figure 4.12 shows the amount of time, as a percentage of

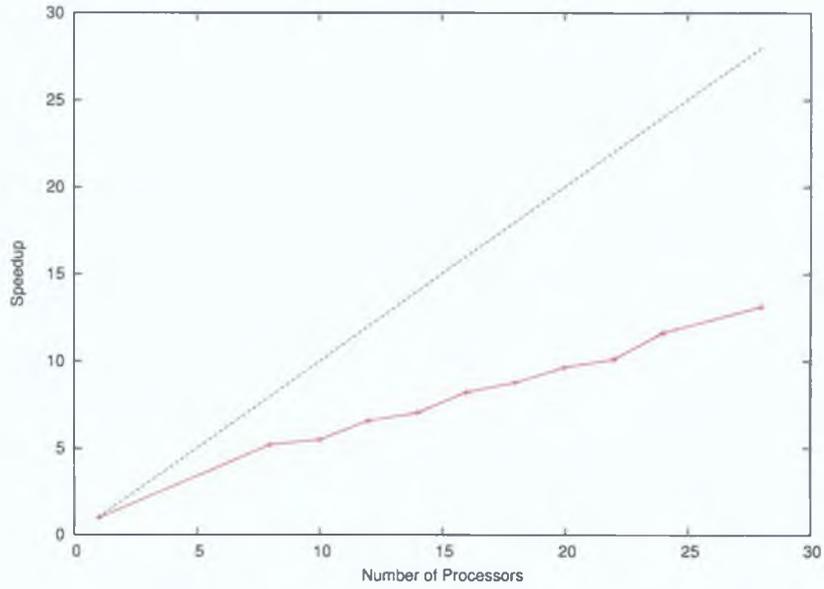


Figure 4.10: Speedup of redistribution parallel code over 1.0×10^4 time-steps. The dashed line represents the ideal speed-up.

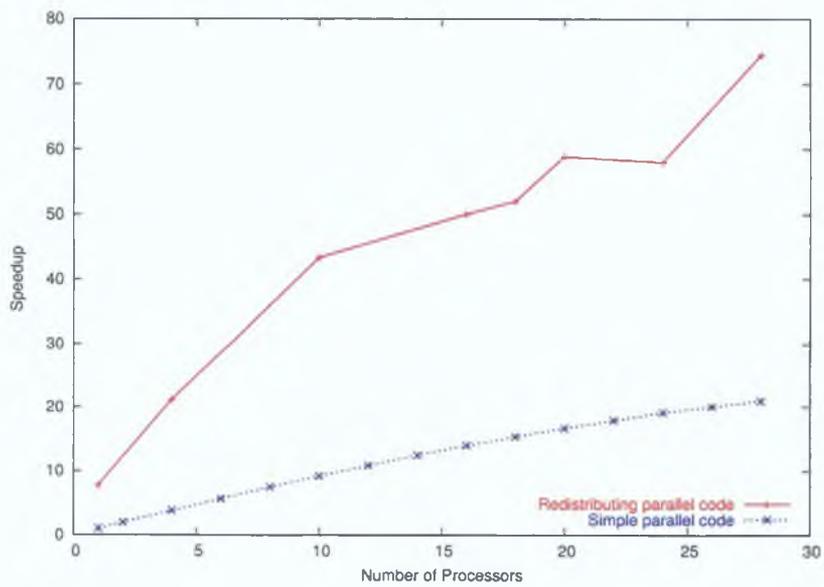


Figure 4.11: Comparison between simple and redistributing parallel codes for an adiabatic jet simulation over 1.0×10^4 time-steps.

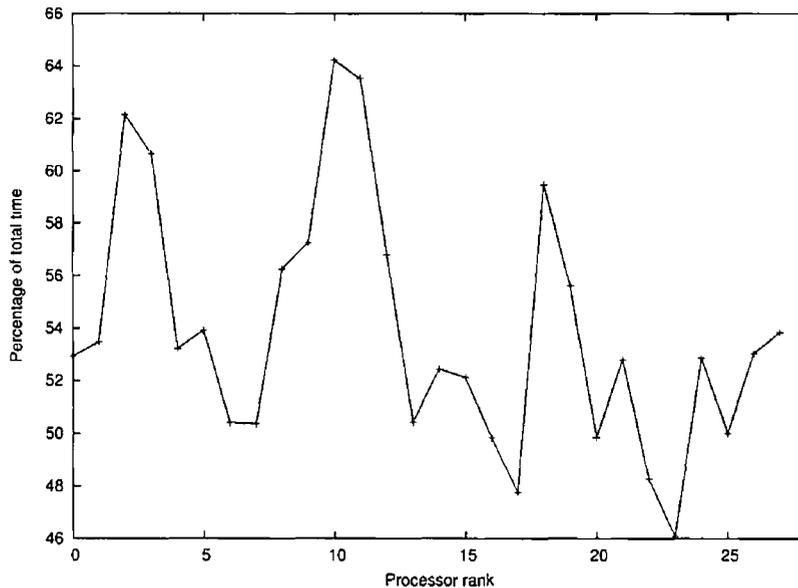


Figure 4.12 Percentage of total execution time to do the flux calculation steps on all processors in a 28 processor simulation running to 1.5×10^4 time-steps. To achieve a speed-up closer to ideal speed-up, the value for this percentage would have to be brought into equality across all the processors. This would be achieved with better load balancing.

overall execution time, spent doing the flux calculation steps over the course of a 1.5×10^4 time-step simulation for all of the processors involved in the simulation. The percentage varies from 46% to 64% depending on the position of the processor on the computational grid. The communication steps cannot take place until all of the processors have finished their flux calculations. Apart from small differences in the cell communication time (figure 4.7), all other parts of the code are executed in the same length of time across all the processors and hence the speed of the code is determined, predominantly, by the slowest processor in the flux calculation steps. Possible solutions to this problem will be discussed in Chapter 7.

4.5 Testing different versions of MPI

The two main, freely available, distributions of MPI are the MPICH version from Argonne National Laboratory and the LAM version from the Open Systems Lab at

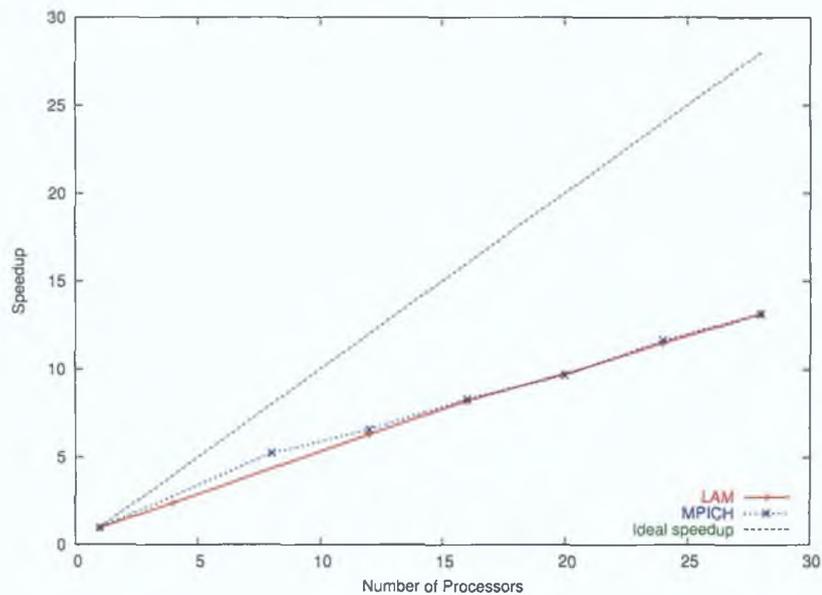


Figure 4.13: Speedup comparison between LAM MPI and MPICH MPI implementations. Here the code was run to 2.0×10^4 time-steps, about 650 years, for an adiabatic jet simulation.

Indiana University. There are several subtle differences between these two implementations which can result in contrasting performances depending on the particular problem. It was therefore important to establish which of these two implementations best suits the needs of the parallelisation of the jet simulation code.

4.5.1 Features of each Implementation

The most significant difference between the two implementations is the use of “daemons” by LAM. Before running a parallel code on a cluster, these daemons need to be set running on each of the machines in the cluster. The function of the daemons is to speed up the process of starting a parallel job and reducing the latency of communications between processors. When a parallel job is initiated with MPICH, it needs to open a connection to each of the machines to be used in the simulation which takes up a certain amount of time. However, whether or not one implementation is better than the other is highly dependent on the parallel code to be executed.

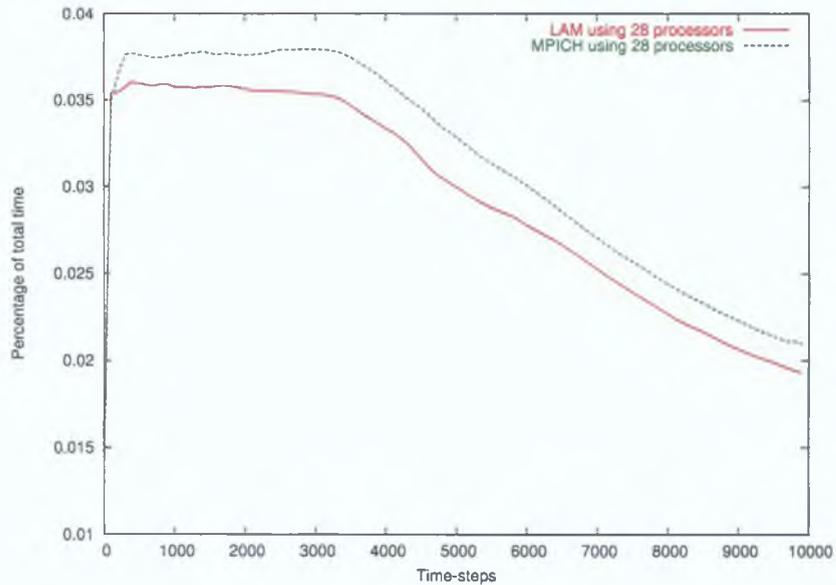


Figure 4.14: Comparison of the percentage of execution time used for communicating cells in the main loop of the code for LAM and MPICH. LAM is marginally more efficient due to the use of daemons. This result is taken from processor 0 in a 28 processor simulation for each case.

4.5.2 Results and Comparisons

Each of the two implementations were used to compile the code and then sample simulations were run, using both versions, to sufficient lengths of time so as to gain a good impression of the contrasts, if any, in their performance. The load-balanced, redistributive code was used as this was the form in which the code would be used for full scale simulations. Again, each test was done for several numbers of processors and speed-up graphs were created from each set of results. As shown in figure 4.13, there is little to choose from between the two implementations. Looking at the percentage time spent communicating, figure 4.14, we see that the LAM code is slightly more efficient, reflecting the benefit of using daemons. However, the speed-up gained is not significant and as the communication time overhead lessens with increasing grid size, the speed-up becomes almost identical.

Chapter 5

Long Duration Simulations of jets from YSOs

In this chapter we shall discuss the results of the long-duration simulations that we carried out using the parallelised code. We shall investigate the results for a jet with a pulsed input velocity profile first and then one with an episodic input velocity profile. The pulsed jet was run to 2300 years and the episodic jet to 2500 years. We shall examine the behaviour of the simulated molecular outflows and how this behaviour compares with that seen in observations of real systems. The results and discussion here is a more detailed version of the work contained in Keegan & Downes (in press, 2005).

For our simulations we have adopted the jet model (see figure 1.1 in Chapter 1). The fundamental theory behind this model is that a supersonic, highly collimated jet propagates into the ambient medium forming a bow-shock surface at the head of the jet. This bow-shock carves into the ambient medium accelerating material at its head in the direction of the jet. For a more detailed description of the bow-shock model see section 1.2.1 in Chapter 1.

5.1 Initial Conditions

Each of the simulations have the same initial conditions apart from the velocity of the input jet. All of the simulations are carried out in cylindrical symmetry. The

jet and ambient densities are matched at 100 cm^{-3} . The pressure is defined such that the temperature of the input jet is 10^3 K , while the temperature of the ambient medium is set to 10^2 K . A value of 9.1 is chosen for the ratio between the number densities of molecular and atomic hydrogen, $\frac{n_{\text{H}_2}}{n_{\text{H}}}$. The gas is assumed to be one of solar abundances. The grid cell spacing is set to $1 \times 10^{14} \text{ cm}$. This is deemed the absolute minimum to get reliable results as the efficiency of momentum transfer is sensitive to grid spacing (Downes & Ray, 1999). The jet radius is set to 50 cells ($5 \times 10^{15} \text{ cm}$) with a shear layer of 5 cells which is used to avoid unphysical effects occurring at the edge of the input jet as it enters the grid.

5.2 The Pulsed Jet

For the pulsed jet we have used a time-averaged jet velocity, v_0 , of 215 km s^{-1} (mach number = 93) with sinusoidal variations superimposed upon it with periods of 5, 10, 20 and 50 years with a total amplitude of v_1 . In this simulation we have set $v_1 = 34.6 \text{ km s}^{-1}$ (mach number = 15).

5.2.1 Morphology

Figure 5.1 contains log-scale plots of the number density at several times in the evolution of this simulation. A striking feature of the bow-shock is the presence of corrugations along its length. These corrugations have been noted before (Smith et al., 1997, Downes & Ray, 1999) and they are thought to come about due to instabilities, such as the Rayleigh-Taylor and/or Vishniac instabilities, occurring at the head of the jet (Blondin et al., 1990, Vishniac, 1994, Dgani et al., 1996). Within the jet, the shocks along its axis can be seen to eventually merge into strong shocks the further away from the source that the jet propagates. This is a result of the slower moving shocks being caught by the faster ones and combining into a single strong shock with their combined momentum.

Another feature is the length-to-width ratio or collimation factor, q . q is calculated by taking the maximum width of the outflow and the length from the boundary where the jet is input to the most advanced point at the head of the bow-shock. At 2300 years the outflow has a collimation factor of about 10.2. Many observed outflows have collimation ratios similar to this value, particularly outflows that are deemed to be in the early stages of their development (Richer et al., 2000). Some typical examples of highly collimated outflows are HH240/241 (Cohen, 1980, Bohigas, Persi & Tapia, 1993), HH211 (Gueth & Gulloteau, 1999) and HH212 (Zinnecker, McCaughrean & Rayner, 1998).

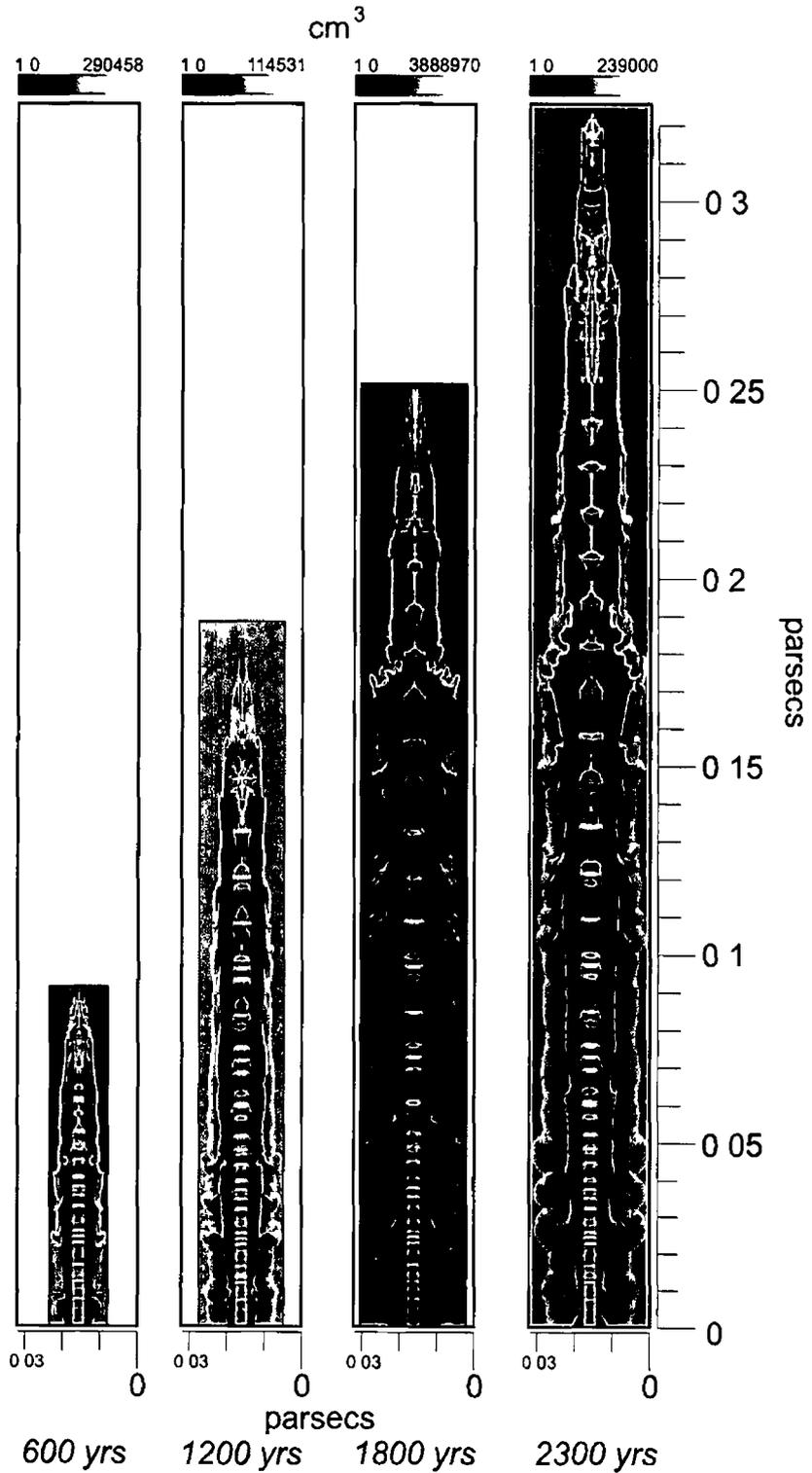


Figure 5.1 Log-scale plot of the number density for the pulsed jet at 600, 1200, 1800 and 2300 years. Note the formation of strong shocks at later times towards the head of the jet and the corrugations along the bow-shock. The corrugations are thought to arise from instabilities occurring at the head of the bow-shock. The outflow is also highly collimated with a length-to-width ratio of 10.2 at 2300 years.

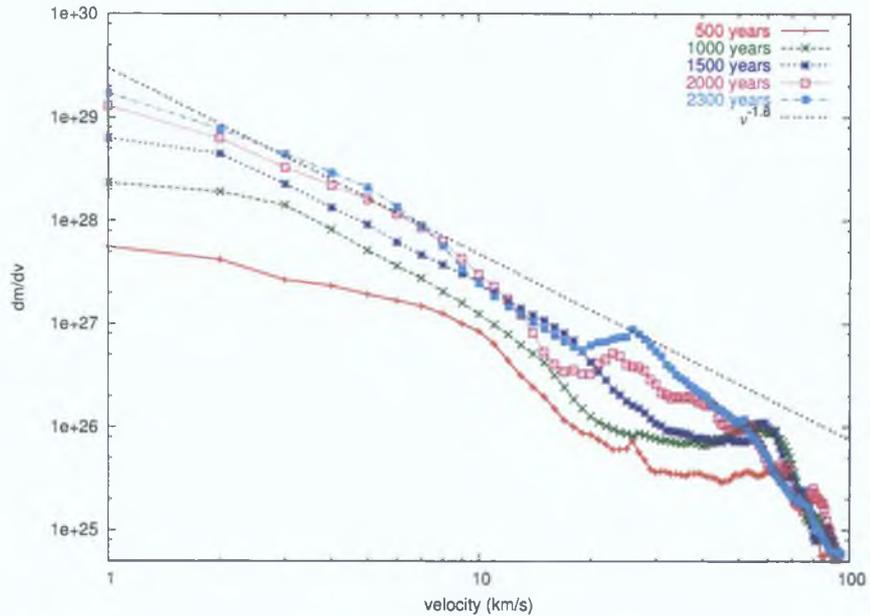


Figure 5.2: The mass-velocity relation, $m(v)$, for the swept-up ambient material at different times during the pulsed jet simulation. Above 1500 years the relation settles to a quasi-stable state. The units for dm/dv are in g s/km .

5.2.2 Mass-velocity Relationship

The behaviour of the mass-velocity relation, $m(v)$, in the pulsed jet simulation is shown in figure 5.2. The total swept-up mass as a function of velocity is plotted at several times during the simulation. From the plot we can see that the relation settles to a quasi-stable state at $t > 1500$ years.

At later times there continue to be substantial variations in the $m(v)$ relation for material moving at velocities greater than 20 km s^{-1} . This stems from the occurrence of corrugations in the bow-shock. These corrugations can contain a large amount of material and appear as “bumps” on the $m(v)$ relation. The bumps first appear as high velocity material corresponding to the formation of a corrugation by an instability at the head of the bow-shock. Over time, as the corrugations “fall off” the head of the bow-shock, these bumps propagate towards the slower moving material in the $m(v)$ relation and are eventually “swallowed-up” by the large amount of material in the lower velocity ranges.

As noted in section 1.1.4 and section 1.2.2 of Chapter 1, from observational results the mass-velocity relation is found to obey a power-law of the form

$$m(v) \propto v^{-\gamma} \quad (5.1)$$

One of the main objectives of this work is to investigate the behaviour of the value for γ , the power-law exponent in the $m(v)$ relation, with the passage of time. Earlier investigations (Smith et al., 1997, Downes & Ray, 1999), for simulations over several hundred years, found that γ tended to agree with observations but was found to increase with time. The question to be answered here is whether or not the upward trend in the value for γ continues to be the case for long duration simulations. Figure 5.3 displays the low velocity component (1-10 km s⁻¹) of the $m(v)$ relation at several different points in time. We can see that for $t < 1500$ years, γ increases monotonically with time. Smith et al. (1997) postulated that the steepening in the value for γ is due to the bow-shock becoming more aerodynamic as the outflow ages. Downes & Cabrit (2003) suggest that the steepening may be due to the increasing size of the sides of the bow-shock as the jet propagates further onto the computational grid. In this simulation, if we examine the $m(v)$ relation for material moving in the velocity range 1-10 km s⁻¹, we can see evidence for this theory. As the length of the outflow increases we have more and more material which has been accelerated by the leading bow-shock falling into the elongated wings of the bow-shock and slowing down. The resulting effect on the $m(v)$ relation is that the slope of the low velocity component, $v < 10$ km s⁻¹, starts with a value ~ 0 and steadily increases until it reaches a stable state at $t > 1500$ years. From figure 5.3, we can see that beyond $t = 1500$ years the value for γ remains approximately constant before beginning to increase slightly again after $t \sim 1800$ years. The reason for this increase at later times is due to large variations at higher velocities propagating into the 1-10 km s⁻¹ velocity range and is not like the steady increase seen at earlier times. These variations are caused by the “corrugations” mentioned in section 5.2.1 caused by instabilities at the head of the

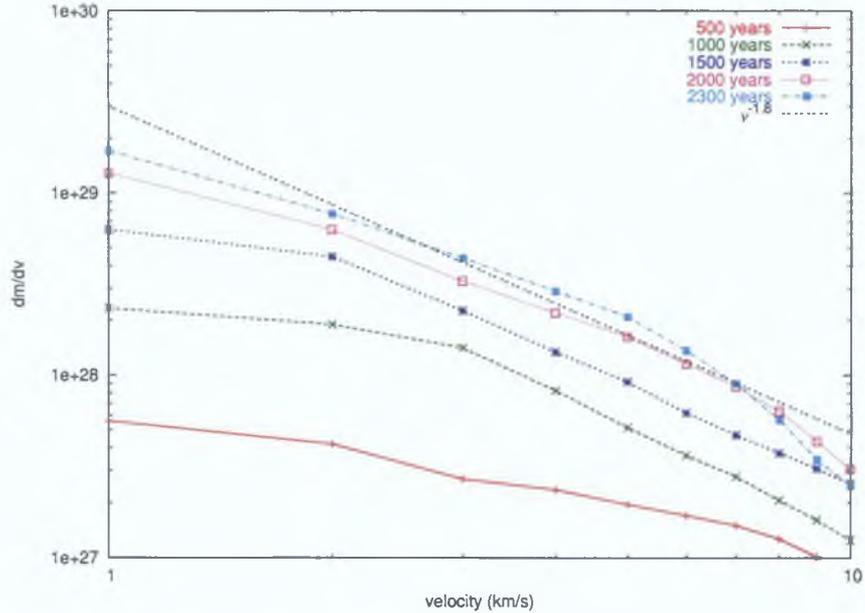


Figure 5.3: Mass-velocity relation for the low velocity component of the outflow, $v \leq 10 \text{ km s}^{-1}$. The relation is shown at 50, 500, 1000, 1500, 2000 and 2300 years. Here, the jet is considered to be at an angle of 30° to the plane of the sky. The units for dm/dv are in g s/km .

bow-shock.

Figure 5.4 shows the time-averaged $m(v)$ relations and also the mass-velocity relation for the swept-up molecular material, $m_{\text{H}_2}(v)$, between 1500 and 2300 years. Figure 5.4 shows no obvious break in the power-law, in agreement with what was found by Downes & Cabrit (2003) and at odds with the assumption that the $m(v)$ relation should follow a broken power-law due to it being assumed that the mass in the flow is proportional to the observed integrated intensity for the CO J=2-1 line (Zhang & Zheng, 1997; Smith et al., 1997). This also appears to contradict the observations but the discrepancy can be resolved by noting that the temperature is not constant throughout the outflow. At higher velocities, the temperatures are higher. When using the CO J=2-1 emission for determining the mass, it is necessary to account for these higher temperatures for the higher velocity material. There is a $1/T$ dependence of emission at temperatures exceeding the upper level of this line. In

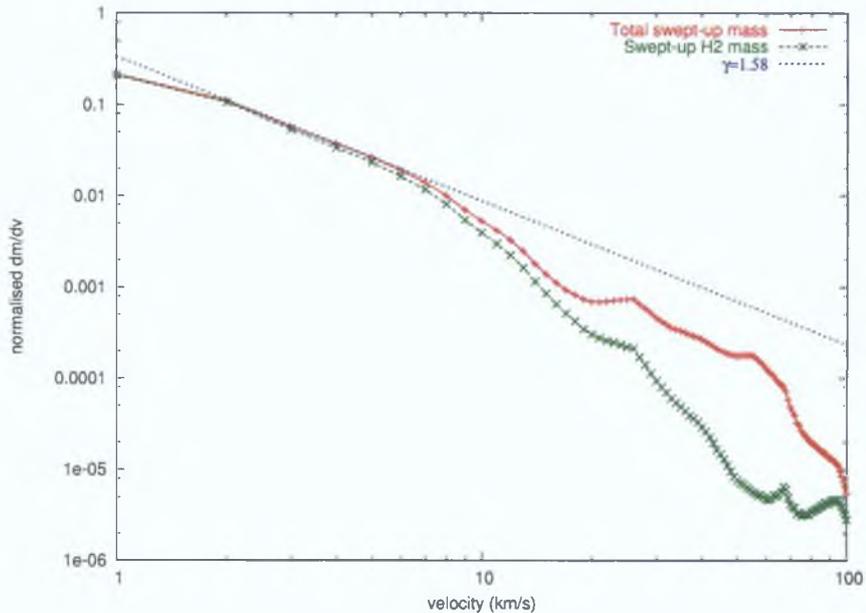


Figure 5.4: Normalised and time-averaged plots for the $m(v)$ and the $m_{\text{H}_2}(v)$ relations for the pulsed jet at 2300 years. The outflow is assumed to be at an angle of 30° to the plane of the sky. A line with a slope of -1.58 is fitted to the low velocity component as a best-fit for the exponent of the relation.

addition, material travelling at these high velocities resides at the head of the bow-shock and the molecules will have been dissociated by passing through the strong shocks in this area.

When we look at the $m_{\text{H}_2}(v)$ relation, we find that it is considerably steeper than the $m(v)$ relation at velocities greater than 20 km s^{-1} . Again, the reason for this is that material at higher velocities has gone through stronger shocks and this leads to dissociation of H_2 .

5.2.3 Intensity–velocity Relationship

Figures 5.5 and 5.6 show plots for the time-averaged, normalised, intensity-velocity relations for CO J=2-1 line, $I_{\text{CO}}(v)$, and for the H_2 S(1) 1-0 line, $I_{\text{H}_2}(v)$, between 1500 and 2300 years. The relations follow a broken power-law in agreement with observations and unlike the $m(v)$ relation. The relation for $I_{\text{H}_2}(v)$ is shallower than

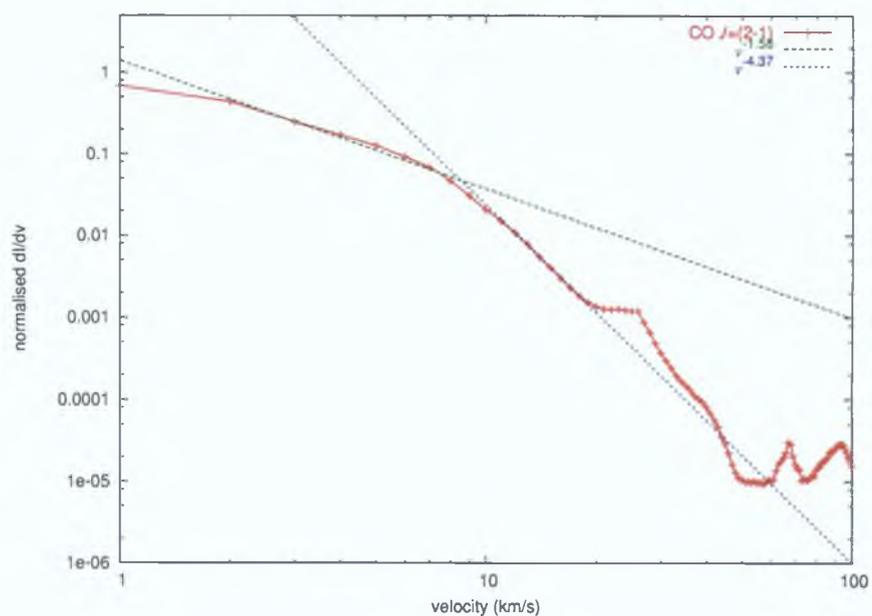


Figure 5.5: Time-averaged intensity-velocity relation for the CO J=2-1 line, $I_{\text{CO}}(v)$, for the pulsed jet.

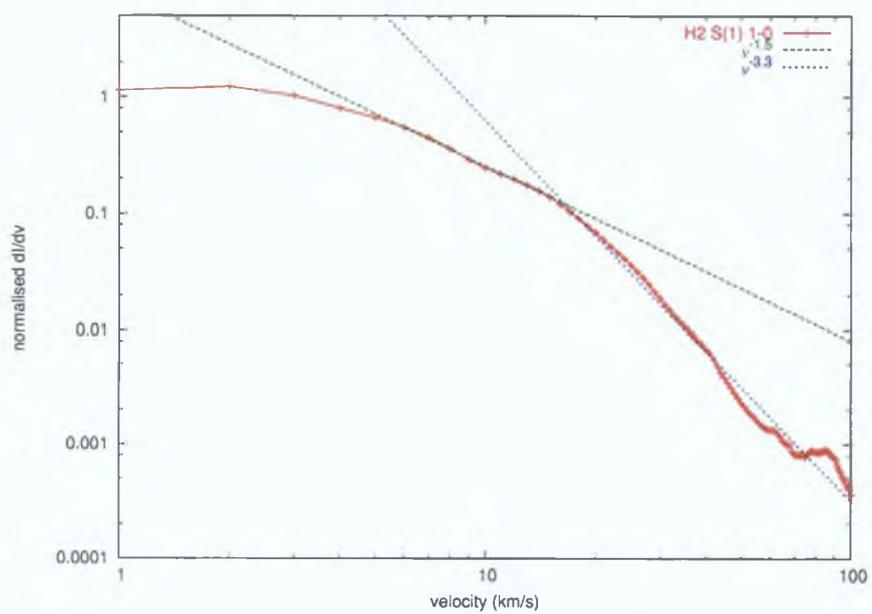


Figure 5.6: Time-averaged intensity-velocity relation for the H₂ S(1) (1-0) line, $I_{\text{H}_2}(v)$, for the pulsed jet.

that of $I_{\text{CO}}(v)$ at all velocities. This is due to H_2 having a higher excitation than the CO line (Downes & Cabrit, 2003). For the CO molecules, at higher temperatures there are higher numbers of energy levels available to the electrons in the molecules to occupy so there are consequently fewer transitions to contribute to the CO J=2-1 emission line. The H_2 S(1) 1-0 line has a higher upper energy level temperature so it has a higher emissivity at the higher jet temperatures. This explains why the slope of this relation is not as steep as that of the CO J=2-1 line. The $I_{\text{H}_2}(v)$ relation does not increase significantly at low velocities unlike the results from Rosen & Smith (2005). However, the results for $I_{\text{H}_2}(v)$ are in qualitative agreement with the observational results (Salas & Cruz-Gonzalez, 2002).

We also note that the results are in good agreement with those of previous simulations, run to shorter time-scales (Smith et al., 1997, Downes & Ray, 1999, Downes & Cabrit, 2003, Rosen & Smith, 2005). This shows that a short simulation (up to ~ 500 -600 years) can be adequate for determining and analysing the characteristics of molecular outflows. This is assuming that there is no precession or episodicity in the outflow.

5.3 The Episodic Jet

With the advantage of being able to run simulations over long time scales, we conducted a simulation of a jet with an episodic nature. We define episodic as meaning a jet with long periods of continuous ejection from the driving source followed by long periods of little or no ejection. Several observed outflows have been seen to exhibit this kind of behaviour (L43, Bence et al., 1998, HH300, Arce & Goodman, 2001b). The initial conditions for this outflow are the same as that for the pulsed jet apart from the input jet velocity. We have set the period for the episodicity to be 800 years, 400 years with a high velocity ejection from the source followed by 400 years with a low velocity ejection. We set the high velocity input, v_h at 215 km s^{-1} and the low

velocity input to be $v_l \sim 30 \text{ km s}^{-1}$. Figure 5.7 shows a log-scale plot of the number density at several stages in the evolution of the episodic jet-driven outflow.

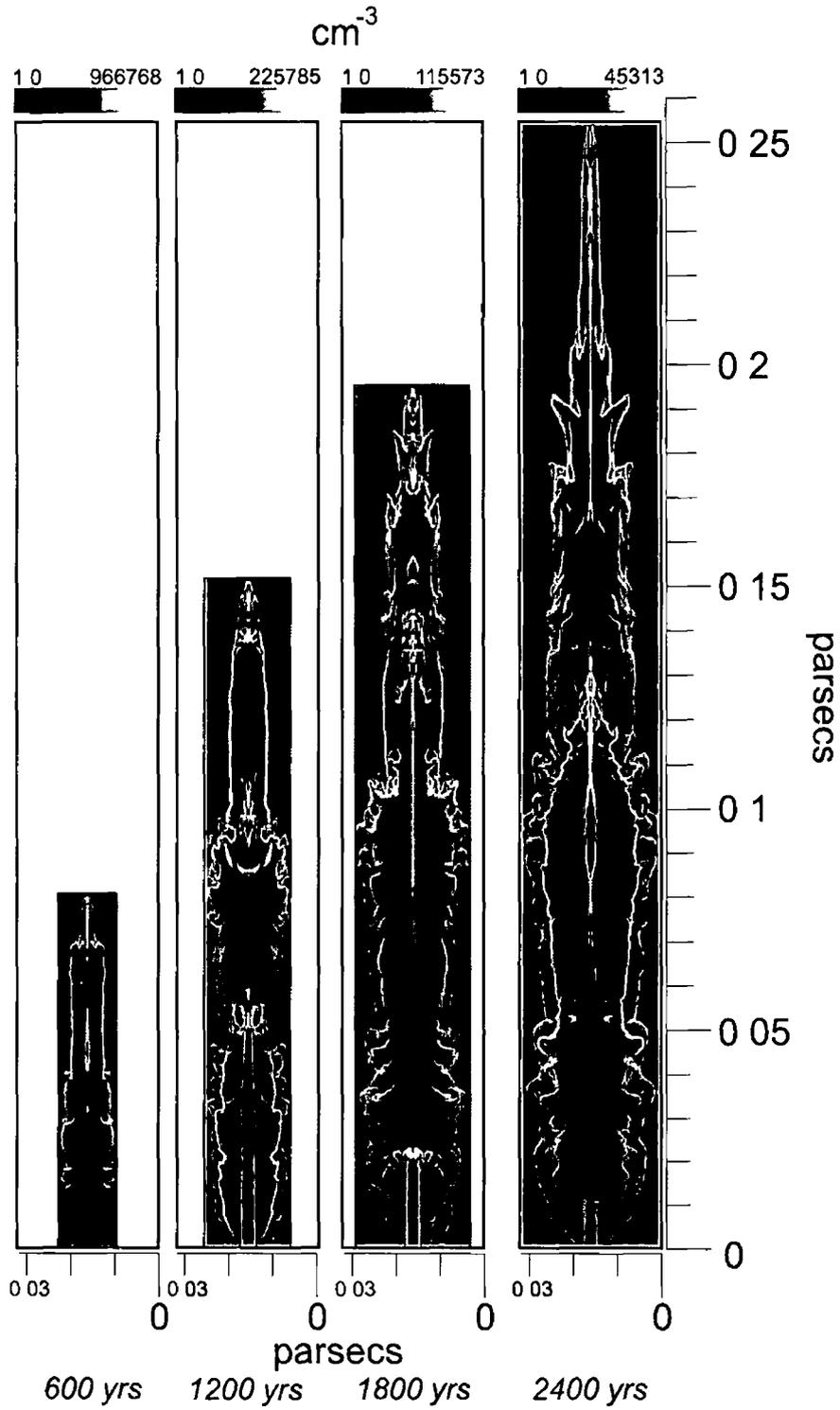


Figure 5.7 Log-scale plot of the number density for the episodic jet at 600, 1200, 1800 and 2400 years. As with the pulsed jet, we see the same corrugation effect in the bow-shock caused by instabilities in the leading bow-shock. Note also that the cocoon of the outflow is broader and less dense than the cocoon in the pulsed jet-driven outflow.

5 3 1 Morphology

The initial ejection episode drives a strong bow-shock into the ambient medium sweeping up a lot of material and only slowing down when it reaches a distance ~ 0.2 pc away from the driving source. The head of the bow-shock consists of extremely dense material relative to its surrounding medium and results in the head of the bow-shock propagating, almost unchecked, through the ambient medium. When the leading bow-shock eventually does slow down it is not long before it is accelerated again by the following episodic ‘burst’.

The collimation ratio, q , for the episodic jet at 2300 years is ~ 8.1 . This is not significantly less than that of the pulsed outflow. Figure 5.8 shows the collimation factor as a function of time for the two simulations. Both outflows experience rapidly increasing collimation for $t < 1000$ yrs. At $t > 1000$ yrs, the leading burst in the episodic outflow begins to slow down as it gets deeper into the ambient medium, resulting in a reduction in the collimation of the episodic jet. This fall-off continues until $t \sim 2100$ years, when the second episodic burst catches the leading burst and injects extra momentum into the head of the bow-shock. The value for q increases rapidly thereafter.

In contrast, the pulsed jet experiences only a minor drop in collimation at $t \sim 1700$ years before increasing again. This is due to the fact that the pulsations in the pulsed jet are continuously impacting on the head of the bow-shock giving it an almost constant velocity throughout the course of the simulation. This difference between the two jets is also reflected in the distances they reach from the driving source by 2300 years. The pulsed jet reaches a distance of ~ 0.32 pc from the source while the episodic jet only reaches a distance of ~ 0.25 pc.

Another difference between the two outflows is the width of the bow-shock. The episodic outflow is $\sim 15\%$ wider than the pulsed outflow and its outflow shell is narrower and denser than that of the pulsed outflow. The reason for this is that the episodic jet ejects more material into the cocoon due to the strong shocks at the head

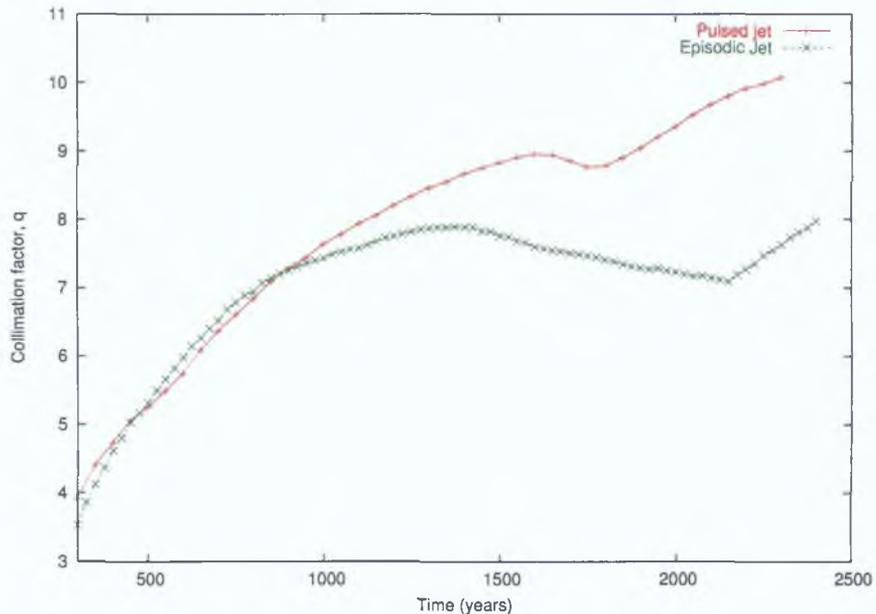


Figure 5.8: Collimation factor, q , as a function of time for the two simulations.

of each ejection burst. As these shocks propagate along the jet beam they continuously eject jet material sideways into the cocoon. This ejected material propagates outwards, away from the jet beam, and hits the inside of the outflow shell which results in an expansion or “inflation” of the shell. This process also contributes to the lower collimation in the episodic outflow when compared with that of the pulsed outflow. The process starts with the introduction of the second mass ejection into the jet at $t \sim 800$ years. From figure 5.8 we can see that this is the time when the collimation factors for the two outflows begin to diverge. The episodic jet-driven outflow collimation factor stops increasing as quickly as the that of the pulsed jet-driven outflow.

5.3.2 Mass-velocity Relationship

Figure 5.9 shows the value for the exponent in the $m(v)$ relation at different times during the course of the episodic and pulsed jet-driven outflow simulations in the velocity range $1-10 \text{ km s}^{-1}$. The trend in the value for γ is similar for both simulations.

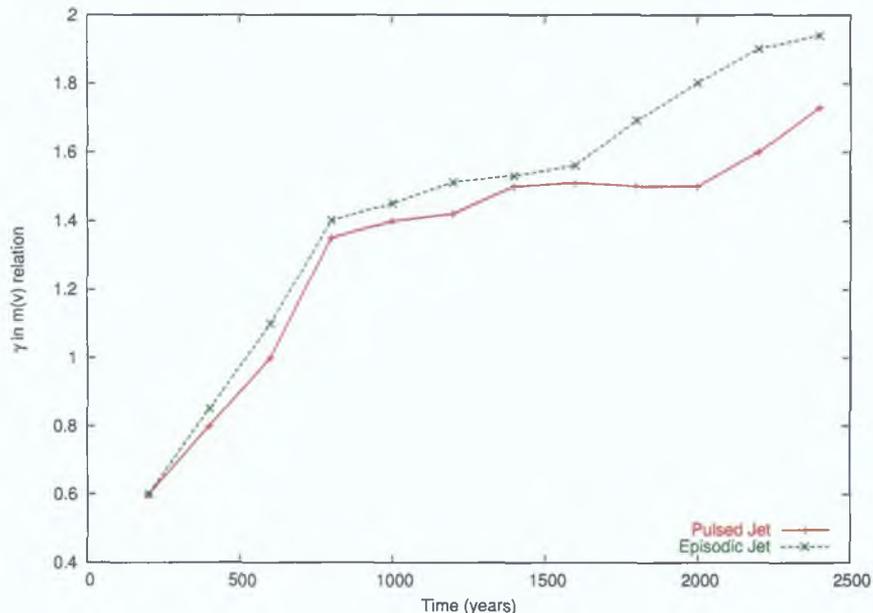


Figure 5.9: The exponent of the $m(v)$ relation, γ as a function of time for both the pulsed and episodic outflows. The values are calculated by fitting over the velocity range $1-10 \text{ km s}^{-1}$.

The value for γ increases rapidly for $t < 1000$ years, followed by a plateau period until $t > 1800$ years, when the value begins to increase again.

In general, we find that the $m(v)$ relation for the total swept-up mass in the episodic jet-driven outflow doesn't differ widely from that of the pulsed jet-driven outflow, particularly at low velocities ($v \leq 30 \text{ km s}^{-1}$). Figure 5.10 shows the time-averaged $m(v)$ relation for both the pulsed and episodic jets. Contrary to the arguments of Arcé & Goodman (2001a), the $m(v)$ relation for the episodic jet is no steeper than that of the pulsed jet. They argued that each ejected burst in an episodic outflow will contribute to an overall $m(v)$ relation with an exponent that is higher than is predicted here. Our simulation has shown that the second and subsequent mass ejections from the driving source have little direct impact on the ambient medium until they reach the head of the outflow (see figure 6.15 in Chapter 6) and, as a result, don't serve to steepen the slope in the $m(v)$ relation.

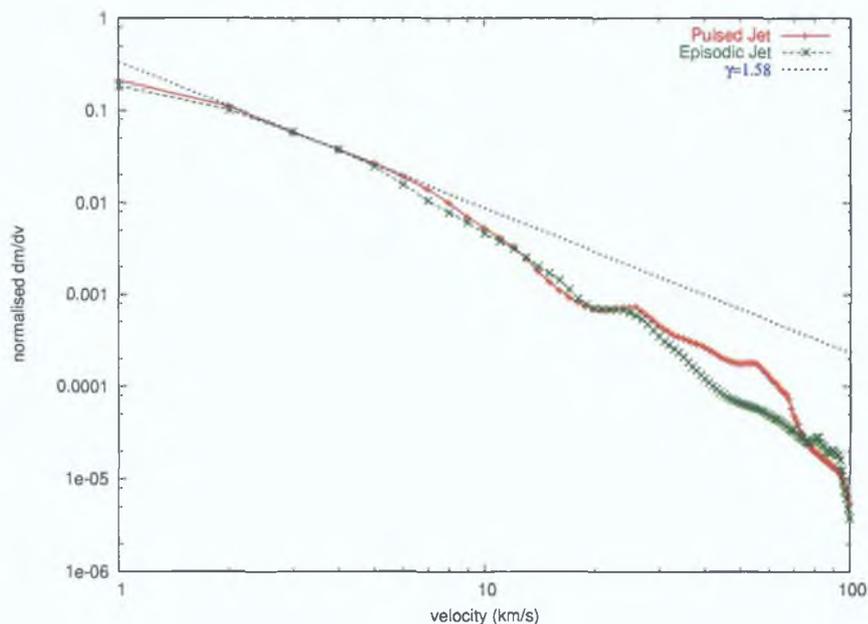


Figure 5.10: The time-averaged normalised $m(v)$ relation for pulsed jet and episodic jet simulations. The average is calculated over $t = 1500$ years and $t = 2300$ years.

5.3.3 Intensity–velocity Relationship

Figures 5.11 and 5.12 show the intensity-velocity relations for the CO J=2-1 line, $I_{\text{CO}}(v)$, and the H₂ S(1) (1-0) line, $I_{\text{H}_2}(v)$, for the episodic jet compared with the same relations for the pulsed jet. As with the $m(v)$ relation, below 20 km s^{-1} the relations for $I_{\text{H}_2}(v)$ and $I_{\text{CO}}(v)$ do not differ much due to similarities in the morphologies of the two outflows. Above 20 km s^{-1} the relations for both lines for the episodic jet can be somewhat steeper than those of the pulsed jet. At 55 km s^{-1} the intensities for both lines are ~ 7 -8 times weaker for the episodic jet. These differences are probably not observable in real systems as they occur at high velocities and low intensities. They arise from slight differences in the bow-shock rather than as a result of the episodicity.

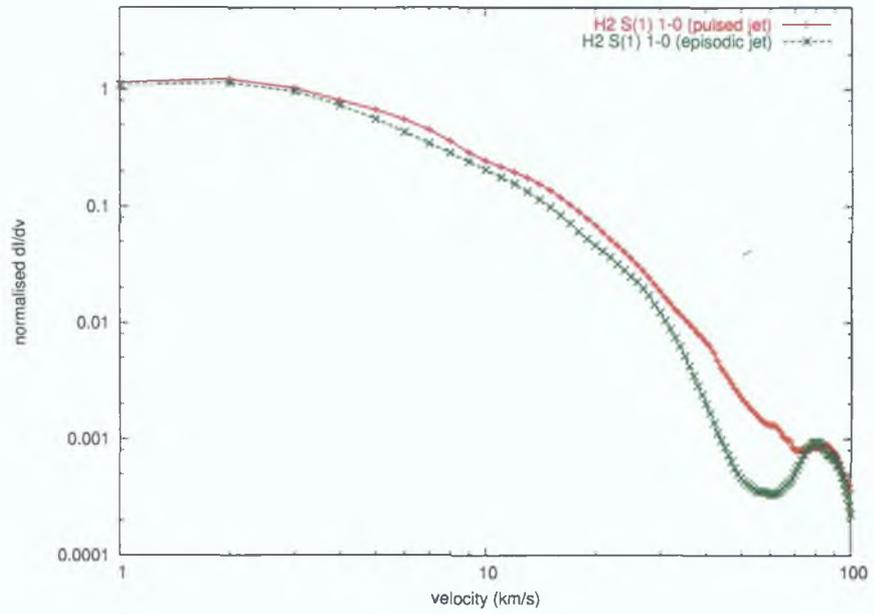


Figure 5.11: The time-averaged $I_{H_2}(v)$ relation for the pulsed and episodic jet-driven outflows.

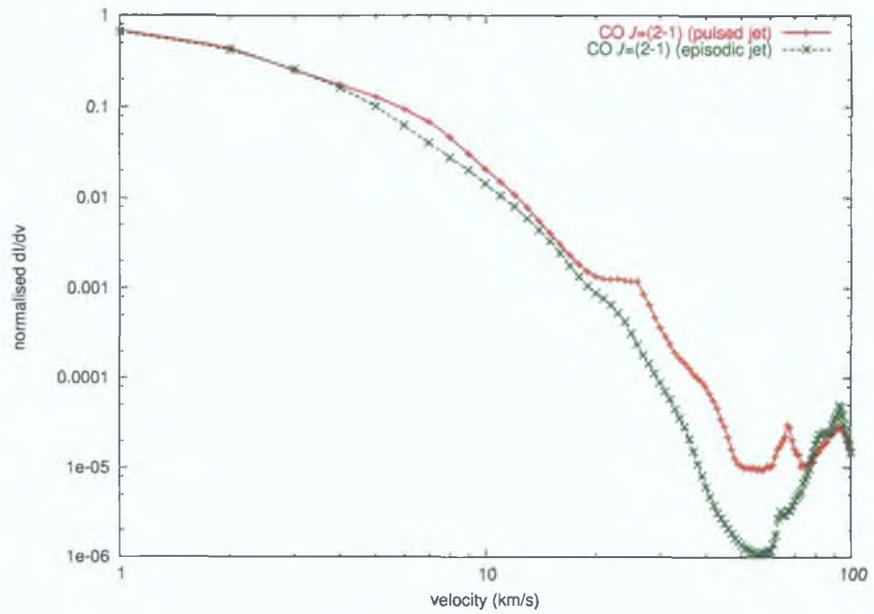


Figure 5.12: The time-averaged $I_{CO}(v)$ relation for the pulsed and episodic jet-driven outflows.

5.4 Behaviour of γ with angle to the plane of the sky

Many authors investigating the jet model (Smith et al , 1997, Downes & Ray, 1999, Lee et al , 2001), have noted that the value for the power-law index γ in the $m(v)$ relation tends to decrease if the outflow is observed at larger and larger angles to the plane of the sky. All of these results were based on outflows that were simulated to relatively short time-scales, the longest being 650 yrs in the case of Lee et al (2001). Figure 5.13 illustrates the behaviour of γ in the $m(v)$ relation at different viewing angles for our long-duration pulsed and episodic outflow simulations. We have found our results to be in agreement with the results of the above mentioned authors.

Previous work using this type of model (Downes & Cabrit, 2003) showed that the position of the break-point velocity, v_{br} , in the $I_{CO}(v)$ relation is dependent on the angle of the observed outflow with respect to the plane of the sky. The smaller the angle the larger the value for v_{br} . Above the break-point velocity, a steepening is observed to occur in the high velocity material. The results from Downes & Cabrit (2003) were based on simulations that were run to ~ 400 years. Figures 5.14 and 5.15 plot the $I_{CO}(v)$ relations for the pulsed and episodic outflows, at various angles to the plane of the sky, at 2300 and 2400 years respectively. We can see that, even for simulations of much longer duration, v_{br} does indeed increase with increasing angle subtended by the outflow to the plane of the sky.

From their observations, Salas & Cruz-Gonzalez (2002), suggested that the viewing angle was not as important as the age of the outflow in determining the position of v_{br} . They observed several outflows in the $2.12 \mu\text{m}$ line of H_2 known to be at different stages in their evolution and found that the value for the break-point velocity decreased with the age of the outflow. Davis et al (1998) observed the same behaviour in CO outflows. Salas & Cruz-Gonzalez (2002) suggested that the reason for this similarity in both molecules may arise from shocked regions within the out-

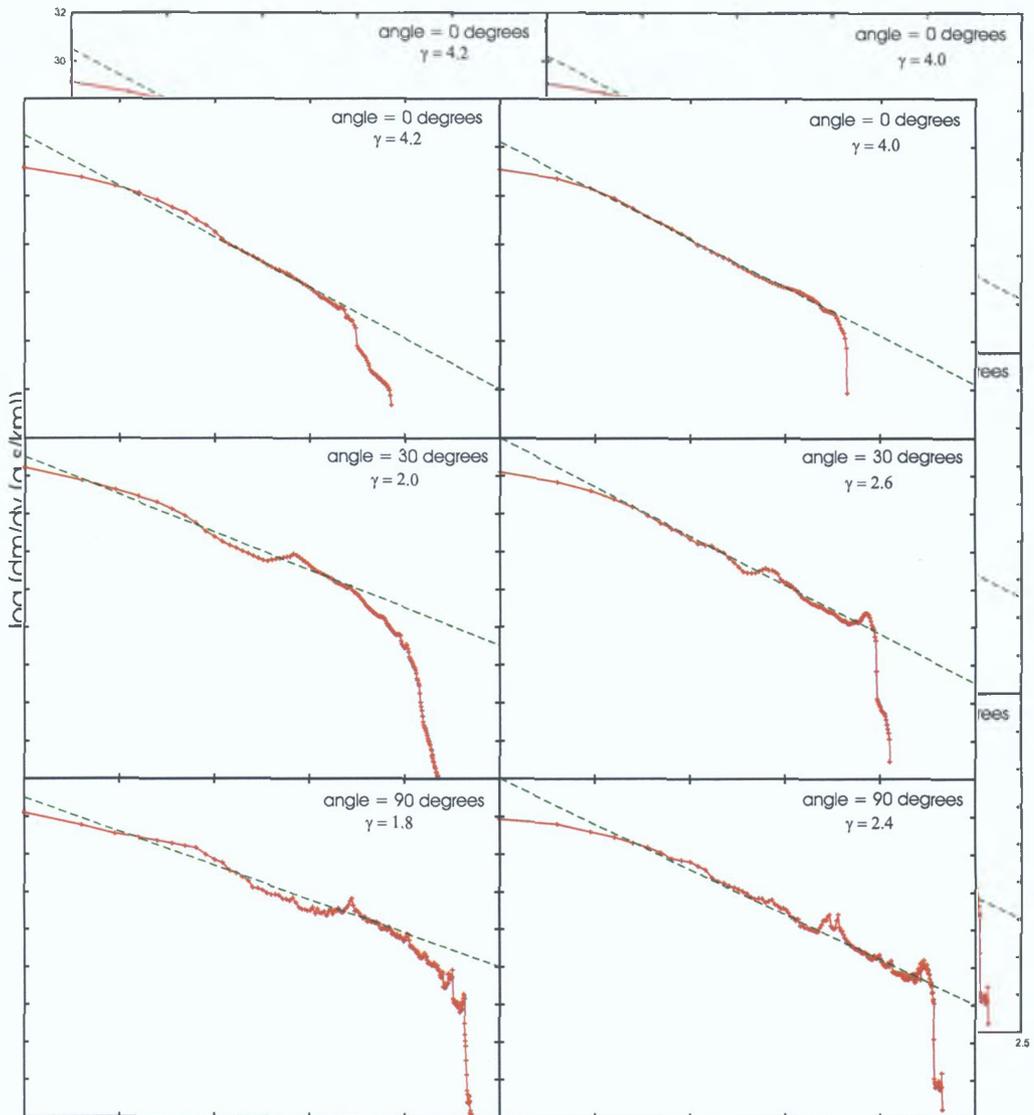


Figure 5.13: Mass-velocity relation for the pulsed jet (left) and episodic jet (right), at 2300 years for different viewing angles to the plane of the sky. Results are presented for angles of 0° , 30° and 90° to the plane of the sky. As with the results from earlier work, the absolute value for the slope of the $m(v)$ relationship, γ , is seen to decrease with increasing viewing angle.

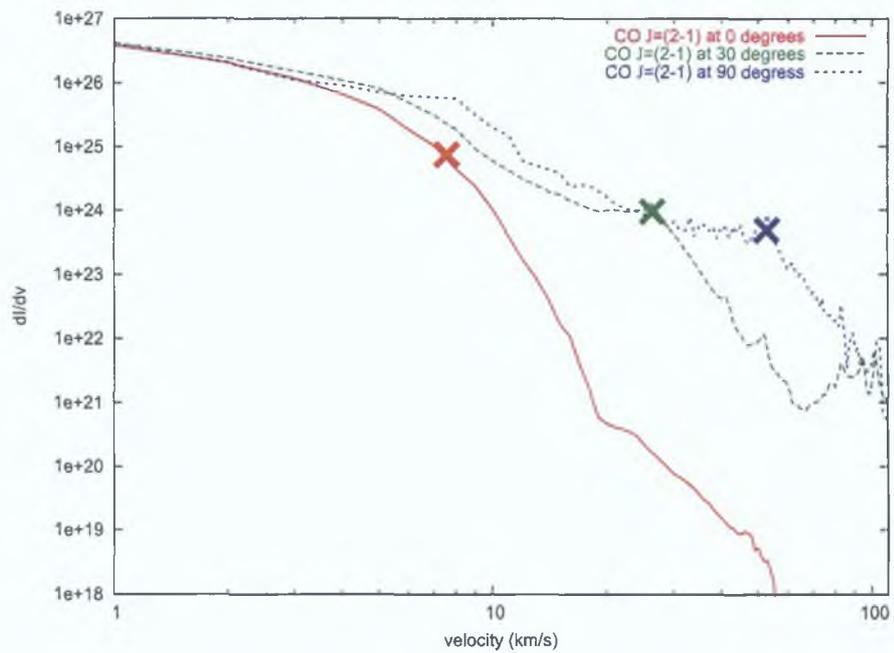


Figure 5.14: Intensity-velocity relation for the CO J=2-1 line, $I_{CO}(v)$, for the pulsed jet at 2300 years for different viewing angles to the plane of the sky. Results are presented for angles of 0° , 30° and 90° to the plane of the sky. The break-point velocity, v_{br} , in the relation, where the slope steepens, is seen to increase with increasing viewing angle. The break-points for each line are marked by an 'x'.

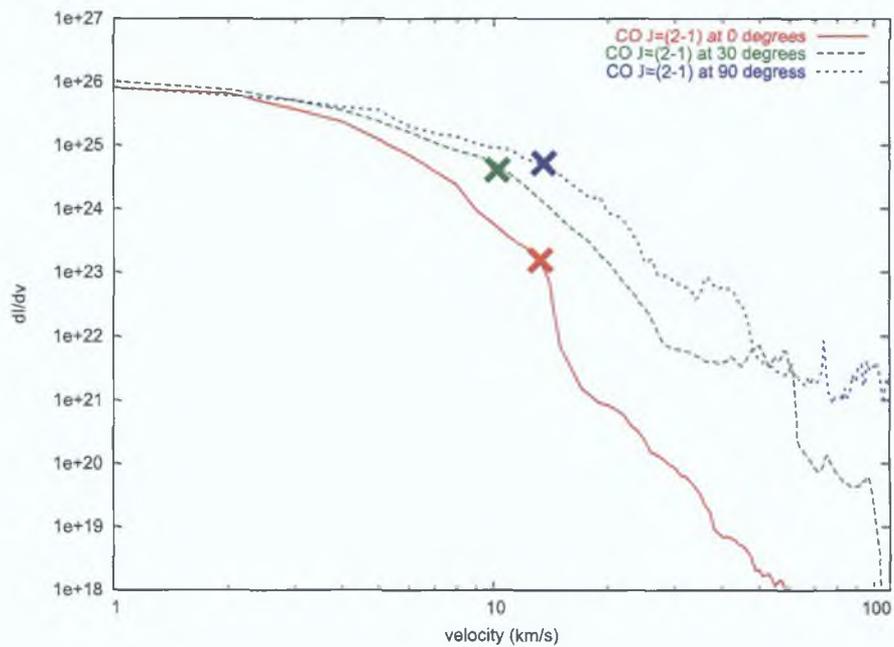


Figure 5.15: Intensity-velocity relation for the CO $J=2-1$ line, $I_{H_2}(v)$, for the episodic jet at 2400 years for different viewing angles to the plane of the sky. Results are presented for angles of 0° , 30° and 90° to the plane of the sky. Here too, at least for 30° and 90° , the break-point velocity, v_{br} , in the relation is seen to increase with increasing viewing angle.

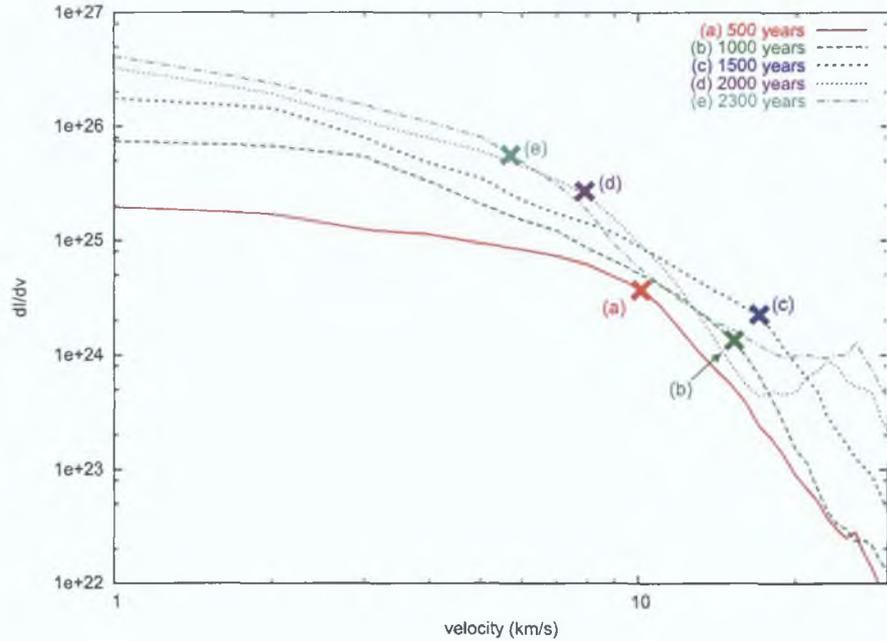


Figure 5.16: CO intensity-velocity relation, $I_{CO}(v)$, for the pulsed jet-driven outflow at 500, 1000, 1500, 2000, & 2300 years for a viewing angle of 30° to the plane of the sky. The break-point velocity, v_{br} , is shown for each of these times.

flow in which both molecules coexist. They also suggested that the steepening in the relations for both molecules at the higher velocities is due to their dissociation by the higher temperatures. Contrary to these findings, Bachiller & Tafalla (2000) found that the break-point velocity, v_{br} , tended to increase with the age of the outflow.

Figures 5.16 and 5.17 show the $I_{CO}(v)$ relations for the pulsed and episodic jets at different times during the course of the two simulations. For $t > 1500$ years, in both outflows, the general trend in the value for the break-point velocity in the relation is that it decreases with time in agreement with the observations of Salas & Cruz-Gonzalez (2002). In the pulsed jet-driven outflow (figure 5.17), for $t < 1500$ years, the trend in the value for v_{br} seems to be more in line with the observations of Bachiller & Tafalla (2000). Clearly, further investigations, both observational and numerical, are required to determine how v_{br} behaves with time.

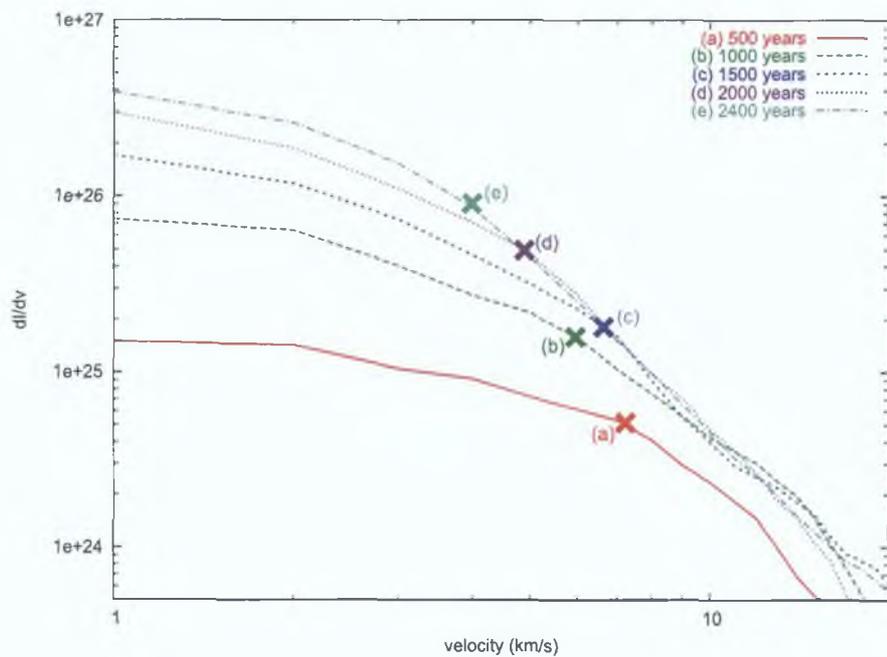


Figure 5.17: CO intensity-velocity relation for the episodic jet at 500, 1000, 1500, 2000, & 2400 years for a viewing angle of 30° to the plane of the sky. The break-point velocity, v_{br} , is shown for each of the times. The trend in v_{br} for this outflow agrees with the findings of Salas & Cruz-Gonzalez (2002).

Chapter 6

An examination of the effects of jet episodicity on molecular outflows

Many observations of molecular outflows from young stars suggest that the driving jets exhibit a variability in their ejection velocity originating in the driving source (HH34, Reipurth et al , 1986, HH111, Reipurth, 1989, HH211, McCaughrean et al , 1994) Raga et al (1990) suggested that knots in HH objects could correspond to “internal working surfaces” within the jet beam, resulting from this variability in the ejection velocity. Several authors have carried out numerical simulations of systems known to show this behaviour by deciphering the past ejection velocity history from observational data (Raga & Noriega-Crespo, 1998, Raga et al , 2002, Raga et al , 2004)

In this chapter we examine how differing input conditions for the jet velocity can affect the resulting outflow. We will look at how this changes the $m(v)$ relation, the $I_{H_2}(v)$ and $I_{CO}(v)$ relations, the collimation of the outflow, the amount of ambient molecular material swept-up by the outflow and how it affects the ratio of red to blue shifted material within the outflow if it were to be observed.

In the previous chapter, one of the simulations we carried out had a sinusoidally varying input jet over a long time period. It is noticeable from the plots of this outflow at large time scales, that the variation in the jet reduces to shocks spaced at 50 year intervals. In this chapter we will also look at what happens to the outflow if we assume these conditions for the input jet velocity from the outset, i.e. a short-period

episodically varying jet

Most calculations done in previous work simulating outflows using the jet model assumed a sinusoidally varying input velocity for the jet. Other possible forms have been explored. De Gouveia dal Pino and collaborators (de Gouveia dal Pino & Benz, 1994) have computed two-step velocity variability with high and low velocity. They found that the resulting jet developed a chain of regularly spaced radiative shocks that were similar in their characteristics to the observed knots in protostellar jets. However, they did not have a very sophisticated treatment for radiative cooling of the gas and could not perform a detailed comparison of their model with observational data. Raga & Noriega-Crespo (1998) simulated the HH34 jet with an input velocity composed of the superposition of three sinusoidal modes which were derived from observational data of HH34. From their simulations they obtained $H\alpha$ intensity maps that were qualitatively very similar to the observations of HH34. As with de Gouveia dal Pino & Benz (1994), they were unable to use high enough resolution to completely resolve the cooling regions.

Cabrit & Raga (2000) also studied the HH34 jet. They attempted to model its slowing-down jet using a model that had a monotonically increasing velocity as a function of time. They found that this could in principle reproduce the observations, but they concluded that the constraints they used to reproduce the observed velocity were deemed unlikely to occur in reality. Raga et al. (1990) and Masciadri et al. (2002) attempted to simulate the past-ejection history of the HH111 outflow with a simulation of a jet with an input velocity which had a sawtooth variability with time (ramps of increasing/decreasing velocity with time followed by a sudden drop/jump in velocity). From their numerical simulations they produced $H\alpha$ images and PV diagrams. In particular, they found that the PV diagrams agreed quite well with the observations.

6.1 Short-period episodically varying jets

To get a clear understanding of the effects of short-period episodically varying jets on the characteristics of a molecular outflow we have run several long-term simulations with slightly different input conditions: three with a simple on-off input jet with periods of 20, 30 and 50 years, and two with the superposition of a short-period on-off component of 50 years and a long-period on-off component with a period of 600 and 800 years. Figure 6.1 illustrates the five input velocity profiles we have used. In each case we use the same “on” and “off” state velocities. For the on state velocity, v_{max} , we have used a value of 215 km s^{-1} and for the off state velocity, v_{min} we have used a value of 30 km s^{-1} . These values are chosen so we can easily compare the characteristics of the resulting outflow with the outflows presented in Chapter 5. When the state changes from on to off or vice versa, the drop or increase in velocity is tapered over the course of a year to allow for a smoother transition and to avoid numerical difficulties arising from an instantaneous, large change in velocity. Apart from the input jet velocity, the initial conditions for the simulations are the same as the initial conditions for the two simulations in Chapter 5 (section 5.1).

6.1.1 Morphology

In figure 6.2 we illustrate the state of each of the five simulations at 1200 years in terms of the number density. In all cases there is a very well-defined excavated cocoon surrounding the jet with a thin, dense shell at the edge of the cocoon. The shell exhibits similar “corrugations” to those that were present in the outflows driven by the pulsed and episodic jets in Chapter 5. Again, these irregularities may be attributed to instabilities at the head of the jet such as the Vishniac instability (Dgani et al. 1996, Downes & Ray, 1999). As noted above, the cocoon region of the outflow appears to be much more prominent in these outflows compared to the simulated outflows of Chapter 5. The reason for this is that the frequent episodic ejections from the driving

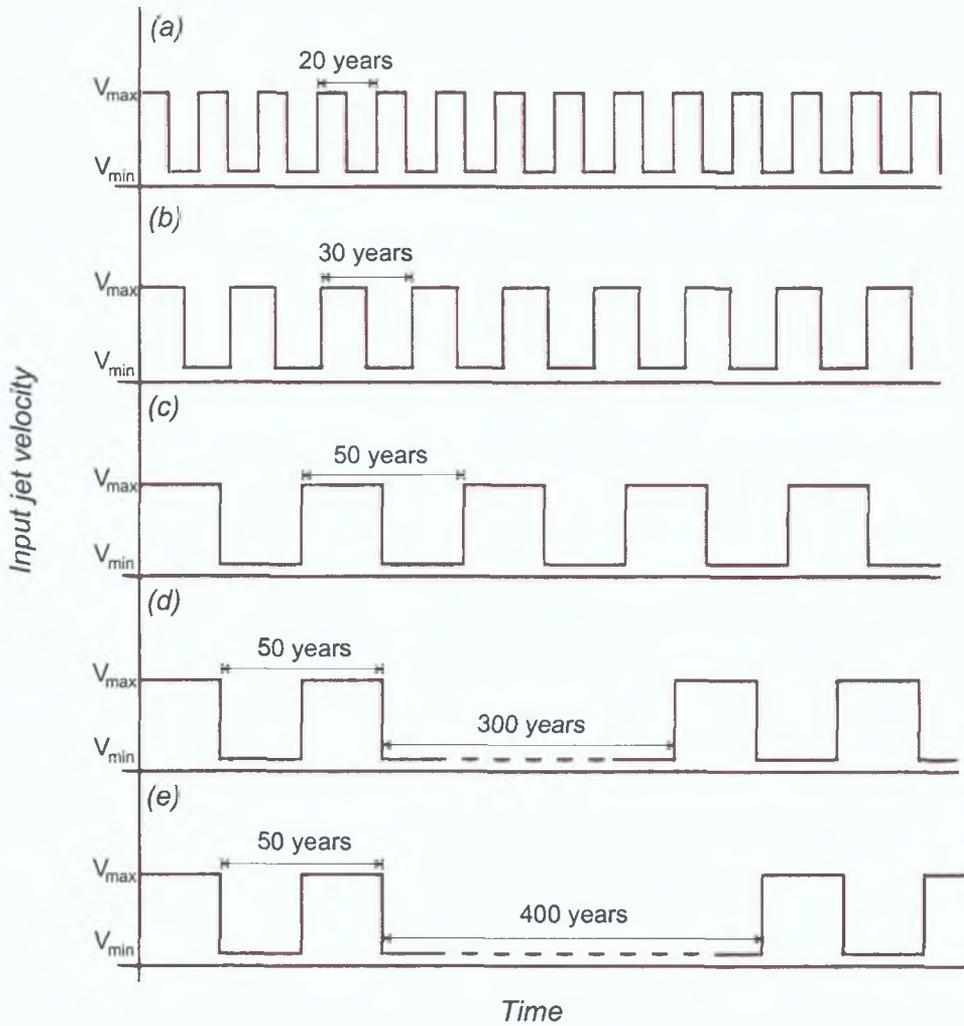


Figure 6.1: Input jet velocity conditions for each of the five simulations. (a), (b) and (c) have a simple “on-off” profile with periods of 20, 30 and 50 years respectively. (d) and (e) have a short on-off component with a period of 50 years and a longer on-off component with periods of 600 and 800 years respectively. In all cases v_{max} is 215 km s^{-1} and v_{min} is 30 km s^{-1} .

source lead to strong shocks forming at frequent intervals along the jet beam. These shocks cause large quantities of jet material to be ejected sideways from the jet beam into the cocoon resulting in an expansion of the outflow shell. In this case, with many shocks along the jet beam, there is a constant force being exerted onto the inside of the outflow shell, steadily expanding it and forming a large cocoon between it and the jet beam. This helps to broaden the overall width of the outflow and reduces its collimation. If we look at the length-to-width ratio or collimation factor, q , (table 6.1) for each of the five simulations we see that the higher the frequency of the mass ejection episodes from the driving source the less well collimated the outflow. This is also reflected in the overall length of the outflow, the higher the frequency of the pulsations, the shorter the outflow. The reason for this is that a greater proportion of the jet's energy is going into broadening the outflow, subsequently, leaving less energy to drive the leading bow-shock forward into the ambient medium. This is clearly evident if we compare simulations (a), (b) and (c) at 1200 years, all having had an equal amount of energy injected into them.

Table 6.1 Comparison of collimation factors and lengths for each of the five simulations at 1200 years. Note that the values for the collimation factor are all substantially lower than the collimation factors for the outflows presented in Chapter 5 and closer to the observed values.

Simulation	Collimation factor at 1200 years	Length at 1200 years (parsecs)
(a)	3.2	0.08
(b)	3.4	0.085
(c)	4.2	0.09
(d)	3.1	0.077
(e)	4.6	0.09

Figure 6.3 shows the behaviour of q as a function of time for each of the five simulations. Beyond $t \sim 700$ years, each one settles into a steady pattern. Above $t \sim 1200$ years, the value for q in all of the outflows, with the exception of (d), reaches a state where it is either constant ((a)) or increasing very slowly ((b), (c) and (e)).

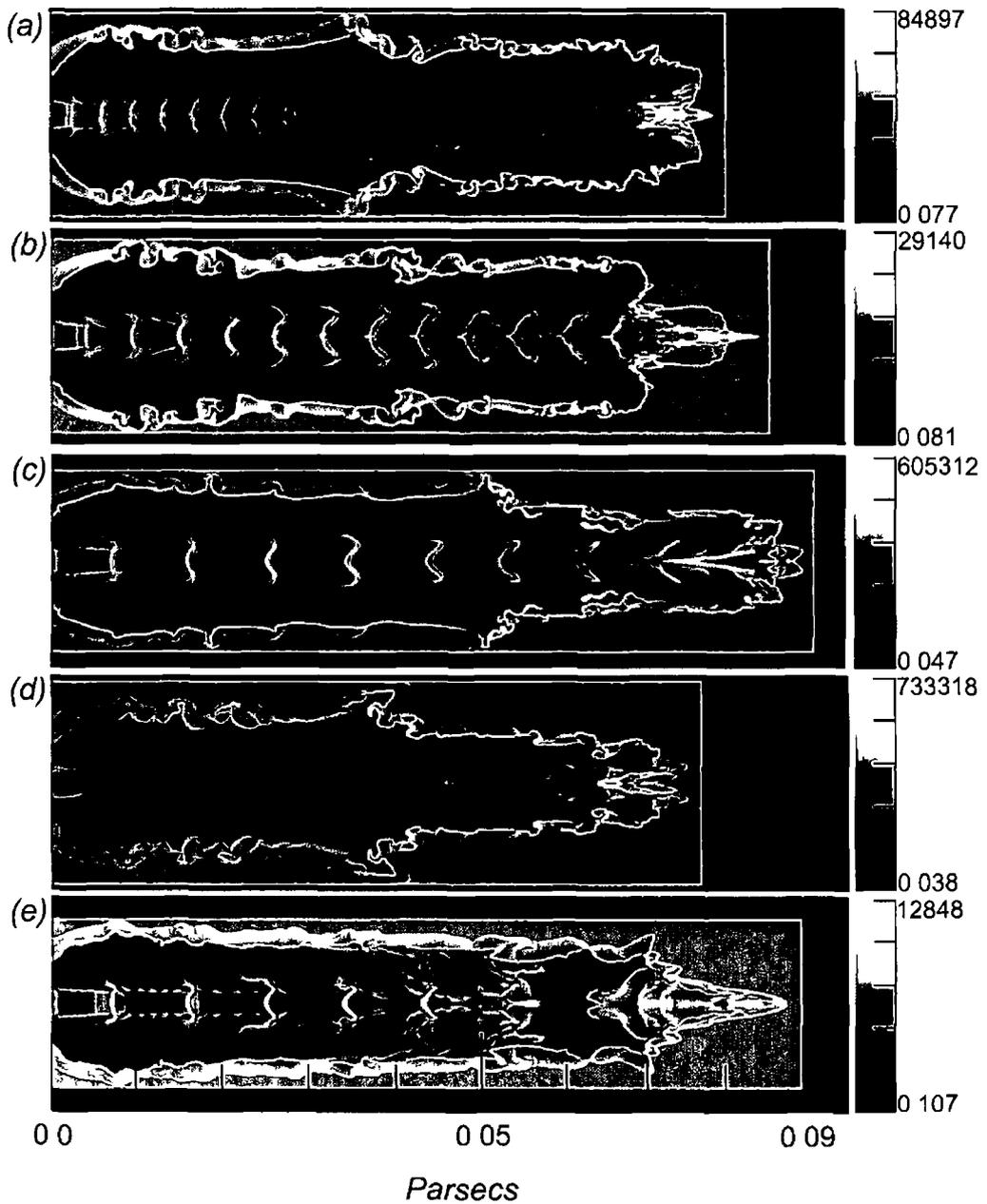


Figure 6.2 Log-scale plots of the number density for several outflows with different input jet velocities at 1200 years (a), (b) and (c) have an on-off input jet with periods of 20, 30 and 50 years respectively (d) and (e) have on-off jets with a period of 50 years and also have a longer period on-off component with periods of 600 and 800 years respectively. In all cases the “on” velocity is 215 km s^{-1} and the “off” velocity is 30 km s^{-1} . The density units are given in cm^{-3} .

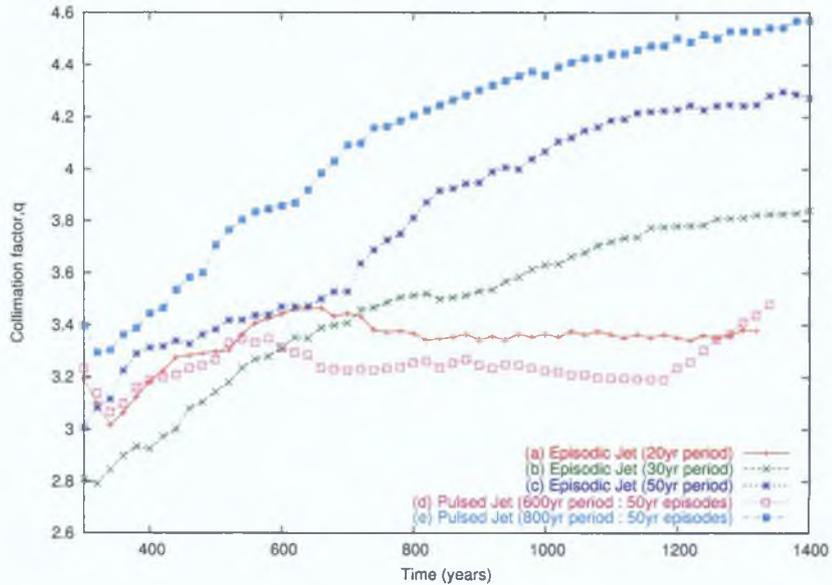


Figure 6.3: Comparison of the collimation factors for each of the five simulations. With the exception of (d), each of the outflows reaches a steady state, (a), or is increasing very slowly ((b), (c) and (e)) in its value for q at 1200 years.

Due to the long periodicity of the mass ejections involved in simulation (d), it has not yet reached a sufficiently mature state at $t = 1200$ years. The maximum value for q is about 4.6 which is a lot less than was found for the long-duration simulations in Chapter 5 and closer to the kind of length-to-width values that are observed in real outflows (Lada, 1985; Wu et al., 2004). As outlined above, this is due to the greater proportion of jet energy going into expanding the outflow shell.

6.1.2 The mass-velocity and intensity-velocity relations

We calculate the mass-velocity and intensity-velocity relations for each of the outflows in the same way as was calculated for the outflows in Chapter 5. Figure 6.4 shows the $m(v)$ relation for the total swept-up mass (atomic and molecular) for each of the five outflows at 1200 years and assuming a viewing angle of 30° to the plane of the sky. Recalling from section 1.1.4 in Chapter 1, it was found that

$$m \propto v^{-\gamma} \quad (6.1)$$

For these outflows, it appears that the value for γ in the $m(v)$ relation does not depend on the variation in the input velocity profile. In the velocity range 1 to 10 km s^{-1} , there is a very small variation in the value for γ (1.6 - 1.8) for each of the five outflows. This value is similar to what we found for the outflows in Chapter 5 and in related earlier work (Downes & Ray, 1999, Downes & Cabrit 2003) and is in agreement with observations.

There is one notable difference in the $m(v)$ relation from that of the outflows presented in Chapter 5. There is a distinct “break” in the slope of the relation between 20 and 30 km s^{-1} for all of the outflows. γ goes from ~ 1.7 to ~ 4.7 . It is not entirely clear why this happens and further work is required to determine the reasons for the break. In section 6.2 we will compare the $m(v)$ relation for a short-period episodic jet-driven outflow with that of the pulsed and long-period episodic jet-driven outflows from Chapter 5 and suggest a possible reason for the break. It is worth noting that the position of this break in the $m(v)$ relation is the same for all of the episodic outflows and is therefore independent of the periodicity of the episodic ejections from the driving source.

The intensity-velocity relations for the CO J=2-1, $I_{\text{CO}}(v)$, and H₂ S(1) 1-0, $I_{\text{H}_2}(v)$, lines for each of the five outflows are presented in figures 6.5 and 6.6. As with the $m(v)$ relation the behaviour of these relations seems to be independent of the input jet velocity profile. The only exception to this is outflow (e) which is the one with the longest period for the input velocity pulsations (800 years) and has not matured enough at this time scale. Both plots exhibit the same distinct break in the relation as was evident in the $m(v)$ relation between 20 and 30 km s^{-1} . The slope of the relation below this break point is almost the same as in the $m(v)$ relation for the $I_{\text{H}_2}(v)$ (~ 1.5), but marginally steeper for the $I_{\text{CO}}(v)$ relation (~ 2.3). The reason for the steeper slope in the CO relation is the lower excitation temperature for this line and is described in section 5.2.3 of Chapter 5. Above the break point the slope of the relations is markedly steeper than that of the $m(v)$ relation. $\gamma \sim 10.3$ for the

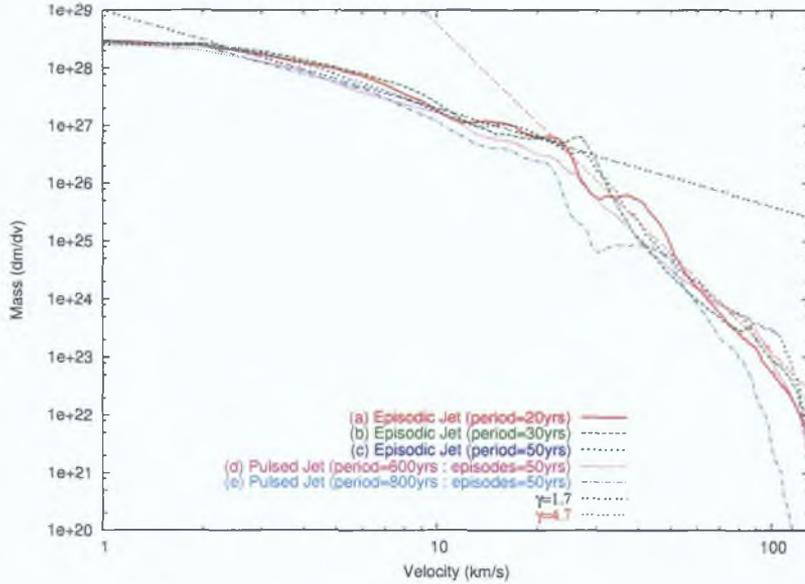


Figure 6.4: Comparison of the mass-velocity relationship for each of the five episodic jet simulations at $t = 1200$ years. We assume that the jet is moving at an angle of 30° to the plane of the sky. The units for dm/dv are in $\text{g s}/\text{km}$.

$I_{\text{CO}}(v)$ relation and ~ 8.8 for the $I_{\text{H}_2}(v)$ relation. There are other factors involved in the diminution of their strength at higher velocities. This is due to dissociation of the molecules as they pass through strong shocks at the higher velocities ($v > 30 \text{ km s}^{-1}$).

6.1.3 Comparison with a sinusoidally varying jet

An alternative method for inserting pulsations into the input jet is to use a sine-wave profile for the velocity. The majority of simulations carried out in previous work have used this type of input velocity profile (Downes & Ray, 1999; Downes & Cabrit, 2003; Smith et al., 1997; Lee et al., 2001). It is interesting to see how an outflow driven by such a jet would compare with the episodically varying jet outflows. To examine this we ran a simulation of an outflow with a sinusoidally varying input jet velocity with a period of 50 years (hereafter referred to as the sinusoidal outflow). Figure 6.7 illustrates the input jet velocity profile for this simulation. We have used the same

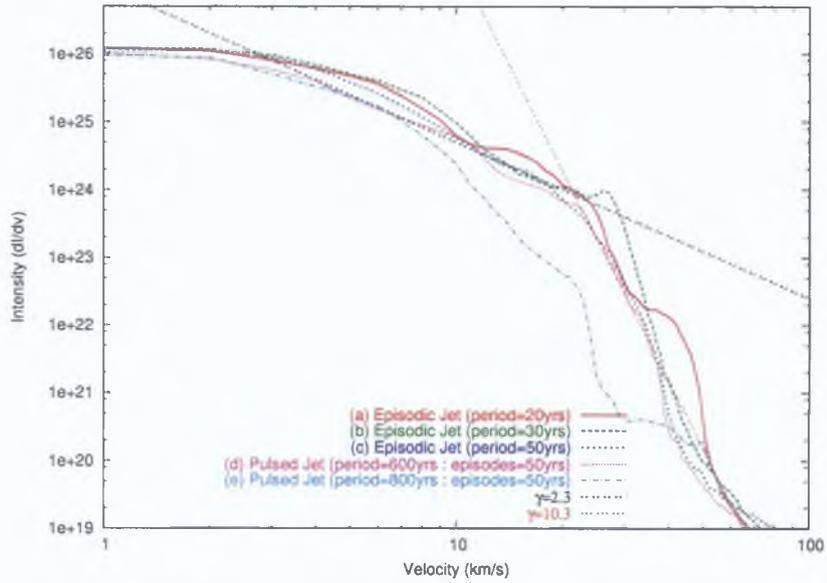


Figure 6.5: Comparison of the intensity-velocity relationship for CO, $I_{CO}(v)$, for each of the simulations. Time $t = 1200$ years and assuming an inclination of 30° to the plane of the sky.

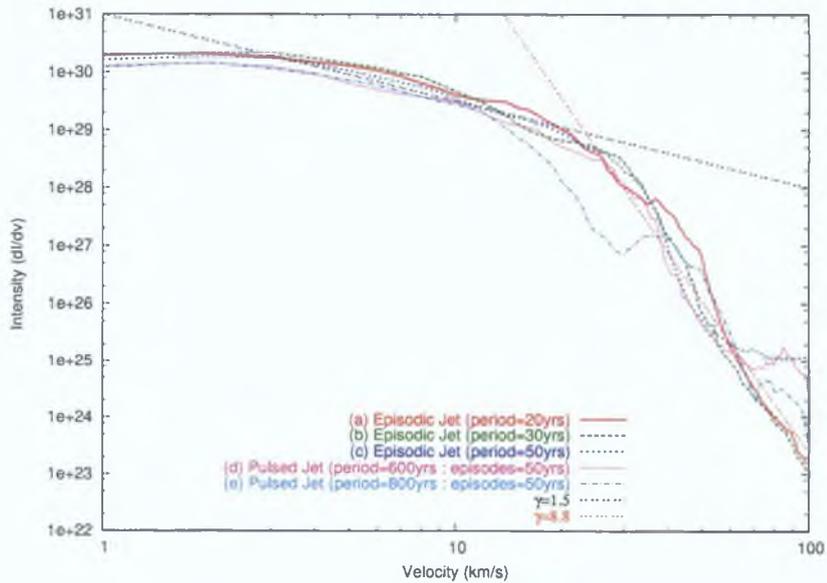


Figure 6.6: Comparison of the intensity-velocity relationship for H₂, $I_{H_2}(v)$, for each of the simulations. Time $t = 1200$ years and assuming an inclination of 30° to the plane of the sky.

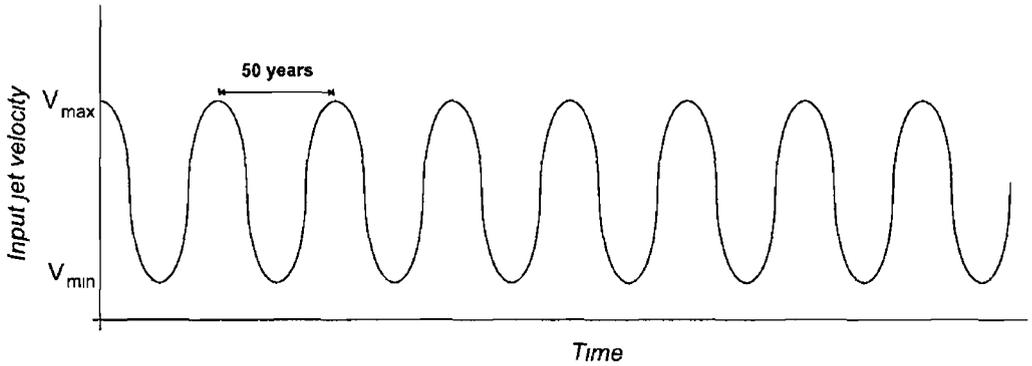


Figure 6.7 Input velocity profile for the sine-wave jet. We have set $v_{max}=215 \text{ km s}^{-1}$ and $v_{min}=30 \text{ km s}^{-1}$, as was done in the five episodic simulations.

values for v_{max} (215 km s^{-1}) and v_{min} (30 km s^{-1}) for this simulation as was used in the five episodic simulations.

We shall only compare this simulation with the episodic jet-driven outflow with a period of 50 years (hereafter referred to as the episodic outflow) as this should provide us with the most insight into the effects on the outflow purely influenced by the difference in the velocity input profile. Figure 6.8 illustrates the state of both outflows at 1000 years in terms of the number density.

Morphological Differences

The sinusoidal outflow is more collimated than the episodically driven one. At $t = 1000$ years, it has a collimation factor of ~ 5 compared to ~ 4 for the episodic outflow. The reason for this is that the shocks in the sinusoidal outflow jet beam are weaker than those in the episodic outflow jet beam. This is due to a more gradual transition from the low velocity mass ejection phase to the high velocity mass ejection phase in the sinusoidal outflow. With weaker shocks, less jet material is ejected into the cocoon and as a result, less jet energy goes into expanding the outflow shell. Figure 6.9 shows the velocity profile along the jet beam for both outflows. Close to the driving source both jets have strong shocks but as the shocks propagate away from the source, the shocks in the sinusoidal jet weaken. The sinusoidal outflow also has

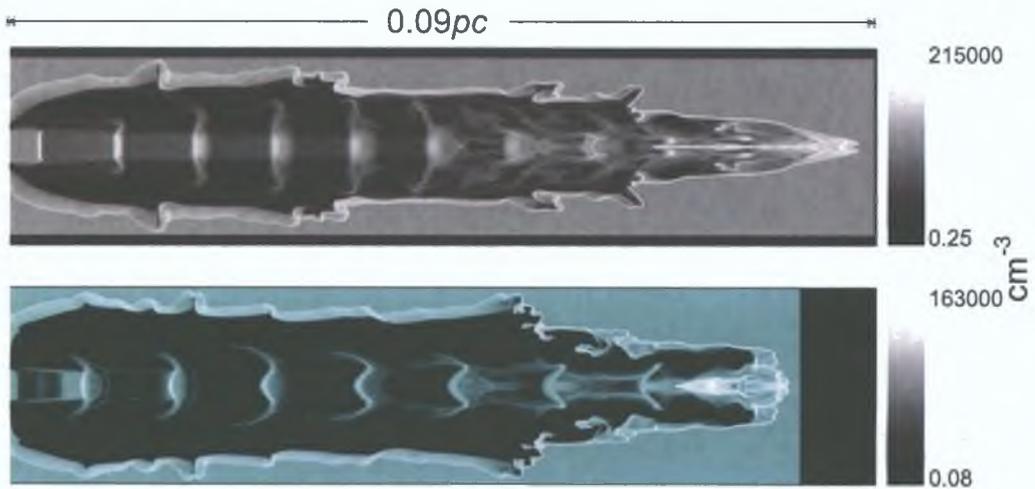


Figure 6.8: Log-scale number density plots for a sine-wave input jet with a period of 50 years, (a), and an episodic jet with a period of 50 years, (b), at a simulation time of 1000 years. The amplitude for the sine-wave input is 90 km s^{-1} , centred on 120 km s^{-1} . The episodic input has an on velocity of 215 km s^{-1} and an off velocity of 30 km s^{-1} . The length scale is in parsecs and the density units are in parts per cm^3 .

a narrower opening angle than the episodic outflow. This is due to the fact that the shocks in the episodic jet form almost instantaneously and start ejecting jet material into the cocoon at the point of the jet input, forcing the outflow shell to broaden in this region. In the sinusoidal jet it is a while after the jet material is injected onto the grid before the shocks form.

6.1.4 Momentum transfer to ambient molecules

Using the same model as we are using here, Downes & Ray (1999) examined the transfer of momentum from the jet to ambient molecular material. They examined jets with steady velocities and multi-mode sinusoidal velocities. For simulations reaching 300 years they found that the mechanism was very inefficient and that only 10 % of the overall momentum is transferred to ambient molecules. Here, we will perform the same calculations on the five episodic outflows and the sinusoidal outflow to see how they compare with each other and with the results of Downes & Ray (1999).

We will use their formula to calculate the fractional momentum transfer to the

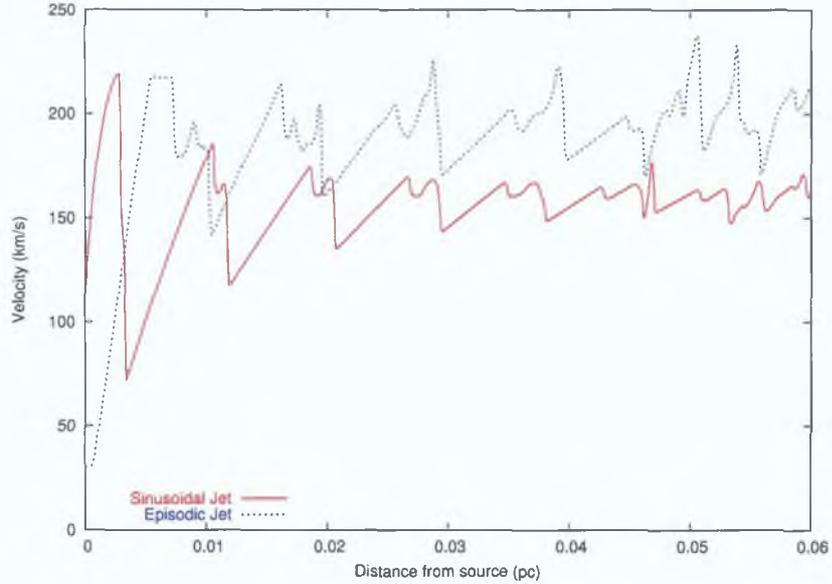


Figure 6.9: Velocity profile along the jet axis for the sinusoidal jet and the episodic jet. The shocks along the episodic jet are stronger than those of the sinusoidal jet.

ambient molecules. It is as follows

$$\mathcal{F}_{\text{H}_2} = \frac{\sum_{ij}(1 - \tau_{ij})u_{ij}m_{\text{H}_2,ij}n_{\text{H}_2,ij}dV_{ij}}{\sum_{ij}u_{ij}n_{ij} \langle m \rangle dV_{ij}} \quad (6.2)$$

where i and j are the cell indices in the z and r directions and dV_{ij} is the volume of cell ij . $n_{\text{H}_2,ij}$ and n_{ij} are the number density of molecular hydrogen and the total number density in cell ij respectively. τ is the jet tracer variable which keeps track of the input jet material and allows us to separate ambient and jet material. m_{H_2} is the mass of a hydrogen molecule, $u_{i,j}$ is the axial velocity in cell ij and $\langle m \rangle$ is the average mass of all the constituent parts of the gas.

Figure 6.10 shows the calculated fraction of the overall momentum in ambient molecules for all of the short-period episodic and the sinusoidal jet-driven outflows up to 1500 years. At 300 years, the fraction is found to be $\sim 10\%$ for all of the outflows, the same as was found in the simulations of Downes & Ray. This indicates that at relatively short timescales the input jet velocity profiles we have used here has no effect on the momentum transfer mechanism. With the benefit of these long-

duration simulations, we can see that the value for the fraction increases with time. The short-period episodic jets, (a) and (b), and the long-period episodic jets, (d) and (e), have a fraction of 30 % or greater by the time they reach 1300 years.

An interesting feature of the long-period episodic jets, (d) and (e), is that they have “peaks” in their plots. The summit of these peaks corresponds to the point where the jets are switched on again. Leading up to these points the jets are in the “off” state and the leading bow-shock is sweeping up more ambient molecules causing a rapid rise in the fraction of momentum in ambient molecules. When the jet is started again, the new momentum being injected into the outflow is only sweeping up jet material within the outflow cocoon and not impacting on ambient molecules. As a result we see a fall in the fractional value for momentum in ambient molecules. This disagrees with the findings of Arce and Goodman (2001a) who postulated that the mass ejection episodes from the driving source, following the leading ejection, should sweep up ambient molecular material.

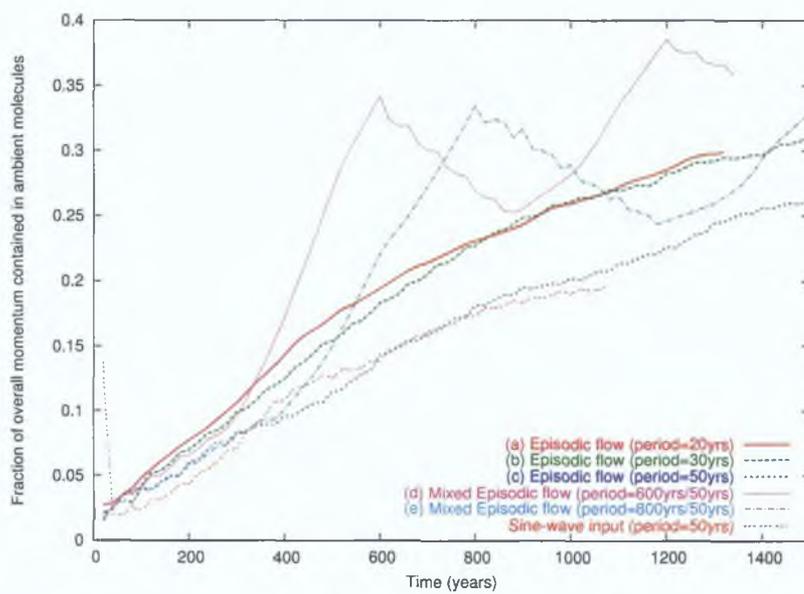


Figure 6.10: Fraction of jet momentum transferred to ambient molecules over the course of the six simulations. The two outflows with long “off” periods, (d) and (e) show peaks in their transfer rates at the end of their “off” periods. This shows that the mass ejections from the driving source do not directly accelerate ambient molecules as is postulated by Arce & Goodman (2001a).

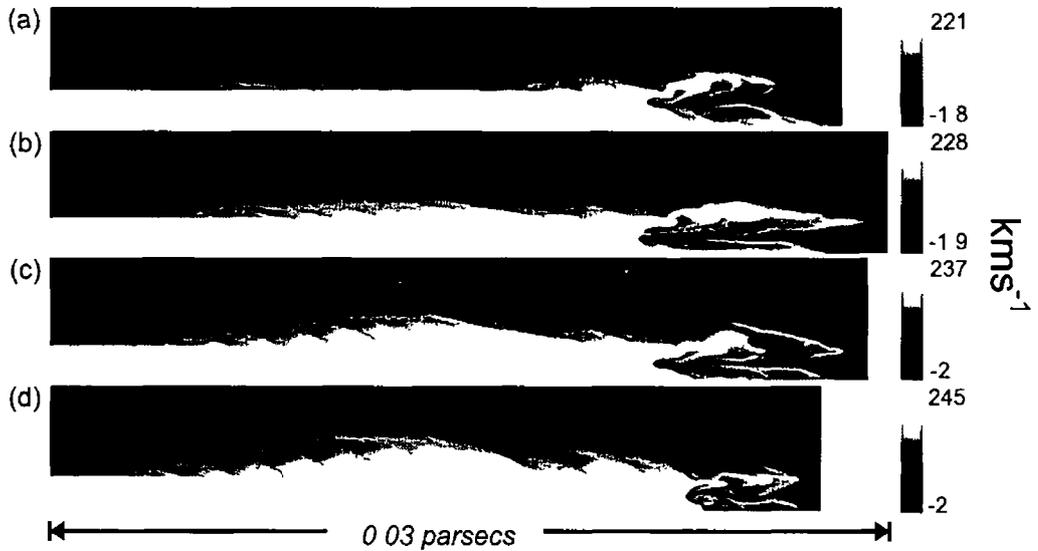


Figure 6.11 Gray-scale plots of the axial-component of the velocity for each of the four velocity variation tests at 275 years. As the amplitude of the variation in input velocity is increased, we see more material ejected from the jet beam shocks into the surrounding cocoon. The value for v_1 in a (a), (b), (c) and (d) are 31, 65, 93 and 127 km s^{-1} respectively.

6.1.5 Pulsed jet and input velocity variation

In the pulsed jet simulations we have used a sinusoidal component to vary the input jet velocity. In order to determine what effect the amplitude of the sinusoidal variation has on the resulting outflow, we have conducted several test simulations (see figure 6.11) where we have varied this amplitude. For the tests we have set the sinusoidal variation in the input velocity to have a period of 20 years with $v_0 = 215 \text{ km s}^{-1}$. We have conducted four tests with $\frac{v_1}{v_0} = 0.75, 0.6, 0.45$ and 0.3 where v_1 is the amplitude variation in the input velocity. This is similar to tests carried out by Downes & Ray (1999). Figure 6.12 illustrates the resulting $m(v)$ relations for the set of tests.

We found that the greater the amplitude of the variation in the input velocity, the stronger the shocks in the jet beam and the more jet material is ejected into the surrounding cocoon. This is similar to what happens in the episodic jets and, as with the episodic jet-driven outflows, there is no effect on the $m(v)$ relation (figure 6.12).

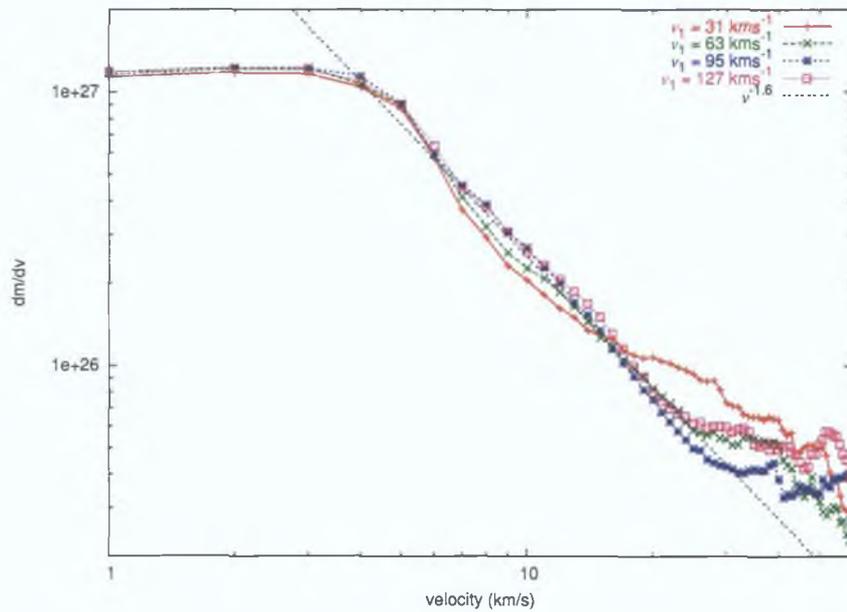


Figure 6.12: Plot of the $m(v)$ relation of the total swept-up mass for several simulations where each one has a different value for v_1 . The pulsations are set to occur every 20 years and $\frac{v_1}{v_0}$ is the extent to which the velocity varies from v_0 . Here $\frac{v_1}{v_0}$ is set to 0.75, 0.6, 0.45 and 0.3. The variation has little impact on the $m(v)$ relation, with $\gamma \sim -1.6$ in each case. These results are taken from jets that reach an age of 275 years. The units for dm/dv are in gs/km .

6.2 Long duration simulations

Figure 6 13 shows the log-scale plot of the number density for the two simulations discussed in Chapter 5 ((a) and (b)), the short-period episodic jet with a period of 50 years, (c), and the short-period episodic jet (50 years) with a longer, on/off, episodic profile with a period of 800 years, (d) The plot illustrates the state of each of the outflows at 2300 years The most striking difference between the short-period episodic outflows and the others is the difference in length In these simulations a greater proportion of the jet energy is going into expanding the outflow shell As a result less energy is going into driving the leading bow-shock forward into the ambient medium This is most obvious if we compare the long-period (800 years) episodic outflow (b) and the short-period (50 years) episodic outflow (c) At 2300 years, both of these outflows ((b) and (c) in figure 6 13) have had an almost equal amount of momentum injected into them (within 5%) We can also see this trend if we look at the collimation factor as a function of time for each of the outflows (figure 6 14) In the long-period episodic jet-driven outflow, (b), the infrequency of the mass ejections from the driving source results in less jet material being ejected into the cocoon and less expansion of the outflow shell as happens with the short period episodic jet-driven outflow, (c) Consequently, more energy goes into driving the leading bow-shock forward in this outflow

In figure 6 15 we see the fraction of momentum contained in ambient molecules (F_{H_2}) as a function of time for each of the outflows The short period episodic outflows are the most efficient at sweeping up ambient molecules Again, we can see the peaks in the relations for the outflows driven by jets with long-period episodicity (see section 6 1 4) For these outflows, when the jet is turned off, F_{H_2} increases more quickly and when the jet is turned on again the value decreases This indicates that the new momentum injected into the jet-outflow system from the switched-on jet does not contribute to the acceleration of ambient molecules The new jet material moves into the already excavated cocoon and does not, directly, interact with the ambient

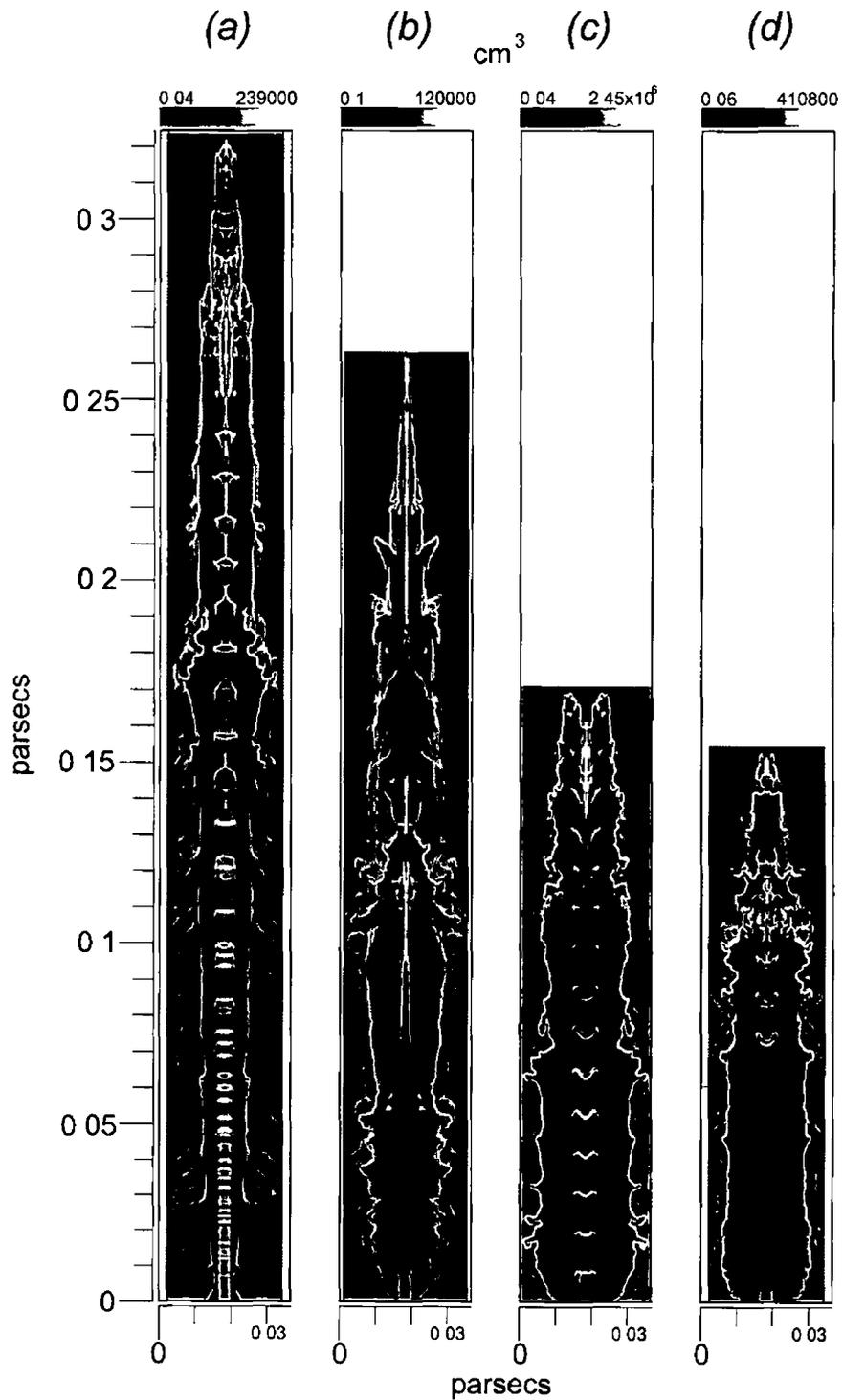


Figure 6.13 Log-scale number density plots for the four simulations at 2300 years. The most striking difference between the two short-period episodic jets ((c) and (d)) and the others ((a) and (b)) is the length difference. The main reason for this is that a greater proportion of the jet energy is going into expanding the outflow shell.

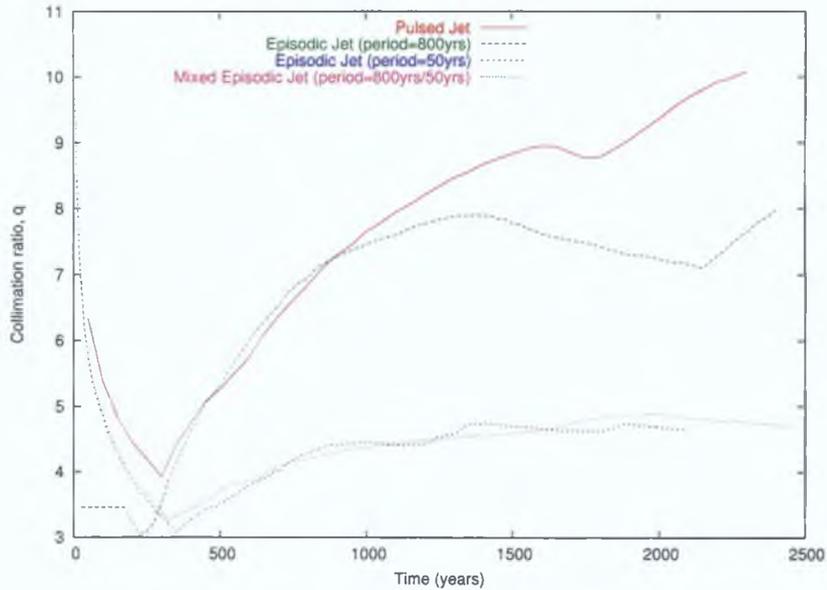


Figure 6.14: Collimation ratio as a function of time for each of the four simulations. The two short period episodic outflows have a constant value for q of between 4 and 5 after the initial conditions of the simulation have been overcome.

medium.

6.2.1 Shell Stripping

One of the reasons the episodic outflow is more efficient than the pulsed jet for accelerating ambient molecules is through a mechanism we have named “shell-stripping”. This involves the acceleration of clumps of material on the inside of the outflow shell by fast moving jet material that has been ejected into the cocoon by the shocks in the jet beam. Figure 6.16 shows an example of this event occurring between 1700 and 1900 years. Ejected jet material propagates with a high velocity at a small angle to the jet axis towards the inside of the outflow shell. It strikes the inside of the shell at an acute angle and in places where the shell is corrugated, drives forward clumps of shell material along the inside of the outflow shell parallel to the jet axis and in the same direction as the jet propagation. As the outflow matures the outflow shell becomes more complicated in its structure with more corrugations forming along it.

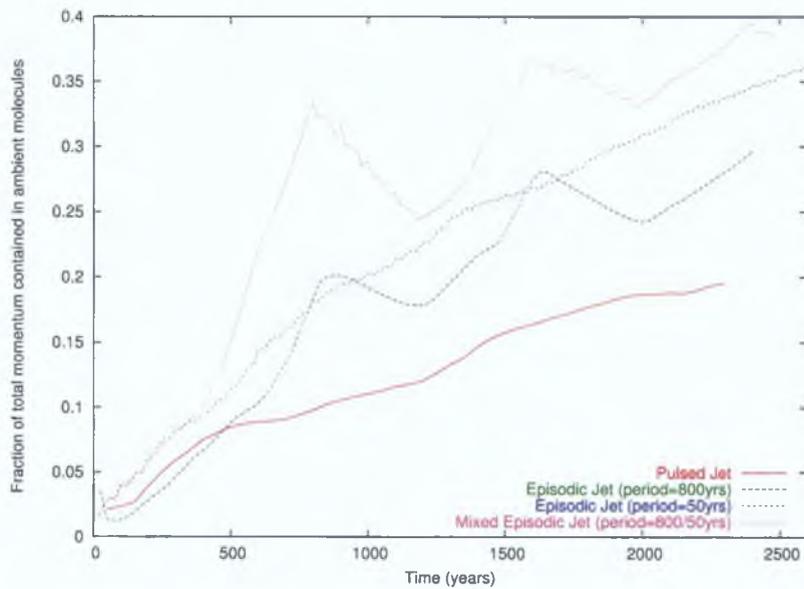


Figure 6.15: Fraction of jet momentum in ambient molecules over the course of each simulation. As the episodic outflows are less collimated than the pulsed outflow, they have a wider bow-shock and are more efficient at sweeping up ambient material. This results in the higher values for the episodic outflows. The long-period outflow has peaks at 800 and 1600 years corresponding to where the jet is switched on. The value for F_{H_2} falls after this point until the jet is switched off again. This implies that the new material being injected into the outflow is not contributing to the sweeping-up of ambient molecules, contradicting the findings of Arcé and Goodman (2001a).

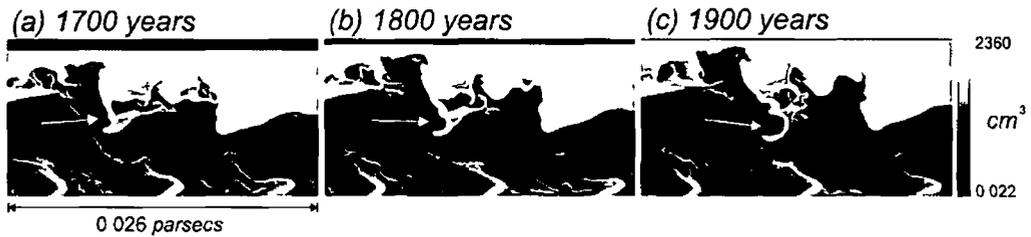


Figure 6 16 An example of “shell-stripping” on the inside of the shell. These images are taken from the short-period episodic jet simulation. This process is found to occur at many places along the inside of the outflow shell and becomes more common with the evolution of the outflow. It is a possible mechanism for the acceleration of ambient molecular material without causing it to dissociate.

As a result, this process is found to occur more frequently. The velocities involved in the process are not high enough to cause strong shocks to occur and there is less likely to be dissociation of the accelerated ambient molecules. Further investigation is required to evaluate how significant this process is in the acceleration of ambient molecules.

6 2 2 Mass–velocity relation

In Chapter 5 we looked at the behaviour of the exponent, γ , in the mass-velocity relation, $m(v)$, for the pulsed and long-period episodic jet-driven outflows over the course of the simulations in the velocity range $1\text{--}10\text{ km s}^{-1}$. We found that the value for γ increased to 1.8 and then remained roughly constant beyond this time. Figure 6 17 shows the same relation for the short-period (50 years) episodic jet-driven outflow. It displays a similar behaviour to the earlier simulations. γ increases rapidly up to $t=1200$ years. Beyond this time, γ remains roughly constant, only increasing marginally from 1.4 to 1.7 between 1200 years and 2700 years. Above 2000 years the value remains constant at 1.7. This is very similar to the behaviour of the relation for the simulations in Chapter 5 and indicates that changing the input velocity profile in the way we have done here does not have an impact on the behaviour of γ .

Figure 6 18 shows a comparison of the $m(v)$ relations for the outflows (a), (b)

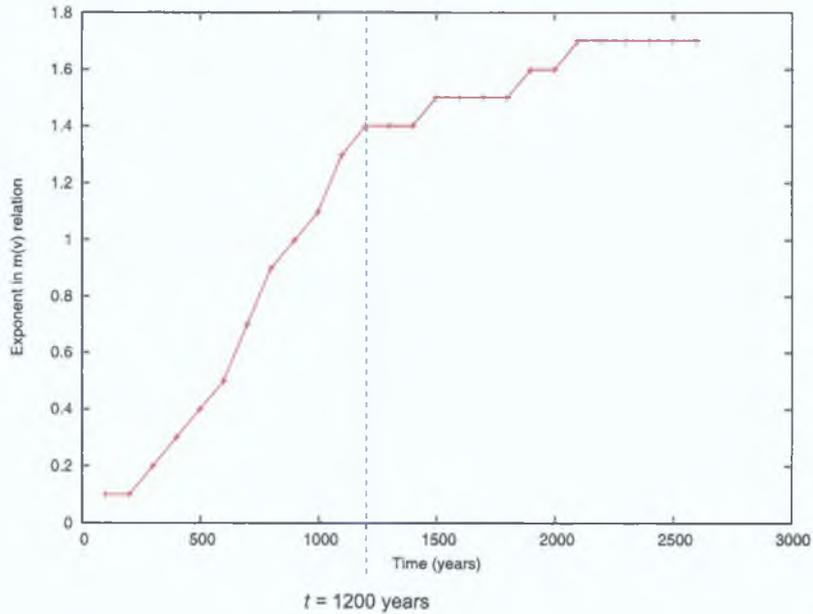


Figure 6.17: Exponent in $m(v)$ relation as a function of time for the outflow driven by the episodic jet with a period of 50 years. The value is calculated for the velocity range $1-10 \text{ km s}^{-1}$. The value increases rapidly until 1200 years. Beyond 1200 years the value remains relatively constant.

and (c) in figure 6.13 at 1000 years. The break in the relation for the short-period episodic jet-driven outflow is clearly visible at $\sim 20 \text{ km s}^{-1}$. The same feature does not appear in either of the other two relations. The reason for this is not clear and requires further investigation but we suggest that it is possibly related to the size of the corrugations that occur along the bow-shock due to the instabilities at the head of the bow-shock.

Figure 6.19 shows grey-scale plots for the log of the number density and the jet tracer variable for each of the three outflows at 1000 years. The jet tracer variable tracks the fraction of injected material in each grid cell. It also allows us to see where swept-up ambient material resides in the outflow. By comparing the density and tracer plots for each of the outflows we can see that more ambient material has penetrated into the outflow shells of the pulsed jet and long-period episodic jet-driven outflows than through the shell of the short-period jet-driven outflow. This,

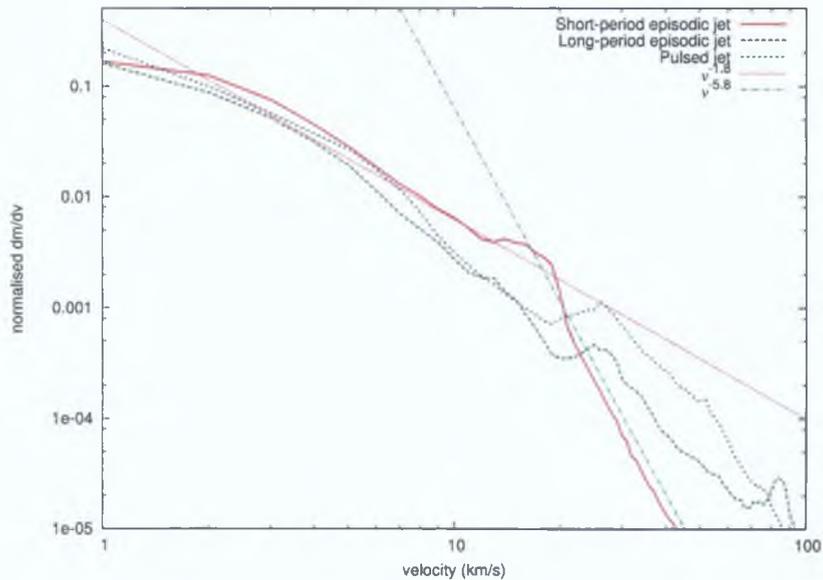


Figure 6.18: Comparison of the mass-velocity relation for the episodic jet-driven outflow with a period of 50 years with that of the long-period episodic and pulsed jet-driven outflows at $t = 1000$ years. The break in the relation for the short-period episodic outflow is not present in the relations of the other two outflows.

we postulate, is due to the formation of larger corrugations in the bow-shocks of these outflows. The gaps between corrugations allows the ambient material to penetrate the bow-shock. Subsequently, this material is close to, or directly in the path of the jet beam and gets accelerated to velocities greater than 20 km s^{-1} . As this process does not happen in to the same extent in the short-period jet-driven outflow there is consequently less ambient material moving at higher velocities ($> 20 \text{ km s}^{-1}$). We suggest that this causes the break in the $m(v)$ relation for this outflow.

6.2.3 Bipolarity

As stated in section 1.1.3 of Chapter 1, another of the key observed characteristics of molecular outflows is their bipolarity. Lada & Fich (1996) observed that the NGC 2264G outflow showed a high degree of bipolarity with a ratio of 20:1 for the amount of blue and red shifted material in each of the outflow lobes. They interpreted this as meaning that most of the material in each lobe was moving in a direction parallel

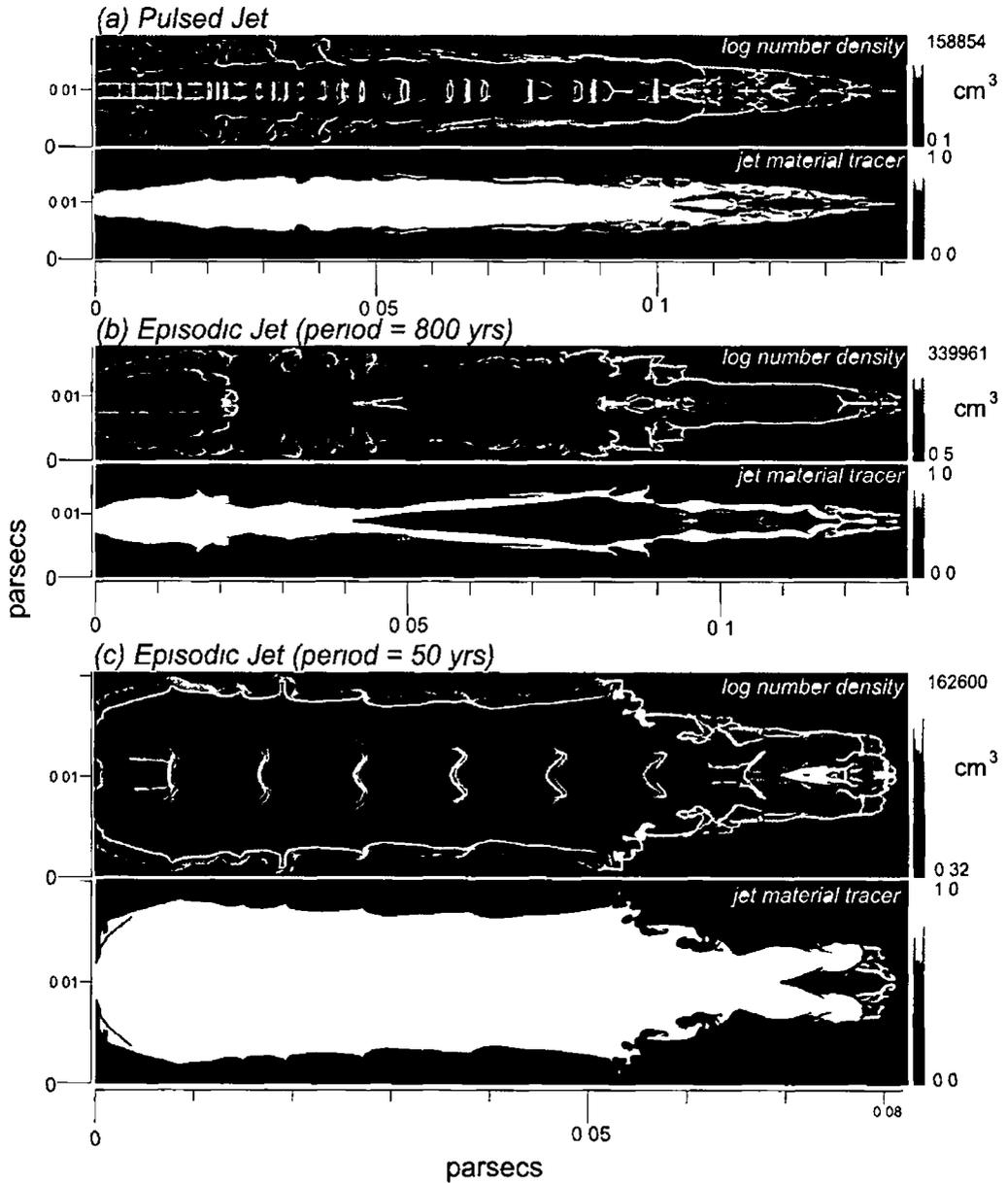


Figure 6.19 Log-scale number density plots and plots of the jet material tracer for the pulsed, long-period episodic and short-period episodic jet-driven outflows at $t = 1000$ years. If we compare each set of images we can see that the level of penetration of ambient material through the bow-shock is lower in the short-period episodic jet-driven outflow. We suggest that this is due to the smaller corrugations that form at the head of this outflow.

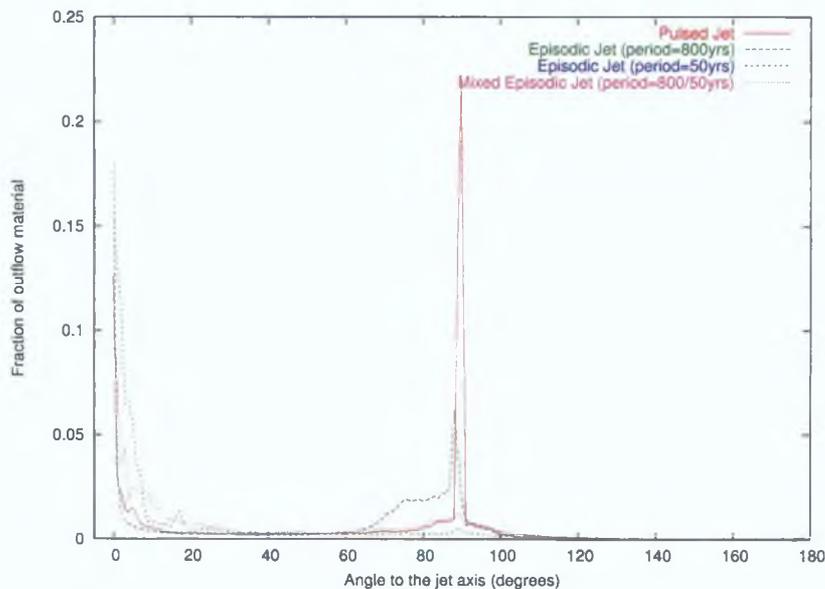


Figure 6.20: Fraction of outflow material with various velocity vector directions relative to the jet-flow axis for each of the four simulations at 2300 years. The outflow driven by the pulsed jet has a large fraction of material moving orthogonal to the jet flow axis. This is due to low jet-outflow shell interaction.

to the jet propagation. We have examined this property for outflows (a), (b), (c) and (d) in figure 6.13 to see if they show the same behaviour. We looked at the fractional amount of material moving at different angles relative to the jet axis in each outflow. Figure 6.20 illustrates the resulting plots at 2300 years. The most noticeable feature in the plot is the large fraction of material moving orthogonal to the jet axis in the pulsed jet-driven outflow. This is due to the lower amount of ejected material from the weaker shocks in the jet beam in this outflow than in each of the episodic outflows. This results in less forward momentum being transferred from the jet to the sides of the outflow shell. Without this forward momentum, the outflow shell material in the pulsed jet-driven outflow propagates orthogonal to the jet axis. From figure 6.20 it is also clear that the material in each of the episodic outflows is predominantly travelling in the same direction as the driving jet and this agrees with the observational findings of Lada & Fich (1996).

Chapter 7

Conclusions

In this chapter will review the work that has been done in this thesis. We will discuss the results for each section and mention possible future developments that could be made to improve upon this work.

7.1 Code

A MUSCL type scheme is used to solve the system of hydrodynamic and chemical equations (Chapter 2). This code was based on the code of Downes & Ray (1999) with a few small modifications. A Riemann solver based on the type described by Falle (1991) is used to calculate the fluxes at the boundaries. A linear solver is used unless the pressures on either side of the cell interface differ by more than 10% in which case a non-linear solver is used. The code also incorporates cylindrical symmetry which allows for a closer approximation to a full three-dimensional outflow than slab symmetry. The problem with cylindrical symmetry is that it can result in over-stability that might not occur in 3-D.

We track the intensities of molecular and atomic hydrogen, along with the ionisation fraction of hydrogen. The CO density is also calculated and is assumed to be a constant (10^{-4}) fraction of the H_2 density.

7.2 Code Parallelisation

The code outlined in Chapter 2 was parallelised for running on distributed memory cluster-type system to allow longer simulations of jets to be run than was previously possible with this code. This was implemented using the MPI libraries (Chapter 3). We used MPI because it is the industry standard for implementing parallel code on a distributed memory machine and allows the code to be portable to many machines. To improve load-balancing and computational performance, a sophisticated parallelisation method was developed which concentrated the combined processing power in the region of the grid occupied by the simulated outflow. We have called this method the “redistributive method”.

We tested the redistributive parallelisation method to see how much of a performance gain we could get from it compared to using the simple parallelisation method (Chapter 4). The theoretical maximum we could expect would be about 4 times faster. In the tests, we achieved a factor of about 3.5 speedup which vindicated the development of the redistributive method. It was also found that the communication overhead of the cell communications became less significant as the outflow got larger and the active computational grid expanded.

Testing was carried out to examine the performance gained by using these methods when compared with the original serial version of the code (Chapter 4). The results showed that the code parallelisation worked extremely well and a sizeable speed-up was gained by using the redistributive method. The overhead incurred by the cell communications between processors became less significant as the simulation progressed and the active computational grid expanded.

However, we found that even with the redistributive method the load balancing was not perfect as some processors would contain regions of the outflow where more cooling and dissociation was taking place resulting in them taking more time to do their calculations than some of the other processors. The calculations would take longer because of the need for more sub-steps to calculate the cooling. Also, at the

cell interfaces where it was used, the iterative routine in the non-linear Riemann solver took more time to calculate the fluxes across cell interfaces

7.3 Long duration simulations

The main reason for parallelising the code was to investigate the long-term behaviour of the mass-velocity relationship, $m(v)$, for molecular outflows. Previous simulations (Smith et al, 1997, Downes & Ray, 1999, Downes & Cabrit, 2003) based on the model we have used here, run to shorter time-scales, had shown good agreement with observational results but it was found that the value for the exponent in the $m(v)$ relation increased with time. As a result, longer term simulations were deemed necessary to investigate whether or not this trend continues. We have run simulations of both pulsed and episodic jets up to 2400 years, a significant fraction of the age of some known outflows. In the episodic jet we have used an on/off period length of 800 years. We found a negative power-law relationship for $m(v)$ of the form $m \propto v^{-\gamma}$ with a value for γ that increased from ~ 1.0 to ~ 1.6 between 0 and 1500 years and became steady at ~ 1.6 beyond this time. We conclude from this that the initial increase corresponds to the outflow simulation overcoming the simplifications in the outflow structure due to the initial conditions. Other than this increase in γ , we found that it did not make any significant difference to the characteristics of the outflow meaning that the earlier work using this model for short-term simulations of outflows could be deemed valid for more evolved outflows.

We also looked at the intensity-velocity relations for the CO $J = 2-1$, $I_{\text{CO}}(v)$, and H₂ S(1) (1-0), $I_{\text{H}_2}(v)$ emission lines. A broken power-law was found for each of these relations. We found that, like the $m(v)$ relation, the value for the slopes in these relations, increased up to $t \sim 1500$ years and became relatively stable beyond this time. For the $I_{\text{CO}}(v)$ relation we found the final value for γ to be ~ -1.6 at lower velocities (1-10 km s⁻¹) and ~ -4.4 at higher velocities (10-100 km s⁻¹). In the

$I_{\text{H}_2}(v)$ relation we found the final value for γ to be ~ -1.5 at lower velocities and ~ -3.3 at higher velocities. Beyond $t = 1500$ years there continued to be fluctuations in the value for γ in the mass-velocity and intensity-velocity relations. This is almost certainly due to instabilities at the head of the bow-shock.

The value for γ was found to decrease with increasing viewing angle to the plane of the sky. We also examined the position of the break-point velocity, v_{br} , in the CO intensity-velocity relations, $I_{\text{CO}}(v)$, for the pulsed and episodic jets. For our simulated outflows with ages greater than 2000 years, the value for v_{br} was found to decrease with increasing viewing angle to the plane of the sky, in agreement with previous, shorter time-scale, results (Smith et al., 1997, Downes & Ray, 1999, Lee et al., 2001). We did not find that the jet model we have adopted conclusively shows an increase or decrease in the value for v_{br} with increasing age of the outflow.

The presence of long-period episodicity in the jet did not significantly alter the behaviour of the mass-velocity or intensity-velocity relations. The reason for this was that apart from the initial mass ejection from the driving source, the ejection episodes in the jet had little direct interaction with the ambient medium until they reached the leading bow-shock. The episodic jet-driven outflow was found to be less collimated than the pulsed jet-driven outflow but not significantly so.

7.4 Input jet velocity variation and its effect on the resulting outflow

We found that outflows driven by jets with a short-period episodicity are less well collimated than sinusoidally jet-driven outflows. A typical collimation factor for this kind of outflow at $t \sim 1200$ years is ~ 4 . This compares with a value of ~ 10 for some of the sinusoidal jet-driven outflows that we simulated in this work. The reason for this is that a large portion of the jet energy goes into widening the outflow leaving less energy to drive the leading bow-shock forward into the ambient medium.

The strong shocks along the jet beam cause jet material to be ejected sideways into the surrounding cocoon resulting in the expansion of the outflow shell. This is a significant result as it is the first time that a simulation of an outflow using the jet model has shown such a low level of collimation. It is true to say that it has, so far, only been found to occur for a very specific jet input velocity profile but the resulting collimation is quite close to what is observed in real outflows.

We found that introducing short-period (~ 50 years) episodicity into a jet results in the introduction of a break in the power-law for the mass-velocity relationship for the swept-up material in the resulting outflow. Further work is required to account for why this occurs. We suggest that it occurs due to the difference in the morphology of the head of the bow-shock in a short-period episodic jet-driven outflow and that of a sinusoidal or long-period episodic jet-driven outflow. The corrugations that form at the head of the latter type of outflow can result in more ambient material propagating through the leading bow-shock. This results in more ambient material travelling at higher velocities ($> 20 \text{ km s}^{-1}$) than is present in the short-period episodic jet-driven outflows.

It was found that mass ejection episodes, other than the initial ejection from the driving source of an episodic jet-driven outflow, does not sweep up additional ambient material as suggested by Arce and Goodman (2001a). The second and subsequent “bursts” of mass are ejected into the jet beam of the outflow and do not directly interact with the ambient medium until they reach the rear of the leading bow-shock where their energy assists the forward movement of the bow-shock into the ambient medium. However, as pointed out above, the strong shocks caused by the episodicity results in the expansion of the outflow shell and the sweeping-up of ambient material. We also found that the fraction of total momentum in ambient molecules increased with the evolution of the outflow. The episodic jet-driven outflows had 30 - 40 % of their momentum in ambient molecules by 2400 years.

Finally, we looked at the bipolarity of our simulated outflows to see did they agree

with the observational results which found that most of the material in an outflow travels in the same direction as the underlying jet propagates. We found that the material in the episodic jet-driven outflows did indeed display this behaviour but the sinusoidal outflow that we examined had a large amount of outflow material moving orthogonally to the jet axis. We suggest that this is due to the weaker shocks in the jet beam of this outflow, resulting in less forward momentum being transferred from the jet to the outflow shell.

7.5 Further Work

7.5.1 Improvements to the code

To improve the underlying code, some of the following improvements should be made:

AMR To further improve the performance of the code Adaptive Mesh Refinement (AMR) could be incorporated. This would allow for the concentration of computational power in regions of the flow where high resolution is needed to resolve the cooling. It would also reduce the need to perform unnecessary calculations where little cooling was taking place. It has been shown in other fields of computational fluid dynamics that this can increase the performance of a code by as much as a factor of 10.

MHD It is commonly believed that magnetic fields play a key role in the formation and propagation of molecular outflows (e.g. O'Sullivan & Ray, 2000). A magnetic field component should be added to the code to investigate this. It would require a more complicated Riemann solver in which the magnetic fluxes across the cell boundaries would need to be determined.

3-D Expansion of the code to 3 dimensions would allow for more realistic simulations of molecular outflows. Cylindrical symmetry can result in over-stability

that might not occur in 3-D. This would also allow for investigations into precession and rotation of the driving source and what effect this might have on the resulting outflow. Rotation of jets has recently been observed by Coffey et al (2004)

7 5 2 Further developments for the parallelisation method

It typically took 3 to 4 weeks to run a long-term simulation on a 32 CPU cluster so even a small improvement in the performance of the parallel code over a single time-step could result in a saving of a few days over the course of the entire simulation. Combined with the constant improvements in the performance of computer hardware, a faster code would make it possible to run even longer simulations capable of achieving the type of time-scales and dimensions that the most evolved outflows that have been observed exhibit. It would also allow for jet-driven outflows to be simulated to a higher resolution allowing for higher densities and better resolution of shocks. Here we suggest some improvements that should be made to the code that do not require large modifications to be made.

Load Balancing To improve the load-balancing in the redistributive parallel method more processing power could be concentrated at the head of the jet where longer time is spent achieving the solution. This could be achieved by the use a dynamic measurement of each processors work load during the course of the simulation which could be used to determine the extent of each processors domain. At set intervals the work load across all processors would be examined and necessary steps would be taken to balance the load. Taking advantage of the shape that outflows take, a straight-forward way to do this would be to move the vertical boundaries between processors so that a processor with a heavy computational load would have fewer cells to work on than a processor with a light load. This would, only involve a small modification to the existing code.

Communications An additional performance gain could be made by the use of persistent communications to do cell communications. Persistent communications is a method for communicating cell values that can take advantage of the fact that the same set of communications is taking place in each time-step. The setting-up process only needs to be done once rather than every time-step, reducing the overhead involved in this process. Although the amount of cells communicated varies from the beginning of the simulation to the end, persistent communications could be set-up for each set of time-steps between each resizing of the grid. As shown in the testing of the parallelisation method, the cell communications occupy a sizeable fraction of the total execution time for each time-step and reducing this fraction should give a significant improvement in the performance of the code.

Check-pointing To aid the debugging and code development process a check-pointing feature should be added to the code. This would be particularly useful for long-term simulations that can take a couple of weeks to perform. Quite often it was found that problems in a simulation might only become apparent after several days. With check-pointing, the complete state of a simulation could be written to disk at set intervals. The simulation could be restarted from any of these states if required. It would require the code to read-in the complete state of a simulation from a file. In a parallel environment there are MPI functions that can get each processor to read-in its portion of the simulation data.

7 5.3 Further investigation of molecular outflows

Further work should also be carried out to investigate the reasons for the break in the mass-velocity relation for short-period episodic jet-driven outflows. Previous work involving sinusoidal jet-driven outflows, using the model we have used, showed no break in the mass-velocity relation (Downes & Ray 1999, Downes & Cabrit, 2003, Keegan & Downes (to appear 2005)). Observational evidence from molecular emission

lines suggests that there may be a break in the relation, making this an important development and worthy of further investigation. The frequency of the episodicity in the mass ejections at which the break begins to appear should be determined and also how the instabilities in the leading bow-shock relate to it.

The shell-stripping phenomenon is another process that has become apparent from the episodic jet-driven outflow simulations. Further effort should be devoted to ascertaining how significant the process is in accelerating ambient molecules. One of the problems with jet-driven outflows is their inefficiency at accelerating ambient molecules. Mechanisms such as this shell-stripping may help to improve the efficiency of the process of accelerating ambient molecules.

It is reasonable to assume that for outflows extending to the length scales that we have simulated here, it is not likely that the ambient medium is uniform, as we have assumed. Simulations should be performed to ascertain what effect a non-uniform ambient medium would have on the resulting outflow. If the ambient density were to fall-off as the jet moves away from its source we might expect that the $m(v)$ relation will become steeper, since the fastest material is that at the head of the bow-shock, and here the density (and hence the amount of mass) drops with time.

Finally, we have just touched on the bipolarity of outflows driven by jets in this work. A more detailed examination should be carried out to understand why the sinusoidal outflows have such a large fraction of material moving orthogonal to the jet axis.

Bibliography

- [1] Amdahl G M , 1967, Validity of the single processor approach to achieving large scale computer capabilities, *Processing AFIPS Spring Joint Computer Conference* 30
- [2] Arce H G , Goodman A A , 2001, *ApJ* 551, L171
- [3] Arce H G , Goodman A A , 2001, *ApJ* 554, 132
- [4] Arce H G , Goodman A A , 2002, *ApJ* 575, 911
- [5] Arce H G , Goodman A A , 2002, *ApJ* 575, 928
- [6] Bachiller R , 1996, *ARA&A* 34, 111
- [7] Bachiller R , Tafalla M , 2000 *Bipolar Molecular Outflows, The Origin of Stars and Planetary Systems, Crete II*
- [8] Bally J , Lada C J , 1983, *ApJ* 265, 824
- [9] Bally J , Reipurth B , Aspin C , 2002, *ApJ* 574, L79
- [10] Bence S J , Padman R , Isaak K G , Weidner M C , Wright G S , 1998, *MNRAS* 299, 965B
- [11] Beuther H , Schilke P , Sridharan T K , Menton K M , Walmsley C M , Wyrowski F , 2002 *A&A* 383, 892
- [12] Blondin J M , Fryxell B A , Konigl, A , 1990, *ApJ* 360, 370
- [13] Bohigas J , Persi P , Tapia M , 1993, *A&A* 267, 168
- [14] Boris J P , Book D L , 1973, *J Comput Phys* 11, 38
- [15] Cabrit S , Guilloteau, S , Andre, P , Bertout, C , Montmerle, T , Schuster, K , 1996, *A&A* 305, 527
- [16] Cabrit S , Raga A , Gueth F , 1997, *IAU Symp 182, Herbig Haro Flows and the birth of Low-Mass Stars* 163

- [17] Cabrit S , Raga A , 2000, *A&A* 354, 667
- [18] Cahill E , 2001, private communication
- [19] Cerncharo J , Reipurth B , 1996, *ApJ* 460, L57
- [20] Churchwell E , 2000 *Massive Star Formation, The Origion of Stars and Planetary Systems, Crete II*
- [21] Coffey D , Bacciotti F , Woitas J , Ray T P , Eisloffel J , 2004, *ApJ* 604, 758
- [22] Cohen M , 1980, *Astron J* 85, 29
- [23] Coquel F , Liou M S , 1995, *NASA TM-106843*
- [24] Cudworth K M , Herbig G H , 1979, *Astron J* 84, 548
- [25] Davis C J , Dent W R F , Matthews H E , Coulson I M , McCaughrean 2000, *MNRAS* 318, 952
- [26] de Gouveia dal Pino, E M , Benz W , 1994, *ApJ* 435, 261
- [27] Devine, D , Bally, J , Reipurth, B , Heathcote, S , 1997, *Astron J* 114, 2095
- [28] Dgani, R , van Buren, D , Noriega-Crespo A , 1996, *ApJ* 461, 927
- [29] Dopita, M A , 1977, *ApJ Supplement S* 33, 437
- [30] Dove J E , Mandy M E , 1986, *ApJ* 311, L93
- [31] Downes T P , 1997, *PhD Thesis, Dublin University*
- [32] Downes T P , Ray T P , 1998, *A&A* 331, 1130
- [33] Downes T P , Ray T P , 1999, *A&A* 345, 977
- [34] Downes T P , Cabrit S , 2003, *A&A* 403, 135
- [35] Edwards S , Snell R L , 1982, *ApJ* 261, 151
- [36] Edwards S , Snell R L , 1983, *ApJ* 270, 605
- [37] Edwards S , Snell R L , 1984, *ApJ* 281, 237
- [38] Falle S A E G , 1991, *MNRAS* 250, 581
- [39] Falle S A E G , Raga A C , 1995, *MNRAS* 272, 785
- [40] Falle S A E G , 2002, *ApJ* 577, 123

- [41] Gingold R A , Monaghan J J , 1977 *MNRAS* 181, 375
- [42] Gueth F , Guilleoteau S , 1999, *A&A* 343, 571
- [43] Haro G , 1952, *ApJ* 115, 572
- [44] Harten A , 1983, *J Comput Phys* 49, 357
- [45] Herbig G H , 1951, *ApJ* 113, 697
- [46] Kwan J , Scoville N , 1976, *ApJ* 210, L39
- [47] Keegan R M , Downes T P , 2005, *A&A*, in press
- [48] Lada C J , 1985, *ARA&A* 23, 267
- [49] Lada C J , Fich M , 1996, *ApJ* 459, 638
- [50] Lee C , Mundy, L G , Reipurth B , Ostriker E C , Stone J M , 2000, *ApJ* 542, 925
- [51] Lee C , Stone J M , Ostriker E C , Mundy, L G , 2001, *ApJ* 557, 429
- [52] Lepp S , Shull M J , 1983, *ApJ* 270, 578
- [53] Lery T , 2002, *RMxAC* 13, 66
- [54] Le Veque R J , 1997, Nonlinear Conservation Laws and Finite Volume Methods for Astrophysical Fluid Flow, *Computational Methods for Astrophysical Fluid Flow*, Springer-Verlag
- [55] Lim A J , Raga A C , Rawlings J M C , Williams D A 2002, *MNRAS*, 335, 817
- [56] Lizano S , Giovanardi C , 1995, *ApJ* 447, 742
- [57] Lucy L B , 1977, *Astron J* 82, 1013
- [58] Masciadri P F , Velazquez P F , Raga A C , Canto, Noriega-Crespo A , 2002, *ApJ* 573, 260
- [59] Masson C R , Chernin L M , 1992, *ApJ* 387, L47
- [60] Masson C R , Chernin L M , 1993, *ApJ* 414, 230
- [61] McCaughrean M J , Rayner J T , Zinnecker H , 1994, *ApJ* 436, L189
- [62] Message Passing Interface Forum MPI A Message-Passing Interface standard 1994, *International Journal of Supercomputing Applications*, 8(3/4), 165

- [63] Message Passing Interface Forum MPI2 A Message-Passing Interface standard 1998, *International Journal of High Performance Computing*, 12(1-2), 1
- [64] Moriarty-Schieven G H , Snell R L , 1988, *ApJ* 332, 364
- [65] Mundt R , Fried J W , 1983, *ApJ* 274, L83
- [66] Nagar N M , Vogel S N , Stone J M , Ostriker E C , 1997, *ApJ* 482, L195
- [67] O'Sullivan S , Ray T P , 2000, *A&A* 363, 355
- [68] Osterbrock D E , 1958, *PASP* 70, 399
- [69] Ostriker E C , 1997, *ApJ* 486, 291
- [70] Raga A , Cabrit S , 1993, *A&A* 278, 267
- [71] Raga A , Canto J , Binette L , Calvet N , 1990, *ApJ* 364, 601
- [72] Raga A , Canto J , Calvet N , Rodriguez L F , Torrelles J M , 1993, *A&A* 276, 539
- [73] Raga A , Noriega-Crespo A , 1998, *ApJ* 116, 2943
- [74] Raga A , Velazquez P F , Canto J , Masciadri E , 2002, *A&A* 395, 647
- [75] Raga A , Riera A , Masciadri E , Beck T , Bohm K H , Binette L , 2004, *ApJ* 127, 1081
- [76] Reipurth B , Bally J , Graham J A , Lane A P , Zealey W J , 1986, *A&A* 164, 51
- [77] Reipurth B , 1989, *Nature* 340, 42
- [78] Richer J S , Shepard D S , Cabrit s , Bachiller R , Churchwell E , 2000, *Protostars and Planets IV*, ed V Mannings, A P Boss, S S Russell (Tuscon Univ Arizona Press), 867
- [79] Rodriguez L F , Ho P T P , Moran J M , 1980, *ApJ* 240, L149
- [80] Rodriguez L F , Carral P , Moran J M , Ho P T P , 1982, *ApJ* 260, 635
- [81] Rosen, A , Smith, M D , 2004, *MNRAS* 347, 1097
- [82] Rosen, A , Smith, M D , 2005, *MNRAS* 347, 579
- [83] Salas L , Cruz-Gonzalez I , 2002, *ApJ* 572, 227
- [84] Sanders R H , Prendergast K H , 1974, *ApJ* 188, 489

- [85] Schwartz R D , 1975, *ApJ* 195, 631
- [86] Shang H , Shu F H , Glassgold A E , 1998, *ApJ*, 493, L91
- [87] Shu, F H , Ruden, S P , Lada, C J , Lizano, S , 1991, *ApJ* 370, L31
- [88] Shu F H , Najita J , Ostriker E C , Shang H , 1995, *ApJ* 455, L155
- [89] Smith M D , Suttner G , Yorke H W , 1997, *A&A* 323, 223
- [90] Snell R L , Loren R B , Plambeck R L , 1980, *ApJ* 239, L17
- [91] Stahler S , 1994, *ApJ* 422, 616
- [92] Steger J L , Warming R F , 1981, *J Comput Phys* 40, 263
- [93] Stone J M , Norman M L , 1993, *ApJ* 80, 753
- [94] Sutherland R S , Dopita M A , 1993, *ApJS* 88, 253
- [95] van Leer B , 1977, *J Comp Phys* 23, 276
- [96] Velusamy T , Langer W D , 1998, *Nature* 392, 685
- [97] Vishniac E T , 1994, *ApJ* 428, 186
- [98] Viti S , Codella C , Benedittini M , Bachiller T , 2004, *MNRAS* 350, 1029V
- [99] Woitas J , Ray T P , Bacciotti F , Davis C J , Eisloffel J , 2002, *ApJ* 580, 336
- [100] Woodward P R , Colella P , 1984, *J Comp Phys* 54, 115
- [101] Wu Y , Haung M , He J , 1996, *A&ASS* 115, 283
- [102] Wu Y , Wei Y , Zhao M , Shi Y , Yu W , Qin S , Huang M , 2004, *A&A* 426, 505
- [103] Zhang Q , Zheng X , 1997, *ApJ* 474, 719
- [104] Zinnecker H , McCaughrean M J , Rayner J T , 1998, *Nature* 394, 892
- [105] Zuckerman B , 1976, *ApJ* 209, L137