

Numerical Solutions of the General Relativistic Equations for Black Hole Fluid Dynamics

Philip Blakely

Selwyn College

University of Cambridge

This dissertation is submitted to the

University of Cambridge

for the degree of

Doctor of Philosophy

August 2009

This thesis is based on research carried out under the supervision of Dr Nikolaos Nikiforakis at the Department of Applied Mathematics and Theoretical Physics at the University of Cambridge from October 2005. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. Further, all computer code referred to herein was written by me unless otherwise stated.

No part of this dissertation has been submitted for any other qualification.

Abstract

The aims of this thesis are to develop and validate a robust and efficient algorithm for the numerical solution of the equations of General Relativistic Hydrodynamics, to implement the algorithm in a computationally efficient manner, and to apply the resulting computer code to the problem of perturbed Bondi-Hoyle-Lyttleton accretion onto a Kerr black hole. The algorithm will also be designed to evolve the space-time metric, and standardised tests will be applied to this aspect of the algorithm.

The algorithm will use up-to-date High-Resolution Shock-Capturing numerical schemes that have been developed for the stable and accurate solution of complex systems of equations. It will be built around the Adaptive Mesh Refinement and overlapping, curvilinear grid methodologies in order to extend these schemes to the efficient solution of two and three-dimensional problems. When implementing the algorithm, we will use previously written code libraries, where appropriate, to avoid excessive software development.

We will validate the algorithm against standard test-cases for Special and General Relativistic Hydrodynamics, and for Einstein's equations for the evolution of the space-time metric. The methodologies we use will be tested to ensure that they lead to the stable and accurate numerical solution of these problems.

Finally, the implemented algorithm will be applied to the problem of Bondi-Hoyle-Lyttleton flow onto a Kerr black hole in three dimensions. It will be validated against existing exact and numerical solutions of the problem, and then be used to perform an extensive parametric study of the problem, varying the spin of the black hole and the incident wind direction, and allowing for the perturbation of the fluid density upstream of the black hole.

We will then analyze the results of the study, and present the complete set of results on a DVD accompanying this thesis.

Contents

1	Introduction	13
1.1	Conventions	14
2	The State of Numerical Relativity	15
2.1	Research Areas in Numerical Relativity	15
2.2	Foundations of Numerical Relativity	17
2.2.1	Gauge freedom	18
2.2.2	Constraints	18
2.2.3	Formulations of Einstein's equations	19
2.2.4	Initial data	20
2.3	Current state of Numerical Relativity	21
2.3.1	Cactus and associated thorns	21
2.3.2	Cornell and CalTech code	23
2.3.3	Wind Accretion by Font, Ibáñez, and Papadopoulos	24
2.3.4	Illinois GRHD code	24
2.3.5	Nada code	24
2.3.6	SACRA code	25
2.3.7	Brigham Young University	25
2.3.8	WHAM: WENO-based GRMHD code	25
2.3.9	COSMOS++	26
2.3.10	Zanotti and Dumbser	26
2.3.11	Code review conclusions	26
2.4	Validating a Numerical Relativistic Code	27
2.5	Aims of this thesis	28

3	System Equations and Metrics	30
3.1	Special Relativistic Hydrodynamics	30
3.2	General Relativistic Hydrodynamics	31
3.3	Fluid types	32
3.3.1	Stiff fluid	32
3.3.2	Ideal fluid	33
3.4	Medium system	36
3.5	Bona-Massó formulation	38
3.6	Fluid validation tests	40
3.6.1	One-dimensional Riemann problems	40
3.6.2	Two-dimensional Riemann problem	43
3.7	Propagating Gauge Pulse	44
3.8	Apples with Apples tests	44
3.8.1	Robustness test	44
3.8.2	Linearized wave test	45
3.8.3	Gowdy wave	45
3.9	Black Hole metrics	46
3.9.1	Kerr-Schild coordinates	46
3.9.2	Boyer-Lindquist coordinates	47
3.9.3	Transformation between Kerr-Schild and Boyer-Lindquist systems	47
3.9.4	Binary black holes	48
3.9.5	Apparent Horizon of a black hole	49
4	Numerical Methods	50
4.1	Choice of numerical methods	50
4.1.1	Evolution schemes	50
4.1.2	Grid Structure	52
4.1.3	Mesh Adaptation	54
4.2	Finite Volume Schemes	55
4.2.1	FORCE and SLIC schemes	57
4.2.2	Multi-Staging approach (MUSTA)	59
4.2.3	GFORCE flux	63
4.2.4	HLL flux	64
4.2.5	Recommendations for the MUSTA approach	65
4.3	Finite Difference Schemes	65

4.4	Source Terms and Splitting	66
4.4.1	Combining the source evolution with the flux	67
4.4.2	Simpson's Rule correction	68
4.4.3	Flux-Source balancing	69
4.5	Multi-dimensions	69
4.6	Curvi-linear grids	70
4.6.1	HLL scheme	72
4.6.2	Determination of time step	72
4.6.3	Free-streaming correction	73
4.7	Adjustments for a non-Minkowski metric	74
4.8	Boundary conditions	75
4.8.1	Excision boundary conditions	75
4.8.2	Boundary conditions at spatial infinity	76
4.8.3	Reflective boundary conditions	76
4.9	Mesh refinement	76
4.9.1	Grid creation	78
4.9.2	Sub-cycling	80
4.9.3	Overlapping grids	81
4.9.4	Refinement criterion	81
4.10	Overlapping grids	82
4.10.1	Cell size considerations	84
4.10.2	Keeping variables physical	84
4.11	Overall time-stepping approach	84
5	Validation of schemes	86
5.1	One-dimensional SRHD Tests	86
5.1.1	Mildly Relativistic Shock	86
5.1.2	Strongly Relativistic Blast Wave	89
5.1.3	Perturbed Density Test	91
5.1.4	One-dimensional AMR	94
5.1.5	One-dimensional non-uniform grid	95
5.1.6	Forcing an exact solution	96
5.2	Two-dimensional SRHD Tests	99
5.2.1	Two-dimensional Quadrants test	99
5.2.2	Adaptive Mesh Refinement	100

5.2.3	Shock propagation across an overlapping grid	104
5.2.4	Front-facing step	106
5.2.5	Reflective cylinder	108
5.2.6	Multiple reflective cylinders	110
5.2.7	Forcing an exact solution	110
5.3	One-dimensional Metric Tests	111
5.3.1	Medium System Tests	111
5.3.2	Apples with Apples tests	114
5.4	Conclusions	121
6	Software development issues	123
6.1	Overture	123
6.1.1	Distributed arrays	123
6.1.2	Grids	124
6.2	Optimisation and Performance	125
6.3	AMR	128
6.4	Conclusions	129
7	Bondi-Hoyle-Lyttleton Accretion	131
7.1	Bondi-Hoyle-Lyttleton Accretion	131
7.1.1	Relativistic BHL Accretion	132
7.1.2	Newtonian BHL accretion	132
7.1.3	Applications of BHL accretion	133
7.2	Model Description	133
7.3	Numerical parameters	135
7.3.1	Evolution methods	135
7.3.2	Grid setup	135
7.4	Post-processing of results	138
7.4.1	Mass and momentum accretion	138
7.4.2	Velocities	139
7.4.3	Shock angles	140
7.4.4	Observable effects	141
7.5	Validation of numerical schemes	142
7.5.1	Exact solutions	142
7.5.2	Validation results	143

7.5.3	Conclusions	150
8	Numerical study of Bondi-Hoyle-Lyttleton Accretion	151
8.1	Base model parameters	151
8.2	General flow morphology	154
8.2.1	Features common to all models	154
8.2.2	Subsonic models	155
8.2.3	Supersonic models	163
8.2.4	Comparison with previous results	169
8.2.5	Conclusions	170
8.3	Parametric study setup	171
8.3.1	Models	172
8.3.2	Further validation	172
8.4	Parametric study results	173
8.4.1	Flow morphology	173
8.4.2	Processed results	188
8.4.3	Independence of initial conditions	196
8.5	Conclusions	198
8.6	Wind Accretion onto a Binary Black Hole system	199
8.6.1	Simulation setup	200
8.6.2	Results	202
8.6.3	Comments	207
9	Conclusions	208
9.1	Computational infrastructure and numerical methodologies	208
9.2	Numerical schemes	209
9.3	Bondi-Hoyle-Lyttleton accretion	209
10	Future Work	211
10.1	Code structure	211
10.2	Numerical Methods	211
10.3	Bondi-Hoyle-Lyttleton Accretion	212
10.4	Final remarks	213

List of Figures

2.1	Illustration of the lapse and shift	18
3.1	Mildly relativistic shock - exact solution	41
3.2	Relativistic blast wave - exact solution	42
3.3	Density Perturbation test - exact solution	43
4.1	Lego excision of a circle from a Cartesian grid	53
4.2	Illustrations of shock capturing.	56
4.3	Multi-staging a simple scheme	60
4.4	Dependence of four flux evaluation schemes on local CFL number	62
4.5	Dependence of multi-staged flux schemes on local CFL number	62
4.6	Dependence of MUSTA approach on CFL number, with scaled α	63
4.7	Example AMR grid setup	78
4.8	Refining a grid based on flagged points	79
4.9	Example overlapping grid generated by Overture	83
5.1	1D simulation of a mildly relativistic shock wave	87
5.2	Comparison of various schemes on mildly relativistic shock problem	88
5.3	Results for a strongly relativistic blast-wave	89
5.4	Results for density perturbation test	92
5.5	Comparison of density perturbation results for various methods	93
5.6	Results for blast-wave test with AMR	94
5.7	Results for blast-wave test on non-uniform grid	95
5.8	Quadrants problem solution, uni-grid.	100
5.9	Evaluation of GFORCE on quadrants problem	103
5.10	Grid for testing curvi-linear routines.	104
5.11	Cross-section of shock evolved across a curvi-linear grid	105

5.12	Contour plot of shock across curvi-linear grid	106
5.13	Grid used for the front-facing step problem.	107
5.14	Evolved solution for front-facing step	107
5.15	State of refined grid at end of front-step problem	108
5.16	Grid used for reflective cylinder problem.	109
5.17	Solution for reflective cylinder test.	109
5.18	Solution for two reflective cylinders	110
5.19	Results for the Medium system, test case 1.	113
5.20	Results for the Medium system, test case 2.	114
5.21	Constraint evolution for Apples with Apples robustness test	115
5.22	Linearized wave test results	116
5.23	Linearized wave test - limiter comparison	116
5.24	Linearized Wave Test - scheme comparison	117
5.25	Linearized Wave Test - resolution comparison	118
5.26	Using different error norms on Gowdy wave test.	119
5.27	Backwards Gowdy wave problem - various flux schemes	120
5.28	Convergence plot for Gowdy wave problem	120
7.1	Capturing a sphere with overlapping grids	136
7.2	Shock-angle location example	141
7.3	Evolution of Petrich <i>et al.</i> 's exact solution	144
7.4	Mass accretion rate for exact solution	145
7.5	Comparison of numerical and exact solutions for stiff fluid accretion	145
7.6	Evolution of stiff fluid to steady state	146
7.7	Mass accretion rate for evolution of stiff fluid	146
7.8	Velocity contours for stiff fluid accretion	148
7.9	Mass accretion rate for stiff fluid	149
8.1	Steady state contour plots for model UB1	153
8.2	Mass accretion rate for UB1 model	153
8.3	Uniform subsonic fluid flow past a non-spinning black hole, model UB0	155
8.4	Uniform subsonic fluid flow past a non-spinning black hole, model UC0	156
8.5	Perturbed subsonic fluid flow past a spinning black hole, model UB0	157
8.6	Perturbed subsonic fluid flow past a spinning black hole, model UB0, $z = 1$ plane	157
8.7	Perturbed subsonic fluid flow past a spinning black hole, model UC0	158

8.8	Accretion regions for subsonic flow	158
8.9	Mass accretion rates for subsonic flows	159
8.10	Stagnation points for subsonic flows	160
8.11	Streamlines for a perturbed flow onto a spinning black hole, model UB0	161
8.12	Streamlines for a perturbed flow onto a spinning black hole, model UC0	162
8.13	Uniform supersonic fluid flow past a non-spinning black hole, model UA1	163
8.14	Uniform supersonic fluid flow past a non-spinning black hole, model UB1	164
8.15	Accretion regions for supersonic flow	164
8.16	Perturbed supersonic fluid flow past a spinning black hole, model UA1	165
8.17	Streamlines for a perturbed flow onto a spinning black hole, model UA1	166
8.18	Perturbed subsonic fluid flow past a spinning black hole, model UB1	167
8.19	Streamlines for a perturbed flow onto a spinning black hole, model UB1	168
8.20	Mass accretion rates for supersonic flows	169
8.21	Stagnation points for supersonic flows	169
8.22	Diagram of wind accretion variables	172
8.23	Effect of spin on Bondi-Hoyle-Lyttleton accretion	176
8.24	Streamlines for a uniform flow onto a spinning black hole	177
8.25	Effect of density perturbation on Bondi-Hoyle-Lyttleton accretion	178
8.26	Streamlines for perturbed flow onto a non-spinning black hole	179
8.27	Effect of density perturbation for spinning black hole	180
8.28	Streamlines for perturbed flow $\epsilon_\rho = -0.2$ onto a spinning black hole	181
8.29	Streamlines for perturbed flow $\epsilon_\rho = 0.2$ onto a spinning black hole	182
8.30	Streamlines for perturbed flow $\epsilon_\rho = 0.2$ onto a spinning black hole	183
8.31	Effect of spin on accretion along spin axis	184
8.32	Streamlines for uniform flow along the spin-axis of a spinning black hole	185
8.33	Effect of density perturbation on accretion along spin axis	186
8.34	Streamlines for perturbed flow $\epsilon_\rho = 0.2$ along the spin-axis of a spinning black hole	187
8.35	Effect of wind direction and density perturbation on accretion volume for $a = 0$	188
8.36	Effect of wind direction and density perturbation on accretion volume for $a = 0.5$	191
8.37	Effect of wind direction and density perturbation on accretion volume for $a = 0.9$	192
8.38	Variation of mass accretion rate with density perturbation	193
8.39	Variation of mass accretion with density perturbation for spinning black hole	193
8.40	Effect of density perturbation on shock-cone angle	194
8.41	Effect of density perturbation on shock-cone angles for spinning black hole	194

8.42	Effect of density perturbation on drag angle	195
8.43	Effect of density perturbation on drag angle for spinning black hole	195
8.44	Mass accretion rate for time-dependent flow.	197
8.45	Mass accretion rate for kicked black hole.	198
8.46	Simple, but asymmetric grid for binary black hole simulation	201
8.47	More complex, but symmetric grid for binary black hole simulation	202
8.48	Accretion onto a binary black hole system	203
8.49	Results for accretion onto a binary black hole system	204
8.50	Mass accretion rate for binary black hole system	205
8.51	Cross-section of the accretion region for binary black hole system	205
8.52	Streamlines for a uniform flow onto a binary black hole system	206

List of Tables

4.1	Values of α required for MUSTA stability	63
5.1	Error norms calculated for comparison of various flux schemes	88
5.2	Density peaks for strong relativistic shock for various methods	90
5.3	\mathcal{L}_1 errors in density for Blast-Wave test.	91
5.4	Demonstration of improvements to one-dimensional fluid evolution	98
5.5	Run-times for quadrants problem, including AMR	101
5.6	Orders of convergence for quadrants problem with AMR	102
5.7	Demonstration of improvements to two-dimensional fluid evolution	111
6.1	Profiling run for a three-dimensional spherical grid with medium resolution	126
6.2	Speed-up statistics for code	127
6.3	Profiling statistics for an AMR simulation	129
7.1	Excision radii	137
7.2	Grid parameters for different resolutions	138
7.3	Dependence on resolution of accretion rates for stiff fluid.	147
8.1	Font and Ibáñez’s test problem parameters	152
8.2	Mass accretion rates	170
8.3	Stagnation point locations	170
8.4	Effect of wind direction on mass accretion rate	173

Contents of DVD

The DVD accompanying this thesis contains the complete results of the parametric study described in Chapter 8. A complete description of its contents is given both on the DVD and in Chapter 8. The content on the DVD should be accessible via any Flash-enabled, standards-compliant web-browser. To start using the DVD, open `index.html` in a web-browser.

Acknowledgements

I am particularly grateful to my supervisor, Nikos Nikiforakis for all his encouragement, support, and helpful suggestions.

I am also grateful to John Stewart, for suggesting the use of the Medium system and for related assistance; and to Tito Toro for comments on the use of the MUSTA approach and the GFORCE scheme.

Thanks also go to Stuart Rankin for helping me with some problems with deploying my code onto the Darwin super-computer, and to Jonathan Gair for making some useful suggestions on astrophysical applications of my code.

I am also grateful to Graeme Leese, Kevin Bates, and other members of the research group for being available for general programming and algorithm advice.

The research was funded by an EPSRC Doctoral Training Grant. Use was also made of the Darwin Supercomputer of the University of Cambridge High Performance Computing Service (<http://www.hpc.cam.ac.uk/>), provided by Dell Inc. using Strategic Research Infrastructure Funding from the Higher Education Funding Council for England.

Chapter 1

Introduction

The theory of General Relativity (GR) was developed by Albert Einstein in the early part of the 20th century. The theory has been shown to be impressively accurate in certain cases, including the perihelion progression of Mercury, and the variation of the period of binary pulsars (see [86, 113]). The Einstein field equations in the presence of matter can be written in the deceptively simple form

$$G_{\mu\nu} = 8\pi T_{\mu\nu}, \tag{1.1}$$

where $G_{\mu\nu}$ is the Einstein tensor, representing the space-time curvature, and $T_{\mu\nu}$ is the energy-momentum tensor, representing any matter or other energy field present in the space-time.

This simple form arises from the fact that GR is a *covariant* theory, that is, it is invariant under local Lorentz transformations. This is the result of the assumption on which special relativity is based, that the speed of light is locally the same for all physical observers. GR extends this to a curved space-time, where we find that what is commonly referred to as the force of gravity can be regarded as the result of matter following geodesics in the curved space-time.

The Einstein equations are highly non-linear second order partial differential equations in the space-time metric, however, and as a result relatively few classes of analytic solutions are known, and these often have a high degree of symmetry. Also, there are very few astrophysically interesting solutions among them. Progress has been made for some systems where GR effects do not dominate, using Post-Newtonian approximations, but it is necessary to solve the full set of Einstein's equations for more extreme cases such as the inspiral and merger of two black holes.

The method of solution is of necessity numerical in nature and, since general relativistic problems often require fully three-dimensional simulations in a large computational domain, considerable effort is being put into finding accurate, robust, parallelisable, and computationally efficient solution techniques. It is on this that we concentrate in this thesis.

1.1 Conventions

We take the speed of light to be $c = 1$, and Newton's gravitational constant to be $G = 1$. This is a simple matter of non-dimensionalisation.

We also adopt the Einstein summation convention throughout, except where it is explicitly stated that we do not. Since we will be splitting the space-time into space-like and time-like parts, we also take the convention that Greek indices μ, ν, \dots represent space-time indices 0, 1, 2, 3, and Latin indices i, j, \dots represent space indices 1, 2, 3.

We take the sign convention that the Minkowski metric is given by $\eta_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$, and that any curved space-time has the same signature.

The Christoffel symbols for a symmetric metric are defined as

$$\Gamma_{j\ k}^i = \frac{1}{2}\gamma^{ip}(\gamma_{jp,k} + \gamma_{pk,j} - \gamma_{jk,p}) \quad (1.2)$$

for a 3-metric γ_{ij} , and

$$\Gamma_{\mu\ \nu}^{\sigma} = \frac{1}{2}g^{\sigma\rho}(g_{\mu\sigma,\nu} + g_{\sigma\nu,\mu} - g_{\mu\nu,\sigma}) \quad (1.3)$$

for a 4-metric $g_{\mu\nu}$, where $_{,i}$ denotes partial differentiation with respect to the coordinate variable x^i . We use $_{;i}$ to denote covariant differentiation with respect to x^i :

$$C^a_{\ b;c} = \partial_c C^a_{\ b} + \Gamma_{c\ i}^a C^i_{\ b} - \Gamma_{c\ b}^i C^a_{\ i}. \quad (1.4)$$

We note that 3-indices are raised and lowered using the appropriate 3-metric, and 4-indices are raised and lowered using the appropriate 4-metric.

The Riemann curvature tensor in three dimensions is defined as

$${}^{(3)}R^i_{\ jkm} = \Gamma_{j\ m,k}^i - \Gamma_{m\ k,j}^i + \Gamma_{p\ k}^i \Gamma_{j\ m}^p - \Gamma_{j\ p}^i \Gamma_{m\ k}^p, \quad (1.5)$$

and the Ricci tensor, Ricci scalar, and Einstein tensor, respectively, as

$${}^{(3)}R_{ij} = {}^{(3)}R^p_{\ ipj}, \quad {}^{(3)}R = \gamma^{ij}{}^{(3)}R_{ij}, \quad {}^{(3)}G^{ij} = {}^{(3)}R^{ij} - \frac{1}{2}\gamma^{ij}{}^{(3)}R. \quad (1.6)$$

There are also four-dimensional equivalents of these.

We note that, when discussing numerical methods, a subscript i refers to a value evaluated at the centre of cell i , and a subscript $i + 1/2$ refers to a value evaluated at the interface between cells i and $i + 1$. A superscript refers to a time-step number in a similar way. Hence, $u_i^{n+1/2}$ refers to the value of the numerically evolved variable u , evaluated at the centre of cell i , and at a point in time half-way between the time-steps n and $n + 1$. This notation could potentially be confused with the use of subscripts and superscripts to refer to space-time indices, but it is usually clear what an index means in a given context, and any cases where there could be confusion will be clarified.

Chapter 2

The State of Numerical Relativity

In this chapter we give a brief overview of the field of Numerical Relativity, its aims, and the current state of research. We also outline the aims of this thesis, and some issues that must be considered in the validation of our results. Two useful reviews of the field are by Alcubierre [1] and Lehner [68].

2.1 Research Areas in Numerical Relativity

There are three main research areas areas that require the numerical solution of Einstein's equations (Numerical Relativity, often abbreviated to NR):

1. It has already been confirmed (see Will [113] §5.1) that binary pulsars, where GR effects dominate, evolve as predicted by GR, with the decay of their orbits due to the emission of gravitational radiation following predictions closely. However, in order to explore the validity of GR further, it is necessary to examine phenomena such as that of binary black hole (BBH) coalescence, where even more extreme GR effects will occur. NR aims to be able to provide templates for the gravitational waves that will result from such events, so that these can be compared to physical observations. This will make it possible to detect any deviations from classical GR. It is likely that this will only require the solution of Einstein's equations for a vacuum.
2. Some highly energetic astrophysical phenomena, such as the iron-core collapse of massive stars, can be modelled to some degree using Newtonian or Post-Newtonian approximations to GR (see [87]). However, more accurate simulations will require GR effects to be taken into account and these effects may be sufficiently strong so as to either change predictions made by these models, or to rule out postulated paths for certain events to occur, such as the physical mechanisms that lead to supernovae. This will require relativistic fluids to be simulated, perhaps including magnetic fields

and micro-physical effects.

3. Once gravitational wave templates are firmly established, it should be possible to use these data banks to process data from gravitational wave detectors so as to extract physical properties of events which would otherwise be undetectable by telescopes relying on the electromagnetic spectrum. This would be yet another tool by which we could gather information about distant astrophysical events.

The gravitational waves referred to above are the main interest in NR. These are predicted by GR through a linearised perturbation of Minkowski space. According to classical GR, they should travel at the speed of light, and are mainly generated by highly dynamic astrophysical events such as BBH coalescence. They cause distortions in space-time which, although they result in variations of the order of one part in 10^{21} by the time they reach the Earth (see Will [113] §6.1), should nonetheless be detectable using experiments such as the Earth-based Laser Interferometer Gravitational-Wave Observatory (LIGO [28]), VIRGO [80], and GEO600 [43]. There is also a space-borne system in the planning stages, the Laser Interferometer Space Antenna (LISA [70]).

These usually function by suspending test masses a large distance apart (of the order of several kilometres for the Earth-bound experiments, and millions of kilometres for LISA) and measuring their motion as gravitational waves pass through. The setup is similar to that of a classic Michelson interferometer, with two perpendicular arms having a beam splitter at their intersection and test masses at the other ends. Changes in the lengths of the arms are then measured using laser interferometry. This is possible since gravitational waves moving in a direction perpendicular to the plane of the interferometer will cause the length of one arm to increase while the other decreases.

There are many technical problems to overcome when building these detectors. These include reducing thermal noise in the detector components, shielding the test masses from local seismic vibrations, such as that caused by passing traffic, and designing laser frequency stabilisation systems to ensure the interferometry is carried out as accurately as possible. Further details of the problems involved with building these experiments can be found in the Living Reviews article by Rowan and Hough [91].

Since gravitational waves pass through low-mass objects such as the Earth virtually unchanged, ascertaining which part of the sky a signal comes from is far more difficult than with conventional telescopes. The different systems will have to work together to rule out effects local to each detector (such as those caused by seismic or thermal vibrations), and to provide a directional estimate for signals by using triangulation based on the times at which signals were received by different detectors. The data analysis required to extract a single signal from the raw data is also highly complex, and is the subject of much research, prompted by the Mock LISA Data Challenge [9].

In fact, although considerable effort has been put into NR, and substantial progress has been made in recent years, there has been little physical verification of any of its predictions. Of course, there has

been rigorous verification of the NR framework by similar results obtained from quite different numerical techniques, and from Post-Newtonian approximations, but this is simply internal consistency of the solution techniques used.

2.2 Foundations of Numerical Relativity

In the absence of any method to solve Einstein's equations numerically in all four dimensions at once, we must split the space-time in some fashion. There are two main approaches to this. The first approach is to perform a 3 + 1 split [116], singling out a time-like coordinate, and evolving a three-dimensional hypersurface forward in time. The second approach is to perform a 2 + 2 split, decomposing the space-time into two null coordinates and two space-like coordinates. A two-dimensional hypersurface is then evolved along the two null coordinate directions simultaneously.

The latter approach has several advantages for vacuum space-times, and for the extraction of gravitational waves, whose path will lie in the plane formed by the null coordinates. However, for evolving fluids, this approach is not the best one, and throughout this thesis we use the 3 + 1 splitting approach, which is as follows:

In order to turn the intrinsically four-dimensional Einstein equations into a system of equations that can be evolved in time from some three-dimensional initial conditions, we decompose the metric as

$$ds^2 = g_{\mu\nu}dx^\mu dx^\nu = -\alpha^2 dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt), \quad (2.1)$$

and we seek to evolve the 3-metric γ_{ij} in the time coordinate t . The metric γ_{ij} is that of a 3-dimensional space-like hypersurface embedded in four-dimensional space-time. The extrinsic curvature of the hypersurface is given by the Lie derivative of the 3-metric with respect to the vector normal to the space-like hypersurfaces:

$$K_{ij} = -\frac{1}{2}\mathcal{L}_n\gamma_{ij} = -\frac{1}{2\alpha}(\partial_t\gamma_{ij} + \nabla_i\beta_j + \nabla_j\beta_i), \quad (2.2)$$

where $n^\mu = -\alpha\nabla^\mu t$,

and the evolution of the slice in the time-like coordinate t is governed by the lapse α and the shift β^i . The lapse is the rate at which the slice moves along the time coordinate perpendicular to the slice, and the shift allows for any change in coordinates in the slice itself. We also define a time vector

$$t^\mu = \alpha n^\mu + \beta^\mu \quad (2.3)$$

where we now write β with a 4-index for consistency, although it remains a space-like vector since $\beta^\mu n_\mu = 0$. These various vectors are shown graphically in Figure 2.1.

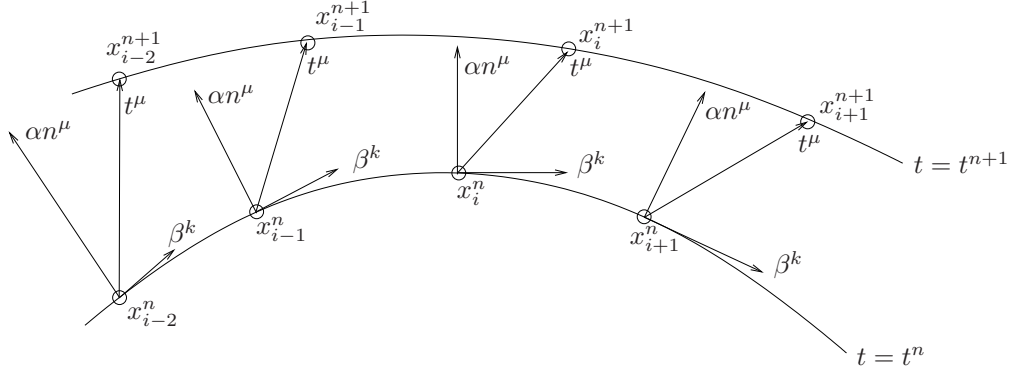


Figure 2.1: Two hypersurfaces at times $t = t^n$ and $t = t^{n+1}$, with lapses α and shifts β^k marked, with t^μ being the time-like vector separating equivalent coordinate points x_i^n and x_i^{n+1} . Here, Greek indices are space-time indices, and Latin indices refer to discretisation points and time-steps, except for those on the shift β^k , where they index the space-like components.

2.2.1 Gauge freedom

Clearly, a given four-dimensional space-time can undergo a $3 + 1$ split in an infinite number of ways, corresponding to different ways for the evolving hypersurface to be embedded into the four-dimensional manifold via the definitions of the space and time coordinates x^i and t , and the lapse and shift α and β^i . This freedom is known as the gauge freedom, since the particular form of the hypersurface does not affect any physical deductions from the evolution. However, some gauges may result in poor numerical evolutions, particularly when the hypersurface becomes strongly curved, resulting in numerical methods becoming inaccurate, often as a result of the lapse approaching zero near singularities.

Much time and effort has been devoted to investigating appropriate gauges for successful evolutions. For example, the work in [3] and references therein. Most of the other references to black hole evolution in this thesis also contain details of the various gauge evolution methods used.

However, this is not something that we investigate here, as our intention is to concentrate on the numerical methods used, and in particular on the test-fluid case where the metric is not evolved.

2.2.2 Constraints

Einstein's equations (1.1) represent ten separate equations in the components of the metric $g_{\mu\nu}$, where we assume that the metric is symmetric. These can be split into two types: those that do not contain any explicit time dependence (four equations) and those that do (the remaining six). Only the last six equations can act as evolution equations for the metric γ_{ij} ; the others can be shown analytically to be satisfied for all time if they are satisfied initially, and are referred to as the constraints of the system.

The four constraints for the Einstein equations can be written in terms of the 3 + 1-split variables as

$$\begin{aligned} \mathcal{H} &= {}^{(3)}R + (\text{tr}K)^2 - K^{ij}K_{ij} - 2\alpha^2 {}^{(4)}G^{00} \\ \text{and } \mathcal{M}^i &= \nabla_j(K^{ij} - \gamma^{ij}\text{tr}K) - \alpha {}^{(4)}G^{0i}. \end{aligned} \tag{2.4}$$

These must be zero for any initial data that is used, which we arrange either by choosing a known analytical solution to Einstein's equations, or by solving the constraint equations numerically, usually for some elements of K .

In any case, the constraints will not remain zero when the space-time is evolved numerically, either due to truncation errors in the numerical method used or round-off errors in the arithmetic. There are three possible ways of dealing with this. The first is to use the amount by which the constraints are violated as a measure of the (in)accuracy of the methods used. The second is to solve the constraint equations frequently, perhaps even after every time-step. It should be noted that this approach does not necessarily result in the constraints being zero throughout the whole evolution if half time-steps occur as a result of the time-stepping algorithm used, or through the use of adaptive mesh refinement (see §4.9). The third approach is to construct a formulation of Einstein's equations in which the constraints will be relaxed to zero. See, for example, the work of Tiglio [108], Gundlach *et al.* [53], and Marronetti [72, 73]. Other work in this area includes that of Bonazzola *et al.* [23], which uses a constrained formulation of Einstein's equations, avoiding constraint-violating modes altogether.

Although we will use constraint monitoring in this thesis, we note that this approach does have its drawbacks. Formulations of Einstein's equations that are supposedly appropriate for numerical solution may rely on the constraints being exactly or approximately satisfied. Once we have left the hypersurface in the space of possible metrics that corresponds to zero constraints, the formulation may not behave as expected. By this we mean that it is by no means certain that evolving a problem through non-zero constraints and then projecting onto the zero-constraint hypersurface will give the same result as evolving the problem correctly on the zero-constraint hypersurface. Some progress has been made in understanding this problem by Holst *et al.* [59], but the methods are not as yet in widespread use.

2.2.3 Formulations of Einstein's equations

In the form (1.1), Einstein's equations do not allow for any obvious numerical approach, and it is therefore necessary to rewrite them in a suitable form. It might initially be thought that the equations are hyperbolic by their very nature, as a result of the finite speed of propagation of physical information imposed by the speed of light as an upper bound. However, the non-physical gauge freedom is not constrained by this, and information about the gauge can travel at any speed through the computational domain.

There is no obvious way of splitting Einstein’s equations into gauge-free and gauge-only parts, which has led to the wide range of formulations with different approaches to evolving the gauge. However, the gauge often corresponds to a part of the formulation with elliptic character, for which elliptic equations would need to be solved. That said, gauge conditions which are hyperbolic in character are often sought, as hyperbolic systems can often be solved using less computational effort than elliptic or parabolic systems, and allow for more efficient parallelisation.

For many years, the most used formulation was the Arnowitt-Deser-Misner (ADM) formulation (see [75]). However, it has been shown that this formulation is ill-posed [47], meaning that it is not possible to put a bound on the growth-rate of a norm of the solution such that the bound is independent of the initial data. This could potentially cause stability problems in numerical evolutions by causing high-frequency modes to grow uncontrollably. With this in mind, several strongly hyperbolic formulations were produced, among them Bona-Massó [22] and Baumgarte-Shapiro-Shibata-Nakamura (BSSN [18]). Subsequent experimentation and experience has resulted in the BSSN formulation being the most widely used.

These formulations differ from each other mainly in their choice of evolved variables. Although all formulations ultimately evolve the space-time metric $g_{\mu\nu}$, supplementary variables such as the first derivatives of the metric are introduced in order to reduce Einstein’s equations to a first-order system of equations, and it is the choice of these variables that results in different formulations. Even within a given approach, there is still the choice of how the gauge, represented by the lapse α and the shift β^i , is to be evolved, and many choices of gauge drivers are available. Further modifications to a formulation may be made by adding multiples of the constraints (2.4) to the system. Since these are zero (at least analytically), this does not alter the solution space of the system, but it may alter its character or stability properties.

2.2.4 Initial data

The problem of initial data is not one that will concern us in this thesis as we deal either with stationary space-times for the purposes of test-fluid evolution or with exact solutions to Einstein’s equations for the purposes of testing. The specific solutions we use are given in §3.7, §3.8, and §3.9.

If we wished to evolve space-times whose form was not known analytically, then we would have to construct a set of initial conditions that had the required properties, and then solve the constraint equations (2.4). For example, in order to construct a space-time containing two black holes, we could take a linear combination of two Kerr black holes with different centres, and then solve the constraint equations numerically.

A further complication arises if we are outside the test-fluid approximation regime, as the fluid enters

into the constraints via the energy-momentum tensor $T^{\mu\nu}$. The constraint equations are elliptic in nature, and therefore require a large amount of computational effort to solve. This is one reason why it is not usual to solve for the constraints at every time-step, and why it is more usual to use different formulations of Einstein's equations which have the property that the constraints are reduced as far as possible.

More details of the issues involved in generating initial data for simulations can be found in the Living Review by Cook [29] and the overview presented byourgoulhon [52]. More recent work on the problem includes that of Foucart *et al.* [45], covering the construction of initial data for black-hole neutron-star binary problems.

2.3 Current state of Numerical Relativity

Numerical Relativity has been a subject of research since the late 1960s [68]. For much of that time, numerical simulations were restricted to one-dimensional or low-resolution two-dimensional cases. However, in the past ten years, computer technology and numerical techniques have developed sufficiently to allow large-scale simulations to be performed in three (spatial) dimensions. Further, in recent years, dramatic improvements have been made in the simulation of binary black-hole coalescence starting with the results of [85] in 2005. Many research groups now have robust codes capable of predicting gravitational wave signals for a wide range of black hole characteristics. For the current state of BBH simulations, see the review by Hinder [58] and references therein.

In this thesis, we intend to concentrate mainly on the evolution of a relativistic fluid on a stationary space-time background, and so we restrict ourselves to detailing the current research in this area. Codes have been developed to evolve the equations of general relativistic hydrodynamics on a variety of metrics, including dynamically evolved metrics, and various types of fluid. We now list the main codes that have been developed for the evolution of General Relativistic Hydrodynamics (GRHD), with their distinguishing features:

2.3.1 Cactus and associated thorns

The Cactus code [5] is a collaborative effort developed primarily by researchers at Louisiana State University and the Max Planck Institute for Gravitational Physics in Germany. The code is modular, so that it can be compiled with different sets of modules (known as thorns) depending on what is required of it. It also has the advantage that independent researchers can develop their own thorns, implementing new features, while taking advantage of all the computational infrastructure and numerical methods already available as thorns or in the main body of the code, known as the flesh.

Cactus is fully parallelised, with almost perfect scaling up to 131 072 cores on the Blue Gene/P

system at the Argonne National Laboratory [5]. It is capable of solving elliptic equations, and of using the Method of Lines approach to solving PDEs. It has the ability to perform multi-patch evolutions, where the domain of solution is covered by more than one numerical grid. These grids may be either abutting or overlapping but can only be Cartesian in nature. It is capable of evolving the space-time metric stably, using the BSSN formulation of Einstein's equations.

Carpet thorn

The Carpet thorn [99] for the Cactus code implements Adaptive Mesh Refinement (AMR), a standard numerical technique for focusing computational effort on areas of interest, often on shocks (see §4.9 for more details).

Whisky thorn

The Whisky thorn, written by Hawke, Baiotti, and Montero [12] for the Cactus code implements the evolution of GRHD simulations, and GRMHD has also been included in it [49].

The evolution is carried out using high-resolution shock-capturing schemes (see §4.2), using approximate Riemann solvers to calculate fluxes. Via the flesh of Cactus, the hydrodynamics are fully coupled with the evolution of the metric. Whisky is capable of excising portions of the computational grid to exclude, for example, the singularity at the centre of a black hole from the evolution. However, due to the restriction to Cartesian grids imposed by Cactus, the edge of the excision region is jagged.

Whisky has been fully tested on various test-cases, including the ability to evolve a stable magnetized neutron star [49], and applied to problems such as that of inspiralling neutron stars collapsing to form a black hole [13].

Curvi-linear Multi-block thorn

Schnetter *et al.* [98] have produced a Cactus thorn that is capable of evolving Einstein's equations on a multi-block grid setup, where the computational domain is constructed from more than one grid, perhaps using multiple coordinate systems which may overlap. They use high-order finite-difference operators and the boundaries between patches are handled using penalty methods. These methods, though, are not suitable for stably evolving fluid flow. They have simulated weak gravitational waves and also the evolution of the scalar wave equation on a Kerr-Schild background. The code is capable of reading grid structures from an external code suited to designing complex patch systems.

Curvi-linear Multi-patch Cactus thorn

There is currently under development by Zink *et al.* a code which has the ability to evolve GRHD problems on curvi-linear multi-patch grids [118]. Note that multi-patch grids are the same as the multi-block grids referred to in the previous section. The code is capable of using either overlapping or abutting grids, and has had a few grid setups implemented to give a spherical domain, including a 6-patch cubed sphere setup, and a similar 13-patch system of grids. The advantage of using curvi-linear grids is that spherical excision can be performed to remove a black hole singularity from the domain, using a smooth inner boundary. As is pointed out in [118], a spherical grid also has the advantage that extending its outer boundary requires a far lower increase in the number of computational cells as compared to a Cartesian grid.

The code uses a monotonised central TVD scheme with the HLL approximate Riemann solver (see §4.2). It has been tested on some one-dimensional shock-tube tests on overlapping grids, as well as the propagation of a one-dimensional shock across a three-dimensional cubed-sphere grid. Its most advanced test is that of evolving an equilibrium accretion torus on a Schwarzschild background.

2.3.2 Cornell and CalTech code

There is a code developed by Duez *et al.* [30] which has the ability to evolve the GRHD system of equations on a dynamic space-time. The code is capable of evolving binary black holes [27, 81], as well as black-hole neutron-star mergers [32].

This code uses pseudospectral methods to evolve the metric, and uses different grid systems for the metric and the fluid. The fluid is evolved using the HLL approximate Riemann solver, either in a finite volume (see §4.2) or in a finite differencing (see §4.3) setup.

The grid system for the metric can be one of several, including a spherical shell or a cubed sphere. The fluid grid, on the other hand, is Cartesian in nature. The use of pseudospectral methods for the metric means that the convergence rate is exponential, assuming the metric to be smooth. However, the convergence rate for the fluid evolution is still polynomial. The intended uses of this code are for cases where the fluid only covers a small part of the domain, so that the fluid grid does not need to cover the same extent as the metric grid.

Excision for the evolution of the fluid is performed by excising specific grid points, and using one-sided differencing near those points as necessary. This is an improvement on excision of a single cube, giving rise to Lego[®] excision, but the excision boundary is still not smooth and could give rise to numerical errors.

The code has been tested on spherical accretion onto a black hole, and also on the evolution of equilibrium relativistic stars. It has also been applied to black hole/neutron star binaries.

2.3.3 Wind Accretion by Font, Ibáñez, and Papadopoulos

This code, detailed in a series of papers [40, 41, 42], was developed with the intention of exploring Bondi-Hoyle-Lyttleton wind accretion onto black holes with and without spin. Although this code is not capable of solving as many types of problem as some of the other codes mentioned, we nevertheless give some details as we intend to perform similar simulations to the ones in the papers mentioned above.

The fluid is evolved using a Roe Riemann solver with monotonic linear reconstruction, and using TVD Runge-Kutta time-stepping. The metric is stationary only, and is restricted to a single black hole in either Boyer-Lindquist or Kerr-Schild coordinates. The grid setup is either that for an axisymmetric evolution, or that for the thin disc approximation to the evolution. We emphasise that both of these are two-dimensional calculations.

Excision is not strictly used for Boyer-Lindquist coordinates as the metric becomes singular at the horizon of the black hole, and so the grid stops outside the horizon. In Kerr-Schild coordinates, the grid goes inside the horizon, and the inner boundary is smooth owing to the annular nature of the grid.

2.3.4 Illinois GRHD code

There is a code developed by researchers at Illinois: Duez, Shapiro, and Yo [31]. This is capable of evolving GRHD coupled with the space-time metric. The grid setup is either axisymmetric or three-dimensional Cartesian. The metric is evolved using Iterated Crank-Nicolson with the BSSN formulation of Einstein's equations. The fluid is evolved using van Leer type advection, and uses artificial viscosity to handle shocks. It is therefore inaccurate for ultra-relativistic fluid flows.

The code has been tested using radial Bondi-accretion onto a Schwarzschild black hole, and has been used to simulate the collapse of rapidly rotating stars.

2.3.5 Nada code

This code, written by Montero, Font, and Shibata [76], is a recent development. It imposes axisymmetry, and so is a two-dimensional code only. However, it couples the space-time and fluid evolution. The space-time and fluid are evolved in time using either an Iterated Crank-Nicolson or a Runge-Kutta approach, and the fluxes are calculated using either the HLLE or the Roe approximate Riemann solver with slope-limited reconstruction.

Black holes are represented using the puncture approach [25], which avoids any excision requirements at singularities. The fluid can have either an ideal or polytropic equation of state.

The code has been shown to evolve rapidly rotating stars and also self-gravitating tori in equilibrium around a black hole. The latter case was tested with either a fixed space-time or a dynamical space-time,

and demonstrated long-term stable evolution in both cases.

2.3.6 SACRA code

The SACRA (SimulAtor for Compact objects in Relativistic Astrophysics) code has been developed by Yamamoto, Shibata, and Taniguchi [115]. This uses AMR on a three-dimensional Cartesian domain to evolve problems where the space-time is coupled to the hydrodynamics. The BSSN formulation of the Einstein equations is solved using a fourth-order finite-difference approach for calculating spatial derivatives, and a fourth-order Runge-Kutta for time integration. The hydrodynamical equations are solved using a high-resolution central scheme.

The code is capable of solving binary systems composed of black holes and neutron stars. Results produced using SACRA have reasonable agreement with those from other constant-refinement codes. This includes post-processed results such as gravitational waves. Further, conservation of energy and angular momentum are satisfied to a reasonable accuracy. As a result of using AMR, simulations with SACRA could be performed on personal computers, although requiring up to one month of CPU time to perform.

2.3.7 Brigham Young University

Anderson, Hirschmann, Lehner, and their collaborators have developed an NR code capable of evolving neutron-star binaries [7]. The code uses the generalized harmonic formulation of Einstein's equations. The space-time metric is evolved using up to fourth-order finite difference operators to calculate spatial derivatives, and the fluid is evolved using an HRSC solver. In both cases time is advanced using a third-order Runge-Kutta solver. The code includes AMR, and uses a three-dimensional Cartesian grid on which to evolve.

The code has been used to evolve a binary neutron-star system which collapses to a black-hole state, and is capable of extracting gravitational waves from the system. The code also has the ability to perform GRMHD simulations [6] of magnetized neutron-star mergers.

2.3.8 WHAM: WENO-based GRMHD code

This code, described in [104], is an improved version of the HARM code by Gammie, McKinney and Tóth [48], and is designed to evolve the equations of general relativistic magnetohydrodynamics on a stationary space-time. It uses the WENO approximate Riemann solver to calculate fluxes, as well as constrained transport to maintain a divergence-free magnetic field. The code can be used in any number of dimensions and in two dimensions can use either axisymmetric or cylindrical coordinates. It has been

applied to various tests both in flat-space and in black hole space-times, including an equilibrium torus in Kerr space-time. Overall, the scheme is fifth-order accurate, while the original HARM code was only second-order.

2.3.9 COSMOS++

Fragile *et al.* [8, 46] have developed a code capable of evolving GRMHD on a cubed-sphere grid setup. This setup allows for smooth excision inside the horizon of a Kerr black hole. The code is capable of adaptive mesh refinement, although done in a different way to Berger and Olinger [19], using a cell-by-cell refinement approach on an unstructured mesh. The methods used are advection based, using a first-order forward Euler scheme.

2.3.10 Zanotti and Dumbser

A quite different approach to those in the preceding sections is being used by Dumbser and Zanotti [33]. Their approach is to use discontinuous Galerkin methods on an unstructured mesh to achieve third to fifth order accuracy in both space and time. These are finite-element methods, which treat the numerical solution as a weighted sum of basis functions on the domain, rather than as values at a point or as integrals over a cell as for the methods used by other codes mentioned here.

Their work currently only covers the resistive (i.e. non-infinite conductivity) relativistic magnetohydrodynamical equations on a flat space-time. However, all other codes mentioned in this section (that include MHD) assume infinite conductivity. This assumption may not be valid for some astrophysical situations, such as pulsars and active galactic nuclei, and the work done by Dumbser and Zanotti may prove useful in simulations of these, particularly since they can use coarser meshes than other codes given in this section, as a result of the higher-order methods used.

The code has been tested against some exact solutions relevant to RRMHD, such as shock-tube tests, as well as the problem of a cylindrical explosion, and a vortex problem which, in the limit of high conductivity, gave results in agreement with those of previous research and for lower conductivity gave qualitatively plausible results.

2.3.11 Code review conclusions

We have seen that there are several mature codes capable of evolving the GRHD system of equations, either on a static space-time, or fully coupled with a dynamical space-time. Some codes are also capable of including magnetic fields in their simulations.

However, most of these codes are restricted to evolving on a Cartesian domain, and those that are not do not have adaptive mesh refinement capabilities, and generally rely on specific grid setups, or take

advantage of space-time symmetries and only evolve in two dimensions, although the code of Schnetter *et al.* mentioned in §2.3.1 is capable of using complex grid structures.

Therefore, the aims of this thesis are to design and validate an algorithm capable of evolving the GRHD system of equations on a general curvi-linear system of grids in three dimensions, including the ability to use AMR to make best use of computational resources, to implement the algorithm, and to apply it to a GRHD problem that requires the above features for its solution. We also intend to include in the algorithm the ability to evolve the space-time metric.

The advantages of our approach over previous work are two-fold. Firstly, by allowing a completely general grid structure, we will be able to implement a spherical grid system without polar singularities, which will allow us to extend the domain of solution to a large radius whilst using fewer grid points than a Cartesian grid of similar resolution near the black hole. Secondly, AMR will allow us to use an underlying coarse grid system, and then to concentrate computational effort on those areas which need it in order to increase accuracy.

We further believe that a general grid structure will be necessary for evolving problems such as fluid flow in the vicinity of a BBH system, where the singularities due to the black holes move in time, and the excised portion of the grid therefore needs to change in order to accommodate this.

2.4 Validating a Numerical Relativistic Code

One of the major difficulties in comparing NR codes is that not all codes are capable of some of the more advanced techniques, such as capturing moving singularities, and also that differences in gauge conditions make it difficult to make comparisons of simulation results. Further, there are a large number of possible combinations of formulations of Einstein’s equations with numerical solvers which, when combined, could have completely different stability and evolution properties.

It would therefore be useful to have some general relativistic tests that can be solved by any numerical relativity solver, which factor out as far as possible any implementation-dependent effects, and focus instead upon the numerical algorithm and formulation of the Einstein equations used. The tests will still need to be fairly stringent, however, and test the full dynamic capabilities of the algorithm. To this end, some suitable tests have been suggested by Alcubierre *et al.* [2] and Babiuc *et al.* [11]. In order to emphasise the fact that these tests are capable of comparing disparate algorithms and implementations, they are called “Apples with Apples” tests.

An online repository for associated information can be found at [44]. The test-cases specify in every detail some, mostly one-dimensional, problems that are straightforward to implement, and do not take large amounts of computational time, but are nonetheless challenging for NR solvers, largely as a result of

requiring many thousands of time-steps to evolve, thus bringing to light any secular stability or accuracy problems in the methods.

There are also some exact solutions available for the evolution of a test-fluid in the vicinity of a black hole, and these are detailed in §7.5. As well as these standardised tests, any solver should have test-cases developed to test its own particular features. We have constructed our own suite of tests including convergence order testing and testing against exact solutions for the numerical methods we have implemented.

We note that BBH simulations have advanced to a sufficient degree that gravitational wave signatures may be extracted, and these are another way of comparing different codes, as the wave signatures should be independent of the code. Such comparisons are now being made as part of the Samurai [55] and NINJA [35] projects. The NINJA project consists of providing templates for use in tests of code which will extract gravitational wave signatures from experimental data, and the Samurai project compares results from several different NR codes, using the predicted phase and amplitude of the gravitational waves.

2.5 Aims of this thesis

The first aim of this thesis is to construct a numerical algorithm applicable to general relativistic problems, and capable of evolving both the space-time metric and a fluid in that space-time, using an overlapping curvi-linear grid approach. The intention is to do this in such a way as should be generally applicable to any initial value problem. In the process of doing this, we will also aim to validate our algorithm extensively, comparing results with those of other numerical approaches, including those of other researchers, and using standardised test-cases. Since a large part of this thesis concerns algorithm development, we aim to spend some time describing and analysing the various numerical techniques used, checking them for consistency, and validating both the algorithms and their implementation.

We shall then aim to apply an implementation of our algorithm to the case of Bondi-Hoyle-Lyttleton accretion of a test fluid onto a stationary black hole, validate our algorithm for this specific problem, and finally perform a parametric study of this problem to explore previously uninvestigated aspects of the flow.

The remainder of this thesis is arranged as follows: in Chapter 3 we present the various systems of equations that we will be using our algorithm to solve, as well as the initial conditions for the test problems that we will use, including black hole metrics in appropriate coordinate systems. Chapter 4 details the numerical methods that we use for the solution of the equations, and our reasons for choosing them. In this we include some numerical techniques that have only recently been developed and have

not previously been applied to relativistic calculations, as well as some adjustments that we have made to some of the methods used in order to increase accuracy.

As promised, we then proceed in Chapter 5 to present extensive validation tests of all the numerical methods presented in the preceding chapter, and also some comparisons with results from other groups. Chapter 6 then covers some of the technical aspects of the implementation of our algorithm, and some of the features of the software libraries we have used to ease the implementation.

Chapter 7 then sees the introduction of the area which we wish to investigate, namely, Bondi-Hoyle-Lyttleton accretion onto a black hole. We detail some known results, the current state of research in this area, and perform some validation tests for this particular problem, before continuing to perform a parametric study of the problem, presented in Chapter 8. In Chapter 9 we detail our conclusions, both those from a consideration of the Bondi-Hoyle-Lyttleton accretion problem, and also those concerning the suitability of the algorithm we have developed for such problems. Finally, Chapter 10 suggests some directions this work could take in the future.

When implementing a new algorithm, a large amount of effort is usually required in areas other than simply the numerical algorithms, such as parallelisation and data input/output. We have therefore decided to use the Overture software [57], developed by Henshaw *et al.*, as a base for our implementation. This provides for overlapping grid generation and interpolation, as well as making available information about the grid geometries and grid functions that will allow us to develop easily our own numerical algorithms using that information. Overture has sufficient generality and scope that it implements, or allows for the easy implementation of, the features that we have defined as being desirable in §2.3.11. It further allows for the dynamic generation of grids from inside a user's code, and also for changes to the grid structure during an evolution.

Overture is set up in a data-distributed way that allows for efficient parallelisation. It also includes visualisation and input/output through a standard file format. All of this has allowed us to concentrate on developing the numerical algorithms, rather than aspects that are a matter of software development and well-understood mathematics.

Chapter 3

System Equations and Metrics

In order to allow for an uncluttered set of results in the validation in Chapter 5, we present here the systems of equations that we will use in this thesis, as well as some sets of initial data and metrics.

3.1 Special Relativistic Hydrodynamics

The evolution of a relativistic fluid on the Minkowski metric is governed by the conservation law

$$\frac{\partial \mathbf{U}}{\partial x^0} + \frac{\partial \mathbf{F}^i}{\partial x^i} = \mathbf{0}, \quad (3.1)$$

where

$$\begin{aligned} \mathbf{U} &= (D, S_j, \tau), \\ \text{and } \mathbf{F}^i &= (Dv^i, S_j v^i + p\delta_j^i, \tau v^i + pv^i). \end{aligned} \quad (3.2)$$

In the case of an ideal gas, the conserved fluid variables \mathbf{U} are defined in terms of the primitive fluid variables as

$$\begin{aligned} D &= \rho W, \quad S_j = \rho h W^2 v_j, \quad \tau = E - D = \rho h W^2 - p - D, \\ W &= (1 - \eta_{ij} v^i v^j)^{-1/2}, \end{aligned} \quad (3.3)$$

where ρ is the fluid density, p is the pressure, and v^i is the velocity vector. The specific enthalpy h characterises the heat content of a fluid per unit volume, and it defines the equation of state for the fluid when written in terms of the pressure and density. The conserved variables are the relativistic density D , the three-momenta S_i , and the relativistic energy E , although we evolve τ in place of E to allow for more accurate evolutions as otherwise variations in p can be swamped by the relatively large relativistic density D . The Lorentz factor W is the length contraction and time dilation factor, and we use the Minkowski metric η^{ij} for its calculation here.

3.2 General Relativistic Hydrodynamics

The evolution of a relativistic fluid on an arbitrary metric is given by

$$\frac{1}{\sqrt{-g}} \left(\frac{\partial \sqrt{\gamma} \mathbf{U}}{\partial x^0} + \frac{\partial \sqrt{-g} \mathbf{F}^i}{\partial x^i} \right) = \mathbf{S}, \quad (3.4)$$

where

$$\begin{aligned} \mathbf{U} &= (D, S_j, \tau), \\ \mathbf{F}^i &= \left(D \left(v^i - \frac{\beta^i}{\alpha} \right), S_j \left(v^i - \frac{\beta^i}{\alpha} \right) + p \delta_j^i, \tau \left(v^i - \frac{\beta^i}{\alpha} \right) + p v^i \right), \\ \mathbf{S} &= \left(0, T^{\mu\nu} \left(\frac{\partial g_{\nu j}}{\partial x^\mu} - \Gamma_{\nu}^{\delta}{}_{\mu} g_{\delta j} \right), \alpha \left(T^{\mu 0} \frac{\partial \ln \alpha}{\partial x^\mu} - T^{\mu\nu} \Gamma_{\nu}^0{}_{\mu} \right) \right), \end{aligned} \quad (3.5)$$

with $g = \det(g_{\mu\nu})$, and $\gamma = \det(\gamma_{ij})$. The expressions for D , S_j , and τ are as in (3.3), and

$$W = (1 - \gamma_{ij} v^i v^j)^{-1/2}. \quad (3.6)$$

The energy-momentum tensor $T^{\mu\nu}$ depends on the fluid-type, and we define it for the fluid-types we use in this thesis in §3.3.

Although (3.4) does not immediately resemble a conservation law, it is nonetheless derived from the conservation law $T^{\mu\nu}{}_{;\nu} = 0$ which follows from Einstein's equations. The terms on the right-hand side arise from the space-time curvature. It can easily be seen that equations (3.4) and (3.5) reduce to equations (3.1) and (3.2) in the case that $\gamma_{ij} = \eta_{ij}$ and $g_{\mu\nu} = \eta_{\mu\nu}$.

We note that the sources can also be written as (see documentation for the Whisky code [12])

$$\begin{aligned} \mathbf{S} = \left(0, T^{00} \left(\frac{1}{2} \beta^l \beta^m \gamma_{lm,j} - \alpha \alpha_{,j} \right) + T^{0i} \beta^l \gamma_{il,j} + \frac{1}{2} T^{lm} \gamma_{lm,j} + \frac{\rho h W^2 v_l}{\alpha} \beta^l{}_{,j}, \right. \\ \left. T^{00} (\beta^i \beta^j K_{ij} - \beta^i \alpha_{,i}) + T^{0i} (-\alpha_{,i} + 2\beta^j K_{ij}) + T^{lm} K_{lm} \right), \end{aligned} \quad (3.7)$$

assuming that the metric satisfies the constraint equations. This definition of the sources is equivalent to that in (3.5), but is better as it only uses variables already defined, rather than having to calculate the time derivatives of γ_{ij} and other space-time variables to incorporate into the four-Christoffel symbols. In the case of a stationary metric, these are exactly equivalent, but for a time-dependent metric, this means that we do not base the fluid source terms on the calculated metric source terms.

The wavespeeds for the full GRHD system of equations are given by ([39] §6.2):

$$\begin{aligned} \lambda^0 &= \alpha v^i - \beta^i \text{ (triple eigenvalue of the system)}, \\ \lambda_{\pm}^i &= \frac{\alpha}{1 - v^2 c_s^2} \left(v^i (1 - c_s^2) \pm c_s \sqrt{(1 - v^2)(\gamma^{ii}(1 - v^2 c_s^2) - v^i v^i (1 - c_s^2))} \right) - \beta^i \end{aligned} \quad (3.8)$$

in each coordinate direction i , where there is no sum over the i index.

3.3 Fluid types

In this thesis, we restrict ourselves to two fluid types: a perfect ideal gas, with a given adiabatic index, Γ , and a stiff, ultra-relativistic fluid.

Given an equation of state through the specific enthalpy h , the energy momentum tensor, found on the right-hand side of Einstein's equations (1.1), is given, for a perfect fluid, by

$$T^{\mu\nu} = \rho h u^\mu u^\nu + p g^{\mu\nu}, \quad (3.9)$$

where the four-velocity u^μ is defined in terms of the three-velocity v^i via

$$v^i = \frac{u^i}{\alpha u^0} + \frac{\beta^i}{\alpha}, \text{ and } W = \alpha u^0. \quad (3.10)$$

Here, α is the lapse function and β^i is the shift vector as defined by the 3+1 splitting in (2.1), and W is the Lorentz factor.

3.3.1 Stiff fluid

This is a special case of an ultra-relativistic fluid where, instead of taking $p = (\Gamma - 1)\rho$ as for ultra-relativistic flow, we now take $p = \rho$. The fluid is defined in terms of a stream-function ψ , which allows for some simplifications to the fluid equations, resulting in the possibility of analytical solutions (e.g. Petrich, Shapiro and Teukolsky [83], and Shapiro [102]). The details of the more general ultra-relativistic fluid can be found in David Neilsen's Ph.D. thesis [78].

We define the fluid variables as follows:

$$\begin{aligned} n &= (-\psi_{,\mu}\psi^{,\mu})^{1/2}, \quad u^\mu = \frac{1}{n}\psi_{,\mu}, \\ \rho = p &= n^2, \quad v^i = \frac{u^i}{\alpha u^0} + \frac{\beta^i}{\alpha}, \\ S_j &= \sqrt{\gamma}(\rho + p)W^2 v_j, \quad \tau = \sqrt{\gamma}(W^2(\rho + p) - p), \end{aligned} \quad (3.11)$$

where, due to the inherent simplifications, we can ignore the equation in D . This explains the absence of a $-D$ term in the expression for τ .

The primitive variables are easily recovered from the conserved variables via the formulae

$$\rho = \tau^2 - S_j S^j \text{ and } v_i = \frac{S_i}{\tau + \rho}. \quad (3.12)$$

We note that Neilsen [78] warns against calculating v_i in this way due to potential numerical precision problems, but the solution he gives is for one dimension only, and is not easily generalisable to multi-dimensions.

This equation of state leads to the the simplification of the conservation of energy and momentum, $T^{\mu\nu}_{;\mu} = 0$ to the linear wave equation $\psi^{;\nu}_{;\nu} = 0$. However, Neilsen notes [78] that not all solutions of the

wave equation can be interpreted as physical fluid solutions, only those that satisfy

$$\nabla\phi \cdot \nabla\phi < 0 \tag{3.13}$$

on the entire space-time.

3.3.2 Ideal fluid

For the ideal fluid, we define the specific enthalpy of the fluid to be

$$h = 1 + \frac{p}{\rho} \left(\frac{\Gamma}{\Gamma - 1} \right), \tag{3.14}$$

where Γ is the adiabatic index of the fluid (often written as γ , but that has already been used for the determinant of the three-metric). The speed of sound in the fluid, c_s , is then given by

$$c_s^2 = \frac{\Gamma p}{\rho} \left(1 + \frac{p}{\rho} \frac{\Gamma}{\Gamma - 1} \right)^{-1}. \tag{3.15}$$

In non-relativistic fluid evolution, the recovery of the primitive variables from the conserved variables is an algebraic operation. However, the presence of the Lorentz factor complicates the issue, since the three-momenta are no longer independent of each other, and we use a Newton-Raphson method to recover the primitive variables. Other techniques are available, including the solution of a quartic, but tend to be computationally involved and expensive.

Initially we followed Martí and Müller’s algorithm presented in [74] and available in the code `rppm.f` accompanying that article. However, their provision for dealing with large velocities, which was to restrict the pressure p by

$$p \geq p_{\min} \equiv (1 + \epsilon) \sqrt{\gamma^{ij} S_i S_j} - \tau - D \tag{3.16}$$

since we know that

$$v^2 < 1 \Leftrightarrow p > \sqrt{\gamma^{ij} S_i S_j} - \tau - D, \tag{3.17}$$

caused problems in precisely the cases it was meant to assist since, when the minimum pressure is used,

$$v^2 = \frac{\gamma^{ij} S_i S_j}{(p_{\min} + \tau + D)^2} = (1 + \epsilon)^{-2} \tag{3.18}$$

which then can lead to problems in calculating W accurately due to issues with finite precision (a background to the issues involved and some possible cures are given by Goldberg [51]).

Therefore, we follow instead the suggestion in Appendix A of Eulderink and Mellema [36]. This requires no calculation of W and, although it does have its own problems, these can be rectified in a

robust way. The equation we solve, using a Newton-Raphson method, is

$$\begin{aligned}
\alpha_4 \xi^3 (\xi - \eta) + \alpha_2 \xi^2 + \alpha_1 \xi + \alpha_0 &= 0, \\
\text{where } \xi &= \frac{\sqrt{(\tau + D)^2 - S^2}}{\rho h W}, \\
\alpha_0 &= -\frac{1}{\Gamma^2}, \quad \alpha_1 = -2C \frac{\Gamma - 1}{\Gamma^2}, \\
\alpha_2 &= \frac{\Gamma - 2}{\Gamma} (C_0^2 - 1) + 1 - C^2 \left(\frac{\Gamma - 1}{\Gamma} \right)^2, \quad \eta = 2C \frac{\Gamma - 1}{\Gamma}, \quad \alpha_4 = C_0^2 - 1, \\
C_0 &= \frac{\tau + D}{\sqrt{(\tau + D)^2 - S^2}}, \quad \text{and } C = \frac{D}{\sqrt{(\tau + D)^2 - S^2}}.
\end{aligned} \tag{3.19}$$

We note that in order to retain numerical accuracy, the expression for calculating $u^0 = W/\alpha$ in equation (A.12) of [36] should be rewritten as

$$W = \frac{1}{2} C_0 \xi \left(1 + \sqrt{1 + 4 \frac{\Gamma - 1}{\Gamma} \cdot \frac{1 - C\xi}{C_0^2 \xi^2}} \right), \tag{3.20}$$

taking $C_0 \xi$ outside the brackets, since for large W the bracketed expression is of order unity and $C_0 \xi$ is large. We then use the following to recover the remaining primitive variables:

$$\rho = \max(D/W, 10^{-8}), \quad h = 1/(C\xi). \tag{3.21}$$

Given that ρ may have been altered from its correct value of D/W , we recalculate the appropriate value of p from h , enforce $p \geq p_{\min}$ if necessary, and then recalculate h before calculating v_i as

$$v_i = \frac{S_i}{\rho h W^2}. \tag{3.22}$$

According to [36], “with some effort it may be shown that the derivative of [the quartic (3.19)] is always at least $2/\Gamma$ in its physical root, thus guaranteeing quadratic convergence for the Newton[-Raphson] method”.

We performed some tests to determine the error in converting conserved variables to primitive variables and back again. We found that with full optimization switched on (-O3 for the Intel and GNU compilers, but with floating-point precision maintained), the adjustment in (3.20) was the most important, i.e. any adjustment of the way in which the ratio C_0 was calculated, or whether C_0^2 was calculated from C_0 or directly from τ, D , and S^2 made no discernible difference to the accuracy.

Recovery from non-physical conserved variables

As explained later in §4.10.2, we may still get unphysical conserved variables if

$$(\tau + D)^2 \geq S^2 \tag{3.23}$$

is violated due to some numerical error in the evolution.

It does not seem appropriate to effectively cap the maximum velocity v^2 by setting D or τ explicitly to satisfy this inequality. Further, specifying a vacuum state of $v = 0$, $\rho = \rho_{\min}$ and $p = p_{\min}$ where (3.23) is violated is unsatisfactory as it is usually violated in regions where v^2 is large, and setting $v^2 = 0$ is therefore inappropriate.

Our solution to this is that, if the inequality is violated, then we specify a minimum pressure p , and then calculate ρ and W from the values of D and S^2 . The choice of variables either to specify or from which to calculate is made for ease of implementation. It would be possible to set either a minimum pressure or a minimum density. However, the viability of such a minimum relies on the positivity of the other variable, as we shall see below. This is trivial to check in the case of density, where $\rho > 0 \Leftrightarrow D > 0$, but much harder to check in the case of pressure, and so we choose to fix a minimum pressure.

Then, since most of our problems arise from violations of $v^2 < 1$, the easiest way to enforce this is to use the known value of S^2 in our recovery of the primitive variables, as this has an overall factor of v^2 , and will therefore lead to the most appropriate value for v^2 . Choosing not to use the known value of S^2 may well lead to the resulting primitive variables not satisfying $v^2 < 1$, so that our initial difficulty remains unsolved. Our subsequent choice of fixing D was mainly due to ease of implementation, but does represent to some extent a fixing of the original density, given that we have already chosen to specify a minimum pressure. In practice, we use $p_{\min} = 10^{-8}$.

It is important to demonstrate that a solution is possible for any value of p_{\min} that we might impose and this is shown by the following argument:

We want to find the range of p s.t. $(\tau + D)^2 > S^2$ can be satisfied, given values of D and S^2 :

$$\begin{aligned}
& (\tau + D)^2 > S^2 \\
\Leftrightarrow & (\rho h W^2 - p)^2 > \rho^2 h^2 (W^4 - W^2) \\
\Leftrightarrow & p^2 - 2W^2 p \left(\rho + p \frac{\Gamma}{\Gamma - 1} \right) > -W^2 \left(\rho + p \frac{\Gamma}{\Gamma - 1} \right)^2 \\
\Leftrightarrow & p^2 \left(1 - W \frac{\Gamma}{\Gamma - 1} \right)^2 + 2pW^2 \rho \frac{1}{\Gamma - 1} + W^2 \rho^2 > 0
\end{aligned} \tag{3.24}$$

and, since all the coefficients of p^2 , p , and 1 are positive, assuming that $\rho > 0$ and $\Gamma > 1$, this is true for any value of p , so any value of p_{\min} is consistent with given $\tau + D$ and S^2 , although not with τ , D , S^2 separately, as this would lead to an over-determined system of equations.

Now, given S^2 and D , we wish to solve the inequality

$$S^2 < (\tau + D)^2 \tag{3.25}$$

for ρ as closely as possible to equality. Some algebra shows that we wish to solve

$$\begin{aligned}
0 < f(\rho) & \equiv -\rho^4 (D^2 + S^2) - 2\rho^3 \lambda D^2 + \rho^2 (D^4 - \lambda^2 D^2) + 2\rho \lambda D^4 + \lambda^2 D^4 \\
& \text{where } \lambda = p_{\min} \frac{\Gamma}{\Gamma - 1}.
\end{aligned} \tag{3.26}$$

The form of this quartic is not obviously amenable to solution by an iterative method such as Newton-Raphson as there are no suitable bounds on its derivative, although we could use an alternative method to find a suitably close solution, and then use Newton-Raphson to converge rapidly to a more accurate solution. However, we know that $\rho \in [0, D]$ and so we have decided to perform a bisection root-finding method to find ρ , stopping when the upper and lower limits for ρ differ by a given relative amount.

This is possible since $f(0) = \lambda^2 D^4 > 0$ and $f(D) = -D^4 S^2 < 0$, so that $f(\rho)$ has at least one root in the range $[0, D]$. If we assume that λ is small (justified by the facts that p_{\min} is small and we do not use Γ close to 1) compared to D and S^2 (which are large for large W), then we can consider the polynomial

$$-\rho^4(D^2 + S^2) + \rho^2 D^4 = 0 \quad (3.27)$$

with roots at

$$\rho = 0, \pm D^2 / \sqrt{D^2 + S^2}. \quad (3.28)$$

We then regard f as a perturbation of this, and note that $f(-\lambda) = -\lambda^4 S^2 < 0$, implying that f has a root in the range $(-\lambda, 0)$.

Since the ρ^4 coefficient of f is negative and the constant coefficient is positive, we know that f has an odd number of negative roots. Therefore, with the assumption that f has a root in the vicinity of the root of the unperturbed polynomial, $-D^2 / \sqrt{D^2 + S^2} \ll -\lambda$, we deduce that f has three negative roots, and hence has precisely one positive root, so that the bisection method will always succeed.

The bisection approach is slower than an iterative scheme would be, but is used far less frequently by the algorithm, and so its efficiency is less important. A first step to improve efficiency is to note that $f(\lambda + D^2 / \sqrt{D^2 + S^2}) < 0$, and so we start our bisection approach with the range $[0, \lambda + D^2 / \sqrt{D^2 + S^2}]$.

We have therefore shown that we have used a fast, robust, and accurate method for recovering primitive variables. Further, we have made provision for the possible case where the conserved variables become unphysical, in that the primitive variables we then recover would convert back to conserved variables close to the original ones.

3.4 Medium system

The Medium system was developed by Barnes *et al.* [16]. It is a specialised version of the full Einstein equations, and describes the evolution of an ultra-relativistic fluid on a Gowdy plane-polarised background. The metric used is

$$ds^2 = e^{2a}(-dt^2 + dx^2) + e^{2b}(e^{2c}dy^2 + e^{-2c}dz^2), \quad (3.29)$$

where a, b, c are functions of x and t alone. The primitive fluid variables are μ , the density and v , the velocity. The fluid pressure is then $P = (\Gamma - 1)\mu$, and the Lorentz factor is $W = (1 - v^2)^{-1/2}$.

The conserved fluid variables are

$$S = (\mu + P)vW^2, \text{ and } \tau = (\mu + P)W^2 - P, \quad (3.30)$$

and we can use the expressions

$$\begin{aligned} \mu &= \frac{1}{\Gamma - 1} \left(-\frac{1}{2}(2 - \Gamma)\tau + \left[\frac{1}{4}(2 - \Gamma)^2\tau^2 + (\Gamma - 1)(\tau^2 - S^2) \right]^{\frac{1}{2}} \right) \\ \text{and } v &= \frac{\Gamma\mu}{2S} \left(\sqrt{1 + \frac{4S^2}{\Gamma^2\mu^2}} - 1 \right) \end{aligned} \quad (3.31)$$

to recover the primitive variables, as suggested by Neilsen [78].

The evolved variables for the full Medium system are then [17]

$$e^{2a}, a_{,t}, a_{,x}, b_{,t}, b_{,x}, c_{,t}, c_{,x}, a_{,tx}, a_{,xx}, b_{,tx}, b_{,xx}, c_{,tx}, c_{,xx}, \tau, S, \quad (3.32)$$

so that second derivatives of the metric are also evolved.

The full set of equations is given by

$$\begin{aligned} \partial_t(e^{2a}) &= 2a_{,t}e^{2a} \\ \partial_t(a_{,t}) &= a_{,xx} + b_{,t}^2 - b_{,x}^2 - c_{,t}^2 + c_{,x}^2 - \frac{1}{2}\kappa e^{2a}(\tau - (Sv + P)) \\ \partial_t(a_{,x}) &= a_{,tx} \\ \partial_t(b_{,t}) &= b_{,xx} - 2b_{,t}^2 + 2b_{,x}^2 + \frac{1}{2}\kappa e^{2a}(\tau - (Sv + P)) \\ \partial_t(b_{,x}) &= b_{,tx} \\ \partial_t(c_{,t}) &= c_{,xx} - 2(b_{,t}c_{,t} - b_{,x}c_{,x}) \\ \partial_t(c_{,x}) &= c_{,tx} \\ \partial_t(a_{,tx}) + \partial_x(-a_{,xx} + \frac{1}{2}\kappa e^{2a}(\tau - (Sv + P))) &= 2b_{,t}b_{,tx} - 2b_{,x}b_{,xx} - 2c_{,t}c_{,tx} + 2c_{,x}c_{,xx} \\ \partial_t(a_{,xx}) + \partial_x(-a_{,tx}) &= 0 \\ \partial_t(b_{,tx}) + \partial_x(-b_{,xx} - \frac{1}{2}\kappa e^{2a}(\tau - (Sv + P))) &= -4b_{,t}b_{,tx} + 4b_{,x}b_{,xx} \\ \partial_t(b_{,xx}) + \partial_x(-b_{,tx}) &= 0 \\ \partial_t(c_{,tx}) + \partial_x(-c_{,xx}) &= -2(b_{,t}c_{,tx} - b_{,x}c_{,xx} + b_{,tx}c_{,t} - b_{,xx}c_{,x}) \\ \partial_t(c_{,xx}) + \partial_x(-c_{,tx}) &= 0 \\ \partial_t\tau + \partial_x S &= -\tau(a_{,t} + 2b_{,t}) - 2S(a_{,x} + b_{,x}) - (Sv + P)a_{,t} - 2Pb_{,t} \\ \partial_t S + \partial_x(Sv + P) &= -\tau a_{,x} - 2S(a_{,t} + b_{,t}) - (Sv + P)(a_{,x} + 2b_{,x}) + 2Pb_{,x}. \end{aligned} \quad (3.33)$$

The constraint equations in this system are algebraic in the variables (3.32):

$$\begin{aligned} 2a_{,t}b_{,t} + 2a_{,x}b_{,x} + b_{,t}^2 - 2b_{,xx} - 3b_{,x}^2 - c_{,t}^2 - c_{,x}^2 &= 8\pi e^{2a}\tau \\ \text{and } -2a_{,t}b_{,x} - 2a_{,x}b_{,t} + 2b_{,tx} + 2b_{,t}b_{,x} + 2c_{,t}c_{,x} &= 8\pi e^{2a}S. \end{aligned} \quad (3.34)$$

We define the actual constraints to be

$$\begin{aligned} \mathcal{H} &= e^{2b}(2a_{,t}b_{,t} + 2a_{,x}b_{,x} + b_{,t}^2 - 2b_{xx} - 3b_{,x}^2 - c_{,t}^2 - c_{,x}^2 - 8\pi e^{2a}\tau) \\ \text{and } \mathcal{M} &= e^{2b}(-2a_{,t}b_{,x} - 2a_{,x}b_{,t} + 2b_{,tx} + 2b_{,t}b_{,x} + 2c_{,t}c_{,x} - 8\pi e^{2a}S). \end{aligned} \quad (3.35)$$

Note that we omit the exponential factor e^{2b} when plotting them as determining this would require a numerical integration of the evolved variables. Barnes *et al.* made the same alteration for their plots in [16] (Barnes - personal communication).

We use the two sets of initial conditions as given in [16], the first being given by:

$$\begin{aligned} v = S = a_{,t} = b_{,t} = 0, \quad a = -2, \quad c_s^2 = \frac{1}{3}, \\ \kappa\mu = \begin{cases} 100 & \text{if } x < 0, \\ 1 & \text{if } x > 0. \end{cases} \\ b_{,x} = -\sqrt{\frac{\kappa e^{2a}\tau}{3}} \tan\left(\frac{x}{2}\sqrt{3\kappa e^{2a}\tau} + \sqrt{3\kappa e^{2a}\tau}\right) \quad \text{where } \kappa = 8\pi, \end{aligned} \quad (3.36)$$

where the expression for $b_{,x}$ has been found analytically, with the conditions that it be continuous at $x = 0$ and that the singularity be well outside the domain. The data is evolved to a time $t = 0.5$.

The second test case is given by

$$\begin{aligned} a_{,t} = 0, \quad a = -3, \quad c_s^2 = \frac{1}{3}, \\ \mu = \frac{40}{\kappa}(1.1 + \tanh(-40x)), \quad v = \frac{20}{\tanh(-40x)} \end{aligned} \quad (3.37)$$

where, in order to complete the initial conditions, we solve numerically for $b_{,t}$ and $b_{,x}$, using a Runge-Kutta second-order integrator, specifying $b_{,t}(x = 0) = b_{,x}(x = 0) = 0$ as a point from which to integrate outwards to the edges of the domain. This initial data is then evolved to time $t = 0.3$.

3.5 Bona-Massó formulation

Although the BSSN formulation [18] is currently the one that is most used in successful evolutions, we here consider the Bona-Massó formulation [22], which is written in a form more amenable to the solution techniques we wish to use.

The Bona-Massó formulation is written in first-order flux-conservative form, with source terms, using the 3+1 splitting approach (2.1). It has been shown to be strongly hyperbolic, and so can form part of a well-posed initial-value problem. Since Einstein's equations are second-order, it is necessary to introduce some auxiliary variables involving first derivatives of the main variables:

$$\begin{aligned} A_i &= \frac{\alpha_{,i}}{\alpha}, \quad D_{kij} = \frac{1}{2}\gamma_{ij,k} \\ V_i &= D_{is}^s - D_{si}^s, \quad B_i^k = \frac{1}{2}\beta^k_{,i}. \end{aligned} \quad (3.38)$$

The time evolution is governed by the following equations [22]:

$$\partial_t \gamma_{ij} = -2\alpha(K_{ij} - s_{ij}) + 2\beta^r D_{rij}, \quad (3.39)$$

$$\partial_t \alpha = -\alpha^2 Q + \alpha \beta^r A_r, \quad (3.40)$$

$$\partial_t D_{kij} + \partial_k(-\beta^r D_{rij} + \alpha(K_{ij} - s_{ij})) = 0, \quad (3.41)$$

$$\partial_t A_k + \partial_k(-\beta^r A_r + \alpha Q) = 0, \quad (3.42)$$

$$\begin{aligned} \partial_t K_{ij} + \partial_k \left(-\beta^k K_{ij} + \alpha [D_{ij}^k - \frac{n}{2} V^k \gamma_{ij} \right. \\ \left. + \frac{1}{2} \delta_i^k (A_j + 2V_j - D_{jr}{}^r) + \frac{1}{2} \delta_j^k (A_i + 2V_i - D_{ir}{}^r) \right) &= 2(K_{ir} B_j{}^r + K_{jr} B_i{}^r - K_{ij} B_r{}^r) + \alpha [-(4)R_{ij} - 2K_i{}^k K_{kj} \\ &+ \text{tr}(K) K_{ij} - \Gamma_r{}^k \Gamma_k{}^r{}_j + 2D_{ik}{}^r D_{rj}{}^k + 2D_{jk}{}^r D_{ri}{}^k \\ &+ \Gamma_k{}^k{}_r \Gamma_i{}^r{}_j - (2D_{kr}{}^k - A_r)(D_{ij}{}^r + D_{ji}{}^r) \\ &+ A_i(V_j - \frac{1}{2} D_{jk}{}^k) + A_j(V_i - \frac{1}{2} D_{ik}{}^k) - nV^k D_{kij}] \\ &+ \frac{n}{4} \alpha \gamma_{ij} [-D_k{}^{rs} \Gamma_r{}^k{}_s + D_{kr}{}^r D^{ks}{}_s - 2V^k A_k \\ &+ K^{rs} K_{rs} - (\text{tr} K)^2 + 2\alpha^2 G^{00}], \end{aligned} \quad (3.43)$$

$$\begin{aligned} \partial_t V_i + \partial_k(-\beta^k V_i + B_i{}^k - B_i{}^k) &= \alpha [\alpha G_i^0 + A_r (K_i{}^r - \text{tr} K \delta_i^r) + K_r{}^s (D_{ir}{}^s - 2D_{ri}{}^s) \\ &- K_i{}^r (D_{rs}{}^s - 2D_{sr}{}^s)] + 2(B_i{}^r - \delta_i^r \text{tr} B) V_r \\ &+ 2(D_{ri}{}^s - \delta_i^s D_{jr}{}^j) B_s{}^r. \end{aligned} \quad (3.44)$$

We note that there is an error in [22], equation (13) which has an extra $A_j(V_i - \frac{1}{2} D_{ik}{}^k)$ term as compared to equation (3.43) here, which destroys the symmetry in i and j .

The parameter n is set to be 1 for our purposes. This corresponds to adding the constraints (2.4) to the equations (3.39-3.44) as compared to when $n = 0$. The advantage of this is that when we tend to the Newtonian limit, small relativistic corrections are not masked by large Newtonian terms, which could result in a loss of accuracy when evolving with numerical methods that introduce truncation errors [21]. The slicing condition Q is defined to be

$$Q = \text{tr} K = K_i{}^i \quad (3.45)$$

for our purposes, giving the harmonic slicing condition, although an extra factor dependent on the lapse α can be introduced. Also, for convenience, and following [22], we have written

$$s_{ij} = \frac{1}{\alpha} (B_{ij} + B_{ji}), \quad (3.46)$$

where the indices of $B_i{}^j$ have been lowered using the three-metric, even though it is not a tensor.

For later use, we will show how the space-time variables can be transformed between coordinate systems. This is not entirely trivial, as some of the quantities are not tensorial in nature. If we wish

to transform between the 3-coordinate systems x^i and r^i , with unprimed quantities in the x^i basis, and primed quantities in the r^i basis, then we can proceed as follows:

The general form of the transformation for a tensor $C^{i\dots k}_{m\dots p}$ is given by

$$C'^{a\dots c}_{d\dots f} = \frac{\partial r^i}{\partial x^a} \dots \frac{\partial r^k}{\partial x^c} \frac{\partial x^m}{\partial r^d} \dots \frac{\partial x^p}{\partial r^f} C^{i\dots k}_{m\dots p}, \quad (3.47)$$

and this applies to the tensors γ_{ij} , K_{ij} , β^i , A_i , and V_i . Although it is not obvious that V_i is a tensor, this can be shown by deriving its transformation law from that of D_{ijk} and finding that it is as would be expected for a tensor.

The transformations for the non-tensors are easily derived, and take the forms are:

$$\begin{aligned} D'_{kij} &= \frac{\partial x^r}{\partial r^k} \frac{\partial x^p}{\partial r^i} \frac{\partial x^q}{\partial r^j} D_{rpq} + \frac{1}{2} \frac{\partial^2 x^p}{\partial r^k \partial r^i} \frac{\partial x^q}{\partial r^j} g_{pq} + \frac{1}{2} \frac{\partial^2 x^p}{\partial r^k \partial r^j} \frac{\partial x^q}{\partial r^i} g_{pq} \\ \text{and } B'^j_i &= \frac{\partial r^j}{\partial x^k} \frac{\partial x^p}{\partial r^i} B_p^k + \frac{1}{2} \frac{\partial x^p}{\partial r^i} \frac{\partial^2 r^j}{\partial x^p \partial x^k} \beta^k. \end{aligned} \quad (3.48)$$

We use these transformations when moving between a computational grid, usually expressed as a unit cube $[0, 1]^3$, and physical coordinate space. The first derivatives of the coordinate transformations in both directions are provided by Overture in this case, and we can calculate the second derivatives as required by using second order finite-differencing from the first derivatives.

We note that the constraints (2.4) can be calculated from these variables since the covariant derivative ∇_i can be found using D_{kij} to calculate the Christoffel symbols, and the Ricci scalar (1.6) can be found as in [22]

$${}^{(3)}R = -2\partial_k V^k + D_k{}^{rs} \Gamma_r{}^k{}_s - D_{kr}{}^r D^{ks}{}_s. \quad (3.49)$$

3.6 Fluid validation tests

In this section we describe some tests which we use to validate the fluid evolution parts of the code. They are all carried out in flat (Minkowski) space-time.

3.6.1 One-dimensional Riemann problems

A Riemann problem is one for which the initial data is given by two constant states:

$$\mathbf{u} = \begin{cases} \mathbf{u}_L & \text{if } x < x_0, \\ \mathbf{u}_R & \text{if } x > x_0. \end{cases} \quad (3.50)$$

For many systems of equations of interest to fluid dynamicists, it is possible to solve such a problem for any (physically valid) states on the left and right sides. For example, the complete solution to the

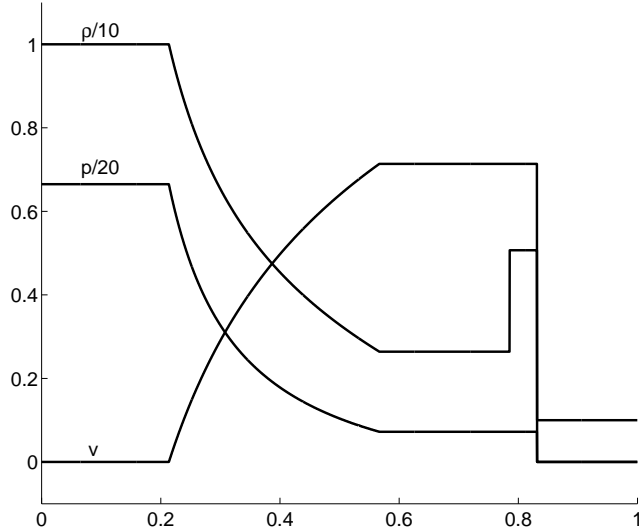


Figure 3.1: *Exact solution for the mild shock Riemann problem given by equation (3.51) at time $t = 0.4$. The solution is composed of a rarefaction, a contact discontinuity, and a shock.*

Riemann problem for the non-relativistic Euler equations can be found in Toro [109]. The approach found therein can then be extended (albeit with some difficulty) to other systems of equations.

An exact solution to the Riemann problem for the one-dimensional special relativistic hydrodynamical equations exists (see [84]), and so numerical evolutions can be compared to this.

The test-cases we use are taken from Del Zanna and Bucciantini [117], so that a comparison can be made with their numerical results, which were obtained using a Convex ENO reconstruction and the HLL Riemann solver.

The three test-cases all use an ideal fluid with adiabatic index $\Gamma = 5/3$ and are performed on a domain of $x \in [-\frac{1}{2}, \frac{1}{2}]$. In order to generate the exact solutions, we use the code `riemann.f` from the Living Review by Martí and Müller [74].

Mildly Relativistic Shock

The initial conditions for this test are

$$(\rho, v, p) = \begin{cases} (10, 0, 13.3) & \text{if } x < \frac{1}{2}, \\ (1, 0, 0) & \text{if } x > \frac{1}{2}. \end{cases} \quad (3.51)$$

It is fairly easy to obtain good results for this simple test, as it contains no extreme velocities or densities. It is evolved to a time $t = 0.4$. The exact solution for all variables is shown in Figure 3.1.

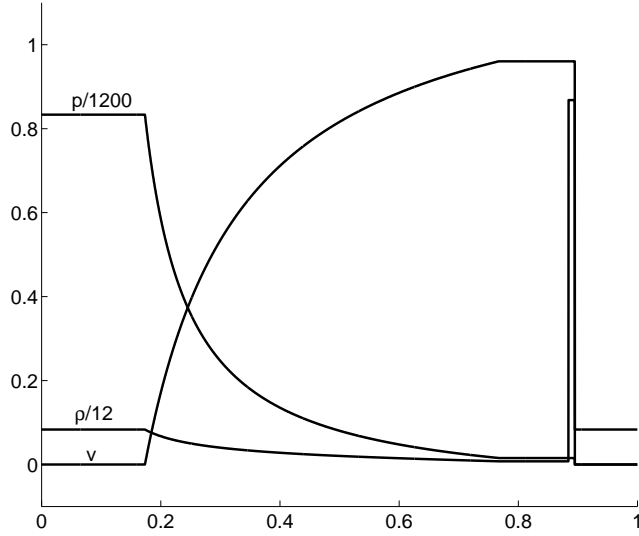


Figure 3.2: *Exact solution for the blast-wave Riemann problem given by equation (3.52) at time $t = 0.4$. The solution is composed of a rarefaction, a contact discontinuity, and a shock. The maximum velocity is $v \approx 0.96040961$, corresponding to a Lorentz factor of $W \approx 3.59$.*

Strongly Relativistic Blast-Wave

The initial data for this problem is given by

$$(\rho, v, p) = \begin{cases} (1, 0, 1000) & \text{if } x < \frac{1}{2}, \\ (1, 0, 0.01) & \text{if } x > \frac{1}{2}. \end{cases} \quad (3.52)$$

This is a more difficult test-case to evolve, as the solution contains a narrow peak of high density, which is hard to capture with a low resolution, and also contains velocities which approach $c = 1$. This too is evolved to a time $t = 0.4$. The exact solution for all variables is shown in Figure 3.2.

Perturbed Density Test

The initial data for this problem is given by

$$(\rho, v, p) = \begin{cases} (5, 0, 50) & \text{if } x < \frac{1}{2}, \\ (2 + 0.3 \sin(50x), 0, 5) & \text{if } x > \frac{1}{2}. \end{cases} \quad (3.53)$$

This is not a Riemann problem but provides a good indicator of how well a numerical method can capture extrema. There is no analytic solution to this, and instead we use a converged numerical solution, generated at high-resolution. This test is evolved to $t = 0.35$, since Del Zanna and Bucciantini's results

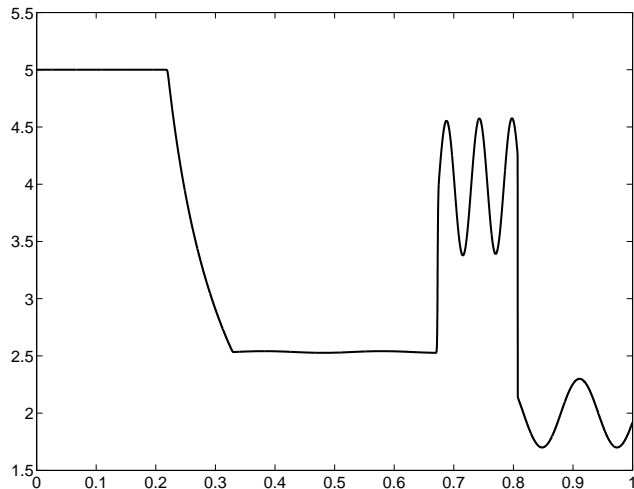


Figure 3.3: *Converged solution to the density perturbation test given in equation (3.53) at time $t = 0.35$, where only the density ρ is shown. The three peaks are particularly hard for a numerical method to capture accurately.*

are consistent with this, not $t = 0.4$ as stated in their main text. The converged solution, evolved using the SLIC¹ scheme with van Leer limiter (see §4.2), and 20 000 cells, is shown in Figure 3.3. We believe this to be sufficiently converged as the solution with 20 000 cells is visually indistinguishable from that with 10 000 cells.

3.6.2 Two-dimensional Riemann problem

We can also test the multi-dimensionalisation of our schemes in a simple two-dimensional test, which is essentially four separate Riemann problems interacting with each other. Due to the two-dimensional nature of the setup, there is no exact solution to this problem, but we can compare our results to those of other authors such as Del Zanna and Bucciantini [117]. The problem is defined on a region $[0, 1] \times [0, 1]$ and is evolved to a time $t = 0.4$.

$$(\rho, v_x, v_y, p) = \begin{cases} (0.5, 0, 0, 1) & \text{if } x < \frac{1}{2} \text{ and } y < \frac{1}{2}, \\ (0.1, 0, 0.99, 1) & \text{if } x > \frac{1}{2} \text{ and } y < \frac{1}{2}, \\ (0.1, 0.99, 0, 1) & \text{if } x < \frac{1}{2} \text{ and } y > \frac{1}{2}, \\ (0.1, 0, 0, 0.01) & \text{if } x > \frac{1}{2} \text{ and } y > \frac{1}{2}. \end{cases} \quad (3.54)$$

This setup is symmetric about the line $x = y$, so that the evolved solution should maintain this symmetry.

3.7 Propagating Gauge Pulse

The propagating gauge pulse metric is taken from [97] and is given by

$$\gamma_{ij} = \text{diag}(e^b, 1, 1), \quad \alpha = e^{b/2}, \quad K_{xx} = \frac{1}{2} b_{,x} e^{b/2}, \quad D_{xxx} = \frac{1}{2} b_{,x} e^b, \quad \text{and} \quad A_x = \frac{1}{2} b_{,x}. \quad (3.55)$$

All other metric variables are zero, and $b = \sin(\pi(x - t))$.

A simple variation of this is given by introducing a constant shift so that

$$\gamma_{ij} = \text{diag}(e^b, 1, 1), \quad \alpha = e^{b/2}, \quad K_{xx} = \frac{1}{2} b_{,x} e^{b/2}, \quad D_{xxx} = \frac{1}{2} b_{,x} e^b, \quad A_x = \frac{1}{2} b_{,x}, \quad \text{and} \quad \beta^x = 1. \quad (3.56)$$

All other metric variables are zero, and now $b = \sin(\pi x)$, i.e. we have a static metric.

3.8 Apples with Apples tests

Here we define some of the Apples with Apples tests [11], referred to in §2.4, that we will use to validate our metric evolution. The tests are generally characterised by a parameter ρ , which defines the resolution of the test grid so as to allow for convergence testing. As this will be relevant for two of the tests, we note that double precision arithmetic has a machine epsilon of $\epsilon = 2.2 \times 10^{-16}$, where ϵ is the smallest positive number such that $1 + \epsilon > 1$ in double precision arithmetic.

3.8.1 Robustness test

The first test that can be performed is that of robustness. Einstein's equations are highly non-linear and, even if they perform well on data that satisfies the constraints (2.4), they may cause departures from the constraints to grow without limit, obliterating the required solution. These original departures may be simply due to finite numerical accuracy.

We therefore evolve Minkowski space, with perturbations of order ϵ , where $1 + \epsilon^2 = 1$ in double precision arithmetic, taking $\epsilon = 10^{-10}$. We add to each space-time variable (i.e. those given in §3.5) a uniformly distributed value in the range $[-10^{-10}/\rho, 10^{-10}/\rho]$. The grid has limits

$$\left[-\frac{1}{2}, \frac{1}{2}\right] \times [-1/(20\rho), 1/(20\rho)] \times [-1/(20\rho), 1/(20\rho)], \quad (3.57)$$

with five grid cells in the y and z directions, and 50ρ cells in the x direction, so that the grid cells are cubic. The time-stepping constraint is chosen to be $\Delta t = 1/(100\rho)$, corresponding to a CFL condition of $c = 1/2$.

The constraints will not be satisfied initially, as we do not insist, for example, that $D_{xxx} = \frac{1}{2}\gamma_{xx,x}$, but allow each variable to vary independently of the others. However, this is close to what might be encountered due to numerical noise in a realistic evolution.

3.8.2 Linearized wave test

This test is that of a small amplitude (linearized) gravitational wave propagating around a periodic domain. For once, limited computational accuracy works in our favour, in that if $A = 10^{-8}$, and A is stored in double precision, then $1 + A^2 = 1$, as required for a linearized solution.

The initial conditions for the non-zero variables are given by

$$\begin{aligned} \gamma_{xx} = 1, \gamma_{yy} = 1 + b, \gamma_{zz} = 1 - b, \alpha = 1, K_{yy} = \frac{1}{2}b_{,t}, K_{zz} = -\frac{1}{2}b_{,t}, \\ D_{xyy} = \frac{1}{2}b_{,x}, \text{ and } D_{xzz} = -\frac{1}{2}b_{,x}, \end{aligned} \quad (3.58)$$

where

$$b = A \sin(2\pi(x - t)), \quad (3.59)$$

and the simulation parameters are given by

$$\begin{aligned} \text{Domain : } x \in \left[-\frac{1}{2}, +\frac{1}{2}\right], y = 0, z = 0, \\ \text{Grid : } x^i = -\frac{1}{2} + (n - \frac{1}{2}\Delta x), n = 1 \dots 50\rho, \Delta x = 1/(50\rho), \rho \in [1, 2, 4] \\ \text{Timestep : } \Delta t = \Delta x/4 = 1/(200\rho) \end{aligned} \quad (3.60)$$

The system should be evolved up to time $T = 1000$. We note that the specified time-step is a quarter of the maximum possible given by the stability criterion. The space-time is linearized about Minkowski space, giving the speed of light to be 1 up to an error of order 10^{-8} , so that the stability limit is given by $\Delta t = \Delta x$. For the methods we will be using, we should not be surprised if we see very diffusive results indeed as a result of the many extra time-steps used, with their associated truncation error.

3.8.3 Gowdy wave

We use the backwards evolution of the Gowdy Wave, with harmonic slicing. The initial conditions for this test case are given by

$$\begin{aligned} \gamma_{xx} = t^{-1/2}e^{\lambda/2}, \gamma_{yy} = te^P, \gamma_{zz} = te^{-P}, \alpha = t^{-1/4}e^{\lambda/4} \\ K_{xx} = \frac{1}{4}t^{-1/4}e^{\lambda/4}(t^{-1} - \lambda_{,t}), K_{yy} = -\frac{1}{2}t^{1/4}e^{-\lambda/4}e^P(1 + tP_{,t}), \\ K_{zz} = -\frac{1}{2}t^{1/4}e^{-\lambda/4}e^{-P}(1 - tP_{,t}), \end{aligned} \quad (3.61)$$

$$\text{where } P = J_0(2\pi t) \cos(2\pi x),$$

$$\begin{aligned} \lambda = -2\pi t J_0(2\pi t) J_1(2\pi t) \cos^2(2\pi x) + 2\pi^2 t^2 [J_0^2(2\pi t) + J_1^2(2\pi t)] - \\ \frac{1}{2} \{ (2\pi)^2 [J_0^2(2\pi) + J_1^2(2\pi)] - 2\pi J_0(2\pi) J_1(2\pi) \} \end{aligned}$$

where J_0 and J_1 are Bessel functions. We note that we took the code for computing the Bessel functions and their derivatives from the Cactus code [5] (translated into C).

Following [44], we re-parameterize the time coordinate as

$$t = F(\tau) = ke^{c\tau}, \quad (3.62)$$

where $c \approx 0.0021195119214617$ and $k \approx 9.6707698127638$,

and we use an initial time $\tau_0 = t_0 \approx 9.8753205829098$.

The problem should be run on a one-dimensional grid $x \in [-0.5, +0.5]$, $\Delta x = 1/(50\rho)$, $\Delta t = \Delta x/4$, where $\rho = 1, 2, 4$. We evolve this problem up to a time of $t = 1000$ if that is possible before the code crashes due to large numerical error.

3.9 Black Hole metrics

Since our intention is to use overlapping grids to excise singularities, we shall need some appropriate space-times. These will be those containing one or more black holes. It is well-known that, in the absence of electromagnetic fields, a black hole can be specified uniquely by its mass and spin. However, the resulting space-time can still be written in several coordinate systems. We have used MapleTM to find the necessary metric derivatives for the coordinate systems, and then to generate C code from the resulting expressions. This reduces the likelihood of coding errors when calculating the metrics.

3.9.1 Kerr-Schild coordinates

The most general black hole we will consider here is the Kerr black hole, with spin, but no charge. The mass of the black hole is M and its dimensionless spin parameter is a where $|a| < 1$. The metric can be expressed in Cartesian coordinates as follows [63, 77]:

$$\begin{aligned} g_{\mu\nu} &= \eta_{\mu\nu} - 2Vl_\mu l_\nu, \\ \alpha &= \frac{1}{\sqrt{1-2V}}, \quad \beta^i = \frac{2V}{1-2V}l^i, \quad \gamma_{ij} = \delta_{ij} - 2Vl_i l_j, \\ K_{ij} &= \alpha(-l_i V_{,j} - l_j V_{,i} - V l_{i,j} - V l_{j,i} + 2V^2(l_i l^m l_{j,m} + l_j l^m l_{i,m}) + 2V l_i l_j l^m V_{,m}), \end{aligned} \quad (3.63)$$

where $V = -\frac{MR^3}{R^4 + M^2 a^2 z^2}$, $l_i = l^i = \left(-\frac{Rx + aMy}{R^2 + M^2 a^2}, -\frac{Ry - aMx}{R^2 + M^2 a^2}, -\frac{z}{R}\right)$,
and $R^2 = \frac{1}{2} \left(\|\mathbf{x}\|^2 - M^2 a^2 + \sqrt{(\|\mathbf{x}\|^2 - M^2 a^2)^2 + 4M^2 a^2 z^2} \right)$,

and this is the form that we use.

There is a more standard form of the metric, in Kerr-Schild spherical coordinates, defined by (see Boyer and Lindquist [24])

$$\begin{aligned} x &= (r \cos \phi - a \sin \phi) \sin \theta \equiv \sqrt{r^2 + a^2} \sin \theta \cos \left(\phi - \tan^{-1} \frac{a}{r} \right), \\ y &= (r \sin \phi + a \cos \phi) \sin \theta \equiv \sqrt{r^2 + a^2} \sin \theta \sin \left(\phi - \tan^{-1} \frac{a}{r} \right), \\ z &= r \cos \theta, \end{aligned} \quad (3.64)$$

where the 3 + 1 split of the metric now takes the form

$$\begin{aligned}\gamma_{rr} &= 1 + \frac{2Mr}{R^2}, \quad \gamma_{r\phi} = -Ma \sin^2 \theta \left(1 + \frac{2Mr}{R^2}\right), \quad \gamma_{\theta\theta} = R^2, \\ \gamma_{\phi\phi} &= \left(r^2 + M^2 a^2 + \frac{2M^3 r a^2}{R^2} \sin^2 \theta\right) \sin^2 \theta, \\ \alpha^2 &= \frac{1}{1 + 2Mr/R^2}, \quad \beta^r = \frac{2Mr}{R^2} \alpha^2,\end{aligned}\tag{3.65}$$

The space-time has an outer horizon at $r = r_+$ and an inner horizon at $r = r_-$, where

$$r_{\pm} = M(1 \pm \sqrt{1 - a^2}).\tag{3.66}$$

3.9.2 Boyer-Lindquist coordinates

Another way of writing the same space-time is in Boyer-Lindquist spherical coordinates, where

$$\begin{aligned}x &= \sqrt{r^2 + a^2} \sin \theta \cos \phi, \\ y &= \sqrt{r^2 + a^2} \sin \theta \sin \phi, \\ z &= r \cos \theta,\end{aligned}\tag{3.67}$$

which differ from Kerr-Schild spherical coordinates only in the ϕ coordinate when $a \neq 0$. In these coordinates, the metric becomes (again, see Boyer and Lindquist [24] and Kelly [63])

$$\begin{aligned}\gamma_{rr} &= \frac{R^2}{\Delta}, \quad \gamma_{\theta\theta} = R^2, \quad \gamma_{\phi\phi} = \frac{\Omega}{R^2} \sin^2 \theta, \quad \alpha^2 = \frac{\Delta R^2}{\Omega}, \quad \beta^\phi = -\frac{2aMr}{\Omega}, \\ \text{where } \Delta &= r^2 + a^2 - 2Mr, \quad \text{and } \Omega = R^2(r^2 + a^2) + 2M^3 a^2 r \sin^2 \theta.\end{aligned}\tag{3.68}$$

This form of the metric has coordinate singularities where $\Delta = 0$, corresponding to the outer and inner horizons. The metric therefore does not extend continuously to inside the event horizon of the black hole, in the same way as the Schwarzschild coordinates, which are a specialisation of the Boyer-Lindquist coordinates for $a = 0$.

3.9.3 Transformation between Kerr-Schild and Boyer-Lindquist systems

We shall also need to transform velocity vectors between the Kerr-Schild and Boyer-Lindquist coordinate systems. The transformations between Cartesian and spherical coordinate systems are achieved via the Jacobians

$$h^i_j = \frac{\partial r^i}{\partial x^j} \quad \text{and} \quad (h^{-1})^j_i = \frac{\partial x^j}{\partial r^i},\tag{3.69}$$

where r^i are the spherical coordinates (given by (3.64) or (3.67)) and x^j are the Cartesian coordinates. One-tensors are then transformed from the Cartesian basis to the spherical basis by

$$v^i_{\text{sph}} = h^i_j v^j_{\text{Cart}}, \quad \text{and} \quad w^{\text{sph}}_i = (h^{-1})^j_i w^{\text{Cart}}_j,\tag{3.70}$$

and from the spherical basis to the Cartesian basis by

$$v_{\text{Cart}}^i = (h^{-1})_j^i v_{\text{sph}}^j, \quad \text{and} \quad w_j^{\text{Cart}} = h_j^i w_i^{\text{sph}}. \quad (3.71)$$

The transformation between Kerr-Schild and Boyer-Lindquist spherical coordinates outside the outer event horizon is then given by Font, Ibáñez, and Papadopoulos [42] as follows:

$$\begin{aligned} v_{\text{BL}}^r &= \Psi \left(v_{\text{KS}}^r - \frac{\beta_{\text{KS}}^r}{\alpha_{\text{KS}}} \right) \\ v_{\text{BL}}^\theta &= \Psi v_{\text{KS}}^\theta \\ v_{\text{BL}}^\phi &= \Psi \left(v_{\text{KS}}^\phi - \frac{Ma}{\Delta} v_{\text{KS}}^r \right) - \Psi \left(\frac{\beta_{\text{KS}}^\phi}{\alpha_{\text{KS}}} - \frac{Ma}{\Delta} \frac{\beta_{\text{KS}}^r}{\alpha_{\text{KS}}} \right) + \frac{\beta_{\text{BL}}^\phi}{\alpha_{\text{BL}}}, \\ \text{where } \Psi &= \frac{W_{\text{KS}}}{W_{\text{BL}}} = \left(\frac{\alpha_{\text{BL}}}{\alpha_{\text{KS}}} - \frac{2Mr}{\Delta} \alpha_{\text{BL}} \left(v_{\text{KS}}^r - \frac{\beta_{\text{KS}}^r}{\alpha_{\text{KS}}} \right) \right)^{-1}. \end{aligned} \quad (3.72)$$

where the subscripts KS and BL refer to variables expressed in Kerr-Schild and Boyer-Lindquist spherical coordinates, respectively, and where $v_{\text{KS}}^i, v_{\text{BL}}^i$ are the components of the Eulerian velocity, related to the proper velocity by (3.10). Note that Ψ here is the reciprocal of that defined in [42]; however, the expressions given here are correct.

3.9.4 Binary black holes

No analytic space-time metric encoding a complete time-evolution of two general Kerr black holes has yet been found (indeed, determining such evolutions is a major aim of NR). We therefore require an initial metric which resembles such a setup for widely separated Kerr black holes. One solution is to impose two superposed Kerr solutions of the form given in (3.63). However, we would then need to solve the constraints numerically, and we have not included such a solver as part of our algorithm, although we note that Overture does have the ability to solve elliptic equations on general overlapping grids.

Instead, we use Brill-Lindquist initial data, which encapsulates two (non-spinning) black holes with arbitrary masses m_1 and m_2 , at positions \mathbf{x}_1 and \mathbf{x}_2 . The metric is then given by [61]:

$$\begin{aligned} ds^2 &= - \left(\frac{1 - \psi/8}{1 + \phi/2} \right)^2 dt^2 + (1 + \phi/8)^4 (dx^2 + dy^2 + dz^2), \\ \text{where } \phi &= 8 \left(\frac{\alpha_1}{r_1} + \frac{\alpha_2}{r_2} \right), \quad \psi = 8 \left(\frac{\beta_1}{r_1} + \frac{\beta_2}{r_2} \right), \\ \alpha_1 &= -\frac{1}{4}(2r_{12} + m_2 - m_1) + \frac{r_{12}}{4} \sqrt{4 + \frac{4}{r_{12}}(m_1 + m_2) + \left(\frac{m_1 - m_2}{r_{12}} \right)^2}, \\ \alpha_2 &= -\frac{1}{4}(2r_{12} + m_1 - m_2) + \frac{r_{12}}{4} \sqrt{4 + \frac{4}{r_{12}}(m_1 + m_2) + \left(\frac{m_1 - m_2}{r_{12}} \right)^2}, \\ \beta_1 &= \alpha_1 \frac{r_{12} + \alpha_1 - \alpha_2}{r_{12} + \alpha_1 + \alpha_2}, \quad \beta_2 = \alpha_2 \frac{r_{12} + \alpha_2 - \alpha_1}{r_{12} + \alpha_1 + \alpha_2}, \\ r_{12} &= \|\mathbf{x}_1 - \mathbf{x}_2\|, \quad r_1 = \|\mathbf{x} - \mathbf{x}_1\|, \quad r_2 = \|\mathbf{x} - \mathbf{x}_2\|. \end{aligned} \quad (3.73)$$

The extrinsic curvature K_{ij} is zero, as is the shift vector β^i . When evolved, this initial data gives rise to a head-on collision of two black holes.

3.9.5 Apparent Horizon of a black hole

We also define the apparent horizon of a black hole to be the outer boundary of the union of all trapped surfaces, which are smooth closed 2-surfaces such that future-pointing outgoing null geodesics have negative expansion

$$\Theta = \nabla_i s^i + K_{ij} s^i s^j - \text{tr}K < 0, \quad (3.74)$$

where s^i is the outward pointing unit 3-vector normal to the surface [107].

The important properties of the apparent horizon are that it is defined *locally* on a hypersurface, i.e. without reference to the gauge, and that it is inside or coincident with the event horizon of a black-hole [107]. The more familiar event horizon, in contrast, is defined in terms of whether or not any future-pointing null-geodesic can reach spatial infinity. Its location requires knowledge of the entire space-time and thus cannot be calculated when we are already trying to calculate the full space-time. We can therefore use the location of an apparent horizon as a surface from the inside of which we can safely assume that no physical information can emerge. In the case of stationary black-holes, such as we use in this thesis, the event horizon is coincident with the apparent horizon.

Chapter 4

Numerical Methods

In this chapter we describe in detail the numerical schemes we have included in our algorithm. Most of the material herein is based on established numerical methods widely used in computational fluid dynamics. Suitable references for a more in-depth guide to these methods are the books of Laney [67], LeVeque [69], and Toro [109].

Some methods described in this chapter, particularly the MUSTA approach, are less well-known, and we include references to specific papers where appropriate. Further, the analysis of the MUSTA approach and the generalisation of the SLIC scheme to a curvi-linear grid and to a non-flat metric are our own.

4.1 Choice of numerical methods

The primary aim of this thesis is to develop a numerical algorithm to evolve a space-time metric and a fluid with reasonable accuracy and in a computationally efficient manner. We therefore detail here the reasoning which led to the choices of numerical methods presented in this section.

4.1.1 Evolution schemes

Firstly, the overall methodology for the evolution must be determined. Our main choice here is between finite-difference and finite-volume methods. We could also consider spectral methods, which are often used for the evolution of the metric (see [30]), but these are certainly not suitable for the evolution of a fluid, as they require functions to be smooth.

Finite-difference schemes approximate derivatives of variables using an approach such as the approximation

$$\frac{\partial u}{\partial x} \approx \frac{u(x + \Delta x) - u(x - \Delta x)}{2\Delta x}, \quad (4.1)$$

which is second-order accurate in Δx if u is a smooth function of x .

Although this can easily be extended to higher orders of accuracy, there is the problem that derivatives are undefined at discontinuities, so that the approximation (4.1) cannot converge to $\partial u/\partial x$ as $\Delta x \rightarrow 0$. This can be overcome by the construction of conservative and shock-capturing finite-difference schemes, such as the conservative finite-difference WENO schemes presented by Xing and Shu [114], but we do not use these approaches here.

Since we wish to evolve fluids, which can develop shocks even from an initially smooth flow, basic finite-difference methods without conservative or shock-capturing corrections are not appropriate for use here. However, the space-time metric is known to be at least twice differentiable (two derivatives are required for the definition of the Einstein tensor), and so it may be possible to use finite-difference methods for this. In fact, we use finite-difference methods as a comparison with other works in testing the metric evolution. However, the most that we can say about $g_{\mu\nu}$ is that it has a second derivative. That this derivative is not necessarily continuous is clear from the fact that $G_{\mu\nu}$, a combination of second derivatives of $g_{\mu\nu}$, is set equal to $8\pi T^{\mu\nu}$, which contains the fluid variables. Since the fluid variables may be discontinuous, the second derivatives of $g_{\mu\nu}$ may also be discontinuous, so that the first derivatives, which include variables that we are evolving, will then have obvious kinks in them, and so not be suitable for evolution with finite-difference schemes.

Finite-volume methods, on the other hand, are based on the concept of a conserved quantity, averaged over a given volume, which is updated using the notion of the flux of that quantity through the surface of the volume. The archetypal conservation law is

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}^i(\mathbf{u})}{\partial x^i} = \mathbf{0}, \quad (4.2)$$

which we can rewrite in its integrated (weak) form

$$\int \mathbf{u} dx^i + \int \mathbf{F}^i(\mathbf{u}) dt = \mathbf{0}, \quad (4.3)$$

or, in a form more suited to relativity:

$$\int_V \nabla_\mu \mathbf{F}^\mu dV = \int_{\partial V} \mathbf{F}^\mu dS_\mu = \mathbf{0}, \quad (4.4)$$

where $\mathbf{F}^\mu = (\mathbf{u}, \mathbf{F}^i)$, and V is a space-time volume. Equation (4.4) best exemplifies the conservation law, but the form (4.3) is more usually seen in CFD literature.

In the case of a fluid, where mass and momentum are conserved, this is easily imagined in terms of the mass-flux through the surface of a region; the only way for mass to pass from inside the region to outside is to go through the boundary. This is more evident in the four-dimensional conservation law (4.4). With the integrated form (4.3) there is no difficulty if a variable is discontinuous somewhere in that volume. The form (4.2) is seen more often, but is of course undefined for discontinuous fluid variables.

Stable finite-volume methods tend to be somewhat harder to design and implement for high orders of accuracy than finite-difference methods. In this work, therefore, we restrict ourselves to second-order methods, and seek to improve the overall accuracy of a scheme within that constraint without using to a higher-order approach. We note that, even allowing for this approach, we can still achieve levels of accuracy similar to those of higher-order schemes such as those used by Del Zanna and Bucciantini [117].

In many systems of equations, the zero right-hand side in (4.2) is replaced by so-called source-terms. These can be seen both in the Bona-Massó equations (3.39)-(3.44) and the GRHD equations (3.5). In both these cases, the sources depend only on the solution variables themselves. However, in general, the sources may depend on derivatives of the variables as well, and this is the case for the BSSN formulation of Einstein's equations [18]. The way in which the evolution of the source terms is calculated and coupled with the flux terms is discussed in §4.4.

4.1.2 Grid Structure

In order to evolve a space-time that includes a black hole, one approach is to excise the singularity so as to avoid having to evolve the space-time or fluid in regions of large, tending to infinite, curvature, which could cause numerical overflows or slow evolution due to the maximum stable time-step being reduced dramatically by small grid cells. Many current codes, such as those described in §2.3, are restricted to Cartesian grids, and therefore need to use Lego[®] excision as demonstrated in Figure 4.1. However, this leads to a jagged excision edge which could cause problems in implementing stable boundary conditions. Note that if we wanted to use a cubic excision region, its size would be severely constrained as noted in the Introduction and Appendix of [26], to the extent that it is not possible to use such an excision for high-spin Kerr black holes. Further discussion on the location of the excision boundary can be found in the work of Szilágyi *et al.* [103].

The excision will be done inside the apparent horizon of any black hole present so that, physically speaking, no information, including any about instabilities, should be capable of leaving the black hole's event horizon. However, numerical instabilities need not be constrained by physical properties of the system, and so there is no guarantee that any instabilities would not affect the space-time outside the horizon.

In fact, as previously discussed in §2.2.1, the speed of propagation of the gauge is not constrained by physical considerations. Errors in computing the gauge as a result of excision could therefore propagate outside the event horizon. Particular care therefore has to be taken when evolving the metric to ensure that the excision boundary conditions do not affect the solution.

There is also the issue of whether, having excised a region inside the apparent horizon, the problem remains well-posed in the discrete sense, since we have introduced a non-physical boundary, on which

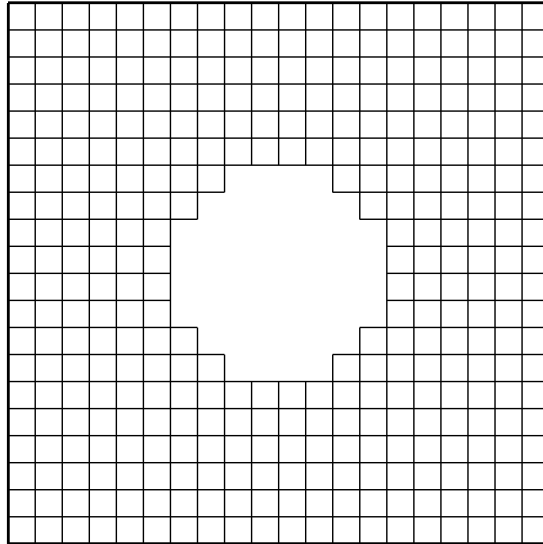


Figure 4.1: *Excising a circle in a Cartesian domain by means of Lego[®] excision.*

boundary conditions must be specified. Work on this topic has been performed by Kreiss and collaborators ([65] and [66]). However, since in this thesis we only use excision when evolving fluid and not the metric, these issues will not concern us further.

It would therefore seem prudent to use a spherical grid for the evolution of a fluid in a black hole space-time. This improves on Cartesian grids in two ways. Firstly, it allows for a smooth excision boundary at some radius inside the apparent horizon of the black hole, which is far easier to deal with from a numerical point of view, and also from an implementation point of view as a Lego[®] excision boundary requires some extra logic to alter the numerical method used on the excision boundary. Secondly, a spherical coordinate system only requires the number of grid cells to be increased by a factor of two for a doubling in the outer radius of the domain.

Spherical coordinates do have drawbacks, however. Near to the poles, the grid cells become very small in their longitudinal dimension, and this leads to requiring very small time-steps in order to keep the simulation stable, so that a large amount of computational time is required. Further, there is also the problem of the coordinate singularity along the axis. This requires a substantial amount of analytical work to overcome, in order to regularise Einstein's equations for solution on a spherical coordinate system. The work of Rinne [89] contains details of such a regularisation.

The issue of prohibitively small grid cells can be avoided by using two or more grids to cover different segments of the domain, so that no one grid contains a singularity. The issue of the coordinate system singularity can also be overcome by using different coordinate systems on the different patches, as demonstrated by Thornburg [106].

However, this requires us to interpolate the evolving solution between the grids. There are two possible approaches to performing this interpolation. The first is to allow the various grids merely to touch. This, however, is not suitable for non-smooth solutions, such as we expect to find in fluids. Having grids that touch means that, at the grid boundaries, only information from one side, i.e. inside the grid, is available. When we need to evolve a solution here, we cannot expect that a one-sided numerical scheme as might be used in the finite-difference approach would be suitable since, if a discontinuity approached a grid boundary, a one-sided reconstruction would not allow the grid onto which the discontinuity was passing to know about the incoming discontinuity. Some way is therefore needed to have information from both sides of a boundary available to the numerical scheme at once, so that a suitable reconstruction can be made, even for non-smooth solutions.

We therefore use the second approach, which is to allow the grids to overlap, and then grid cells on the boundary of one grid can interpolate from cells inside the evolved domain of the other grid. We would, however, need to ensure that the interpolation be robust and not lead to numerical oscillations or unphysical instabilities.

We note that making this generalisation will allow for the use of completely general overlapping sets of grids. An example would be the use of a global spherical set of coordinates (constructed out of two or more grids to avoid singularities), and another local spherical set of grids, so as to excise the singularities of two black holes (see §8.6). Other approaches to the same problem require either Lego[®] excision or carefully constructed domains.

We believe that the advantages of using overlapping grids are greater than the problems caused by having to implement methods capable of evolving on the general grids. Several of these problems are obviated by using a system such as Overture that implements all the interpolation between different grids, for a completely general grid structure. Overture is also capable of generating grids for AMR (see §4.9), which is an extra advantage.

4.1.3 Mesh Adaptation

Second-order finite-volume schemes are so-called because they are second-order on smooth solutions. However, near to discontinuities, they reduce to first-order convergence in order not to introduce numerical instabilities nearby, in accordance with Godunov's theorem (see §4.2.1). It is therefore the case that finite-volume schemes are at their least accurate near discontinuities, the very features we wish to capture. Ideally, we should like to be able to run the simulation at a sufficiently high resolution that the discontinuities would be captured reasonably well. However, our computational resources are finite, so it would seem to be worthwhile to seek to increase resolution only at shocks, so that smooth parts of the solution are evolved at a low resolution, and shocks are evolved at a higher resolution. We would

therefore require that the regions of high resolution follow the shocks dynamically. Such an approach was invented by Berger and Olinger [19], and is known as Adaptive Mesh Refinement, or AMR. We have included this in our algorithm. More details are given in §4.9.

4.2 Finite Volume Schemes

Finite Volume (FV) schemes are applied to PDEs written in the form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x^{(1)}} \mathbf{f}_1(\mathbf{u}) + \cdots + \frac{\partial}{\partial x^{(n)}} \mathbf{f}_n(\mathbf{u}) = \mathbf{S}(\mathbf{u}), \quad (4.5)$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is a set of variables to be evolved, the fluxes \mathbf{f}_i are functions of \mathbf{u} only, with the same dimension, and $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$ forms the initial conditions. The source terms \mathbf{S} usually arise from either chemical reactions or geometrical effects not contained within the physical grid structure, and do not depend on derivatives of \mathbf{u} .

Clearly, (4.5) is undefined when the solution has discontinuities, and therefore we use the integral form of the equation when expressed over the region $[x - \frac{1}{2}\Delta x, x + \frac{1}{2}\Delta x] \times [t, t + \Delta t]$ for some Δx and Δt . We only outline the FV approach for one space dimension; the extension to more dimensions is straightforward. From now on in this section, subscripts i refer to discretisation in space, and superscripts n refer to discretisation in time. We obtain

$$\Delta x (\bar{\mathbf{u}}_i^{n+1} - \bar{\mathbf{u}}_i^n) + \Delta t (\hat{\mathbf{f}}_{i+\frac{1}{2}}^n - \hat{\mathbf{f}}_{i-\frac{1}{2}}^n) = \mathbf{0}, \quad (4.6)$$

where

$$\bar{\mathbf{u}}_i^n = \frac{1}{\Delta x} \int_{x_i - \frac{1}{2}\Delta x}^{x_i + \frac{1}{2}\Delta x} \mathbf{u}(x', t^n) dx' \quad \text{and} \quad \hat{\mathbf{f}}_i^n = \frac{1}{\Delta t} \int_t^{t+\Delta t} \mathbf{f}(x_i, t') dt' \quad (4.7)$$

are averages over the cell-volume and period covered by a time-step, respectively. The name “finite-volume” arises from the fact that we take integrals over the cells and hence that we evolve the value averaged over each cell volume.

We obtain a stable time-step Δt from the Courant-Friedrich-Lewy (CFL) condition

$$\Delta t \leq c_{\text{CFL}} \frac{\Delta x}{S_{\text{max}}}, \quad (4.8)$$

where S_{max} is the fastest wave speed over the whole domain and c_{CFL} is the CFL or Courant number. Note that it is not necessarily true that $S_{\text{max}} \leq 1$, as S_{max} includes the effects of information about the gauge, which may travel faster than the speed of light. The CFL number is usually required to satisfy $0 \leq c_{\text{CFL}} \leq 1$ for stability, although the upper bound may vary, depending on the precise scheme used. Having an upper bound of one is the most intuitive, however, as it corresponds to the situation where information may propagate by precisely one cell in one time-step, and this satisfies the notion that the

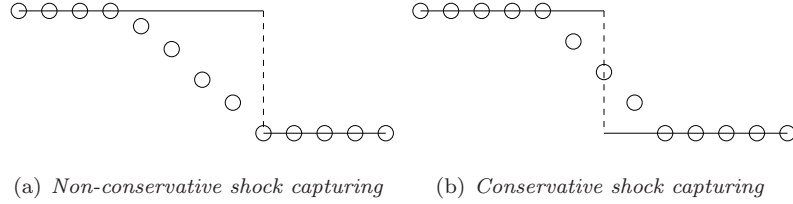


Figure 4.2: *Illustrations of computed solutions to shocks. The lines represent the exact solution, and the circles represent the computed solution.*

physical domain of dependence must lie within the numerical domain of dependence if we are to obtain a consistent evolution. As information usually only travels by one cell per time step, very few stable schemes have an upper bound greater than one, but many have an upper bound somewhat below one.

All FV schemes follow the same basic evolution step, derived from (4.6):

$$\bar{\mathbf{u}}_i^{n+1} = \bar{\mathbf{u}}_i^n + \frac{\Delta t}{\Delta x} \left(\hat{\mathbf{f}}_{i-\frac{1}{2}}^n - \hat{\mathbf{f}}_{i+\frac{1}{2}}^n \right). \quad (4.9)$$

This states that the change in the amount of \mathbf{u} contained in cell i increases by an amount equal to the time-step Δt multiplied by the rate at which \mathbf{u} enters through cell-face $i - 1/2$ minus the rate at which it leaves through cell-face $i + 1/2$.

The difference between various FV schemes comes when we consider how to approximate the fluxes $\hat{\mathbf{f}}_{i+\frac{1}{2}}^n$ between the cells.

The evolution equation (4.9) makes one of the important properties of FV schemes immediately clear; regardless of the approximation used to obtain $\hat{\mathbf{f}}_{i+\frac{1}{2}}^n$, it is always the case that the quantity $\sum_i \bar{\mathbf{u}}$ is conserved if we ignore the effect of boundary conditions. In the case of a fluid, where it is vital that mass, momentum, and energy are conserved, this is an important property, and is one of the main reasons for FV schemes being widely used in computational fluid dynamics.

A corollary of the fact that FV schemes conserve the integral of quantities is that they capture shock-speeds accurately. This is not the case, in general, for other types of numerical scheme. Figure 4.2 shows two possible ways of capturing a shock wave. The right-hand solution, although it smears the shock to the same extent as the left-hand, is considered to be better since the smear is centred on the exact solution. It is clear that the integral of the numerical solution on the left-hand side is not the same as that of the exact solution, and so this cannot be the result of a conservative scheme. A finite-volume scheme, on the other hand, being conservative, will always capture a shock in the manner of the right-hand figure.

If insufficient care is taken in calculating the flux approximations, then it is possible to generate unphysical oscillations in the solution, often in the vicinity of discontinuities. These tend to increase in size exponentially in time, and come to dominate the solution as high frequency modes increase in amplitude.

Some care is taken, therefore, to seek numerical methods that are Total Variation Diminishing (TVD). Total Variation (TV) is calculated for a discrete solution as

$$\text{TV}(u) = \sum_i |u_i - u_{i+1}|. \quad (4.10)$$

This calculates the sum of the height differences between adjacent peaks and troughs in the solution. If we have a scheme which obeys

$$\text{TV}(u^n) \geq \text{TV}(u^{n+1}) \quad \forall n, \quad (4.11)$$

then it is said to be TVD, and clearly no extra oscillations can be generated. It can be shown that the exact solutions to convex *scalar* conservation laws obey this restriction, so that TVD schemes are seen to be relevant to the problem in question. This is not necessarily the case for systems of equations, however, but experience has shown that schemes that are TVD for scalar equations tend to perform well, and not usually effect oscillations, for systems of equations.

The usual approach to considering how best to approximate the flux across a cell boundary is to consider the Riemann problem (see (3.50)) formed by the reconstructed solution on both sides of the boundary. This reconstruction can be to any order, although higher order schemes are more computationally intensive.

From now on, we shall omit the line and hat indicating the cell and time averages, and simply refer to \mathbf{u}_i^n and $\mathbf{f}_{i+\frac{1}{2}}^n$.

A subset of finite volume schemes is the set of High-Resolution Shock-Capturing (HRSC) schemes. These are capable of capturing shocks with very little smearing, even at low resolution, and are of at least second-order accuracy.

4.2.1 FORCE and SLIC schemes

One of the most basic FV schemes is FORCE (First ORder CEntred). It can be applied to any set of PDEs of the form (4.5) (omitting source terms) with little effort. The flux approximation is calculated as follows:

$$\mathbf{f}_{i+\frac{1}{2}}^{\text{FORCE}}(\mathbf{u}_i^n, \mathbf{u}_{i+1}^n) = \frac{1}{2}\mathbf{f}_{i+\frac{1}{2}}^{\text{LW}} + \frac{1}{4}[\mathbf{f}_i^n + \mathbf{f}_{i+1}^n] + \frac{1}{4}\frac{\Delta x}{\Delta t}(\mathbf{u}_i^n - \mathbf{u}_{i+1}^n), \quad (4.12)$$

where $\mathbf{f}_i^n = \mathbf{f}(\mathbf{u}_i^n)$, $\mathbf{f}_{i+\frac{1}{2}}^{\text{LW}} = \mathbf{f}\left(\mathbf{u}_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right)$, and $\mathbf{u}_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}(\mathbf{u}_i^n + \mathbf{u}_{i+1}^n) + \frac{1}{2}\frac{\Delta t}{\Delta x}(\mathbf{f}_i^n - \mathbf{f}_{i+1}^n)$.

This is first-order accurate, and is a centred scheme, meaning that it does not have a (solution dependent) bias to information from either the left or the right. As will become relevant later, FORCE is the simple average of two other numerical schemes: the Lax-Wendroff scheme, being the term $\mathbf{f}_{i+\frac{1}{2}}^{\text{LW}}$, and the Lax-Friedrichs scheme, being the remaining terms up to a factor of $\frac{1}{2}$. The Lax-Wendroff scheme

is second-order but, in accordance with Godunov's theorem¹, causes oscillations near shocks, and the Lax-Friedrichs scheme is first-order, but very diffusive. It turns out that the FORCE scheme is the least diffusive non-oscillatory first-order scheme that is not dependent on local solution information.

The FORCE scheme is based on the piecewise constant reconstruction of the solution, i.e. the left and right states of the Riemann problem are taken directly from the solution vectors \mathbf{u}_i^n and \mathbf{u}_{i+1}^n . One improvement to this, known as the SLIC scheme (Slope-Limited Centred), is to use cells $i - 1$ and $i + 2$ to make a piecewise linear reconstruction of the solution at time $n + \frac{1}{2}$ to provide initial states for the Riemann problem. However, we must take care that, in reconstructing the solution, we do not introduce any new extrema into the solution. We first reconstruct the slope of the solution, using data from either side of the current cell:

$$\Delta_i^- = \mathbf{u}_i^n - \mathbf{u}_{i-1}^n, \quad \Delta_i^+ = \mathbf{u}_{i+1}^n - \mathbf{u}_i^n, \quad \Delta_i = \frac{1}{2}(\Delta_i^- + \Delta_i^+), \quad (4.13)$$

and then change the slope Δ_i by multiplying each component by a factor dependent on the local solution behaviour:

$$\Delta_i^* = \Delta_i \cdot \xi(\Delta_i^-, \Delta_i^+), \quad (4.14)$$

where

$$\xi(\Delta_i^-, \Delta_i^+) \equiv \xi\left(\frac{\Delta_i^-}{\Delta_i^+}\right) \equiv \xi(r). \quad (4.15)$$

The preceding two equations are applied to each individual component of Δ_i , Δ_i^+ , and Δ_i^- to form a limited slope Δ_i^* .

The SLIC flux is then calculated as

$$\mathbf{u}_{i,L}^n = \mathbf{u}_i^n - \frac{1}{2}\Delta_i^*, \quad \mathbf{u}_{i,R}^n = \mathbf{u}_i^n + \frac{1}{2}\Delta_i^*, \quad (4.16)$$

$$\mathbf{u}_{i,L}^{n+\frac{1}{2}} = \mathbf{u}_{i,L}^n + \frac{\Delta t}{2\Delta x} (\mathbf{f}(\mathbf{u}_{i,L}^n) - \mathbf{f}(\mathbf{u}_{i,R}^n)), \quad (4.17)$$

$$\mathbf{u}_{i,R}^{n+\frac{1}{2}} = \mathbf{u}_{i,R}^n + \frac{\Delta t}{2\Delta x} (\mathbf{f}(\mathbf{u}_{i,L}^n) - \mathbf{f}(\mathbf{u}_{i,R}^n)), \quad (4.18)$$

$$\mathbf{f}_{i+\frac{1}{2}} = \mathbf{f}_{i+\frac{1}{2}}^{\text{FORCE}} \left(\mathbf{u}_{i,R}^{n+\frac{1}{2}}, \mathbf{u}_{i+1,L}^{n+\frac{1}{2}} \right). \quad (4.19)$$

where we use $\mathbf{u}_{i,R}^{n+\frac{1}{2}}$ and $\mathbf{u}_{i+1,L}^{n+\frac{1}{2}}$ as the left and right states for the FORCE flux.

There are many options for the slope-limiter ξ , even allowing for the constraint that the overall numerical scheme must be TVD. The constraints on $\xi(r)$ can be shown to be ([109] p505, setting $\omega = 0$):

$$\xi(r) = 0 \text{ for } r \leq 0, \text{ and } 0 \leq \xi(r) \leq \min\{\xi_L(r), \xi_R(r)\} \text{ for } r > 0, \quad (4.20)$$

¹Godunov's theorem states that "linear monotonicity preserving methods are first-order accurate, at best", where the TVD property implies monotonicity preservation. This statement of the theorem comes from Laney [67] p277, as the original theorem is in Russian [50].

where

$$\xi_L(r) = \frac{4r}{(1 + c_{\text{CFL}})(1 + r)}, \text{ and } \xi_R(r) = \frac{4r}{(1 - c_{\text{CFL}})(1 + r)}, \quad (4.21)$$

with c_{CFL} being the Courant number.

We see that r in (4.15) is a measure of the change in slope in the solution around cell i . It is negative if i lies at an extremum of the solution, as one of Δ_i^+ and Δ_i^- will be positive and the other negative, and this is the reason that we require $\xi(r) = 0$ for $r < 0$ as any non-constant reconstruction would result in an overshoot or undershoot of the extremum.

The slope-limiters that we will use in this thesis are:

$$\begin{aligned} \xi_{\text{super-bee}}(r) &= \begin{cases} 0 & \text{if } r \leq 0, \\ 2r & \text{if } 0 < r \leq \frac{1}{2}, \\ 1 & \text{if } \frac{1}{2} < r \leq 1, \\ \min\left(r, 2, \frac{2}{1+r}\right) & \text{if } r > 1, \end{cases} \\ \xi_{\text{van Leer}}(r) &= \begin{cases} 0 & \text{if } r \leq 0, \\ \min\left(\frac{2r}{1+r}, \frac{2}{1+r}\right) & \text{if } r > 0, \end{cases} \\ \xi_{\text{min-bee}}(r) &= \begin{cases} 0 & \text{if } r \leq 0, \\ r & \text{if } 0 < r \leq 1, \\ \min\left(1, \frac{2}{1+r}\right) & \text{if } r > 1. \end{cases} \end{aligned} \quad (4.22)$$

These have been arranged in order from least diffusive to most diffusive. For a scalar conservation law such as linear advection, all of these are guaranteed to be TVD. However, there is no such guarantee for other systems of equations, and in general we use the van Leer limiter as the super-bee limiter can in extreme circumstances cause overshoots.

4.2.2 Multi-Staging approach (MUSTA)

When considering a numerical scheme, there are two types of accuracy that need to be considered. Firstly, the order of accuracy, which determines the rate at which the numerical solution tends to the exact solution with increasing resolution. Secondly, the overall accuracy, which is the actual difference between the exact and computed solution. The point to note is that it is possible to increase the overall accuracy of a method without increasing its *order* of accuracy.

In order to improve on the accuracy of SLIC, we would usually consider Riemann solvers, which use information about the wave types making up the solution to calculate the flux. These tend to be computationally intensive, often requiring iterative or matrix methods to determine the state at the cell

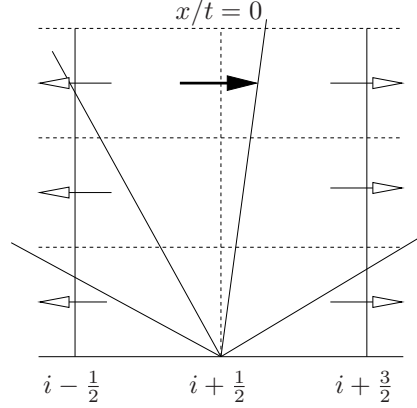


Figure 4.3: How to multi-stage a simple numerical scheme (e.g. FORCE). The Riemann problem is solved for several time steps using the simple scheme on the two-cell domain, with transmissive boundary conditions as shown by the empty arrows. The solid lines show the wave structure of the solution. As the number of stages increases, the waves move away from the centre, so that the flux at the centre tends to the correct limit, which is represented by the filled arrow at the top. This figure corresponds to calculating the FORCE³ scheme, if FORCE is used as the underlying simple scheme.

face. In order to increase the order of accuracy, we could use a higher-order reconstruction than the linear one that SLIC uses. Again, this would require more computational expense.

However, Toro [110] has suggested a simple way of improving the accuracy of any numerical scheme, which can produce improved accuracy for less computational effort than would be required if the accuracy were to be attained by increasing resolution. This is called MUSTA, which stands for Multi-STaging and can be understood as follows:

The FORCE flux provides a simple way of estimating the flux between two cells, given a Riemann problem. However, if we were to evolve a Riemann problem for a few time steps, the exact flux between the two cells would remain constant due to the similarity of the solution in x/t , and the values of the solution immediately to either side of the initial discontinuity would become closer. The FORCE flux can provide a better estimate of the correct flux given two close states than two separated states, and so its approximation of the central flux would be improved. Therefore, the MUSTA approach is to evolve the Riemann problem on a two cell grid, with transmissive boundary conditions, for a few time steps, or stages, and to take the central flux at the last stage as the estimate for the final flux. This can be represented as in Figure 4.3. In terms of an evolution algorithm, multi-staged FORCE can be written as the iterative scheme

$$\mathbf{f}_{i+\frac{1}{2}}^{\text{FORCE}^k} = \mathbf{f}_{i+\frac{1}{2}}^{\text{FORCE}}(\mathbf{u}_L^{k-1}, \mathbf{u}_R^{k-1}), \quad (4.23)$$

$$\mathbf{u}_L^k = \mathbf{u}_L^{k-1} - \frac{\Delta t}{\Delta x} \left(\mathbf{f}_{i+\frac{1}{2}}^{\text{FORCE}^k} - \mathbf{f}(\mathbf{u}_L^{k-1}) \right), \quad (4.24)$$

$$\mathbf{u}_R^k = \mathbf{u}_R^{k-1} - \frac{\Delta t}{\Delta x} \left(\mathbf{f}(\mathbf{u}_R^{k-1}) - \mathbf{f}_{i+\frac{1}{2}}^{\text{FORCE}^k} \right), \quad (4.25)$$

where $u_L^0 = u_L$ and $u_R^0 = u_R$ give the initial conditions.

The best improvements in accuracy for least computational expense are likely to be found when multi-staging is applied to as simple a scheme as possible. Therefore, in order to improve the accuracy of SLIC, we do not multi-stage SLIC itself, but rather multi-stage the FORCE scheme on which SLIC is based. We write the multi-staged FORCE scheme as FORCE^k , where k is the number of stages, so that $\text{FORCE}^1 \equiv \text{FORCE}$. This notation is as in [110] but differs from that in [112].

Although the preceding explanation seems plausible enough, we can examine the theoretical effect of MUSTA for the linear advection equation. As mentioned in §4.2.1, the FORCE scheme can be expressed as

$$\mathbf{f}_{i+\frac{1}{2}}^{\text{FORCE}} = \omega \mathbf{f}_{i+\frac{1}{2}}^{\text{LW}} + (1 - \omega) \mathbf{f}_{i+\frac{1}{2}}^{\text{LF}} \quad \text{with } \omega = \frac{1}{2}. \quad (4.26)$$

The local CFL number c_L is derived from the *local* wave speed $a_{i+\frac{1}{2}}$ and *global* time step Δt as

$$c_L = \frac{\Delta t}{\Delta x} a_{i+\frac{1}{2}}. \quad (4.27)$$

For linear advection, all numerical schemes with a centred stencil of width at most three can be written in the form (4.26), and so we can plot the dependence of ω on c_L for various schemes, as in Figure 4.4. For the linear advection problem, other schemes can be plotted on the same diagram. The exact Riemann solver (Godunov scheme) falls on the curved line shown. This splits the diagram into two parts. Above the curve lie schemes that are oscillatory, and below it lie schemes that are non-oscillatory, but are more diffusive than the ideal. The exact Riemann solver is the least diffusive non-oscillatory first-order method for the linear advection equation. The FORCE scheme can therefore be seen to be the least diffusive non-oscillatory first-order method not to depend on the local wave speed information given by c_L .

Although the MUSTA approach does not depend explicitly on c_L , it does depend on it implicitly, through its dependence on the local data for the predictor step (4.23). This dependence can be calculated in a straight-forward, but algebraically intensive manner, which we have done using MapleTM, and we plot the results for FORCE^n , $n = 1, \dots, 5$ in Figure 4.5. It is immediately clear that MUSTA may not be a suitable evolution scheme at high CFL numbers, even for such a simple equation as linear advection, as all of the schemes now protrude into the oscillatory region. Using very high numbers of stages shows that FORCE^k tends to the Lax-Wendroff scheme as $k \rightarrow \infty$ (except for $c_{\text{CFL}} = 1$ where $\omega = \frac{1}{2}$ for all k , so that the convergence is not uniform).

However, some improvement can be made to this. If we replace Δt by $\alpha \Delta t$ all the way through, i.e. in equations (4.24),(4.25) and (4.12), keeping α the same throughout a multi-staging calculation, then using the values of α in Table 4.1, we find the situation in Figure 4.6. Here we see that there is no protrusion into the oscillatory region, and that we have reduced the diffusivity of the FORCE scheme.

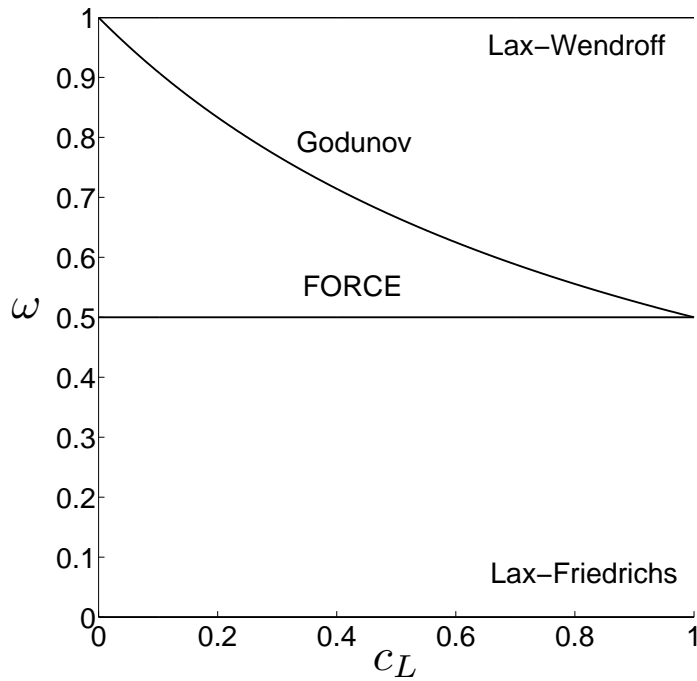


Figure 4.4: *Four basic schemes for evolving the linear advection equation, displayed in terms of the local CFL number c_L , which affects the weighting used to construct the method from the Lax-Friedrichs and Lax-Wendroff schemes.*

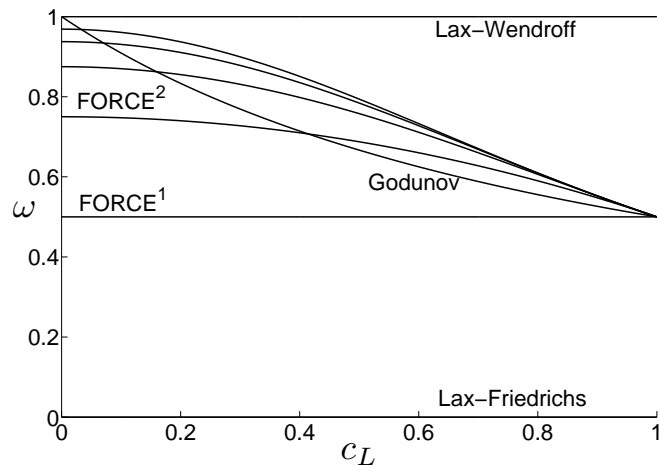


Figure 4.5: *Same as Figure 4.4, with additional multi-staged versions of $FORCE^k$ where $k = 1, \dots, 5$ and k increases up the page (labels for $k = 3, 4, 5$ are omitted for reasons of clarity).*

However, increasing the number of stages beyond two is of little use. The values for α were calculated using MapleTM, the condition being that the curves should lie below the Godunov curve for all values of

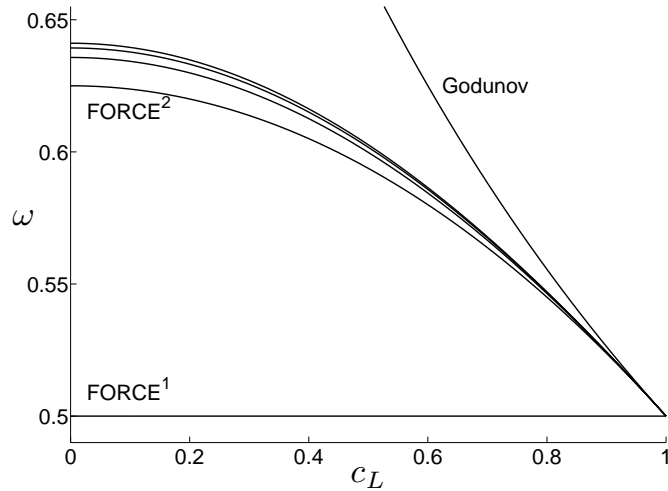


Figure 4.6: Same as Figure 4.5, except that Δt in (4.24) and (4.25) has been scaled by a factor α to maintain the monotonicity of the scheme overall. Note the expanded ω scale as compared to the other two figures.

stages	α
1	not applicable
2	$\frac{1}{2}$
3	$1 - \frac{1}{\sqrt{2}}$
4	$1 - \frac{1}{\sqrt[3]{2}}$
5	$1 - \frac{1}{\sqrt[4]{2}}$

Table 4.1: Values of α that make a multi-staged FORCE scheme stable for the linear-advection problem when replacing Δt by $\alpha\Delta t$.

c_{CFL} . We have not been able to prove the apparent relation that $\alpha = 1 - 2^{1-k}$ in general.

4.2.3 GFORCE flux

The impetus for the development of MUSTA was the fact that even for the simplest of all conservation laws, the linear advection equation, the FORCE method was not equivalent to the exact Godunov Riemann solver, and the approach was to improve the accuracy of a scheme, without requiring local wave speed information. However, we have seen that the MUSTA approach is not ideal as it leads to oscillatory methods, even for linear advection.

Toro and Titarev presented a solution to this in the same paper as their MUSTA approach [112]. In this, we calculate the generalised FORCE flux (GFORCE) as a weighted average of the Lax-Friedrichs

and Lax-Wendroff fluxes, now dependent on local wave speed information:

$$\mathbf{f}_{i+\frac{1}{2}}^{\text{GFORCE}} = \frac{1}{1 + c_L} \left(\mathbf{f}_{i+\frac{1}{2}}^{\text{LW}} + c_L \mathbf{f}_{i+\frac{1}{2}}^{\text{LF}} \right) \quad (4.28)$$

where c_L is the *local* CFL number, derived from the *local* wave speed $a_{i+\frac{1}{2}}$ and *global* time step Δt as

$$c_L = \frac{\Delta t}{\Delta x} a_{i+\frac{1}{2}}. \quad (4.29)$$

With these definitions, GFORCE replicates the Godunov upwind scheme for the linear-advection case. This suggests that the GFORCE flux could well provide improved accuracy, beyond that of MUSTA, for more complex systems of equations. The wave speed calculation is carried out in the local computational grid coordinates so that the wave speeds are aligned with the grid, and the wave-speeds we use for GRHD are those given in equation (3.8).

As the GFORCE scheme already coincides with the Godunov scheme for the linear advection equation, applying the MUSTA approach makes no difference in this case, simply maintaining the Godunov scheme. However, this is not the case for more complicated systems of equations, and we shall see that multi-staging GFORCE provides some increase in accuracy.

4.2.4 HLL flux

The GFORCE flux, being equivalent to the Godunov exact Riemann solver in this case, is the most accurate non-oscillatory first-order scheme for the linear advection equation. However, more complex systems of equations have more complex solutions to their Riemann problems. It is possible, in many cases, to derive an exact Riemann solver for a system of equations. However, this will usually involve either calculating a matrix inverse or using an iterative method to find the flux. This tends to be computationally expensive, and is therefore often avoided unless the pay-off of computational time versus accuracy is worth it.

However, an approximate Riemann solver, applicable to any system of equations, was suggested by Harten, Lax, and van Leer [56], known as the HLL flux. The rationale is that if all the waves in the solution are moving in one direction only, then the flux is taken wholly from information on one side or the other, and an appropriately weighted average is taken otherwise:

$$\mathbf{f}_{i+\frac{1}{2}}^{\text{HLL}} = \begin{cases} \mathbf{f}(\mathbf{u}_i) & \text{if } 0 < a_L, \\ \frac{a_R \mathbf{f}(\mathbf{u}_i) - a_L \mathbf{f}(\mathbf{u}_{i+1}) + a_L a_R (\mathbf{u}_{i+1} - \mathbf{u}_i)}{a_R - a_L} & \text{if } a_L < 0 < a_R, \\ \mathbf{f}(\mathbf{u}_{i+1}) & \text{if } a_R < 0. \end{cases} \quad (4.30)$$

This is only dependent on the local wave speeds a_L and a_R . These can be calculated from the fluid variables as suggested by Schneider *et al.* [96]

$$a_L = \min \left(0, \frac{\bar{v} - \bar{c}_s}{1 - \bar{v} \bar{c}_s} \right), \quad a_R = \max \left(\frac{\bar{v} + \bar{c}_s}{1 + \bar{v} \bar{c}_s}, 0 \right), \quad (4.31)$$

where \bar{v} and \bar{c}_s are the arithmetic means of the initial velocities and sound speeds in cells i and $i + 1$.

The wave speeds in (4.31) are the special relativistic simplification of those given in (3.8). For the general relativistic case, we use the speeds from (3.8) based on averaging the metric components as well as the fluid variables. Indeed, the HLL flux for metric variables and fluid variables can be calculated separately as the fluid wave speed is always less than the metric wave speed, and we use this property in order to find the appropriate fluxes for both the metric and the fluid, even when evolving both at the same time. In the case of a curvi-linear grid, we note that we have to transform the variables into the computational grid's local coordinate system in order to find the wave speed parallel to the grid coordinate direction. We elaborate on this in §4.6.

4.2.5 Recommendations for the MUSTA approach

In summary, therefore, the multi-staging approach at first sight appears to produce a straightforward and cheap way of improving the accuracy of a simple flux approximation scheme. However, some investigation shows that applying it to the FORCE scheme alone is not a viable approach as this results in oscillatory schemes even for linear advection. An improvement on this appears to be to apply it to the GFORCE scheme, where no oscillatory schemes emerge for linear advection. The same applies to the HLL scheme.

However, we have not been able to extend our analysis to more complex systems of equations, and therefore our recommendations are as follows: use of FORCE^k may be advantageous, but any improvements are likely to be at part due to a failure of the TVD property of the scheme. The use of GFORCE^k or HLL^k does not, as yet, have any evidence of causing the loss of the TVD property, although, given the analysis for FORCE, we suggest caution as to whether this result can be applied to more complex systems of equations.

In any event, we may tend to be saved by the inherent diffusivity of higher-order schemes when applied to complex systems of equations, so that an increase in multi-staging may in fact do what it was designed to do without causing unstable schemes. We attach a *caveat emptor* here, therefore, to any use of multi-staging, but suggest that it may be useful if used with care, and with only a few extra stages, perhaps around three or so. We present some numerical results and comparisons in Chapter 5.

4.3 Finite Difference Schemes

The majority of NR codes use finite differencing methods to evolve their space-time variables. As we intend to make comparisons between our finite-volume methods and other current NR codes, we present here the finite difference scheme we will use.

We use the Iterated Crank-Nicolson method, an iterative, explicit version of the implicit Crank-Nicolson method. Given a numerical solution \mathbf{u}_i , we calculate the l th iteration as follows:

$${}^{(l-1)}\mathbf{u}_i^{n+\frac{1}{2}} = \frac{1}{2} \left({}^{(l-1)}\mathbf{u}_i^{n+1} + \mathbf{u}_i^n \right), \quad (4.32)$$

$$\frac{{}^{(l)}\mathbf{u}_i^{n+1} - \mathbf{u}_i^n}{\Delta t} = - \left(\frac{\mathbf{f} \left({}^{(l-1)}\mathbf{u}_{i+1}^{n+\frac{1}{2}} \right) - \mathbf{f} \left({}^{(l-1)}\mathbf{u}_{i-1}^{n+\frac{1}{2}} \right)}{2\Delta x} \right), \quad (4.33)$$

using the starting condition ${}^{(0)}\mathbf{u}_i^n = \mathbf{u}_i^n$.

The solution for the next time-step is then given by $\mathbf{u}_i^{n+1} = {}^{(l)}\mathbf{u}_i^{n+1}$. In terms of the number of iterations that are necessary, it is possible to perform von Neumann stability analysis in the case of the linear advection equation (see Teukolsky [105]). The result of this is that we should use three iterations, corresponding to $l = 3$, and no more. We have a slight difference of notation here to make the form of (4.32)-(4.33) more compact, which accounts for the fact that [105] suggests two iterations, rather than the three we use here. Also, the extra minus sign on the right-hand side of (4.33) as compared to equation (11) in [105] comes from the fact that Teukolsky has the x -derivative on the right-hand side of the advection equation. The schemes are in fact equivalent.

4.4 Source Terms and Splitting

So far we have only considered the left-hand side of equation (4.5). The right-hand side of the equations are known as source terms, and often arise as the result of chemical reactions. In GRHD, they arise from geometry due to the fact that we are evolving on a non-flat three dimensional hypersurface embedded in a four-dimensional manifold.

We evolve the source terms using one of the following methods:

- Euler method

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \Delta t \cdot \mathbf{S}(\mathbf{u}_i^n) \quad (4.34)$$

- Runge-Kutta 2nd order (RK2)

$$\begin{aligned} \mathbf{u}_i^{n+\frac{1}{2}} &= \mathbf{u}_i^n + \frac{\Delta t}{2} \cdot \mathbf{S}(\mathbf{u}_i^n) \\ \mathbf{u}_i^{n+1} &= \mathbf{u}_i^n + \Delta t \cdot \mathbf{S} \left(\mathbf{u}_i^{n+\frac{1}{2}} \right) \end{aligned} \quad (4.35)$$

- Runge-Kutta 4th order (RK4)

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{S}(\mathbf{u}_n^i) \\
\mathbf{k}_2 &= \mathbf{S}\left(\mathbf{u}_n^i + \frac{\Delta t}{2} \mathbf{k}_1\right) \\
\mathbf{k}_3 &= \mathbf{S}\left(\mathbf{u}_n^i + \frac{\Delta t}{2} \mathbf{k}_2\right) \\
\mathbf{k}_4 &= \mathbf{S}\left(\mathbf{u}_n^i + \Delta t \mathbf{k}_3\right) \\
\mathbf{u}_i^{n+1} &= \mathbf{u}_i^n + \frac{\Delta t}{6} (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4).
\end{aligned} \tag{4.36}$$

The Euler method is first-order accurate, the RK2 scheme is second-order accurate, and the RK4 scheme is fourth-order accurate. However, in order to retain second-order accurate evolution overall, when combined with the evolution of the flux terms, we need to take care.

The need for this arises from the fact that if we approximate the evolution over one time-step as the evolution of two separate PDEs:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}^i}{\partial x^i} = \mathbf{0}, \text{ and } \frac{\partial \mathbf{u}}{\partial t} = \mathbf{S}(\mathbf{u}), \tag{4.37}$$

and combine them as

$$E(\Delta t) = S(\Delta t) T(\Delta t), \tag{4.38}$$

where S evolves the source terms and T evolves the flux terms, and they are applied individually, from right to left, then the full evolution operator E is then only first-order in time. However, if we use the Strang splitting

$$E(\Delta t) = S\left(\frac{1}{2}\Delta t\right) T(\Delta t) S\left(\frac{1}{2}\Delta t\right), \tag{4.39}$$

then E is second order in time if both T and S are. We could split the T operator instead of the S operator, but we generally need to evolve T by a full timestep to attain as much accuracy as possible and, in any case, T tends to be by far the more computationally expensive operator.

4.4.1 Combining the source evolution with the flux

A disadvantage of the above approach is that it may not be entirely consistent. The operators S and T have been constructed on the assumption that they will act on an exact solution of the system. If we consider the hypersurface of exact solutions of the system, we wish the numerical evolution to keep us on or close to this manifold, and we know that evolving the full system of equations exactly would do this. However, applying the operators S and T individually does not necessarily keep us on the manifold. In particular, applying a half-step of operator S and then a full step of operator T would not appear to leave us in a suitable position to apply operator S again. It would appear to be preferable to apply S to the result of a half-step of S and a half-step of T which, while it is not likely to leave us exactly on the

solution hypersurface, is at least similar to evolving the system for time $\Delta t/2$, for which we might expect to remain close to the hypersurface, and hence that applying S might be valid.

The approach that we have developed, therefore, is as follows, beginning with state \mathbf{u}^n :

1. Apply the source term operator using RK4 for time $\Delta t/2$ to obtain state $\mathbf{u}^{n+\frac{1}{2}S}$
2. Calculate the fluxes for time Δt for the state $\mathbf{u}^{n+\frac{1}{2}S}$, and add a half of them to that state to obtain state $\mathbf{u}^{n+\frac{1}{2}}$ (Note that this will not, in general, give the same fluxes as finding the fluxes for time $\Delta t/2$, but since the fluxes are the most computationally expensive part of the algorithm, we try not to have to calculate them any more than necessary.)
3. Calculate the sources for time $\Delta t/2$ for state $\mathbf{u}^{n+\frac{1}{2}}$ using RK4.
4. Add the full fluxes (i.e. for time Δt) from step (2) and the sources from step (3) to $\mathbf{u}^{n+\frac{1}{2}S}$ to obtain the final state \mathbf{u}^{n+1} .

The way in which this differs from the usual Strang splitting is the way in which only half the fluxes are added before calculating the second set of sources.

4.4.2 Simpson's Rule correction

Another improvement that can be made is to remember that, although we often calculate the source term as a point value, it is in fact an integral over the whole cell. We can therefore improve the accuracy of the source term by applying Simpson's Rule, which is that [88]

$$\int_{x-\frac{1}{2}\Delta x}^{x+\frac{1}{2}\Delta x} f(x) dx \approx \frac{1}{6} \Delta x (f(x - \frac{1}{2}\Delta x) + 4f(x) + f(x + \frac{1}{2}\Delta x)) + \frac{1}{2880} (\Delta x)^4 \langle f^{(4)} \rangle, \quad (4.40)$$

where $\langle f^{(4)} \rangle$ is an average of $f^{(4)}$ over $[x - \frac{1}{2}\Delta x, x + \frac{1}{2}\Delta x]$

for a smooth function f in one dimension, and is easily extended to multi-dimensions by multiple applications of the one-dimensional case.

We can include the effects of the Jacobian for non-uniform grids (see §4.6 for an explanation of terms used), multiplying each source term by the Jacobians averaged in the same way as the evolution variables, before dividing by the correct combination of Jacobians. In the one-dimensional case, for example, we take

$$\mathbf{S}_i = \frac{J_{i-\frac{1}{2}} \mathbf{S}(\mathbf{u}_{i-\frac{1}{2}}) + 4J_i \mathbf{S}(\mathbf{u}_i) + J_{i+\frac{1}{2}} \mathbf{S}(\mathbf{u}_{i+\frac{1}{2}})}{J_{i-\frac{1}{2}} + 4J_i + J_{i+\frac{1}{2}}}. \quad (4.41)$$

Usually, the values of $\mathbf{u}_{i-1/2}$ for the cell faces are approximated by taking the simple average $\frac{1}{2}(\mathbf{u}_{i-1} + \mathbf{u}_i)$.

4.4.3 Flux-Source balancing

Yet another improvement that can be made, at least in one dimension, is to attempt some form of flux-source balancing. A problem often arises when a solution close to a steady state is sought, but where the fluxes and sources are relatively large as compared to the perturbation from steady state. As the fluxes and sources are not calculated in the same manner as each other, the terms do not cancel exactly, and the loss of cancellation can swamp the effect sought. Much work has been done on certain sets of equations, particularly the shallow water equations [60], deriving numerical methods that allow the terms to cancel to round-off error.

For GRHD, such an analysis would take a large amount of work due to the highly non-linear nature of the problem. However, if we have a static solution $u(x)$ of the equation

$$\frac{\partial u(x)}{\partial t} + \frac{\partial f(u(x))}{\partial x} = s(u(x)), \quad (4.42)$$

then (writing $f(x)$ for $f(u(x))$ and $s(x)$ for $s(u(x))$ as a convenient shorthand):

$$\begin{aligned} & \frac{1}{\Delta x} (f(x + \frac{1}{2}\Delta x) - f(x - \frac{1}{2}\Delta x)) \\ &= \frac{1}{\Delta x} \left((f(x) + \frac{1}{2}\Delta x f'(x) + \frac{1}{8}\Delta x^2 f''(x) + \frac{1}{48}\Delta x^3 f'''(x)) \right. \\ & \quad \left. - (f(x) - \frac{1}{2}\Delta x f'(x) + \frac{1}{8}\Delta x^2 f''(x) - \frac{1}{48}\Delta x^3 f'''(x)) + \mathcal{O}(\Delta x^4) \right) \\ &= f'(x) + \frac{1}{24}\Delta x^2 f'''(x) + \mathcal{O}(\Delta x^3) \\ &= s(x) + \frac{1}{24}\Delta x^2 s''(x) + \mathcal{O}(\Delta x^3), \text{ assuming } u \text{ to be an exact solution} \\ &= s(x) + \frac{1}{6}(s(x + \frac{1}{2}\Delta x) + s(x - \frac{1}{2}\Delta x) - 2s(x)) + \mathcal{O}(\Delta x^3). \end{aligned} \quad (4.43)$$

Unfortunately, it is not possible to extend this approach to more than one dimension, as the resulting mixed derivatives of the fluxes cannot be simplified by assuming the solution to be exact.

4.5 Multi-dimensions

All the finite-volume schemes presented so far are for one dimension alone. There is a simple way to apply FV schemes to more than one dimension, namely, to apply the 1D operator to each dimension in turn:

$$T(\Delta t) = T_x(\Delta t) T_y(\Delta t) T_z(\Delta t), \quad (4.44)$$

and we permute the order of the operators with each timestep to avoid letting asymmetric effects creep in, although the symmetry in the axes is only maintained after two steps in the 2D case, and after three steps in the 3D case and then only if the time-step is fixed. However, this simple scheme results in a decrease in stability. In order for this to be stable, we have to take $c_{\text{CFL}} \leq \frac{1}{2}$ in the 2D case, and $c_{\text{CFL}} \leq \frac{1}{3}$ in the 3D case.

It is possible to avoid this by using a somewhat more complex approach to multi-dimensionalising the scheme, such as that developed by Toro and Hu [111]. In this scheme, the flux in the x direction can be found as

$$F_{x,\Delta t}^{\text{SLIC}} \left(F_{y,\Delta t/2}^{\text{FORCE}} \right) \quad (4.45)$$

in two dimensions and as

$$F_{x,\Delta t}^{\text{SLIC}} \left(F_{y,\Delta t/2}^{\text{FORCE}} \left(F_{z,\Delta t/2}^{\text{FORCE}} \right) \right) \quad (4.46)$$

in three dimensions.

In other words, we do not calculate the x flux directly from the data at the beginning of the time step, but from data that has been advanced by half a time-step in each of the other coordinate directions by a first-order scheme. The fluxes for the other coordinate directions are calculated in an equivalent way. In three dimensions, it is best to keep the cyclic order of the axes fixed for symmetry.

Although this requires more flux evaluations per time-step overall, the advantage is two-fold: firstly, it is completely symmetric in all the coordinate directions after each time-step, and secondly, there is an improved CFL stability condition.

With regard to the latter point, Toro and Hu claim that the scheme has a stability limit of $c_{\text{CFL}} \leq 1$ for two dimensions, and $c_{\text{CFL}} \leq \frac{1}{2}$ for three dimensions. However, this claim is based only on the linear-advection scheme, using von Neumann analysis. Simulations that we have done suggest that this result is valid even for more complex systems of equations. As a result, we can use $c_{\text{CFL}} = 0.95$ in 2D evolutions, which means that we introduce as little diffusion as possible.

However, Toro and Hu do not appear to have looked at the stability of the centred scheme in three dimensions, although they have used the obvious extension of their approach in constructing a multi-dimensional WAF scheme [20] and investigated the stability of that scheme.

We have examined a simplified version of equations (4.45) and (4.46) by replacing the SLIC scheme with FORCE or GFORCE. We have shown, using MapleTM to perform a numerical sampling of frequency space in a von Neumann approach, that for the linear advection equation in three dimensions, the scheme has stability region $\max(c_x, c_y, c_z) \leq 1$ as we might hope.

4.6 Curvi-linear grids

If we wish to evolve PDEs in coordinate systems other than Cartesian, we have two choices. We can either transform the PDEs themselves, i.e. transform (3.4)-(3.5) or (3.39)-(3.44) and the corresponding variables from Cartesian coordinates into general coordinates, or we can transform the Cartesian fluxes into the general coordinate system, while leaving the variables defined everywhere in Cartesian coordinates.

Since we need to interpolate between grids with different coordinate systems, and Overture performs the interpolation itself, without doing any coordinate transformations, it is easiest to have all the variables in Cartesian coordinates. We write Cartesian coordinates as x^i and the coordinate system of the grid as r^i . We assume that there exists a smooth mapping between these, and that the coordinate system of the grid is rectangular, which is the case in Overture. This means that we transform the flux equations as follows

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}^i(\mathbf{u})}{\partial x^i} = \mathbf{0} \mapsto \frac{\partial J\mathbf{u}}{\partial t} + \frac{\partial}{\partial r^i} \left(J \frac{\partial r^i}{\partial x^j} \mathbf{f}^j(\mathbf{u}) \right) = \mathbf{0}, \quad (4.47)$$

where J is the Jacobian of the transformation:

$$J = \left| \left[\frac{\partial x^i}{\partial r^j} \right] \right|, \quad (4.48)$$

although an extra term needs to be added when discretising this, as shown in §4.6.3.

The standard finite-volume methods can then be applied to the transformed equations. However, we have to look carefully at the transformed equations, and consider how they behave if, for example, we attempt to evolve a static fluid on a curvilinear background, where we would hope to keep the fluid static, regardless of the grid.

The slope-limiting step (4.14) will potentially limit the piecewise linear solution, where the cell volume is the only non-constant quantity. Further, since the quantities $\mathbf{u}_{i,L}^{n+\frac{1}{2}}$, $\mathbf{u}_{i,R}^{n+\frac{1}{2}}$ are meant to be evaluated at the cell faces, rather than the cell centres, we have to make a further adjustment. We replace (4.13)-(4.18) by

$$\begin{aligned} \Delta_i^- &= J_i \cdot \left(\frac{\mathbf{u}_i^n}{J_i} - \frac{\mathbf{u}_{i-1}^n}{J_{i-1}} \right), \quad \Delta_i^+ = J_i \cdot \left(\frac{\mathbf{u}_{i+1}^n}{J_{i+1}} - \frac{\mathbf{u}_i^n}{J_i} \right), \quad \Delta_i = \frac{1}{2} (\Delta_i^- + \Delta_i^+), \\ \mathbf{u}_{i,L}^n &= \frac{J_{i-\frac{1}{2}}}{J_i} \left(\mathbf{u}_i^n - \frac{1}{2} \Delta_i \right), \quad \mathbf{u}_{i,R}^n = \frac{J_{i+\frac{1}{2}}}{J_i} \left(\mathbf{u}_i^n + \frac{1}{2} \Delta_i \right), \\ \mathbf{u}_{i,L}^{n+\frac{1}{2}} &= \mathbf{u}_{i,L}^n + \frac{\Delta t}{2\Delta x} \left(\mathbf{f}(\mathbf{u}_{i,L}^n) - \frac{J_{i-\frac{1}{2}}}{J_{i+\frac{1}{2}}} \mathbf{f}(\mathbf{u}_{i,R}^n) \right), \\ \mathbf{u}_{i,R}^{n+\frac{1}{2}} &= \mathbf{u}_{i,R}^n + \frac{\Delta t}{2\Delta x} \left(\frac{J_{i+\frac{1}{2}}}{J_{i-\frac{1}{2}}} \mathbf{f}(\mathbf{u}_{i,L}^n) - \mathbf{f}(\mathbf{u}_{i,R}^n) \right), \end{aligned} \quad (4.49)$$

where J_i is the Jacobian of cell i , and $J_{i+\frac{1}{2}} = \frac{1}{2}(J_i + J_{i+1})$. We also take care to evaluate the fluxes, which require J and $\partial r^i/\partial x^j$ for their calculation, at the correct part of the cell, be it cell-centre or cell-face. We use a simple average of cell-centred Jacobians to find the Jacobian as required on cell-faces. The derivatives $\partial r^i/\partial x^j$, however, are averaged differently since, in one dimension, we know that $J^{-1} = \partial r/\partial x$, and so we want to ensure that

$$J \Big|_{i+\frac{1}{2}} \frac{\partial r^j}{\partial x^j} \Big|_{i+\frac{1}{2}} = 1 \quad \forall j, \quad (4.50)$$

where there is no sum over j .

One way to ensure that this is always true is to average the derivatives of the inverse coordinate mapping using the harmonic mean

$$D_{i+\frac{1}{2}} = \frac{2}{D_i^{-1} + D_{i+1}^{-1}}, \quad (4.51)$$

where

$$D_i = \left[\frac{\partial r^l}{\partial x^m} \right] \Big|_{x=x_i}. \quad (4.52)$$

Even in the case of multi-dimensions, we never need to calculate the fluxes across more than one cell-face at a time, so the above averaging is only ever done in one computational grid direction at a time.

4.6.1 HLL scheme

In order to calculate the HLL flux in curvi-linear coordinates, we first transform the Cartesian components into the local grid coordinate basis. The wave speeds are then calculated for the required coordinate direction from the transformed variables. As for SLIC, we calculate the fluxes in the local grid coordinate direction, multiplied by the local Jacobian. In order to achieve second order convergence, we adjust the HLL flux to read:

$$\mathbf{f}_{i+\frac{1}{2}}^{\text{HLL}} = \begin{cases} \mathbf{f}(\mathbf{u}_i) & \text{if } 0 < a_L, \\ \frac{a_R J_{i+\frac{1}{2}} \mathbf{f}(\mathbf{u}_i) - a_L J_{i+\frac{1}{2}} \mathbf{f}(\mathbf{u}_{i+1}) + a_L a_R (J_{i+\frac{1}{2}} \mathbf{u}_{i+1} - J_{i+\frac{1}{2}} \mathbf{u}_i)}{a_R - a_L} & \text{if } a_L < 0 < a_R, \\ \mathbf{f}(\mathbf{u}_{i+1}) & \text{if } a_R < 0, \end{cases} \quad (4.53)$$

where the fluxes are calculated parallel to the local grid coordinate i . We note that the local CFL number required for the calculation of GFORCE is also derived from the wave speeds calculated along the local grid coordinate directions.

4.6.2 Determination of time step

On Cartesian grids, the maximum time-step that retains stability is given by equation (4.8). On curvi-linear grids, we determine the time-step on a cell-by-cell basis, where the cell size is matched to the data in that cell. The alternative, taking the minimum cell size and the maximum wave speed over the whole grid, can lead to time steps that are more restrictive than necessary. Further, when we wish to calculate the wave-speeds, we in fact transform the numerical solution into the local grid basis so that the exact wave speeds along the coordinate directions can be found, as these may not be aligned with Cartesian directions, which could lead to an underestimate of the local wavespeed.

However, a drawback of using curvi-linear grids is that the cells are not of similar size everywhere. Even with the above method of calculating the time-step, this still leads to an effectively lower CFL number for evolution on larger grid cells, as the time-step has been restricted by the stability condition

on smaller grid cells. Therefore a shock can become somewhat smeared on a coarser area of the grid only because of finer resolution elsewhere.

We reduce the effect of this problem by using as accurate a scheme as possible, namely GFORCE or HLL, and also by using AMR (see §4.9) appropriately.

4.6.3 Free-streaming correction

The obvious way to discretise the transformed fluxes is given in (4.47) as this leaves the new system of equations in the same form as the Cartesian form, and is therefore easy to implement in a pre-existing code. Although valid in the continuous case, (4.47) is not valid when discretised since

$$\begin{aligned} & \frac{\partial J\mathbf{u}}{\partial t} + J \frac{\partial \mathbf{f}^i(\mathbf{u})}{\partial x^i} = \mathbf{0} \\ \Rightarrow & \frac{\partial J\mathbf{u}}{\partial t} + J \frac{\partial r^j}{\partial x^i} \frac{\partial \mathbf{f}^i(\mathbf{u})}{\partial r^j} = \mathbf{0} \\ \Rightarrow & \frac{\partial J\mathbf{u}}{\partial t} + \frac{\partial}{\partial r^i} \left(J \frac{\partial r^i}{\partial x^j} \mathbf{f}^j(\mathbf{u}) \right) - \frac{\partial}{\partial r^i} \left(J \frac{\partial r^i}{\partial x^j} \right) \mathbf{f}^j(\mathbf{u}) = \mathbf{0}. \end{aligned} \quad (4.54)$$

The additional term as compared to (4.47) is analytically zero for a continuous mapping. However, it will not necessarily be so when discretised, and therefore needs to be considered when implementing a numerical algorithm, since we wish to maintain exact conservation on a discrete grid.

This extra term is composed of an evaluation of \mathbf{f} on a cell-centred value of \mathbf{u} , multiplied by a geometric term. This geometric term must be calculated in a certain way, dependent on the precise numerical method used, so that, for a constant state \mathbf{u} , the geometric terms will cancel exactly (at least up to round-off error).

The process of determining the appropriate term is fairly straightforward in practice. If we start with a fluid with constant density, velocity, and pressure (on the Minkowski metric for the time being), then it is possible to work through the SLIC algorithm by hand, ensuring that the states $\mathbf{u}_{k,*}^n$ and $\mathbf{u}_{k,*}^{n+\frac{1}{2}}$ are again this constant state, regardless of the geometry or the slope-limiter used. This is what has been done in order to construct the form of (4.49). The outcome is that the flux for each cell face is composed of the product of a flux, derived from the given constant state, and a geometry term of the form $J_{k+1/2} \frac{\partial r^i}{\partial x^j} \Big|_{k+1/2}$. Since the final flux for a given coordinate direction r^i is then simply a difference of these, the term that cancels these is

$$\left(J_{k+\frac{1}{2}} \frac{\partial r^i}{\partial x^j} \Big|_{k+\frac{1}{2}} - J_{k-\frac{1}{2}} \frac{\partial r^i}{\partial x^j} \Big|_{k-\frac{1}{2}} \right) \mathbf{F}(\mathbf{u}_k), \quad (4.55)$$

where the values of the Jacobian and coordinate derivatives have to be calculated in exactly the same way as in the main SLIC algorithm.

This simple additional term improves evolutions on curvi-linear grids greatly, keeping a constant state

fluid static up to round-off error (of the order of 10^{-17} for double precision) on even the most distorted grids.

4.7 Adjustments for a non-Minkowski metric

So far we have not made any mention of how we deal with a non-flat metric. Since our eventual aim is to evolve the metric with the fluid, we can only assume that we know the metric at the cell centres (i.e. at the same points as the fluid), rather than using analytic values wherever they are needed. The main issue is how we alter the SLIC scheme from §4.2.1 when it requires slope-reconstructed values to be evaluated at cell faces. A careful and rigorous approach to this is presented by Rossmanith *et al.* [90]. However, as this approach is somewhat involved, and computationally expensive, we use the following approach, which includes considerations for a curvi-linear coordinate system. This is a generalisation of the SLIC method on flat space, given in (4.49), and explicitly uses slope-limiting on primitive variables, rather than conserved variables. In the following we use \mathbf{p} for the primitive fluid variables, \mathbf{u} for the conserved fluid variables, and g for the metric variables. We also use $\mathbf{u}(g_i, \mathbf{p}_i)$ to define the conserved variables \mathbf{u} as a function of the primitive variables \mathbf{p} and the metric g , where the subscript i refers to the cell in which the quantities are calculated. The following is for the case where the metric is stationary, and is not being evolved:

$$\text{Define } \Delta_i^- = \mathbf{p}_i - \mathbf{p}_{i-1}, \Delta_i^+ = \mathbf{p}_{i+1} - \mathbf{p}_i, \Delta_i = \frac{1}{2}(\Delta_i^+ + \Delta_i^-), \xi = \xi(\Delta_i^-, \Delta_i^+), \quad (4.56)$$

then find slope-limited values

$$\mathbf{u}_{i,L} = \mathbf{u}(g_i, \mathbf{p}_i) - \frac{1}{2}\xi\Delta_i, \mathbf{u}_{i,R} = \mathbf{u}(g_i, \mathbf{p}_i) + \frac{1}{2}\xi\Delta_i, \quad (4.57)$$

average the metric for cell-faces

$$g_{i-\frac{1}{2}} = \frac{1}{2} \left(g_i + \frac{J_i}{J_{i-1}} g_{i-1} \right), g_{i+\frac{1}{2}} = \frac{1}{2} \left(g_i + \frac{J_i}{J_{i+1}} g_{i+1} \right), \quad (4.58)$$

adjust the fluid variables for the cell-face metric

$$\mathbf{u}_{i-\frac{1}{2},L} = \frac{J_{i-\frac{1}{2}}}{J_i} \mathbf{u}(g_{i-\frac{1}{2}}, \mathbf{p}(g_i, \mathbf{u}_{i,L})), \mathbf{u}_{i+\frac{1}{2},R} = \frac{J_{i+\frac{1}{2}}}{J_i} \mathbf{u}(g_{i+\frac{1}{2}}, \mathbf{p}(g_i, \mathbf{u}_{i,R})), \quad (4.59)$$

and then continue as in (4.49), replacing $\mathbf{u}_{i,*}^n$ by the $\mathbf{u}_{i+\frac{1}{2},*}^n$ found above. These calculations are somewhat expensive since they involve several conversions between primitive and conserved variables. However, we have found them to improve accuracy substantially, particularly for three-dimensional simulations.

The main improvement to the fluid evolution comes through the use of averaging the metric correctly for the cell faces as above. Our reason for using the primitive variables on which to limit is due to the fact that, on a curvilinear grid, the mixing that occurs between different components of the momentum

can cause a flow in the x -direction to develop non-zero velocities in the y -direction. Removing the mixing effect of the Lorentz factor on the momenta by limiting on primitive variables reduces this effect, but does not remove it altogether.

Any further adjustments to the metric averaging process tend to be inconsistent with the free-streaming correction given in §4.6.3. We have not found a way of combining a metric averaging with the final free-streaming correction to correct the coordinate mixing errors completely, so as to maintain a static fluid on a stationary metric.

We note that we only limit on the primitive variables when there is a non-flat metric involved. Otherwise, we limit on the conserved variables.

4.8 Boundary conditions

In dealing with excised black holes, there are two types of boundary condition that must be considered: inner (excision) boundary conditions, and outer (spatial infinity) boundary conditions.

4.8.1 Excision boundary conditions

In theory, boundary conditions should not be needed at an excision boundary, provided that it is within the black hole’s apparent horizon, since no physical information can emerge from an apparent horizon. However, most numerical methods do not obey this precisely, as a method may use cells further inside the black hole in order to provide a local reconstruction of the solution. There are methods which overcome this, for example, the causal differencing approach of Gundlach and Walker [54], but they tend to be computationally expensive. Simpler methods can lead to satisfactory solutions if sufficient care is taken with the boundary conditions.

When the metric is evolved, its values are extrapolated to second order, i.e. linearly, into two ghost cells outside the boundary. Metric variables tend to be smooth, and so no problems are caused by this. Indeed, this is sufficient to maintain overall second-order accuracy.

If we are evolving the fluid, then we must restrict ourselves to a simple copying of the *conserved* variables from just inside the grid onto the two ghost cells, multiplying by the ratio of the metric determinants in the respective cells. Any higher-order extrapolation, unless checked carefully with extra routines, could lead to negative densities or pressures, which would immediately cause the method to fail.

If we are evolving the fluid alone, we do not extrapolate the metric at all, but set it to its analytic value. This is usually not necessary since the metric is set at the start of the simulation, but is sometimes required for the boundaries of AMR grids (see §4.9) that are generated during the simulation and which touch the excision boundary.

4.8.2 Boundary conditions at spatial infinity

We should like to have our outer boundaries at spatial infinity. However, due to limited computational resources, we must restrict ourselves to a finite domain. There do exist ways of capturing infinite spacetimes on a finite grid, such as the use of a conformal mapping of the metric [4] but these require more computational expense and more complex numerical algorithms.

Metric outer boundary conditions are again performed using linear extrapolation if necessary, although exact metric values are used if we do not evolve the metric. In this thesis, this suffices, although if it is wished to extract gravitational wave data from the outer boundaries, then especial care must be taken not to generate spurious waves at the boundary.

The fluid boundary conditions depend on the problem we are trying to solve. In the case of wind accretion (Chapter 7), the incoming fluid is defined using fixed Dirichlet boundary conditions, and the boundary conditions where the fluid is outgoing are that the fluid is copied in conserved variables, where no allowance is made for the change in metric as at the excision boundary, since we assume the metric to be nearly flat there.

4.8.3 Reflective boundary conditions

Reflective boundary conditions, which we will need for some SRHD tests, are fairly straightforward. As a first attempt, we apply even symmetry to the D and τ variables, setting values in the n^{th} ghost cell to be equal to those in the n^{th} cell inside the boundary. The momentum variables \mathbf{S} then use Overture's built in `vectorSymmetry` boundary condition, which ensures that $\mathbf{n} \cdot \mathbf{S}$ is an odd function across the boundary, and $\mathbf{t} \cdot \mathbf{S}$ is an even function, where \mathbf{n} is the normal to the boundary, and \mathbf{t} is any vector tangential to the boundary.

However, this is not completely robust. It is possible that a strong rarefaction may occur at a reflective boundary, if fluid is moving away from it, creating a near vacuum, and this could lead to a negative density if the numerical method is not TVD. To avoid this case, we calculate $\mathbf{n} \cdot \mathbf{S}$ and if it is more negative than a given parameter, then we set the ghost cell values for the momentum using even symmetry.

4.9 Mesh refinement

One problem often encountered in numerical computations is that of resolution. For example, in the case of a fluid falling into a black hole, there is very little going on far from the black hole. However, the fluid may develop very fine features near the black hole which would need to be resolved accurately. If a sufficiently high resolution to resolve these were to be used over the whole domain, there would be a lot of unnecessary computation far from the black hole, and the overall evolution would proceed very slowly.

There are two approaches to mesh refinement, depending on whether it can be predicted in advance where the most accuracy will be required.

Adaptive Mesh Refinement (AMR) uses a hierarchical set of grids, where one relatively coarse grid is used to cover the complete domain, and levels of successively finer grids are used in regions where high resolution is required. See Figure 4.7 for an example. The finer grids used are repositioned every few timesteps, so that they adapt to the evolving solution. This decreases the amount of computation used in regions of low activity and smooth solution, as we know that the numerical methods we use have a higher order of accuracy on smooth solutions than near discontinuities. Further, since the grids can be evolved independently of each other, apart from some interaction at the boundaries and when interpolating or extrapolating, the computation can be efficiently parallelised, giving different grids to different processors of a multi-processor computer, although Overture in fact splits individual grids over different processors instead.

The most widely used approach for AMR was developed by Berger and Olinger [19], which refines in both space and time, thus maintaining the effective CFL number of the finite volume schemes being used, and therefore maintains accuracy by minimising diffusion.

In theory, using AMR with a certain effective resolution (the resolution of the finest grid) should give the same results as evolving the entire system at that finest resolution. In practice this may not quite be achieved, but it is possible to come very close to the ideal.

The second approach is Fixed Mesh Refinement (FMR), which is a special case of AMR where the grid structure is not changed throughout the simulation. In this case, the grid structure can be carefully specified beforehand, and then no overhead regarding grid generation need be incurred. However, since in Overture we have sufficient routines to enable us to implement AMR, we do not consider this simplification again.

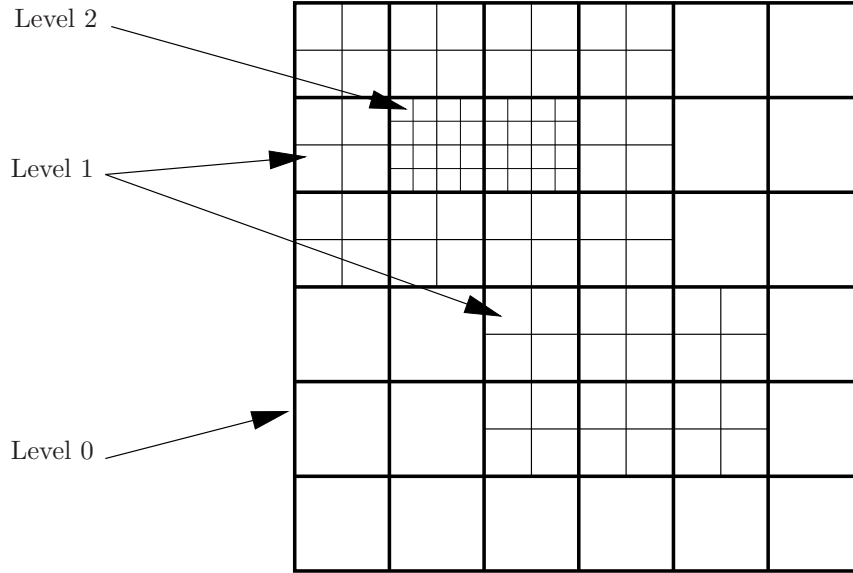


Figure 4.7: *Example AMR grid setup. The finer lines correspond to refined grids, with a refinement ratio of two. Levels 0 and 2 consist of one rectangular grid each, and Level 1 consists of two separate rectangular grids.*

4.9.1 Grid creation

In order to be able to use AMR, we must create an appropriate grid structure. This is done as follows:

Error flagging

The first step is to flag points on the coarsest grid where we expect large numerical errors to occur. One way of doing this is to evolve the coarse grid and an even coarser version of the grid forward by one time-step and to compare the solutions. Where the solutions differ by more than a certain threshold, we flag a grid point on the original coarse grid.

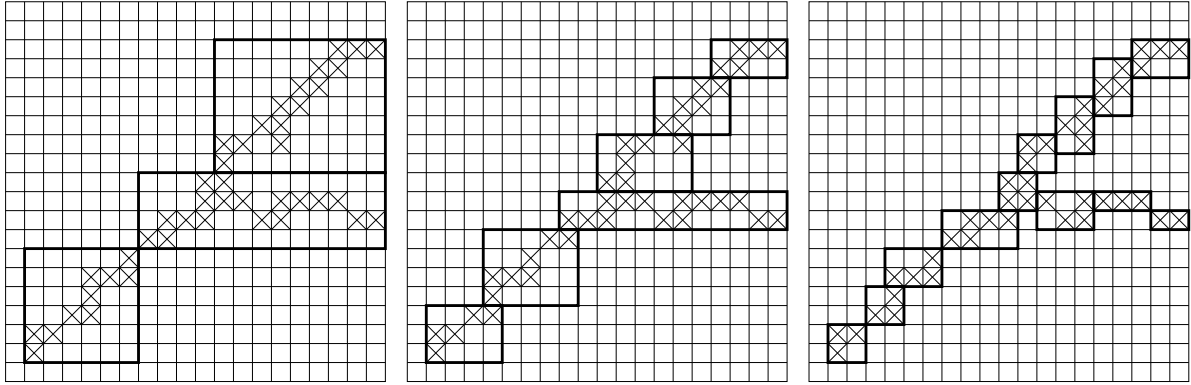
An alternative, which we follow here, is to create some criterion, based on the evolution variables and their derivatives, which tends to pick out places where large errors in the numerical method are likely to occur. For example, we could refine where the density gradient exceeds a certain threshold:

$$\rho^{-2} \gamma^{ij} \rho_{,i} \rho_{,j} > \epsilon_{\text{tol}}, \quad (4.60)$$

which would tend to pick out regions where there are steep gradients in the fluid density. It has the added advantage that it is scale invariant, i.e. if we double the density everywhere, then the refinement occurs in exactly the same places.

Another possible refinement criterion is to consider the divergence of the momentum

$$\gamma^{ij} S_{i,j} < -\epsilon_{\text{tol}}. \quad (4.61)$$



(a) Possible grid refinement with an efficiency of 0.28 (b) Possible grid refinement with an efficiency of 0.47 (c) Possible grid refinement with an efficiency of 0.73

Figure 4.8: Series of possible grid structures to refine a grid given the marked points as being flagged.

as suggested by Ruffert [93] since “in shocks matter is decelerated and thus a high negative value of the divergence [of the velocity] appears.”

Once the initial flagging is complete, we extend the flagging regions slightly so as to include nearby regions. This means that when a shock moves, it will still be inside the flagged region, thus retaining the accuracy of the method.

Boxing flagged points

Once we have flagged points, we must create grids which contain them. This must be done with some care; if we simply cover all the flagged points with a single grid, we are likely to cover too much space and to waste computational resources. On the other hand, if we cover only points that have been flagged, we run the risk of creating too many small grids with a lot of computational overhead. We therefore specify an efficiency (we choose 0.7), which dictates the minimum proportion of each grid that should be filled with flagged points. An example with various efficiencies is shown in Figure 4.8.

Once the grids have been decided upon, we also extend them by a few cells in all directions so that the features we are seeking to refine upon do not immediately move outside the refined region. With this, we can avoid redefining the refined grid structure every time-step and only redefine it every four steps, for example.

Child grid creation

The grids are created as described above, with an appropriate refinement factor. This is often taken to be a power of two, although this is not necessary, and any integer ratio is in fact permissible. However, we use a refinement ratio of two throughout this work. Once the first set of grids has been created, we repeat

the process on the new grid level to obtain another refined level, up to some user-defined maximum.

In this way, each refined level is entirely contained within the level one stage coarser than it. In some implementations, it is further required that a refined grid must be contained entirely within a single grid on the next coarser level, although this is not a necessary part of the AMR algorithm, and is not followed by Overture itself, for example. We therefore have the concept of a hierarchical grid structure, which can be coded efficiently within modern programming languages.

When we change the grid structure, we need to find solution data on the newly created grids. This is done by interpolation or copying from the most refined data available. For example, if the grid structure for the third refined level is changed, then the data for those grids can be obtained from data on the second refined level and any solution data previously obtained on the third refinement level. This can always be done due to the parent-child relationship described above.

With the exception of specifying and evaluating the refinement criterion, all the above grid creation steps can be carried out by Overture.

4.9.2 Sub-cycling

Once the grid structure has been determined, we need to find a way of evolving the system consistently so that it retains the accuracy of the other numerical methods used, and does not introduce any new errors. We could simply evolve the grid using the time-step found from the finest grid. However, this will give rise to more diffusive results than necessary on the coarsest grid.

We therefore use a scheme in which all grids are evolved with the time-stepping appropriate to their resolution in a process called sub-cycling. This proceeds as follows for a grid with a single level of refinement:

- Start with all grid levels at time t .
- Impose physical boundary conditions on Level 0
- Store the solution on Level 0 in a temporary array
- Advance the solution on Level 0 by one time-step Δt
- Advance the solution on Level 1 by one time-step Δt as follows:
 - Impose physical boundary conditions on Level 1
 - Advance the solution on Level 1 by a half time-step $\Delta t/2$
 - Interpolate non-physical boundary conditions on Level 1 grids from the two stored solutions on Level 0 at times t and $t + \Delta t$.

- Impose physical boundary conditions on Level 1
- Advance the solution on Level 1 by a half time-step $\Delta t/2$
- Project data on Level 1 back onto any relevant cells on Level 0.
- All levels now have the most accurate data available for time $t + \Delta t$.

This can be easily extended to multiple levels of refinement. In practice, the sub-cycling algorithm is implemented as a recursive procedure.

In order to maintain flux conservation of our scheme, we should apply a fix-up for the fluxes at cell-faces on grid cells that border a refined grid. However, we have not found this loss of conservation to be a problem, and we postpone applying this correction for future work.

4.9.3 Overlapping grids

With regard to overlapping grids and AMR, there is no increase in complexity once both are implemented individually within Overture. AMR grids are defined as refinements of the main base grids of the overlapping grid setup. We have to be somewhat careful as to when we interpolate between grids, and not to mix data from different times. If, however, we only ever interpolate between grids on the same refinement level at once (something Overture lets us do quite simply), then this is not a problem.

4.9.4 Refinement criterion

One other consideration that must be made is that of the refinement criterion. In CFD, this may be related in some way to local truncation error, which requires the use of shadow grids to calculate. Also, in CFD we usually only refine where shocks, or at least steep gradients, occur. In NR, it would also be useful to refine in the presence of strong gravitational fields, which could nevertheless be smooth, since we expect highly dynamic flows here.

When considering the refinement criterion, there is also the issue of coordinate system. If we refine a spherically symmetric black hole then, even if it is represented in a Cartesian coordinate system, we would wish that the refinement be spherically symmetric. In order to achieve this, our refinement criterion must be a scalar comparison, i.e. independent of the coordinate system used. We use separate criteria for the metric and the fluid, respectively:

$$\begin{aligned}
& {}^{(3)}R = -2\partial_k V^k + D_k{}^{rs}\Gamma_r{}^k{}_s - D_{kr}{}^r D^{ks}{}_s > \epsilon_{\text{tol}}, \\
& \text{and } \frac{\gamma^{ij}\partial_i\rho\partial_j\rho}{\rho^2} > \epsilon_{\text{tol}} \text{ in three dimensions, or } \gamma^{ij}S_{i,j} < -\epsilon_{\text{tol}} \text{ in two dimensions.}
\end{aligned} \tag{4.62}$$

We note that these are derived from 3-vectors in the space-like hypersurface, not from 4-vectors in the full space-time, and are therefore not gauge independent.

4.10 Overlapping grids

Having decided to use overlapping grids on which to evolve the fluid and the metric, it is necessary to make a choice as to whether to have the variables on each grid defined in a coordinate system specific to each grid, or to have all the variables defined in a global coordinate system.

The use of a local coordinate system could be advantageous on a single grid as we would define the variables in the coordinate system of that grid, which could lead to simpler numerical methods which would more easily satisfy conservation conditions. However, the interpolation between grids would require the construction of a transformation between the two coordinate systems of the grids involved.

A global coordinate system removes the latter problem, but requires consideration of the issues presented in §4.6. Also, for some space-times, a global coordinate system may not exist. However, all the metrics we use do have a global coordinate system, at least on the portion of the space-time we wish to cover. For example, the Boyer-Lindquist coordinate system (see §3.9) does not have a global continuous coordinate basis if we were to extend our grid inside the black hole, but we do not do this.

In our case, Overture has made the decision for us. Overture already performs the interpolation between grids itself, and we would need to rewrite and specialise that part of the Overture software if we wished to have some variables expressed in local coordinate systems. We have therefore decided to use a global coordinate system, specifically, Cartesian for ease of implementation and use.

There are two main issues with regard to dealing with overlapping grids. The first is that of constructing the full grid structure. This is one of the tasks that Overture has been designed to perform for even very complicated grids. Given a set of logically rectangular grids, along with mappings to the physical grid coordinates, as well as boundary conditions, Overture can remove parts of grids that are obscured by other grids and calculate which cells on which grids should be used to interpolate data onto grid cells that are now near cut-out portions of the grid. In implementing our algorithm, we have treated this very much as a black-box algorithm and, since the grids we use have been relatively simple, we have not needed to investigate the algorithm in any detail.

Cells that are obscured by other grids are masked out, and it is possible to determine, within a code, which these points are so that we can avoid expending any computational resources in evolving them.

Once we know which cell values will be interpolated from which other cell values on which grids, we still need to decide what form the interpolation will take. If we have allowed an interpolation point to interpolate from points on another grid that are themselves interpolation points, then we need to use an implicit, matrix-based method to find the interpolated values. This tends to be fairly computationally expensive, but does allow for relatively small grid overlap, which may be needed in complex grid setups in order to get a valid interpolation scheme for the entire grid.

Alternatively, we can use explicit interpolation. Here, interpolation points only need to use values

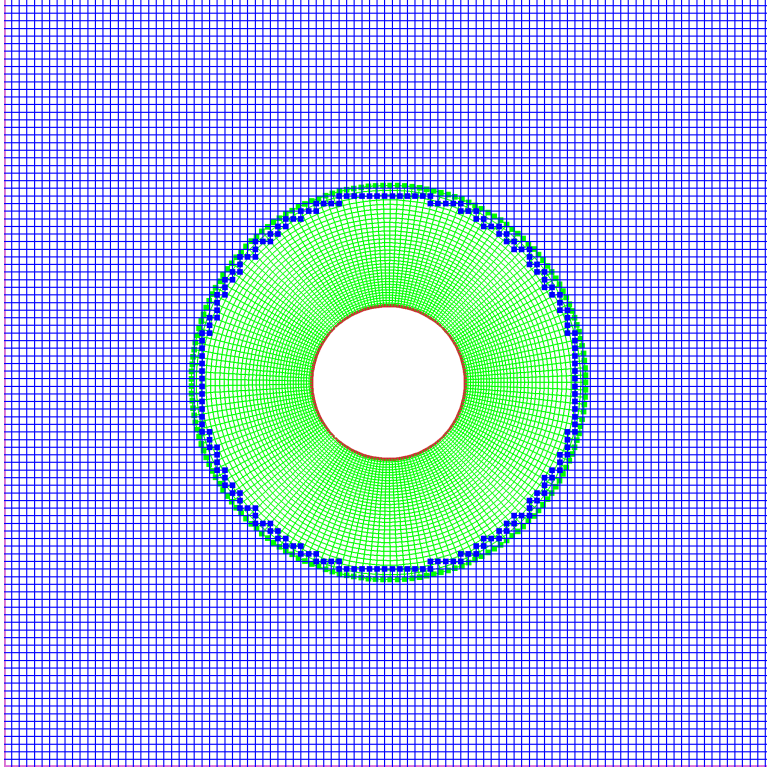


Figure 4.9: *Overlapping grid generated by Overture. Interpolation points are marked.*

from the interior of other grids. Each point can therefore be determined as an explicit function of a set of values on the interpollee grid. Overture does this using Lagrange polynomials, and takes into account the grid geometries involved. The main drawback of this is that it is not guaranteed that the interpolated values will be contained in the same range as the values from which they are interpolated. This could lead to overshoots developing at grid boundaries. However, we have not found this to be a problem, and correcting for it was found to be very computationally intensive as a result of the conditional statements involved.

In practice, we used fifth-order interpolation, which allowed for two lines of interpolation points at the edge of each grid. We needed this as the slope-limiting methods we use require two ghost cells at the grid boundaries. However, we note that we could have used a lower order of interpolation while still having sufficient interpolation points for our scheme.

Figure 4.9 shows an example grid generated by Overture with the interpolation points marked. This demonstrates the smooth excision of a circle from a rectangular region using an annular grid overlapping a Cartesian grid.

4.10.1 Cell size considerations

One thing that is important when defining suitable grids is the grid spacing. It is best for the interpolation routines if the grid cells that overlap are of approximately the same size, as then the interpolated values have an approximate one-to-one correspondence with the interpolatee values. If there were fewer interpolated points than points to be interpolated from, then information could be lost, and if there were more interpolated points, then the interpolated solution could be less accurate as several cells would be interpolated from the same points on the other grid.

Also, due to the CFL number, if there were some cells on the grid that were much smaller than others, then the small cells would probably have the dominant effect on the CFL condition, lowering it with respect to the larger cells. The result would be that solution would become more diffused on the larger cells than would otherwise have been the case.

We reduce the effect of this problem by only using a very few cells in the radial direction of the annulus shown in Figure 4.9. We also apply a stretching to the radial coordinate of the annulus so that the cells are all approximately square (see §7.3.2 for details). When we consider a spherical grid later on, we use the same approach.

4.10.2 Keeping variables physical

The conserved fluid variables do not, by themselves, define the primitive variables, but require the metric for the transformation. It is therefore the case that simply copying or interpolating conserved fluid variables may cause them to become unphysical if the metric in the cell to which they are copied causes the violation of the inequality

$$(D + \tau)^2 \geq \gamma^{ij} S_i S_j . \tag{4.63}$$

Some physical boundary issues have been considered in §4.8, and in practice no problems have been found at interpolation boundaries. Further, it tends to be quite a demanding exercise to enforce physicality of the variables, due to the presence of the $\sqrt{\gamma}$ factor. In practice, all we can easily enforce is $D > 0$, and this is our reason for using the fall-back techniques given in §3.3.2.

It is still possible for the numerical methods to cause unphysical fluid conditions, as they have not been designed to deal with such problems. However, the above considerations certainly reduce the number of places where such errors can arise.

4.11 Overall time-stepping approach

Now that we have described all the numerical methods we intend to use, we need to explain how these fit together to create a consistent, robust, and accurate algorithm. For clarity, we omit any AMR related

operations; it should be clear how the following combines with the AMR sub-cycling approach in §4.9.2. The following is the procedure we used to perform a single time-step of a single refinement level:

- Update grid ghost boundaries (ghost cells used due to the splitting of a grid across multiple processors)
- Interpolate between all grids on this refinement level
- Impose physical boundary conditions
- Advance solution by source evolution for time $\Delta t/2$
- Update grid ghost boundaries
- Interpolate between all grids on this refinement level
- Advance solution by HRSC flux method for time Δt (if we are evolving both the fluid and the metric, then either both are evolved using the same scheme, or we use Strang splitting to evolve them using different schemes, i.e. half a time-step for the fluid, a full time-step for the metric, and a final half time-step for the fluid)
- Update grid ghost boundaries
- Interpolate between all grids on this refinement level
- Impose physical boundary conditions
- Advance solution by source evolution for time $\Delta t/2$
- Impose physical boundary conditions
- Interpolate between all grids on this refinement level

We have not included issues such as solution output and grid generation. These are only performed after a time-step and consist largely of calls to the Overture infrastructure. Also, we assume that the solution is stored in conserved variables. The entire solution is only converted back to primitive variables when necessary for output although, of course, individual cells may be converted to primitive variables for flux calculations. Further, it is possible that not all of the interpolations carried out in the above are, in fact, necessary. In particular, the interpolation following the source evolution may not be needed, and the interpolation at the end of the time-step is equivalent to that carried out at the beginning of the (following) time-step.

The way we have done this means that the solution remains consistent and the evolution remains second-order in both time and space. We demonstrate this in the next chapter.

Chapter 5

Validation of schemes

In this chapter we present the various tests that we have performed on our algorithm and its implementation in order to confirm that they perform as intended. The aim of these tests is twofold. Firstly, we use them to confirm that we have implemented standard schemes correctly. Secondly, we use them as a test-bed for some of the new schemes and approaches that are being used in this algorithm, to demonstrate that they improve accuracy, and are well-suited to the types of problems we are addressing.

5.1 One-dimensional SRHD Tests

In this section we present validations of the numerical schemes presented in Chapter 4, using the tests presented in §3.6.1. The first three tests were carried out using a simple one-dimensional uniform resolution version of our algorithm implementing the same numerical schemes as the Overture-based implementation, but without the overhead imposed by Overture. The last two tests were carried out using the Overture-based implementation as they required AMR and non-uniform grid capabilities.

5.1.1 Mildly Relativistic Shock

Here we evolve the IVP found in (3.51). It has been noticed by other researchers (e.g. Del Zanna and Bucciantini [117]) that, for certain schemes, the initial zero pressure has to be set to some small non-zero value. We use a minimum pressure of 10^{-8} .

An example output is shown in Figure 5.1 using the van Leer slope limiter, a CFL number of 0.99, based upon GFORCE³. The right-most discontinuity, corresponding to a shock wave, is captured with only four points. In this, and in the capturing of other discontinuities, we see that our results are as good as those of Del Zanna and Bucciantini. However, the capturing of the density is seen to be somewhat better in Lucas-Serrano *et al.* [71]. We also note that the kink we find in the velocity at $x \approx 0.6$ is present

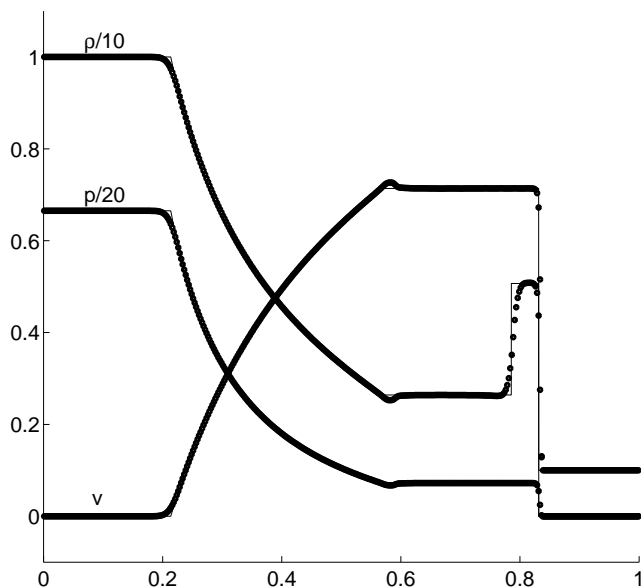


Figure 5.1: Results for a mildly relativistic shock wave, $(\rho, u, p)^L = (10, 0, 13.3)$ and $(\rho, u, p)^R = (1, 0, 0)$, using 400 cells, a CFL number of 0.99, and the van Leer limiter, with the SLIC scheme based on the GFORCE³ flux. The exact solution is give by the solid line.

to a slightly lesser extent in [71], and not at all in [117].

Figure 5.2 shows the result of using four different numerical schemes for the same problem. All the methods use the SLIC scheme, based on four different first-order fluxes: FORCE, GFORCE, GFORCE³, and HLL. It can be seen that there is very little difference indeed between the methods. This is due to the mild and simple nature of the problem.

However, we can make the differences between the methods more apparent by evaluating the \mathcal{L}_1 error norm on the density for each of several methods, using a range of CFL numbers. The results are shown in Table 5.1. The error norm is taken over the whole domain and so includes the discontinuities.

We see that increasing the number of stages used for a scheme reduces the error. In all cases, the GFORCE scheme is better than the FORCE scheme, although the FORCE² scheme is always better than the GFORCE² scheme, except for the case with CFL 0.4, where the numerical evolution blows up. We note this data as evidence for the *caveat emptor* in §4.2.5. The reason for the improvement of FORCE^k over GFORCE is that the FORCE^k scheme is usually above the GFORCE scheme as shown in Figure 4.5, as a result of its lessening diffusion at the expense of potentially introducing oscillations into the solution. The HLL scheme is always better than the GFORCE scheme, but usually worse than the GFORCE² scheme. In this problem, there is very little improvement when increasing the number of

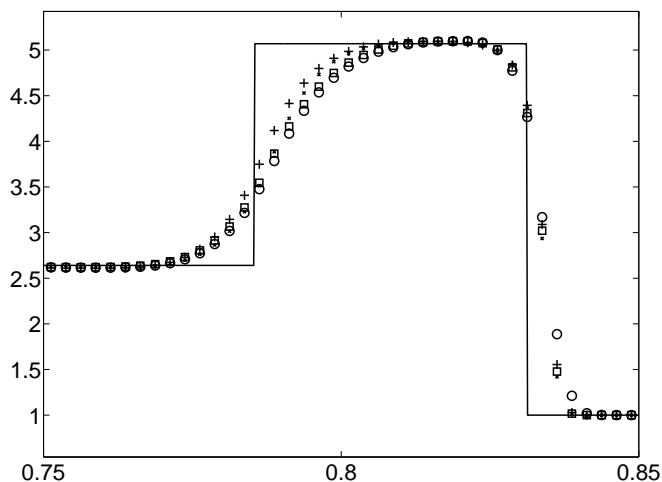


Figure 5.2: Results for the density of a mildly relativistic shock wave, $(\rho, u, p)^L = (10, 0, 13.3)$ and $(\rho, u, p)^R = (1, 0, 0)$, using 400 cells, a CFL number of 0.99, and the van Leer limiter, with the SLIC scheme based on the fluxes: FORCE (circles), GFORCE (squares), GFORCE³ (crosses), and HLL (+ signs). The exact solution is given by the straight line. We restrict the region plotted so that the differences between the methods can be seen.

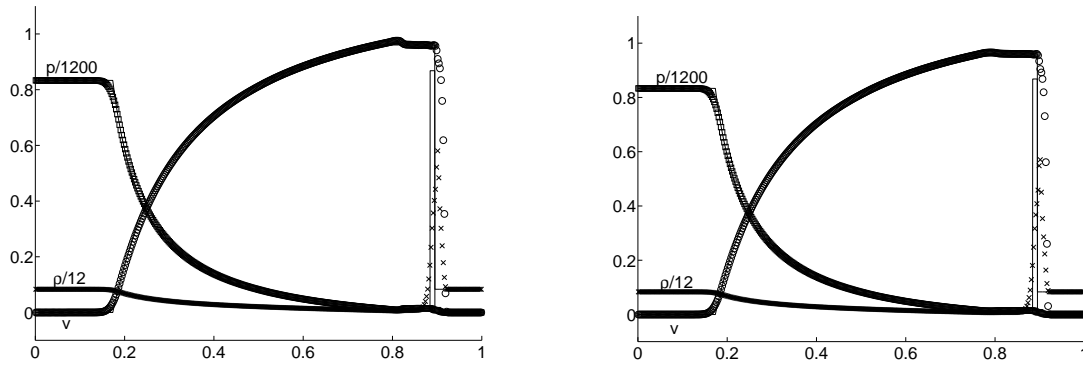
Scheme	CFL 0.4	CFL 0.8	CFL 0.9	CFL 0.99
FORCE ¹	8.68×10^{-2}	5.81×10^{-2}	5.14×10^{-2}	4.61×10^{-2}
FORCE ²	6.94×10^{48}	4.62×10^{-2}	4.08×10^{-2}	3.63×10^{-2}
FORCE ³	4.37×10^{-2}	4.00×10^{-2}	3.69×10^{-2}	3.32×10^{-2}
GFORCE ¹	6.14×10^{-2}	5.37×10^{-2}	4.53×10^{-2}	4.06×10^{-2}
GFORCE ²	5.89×10^{-2}	5.02×10^{-2}	4.31×10^{-2}	3.82×10^{-2}
GFORCE ³	5.70×10^{-2}	4.82×10^{-2}	4.28×10^{-2}	3.81×10^{-2}
HLL ¹	5.88×10^{-2}	5.44×10^{-2}	5.30×10^{-2}	5.13×10^{-2}

Table 5.1: Error norms of the density using $\|\cdot\|_1$ for various base schemes for the SLIC method applied to the mildly relativistic shock test. All were evaluated using the van Leer slope limiter on 400 cells. Note the blow-up evident for FORCE² with CFL 0.4.

stages used for the GFORCE scheme.

We also note that the error decreases for higher CFL numbers, as a result of the reduced number of time-steps required to reach the final time.

All of these conclusions are roughly in line with what we would expect from the analysis of the various



(a) Same scheme used for all time-steps

(b) Used FORCE scheme alone for first five time-steps

Figure 5.3: Results for a strongly relativistic shock wave, $(\rho, u, p)^L = (1, 0, 1000)$ and $(\rho, u, p)^R = (1, 0, 0.01)$, using 400 cells, a CFL number of 0.99, and the van Leer limiter, with the SLIC scheme based on the GFORCE flux. The exact solution is given by the solid lines. Note that the kink in the velocity seen in (a) is not present in (b), as a result of using a first-order scheme for the first few time steps.

schemes on the linear advection equation in §4.2.

5.1.2 Strongly Relativistic Blast Wave

The initial conditions from (3.52) produce a narrow region of high density moving close to the speed of light, as shown in Figure 3.2. The maximum density is very difficult to capture at a moderate resolution such as we are using due to the narrowness of the peak. A good test of the accuracy of a scheme, therefore, is the fraction of the peak value that is attained.

An example solution, obtained using the SLIC scheme based on GFORCE, using the van Leer limiter, with a CFL of 0.99, is shown in Figure 5.3(a). The solution is already quite accurate, and the density peak is at 65.6% of its correct value. Del Zanna and Bucciantini achieved peaks of 70.1% and 62.4% for two different versions of their method, and Lucas-Serrano *et al.* [71] attained a density peak of 76% of the exact value (the comparison is not quite valid as Del Zanna and Bucciantini only evolved to time $t = 0.35$, rather than $t = 0.4$ as used here and in [71]).

However, we note that there is an odd kink in the velocity solution, at $x \approx 0.8$. This is just visible in Del Zanna and Bucciantini [117], but to a lesser degree. We can remove this by setting the slope-limiter to zero, effectively using the GFORCE scheme alone, for the first five time steps only (out of 287 time steps for the whole evolution). The results are now shown in Figure 5.3(b). We use this approach for this problem from now on.

Scheme	Density peak
Exact	10.42
FORCE	6.84
FORCE ²	7.56
FORCE ³	8.04
FORCE ⁵	8.24
GFORCE	6.99
GFORCE ²	7.45
GFORCE ³	7.78
GFORCE ⁵	7.98
HLL	7.26

Table 5.2: *Density peak for strongly relativistic shock as predicted by various methods. The approximate methods all used CFL 0.99, van Leer limiter, and the first-order reduction for the first five steps as indicated in the main text.*

Once again, there is little visual difference between the various schemes, and we give the density peaks in Table 5.2. These results are in line with what we have already seen, that FORCE^k tends to be more accurate than GFORCE^k for $k \geq 2$, and that HLL is more accurate than GFORCE when no multi-staging is used.

In Table 5.3 we give the \mathcal{L}_1 errors in the density for various resolutions and methods. This shows near first-order accuracy over the whole problem, which is the best that we can expect over a solution containing discontinuities.

Resolution	Scheme	$\ \rho_{\text{exact}} - \rho_{\text{num}}\ _1$
400	FORCE	1.59×10^{-1}
	GFORCE	1.54×10^{-1}
	GFORCE ³	1.44×10^{-1}
	HLL	2.05×10^{-1}
800	FORCE	1.16×10^{-1}
	GFORCE	1.13×10^{-1}
	GFORCE ³	9.95×10^{-2}
	HLL	2.00×10^{-1}
1600	FORCE	6.27×10^{-2}
	GFORCE	6.13×10^{-2}
	GFORCE ³	5.07×10^{-2}
	HLL	1.11×10^{-1}

Table 5.3: \mathcal{L}_1 norm errors in density for the blast-wave test, evaluated over the whole domain. All simulations used the van Leer limiter, CFL 0.99, and the first-order reduction for the first five time-steps.

5.1.3 Perturbed Density Test

We now turn to the perturbed density test given by (3.53), which shows the effect of the multi-staging more clearly. As previously noted, there is no analytic solution to this problem, and so any reference to exact solution is to the result of a numerical evolution using the SLIC scheme based on FORCE¹, with a CFL of 0.99, the van Leer limiter, and 20 000 cells which we believe to have converged to very close to the correct solution since it is visually indistinguishable from that using 10 000 cells.

An example evolution, using 400 cells and the SLIC scheme based on FORCE, is shown in Figure 5.4. Points to note are that the three sharp peaks are not particularly well captured, and that the final peak at the right of the figure is clipped. We note that Del Zanna and Bucciantini manage to capture the peaks somewhat better than we do whilst using half the number of cells.

Figure 5.5 shows the results for various schemes, focusing on the region $0.6 \leq x \leq 1$. Now we see that increasing the number of stages used improves the capturing of the three peaks and the knees at $x \approx 0.67$ and $x \approx 0.85$. The ordering of the plots has been chosen to show a progressive increase in accuracy. We see that multi-staging GFORCE out-performs HLL, and that multi-staged FORCE still produces better accuracy than multi-staged GFORCE, though possibly at the risk of being unstable.

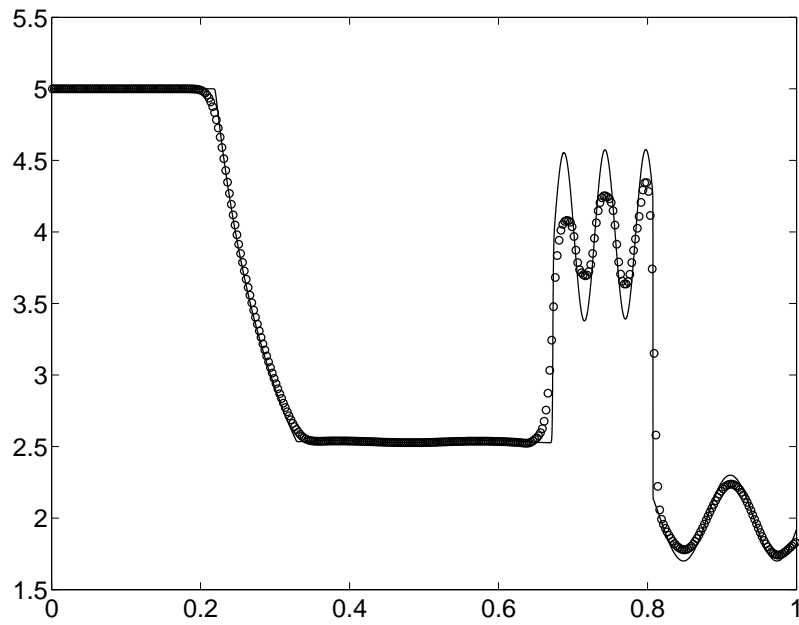
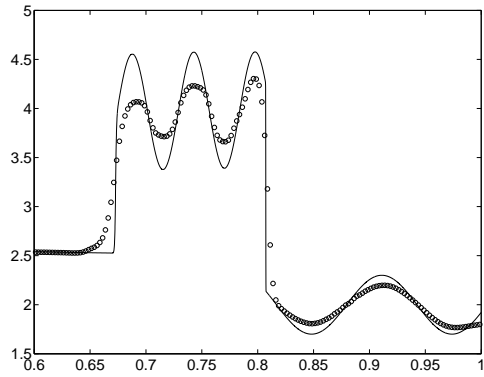
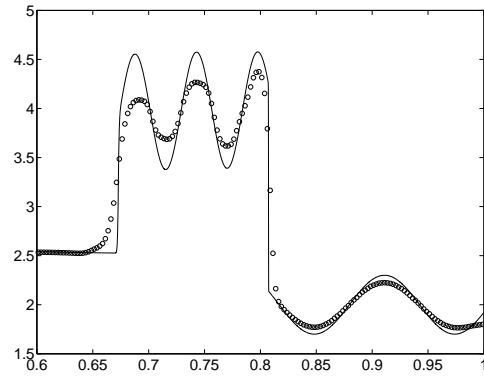


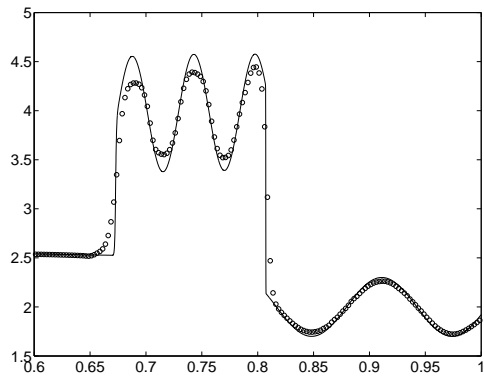
Figure 5.4: Results for the density for density perturbation initial conditions. The SLIC scheme based on $FORCE^1$ was used, with 400 cells, a CFL 0.99, and the van Leer limiter. The “exact” solution (see main text) is given by the solid line.



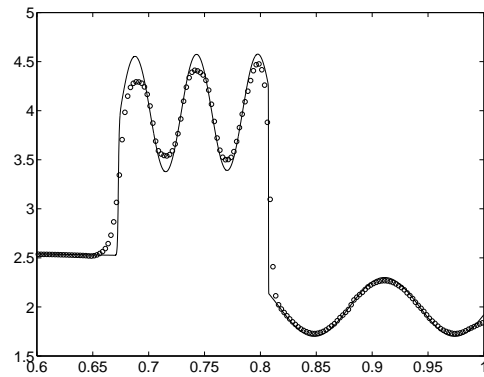
(a) *HLL*



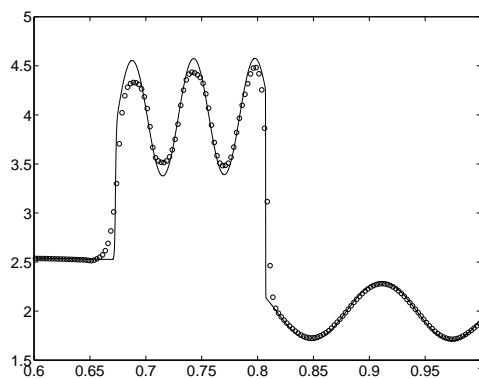
(b) *GFORCE*¹



(c) *GFORCE*³



(d) *FORCE*³



(e) *GFORCE*⁵

Figure 5.5: Close-ups of density results from the density perturbation test. For all results, a CFL of 0.99 and the van Leer limiter were used, on a grid of 400 cells.

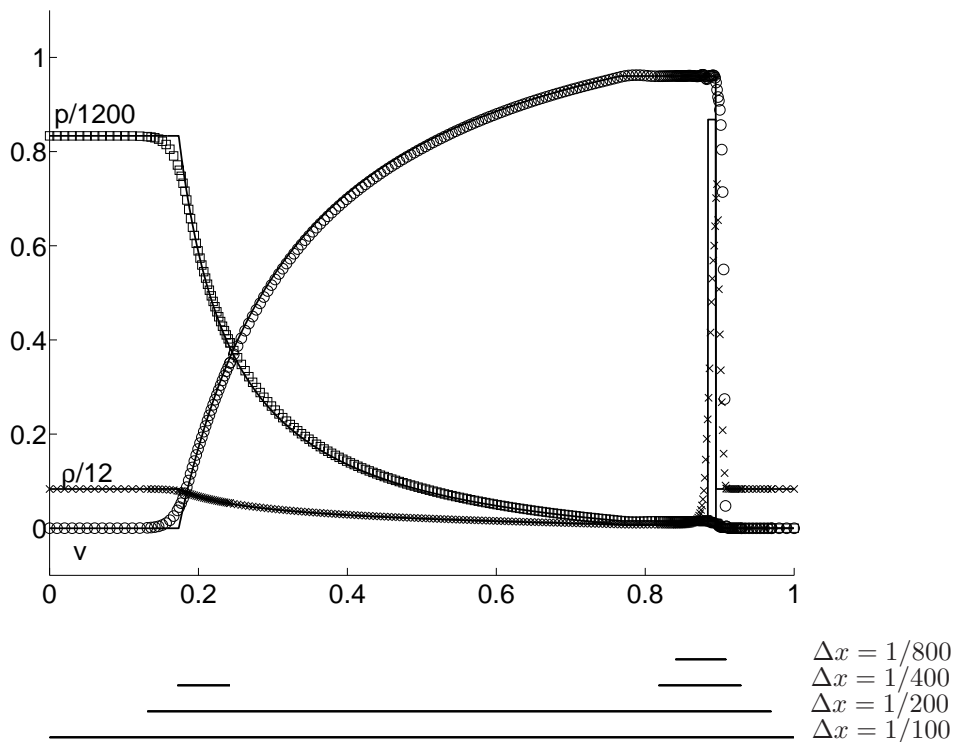


Figure 5.6: *Evolution of the blast wave test, equation (3.52) using a base grid of 100 cells, with the van Leer slope-limiter, GFORCE¹, CFL 0.95, and 3 levels of AMR, giving an effective resolution of 800 cells. The final grids are shown underneath.*

5.1.4 One-dimensional AMR

We can also check the AMR parts of the algorithm in one dimension. We use the blast wave test of (3.52) and evolve it on a uniform domain with 100 cells, but using three levels of refinement factor two, taking the effective resolution to 800 cells. The refinement criterion we use is that we flag a refinement point on a grid level k (coarsest level is zero) if

$$\frac{\rho^2_{,x}}{\rho^2} > 4 \cdot 2^{2k}. \quad (5.1)$$

The results are shown in Figure 5.6. The total number of cells advanced, summed over all time-steps and sub time-steps, for this calculation was 129 960 cells, about 27% of the equivalent figure for a single grid of 800 cells, 484 000 cells (i.e. 605 time steps with 800 cells at each step). The times are less than favourable, though, as the AMR calculation took 44.2s and the uniform calculation took 22.0s. These timings, however, will be improved upon for multi-dimensions, where the overhead due to AMR, in particular the regridding which we carried out here after every time-step, is proportionally far less.

The maximum density for the AMR simulation was 8.99, and that for an equivalent simulation on

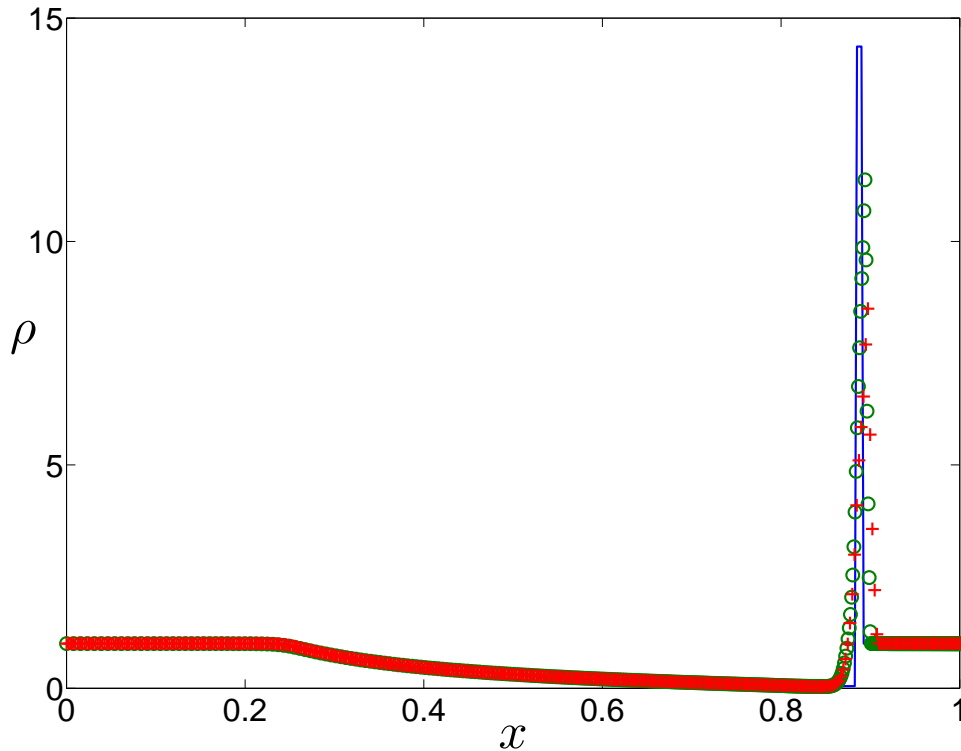


Figure 5.7: *Blast wave test, evolved using two different grids. The red + signs show the evolution on a grid with 400 equally spaced cells, and the green o signs show the evolution on a grid with 400 cells which are concentrated towards the right-hand end of the grid.*

a single grid of 800 cells was 8.91. While not identical, these are very close, suggesting that the AMR techniques are working robustly, and to the correct order of accuracy.

5.1.5 One-dimensional non-uniform grid

We should also test whether our algorithm is capable of attaining second-order accuracy on non-uniform grids as well. We use a grid in the range $x \in [0, 1]$, with smoothly varying grid spacing, such that the cells are smaller towards the right-hand end than at the left-hand end. We therefore have an effectively increased resolution where the density peak will finally form.

We ran the simulation using the GFORCE¹ flux, the van Leer slope-limiter, and a CFL of 0.95. The non-uniform grid we used had 400 cells. We compare the evolution on a non-uniform grid to that on a uniform grid in Figure 5.7. The extra resolution has clearly resulted in a better capturing of the density peak, with no apparent degradation of the solution in the relatively smooth region to the left. The maximum density in this case was 9.22, higher than for the uniform grid.

For the purposes of comparison, we note that the evolution with the uniform grid took 267 time-steps and 5.49s to run, while the non-uniform grid required 271 time-steps and 5.53s to complete. We have therefore shown that an improvement in accuracy may be obtained without a substantial increase in computational expense if we place grid-points appropriately. We have also shown that there is no obvious error in our implemented algorithm when applied to non-uniform grids.

5.1.6 Forcing an exact solution

We have been unable to locate any one-dimensional exact solutions to Einstein's equations incorporating a fluid (even for the test-fluid approximation) for a non-Minkowski space-time (in Cartesian coordinates), and so we use a different approach. The intention is to test the algorithm on the evolution of a fluid in a non-Minkowski regime. The flux terms are fairly straightforward, with little difference between those for Minkowski space and the general metric case. The source terms, however, require a large amount of calculation, and could well be the source of errors.

One approach to testing these is to implement a forced exact solution to Einstein's equations. This technique can be applied to any system of equations, and can be used to track down coding errors.

Given a general set of PDEs of the form

$$\frac{\partial u}{\partial t} + \frac{\partial F^i(u)}{\partial x^i} = S(u), \quad (5.2)$$

we can take an analytic function $u_{\text{exact}}(x^i, t)$, and define

$$R(x^i, t) = \frac{\partial u_{\text{exact}}}{\partial t} + \frac{\partial F^i(u_{\text{exact}})}{\partial x^i} - S(u_{\text{exact}}). \quad (5.3)$$

We now know that the function $u_{\text{exact}}(x^i, t)$ is an exact solution of the set of PDEs

$$\frac{\partial u}{\partial t} + \frac{\partial F^i(u)}{\partial x^i} = S(u) + R(x^i, t), \quad (5.4)$$

and therefore, if we incorporate the extra source term R into our code, we can check to see whether our calculation of R , usually generated using software such as MapleTM, agrees with that of F and S . We can test fluxes and sources separately, as we use MapleTM to calculate both exact fluxes and exact sources, so that the exact flux term can be combined with the numerically calculated sources, and vice-versa. Note that we may also need to adjust any boundary conditions used in order to take into account the new setup.

This system is not perfect for the following reasons: we do not check that the coded expressions are correct, as such, only that the two different ways of calculating them, through MapleTM and through the main code, agree. It is sufficient, however, to catch typographical errors in the implemented algorithm and probably some errors due to algebraic manipulation.

Even with an exact solution, the sources and fluxes will not necessarily balance, since we advance the flux terms using a conservative scheme, and the source terms using a Runge-Kutta scheme. It is not likely that this will be a problem for us, but there do exist systems of equations where this failure to balance is a major problem, and much research has been done to rectify it, e.g. Hubbard and Garcia-Navarro [60].

Note that, since the new system only differs from the original by the addition of source terms, it will have the same stability character as the original system, and therefore any error reductions in this system will likely be present in the original system as well.

Even given all these caveats, this approach still provides another way to check whether the algorithm and its implementation are functioning as they should. We note that it is not necessary only to look at final numerical evolutions; we can also compare the raw source terms calculated using the two different methods inside a debugger.

We can use this approach to test the effect of some of the numerical corrections suggested in §4.4 and §4.7. We evolve a simple test-case using this forced-exact solution, and compare the resulting errors. The corrections are simple variations on a theme that we know to be largely correct, so that we know that they should be applicable to evolving fuller, unforced, systems. (Simply aiming for accurate capturing of the fluxes and sources would tend to suggest high-order finite-difference methods, which would not be suitable.)

The five corrections we test are:

1. Whether to limit on primitive or conserved variables (§4.7)
2. Whether or not to perform the more careful metric averaging suggested in §4.7
3. Whether or not to include the flux/source balancing term (§4.4.3 - 1D only)
4. Whether or not to use the Simpson's rule improvement to the source calculation (§4.4.2)
5. Whether to use the combined source splitting, or Strang splitting (§4.4.1)
6. Whether to use the GFORCE flux or the HLL flux (both without multi-staging) (§4.2.3 and §4.2.4)

We test these in all possible combinations as applied to the following setup:

- Metric given by the stationary propagating gauge pulse of §3.7. This avoids having to use the metric evolution terms. We have to use a solution of the Einstein equations here as we wish to compare different ways of calculating the source terms that rely on Einstein's equations holding (see §3.2).
- Ideal fluid (§3.3.2) with adiabatic index $\Gamma = 4/3$ and initial state given by

$$\rho = 10^{-1} + 10^{-2} \cos(2\pi x), \quad v_x = 10^{-1} + 10^{-2} \cos(2\pi x), \quad p = 2 \cdot 10^{-3} + 2 \cdot 10^{-4} \cos(2\pi x) \quad (5.5)$$

1	2	3	4	5	6	$\ \rho_{\text{error}}\ _{\infty}$
✓	×	×	×	×	GFORCE	0.013297
✓	×	×	×	✓	GFORCE	0.012816
✓	×	×	✓	✓	GFORCE	0.012765
×	×	×	✓	✓	GFORCE	0.012753
×	✓	×	✓	✓	GFORCE	0.012656

Table 5.4: We show a short chain of gradual improvements to the evolution of a fluid on a curved space-time. The key to the columns is as follows: 1) Primitive variable limiting (§4.7) 2) Metric averaging correction (§4.7) 3) Flux-source balancing (§4.4.3) 4) Simpson’s Rule correction (§4.4.2) 5) Flux-source combination (§4.4.1) 6) GFORCE or HLL scheme (§4.2.3 and §4.2.4)

- Periodic boundary conditions (the initial conditions are periodic, and this removes any problems caused by boundary effects)
- Extra forcing term calculated so as to keep the fluid fixed at its initial state, assuming no numerical errors in the evolution.
- Evolve on a domain $[-1, 1]$, with 100 cells, i.e. $\Delta x = 1/50$ up to time $T = 0.1$.

In order to compare the methods, we calculate the error that occurs in each of the variables ρ, v_x , and p , and take two norms of this, the \mathcal{L}_{∞} norm and the \mathcal{L}_1 norm. The effect of the corrections is, in most cases, sufficiently small for these tests that whether the correction appears to be beneficial depends on the variable and norm used.

As an example, we show in Table 5.4 the sequence of changes made between the greatest and least errors in the maximum error in ρ . As we can see, the improvement overall is only about 5%, and we have chosen to show this chain for the variable and norm showing the greatest improvement.

Looking at the combinations that give the smallest error for each of the variables and error norms, we see that we should

1. Limit on the conserved variables
2. Use the more careful metric averaging
3. Not use the flux/source balancing term
4. Possibly use the Simpson correction, which only contributes to the least error for the velocity
5. Use combined source splitting (only fails to contribute to the least error when taking the \mathcal{L}_{∞} norm of the pressure)

6. Use the GFORCE flux (only fails to contribute for \mathcal{L}_∞ norm of velocity)

We have performed more general checks where we see how often setting a specific option to one state reduces the error as compared to the identical setup with the option set to its other state. These are not completely conclusive either, except that using the combined sources almost always helps and that using the metric averaging correction usually helps. The Simpson correction helps to reduce the \mathcal{L}_∞ norm, not the \mathcal{L}_1 norm.

We emphasise, however, that the previous tests were carried out in fairly simple situations, and may not reflect the performance of the numerical schemes proposed in far more testing regimes. Also, most of these schemes require extra computation, and so it may be more beneficial to reduce the error by increasing the resolution, rather than introducing new schemes. This applies in particular to our method of combining the sources with the fluxes.

5.2 Two-dimensional SRHD Tests

In this section we run some tests of the fluid evolution that test the two-dimensional aspects of the algorithm. We first run a two-dimensional Riemann problem, and compare our results to some already published. We then proceed to check the accuracy and robustness of our algorithm with regard to overlapping grids and the interpolation required. We then test our corrections as in §5.1.6 and find some improvements over the one-dimensional results.

5.2.1 Two-dimensional Quadrants test

In this section we show the results for the two-dimensional Riemann problem given in §3.6.2. Figure 5.8 shows the results for two different schemes, FORCE and GFORCE, both using the van Leer limiter. The resolution used is 400 cells in each direction, using a CFL of 0.95. Although there is no exact solution for this initial data, the results compare favourably with those of [117]. We note that the long density peak in the upper-right of the figure is more pronounced in our scheme than in [117], but that the feature just below and to the left of the centre is somewhat less well-defined in our plot than in [117]. It is in fact that case that, within this region, the density drops to a very small value indeed, of the order of 10^{-6} . It is also the case, although not entirely clear on the plots shown, that the discontinuities are slightly narrower for the GFORCE scheme than for the FORCE scheme.

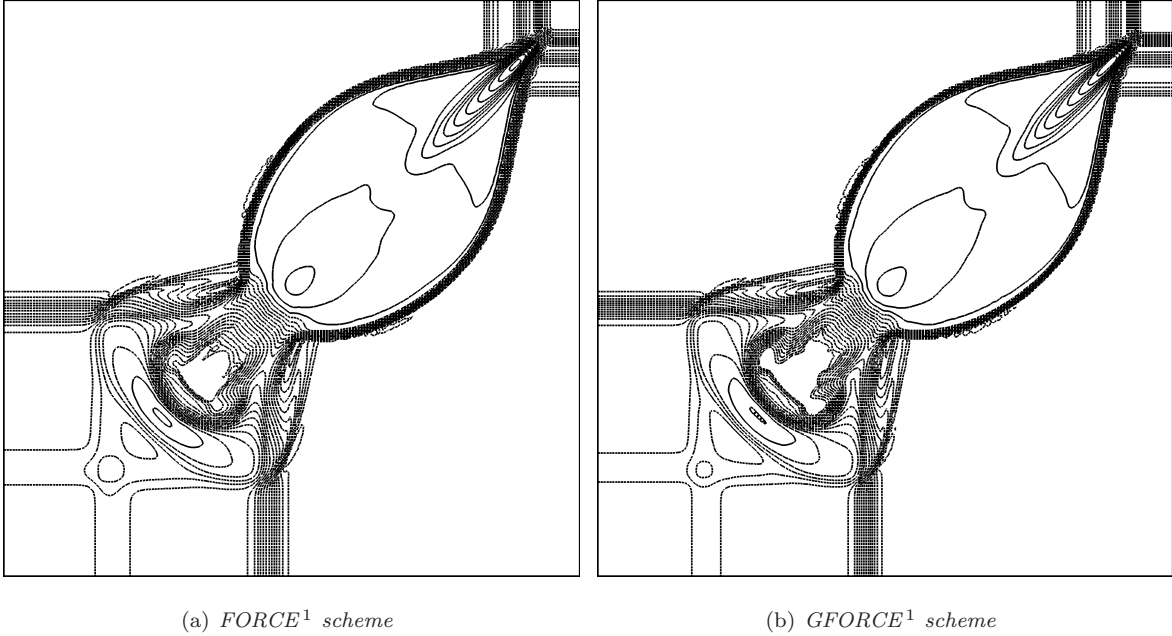


Figure 5.8: The quadrants problem evolved at a resolution of 400×400 , with a CFL of 0.95, using the $FORCE^1$ and $GFORCE^1$ schemes with the van Leer slope-limiter. We plot the log of the density ρ , using 30 equally spaced contours from $\ln 0.1 = -2.303$ to $\ln 3.927 = 1.368$

5.2.2 Adaptive Mesh Refinement

Although we are essentially evolving the Riemann problem for more than one time-step in order to calculate the multi-staged flux, this is in no way related to the time-stepping of each grid, and so does not affect the standard approach to AMR.

The efficiency of the AMR approach is evaluated by comparing the CPU times between constant cell-width and adapted results (of the same effective resolution). The timings are shown in Table 5.5, and some results are shown in Figure 5.9, where the boxes show the regions where the mesh has been refined. The benefits of AMR are well known and the timings shown illustrate these since as the resolution of the coarsest grid decreases, the run-time reduces, even though the effective resolution is the same.

The timings also demonstrate the effect of multi-staging. Each additional stage adds approximately the same amount of time to the simulation as a result of having to calculate the fluxes more often. We note that the density peak values are roughly independent of the number of refinement levels used, showing that AMR has been correctly implemented.

Figure 5.9 shows the same plots as Figure 5.8, except that we now show evolutions performed at a resolution of 100 cells in each direction with two extra levels of refinement factor two, and we can see that the finest grid level contains the regions where we would expect most resolution to be needed in order to obtain an accurate solution. The main differences between the uni-grid and AMR simulations

Resolution	400×400		$200 \times 200 (\times 2)$		$100 \times 100 (\times 2 \times 2)$	
Scheme	Max. ρ	Time (s)	Max. ρ	Time (s)	Max. ρ	Time (s)
FORCE ¹	3.685	356.6	3.746	258.1	3.663	244.0
FORCE ²	4.437	463.1	4.505	302.9	4.743	289.5
FORCE ³	4.967	571.1	5.060	359.5	5.519	328.4
GFORCE ¹	3.928	442.5	3.945	294.8	3.999	271.2
GFORCE ²	4.351	594.2	4.372	374.1	4.484	357.8
GFORCE ³	4.639	742.2	4.654	454.6	4.854	435.8
HLL ¹	4.247	564.2	4.283	338.6	4.387	326.3
HLL ²	4.470	875.8	4.492	505.1	4.693	465.1
HLL ³	4.625	1188.5	4.669	656.4	4.883	606.3

Table 5.5: *Comparison of run-times and maximum densities attained when evolving the quadrants problem in AMR, all for an effective resolution of 400×400 . The notation $n \times n (\times r)$ refers to a base grid of $n \times n$ with one level of refinement of factor r . All the schemes stated are embedded in the SLIC scheme, using the van Leer slope limiter.*

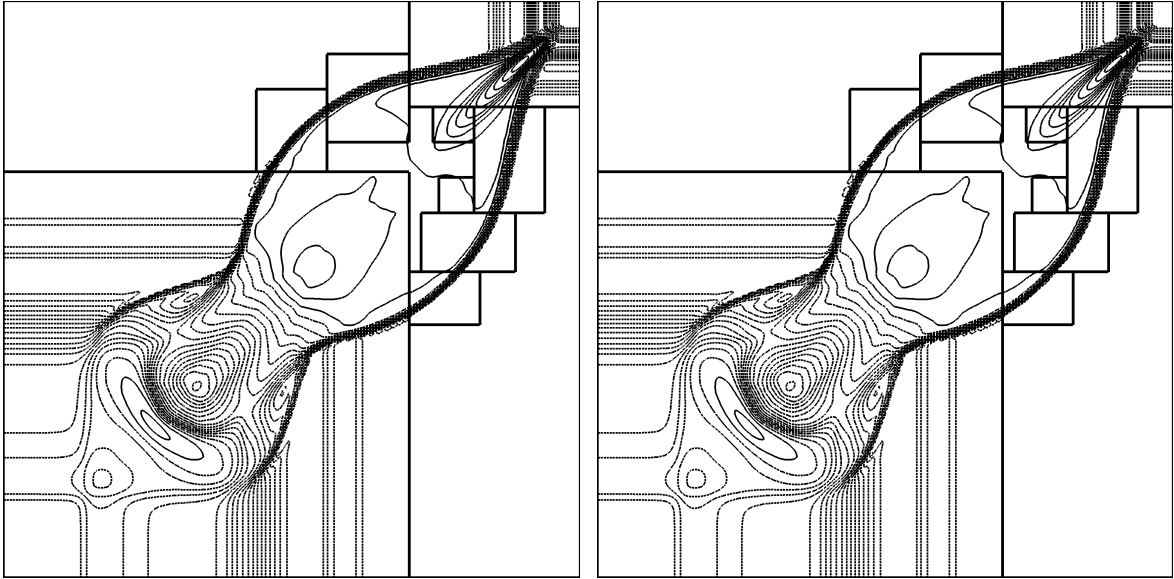
are the smoothing of the region below and to the left of centre and the fairly significant diffusion of the shocks leaving to the grid at the left and the bottom.

In Table 5.6 we quantify the convergence properties of AMR for various numerical schemes. We see that we do not achieve second order convergence, but this would not be expected anyway due to the presence of discontinuities in the data. The convergence rates, however, are somewhat in excess of first-order, and so we conclude that our approach has been designed and implemented correctly to retain second-order convergence.

However, we also conclude that the algorithm may be ill-equipped to deal with shocks in some cases, as we have seen greater diffusion than for a single grid. This may require some further investigation into the refinement criteria, particularly at the start of the simulation, where the shocks were not sufficiently well refined.

Scheme	Resolution	Est. $\ \rho\ _1$ Error	Time (s)	Order
GFORCE ¹	100	6.83×10^{-2}	11.1	1.37
	100×2	2.64×10^{-2}	75.6	
	$100 \times 2 \times 2$	1.02×10^{-2}	416.7	
GFORCE ²	100	6.32×10^{-2}	13.7	1.41
	100×2	2.38×10^{-2}	88.8	
	$100 \times 2 \times 2$	8.97×10^{-3}	498.2	
GFORCE ³	100	9.60×10^{-2}	16.3	0.78
	100×2	5.60×10^{-2}	107.4	
	$100 \times 2 \times 2$	3.26×10^{-2}	592.0	
HLL ¹	100	6.05×10^{-2}	10.6	1.36
	100×2	2.35×10^{-2}	74.1	
	$100 \times 2 \times 2$	9.17×10^{-3}	414.0	
HLL ²	100	6.17×10^{-2}	13.5	1.23
	100×2	2.64×10^{-2}	96.3	
	$100 \times 2 \times 2$	1.13×10^{-2}	552.6	
HLL ³	100	5.95×10^{-2}	16.4	1.20
	100×2	2.59×10^{-2}	118.7	
	$100 \times 2 \times 2$	1.12×10^{-2}	697.8	

Table 5.6: *Orders of convergence for the Quadrants test problem for various schemes using a base grid of resolution 100, and varying numbers of levels of refinement. All calculations were done using CFL 0.95 and the van Leer slope limiter. The errors were estimated using Richardson extrapolation, using a tool available in Overture.*



(a) Evolved using $FORCE^1$

(b) Evolved using $GFORCE^1$

Figure 5.9: The quadrants problem evolved at a resolution of $100(\times 2 \times 2)$, with a CFL of 0.95, using the $FORCE^1$ and $GFORCE^1$ schemes with the van Leer slope-limiter. We plot the log of the density ρ , using 30 equally spaced contours from $\ln 0.1 = -2.303$ to $\ln 3.927 = 1.368$. The boxes show the edges of the grids with the finest resolution. The refinement to the resolution of 200 cells covers the whole domain, which explains the apparent presence of only one refinement level.

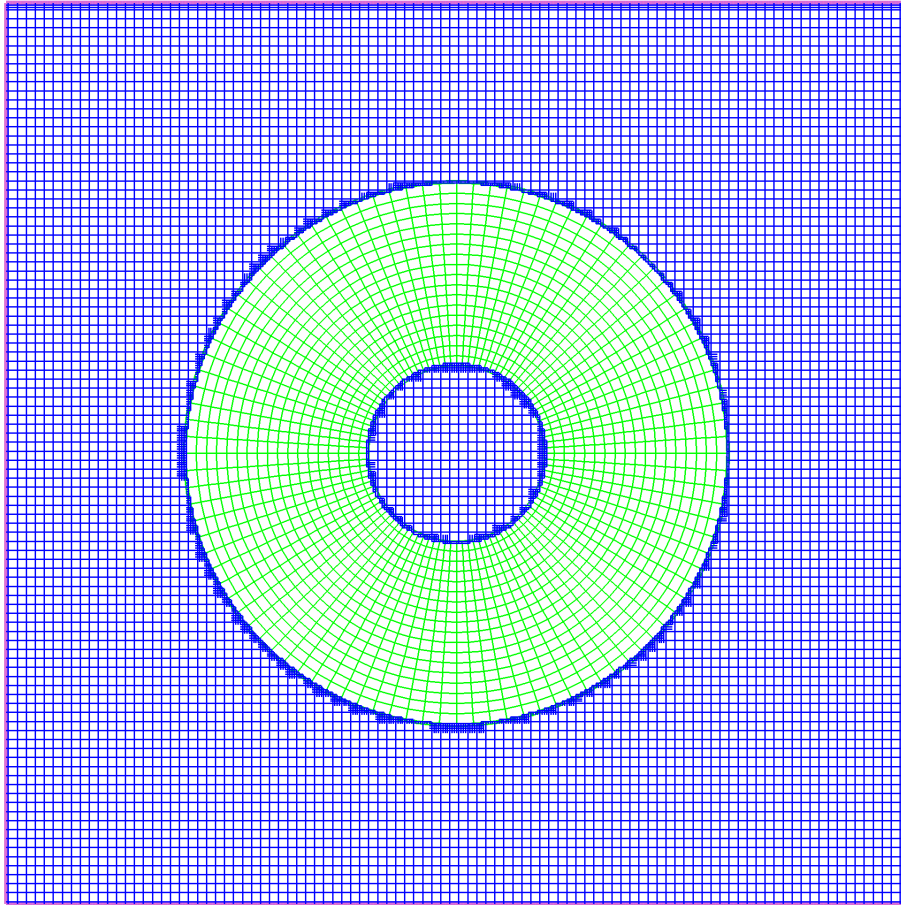


Figure 5.10: *Grid for testing curvi-linear and overlapping routines. An annulus is superimposed on a Cartesian grid and used to replace part of that grid. The Cartesian grid has its vertices at $(\pm 0.5, \pm 0.5)$, a resolution of 400×400 , and the annulus resolution matches that of the Cartesian grid at the inner circumference. We only show every fourth gridline in this plot for clarity.*

5.2.3 Shock propagation across an overlapping grid

It is important to test the accuracy of evolution across a set of curvi-linear, overlapping grids in order to validate these approaches. To this end, we evolve a Riemann problem with a known solution across the two-dimensional domain shown in Figure 5.10, where the Cartesian grid has resolution 400×400 , and the annulus matches its resolution to that of the Cartesian grid at its *inner* circumference. Note that the entire space is covered, so that we are *not* testing any outgoing boundary conditions on the annulus. Also, it is necessary for testing purposes that we use a two dimensional grid since, in one dimension, the Jacobian and the inverse coordinate derivative multiply to give one, thus simplifying the calculation somewhat. We use flat Minkowski space-time for the evolution, and as initial conditions we use the

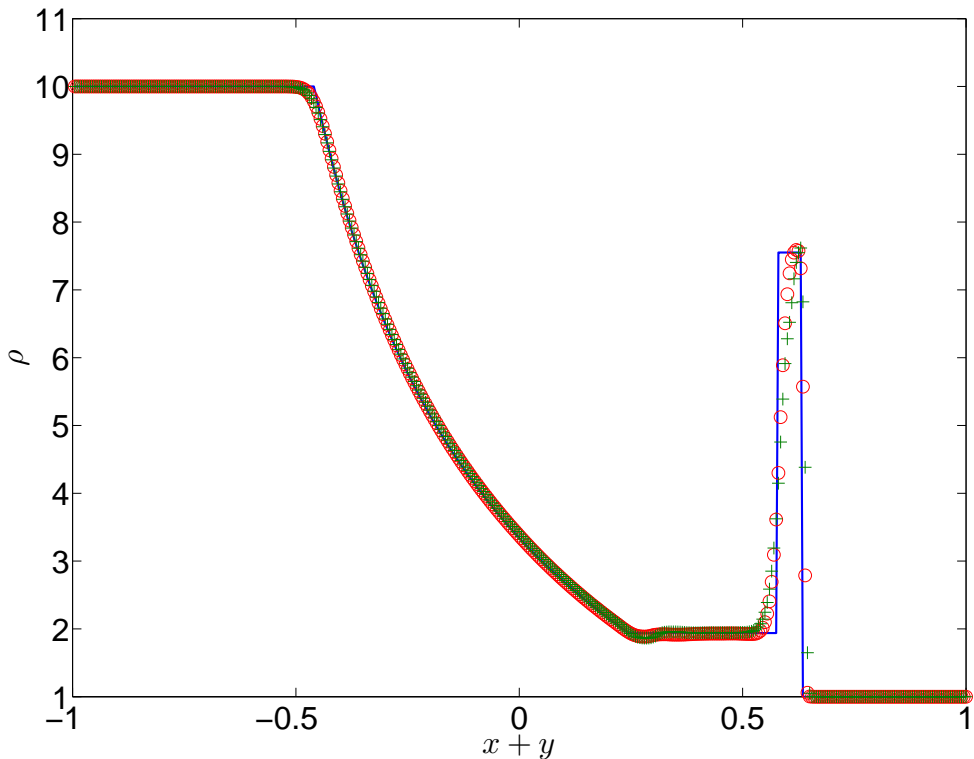


Figure 5.11: *Cross-section of the evolution of the mildly relativistic shock-wave across the grid shown in Figure 5.10. The run was performed using the GFORCE¹ scheme at a CFL of 0.95, and using the van Leer slope-limiter. The green + signs are the numerical solution on the overlapping grid, the red circles are from an equivalent simulation on a single Cartesian grid, and the blue line is the exact solution for the problem. The two numerical solutions have been interpolated from along the line $x = y$.*

Riemann problem

$$(\rho, v_x, v_y, p) = \begin{cases} (10, 0, 0, 13.3) & \text{if } x + y < 0, \\ (1, 0, 0, 10^{-6}) & \text{if } x + y > 0, \end{cases} \quad (5.6)$$

so that the shock is propagating diagonally across the grid.

As this is only a 1D test evolved on a 2D grid, the exact solution can be found as in §3.6.1. We perform the evolution using the grid described in Figure 5.10, and sample the solution at 400 points (corresponding to the Cartesian grid's resolution) along the line $x = y$, parallel to the shock direction, at a time of $t = 0.4$. Figure 5.11 shows the results from using the GFORCE¹ flux solver, with a CFL of 0.95 and the van Leer limiter. We can see that the solutions on the plain Cartesian grid and on the overlapping grid are very close together indeed. The solution on the overlapping grid has a small oscillation at the base of the rarefaction wave, but is otherwise clean. In fact, the solution from the Cartesian grid has a

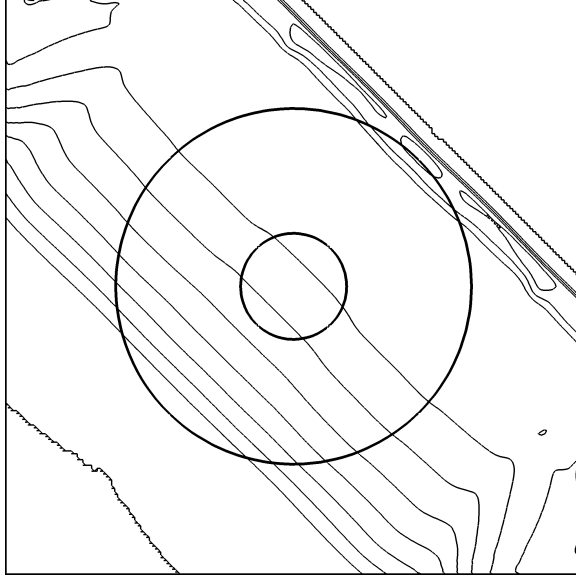


Figure 5.12: *Contour plot of the evolution across an overlapping grid as described in Figure 5.11. We show contours of density at $\rho = 1, 2, \dots, 10$ and the location of the annular grid.*

similar, but smaller oscillation.

We also show in Figure 5.12 the contour lines resulting from the evolution on the overlapping grid. The location of the annulus grid is also shown overlaid. Ignoring the effect of using simple transmissive outer boundary conditions for the Cartesian grid, the contour lines are fairly straight. In particular, there are no deviations noticeable at the edges of the annular grid. It therefore appears that the algorithm is capable of evolving a shock cleanly on an overlapping grid.

5.2.4 Front-facing step

In order to test the overlapping grid approach in the case of a non-trivial boundary, we apply our algorithm to the problem of a fluid shock hitting a solid front-facing step. This test has been used by Lucas-Serrano *et al.* [71]. Due to constraints on the way grid boundaries can interact in Overture, we cannot use a sharp corner, but we replace it with a curved corner to the step, provided by a smoothed polygon. The grid is shown in Figure 5.13. This accounts for the difference in solution between our results and those of [71].

The fluid is defined to have an initial density of 1.4, with an adiabatic index of 1.4. The fluid comes in from the left-hand boundary at velocity $v_x = 0.8$, and at Mach 3. Initially, the fluid is given constant density and velocity, and it is then evolved to time $t = 10$. The resulting flow pattern is shown in Figure 5.14, where we plot iso-contours of the log of the density.

In performing this evolution, we have also used AMR, refining where $\nabla \cdot \mathbf{S} < -0.1$, and in Figure 5.15

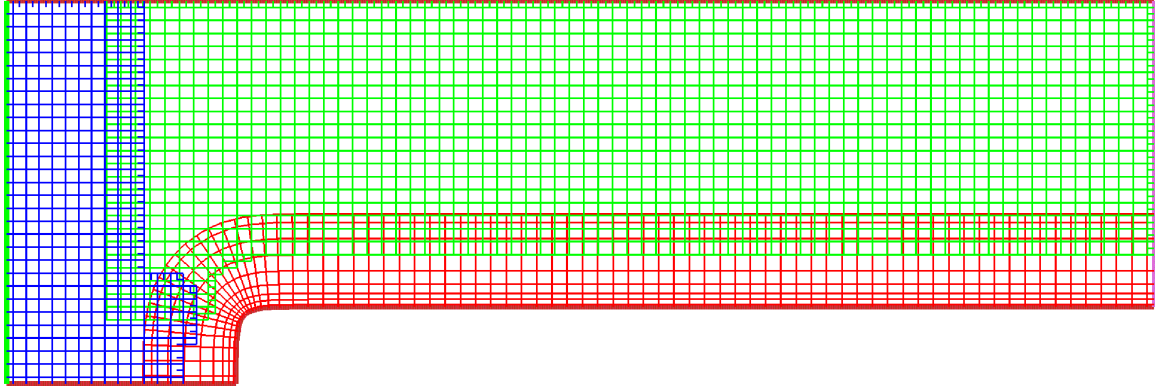


Figure 5.13: *Grid used for the front-facing step problem. The grid is made up of three separate grids: one for the incoming channel, one for the rest of the channel, and one for the step itself. The last has stretched grid cells that become narrower towards the edge of the step. For clarity we only show every other grid line.*

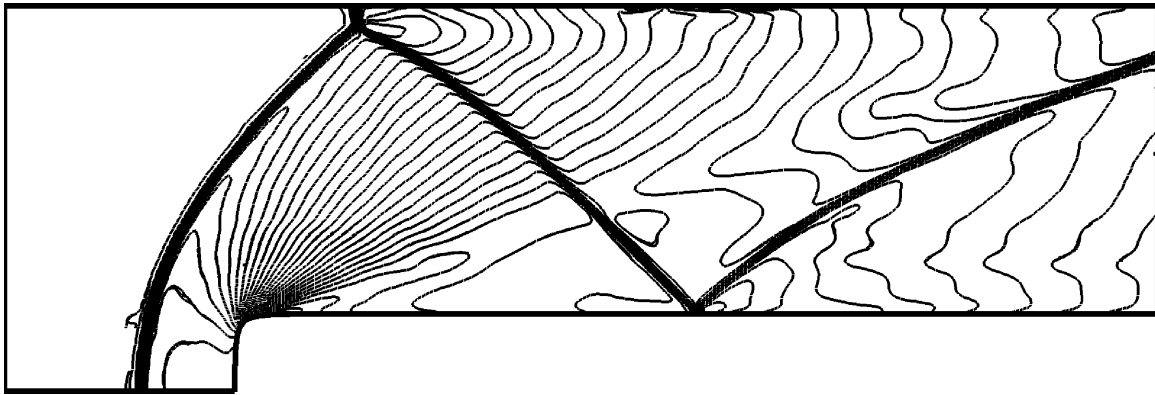


Figure 5.14: *Solution at $t = 10$ for a shock hitting a front-facing step. The fluid has adiabatic index $\Gamma = 1.4$, velocity $v = 0.8$ and is travelling at Mach 3. The scheme used was GFORCE¹ with the van Leer slope-limiter, and with CFL 0.95. We plot 30 contours of the log of the density.*

we show the refined grid structure at the end of the simulation. Most of the grid has been refined, except for the region at the very left of the domain, just before the step.

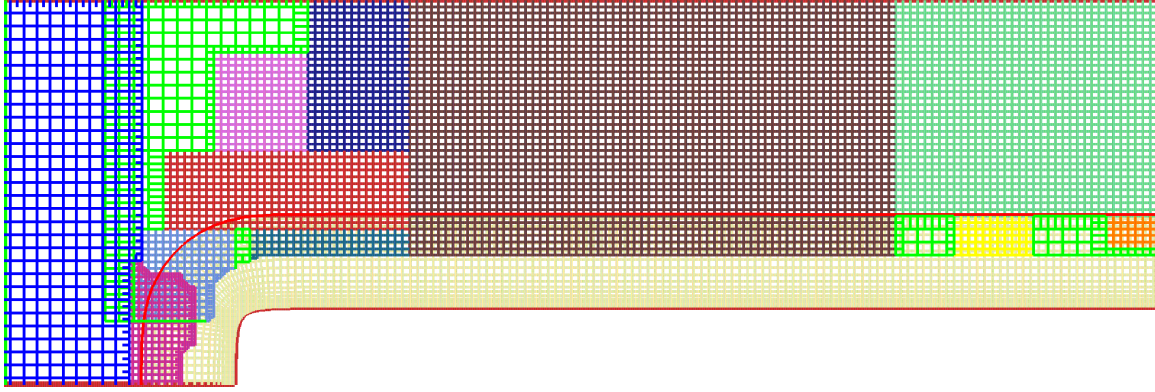


Figure 5.15: We show adaptively refined grid at the end of the evolution shown in Figure 5.14. Again we only show every other grid-line. Each grid is given a different colour.

5.2.5 Reflective cylinder

Since we intend to apply this algorithm to problems in General Relativity, we demonstrate a problem that is somewhat similar in its grid setup and initial conditions to that of a fluid accreting onto a black hole. In this problem, we immerse a solid cylinder in a fluid with adiabatic index $\Gamma = 1.4$, travelling with constant velocity $0.8c$ at Mach 3, and allow it to evolve to steady-state. This general setup has been much studied in the Eulerian case, and we expect to see qualitatively similar results here. Our grid is constructed from a Cartesian channel, containing an annulus, the inner circumference of which is a reflective boundary. The channel's walls are given transmissive boundary conditions, with the exception of the left boundary, which is an inflow boundary. The initial grid is shown in Figure 5.16.

The state of the simulation at time $t = 10$ is shown in Figure 5.17. In order to demonstrate the grid setup possible with AMR, we show a split diagram where the upper half shows the numerical result and the lower half shows the adaptive grid structure. Here our criterion for refining the grid is $\nabla \cdot \mathbf{S} < -0.05$.

We see that the result is qualitatively similar to the well-known case of a cylinder embedded in a non-relativistic flow.

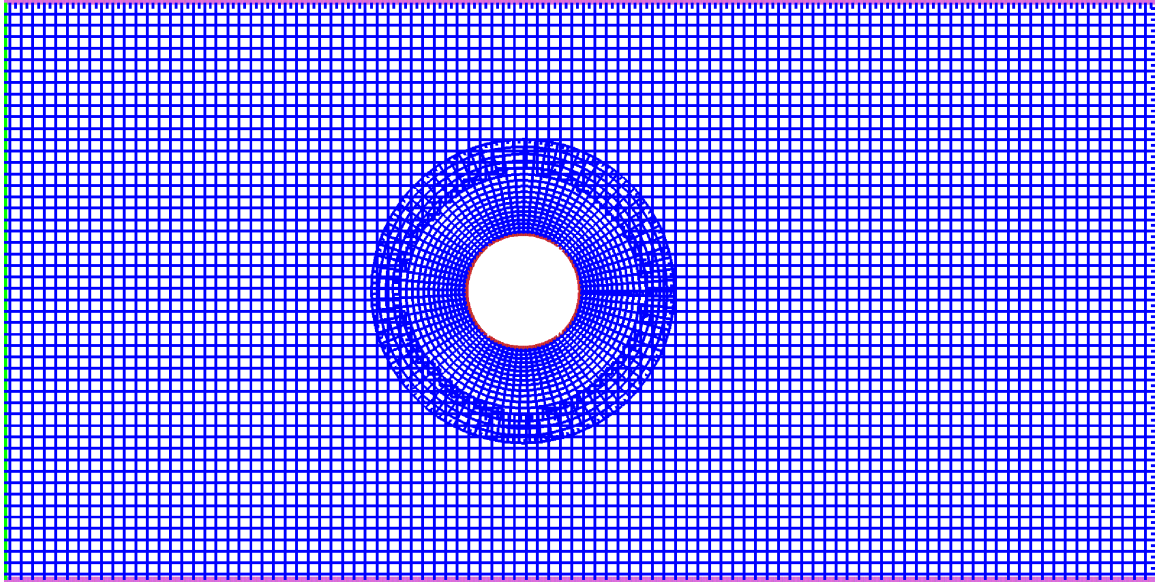


Figure 5.16: *Grid used for the problem of a shock hitting a reflective cylinder. The main channel has length 10 and width 5, composed of 100×200 cells. The annulus has inner radius 0.5 and outer radius 1.25, with 168 cells in the angular direction, and 32 cells in the radial direction, where an exponential stretching function has been applied in the radial direction to restrict the radial size of cells nearer the inner boundary. For clarity, we only show every other grid line.*

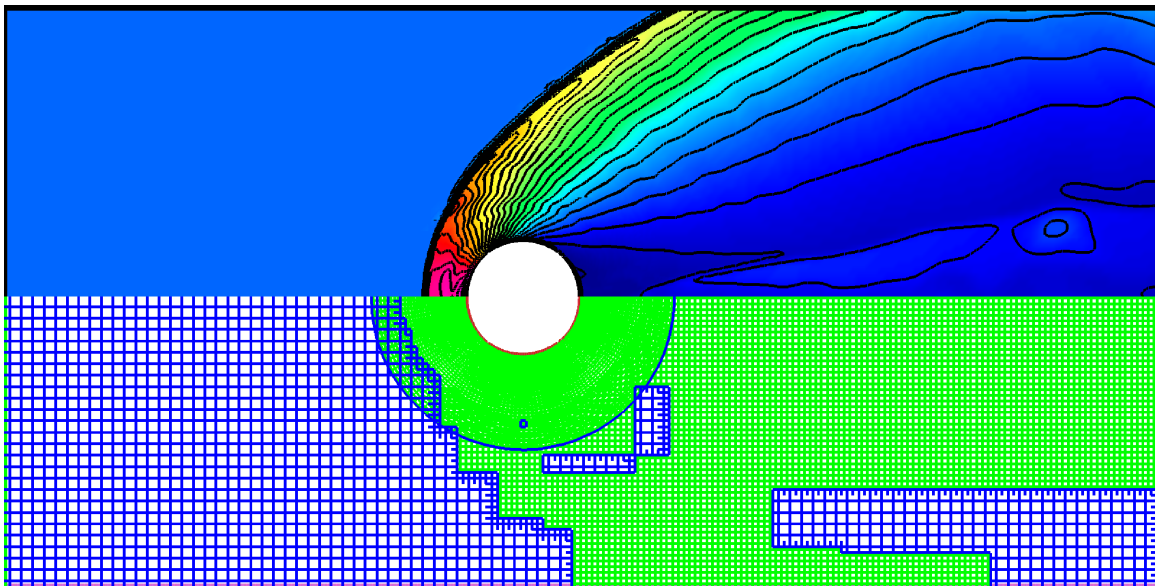
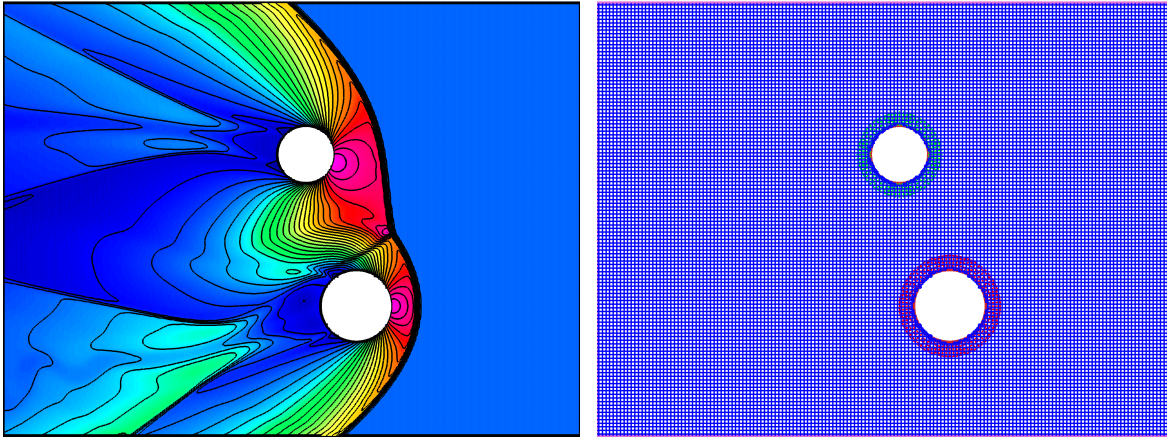


Figure 5.17: *Contours of ρ at time $t = 10$ of a reflective cylinder initially immersed in a constant velocity flow of Mach 3, velocity $0.8c$, adiabatic index $\Gamma = 1.4$, and density 1.4. The grid used was that of Figure 5.16, with one level of refinement factor 2. The scheme used was $GFORCE^1$, with the van Leer slope-limiter, at a CFL of 0.95. There are 30 contours, equally spaced between 0 and 10.*



(a) Solution for two reflective cylinders

(b) Grid used for multi-cylinder flow, showing only every other grid line for clarity.

Figure 5.18: We plot 30 contours of ρ from 0 to 10 for two solid cylinders initially immersed in a fluid of constant state, Mach 3 and velocity 0.8, with density 1.4 and adiabatic index 1.4. The solution is shown at time 10. The GFORCE¹ scheme was used, with the van Leer slope-limiter, and a CFL of 0.95.

5.2.6 Multiple reflective cylinders

In Figure 5.18(a) we show an extension of this problem, to the case where there is more than one reflective cylinder in the domain. The same constant state initial data was used, this time incoming from the right. This simulation required no more extra setup over and above the single cylinder than generating a new grid using ogen, the interactive overlapping grid generator that comes with Overture.

Although there are no prior solutions with which to compare it, the resulting flow appears qualitatively correct, and no numerical problems are visible in the vicinity of the overlapping regions.

5.2.7 Forcing an exact solution

In §5.1.6, we displayed the results for applying some corrections to our evolution of a one-dimensional solution that we forced to be exact. Here we apply a very similar test to a two-dimensional grid and display the equivalent results.

The grid we use for this test is the one in Figure 5.10, with periodic boundary conditions, and 100 cells in each direction on the Cartesian grid. The problem setup is otherwise identical to the 1D case. Although the problem itself is entirely 1D, the 2D nature of the curvi-linear grid means that the y -velocity will not remain zero as it should. We therefore want to reduce the error in the y -velocity as far as possible.

1	2	3	4	5	6	$\ v_y\ _\infty$
✓	✓	✓	×	×	GFORCE	1.3665×10^{-2}
✓	✓	✓	✓	×	GFORCE	7.9786×10^{-3}
✓	×	✓	✓	×	GFORCE	1.2234×10^{-3}
✓	×	×	✓	×	GFORCE	5.0693×10^{-4}

Table 5.7: Here we show the \mathcal{L}_∞ norm of v_y after the evolution of the two-dimensional fluid test on a curved metric. The key to the columns is as follows: 1) Primitive variable limiting (§4.7) 2) Metric averaging correction (§4.7) 3) Simpson’s Rule correction (§4.4.2) 4) Free-streaming correction (§4.6.3) 5) Flux-source combination (§4.4.1) 6) GFORCE or HLL scheme (§4.2.3 and §4.2.4)

Some of the results for the \mathcal{L}_∞ norm of the error in v_y are shown in Table 5.7. From this we see that the free-streaming correction makes a large improvement to the error. Further, removing the metric averaging correction and the Simpson’s rule correction actually reduces the error. This may be due to the fact that both of these cause some mixing between the different Cartesian components when performed on a curvi-linear grid, thereby transferring some errors from the x -component to the y -component. For example, the metric averaging correction does tend to reduce the \mathcal{L}_∞ norm of the error in v_x at the expense of that in v_y .

Overall, these results are not conclusive, and more investigation would need to be done to find ways of making these simple schemes more accurate.

5.3 One-dimensional Metric Tests

In this section we give the results of some tests of the metric evolving aspects of our algorithm. Firstly we apply our numerical methods to a specialised system of Einstein’s equations, in order to test the suitability of our methods for evolving large systems. We then proceed to apply our algorithm to some standard test-cases for metric evolution, and demonstrate similar accuracy to other, more elaborate, methods.

5.3.1 Medium System Tests

In this section we describe some results we obtained using the Medium system defined in §3.4. The main purpose of this section is to demonstrate the accuracy of the GFORCE^k schemes when applied to a set of equations based on a specialised form of Einstein’s equations. In particular, the Medium system includes second derivatives of the metric among its variables, and we therefore expect to see, and in fact do see, discontinuities in these variables, as permitted.

We note that Barnes *et al.* [16] used WENO decomposition and the Marquina solver for their evolutions. Although the methods we use have a lower formal order of accuracy, we shall see that the results show at least similar, and sometimes better, visual accuracy, particularly around discontinuities.

Decreasing Constraint Violations

The original paper of Barnes *et al.* [16] demonstrated sizable constraint violations near discontinuities in either the fluid or second derivatives of the metric. We have managed to reduce these errors somewhat, partly by our use of a simpler numerical method, and partly by a small adjustment to the boundary conditions, which we now outline.

These tests were carried out using a simple one-dimensional code, not using Overture. However, the numerical methods used were otherwise identical. Initial tests of the system resulted in quite large constraint violations from the boundary. After some investigation, we discovered that this was due to a slight inconsistency between source and flux evolutions, which was being propagated into the domain. Our solution was to increase the number of ghost cells used at the boundary from two (as one would expect to be used for the SLIC scheme) to three, and to solve the constraint equations (3.35) exactly for the variables $b_{,xx}$ and $b_{,xt}$ on the ghost cell neighbouring the computational domain, at either end. This reduced the constraint violation substantially. We note, however, that this is a somewhat specialised solution to the problem as in this case the constraint conditions are algebraic in nature, and easily solved. It is not easily applicable to multi-dimensional evolutions, where a more expensive elliptic solver would be needed to perform the same adjustment.

Numerical Results

In a similar manner to that used for the 1D SRHD case, we reduce the time-step by a factor of four for the first four time-steps. Our domain was $x \in [-1, 1]$, and we used 800 cells to cover it. Following Barnes *et al.* [16], we restrict our plots to the domain $[-0.6, 0.6]$.

The results for this implementation of a simplified version of our algorithm as applied to the tests described in §3.4 can be seen in Figures 5.19 – 5.20. We plot the same variables as in [16]. Our axes are also identical, except for those for the constraints, and those for $b_{,tx}$ and $b_{,xx}$ for the second test. We can see that our results are mostly visually identical to those of Barnes *et al.* One noticeable difference is that for $a_{,tx}$ and $a_{,xx}$ in Test 1, where the disturbance at $x \approx 0.5$ is squarer than theirs. Since their zoomed version of that plot shows that part of the signal has propagated beyond $x = 0.5$, i.e. faster than the speed of light, we assume that our results are the more correct, not having smeared the discontinuity as much.

However, the main point of interest is that our constraint violations are somewhat smaller than those

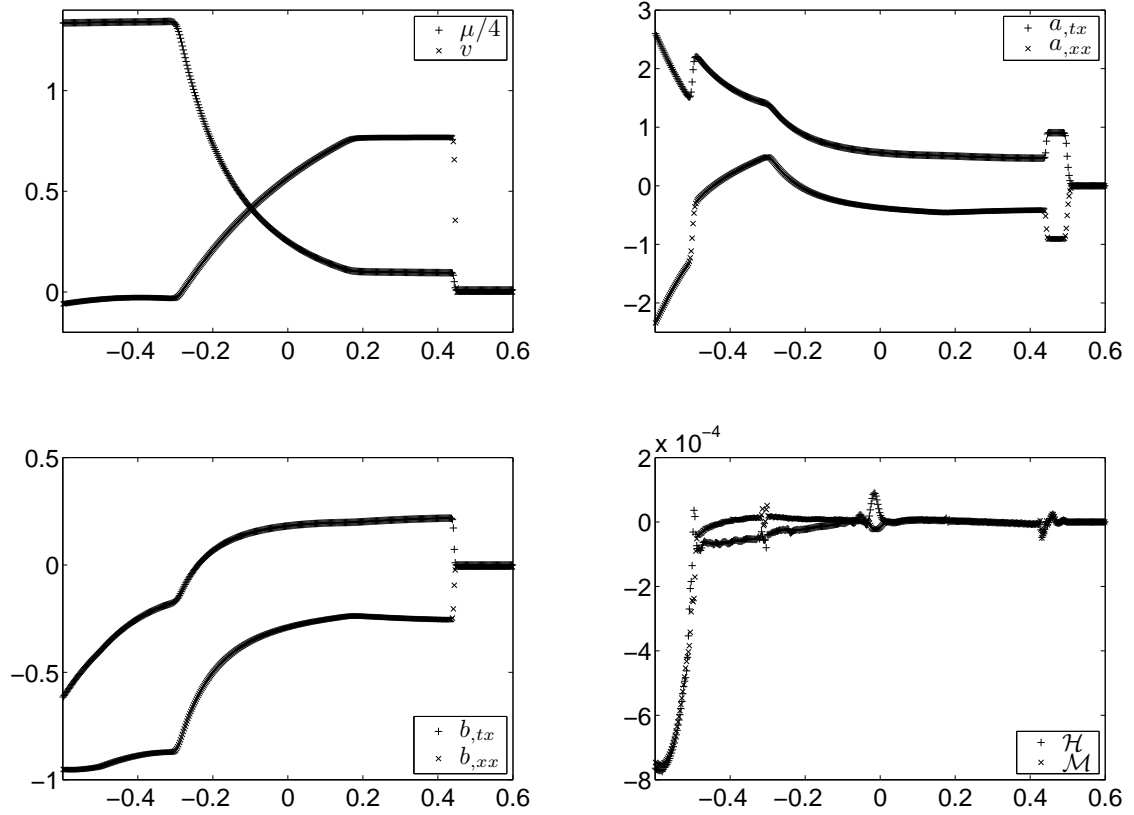


Figure 5.19: Results for the Medium system evolving the first test case, using a CFL of 0.95, the GFORCE¹ scheme and the super-bee limiter.

in [16]. We attribute this partially to our improved boundary conditions, but mostly to the different numerical scheme. We were able to use a CFL number of 0.95, whereas [16] only used 0.5, which would lead to more smeared discontinuities, and hence larger errors nearby.

We therefore conclude that SLIC based on GFORCE is suitable for the evolution of complex systems of equations, including those that are simplifications of Einstein's equations.

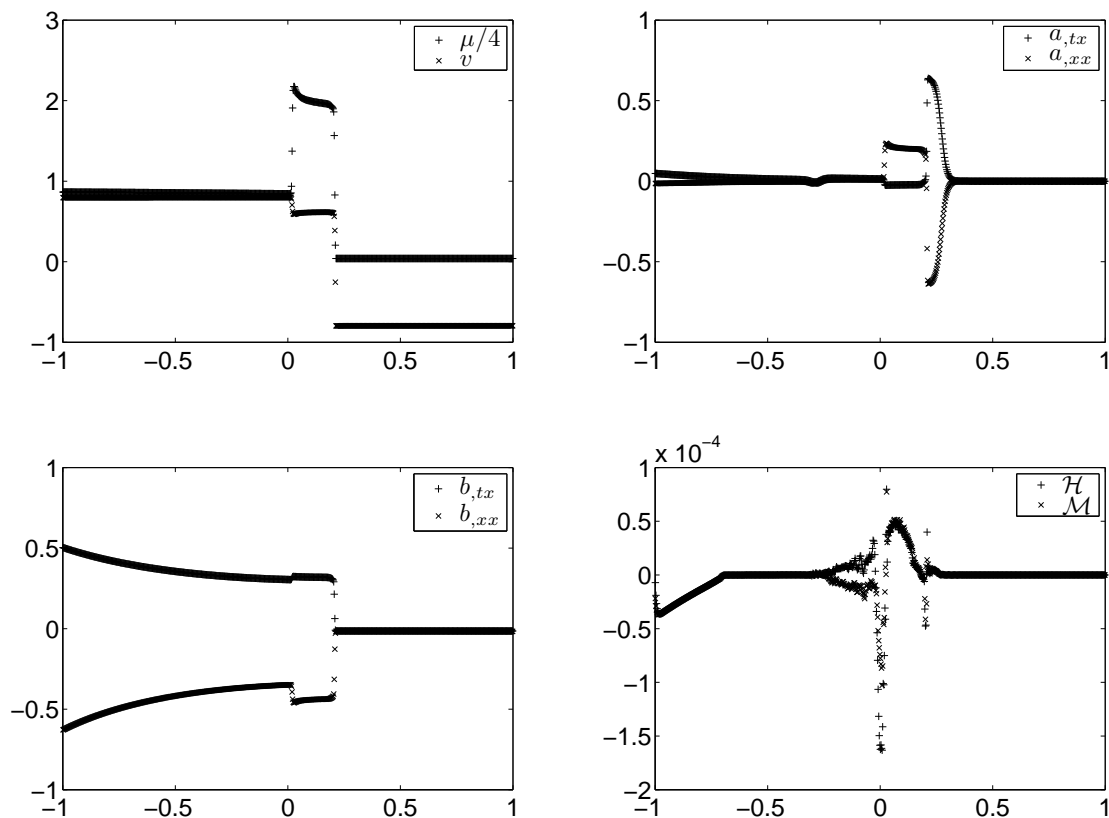


Figure 5.20: *Medium system evolving the second test case with CFL 0.95, and using GFORCE³ with the van Leer slope-limiter*

5.3.2 Apples with Apples tests

We can use the Apples with Apples tests as outlined in §3.8 to compare the various numerical schemes. We can also check that the convergence rates are as expected.

Robustness test

In Figure 5.21 we show the results of evolving the robustness test described in §3.8.1. We use two approaches: the finite-volume scheme GFORCE with the van Leer limiter, and the finite-difference scheme ICN. The finite-difference scheme damps the constraints to a greater extent than the finite-volume scheme. Both schemes exhibit some oscillations in the constraints, as a result of the randomness of the initial conditions. However, neither demonstrates any essential difficulty in preventing small constraint violations from increasing in size.

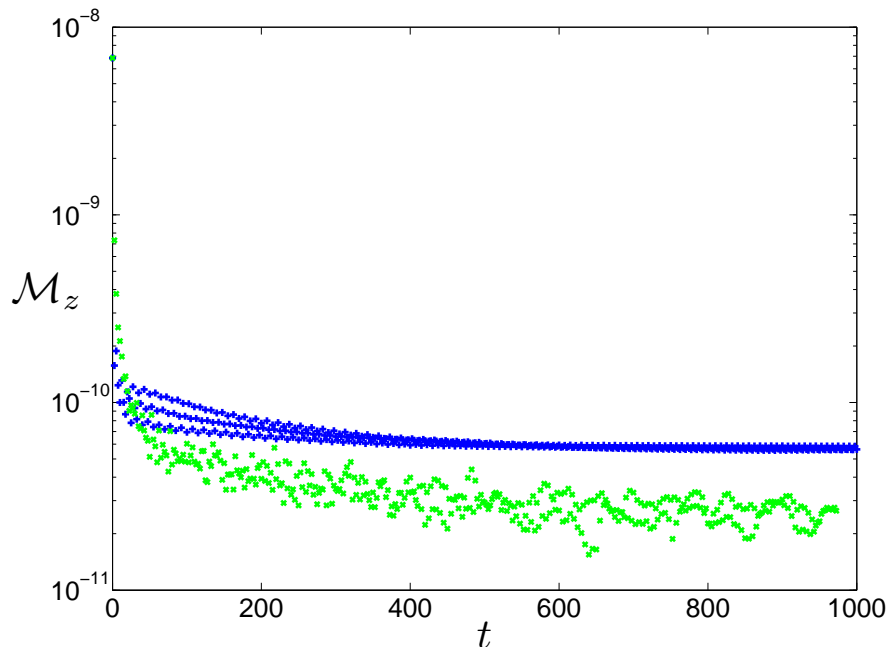


Figure 5.21: *Evolution of constraint \mathcal{M}_z for the Apples with Apples robustness test. The blue +s represent the finite-volume scheme GFORCE, with the van Leer slope limiter, and the green \times s represent the finite-difference scheme ICN. Both simulations use the RK2 scheme for the source terms, and the resolution $\rho = 2$. The ICN simulation did continue to time $t = 1000$, but the data for the last few time-steps were lost for technical reasons.*

Linearized Wave test

We evaluate the linearized wave test as described in §3.8.2. Figure 5.22 shows the difference that the CFL parameter can make, comparing the original test-case with one where $dt = c_{\text{CFL}} dx$ and $c_{\text{CFL}} = 0.95$. The numerical parameters are otherwise the same. The main errors for the GFORCE scheme are that there is a phase error in the solution, which is more pronounced for a lower CFL number, substantial diffusion, again more pronounced for a lower CFL number, and that the solution is not vertically centred on zero, which is most noticeable where the solution for CFL 0.95 goes below the exact solution at $x \approx 0.25$. An effect similar to the last error is seen in the results from the CCATIE_BSSN solver in [11].

The diffusive effect apparent with the lower CFL condition is more pronounced when using a more diffusive limiter. This comparison can be seen in Figure 5.23.

The ICN scheme is less diffusive than the GFORCE scheme, as we would expect. However, it gives a much larger phase shift, which is larger for the higher CFL number. It is not mentioned in [11] that a low CFL number is being used, and this would seem to unduly penalize methods which depend heavily on the CFL condition. However, it is highly likely that a propagating gravitational wave far from a highly

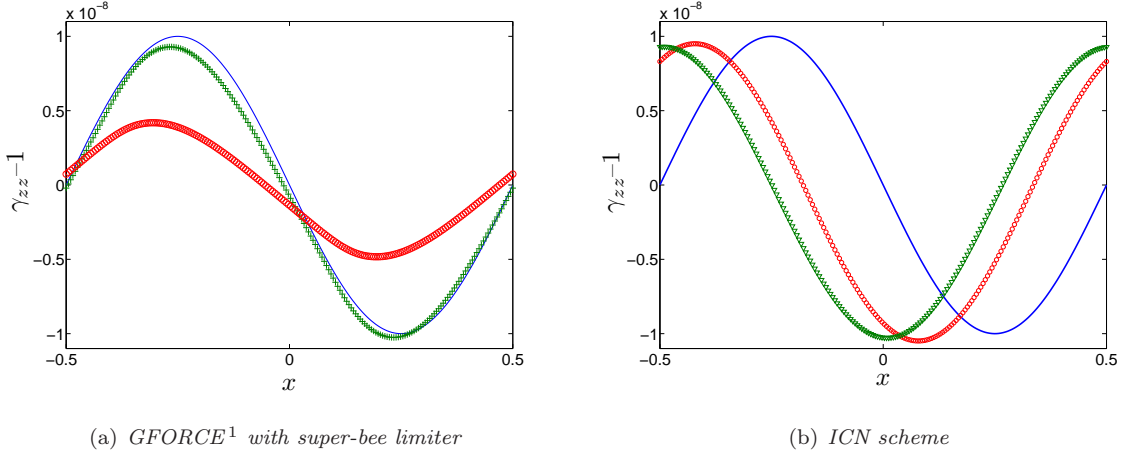


Figure 5.22: The effect of the CFL number on the Linearized Wave test described in §3.8.2. The green + signs use CFL 0.95 and the red circles used the CFL condition prescribed in the test. A resolution of $\rho = 4$, corresponding to 200 cells, was used, and RK2 was used to evolve the source terms. The solid line shows the exact solution.

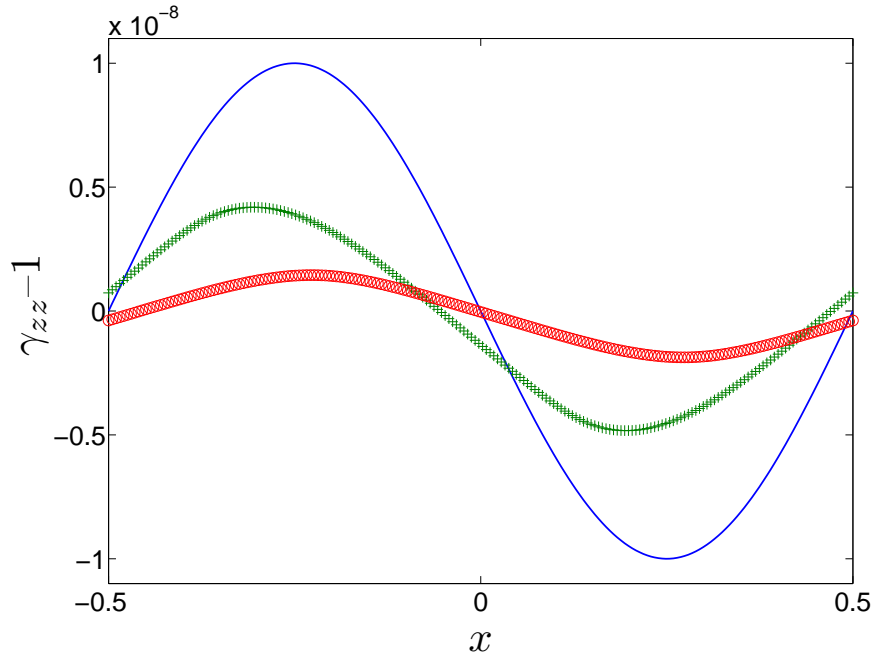


Figure 5.23: The effect of slope-limiter on the Linearized Wave test. The green + signs use the super-bee limiter, and the red circles use the van Leer limiter. The CFL condition is that prescribed by the test, and we use $\rho = 4$, the $GFORCE^1$ scheme, and RK2 for the source terms.

active source will effectively be subject to a low CFL number due to the high wave and gauge speeds found in the more active regions of the domain. In this regard, the test may be more realistic than might

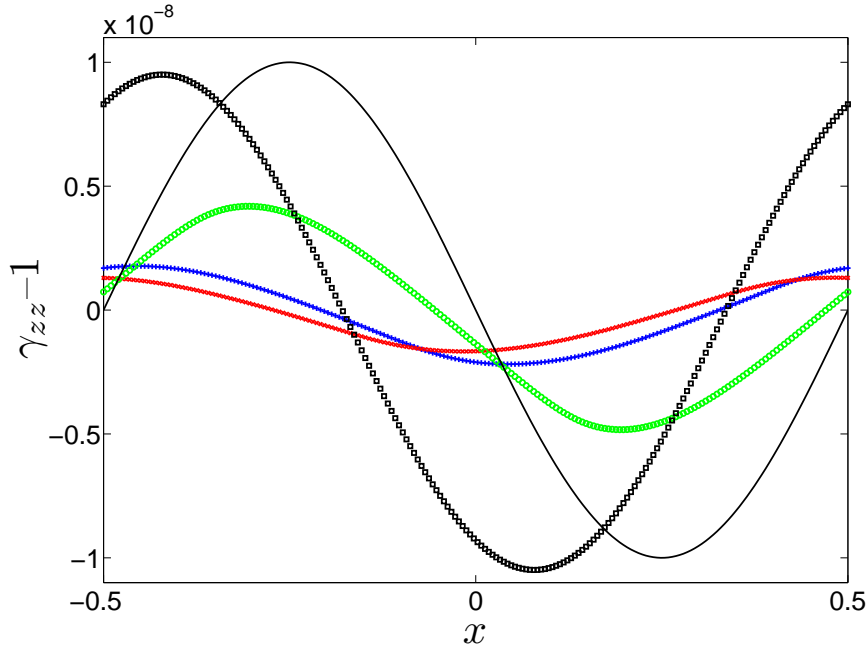


Figure 5.24: *The effect of different flux schemes on the Linearized Wave test. The blue + signs are for the FORCE¹ scheme, the red × signs are for the FORCE³ scheme, the green circles are for the GFORCE¹ scheme, and the black squares give the solution with ICN. The HLL scheme is not shown, but coincides with the GFORCE¹ scheme. The solid line gives the exact solution.*

at first be thought, and hence points up a potential weakness of the combined scheme and Bona-Massó formulation in producing accurate gravitational wave profiles.

Keeping with the time-stepping condition given in the test case, we show in Figure 5.24 a comparison of different schemes. All of these are done at a resolution $\rho = 4$, use the super-bee limiter, and the RK2 scheme for source terms. We see that once again the GFORCE¹ scheme out-performs the FORCE^k schemes, both in terms of amplitude and the phase shift error. The solution for the FORCE¹ scheme is better than that for the FORCE³ scheme. There is no visible improvement attained by multi-staging the GFORCE¹ scheme and, in fact, the results for the HLL scheme coincide with those for the GFORCE¹ scheme. The ICN scheme shows quite a large phase-shift, although it shows very little diffusion, maintaining the amplitude very well, and the wave is slightly off-centre in the vertical direction.

We also compare the results attained for different resolutions $\rho = 1, 2, 4$ for GFORCE¹ and for ICN in Figure 5.25, using a CFL of 0.25 and RK2 for source terms. The GFORCE scheme uses the super-bee slope-limiter.

Clearly, GFORCE¹ is very diffusive indeed; the solution only retains about 45% of its original amplitude even at resolution $\rho = 4$, and has almost decayed away completely for $\rho = 1, 2$. ICN is much less

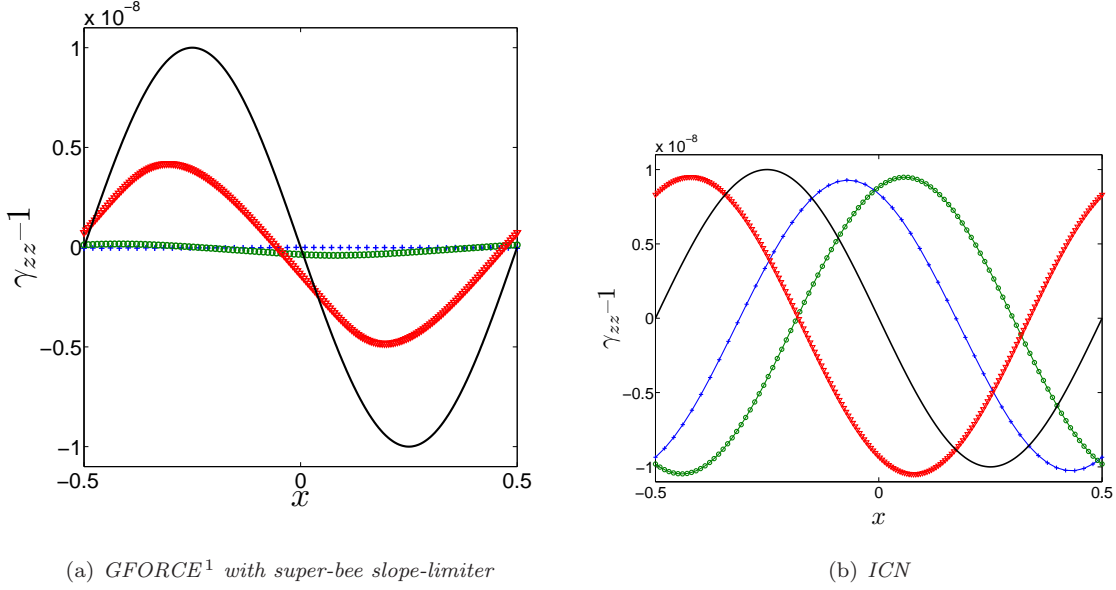


Figure 5.25: The effect of varying the resolution for the linearized wave test. The blue + signs are for $\rho = 1$, the green circles are for $\rho = 2$, and the red triangles are for $\rho = 4$. The exact solution is given by the solid line.

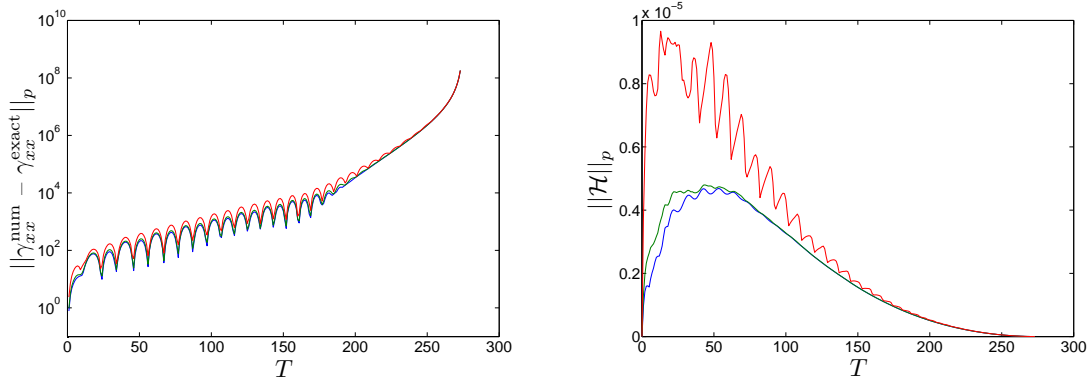
diffusive, even for low resolutions, but has a large phase shift, where $\rho = 1$ has a phase shift of almost a full cycle, $\rho = 2$ about half a cycle, and $\rho = 4$ is the only resolution to come close to the exact solution.

In conclusion, we note that this demonstrates the highly diffusive nature of the schemes that we are using, and also the relatively high phase shift when compared with other schemes and formulations used to evolve this test.

Gowdy wave test

We now show our results for the Gowdy wave test as described in §3.8.3.

Measuring errors We first show, in Figure 5.26, six different ways of evaluating the accuracy of the schemes used. In Figure 5.26(a), we show the \mathcal{L}_1 , \mathcal{L}_2 , and \mathcal{L}_∞ norms of the error in the γ_{xx} component, and in Figure 5.26(b), we show the same norms of the Hamiltonian constraint \mathcal{H} . We see that the Hamiltonian constraint increases to a maximum value before decreasing in time until the run crashes due to large accumulated errors. This is somewhat counter-intuitive, as we would expect \mathcal{H} to increase with time until the run ended. Indeed, this is what we see in equivalent plots in Babiuc *et al.* [11] where the test is applied to various codes implementing various formulations of the Einstein equations. It is the case, however, that \mathcal{H} converges to zero with the appropriate order of convergence as resolution increases, so we believe that it is correctly calculated, and that the differing behaviour is due to the Bona-Massó



(a) \mathcal{L}_1 (blue line), \mathcal{L}_2 (green line), and \mathcal{L}_∞ (red line) norms of the error in γ_{xx} (b) \mathcal{L}_1 (blue line), \mathcal{L}_2 (green line), and \mathcal{L}_∞ (red line) norms of \mathcal{H} , the Hamiltonian constraint.

Figure 5.26: Plots showing six different ways of measuring the accuracy of our numerical solutions to the backward Gowdy wave test. The scheme used here is the van Leer limiter, the FORCE¹ scheme, the RK2 scheme for the source terms, and a resolution of $\rho = 4$.

formulation rather than an error in our algorithm or its implementation. Nevertheless, from now on, we use the \mathcal{L}_1 norm of the error in γ_{xx} to compare the different schemes used.

Method comparison In Figure 5.27 we show a comparison between the FORCE¹, FORCE³ and GFORCE¹ schemes as used to evolve the Gowdy test. We used the RK2 scheme for evolving the source terms. As we would expect from the analysis in §4.2.2, increasing the number of stages used in the FORCE scheme increases the accuracy of the simulation. However, switching to the GFORCE scheme instead brings a much better improvement. An improvement to the basic FORCE¹ scheme is brought about by using the super-bee slope-limiter in place of the van Leer limiter. This is as we would expect as the wave is entirely smooth with no discontinuities, so any slope-limiting at all actually decreases the accuracy of the solution. There was very little difference between the results using the RK2 and the Euler or RK4 schemes. Also, the HLL solver results were very close to those for GFORCE, and increasing the number of stages on GFORCE did not improve upon the results, and so we do not show them. However, the best results of all are obtained using the ICN scheme, which manages to evolve the problem nearly up to time 600 before crashing. In fact, using the GFORCE scheme gives results fairly close to those of ICN for early times.

Convergence rate We can further demonstrate the convergence rate of the schemes by plotting the logarithm of the ratio of the \mathcal{L}_1 norms of the errors at $\rho = 2$ and $\rho = 4$, divided by the log of 2. This gives a plot of the order of convergence of the method, as seen in Figure 5.28. We see that for both GFORCE and ICN, the order of convergence starts off at around 2, and then fluctuates somewhat as

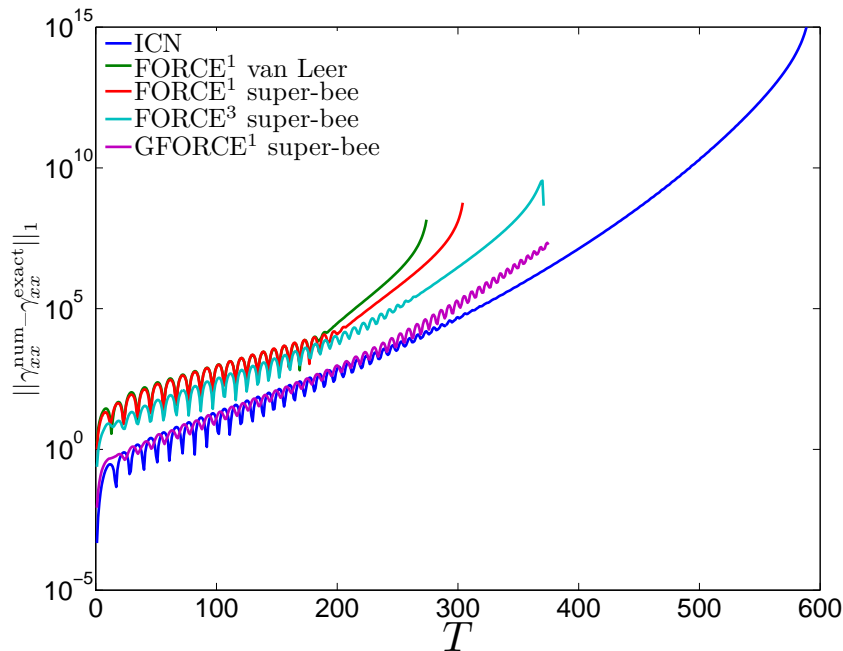


Figure 5.27: Comparison of various schemes used to evolve the backwards Gowdy wave. The simulations are all run at resolution $\rho = 4$, using the RK2 scheme to evolve the sources. The run using ICN continues to about time 600 before crashing completely.

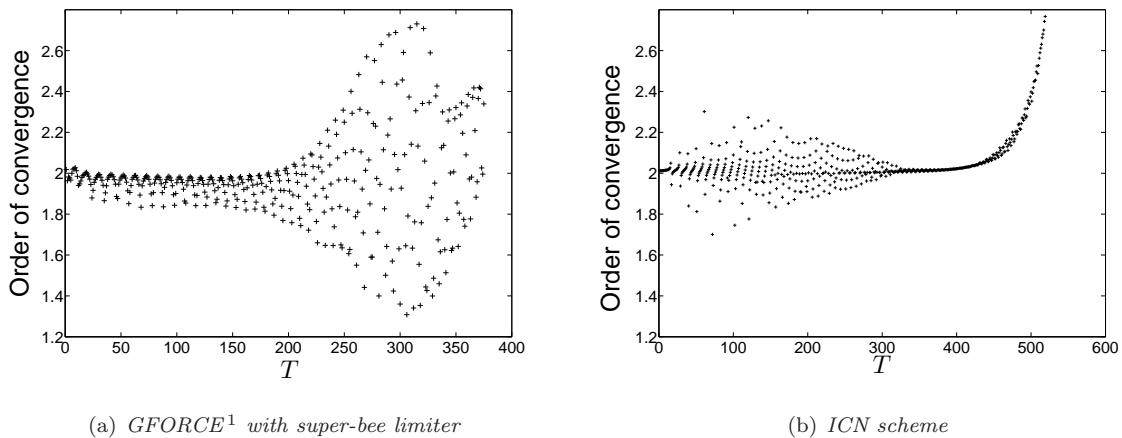


Figure 5.28: Order of convergence varying in time, calculated using $\rho = 2$ and $\rho = 4$ for the Gowdy wave test-case with the time-stepping condition as defined in the test description, and with RK2 for the source evolution.

the accumulated error in the evolution means that we pass outside the domain of convergence of the scheme. However, the order does stay above 1.8 for most of the first half of the evolution, providing further evidence that our schemes achieve the expected convergence rate of 2 on smooth solutions.

Conclusions The Gowdy wave evolution is deliberately designed to test a numerical scheme as there are three orders of magnitude between the γ_{xx} and γ_{yy} coefficients, so that a mixing of the components can lead to inaccurate arithmetic. As finite precision arithmetic is *not* associative, it could benefit the accuracy of the implemented algorithm if we were to arrange the flux calculations so as to reduce arithmetic errors. However, there is no sense in which any one variable can be said to be always larger or smaller than another, for all possible space-times, so this would be an impossible task. The way that the flux calculations are written is to calculate each tensor and term of equations (3.39)-(3.44) individually and then to combine them. This leads to the most readable and easy to debug code, which is a strong advantage. A brief experiment with optimised code generated by MapleTM led to longer run-times, either due to an increase in expression complexity and hence a decrease in the compiler's ability to optimise further, or an increase in cache misses, so this line of thinking was abandoned.

Although the finite-volume approach required more computational time, and produced somewhat less accurate results than ICN, we still persevere with this approach, since the Gowdy wave test is a particularly strong test, and may well not be representative of space-times found in black hole and neutron star interactions. ICN will only cope with smooth space-times, and we believe that we should investigate the suitability of schemes that can cope with space-times that do not have continuous second derivatives.

5.4 Conclusions

In this chapter, we have demonstrated that our algorithm can evolve standard test cases stably and accurately, and that most of the features that we outlined in Chapter 4 are beneficial in reducing errors in the evolution. In particular, we have demonstrated the following:

- Standard test cases in SRHD are easily and accurately evolved in one dimension, giving results of similar accuracy to previously published results using more complex and more computationally expensive schemes (§5.1.1, §5.1.2, and §5.1.3)
- AMR works in one dimension, although it does result in greater run-times, but this is due to overheads that will become less important in higher dimensions. (§5.1.4)
- No problems are caused by the use of a non-uniform grid in one dimension (§5.1.5)
- The GFORCE^k schemes all converge to second order, and increasing the number of stages k reduces the overall error substantially.
- Evolving 1D test cases on a non-Minkowski metric shows that the introduction of source terms does not affect the accuracy of the schemes.

- AMR works in two dimensions, and demonstrates greater than first-order convergence properties for various schemes. (§5.2.2)
- The evolution of one dimensional test-cases on two dimensional overlapping grids is successful. (§5.2.3)
- Evolution on complex overlapping grids, including the use of AMR, gives results which are qualitatively comparable with other similar evolutions. (§5.2.4 and §5.2.5)
- Various improvements, designed to help reduce evolution errors, particularly in the velocity, are partly successful (§5.1.6 and §5.2.7)

In this section we have not shown any major tests in three dimensions. Our validation in three dimensions will be specific to the application to which we intend to apply our algorithm, and we present the relevant background and equations in Chapter 7.

As a result of the test cases examined in this chapter, we conclude that the methods are sufficiently accurate to evolve GRHD problems, and so we look for an application in this area.

We shall not apply our algorithm to any problems which require evolving the metric. Although we have seen performance similar to other formulations on standard tests, we do not have the necessary features of gauge evolution or constraint damping which are likely to be needed for physically interesting problems. However, we have shown that the methods we use are applicable to large systems of equations, and this is a result in itself.

Chapter 6

Software development issues

In this chapter we describe some of the technical aspects of the code we have written for this thesis. We include a short description of Overture, and investigate some performance issues with our code.

6.1 Overture

Overture is a framework for developing software to solve PDEs on overlapping grids. It is developed by Henshaw, Quinlan, and Chand [57], and is written in C++ and FORTRAN, and hence can be compiled and run on any Linux based operating system. It is open source, which allows for careful examination of the code when results are not quite as expected, and for the addition of any features not already implemented.

Overture allows for the generation of arbitrary grids and the interpolation of numerical solutions between them. Although Overture does not provide any numerical methods itself, it does provide all the information, such as local derivatives of the grid mappings, required to construct codes which can apply any numerical method required. Furthermore, it is parallelised.

Although extensive documentation is available for Overture, much of it is designed for researchers developing codes which use Overture. We present a short overview of how the various Overture classes that we have used fit together.

6.1.1 Distributed arrays

Overture uses the A++/P++ set of classes, developed by Quinlan (see relevant documentation at [57]) to distribute data over multiple processors. These allow for the creation of multi-dimensional arrays of integer, single precision, or double precision values. The arrays can be built on one processor only, or distributed across all or some of the processors being used. The data is distributed as evenly as

possible, and allows for the use of ghost values, where some data is duplicated in order to allow numerical methods to access all the data they require without having to fetch data from other processors, requiring time-consuming communication.

Although the A++/P++ classes incorporate methods to access data given appropriate indexes, and also operators such as $+$, $-$, \times , $/$ as well as \sin , \cos and similar functions which can be applied to whole arrays at once, these methods can be somewhat slow as they may involve communication between processors even when it is not necessary. However, A++/P++ also permits direct memory access by way of pointers, although this requires some care to ensure that we do not attempt to access values that are not present on the current processor.

A++/P++ uses column-major ordering¹ of its elements. Then, it is usually assumed in Overture that the coordinate indexes are the first three indexes of the array, and any solution variables stored in the array are indexed by the fourth and subsequent component indexes. This is somewhat less than optimal for our purposes, however, as we frequently require to treat each data-point separately, and this requires reading values from points widely separated in memory, resulting in many cache misses and the subsequent speed penalty. It would be possible to change this within Overture if it were found to be appropriate.

An array's data can be split across any number of processors in any way, although the default way is to ensure a load-balanced approach. An advantage of using the default inter-processor distribution is that the data for the grid geometry, also being split across the processors, is distributed in the same way. Therefore, the grid geometry for a particular cell is on the same processor as the numerical solution data, thus improving potential speed-up.

6.1.2 Grids

The main classes that Overture provides are those concerned with computational grids. The basic grid is a `MappedGrid`, several of which can be collected into a `GridCollection`, which can then be incorporated into a `CompositeGrid`. The `CompositeGrid` includes details of how various `MappedGrids` are combined, with information about how they interpolate from each other, and to which AMR level they belong.

The basic type of class used to define a grid is a `Mapping`. There are several types of `Mapping` included in Overture, including analytically defined mappings such as `SphereMappings` and `BoxMappings`, as well as `Mappings` that are designed to be applied to other `Mappings` such as the `StretchMapping` and the

¹An array `A` of dimension 3×3 is stored in memory in the order `A(0,0)`, `A(0,1)`, `A(0,2)`, `A(1,0)`, ... in row-major ordering and in the order `A(0,0)`, `A(1,0)`, `A(2,0)`, `A(0,1)`, ... in column-major ordering. `C` uses row-major ordering and FORTRAN uses column-major ordering. It is therefore necessary to be careful with array access when using software written in both `C` and FORTRAN.

`MatrixTransform`. These all provide a mapping of the region $[0,1]^m$ into \mathbb{R}^n where we usually have $m = n =$ number of dimensions. The `Mapping` can also provide various information such as the derivatives of the analytical mapping, including all the information we need for the coordinate transformation given by (4.54).

The class `MappedGrid` is then derived from a specific `Mapping` and adds information specific to a grid, such as the various boundary conditions and number of cells in each direction. Several `MappedGrids` can then be made into a `CompositeGrid`. Once the overlapping grid generator `ogen` has been applied to the `CompositeGrid`, the latter includes all the information required about how the grids will interact with each other. Refined grids as required for AMR are incorporated into a `CompositeGrid` with a parameter specifying to which level they belong.

6.2 Optimisation and Performance

In any large parallel computer code, it is important to ensure that the best performance possible is being extracted from the hardware. Various tools are available for profiling codes in order to see where most processing time is being consumed. In this project, much use has been made of Valgrind [79], which has the facility to pinpoint the specific lines of source code that are taking the most time.

We note that “Premature optimisation is the root of all evil” [64], but in this case it was necessary to optimize the code before it was fully tested in order that the tests themselves might be completed sufficiently quickly.

Some initial optimizations included using a few lines of code specially written to invert a symmetric 3-by-3 matrix, instead of relying on a more versatile, but more computationally expensive, library routine. It was also necessary to reduce the amount of inter-processor communication being done by the A++/P++ library. This was done using direct memory-access pointers made available by the library rather than other more robust, but slower, routines to access elements of an array of variables distributed across many processors. Also, we confirmed that each processor could, for example, perform a flux-evolution update on its local grid segment, without communication with another processor. This improved the parallel efficiency of the code. For example, before performing a flux update calculation, we first compute all relevant derivatives of the grid mappings, which may require inter-process communication, so that the geometry information is then stored on the processor where it will be required for the flux calculation so that no further inter-processor communication is required for the flux-update.

A small amount of improvement could perhaps have been made if the A++/P++ arrays had been stored in memory using row-major ordering rather than column-major ordering since extracting the variables associated with a single grid points required accessing widely separated memory locations, with the

Procedure	Time (s)	Proportion of one time-step
Initial grid generation	17.63	79.0%
Initial overlap calculation	7.90	35.4%
Initial data	4.75	21.3%
Solution output	0.78	3.5%
Single time-step	22.32	100%
Calculate Δt	2.84	12.7%
Advance solution - total time	19.48	87.3%
Boundary conditions	2.39	10.7%
RK2 update	0.95	4.3%
Flux update	14.77	66.2%
Interpolate	1.34	6.0%

Table 6.1: *Profiling run for a three-dimensional spherical grid with medium resolution. The run was done on two nodes of the super-computer, using four CPUs per node. Caching and communication overheads may therefore be significant. Note that the first few operations are not performed every time-step, and so their times are not included in the total single time-step.*

substantial numbers of cache-misses that implies. Profiling finally suggested that there were no obvious places to reduce run-time, and at this point more effort was put into perfecting the algorithm, rather than speeding up its implementation.

We used the Intel[®] C/C++ compiler for our final runs, using the compile parameters:

```
-O3 -ipo -xhost -fp-model precise -fp-model source,
```

these having been chosen to provide best speed and accuracy. We note that, presumably due to expression rearrangements performed by the optimizer, the `-fp-model` options were necessary to retain the numerical accuracy of no optimization, even when using `-O3` optimization. This was evident in some tests of recovering primitive variables from conserved variables for velocities close to the speed of light. We have used double precision arithmetic throughout, as it is likely that the numerical methods, particularly the primitive variable recovery, will need this level of accuracy.

We performed a profiling run of our code, to see where most of the computational time was spent. This run was done for the base model UB1 run on a medium resolution grid as described in Table 7.2. The timings are shown in Table 6.1.

We see that the flux calculations are by far the most computationally expensive part of the code. The

Nodes \times Processors per node	Time(s)	Speed-up	Efficiency
1 \times 1	31254.9	1.00	1
1 \times 2	17441.9	1.79	0.896
2 \times 1	17262.6	1.81	0.905
1 \times 4	11247.6	2.78	0.695
2 \times 2	10257.8	3.05	0.762
2 \times 4	7130.31	4.383	0.550

Table 6.2: *Speed-up statistics for the code. This was a three-dimensional run using the low resolution grid over 1000 time steps. The speed-up is the ratio of the time taken to run the code in parallel to that taken for a serial run. The efficiency is the speed-up divided by the number of processors.*

next most expensive part is the calculation of the time-step which requires a change of coordinate basis to get the correct time-step for a curvi-linear grid, which accounts for its relatively large contribution to the CPU time.

The interpolation is also fairly time consuming. This is due to the fact that we use fifth-order Lagrange interpolation. This requires information from 125 cells per interpolation point. Due to the column-major ordering used by Overture, there will be substantial caching overheads involved here. Also, we have interpolated the metric values as well as the fluid values, so there is some unnecessary overhead involved here. The RK2 update is entirely local to each cell, and requires no geometric data to evaluate, and so takes the least time.

We also tested our code to ascertain how well it had been parallelised. We ran exactly the same model parameters in each case on various combinations of processors. The super-computer setup² is such that there are 4 CPUs per node, with 8GB of RAM per node, so that a reasonably sized problem could be run even on only 1 processor, since it had 8GB available to it. We ran our simulation for 1000 time steps, and recorded the total time for the program to run. The timings can be seen in Table 6.2.

The speed-up from one to two processes is encouraging, and suggests a reasonable level of parallelisation. However, the subsequent speed-up for four processes on one node is less so. It is likely that this is due to the super-computer’s architecture. Each node consists of two dual-core processors, so that, when we go from one to two processes, each process sits on a different processor, with separate caches. However, four processes will require that two processes sit on each processor, which will result in less performance gain due to clashes between the cache requirements of each core. This is particularly noticeable when comparing four processes on one node (efficiency 0.695), and two processes on each of two nodes

²We used the Darwin supercomputer of the University of Cambridge High Performance Computing Service. Details of its architecture can be found at <http://www.hpc.cam.ac.uk/services/darwin.html>.

(efficiency 0.762), so that the caching problems outweigh any communication overhead between nodes.

However, for the grid resolutions we have used, the efficiencies and timings demonstrated here are sufficient to allow us to perform the validation and studies we need in an acceptable length of time, and therefore we do not seek any further optimizations.

6.3 AMR

Our AMR routines work in parallel, and maintain the convergence and accuracy shown in §5.2.2. However, the efficiency is not sufficient to give any appreciable performance gain over running on a uni-grid resolution. We therefore have not used AMR in any of our subsequent testing.

The times taken for a run including AMR are shown in Table 6.3. These were generated for a simulation of UB1 on a grid of slightly lower resolution than the lowest given in Table 7.2, with $n = 30$, and with a single level of refinement factor two, making it equivalent to the high-resolution grid for $n = 60$. The code was run on two nodes with four processes running on each, so that caching overheads are probably significant here.

For the time-step that we used for these timings, the coarsest level contained 470 862 cells and the refined level contained 988 828 cells. This would suggest that advancing the refined level would take approximately

$$\frac{470682}{2 \times 988828 + 470862} = 80.8\% \tag{6.1}$$

of the time, which is very close to the actual value found, suggesting that AMR does not add overhead disproportionately to different refinement levels.

We see that the interpolation at level 1 is very time consuming, almost as much as the flux update. The interpolation contributes much more to the CPU time than for a uni-grid run; this is presumably due to the extra effort required in checking the interpolation points for refined grid levels.

Procedure	Time (s)	Proportion of one time-step
Initial grid generation	3.30	8.0%
Initial overlap calculation	1.96	4.7%
Initial data	1.42	3.4%
Solution output	1.08	2.6%
AMR regrid	31.57	76.3%
Single time-step	41.36	100%
Calculate Δt	1.81	4.4%
Level 0 - total time	6.07	14.7%
Boundary conditions	0.91	2.2%
RK2 update	0.35	0.8%
Flux update	3.93	9.5%
Interpolate level 0	0.50	1.2%
Store level 1 boundary cells	0.38	0.9%
Level 1 - total time	33.21	80.3%
Boundary conditions	0.15	0.4%
RK2 update	1.16	2.8%
Flux update	18.23	44.1%
Interpolate level 1	13.67	33.1%
Interpolate level 0 cells from level 1	0.27	0.7%

Table 6.3: *Profiling statistics for a three-dimensional run including AMR. The first few operations are not performed every time-step and so their times are not included in that for a single step. Also, although we do two RK2 updates per level, these are all included in a single time, and both level 1 sub-steps are included in the level 1 update time.*

6.4 Conclusions

In this chapter we have seen some of the issues involved with implementing a substantial numerical algorithm, and outlined the way in which this has been handled by Overture. The complexity of such an algorithm requires a lot of time to be spent in its implementation, and the issues of arbitrary grid geometries, overlapping grids, and parallelisation are particularly complex. Therefore, the use of Overture as a base has saved a substantial amount of time in moving from the algorithm to its implementation.

Further, we have outlined some issues of optimization and performance. Although we have managed to produce a code that produces the results we needed sufficiently quickly, there are still some concerns over computational efficiency, particularly with regard to parallelisation.

Chapter 7

Bondi-Hoyle-Lyttleton Accretion

We have designed our algorithm to evolve both a fluid and the metric using the same numerical methods. However, our tests show that the scheme and formulation used to evolve the metric are not particularly well-suited to this, being somewhat less accurate than some other commonly used formulations. We therefore restrict our attention to evolving a general relativistic fluid on a stationary metric background.

The specific problem we have chosen to consider is that of Bondi-Hoyle-Lyttleton accretion. In this chapter we outline the background of this problem, work done previously in this area, and some tests of our code to ensure that it can produce good results for this problem.

7.1 Bondi-Hoyle-Lyttleton Accretion

Although we have analytic solutions of Einstein's equations such as the Kerr black hole, black holes will never exist in a complete vacuum in the physical universe. It is more likely that they will either be close to one or more stars or that they will be moving through an interstellar gas. It is this latter case that is known as Bondi-Hoyle-Lyttleton (BHL) accretion. BHL accretion usually refers to the more general case of supersonic flow of a gas onto a gravitating accretor, but we specialise to a black hole for our purposes.

The starting assumptions for BHL accretion, then, are that a stationary (fixed centre) black hole is subject to a wind composed of a fluid coming from infinity at Mach $\mathcal{M}_\infty = v_\infty/c_\infty$, where c_∞ is the sound speed in the fluid and v_∞ is the fluid speed, both measured at a large distance upstream of the black hole. The black hole has mass M and spin a , and the spin axis may be at any angle to the wind direction. The gas may be of any composition, but we restrict ourselves to the cases of an ideal gas, with adiabatic index Γ , and a stiff fluid. We note that by a simple transformation, this is equivalent to

a black hole moving at some constant velocity through a stationary gaseous medium. We assume that the accretion of the fluid onto the black hole does not change its mass significantly, i.e. we are using the test-fluid approximation.

7.1.1 Relativistic BHL Accretion

Some axisymmetric simulations of accretion onto a moving black hole were performed by Petrich *et al.* [82], although they used finite-difference methods with artificial viscosity. Following them, there is a series of papers on numerical calculations of accretion rates by Font, Ibáñez, and Papadopoulos [40, 41, 42]. The most general case studied was that of wind accretion onto a spinning Kerr black hole, using a range of Mach numbers, adiabatic indices, and spin rates. However, these were done either assuming axisymmetry or in the equatorial plane (thin disk approximation), although this latter “dimensional simplification still captures the most demanding aspect of the Kerr background, which is encoded in the large azimuthal shift vector near the horizon.” [42]. The reason for this was limited computational resources, which at the time would not allow high-resolution three-dimensional calculations to be performed in a reasonable time.

Further, the first two of these papers [40, 41] did not use excision, as such. Rather they used Boyer-Lindquist coordinates (3.68), which have a coordinate singularity at the horizon. They made use of a tortoise coordinate r_* in the radial direction such that [42]

$$dr_* = \frac{r^2 + a^2}{r^2 + a^2 - 2Mr} dr, \quad (7.1)$$

so that on the computational grid, the outer event horizon is situated at $r_* = -\infty$. This defines a grid that becomes ever more refined as it approaches the horizon, without ever reaching it. Any adverse effects caused by not performing the excision within the horizon are thus put at a sufficiently large distance (in computational space) that their effect on the numerical solution are minimised. The third paper in the series [42], however, did use Kerr-Schild coordinates, and so used excision inside the horizon, although since the setup was two-dimensional only, this was easily achieved by using plane-polar coordinates and enforcing suitable outgoing boundary conditions at some non-zero radius.

7.1.2 Newtonian BHL accretion

Another body of work on wind accretion is due to Ruffert [92, 93, 94, 95]. These look at Newtonian BHL accretion onto an absorbing compact object. This is similar in nature to the situation in which we have a black hole, but we expect that relativistic effects will affect the results, possibly giving a stabilising effect. Ruffert’s simulations were done in three dimensions for a range of accretor sizes and incident wind directions, using a nest of Cartesian grids at various resolutions (fixed mesh refinement).

Ruffert’s aim was to study the stability of the flows by introducing perturbations into the wind (see [94] and [95]), and some preliminary suggestions were made, particularly in [93]. These were that instabilities were present mainly for flow at high Mach numbers ($\mathcal{M}_\infty \gtrsim 3$) and for small accretor size (equivalent to low asymptotic velocity v_∞ for our purposes).

We have found that our algorithm does not deal well with the very low wind velocities required to probe the regions of parameter space in which instabilities might be expected to occur. This may be to do with the (necessary) use of upwind boundary conditions at a finite radius, but we have not investigated further.

7.1.3 Applications of BHL accretion

There are several applications of BHL accretion to realistic astrophysical situations, and some are outlined in a review by Edgar [34]. The most relevant application is that of extreme mass ratio inspirals (EMRI). These are systems where a stellar mass black hole is orbiting a super-massive black hole (SMBH). The smaller black hole will gradually spiral in towards the larger black hole as a result of gravitational wave emission. However, it is likely that the SMBH will be surrounded by an accretion disk or torus of gas or dust, and if the smaller black hole is orbiting within this, then it will experience a drag due to accretion of the disk material, which will affect its inspiral path.

Barausse and Rezzolla [15] showed that the hydrodynamic drag was of less importance than the loss of energy due to gravitational wave emission (radiation reaction). Further, the main effect in the hydrodynamical drag was that of the gravitational field generated by the fluid’s wake. However, it could still be of interest to examine the effect of the drag rate and direction on the black hole.

Another application is that of high-mass X-ray binaries. In this case, a black hole (or some other compact object), is orbiting a massive star and accreting mass from it. The accreting fluid reaches a very high temperature due to the high speeds that result, and emits X-rays, which can be detected. In this case, the interest would be in the effect of the spin of the black hole, or any inhomogeneity of the wind, on the X-ray luminosity, which could be detected by conventional electromagnetic spectrum telescopes.

7.2 Model Description

As stated above, we use a Kerr-Schild coordinate system for the black hole. The inner excision boundary is always placed between the outer horizon at r_+ and the inner horizon at r_- , although its precise location varies depending on the spin parameter. The outer boundary always lies at $50M$.

As initial data for our fluid we use constant density, constant pressure, and constant velocity (in Kerr-Schild coordinates) over the whole domain, and allow these to evolve to a stationary state. The

density and pressure are defined so as to give a constant sound speed of c_∞ if required (it is not, for example, required for a stiff fluid where the sound speed is always $c = 1$). The velocity is normalised as follows: given a wind direction vector, n^i , and a velocity at spatial infinity, v_∞ , the wind velocity at a point is defined as

$$v^i = \frac{v_\infty n^i}{\sqrt{\gamma_{kl} n^k n^l}}. \quad (7.2)$$

We also allow for the fluid to have a non-uniform density or velocity upstream of the black hole, following Ruffert, who defines the density and velocity to be

$$\begin{aligned} \rho_\infty &= \rho_0 \left(1 - \frac{1}{2} \tanh \left[2\epsilon_\rho \frac{y}{R_A} \right] \right), \\ v_{x\infty} &= v_0 \left(1 + \frac{1}{2} \tanh \left[2\epsilon_v \frac{y}{R_A} \right] \right), \quad v_{y\infty} = v_{z\infty} = 0, \end{aligned} \quad (7.3)$$

where the accretion radius is defined to be

$$R_A = \frac{2GM}{v_0^2}, \quad (7.4)$$

although Font and Ibáñez use a different definition:

$$r_a = \frac{M}{v_\infty^2 + c_\infty^2}. \quad (7.5)$$

One would ordinarily use a linear modulation for the density or velocity perturbations but, for sufficiently large ϵ_ρ or ϵ_v , this leads to negative velocity or density at large distances from the black hole. Ruffert therefore uses the tanh variation as it is approximately linear near the origin but tends to ± 1 as $r \rightarrow \infty$, and so avoids this problem while maintaining approximately the same problem setup. In our setup, the pressure is defined such that the sound speed is given by a constant value c_∞ upstream.

Following (7.3), therefore, we generalise to the case of a non-flat metric, and our initial density and velocity are given by

$$\begin{aligned} \rho(\mathbf{x}) &= \rho_\infty \left(1 - \frac{1}{2} \tanh \left[2\epsilon_\rho \frac{\mathbf{x} \cdot \mathbf{p}}{R_A} \right] \right), \\ v^i(\mathbf{x}) &= \frac{v_\infty n^i}{\sqrt{\gamma_{kl} n^k n^l}} \left(1 + \frac{1}{2} \tanh \left[2\epsilon_v \frac{\mathbf{x} \cdot \mathbf{p}}{R_A} \right] \right), \quad \text{with } R_A = \frac{2M}{v_\infty^2}, \end{aligned} \quad (7.6)$$

where n^i gives the wind direction, and \mathbf{p} is orthogonal to \mathbf{n} , and specifies the direction in which the perturbation occurs. The vectors \mathbf{n} and \mathbf{p} are unit vectors defined at spatial infinity, so that the notions of orthogonality and normalisation are defined in terms of the flat-space Euclidian metric $\delta^i_j = \eta^i_j$. We note that we could allow the density and velocity perturbations to be in different directions, but this would add significantly to the size of our parameter space, and so we do not follow this variation.

7.3 Numerical parameters

In this section we detail the specific numerical schemes that we use, and also the grid structure that we use for the problems described above.

7.3.1 Evolution methods

From our previous validation experiments, and from a desire to use as little computing power as possible, we have decided to use the GFORCE¹ scheme for flux evaluation, using piecewise-linear reconstruction, with the van Leer slope-limiter. We use the RK2 scheme for evaluating source terms, and a CFL parameter of 0.95. The dimensionally split scheme is used as it is far less computationally expensive, and has been found to be stable for our problems. We also use the correct metric averaging approach from §4.7, the free-streaming correction §4.6.3, and we limit on the primitive variables. None of the other corrections referred to in §5.1.6 and §5.2.7 have been used, as they tend to take a substantial amount of computational time for very little improvement.

7.3.2 Grid setup

There are several possibilities for creating a spherical grid with no polar singularities. The most well-known are the Yin-Yang grid [62], where two identical grids mesh to form the entire sphere and the cubed sphere, where six identical patches are used to cover the sphere, arranged like the sides of a cube. The latter is the approach used by Zink *et al.* [118] and adapted to cover a full spherical domain if necessary by the inclusion of a central seventh grid.

However, for our simulations, we have used a scheme as follows to remove the polar singularity:

- Take two identical hollow spheres
- Remove their polar singularities by restricting the latitudinal coordinate to $\theta \in [0.18\pi, 0.82\pi]$
- Rotate one sphere through $\pi/2$ about a line perpendicular to its axis.
- Combine the two spheres and form an overlapping grid.

The resulting grid is shown in Figure 7.1.

As stated in §4.10.1, we have defined our grid so that the grid cells are always approximately cubic. To this end, we have used a stretching transformation that makes the grid spacing in the radial direction proportional to the radius. Further, the numbers of cells in the ϕ (longitudinal) and θ (latitudinal) directions are determined so that the cells are approximately cubic. The numbers of cells in the ϕ and θ

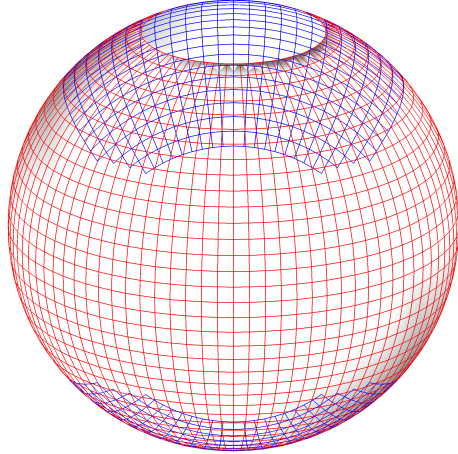


Figure 7.1: *Overlapping grid structure used for three-dimensional simulations. It is formed from two hollow spheres. The blue sphere is not wholly present as parts of it have been removed where it covers the same regions as the red sphere.*

directions are defined as

$$\phi_{\text{cells}} = 1 + 4 \left\lceil \frac{\pi}{4} r_{\text{inner}} n \right\rceil, \quad \theta_{\text{cells}} = 1 + 2 \left\lceil (\theta_{\text{max}} - \theta_{\text{min}}) \frac{\pi}{4} r_{\text{inner}} n \right\rceil \quad (7.7)$$

where n is a number characterising the resolution of the grid, and r_{inner} is the excision radius. The number of radial cells is then defined as

$$r_{\text{cells}} = \left\lceil \frac{n}{2} k r_{\text{inner}} \right\rceil, \quad \text{where } k = \ln \left(\frac{r_{\text{outer}}}{r_{\text{inner}}} \right), \quad (7.8)$$

and the transformation of the radial coordinate is defined as

$$r^* = r_{\text{inner}} \exp \left(k \frac{r - r_{\text{inner}}}{r_{\text{outer}} - r_{\text{inner}}} \right). \quad (7.9)$$

This is a transformation from $[r_{\text{inner}}, r_{\text{outer}}]$ to itself, and $\frac{dr^*}{dr} \propto r^*$ as required.

Although the resolution parameter n does not have a trivial relationship to the number of cells in the grid, the setup remains such that doubling n will multiply by eight the number of cells in the grid overall. The relation with the minimum grid-cell size is not as trivial, however, due to the non-linear mapping from computational space onto physical space.

Although the outer horizon of a spinning black hole is not spherical, we always use a spherical excision boundary. The ellipsoidal outer horizon has a circular cross-section in the equatorial plane, with (Cartesian) radius

$$\sqrt{x^2 + y^2 + z^2} = \sqrt{2 + 2\sqrt{1 - a^2}}, \quad (7.10)$$

and the minor axis of the horizon lies on the z -axis, and corresponds to a radius

$$\sqrt{x^2 + y^2 + z^2} = 1 + \sqrt{1 - a^2}. \quad (7.11)$$

Spin a	0	0.5	0.9
$r_+(\theta = 0, \pi)$	2	1.866	1.436
$r_+(\theta = \pi/2)$	2	1.932	1.695
Excision radius	1.4	1.477	1.307

Table 7.1: *Set of radii at which we perform spherical excision, ensuring that the excision is inside the outer horizon.*

We therefore ensure that our spherical excision boundary is within the latter radius, so that the whole of the region (up to a large spatial distance) outside the outer horizon is covered by the computational grid. The excision radii for the three spins we use are given in Table 7.1. The scheme used to calculate them also ensures that the two ghost cells inside the excision boundary still lie outside the inner horizon.

Our reason for using this particular grid setup is that within Overture there is no explicit mapping for a cubed sphere or the Yin-Yang grid. However, one could be written and incorporated into our implementation using the analytic mappings available in the literature [62, 118]. Alternatively, Overture allows for the definition of a grid from its vertices alone, and this could be more easily implemented. This does tend to lead to lower accuracy, however, as the mapping derivatives have then to be calculated using finite differences, rather than from an analytic mapping. These calculations are also more computationally expensive than the analytic mappings. The parameters of the grids we have used for our results are shown in Table 7.2. There are three resolutions, which we will refer to as low, medium, or high, corresponding to $n = 35, 50, 60$ respectively.

Resolution	low	medium	high
n	35	50	60
θ cells	37	53	63
ϕ cells	113	161	193
r cells	69	98	118
cells per grid	288 489	836 234	1 392 768
total cells	576 978	1 672 468	2 785 536
interpolation points	62 928	120 736	171 808
$\Delta\theta$	5.58×10^{-2}	3.86×10^{-2}	3.24×10^{-2}
Minimum $\Delta\phi$ at $r = 2M$	6.01×10^{-2}	4.21×10^{-2}	3.50×10^{-2}
Maximum $\Delta\phi$ at $r = 2M$	1.12×10^{-1}	7.84×10^{-2}	6.56×10^{-2}
Minimum Δr	5.87×10^{-2}	4.08×10^{-2}	3.37×10^{-2}
Maximum Δr	2.77	1.96	1.63

Table 7.2: Grid cell statistics for the resolutions used. Points removed by the overlapping grid routines are still included in these counts. These statistics are given for the case when $a = 0$. In this case, the inner grid radius is $1.4M$, and the outer grid radius is $50M$. For spins $a \neq 0$, the inner radius changes as in Table 7.1, thereby changing the radial spacing, but the numbers of cells in all dimensions and the angular resolutions remain fixed.

7.4 Post-processing of results

After running the simulations, we must process them in order to extract useful conclusions from them and to compare different models in a quantitative way.

7.4.1 Mass and momentum accretion

In order to calculate the mass accretion rate \dot{M} from our numerical output, we note that [82]

$$\partial_\mu(\sqrt{-g}J^\mu) = \sqrt{-g}\nabla_\mu J^\mu = 0, \quad (7.12)$$

where $J^\mu = \rho u^\mu$ is the conserved four current, and $\sqrt{-g} = \alpha\sqrt{\gamma}$.

Then, the mass M inside a volume V is given by

$$\begin{aligned} M &= \int_V \sqrt{\gamma}\rho W dV = \int_V \sqrt{-g}J^0 dV \text{ since } u^0 = W/\alpha, \\ \text{so that } \dot{M} &= \int_V \frac{\partial}{\partial x^0}(\sqrt{\gamma}D) dV = \int_V \frac{\partial}{\partial x^0}(\sqrt{-g}J^0) dV \\ &= \int_V \sqrt{-g}\nabla_\mu J^\mu \stackrel{0}{=} \partial_i(\sqrt{-g}J^i) dV \end{aligned} \quad (7.13)$$

$$\begin{aligned}
&= - \int_S \sqrt{-g} \rho u^i d\mathbf{S}_i \\
&= - \int_S \alpha \sqrt{\gamma} \rho W \left(v^i - \frac{\beta^i}{\alpha} \right) d\mathbf{S}_i .
\end{aligned}$$

In order to evaluate this numerically, we interpolate the integrand at evenly spaced points in the θ and ϕ coordinates around a sphere with some radius r and sum them weighted by the infinitesimal area $d\mathbf{S}_i = \gamma_{ij} \mathbf{n}^j dA$ at those points. We use 150 points in the θ (latitudinal) direction and 300 points in the ϕ (longitudinal) direction. This is sufficient as we show in §7.5.2.

We carry out the calculation in Kerr-Schild coordinates, as opposed to Font and Ibáñez, who calculated in Boyer-Lindquist coordinates. As the result is a scalar, it should be the same in both coordinate systems.

In [83], however, the density ρ in the above is replaced by $n = \sqrt{\rho}$, the baryon density, and so for comparisons with their exact solution, we use this slightly altered expression when dealing with a stiff fluid.

We also approximate the rate of accretion of linear momentum onto the black hole in the manner suggested by Petrich *et al.* [82]:

$$\dot{p}^i = - \int_{\partial V} T^{ij} \sqrt{-g} dS_j . \tag{7.14}$$

This is only an approximation since it neglects long-range gravitational effects, and also needs to be evaluated at a large radius in flat space-time, whereas we evaluate it at the same radius as the mass accretion rate.

7.4.2 Velocities

In order to provide a suitable comparison with previous work, we transform the velocities we obtain into those in Boyer-Lindquist coordinates, using the transformation (3.72). This is only possible outside the horizon of the black hole, and so results derived from these cannot extend inside the black hole. We can calculate the total velocity $v = \sqrt{\gamma_{ij} v^i v^j}$ and, in some cases, locate the stagnation point of the flow, which lies downstream of the black hole.

We can also extract some more information about the flow structure by calculating streamlines. These are found from a single snapshot of the flow, and so would not be streamlines in the strict sense, except that, since we reach steady state, the flow has no time dependence. When plotting streamlines, we use the vector

$$\mathbf{w} = \sqrt{\frac{\gamma_{ij} v^i v^j}{\delta_{ij} v^i v^j}} (v^x, v^y, v^z) , \tag{7.15}$$

which satisfies

$$w^2 = \delta_{ij} w^i w^j = \gamma_{ij} v^i v^j = v^2 , \tag{7.16}$$

and then requires no further alterations to Overture’s streamline plotter, which is designed for Newtonian flows.

7.4.3 Shock angles

In §8.2 we shall see that under some circumstances, including the ones we investigate, a shock-cone forms. Far from the black hole, the sides of the cone become straight, and we use this part of the flow to calculate the opening angle of the cone.

In order to find the shock location, we plot the velocity $\sqrt{\gamma_{ij}v^iv^j}$ in Boyer-Lindquist coordinates around circles centred on the origin, at various radii. We can then take the centre of the jump in velocity as the shock location, since conserved schemes preserve the shock location (see §4.2). The circles are in the plane containing the wind direction and the y -axis (or the perturbation direction, if different). We show one of these plots in Figure 7.2, where we see that, although the shock is spread over about 11° , the centre of the shock is fairly easy to find. In particular, we note that the centre of the shock in the angular direction corresponds to the centre of the shock in the velocity component, as we would expect from the conservation property of finite-volume methods. The data points are interpolated; however, we interpolate to the same number of points as there are grid cells, so there is a close correspondence between the data points and the actual values computed.

However, since the shock-cone is not centred at the origin, we then have to calculate the actual angle of the shock with the wind direction as

$$\theta = \tan^{-1} \left(\frac{r_1 \sin \theta_1 - r_2 \sin \theta_2}{r_1 \cos \theta_1 - r_2 \cos \theta_2} \right), \quad (7.17)$$

where (r_1, θ_1) and (r_2, θ_2) are two points on the shock. This formula is derived to find the angle of the line passing through these two points to the horizontal axis. The radii used here should be sufficiently large that the shock has become straight. In practice, we use $r_1 = 20M$ and $r_2 = 30M$. We calculate the angle of the shock-cone both above and below the axis. This is illustrated in Figure 8.1 in Chapter 8.

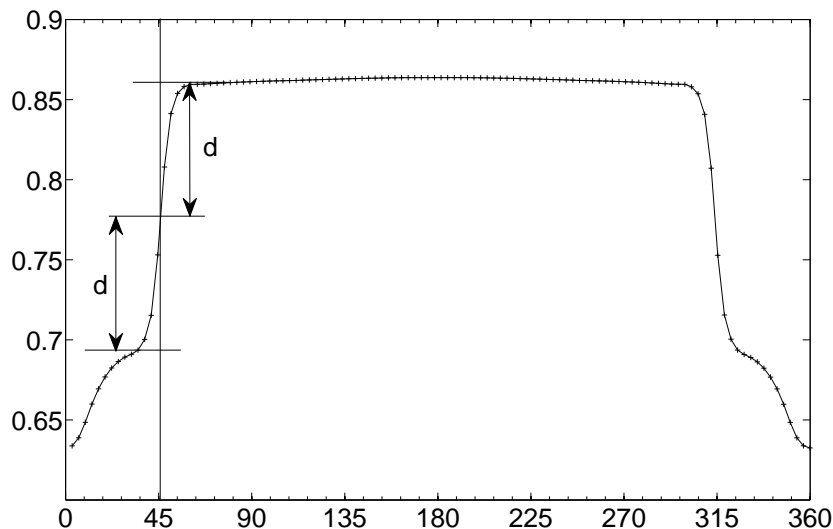


Figure 7.2: *Example of how to calculate the shock angle from a plot of velocity. The upper and lower horizontal lines mark the upper and lower bounds of the shock, the central horizontal line is midway between these, and the shock location is taken to be where this crosses the numerical solution, at 45° in this case. The plot is of v against angle around a circle at radius $r = 30M$ for the model UB1 (see Chapter 8).*

7.4.4 Observable effects

In practice, it is hard to resolve the fluid flow of real black hole systems at anywhere near the required resolution to distinguish the above effects, using current technology. However, two effects may be observable:

Firstly, the high densities and temperatures which occur near the black hole horizon lead to very high fluid temperatures, and therefore to the fluid emitting X-rays, which can be detected using current technology. An approximate procedure for calculating the X-ray luminosity can be found in [101]. An extension to include relativistic effects can be found in [100]. Since this makes a significant difference, of around 50%, to the luminosity, and would take substantial effort to calculate, including a generalisation of the argument in [100] to non-axisymmetric effects, we did not perform the calculation, but it could be performed in theory using the results we have obtained.

Secondly, when applied to the problem of EMRI, outlined in §7.1.3, the different drag rates could affect the rate of inspiral, although the overall impact will be low. This inspiral rate could be detected, although gravitational wave detectors may be required to provide accurate data.

7.5 Validation of numerical schemes

Although we have validated our algorithm in simple 1D cases, and also in 2D cases, we have not yet shown that our setup is robust enough to evolve three-dimensional flows for very long periods of time accurately. Further, there may be boundary condition issues, both at the excision boundary and at spatial infinity that need to be rectified. We therefore seek to test our implemented algorithm against known results in this area.

7.5.1 Exact solutions

Analytical results exist for some special cases of wind accretion. The first involves a perfect fluid with adiabatic index Γ . The special case in which $\mathcal{M}_\infty = v_\infty = 0$ is known as Bondi-accretion in the non-relativistic case, and the equivalent relativistic result is that the rate of mass accretion is: [101]

$$\dot{M} = 4\pi\lambda_s M^2 \rho_\infty c_\infty^{-3},$$

$$\text{where } c_\infty = \sqrt{\frac{p_\infty \Gamma}{\rho_\infty}} \text{ and } \lambda_s = \left(\frac{1}{2}\right)^{(\Gamma+1)/2(\Gamma-1)} \left(\frac{5-3\Gamma}{4}\right)^{-(5-3\Gamma)/2(\Gamma-1)}. \quad (7.18)$$

Note that this result is for a non-rotating black hole, and applies only when equilibrium has been established, which is why there is no radial dependence.

Secondly, there is a surprisingly general solution for wind accretion, derived by Petrich, Shapiro, and Teukolsky [83]. Here, an exact, stationary solution for the accretion of a stiff fluid (see §3.3.1) onto a Kerr black hole in Boyer-Lindquist coordinates is presented. The solution is given for any spin, and for any direction of incidence of the wind, by the components of the four-velocity u_μ and the density n as follows:

$$\begin{aligned} n \cdot u_t &= -u_\infty^0, \\ n \cdot u_r &= -(r_+^2 + a^2) \frac{u_\infty^0}{\Delta} + u_\infty \cos \theta \cos \theta_0 + u_\infty \Re\left([1 + ia(r - M + ia)/\Delta] \right. \\ &\quad \left. \times \sin \theta \sin \theta_0 e^{i(\phi - \phi_0 - \chi)}\right), \\ n \cdot u_\theta &= -u_\theta(r - M) \sin \theta \cos \theta_0 + u_\infty \Re[(r - M + ia) \cos \theta \sin \theta_0 e^{i(\phi - \phi_0 - \chi)}], \\ n \cdot u_\phi &= -u_\infty \Im[(r - M + ia) \sin \theta \sin \theta_0 e^{i(\phi - \phi_0 - \chi)}], \\ n^2 &= [(r^2 + a^2)u_\infty^0 - a \cdot (n \cdot u_\phi)]^2 / (\Sigma \Delta) - [(nu_\phi) - a \sin^2 \theta (u_\infty^0)]^2 / (\Sigma \sin^2 \theta) \\ &\quad - (\Delta/\Sigma)(nu_r)^2 - (nu_\theta)^2 / \Sigma, \end{aligned} \quad (7.19)$$

where

$$\chi = \frac{1}{2\sqrt{a(M^2 - a^2)}} \ln \left(\frac{r - r_-}{r - r_+} \right) \text{ and } \Sigma = r^2 + a^2 \cos^2 \theta. \quad (7.20)$$

The incident direction angles θ_0 and ϕ_0 are extracted from the vector v^i , being the polar angles corresponding to the wind direction, and Δ is defined in equation (3.68).

The mass accretion rate is then found to be

$$\dot{M} = \frac{4\pi(r_+^2 + a^2)n_\infty}{\sqrt{1 - v_\infty^2}}, \quad (7.21)$$

and the stagnation point of the fluid is at

$$r = M \left[1 + \sqrt{1 + \frac{4}{v_\infty}} \right] \quad (7.22)$$

for the case of zero spin.

Even more remarkably, a similar solution for the (charged) Kerr-Newman black hole was presented by Babichev *et al.* [10]. We do not follow this extension here, but note that that reference makes the definition of the basic fluid variables in terms of the potential somewhat clearer.

This solution is not quite as we would like, however, as it uses Boyer-Lindquist coordinates which are singular at the horizon. We therefore use the coordinate transformation given by (3.72) to transform the velocity to the Kerr-Schild coordinate system that we use. The transformation is implemented in the code itself as the final expressions involved are too complex for MapleTM to handle easily. This is not a problem in terms of computational expense, however, as the calculation is only performed for the initial conditions.

7.5.2 Validation results

We now present the results of applying our algorithm to the exact solution of Petrich *et al.* presented above. When taking cross sections of the solutions, we do so along the x -axis, since we have the wind coming in from $x = -\infty$ by default. We do not plot actual values of the solution, however, but interpolate 800 equally spaced points between $x = \pm 50M$. Therefore, our plots show the correct shape of our evolved solution, but not exact numerical values. We note that for the low resolution grid (see Table 7.2), $100M/\Delta r_{\min} \approx 1700$, so this is certainly not too many points to use for the interpolation, at least near the horizon.

Evolving the solution of Petrich *et al.* will test all aspects of our algorithm, as it requires the capturing of effects that do not have spherical or Cartesian symmetry, so that all flux components are tested, along with our implementation of evolution on curvilinear grids. The suitability of both the inner and outer boundary conditions will also be tested.

Unless otherwise stated, the following tests are all carried out using a low resolution grid (see Table 7.2) with an inner boundary at $r = M$ and an outer boundary at $r = 50M$. We use a fluid velocity at infinity of $v_\infty = 0.6$, so as to compare to Petrich *et al.*'s plot.

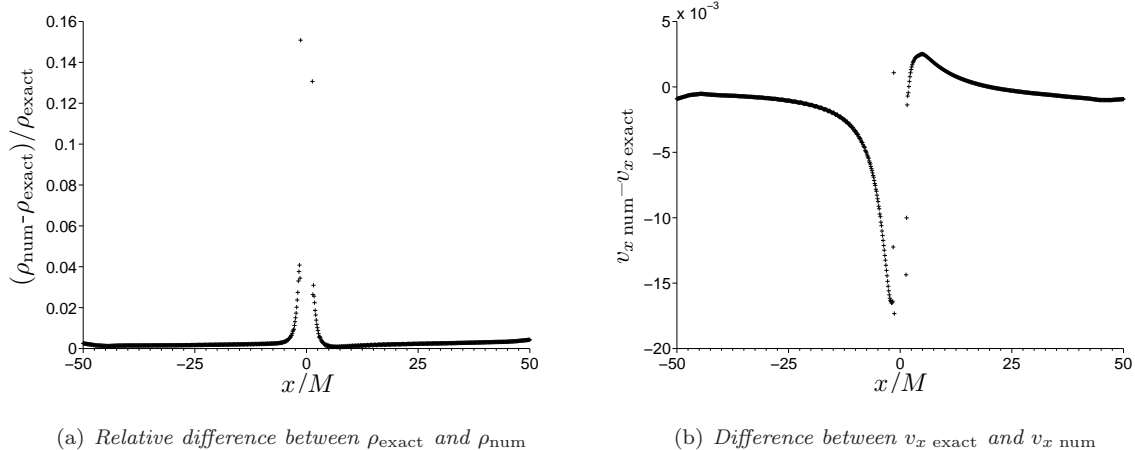


Figure 7.3: The exact solution of Petrich et al. [83] is used as initial data for the simulation and evolved to time $t = 200M$. We plot the difference between the exact solution and the evolved solution at this time. We see that the solutions are very close, except near to the excision boundary.

Our first test of the algorithm is to begin with the exact solution and to evolve it for some time. In Figure 7.3 we show a comparison of the simulated solution with the exact solution.

The density is within about 0.2% of the exact solution outside $r = 5M$ and within 2% outside the horizon at $r = 2M$. The x -velocity is everywhere within about 0.015 of the exact solution. We also compare the mass accretion rates for the evolved exact solution with the analytic value, and this is shown in Figure 7.4. We see that the accretion rate stabilises slightly above the exact solution. However, the limiting value is within 1.2% of the exact solution.

We now perform the same evolution with the constant density and velocity initial data that we intend to use for later problems. A comparison of the exact and numerical solutions is shown in Figure 7.5. The differences between the exact solution and the numerical solution is shown in Figure 7.6. We see that although the solutions appear fairly close except near the outer boundaries, the relative difference between the exact and numerical solutions are somewhat larger than those from evolving the exact solution as initial data.

The mass accretion rate, shown in Figure 7.7, stabilises to a constant limit after a time of about $130M$. The limit is again slightly in excess of the analytic value, exceeding it by about 4.3%.

We have repeated these plots for different resolutions with results as in Table 7.3, and we observe that the numerical solution does not converge to the analytic solution. We have eliminated some possibilities for this discrepancy as follows:

1. Generating the exact solution using our numerical code and then calculating the mass accretion rate from that yields a value within $10^{-4}\%$ of the exact value. The low resolution of the grid is therefore not an issue, and we are calculating the mass accretion rate correctly.

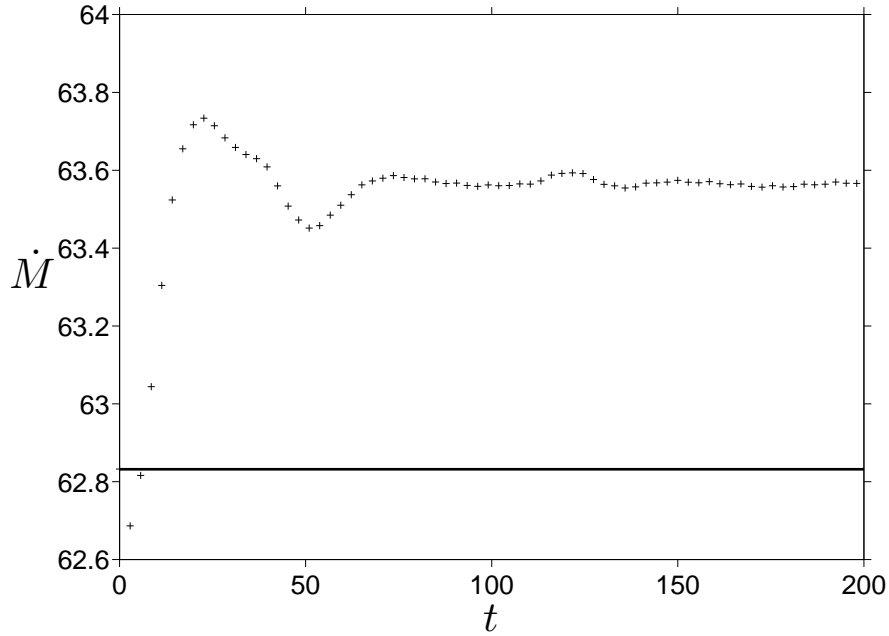
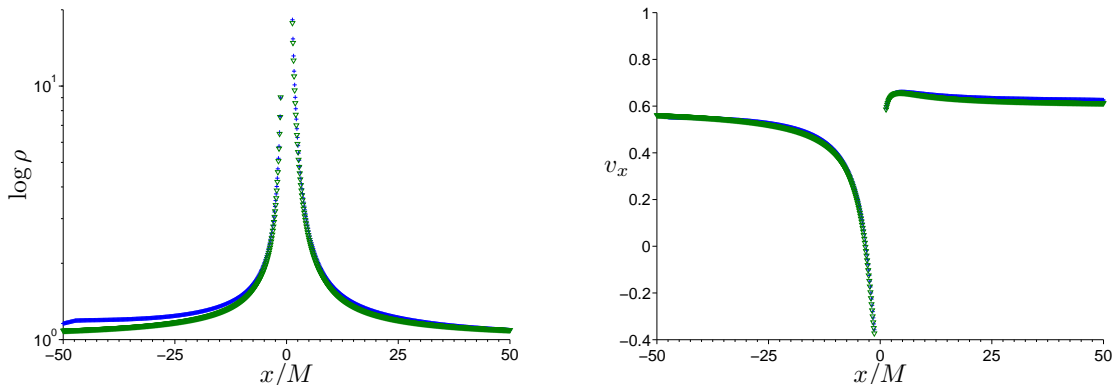


Figure 7.4: Mass accretion rate resulting from evolving Petrich et al.’s exact solution to time $t = 200M$, evaluated at radius $r = 5M$. The solid line shows the exact solution, and the crosses the numerical solution.



(a) Comparison of the numerical and exact solutions for $\log \rho$. (b) Comparison of the numerical and exact solutions for v_x .

Figure 7.5: Comparison of the numerical and exact solutions for $\log \rho$ and v_x . The numerical solution is given by the blue +s and the exact solution is given by the green ∇ s.

2. Varying the radius at which we calculate the mass accretion rate allows for some variation, as shown in Table 7.3. However, they do not vary sufficiently to encompass the exact solution.
3. Varying the outer radius of the grid does not alter the limiting value significantly.

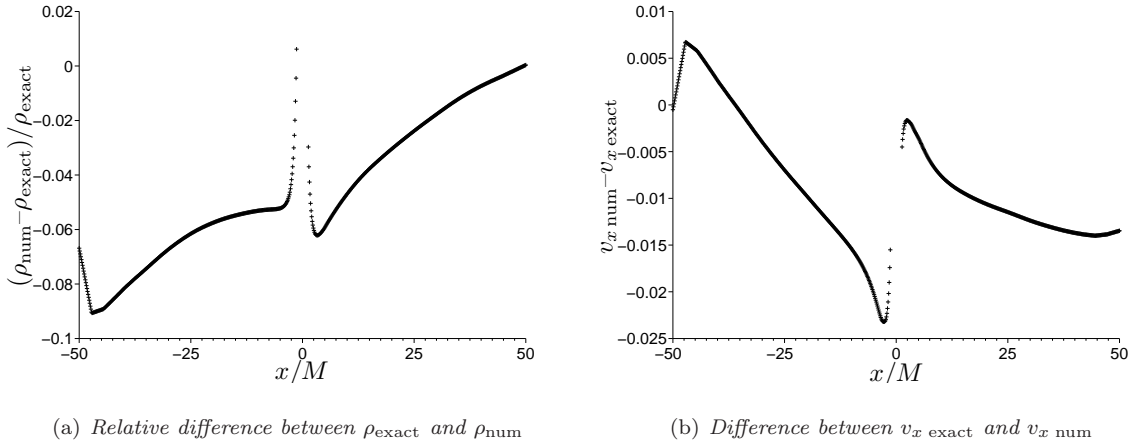


Figure 7.6: A stiff fluid with an initial state of $\rho = 1$ and $\sqrt{v_i v^i} = 0.6$ is evolved to time $t = 197M$. We plot the difference between the exact and evolved solutions at this time.

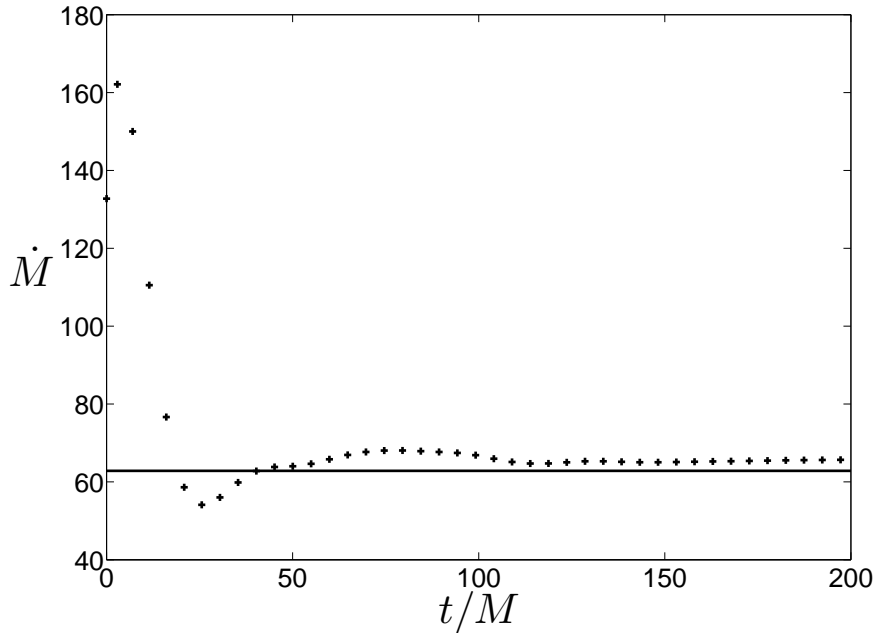


Figure 7.7: Mass accretion rate resulting from evolving constant initial conditions to time $t = 200M$, evaluated at radius $r = 5M$. The solid line shows the exact solution, and the +s the numerical solution.

Further, plotting $\sqrt{v^2}/u^0$ in Boyer-Lindquist coordinates for a high resolution run gives a plot visually very close to that given in [83], as shown in Figure 7.8.

Applying our code to a stiff fluid accretion onto a black hole with spin $a = 0.5$ results in the mass accretion rate shown in Figure 7.9. We again see that the limiting value resulting from the evolved solution is slightly in excess of the analytic solution, this time by about 4.1%.

Resolution	Radius	$\dot{M}_{\text{num}}/\dot{M}_{\text{exact}}$
$n = 35$	2.50	1.029
	3.00	1.033
	3.80	1.037
	5.00	1.043
	6.00	1.043
$n = 50$	2.50	1.028
	3.00	1.031
	3.80	1.033
	5.00	1.036
	6.00	1.037
$n = 60$	2.50	1.028
	3.00	1.030
	3.80	1.032
	5.00	1.034
	6.00	1.035

Table 7.3: *Final steady state accretion rates for various resolutions and extraction radii. The mass accretion rates are calculated at a time $T = 200M$, where the solution has settled down to a steady state for all simulations.*

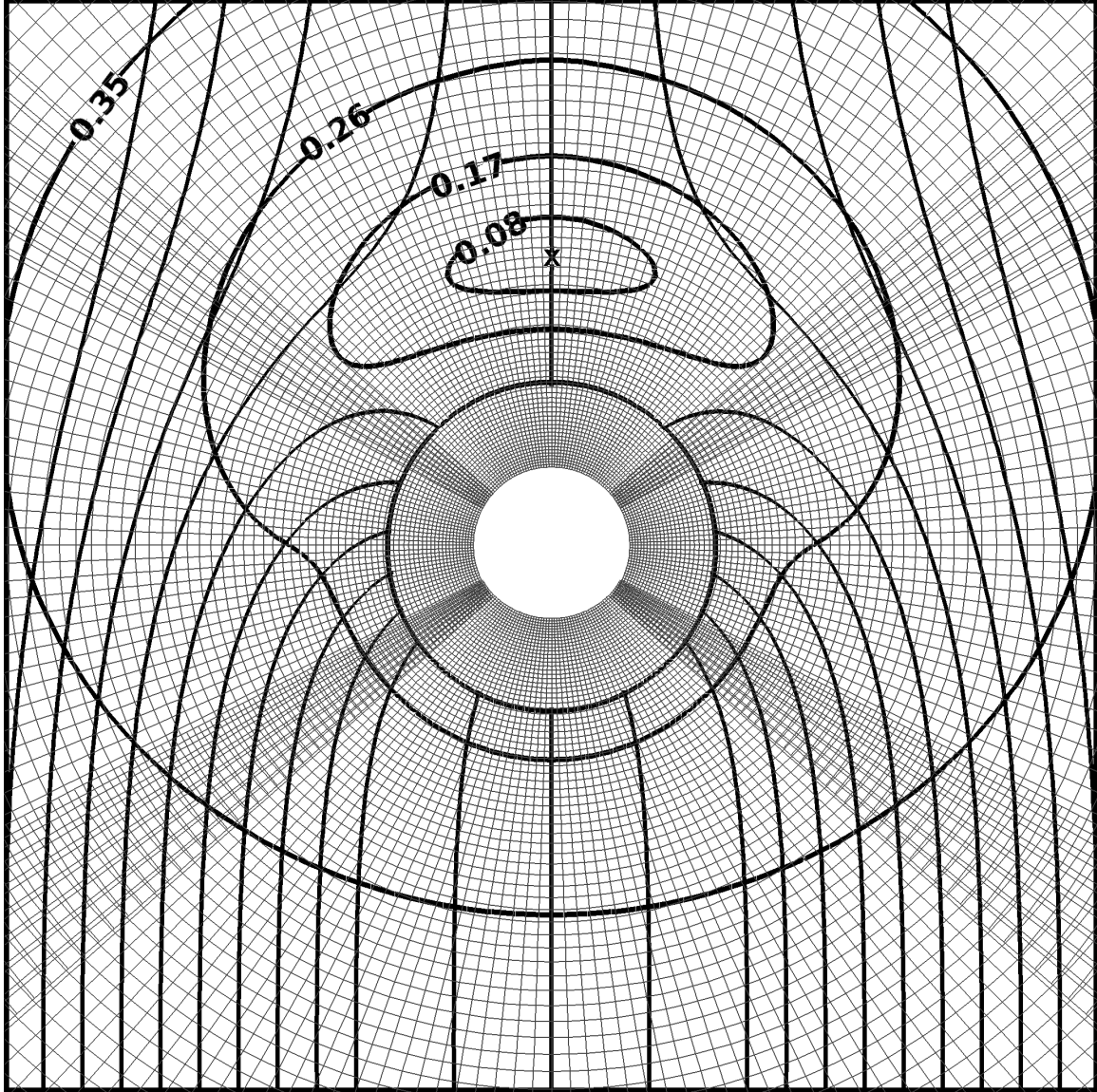


Figure 7.8: *Contours of $\sqrt{v^2}/u^0$ in Boyer-Lindquist coordinates of the accretion of a stiff fluid onto a non-rotating black hole with $v_\infty = 0.6$. The high resolution grid was used (with inner radius $r = M$), and the simulation ran to a time $t = 400M$. The contour levels are at 0.08, 0.17, 0.26, 0.35, with figure limits at $\pm 7M$, the same as in the plot in [83]. The stream-lines are calculated using velocities in Boyer-Lindquist coordinates, which have a coordinate singularity at the horizon, which is why the streamlines do not continue to the excision boundary. The grid lines for the $z = 0$ plane are shown, and the regions of overlap can be seen.*

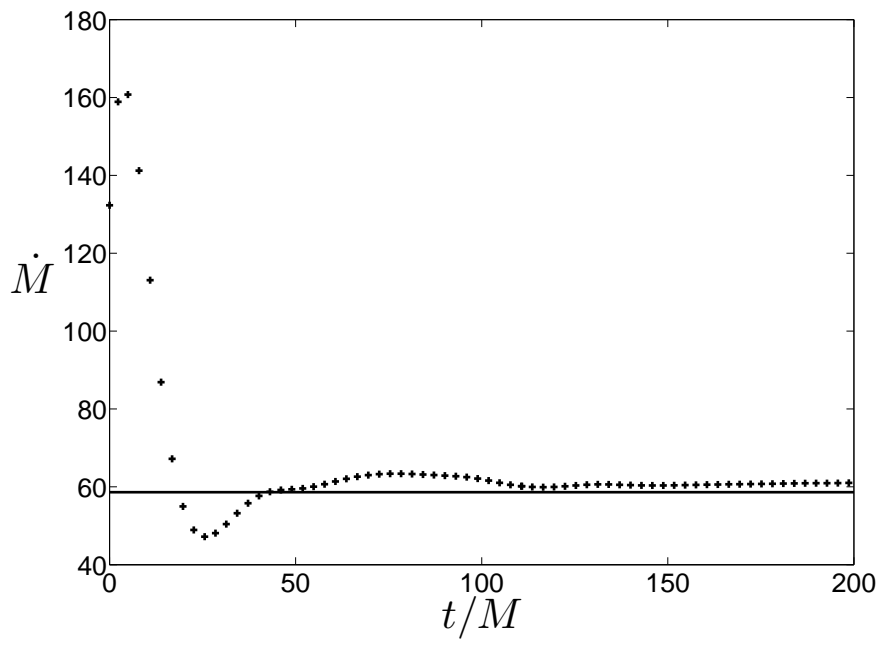


Figure 7.9: Variation in time of the mass accretion rate of a stiff fluid with velocity $v_\infty = 0.6$ onto a Kerr black hole with spin $a = 0.5$. The analytic solution as derived by Petrich et al. is shown by the line, and the numerical solution is shown by the +s.

7.5.3 Conclusions

We have demonstrated that our algorithm is capable of reproducing the exact solution of Petrich *et al.* to a good accuracy. The fact that the solution converges to a steady state shows that the appropriate quantities are conserved. It is clear that our outer boundary conditions, although only defined at a finite radius, are sufficiently far away not to affect the solution adversely. Further, the results have been obtained at a relatively low resolution, which points to the overall accuracy of our method.

Although the mass accretion rate converges with increased resolution, it does not converge to the exact solution. We suspect that the reason is concerned with the implementation of the outer boundary conditions for a subsonic flow.

Therefore, we conclude that our numerical methods are robust and accurate, and that the results and conclusions we present in the next chapter are therefore valid.

Chapter 8

Numerical study of Bondi-Hoyle-Lyttleton Accretion

In this chapter we present the results of the simulations of Bondi-Hoyle-Lyttleton accretion that we have performed.

In order to understand the general properties of the accretion flow, we first show the results of some simulations also performed by Font and Ibáñez, but now including the addition of some effects that we can investigate using our newly implemented and validated algorithm. We then select one of these models and perform a full parametric study on it, exploring some new effects now open to exploration due to the features of our implemented algorithm.

8.1 Base model parameters

In order to comprehend the range of flow patterns covered by Bondi-Hoyle-Lyttleton accretion, we make use of some models suggested by Font and Ibáñez [41]. We present the parameters for these in Table 8.1. For a more direct comparison with [41], we have scaled the mass accretion rate in the same way as Font and Ibáñez, by a factor

$$\frac{4\pi\lambda_s M^2 \rho_\infty}{(v_\infty^2 + c_\infty^2)^{3/2}}, \quad (8.1)$$

where λ_s was defined in equation (7.18).

In Figure 8.1 we show the results for running the UB1 model to time $t = 400M$ on a medium resolution grid (see Table 7.2). We have plotted the same contour levels and used the same figure limits as Font and Ibáñez [41]. The main difference is that our results are shown at a later time than theirs. However, since we are evolving to steady state, this should not be important. For the velocity plot, we have transformed

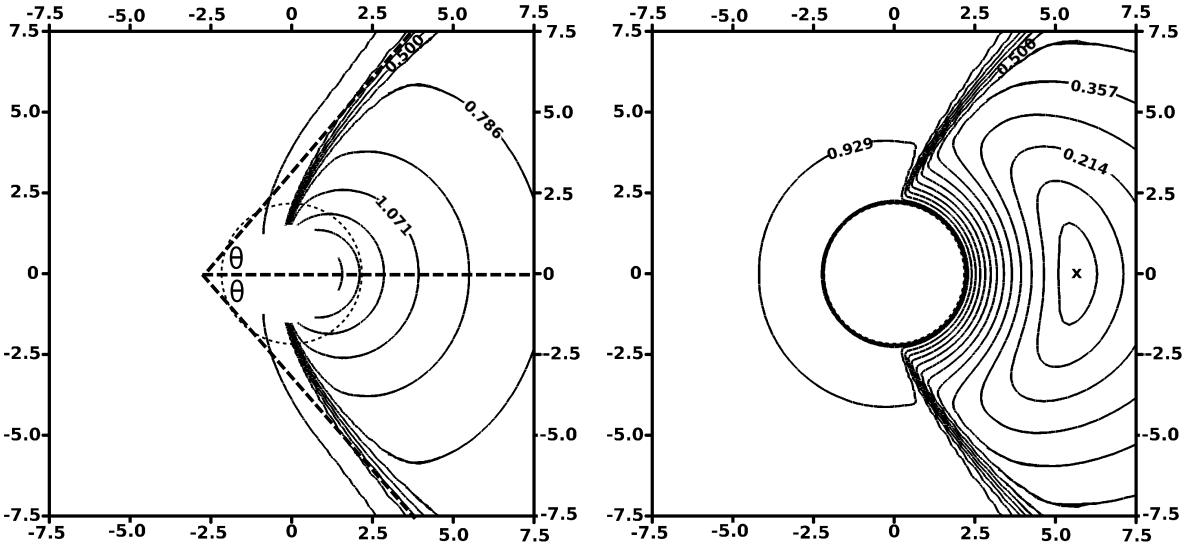
Model	c_∞	Γ	\mathcal{M}_∞	v_∞
UB0	0.570	4/3	0.6	0.342
UC0	0.810	5/3	0.6	0.486
UA1	0.310	1.1	1.5	0.465
UB1	0.570	4/3	1.5	0.855

Table 8.1: *Test model parameters, as taken from Font and Ibáñez [41]. We emphasise that the values for the sound speed are exact.*

into Boyer-Lindquist coordinates.

The plot of density shows that we have found higher densities than those of Font and Ibáñez. The velocity plot shows the stagnation point as being further downstream than that found by Font and Ibáñez, but the contours look otherwise the same. In Figure 8.2 we plot the mass accretion rate as a function of time. As expected, it tends to a constant limit since we have reached a steady state. Evaluating the mass accretion rate at a smaller radius does not affect the limiting value, either, again showing that the flow is at a steady state.

However, the final rate is somewhat higher than the final (scaled) mass accretion rate found in [41], which suggested it to be $\dot{M} = 27.0$. The difference in the evolution of the mass accretion rates (ours shows a peak at time $t \approx 10M$, while [41] shows a monotonic increase to the final limit) is explained by the difference in coordinate systems, and hence initial conditions, as we specified initial velocity to be constant in Kerr-Schild coordinates, as opposed to Font and Ibáñez’s use of Boyer-Lindquist coordinates.



(a) Plot of $\log_{10}(\rho)$. The dashed lines indicate the shock- (b) Plot of $\sqrt{v^2}$, evaluated in Boyer-Lindquist coordinates, cone angle, θ , which for this case is the same above and so that the contours cannot be calculated inside the horizon. The stagnation point is marked.

Figure 8.1: Steady-state fluid for model UB1 with uniform flow and a non-spinning black hole at time $t = 400M$. The medium resolution grid was used, and the axes are scaled in terms of the accretion radius $r_a = 0.92M$.

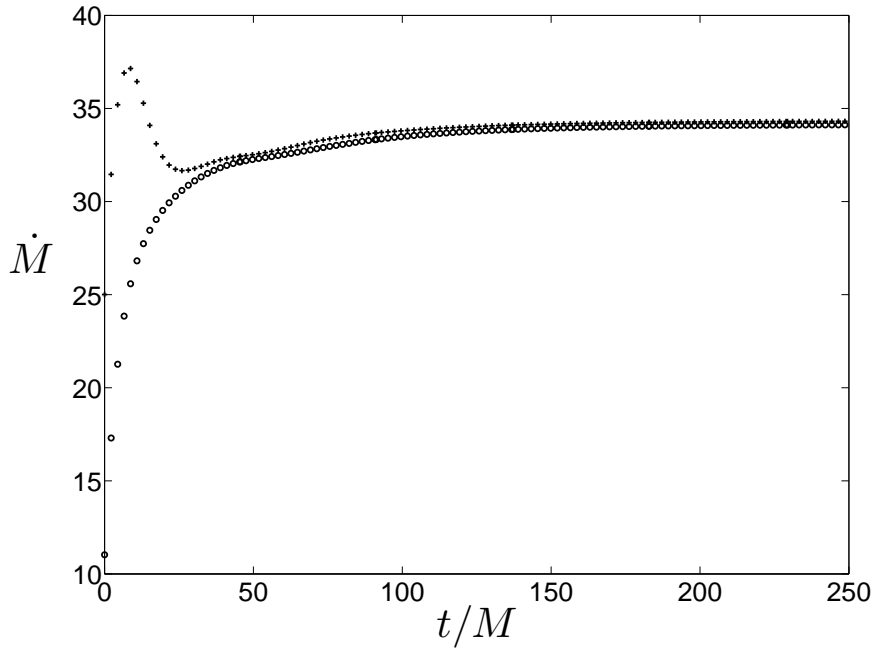


Figure 8.2: Mass accretion rate for the UB1 model evaluated at $r = 2.5M$ (circles) and $r = 5M$ (+ signs).

8.2 General flow morphology

We examine the effect of varying the adiabatic index, Γ , and the flow velocity, using some models suggested by Font and Ibáñez, as given in Table 8.1. Qualitatively, our results are the same as those of Font and Ibáñez, although there are some differences in the numerical values that we find.

In order to have our numerical solution converge to steady state in a reasonable time, we have chosen all our models to be ultra-relativistic, meaning that the sound speed in the fluid upstream of the black hole is close to its theoretical maximum of $\sqrt{\Gamma - 1}$. The effect of the black hole can therefore propagate throughout the computational domain in a fairly short period of time, of the order of a few hundred M . We use the medium resolution grid whose parameters are given in Table 7.2.

As well as testing the base models, we also test variations on these models with a spinning black hole and a density perturbation. We use a spin of $a = 0.9$, with the spin-axis along the z -axis, and a density perturbation of $\epsilon_\rho = 0.2$, perturbing along the y -axis (i.e. $\mathbf{p} = \mathbf{e}_y$ in (7.6)). The fluid direction is along the x -axis in all cases.

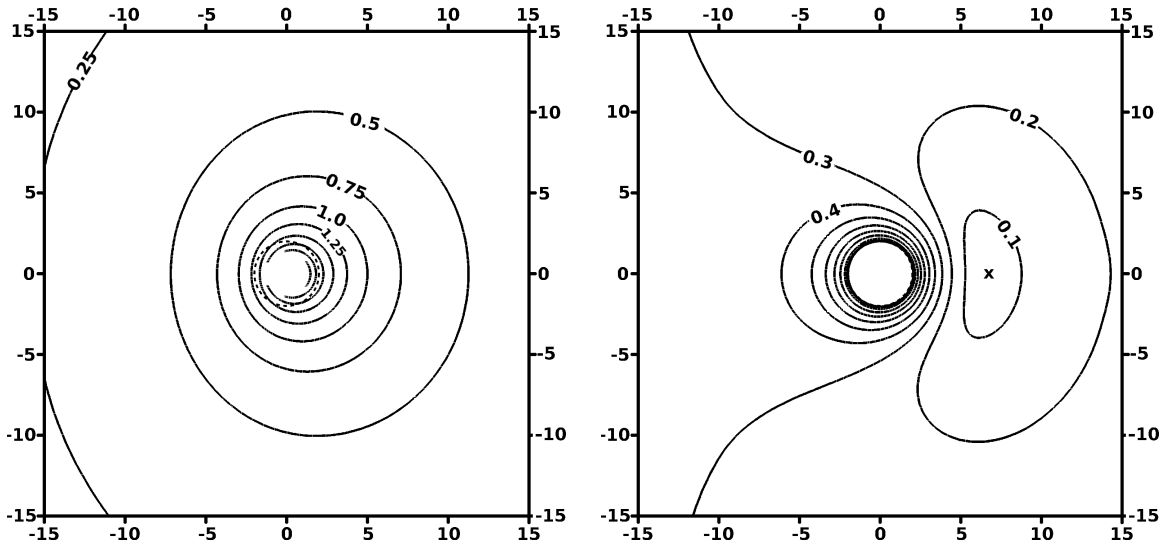
The DVD accompanying this thesis includes, for each model examined here, the final steady-state of the fluid in terms of its density, velocity, and streamlines of the flow. More details about these plots can be found in the main results section, §8.4.

8.2.1 Features common to all models

Upstream of the black hole, the fluid motion is mostly undisturbed by the presence of the hole, although the fluid is always attracted towards the singularity. Some fluid will fall directly into the black hole. Some fluid will only be slowed by the gravity of the black hole, but be moving fast enough to go past it, before finally being turned around to fall into the hole from the downwind direction. The rest of the fluid will be affected by the hole, but will ultimately escape to infinity, downstream of the hole. There will therefore be a stagnation point, where the fluid has zero velocity, downstream of the black hole, which denotes a critical point between fluid which ultimately falls into the black hole, and fluid which escapes. As a result of the fluid which initially passes the black hole before falling in, there is always a region of high density on the downstream portion of the horizon.

Extending the idea of the stagnation point, we can also identify a region at a large distance upstream of the hole, inside which fluid will eventually fall onto the hole, and outside which it will not do so. In the case of symmetrical Bondi-Hoyle-Lyttleton accretion, this region always has a circular cross-section, and is therefore referred to as the *accretion cylinder*. We shall see that our variations cause the cross-section to deviate from the circular, and so we refer to it instead as the *accretion region*.

When considering fluid velocities, we always use Boyer-Lindquist coordinates (via the transformation



(a) Equally spaced contours of $\log \rho$.

(b) Velocity contours evaluated in Boyer-Lindquist coordinates. The stagnation point is marked.

Figure 8.3: Contour plots of density and velocity for a uniform subsonic flow past a non-spinning black hole with flow parameters given by model UB0. The plots are evaluated at time $t = 400M$.

given in (3.72)), which have a coordinate singularity at the horizon. In these coordinates, therefore, the fluid velocity approaches the speed of light at the horizon, although this is not the case in Kerr-Schild coordinates.

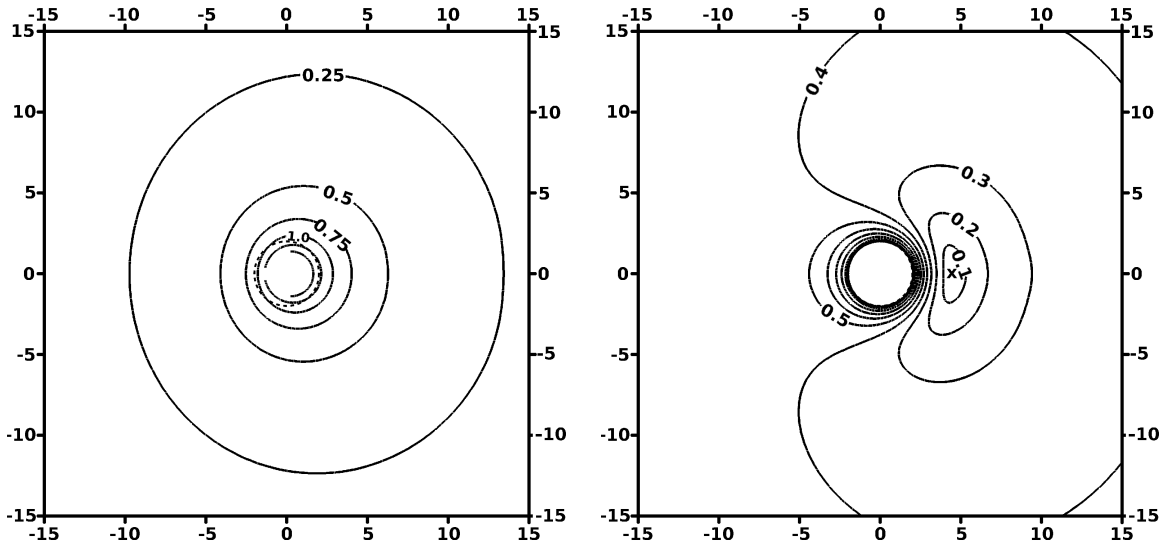
We now split our discussion into two cases, those of subsonic and supersonic flow.

8.2.2 Subsonic models

The subsonic models that we have tested are UC0 and UB0 (see Table 8.1), with a Mach number of $\mathcal{M} = 0.6$. These models differ primarily in their adiabatic indices, although the fluid velocity has to be changed to allow for the maximum velocity permitted by $c_s < \sqrt{\Gamma - 1}$. We run the simulations to time $t = 400M$, and the results are shown in Figures 8.3 and 8.4.

For both models we see that the density contours near the hole are nearly circular, but offset slightly in the downstream direction relative to the centre of the hole so that the density at the horizon is higher on the downstream side. Model UC0, with the higher adiabatic index, gives a lower fluid density close to the horizon than does model UB0, since the fluid is less compressible for higher adiabatic index.

The fluid velocity contour plots are smooth, showing the features we would expect, namely, that the fluid falls in with increasing velocity, and that there is a stagnation point downstream. The velocity at a given distance upstream of the singularity is higher for model UC0, and the velocity at a given distance



(a) Equally spaced contours of $\log \rho$.

(b) Velocity contours evaluated in Boyer-Lindquist coordinates. The stagnation point is marked.

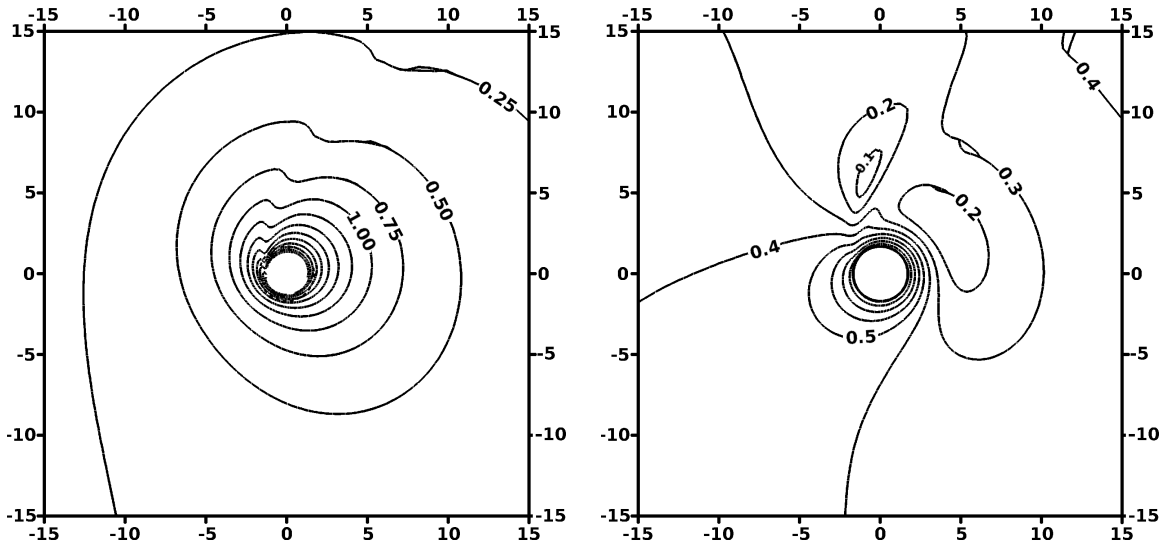
Figure 8.4: Contour plots of density and velocity for a uniform subsonic flow past a non-spinning black hole with flow parameters given by model UC0. The plots are evaluated at time $t = 400M$, and the axes are in units of M .

downstream of the hole is higher for model UB0, the stagnation point for which lies further downstream than for model UC0.

In Figures 8.5 and 8.7 we show the effect on these models of including spin $a = 0.9$ and density perturbation $\epsilon_\rho = 0.2$. The flow is still in the equatorial plane of the black hole, however. In both cases the flow is wrapped around the horizon significantly, in the direction of the spin. The density contours now have slight indentations, showing that there is a line of slightly lower density fluid. This is more evident for model UB0. Examination of cross-sections for other values of z , such as $z = 1$ (see Figure 8.6), shows that the indentation is more prominent to either side of the $z = 0$ plane.

The velocity contours show the development of *two* regions of lower velocity flow. From the form of the ellipse-shaped contours extending into the lower-left of the plots, it seems that the main flow morphology has been twisted around the horizon by about 45° , although there are now no stagnation points, due to the added effect of the fluid rotating around the hole before falling in. Examining iso-surfaces of the velocity shows that neither are there are any stagnation points away from the $z = 0$ plane.

These flow features arise because the spin of the black hole tends to pull fluid in the lower half of the plot around in the anti-clockwise direction, while fluid in the upper half of the plot is decelerated in the same (angular) direction. The fluid from the lower half therefore moves further anti-clockwise around the black hole than it would otherwise have done before it is decelerated sufficiently to fall into the black



(a) Equally spaced contours of $\log \rho$.

(b) Velocity contours evaluated in Boyer-Lindquist coordinates.

Figure 8.5: Contour plots of density and velocity for a subsonic flow with density perturbed by $\epsilon_\rho = 0.2$ past a black hole with spin $a = 0.9$ and flow parameters given by model UB0. The plots are evaluated at time $t = 400M$, and the axes are in units of M .

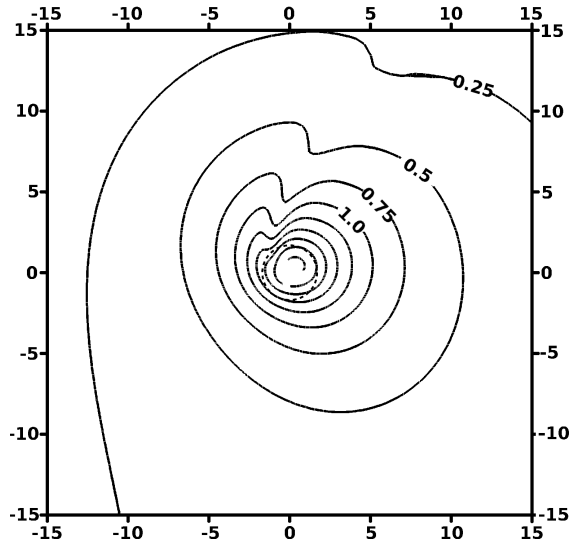
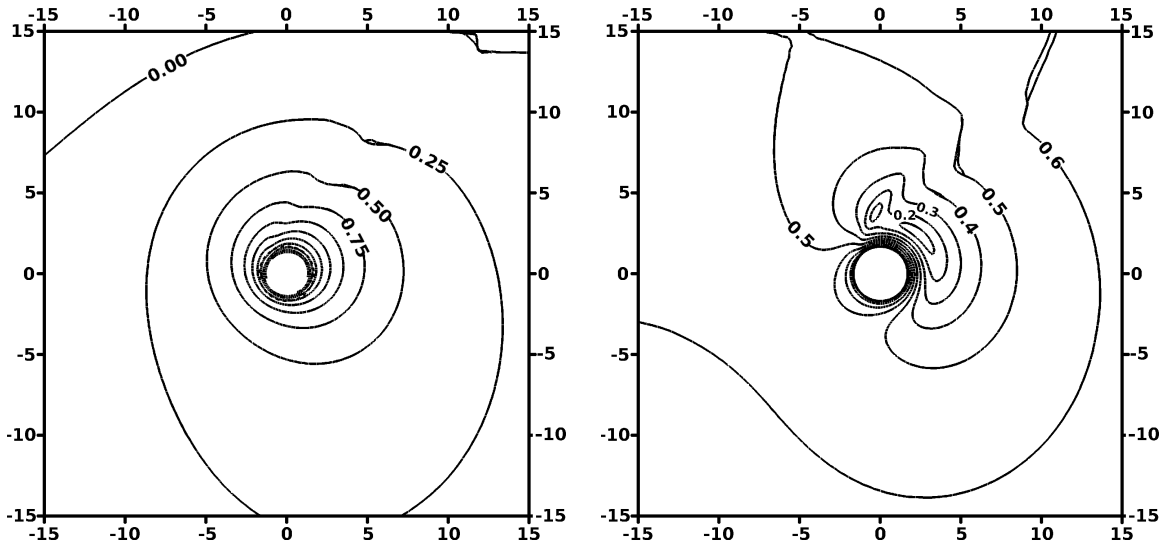


Figure 8.6: Density as in Figure 8.5, now evaluated on $z = 1$ plane.

hole. This accounts for the change in the locations of the regions of low velocity.

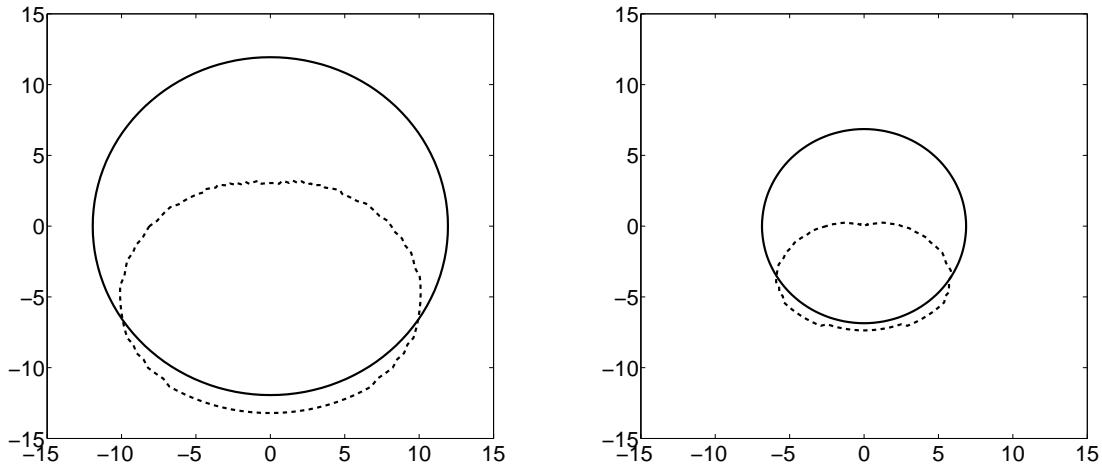
The density perturbation gives a higher density towards the bottom of the plot (on the left of the figure, upstream of the black hole). The effect of the perturbation is similar to that caused by the spin in



(a) Equally spaced contours of $\log \rho$.

(b) Velocity contours evaluated in Boyer-Lindquist coordinates.

Figure 8.7: Contour plots of density and velocity for a subsonic flow with density perturbed by $\epsilon_\rho = 0.2$ past a black hole with spin $a = 0.9$ and flow parameters given by model UC0. The plots are evaluated at time $t = 400M$, and the axes are in units of M .



(a) Model UB0

(b) Model UC0

Figure 8.8: Accretion regions for all four subsonic models. The solid line shows the accretion region for $a = 0$, $\epsilon_\rho = 0$, and the dashed line shows the accretion region for $a = 0.9$, $\epsilon_\rho = 0.2$.

that, due to the lower momentum of the fluid at the top of the plot (being less dense), the fluid coming from the lower part of the plot (being more dense) can travel further around the black hole before being

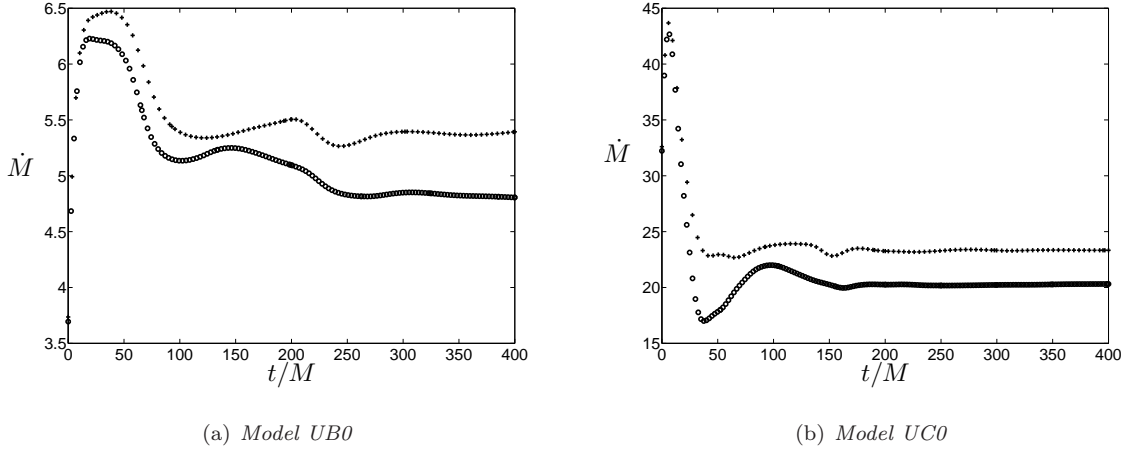


Figure 8.9: Variation of mass accretion rate with time for two subsonic models. In both plots the + signs correspond to the uniform, non-spinning case, and the circles correspond to the perturbed, $\epsilon_\rho = 0.2$, spinning, $a = 0.9$ case. The accretion rates are evaluated on a sphere of radius $5M$.

decelerated sufficiently to fall into it.

The cross-sections of the accretion regions upstream of the black hole are shown in Figure 8.8. As expected by symmetry, the accretion regions for the unperturbed, non-spinning cases are circular. The plots also show that the combined effect of the spin and density perturbation is to deflect fluid from the upper half of the domain away from the black hole so that it does not accrete onto it. We also see that the accretion regions for the higher adiabatic index (model UC0) are somewhat smaller than those for model UB0.

We note that the tilted form of the flow and the development of a line of lower density are features that also appear in Ruffert’s simulations involving a density perturbed fluid onto a compact object [95]. This suggests that the major changes to the flow features are the result of the density perturbation, rather than the spin of the black hole, as such an effect was not included in Ruffert’s simulations, not being relevant in the Newtonian regime.

In Figure 8.9 we show the mass accretion rates for the four subsonic models we have presented. We see that although all the rates have reached a nearly constant value, not showing any short-period oscillations, the rates for model UB0 are slowly decaying. However, the fact that the rates are very nearly constant shows that we have attained steady-state flow, thus validating our numerical methods.

In both cases we see that the joint effect of the spinning black hole and the density perturbation is to lower the rate at which mass is accreted onto the black hole. This is as suggested by the reduction in the size of the accretion regions. The reasons for this will be discussed in the full parametric study in §8.4.

Note that the mass accretion rates for the two models cannot be compared directly, as they have already been scaled by the factor given in equation (8.1).

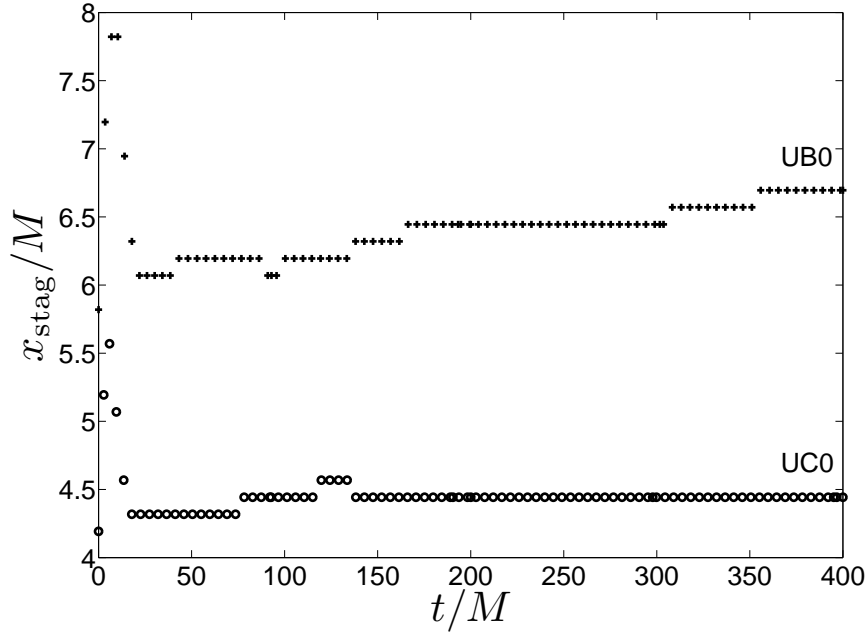


Figure 8.10: Evolution of stagnation point locations with time for the subsonic models UB0 (+ signs) and UC0 (circles).

In Figure 8.10 we show the evolution of the stagnation point for the two non-spinning, non-perturbed models we have presented. We see that they both reach a constant state by the end of the evolution. These plots are derived from interpolating the total velocity along the x -axis (the direction of the wind in this case) at equally spaced points, and finding the location of the minimum. This accounts for the discrete nature of the points at which the stagnation point is located.

The streamlines for the perturbed model UB0 with spinning black hole are shown in Figure 8.11. Overall, these show the flow being diverted from its original direction by the presence of the black hole. However, at the base of the main outgoing tracer pattern, we see a concave feature, which suggests that the fluid is forming two vortices. We shall see this feature developed more strongly in supersonic flows.

The streamlines for the perturbed and spinning UC0 model are shown in Figure 8.12. These essentially show only that the flow has been diverted from its original direction by the black hole. However, there is a small feature at the bottom of the main circle of tracers. This may well be an undeveloped version of the vortex structure previously referred to.

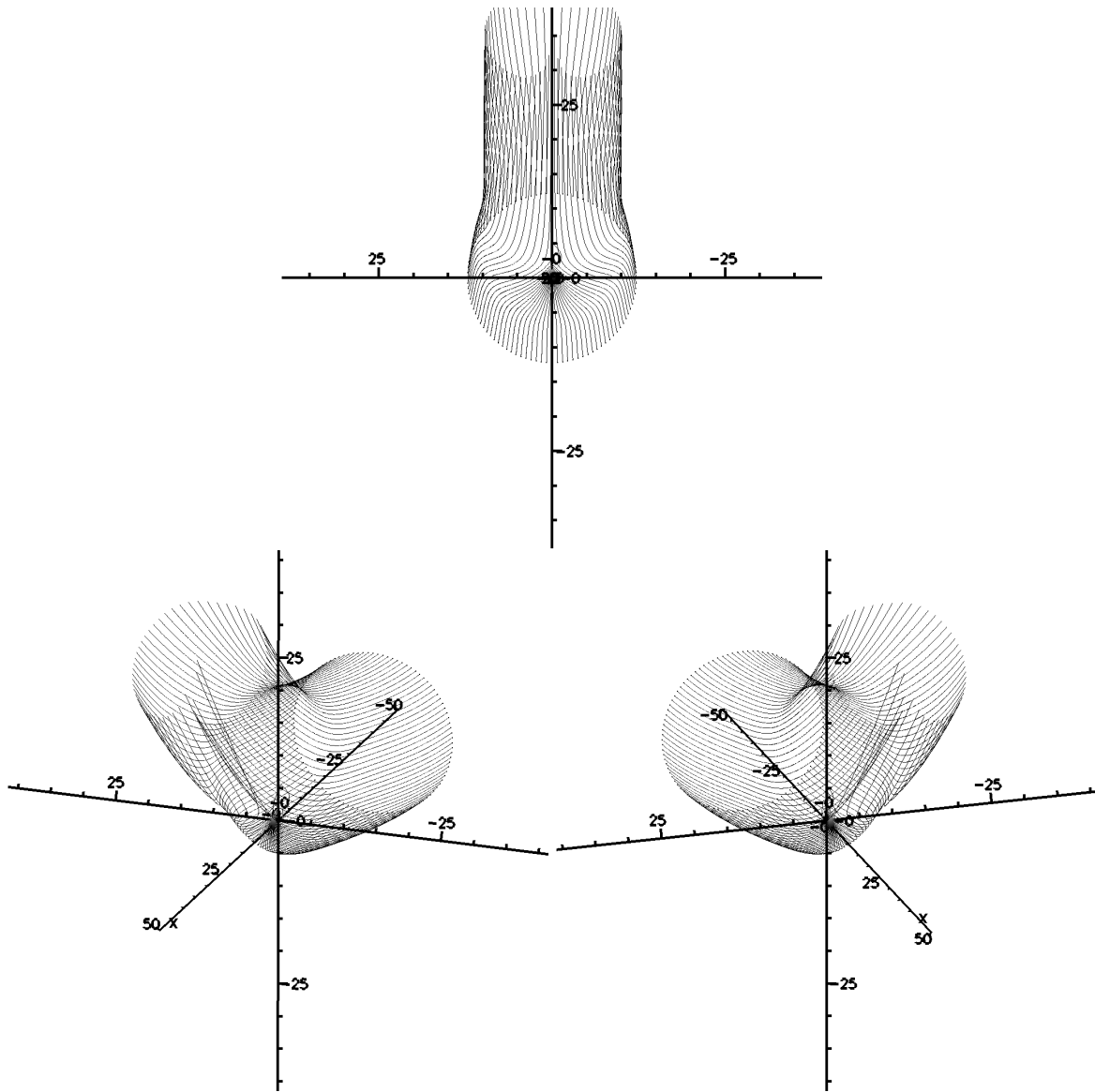


Figure 8.11: Plot of streamlines for a perturbed flow with $\epsilon_\rho = 0.2$, model UB0, parallel to the equatorial plane, onto a Kerr black hole with spin $a = 0.9$.

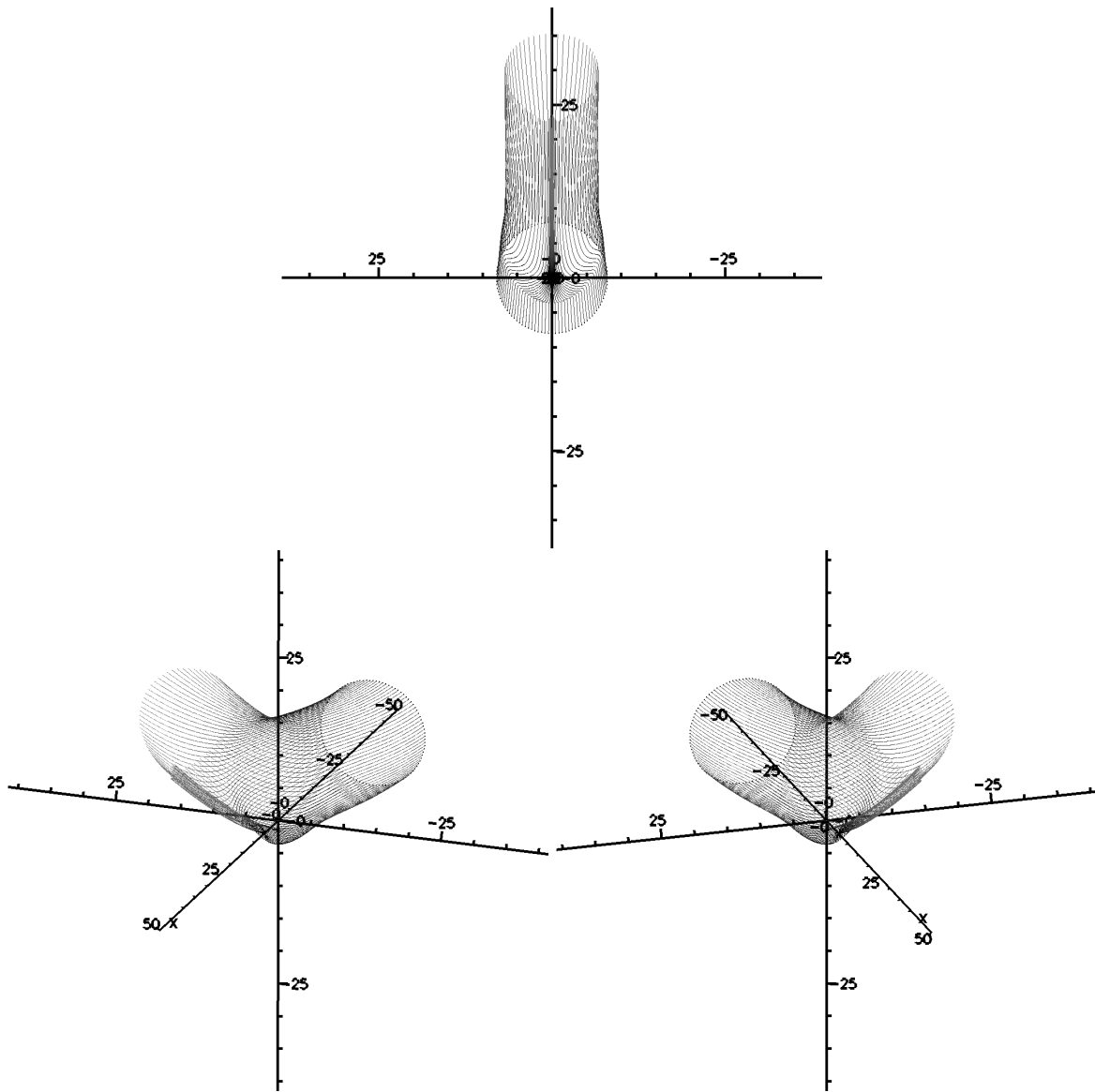
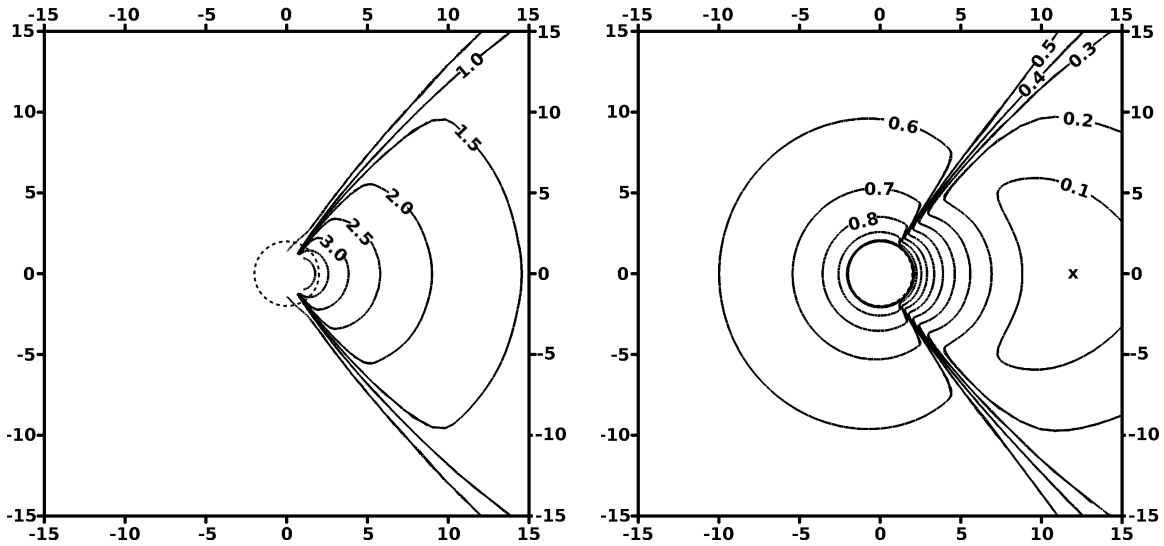


Figure 8.12: Plot of streamlines for a perturbed flow with $\epsilon_\rho = 0.2$, model UC0, parallel to the equatorial plane, onto a Kerr black hole with spin $a = 0.9$.



(a) Equally spaced contours of $\log \rho$.

(b) Velocity contours evaluated in Boyer-Lindquist coordinates. The stagnation point is marked.

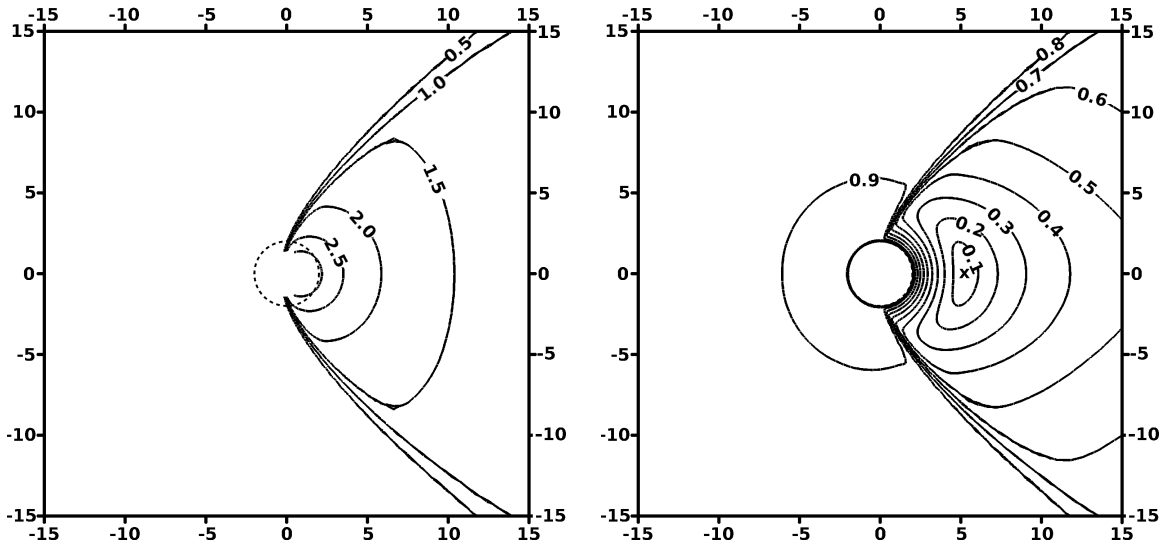
Figure 8.13: Contour plots of density and velocity for a uniform supersonic flow past a non-spinning black hole with flow parameters given by model UA1. The plots are evaluated at time $t = 300M$.

8.2.3 Supersonic models

The supersonic models we have tested are UA1 and UB1, at Mach number $\mathcal{M} = 1.5$, and the results for the density and velocity are shown in Figures 8.13 and 8.14. The main feature of these plots is the shock-cone that has formed downstream of the black hole, in a manner similar to that of a solid body immersed in an Eulerian flow in flat space-time. Analogously to that case, the opening angle of the shock-cone at a large distance from the hole should be $\sin^{-1}(1/\mathcal{M})$, with higher Mach numbers corresponding to a narrower shock-cone (see Appendix B in [82] and references therein for more details).

For a particularly stiff fluid, corresponding to a high adiabatic index, the shock-cone can detach from the black hole and begin upstream of it. However, we do not show any models with this feature here, although some examples can be seen in Font and Ibáñez [41].

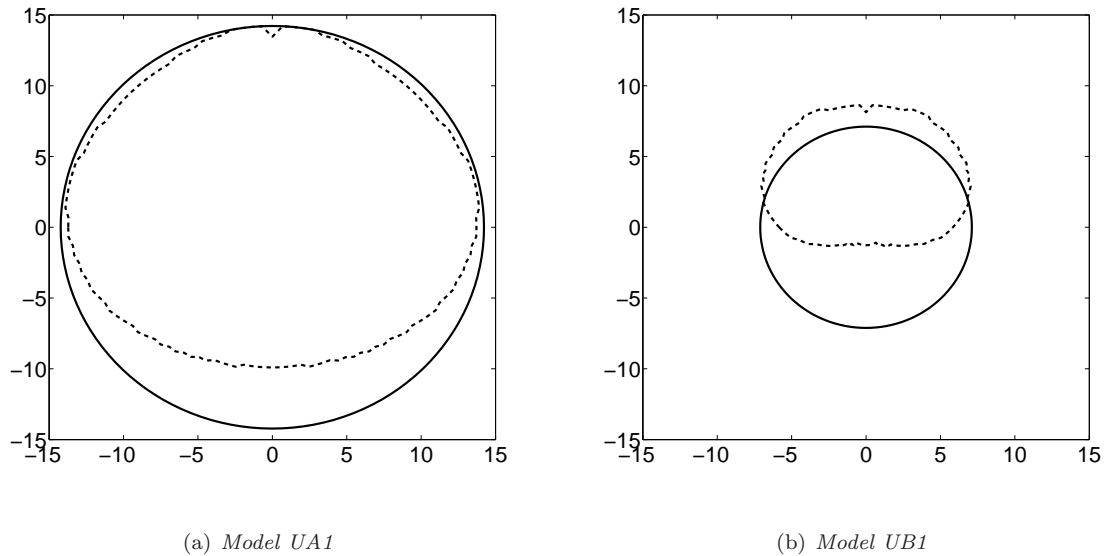
In both cases, the density of the fluid increases sharply across the shock-cone, and also increases as the horizon is approached, both from the downstream direction and from the upstream direction. We now have a considerably larger ratio between the density on the upstream and downstream sides of the horizon than for the subsonic models, with a high density inside the shock-cone as fluid is focused through it onto the horizon. The UA1 model, with the lower adiabatic index, exhibits a higher density inside the shock-cone and, in particular, at the horizon, than the UB1 model. As in the subsonic case, this is due to the fact that the fluid with the higher adiabatic index is less compressible.



(a) Equally spaced contours of $\log \rho$.

(b) Velocity contours evaluated in Boyer-Lindquist coordinates. The stagnation point is marked.

Figure 8.14: Contour plots of density and velocity for a uniform supersonic flow past a non-spinning black hole with flow parameters given by model UB1. The plots are evaluated at time $t = 300M$, and the axes are in units of M .

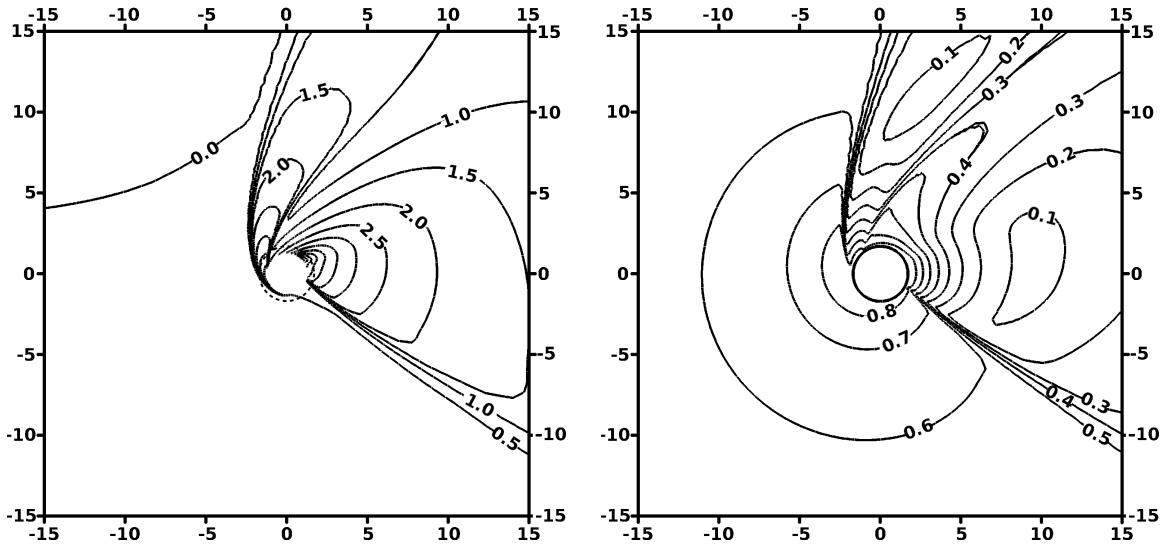


(a) Model UA1

(b) Model UB1

Figure 8.15: Accretion regions for all four supersonic models. The solid line shows the accretion region for $a = 0, \epsilon_\rho = 0$, and the dashed line shows the accretion region for $a = 0.9, \epsilon_\rho = 0.2$.

Upstream of the black hole, the velocity contours are almost circular and show the fluid increasing in velocity as it approaches the horizon. As the fluid passes through the shock, it is decelerated significantly.



(a) Equally spaced contours of $\log \rho$.

(b) Velocity contours evaluated in Boyer-Lindquist coordinates.

Figure 8.16: Contour plots of density and velocity for a supersonic flow with density perturbed by $\epsilon_\rho = 0.2$ past a black hole with spin $a = 0.9$ and flow parameters given by model UA1. The plots are evaluated at time $t = 400M$, and the axes are in units of M .

Close to the horizon inside the shock-cone, the contours are once again circular arcs, but upstream the contours distort to surround the stagnation point. The flow for model UB1 has its stagnation point closer to the black hole than that for model UA1 but upstream, outside the shock-cone, the flow velocity is lower for model UB1. Since both models have the same Mach number, $\mathcal{M} = 1.5$, we see that both shock-cones have approximately the same opening-angles.

In Figures 8.16 and 8.18 we show the effect on these models of including a spin of $a = 0.9$ and a density perturbation of $\epsilon_\rho = 0.2$. The flow is still in the equatorial plane of the black hole, however. We see that the shock-cone is pulled round the hole in the positive (anti-clockwise) direction. This effect is so significant near to the black hole that the shock-cone attaches to the *upstream* side of the horizon and far from the horizon the cone is still at an angle to the horizontal. We also see that there is a line of significantly lower density dividing the shock-cone into two sections. This is a more extreme case of the effect we noticed in the subsonic regime. Examining iso-surfaces of the density reveals that there are two regions of lower density either side of the $z = 0$ plane that extend down towards the horizon. Unsurprisingly, model UA1 still has a higher density near to the horizon than model UB1.

The velocity contour plots show similar twisting and distortion to the subsonic case. The velocity drops significantly across the shock-cone, and outside the cone, the contours retain their circular appearance. There are then two regions of low velocity. It is not clear from the numerical solution whether either of

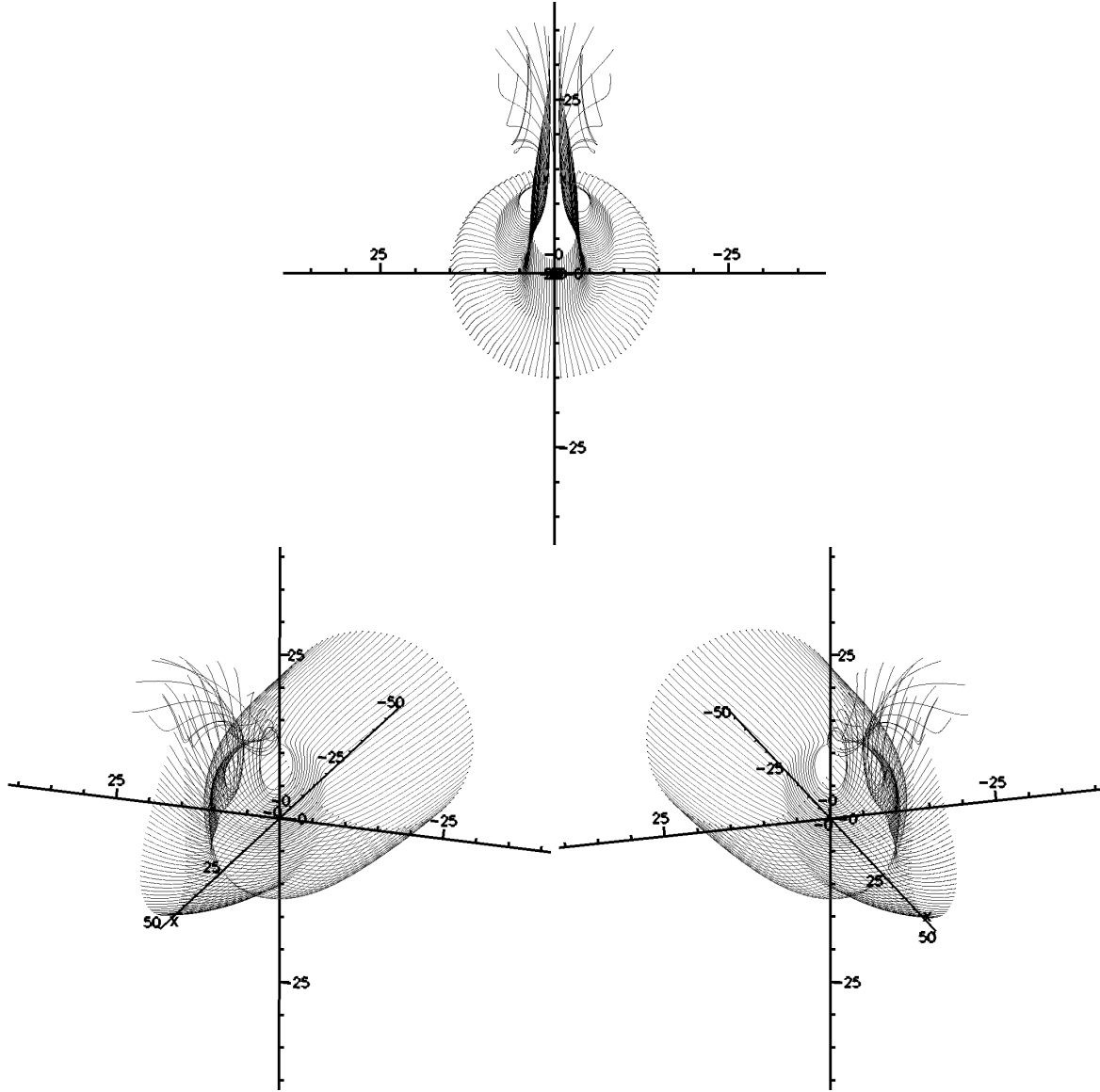
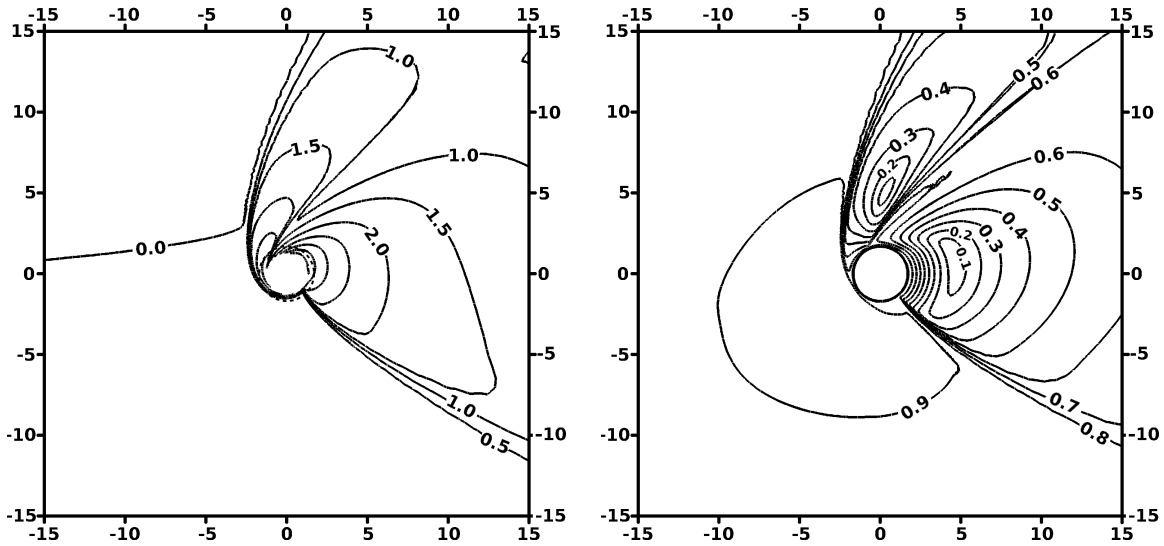


Figure 8.17: Plot of streamlines for a perturbed flow with $\epsilon_\rho = 0.2$, model UA1, parallel to the equatorial plane, onto a Kerr black hole with spin $a = 0.9$.

these contain stagnation points.

It is more instructive, however, to examine the streamlines (shown in Figure 8.19). For the tracers that we choose to examine for model UB1, we see that all of those approaching the black hole where it is rotating against the flow direction fall into it, whereas none of those approaching it on the co-rotating side do so. The flow is compressed in from the sides, and the tracers that escape the domain downstream of the black hole form a tear-drop shape.

For model UA1, where the streamlines are shown in Figure 8.17 we note that the tracers were started on a larger radius circle, but we still see a qualitatively fairly similar flow. However, at the top of the



(a) Equally spaced contours of $\log \rho$.

(b) Velocity contours evaluated in Boyer-Lindquist coordinates.

Figure 8.18: Contour plots of density and velocity for a supersonic flow with density perturbed by $\epsilon_\rho = 0.2$ past a black hole with spin $a = 0.9$ and flow parameters given by model UB1. The plots are evaluated at time $t = 300M$, and the axes are in units of M .

tear-drop, we see that two vortices have developed in the flow. These correspond to the two low density regions mentioned above.

The cross-sections of the accretion regions upstream of the black hole are shown in Figure 8.15. Again the accretion regions corresponding to the unperturbed, non-spinning cases are circular. We also see that the combined effect of the spin and perturbation is to move the accretion region towards the top of the domain, and to cause it to decrease in size. We also see that the accretion regions for the higher adiabatic index (model UB1) are somewhat smaller than those for model UA1.

Again, the gross flow features are borne out by an examination of Ruffert's simulations in a density perturbed flow [95]. We also note that the angle by which the shock-cone is tilted is approximately the same in both models.

In Figure 8.20 we show the mass accretion rates for the four supersonic models we have presented. The rates for model UA1 have not quite reached a constant value at time $t = 400M$, but those for model UB1 have already attained their limit at $t = 250M$. Again we see that the effect of spinning the black hole and introducing a density perturbation is to lower the mass accretion rate, as suggested by the decrease in area of the accretion regions.

In Figure 8.21 we show the evolution of the stagnation point for the two non-spinning, non-perturbed supersonic models we have presented. We see that they both reach a constant state by the end of the

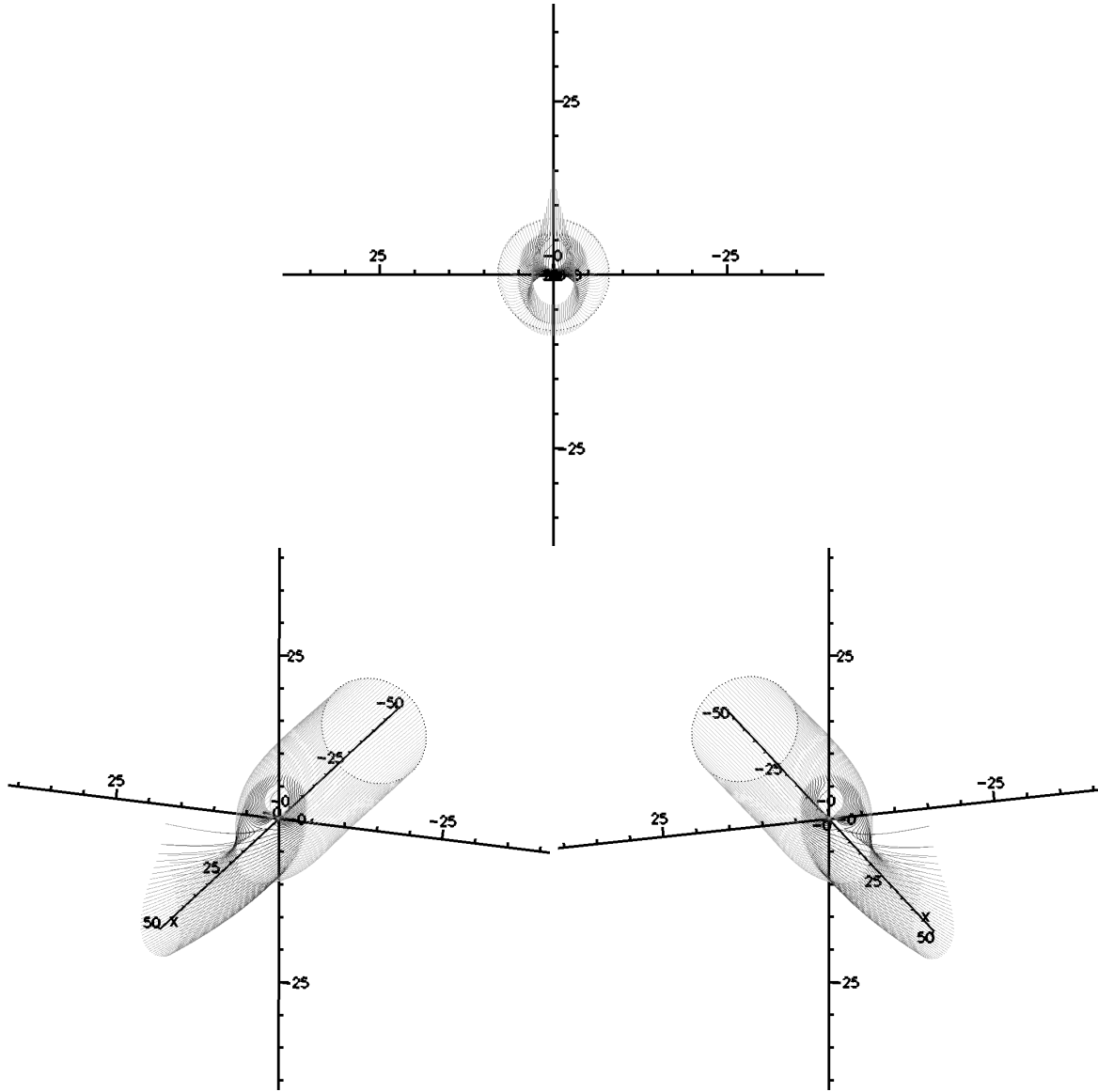


Figure 8.19: Plot of streamlines for a perturbed flow with $\epsilon_\rho = 0.2$, model UB1, parallel to the equatorial plane, onto a Kerr black hole with spin $a = 0.9$.

evolution.

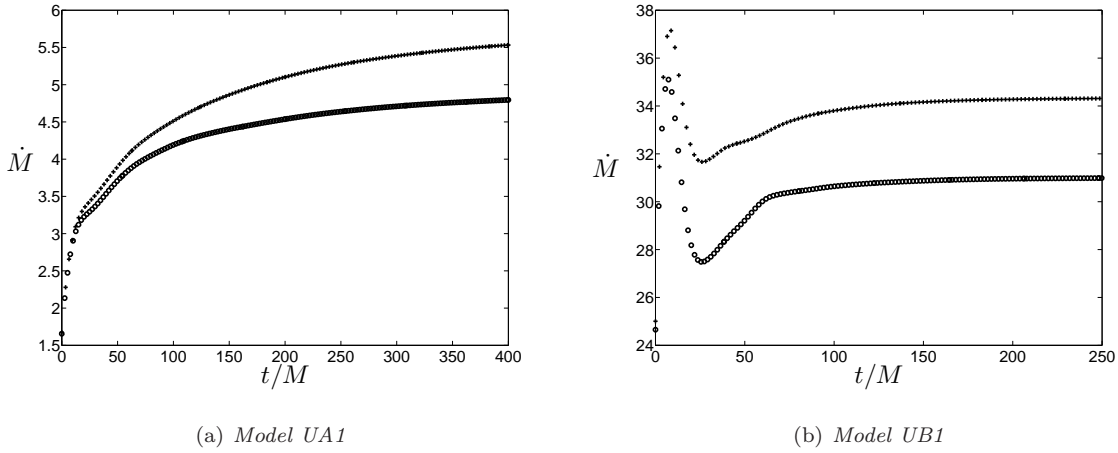


Figure 8.20: Variation of mass accretion rate with time for two supersonic models. In both plots the + signs correspond to the uniform, non-spinning case, and the circles correspond to the perturbed, $\epsilon_\rho = 0.2$, spinning, $a = 0.9$ case.

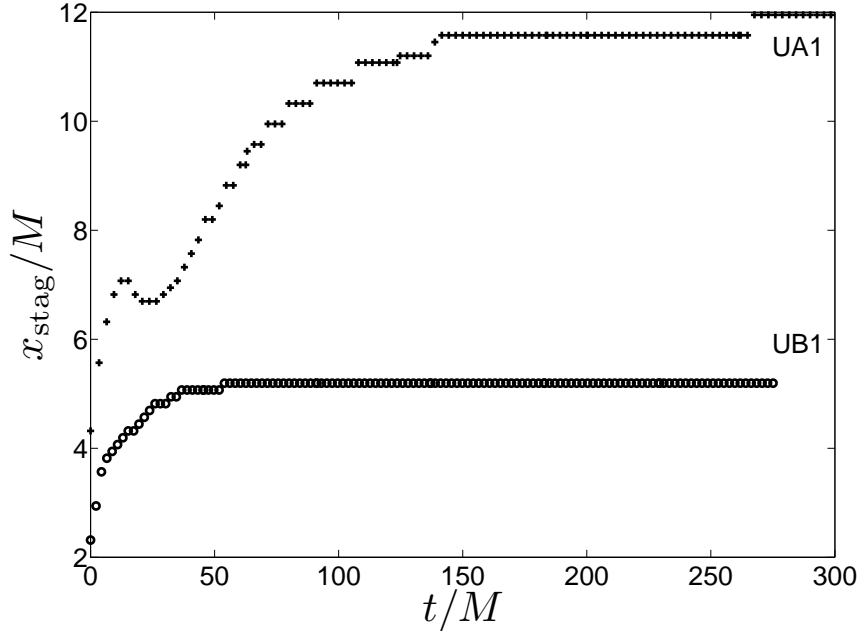


Figure 8.21: Evolution of stagnation point locations with time for the supersonic models UA1 (+ signs) and UB1 (circles). A loss of data, for technical reasons, caused model UB1 to stop earlier, rather than any numerical effects.

8.2.4 Comparison with previous results

We show the limiting values for the mass accretion rates in Table 8.2, with Font and Ibáñez’s results [41] for comparison. Although our results are similar in magnitude to those previously found,

Model	\dot{M}	\dot{M}_{FI} [41]	$\dot{M}/\dot{M}_{\text{FI}}$
UB0	5.39	4.6	1.17
UC0	23.33	21.0	1.11
UA1	5.53	3.7	1.49
UB1	34.32	27.0	1.27

Table 8.2: Comparison of mass accretion rates found using our algorithm with those of Font and Ibáñez. Both sets of results were evaluated at $r = 5M$ and scaled in the same way.

Model	x_{stag}/M	$x_{\text{stag}}^{\text{FI}}/r_a$ [41]	r_a/M	$x_{\text{stag}}^{\text{FI}}/M$
UB0	6.70	2.70	2.2	5.94
UC0	4.44	3.88	1.1	4.27
UA1	11.95	3.24	3.2	10.37
UB1	5.19	5.22	0.92	4.80

Table 8.3: Comparison of stagnation point locations found using our algorithm with those of Font and Ibáñez, which are then scaled to be in units of M rather than of r_a .

our rates are persistently higher, by up to 50%. Although our results will become more accurate with increased resolution, the results in Table 7.3 suggest that the effect will not be substantial.

In Table 8.3 we show the location of the stagnation points for the four non-spinning unperturbed models we have presented above. An examination of the velocity plots in [41] suggests that the stagnation point locations are written in units of the accretion radius r_a . The location of the stagnation points found using our algorithm is similar to, but consistently further out than those found by Font and Ibáñez.

The results we have found, therefore, are sufficiently similar to those of previously published results to give us confidence in the accuracy of our algorithm, and in our results.

8.2.5 Conclusions

In this section we have shown some of the flow morphologies that can arise from accretion onto a black hole. We have, to a large extent, confirmed the results of Font and Ibáñez, although there is a slight disagreement in the mass accretion rates and locations of the stagnation points.

We have also demonstrated the combined effects of allowing a spinning black hole and a fluid whose density is perturbed upstream. These show significant alterations in the flow properties, some of which were demonstrated by Ruffert in the Newtonian regime. In particular, including spin and a density perturbation cause the flow to twist around the spin-axis of the black hole. The flow also develops

complex features; all the models show the development of a line of lower density in the $z = 0$ plane, and one model develops vortices.

The overall effect for all the models is to lower the mass accretion rate, and to change the area and location of the accretion region. However, in order to understand these effects more fully, we shall need to perform a parametric study, examining the separate and combined effects of spin, density perturbation, and wind direction.

8.3 Parametric study setup

Within the framework of Bondi-Hoyle-Lyttleton flow, we have the following parameters that can be varied entirely independently of each other:

M	Black hole mass
a	Black hole spin ($0 \leq a < 1$)
\mathcal{M}	Upstream asymptotic Mach number of the flow
v_∞	Upstream asymptotic velocity of the flow (The sound speed is $c_\infty = v_\infty/\mathcal{M}$)
Γ	Adiabatic index of ideal fluid
n^i	Incident wind direction
p^i	Perturbation direction
ϵ_ρ	Strength of density perturbation ($\epsilon_\rho \ll 1$)
ϵ_v	Strength of velocity perturbation ($\epsilon_v \ll 1$)

Varying these in all possible combinations would lead to a vast and unwieldy set of results, and so we seek to find representative sample of the parameter space to investigate.

Since it is just a scaling parameter, we fix the black hole to have mass $M = 1$. The spin axis is fixed to be the z -axis, and so we vary the incident wind direction \mathbf{n} in the $y = 0$ plane, at angles of $\theta_w = 0, \pi/6, \pi/3, \pi/2$ to the x -axis. This corresponds to $\mathbf{n} \cdot \mathbf{e}_y = 0$ and $\mathbf{n} \cdot \mathbf{e}_z = \cos \theta_w$.

The perturbations will have most effect if they occur perpendicular to the wind direction. Due to the asymmetry induced by any spin that might be present, this still leaves us with one degree of freedom, and we fix the perturbation direction to be parallel to the y -axis (i.e. $\mathbf{p} = \mathbf{e}_y$ in (7.6)). The various vectors can be seen in Figure 8.22.

Since low spins do not seem to affect the flow by a large amount, according to the results of Font and Ibáñez, we use three spin parameters: $a = 0, 0.5$, and 0.9 .

Ruffert *et al.* have used $\epsilon_\rho = 0, 0.03, 0.2$. Since we have specified the spin always to be positive, we have to allow for this handedness and use negative values for the perturbations as well, where appropriate. We have chosen not to perturb the fluid velocity for this study.

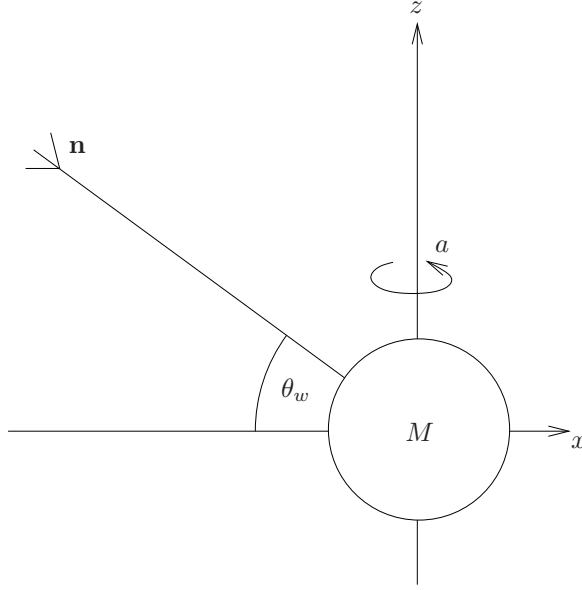


Figure 8.22: *Geometric setup of wind accretion parametric study. The black hole spin is always in a positive direction around the z -axis, the wind direction is in the x - z plane at an angle θ_w to the x -axis, and any fluid perturbations are in the y -direction (positive y goes into the page).*

8.3.1 Models

We choose to apply perturbations to one of the models suggested by Font and Ibáñez: UB1. This is an ultra-relativistic flow of a medium stiffness fluid at a moderate Mach number. Note that the sound speed $c_s = 0.57$ is close to the maximum permitted value of $\sqrt{\Gamma - 1} \approx 0.57735$ (see (3.15)). This model has the advantage that it converges fairly quickly to steady state as seen in Figure 8.2.

We allow for the following perturbations to the base solution:

$$\begin{aligned}
 a &= 0, 0.5, 0.9, \\
 \theta_w &= 0, \pi/6, \pi/3, \pi/2, \\
 \text{and } \epsilon_\rho &= -0.2, -0.03, 0, 0.03, 0.2,
 \end{aligned}
 \tag{8.2}$$

These are applied in all possible combinations, although some combinations are not necessary as they duplicate others. For example, θ_w is irrelevant when $a = 0$, as shown in the next section.

8.3.2 Further validation

Although our grid setup is spherical, it does not have spherical symmetry, due to the overlapping grid structure (see Figure 7.1). In Table 8.4, we show the effect of θ_w on the mass accretion rate for a non-spinning black-hole when using the low resolution grid. In theory, of course, there should be no difference due to the symmetry of the situation. We do, however, see a small difference in the accretion

θ_w	\dot{M}
0	34.26
$\pi/6$	34.19
$\pi/3$	34.19
$\pi/2$	34.28

Table 8.4: *Effect of θ_w on mass accretion rate for non-spinning black hole, at time $t = 300M$, evaluated at radius $r = 5M$. The low resolution grid was used.*

rates. The fact that the results for $\theta_w = 0, \pi/2$ and $\theta_w = \pi/6, \pi/3$ are very similar suggests that the problem is either linked to the angle with a Cartesian axis, or to the interpolation boundary at $\theta_w = \pi/4$.

The effects we have noticed here, though, are smaller than the effects we have found in the results of our study. We therefore conclude that we can safely use the low resolution grid to obtain good results, although it may be necessary to confirm our findings at a later date using a higher resolution grid.

8.4 Parametric study results

In this section we present the results of our parametric study. As performing a parametric study is very CPU-intensive, we calculated these results at the low resolution. However, from our validation results, we believe that the effects of the low resolution on our results are not significant.

8.4.1 Flow morphology

Before showing the processed results for all the simulations we performed, we show how the qualitative form of the flow varies depending on the spin of the black hole, the wind direction, and the strength of the density perturbations.

Accompanying DVD

In selecting plots for this thesis, we encountered two difficulties. Firstly, it was not possible to present the streamlines of the flow adequately in a two-dimensional plot, although we have produced plots showing the main features thereof for inclusion in this thesis. Secondly, although we have selected plots which demonstrate the essential properties of the flows encountered, we felt that there would be a benefit in making available the results for all the parametric variations simulated.

We have therefore included a DVD with this thesis that presents all the results of the simulations in an interactive form. The starting point is to open the file `index.html` in any web browser with an Adobe

FlashTM Player compatible plugin. In the sections below, we explain the information included for each model.

Streamlines

We generate streamlines of the flow by starting a set of tracers on a circle at a distance of $40M$ upstream of the black hole, and with radius $8M$, and evolving them forwards in time. This is done on the final state of the velocity at $t = 300M$, using velocity vectors in the Boyer-Lindquist coordinate basis as described in §7.4. These are shown on the DVD, with a fly-around of the three-dimensional plot, and in this thesis we show the streamlines from different viewpoints to emphasise the relevant features.

The streamlines show up some slight issues with the low resolution used, such as the fact that for the base model with no spin and uniform flow, the cross-section of the streamlines at the end is seen to be not quite circular. The higher resolution simulations described in §8.2 do not have this problem, but otherwise display the same flow features. We are therefore justified in drawing conclusions from the lower-resolution plots.

The streamline algorithm has also been used to determine the size and shape of the accretion region, by starting a set of streamlines covering a circle at a distance of $40M$ upstream of the black hole and checking which tracers fall into the hole. From this we can determine the shape of the accretion region, shown on the DVD for each model, and also in Figures 8.35-8.37.

Density and Velocity plots

The DVD included with this thesis also contains a fly-around of the final steady state of the fluid for the simulations performed. We plot the logarithm of the density, and also the fluid speed as evaluated in Boyer-Lindquist coordinates. These are plotted on three orthogonal planes: the $y = 0$ plane, the plane containing the y -axis and the wind direction, and a third plane orthogonal to the first two.

Mass accretion rates

The mass accretion rates are calculated on the surface of a sphere radius $r = 5M$. Although we have found some discrepancy between our results for mass accretion and those of Font and Ibáñez, we note that our plots always tend to a constant limit, as opposed to some simulations in [42] which exhibit oscillations even at late times in the simulation. A plot of the mass accretion rate, as well as the final value, is shown with the density plot on the DVD for each simulation.

Plot parameters

All the plots in the following sections are standardised for easy comparison. We plot the density with equally spaced contours at $\ln \rho = -0.5, 0, 0.5, \dots, 4, 4.5$. The outer horizon of the black hole is shown with a dashed line at $r = r_+$ (and is non-circular when the spin-axis is not perpendicular to the plot plane), and the plot ranges are $\pm 15M$. The plane in which we plot the contours contains the y -axis and the incoming wind direction. That is, for $\theta_w = 0$, the black hole spin axis is normal to the plot, and for $\theta_w = \pi/2$, the spin axis lies in the plot plane, from left to right. The wind direction is always from the left of the plot.

Increasing the spin of the black hole

In Figure 8.23, we see the effect of varying the spin of the black hole for a uniform flow. The flow is parallel to the equatorial plane of the hole, and we expect our results to be close to those of Font and Ibáñez [42], although their results were for the thin-disc approximation, not the fully three-dimensional results we have here.

As we increase the spin of the black hole, the points where the shock cone meets the horizon move around in the direction of spin. The fluid on the co-rotating side of the hole is pulled further round by the black hole before it is decelerated sufficiently to fall into the hole and, on the counter-rotating side, it is decelerated by the black hole spin so that it falls in earlier.

Although the flow is altered near the horizon, the effect further out is far less, and is nearly indistinguishable from that of the non-spinning case. This is particularly evident from the results when measuring the shock angles, as we shall see in §8.4.2.

Examining the streamlines for the $a = 0.9$ case (Figure 8.24) shows that fluid on the co-rotating side of the hole can escape to infinity when it starts from a smaller radius as compared to fluid on the counter-rotating side, and the minimum impact parameter for which fluid can escape is smaller for higher spin. This is evident in the accretion region which for spin $a = 0.9$ is entirely in the upper half of the domain.

Perturbing the density for a non-spinning black hole

Perturbing the density of the flow onto a non-spinning black hole gives results as in Figure 8.25. The density upstream of the black hole increases down the page.

The shock-cone is visibly skewed round in the anti-clockwise direction in both cases, although more so for $\epsilon_\rho = 0.2$, and this effect persists far away from the hole, unlike the effect of the spin. There is also a line of lower density in the upper part of the shock-cone, so that there are almost two separate cones,

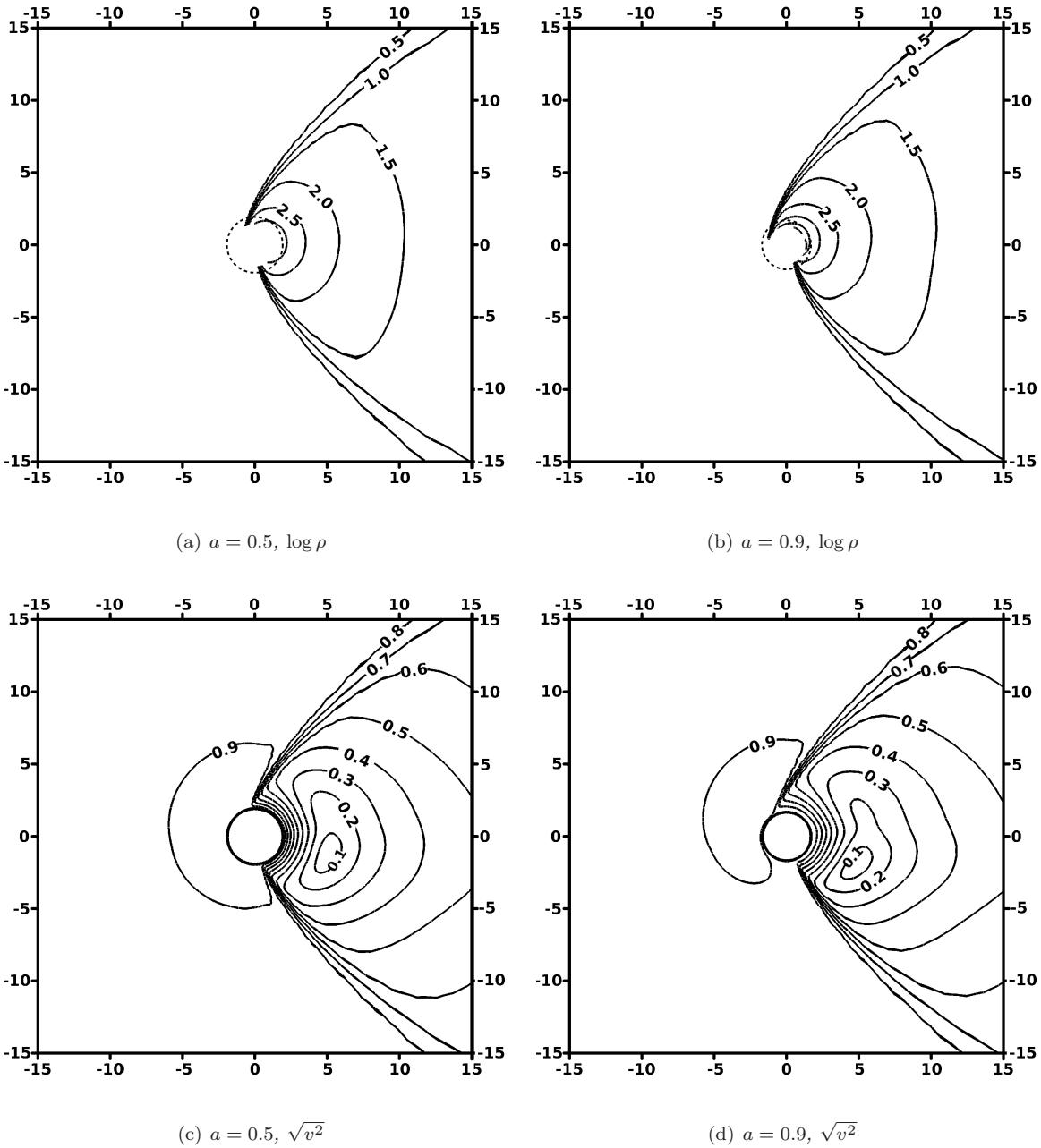


Figure 8.23: *Effect of increasing black hole spin on uniform wind accretion parallel to the equatorial plane of a Kerr black hole. We plot the log of the density in Kerr-Schild coordinates, and the velocity in Boyer-Lindquist coordinates. The black hole spin is in the positive direction.*

although this is only really visible for $\epsilon_\rho = 0.2$. These features are similar to those for highly perturbed Newtonian flow seen in [95].

This effect arises because fluid approaching the black hole on the higher density side has a larger momentum than that approaching on the lower density side, and will therefore continue further around

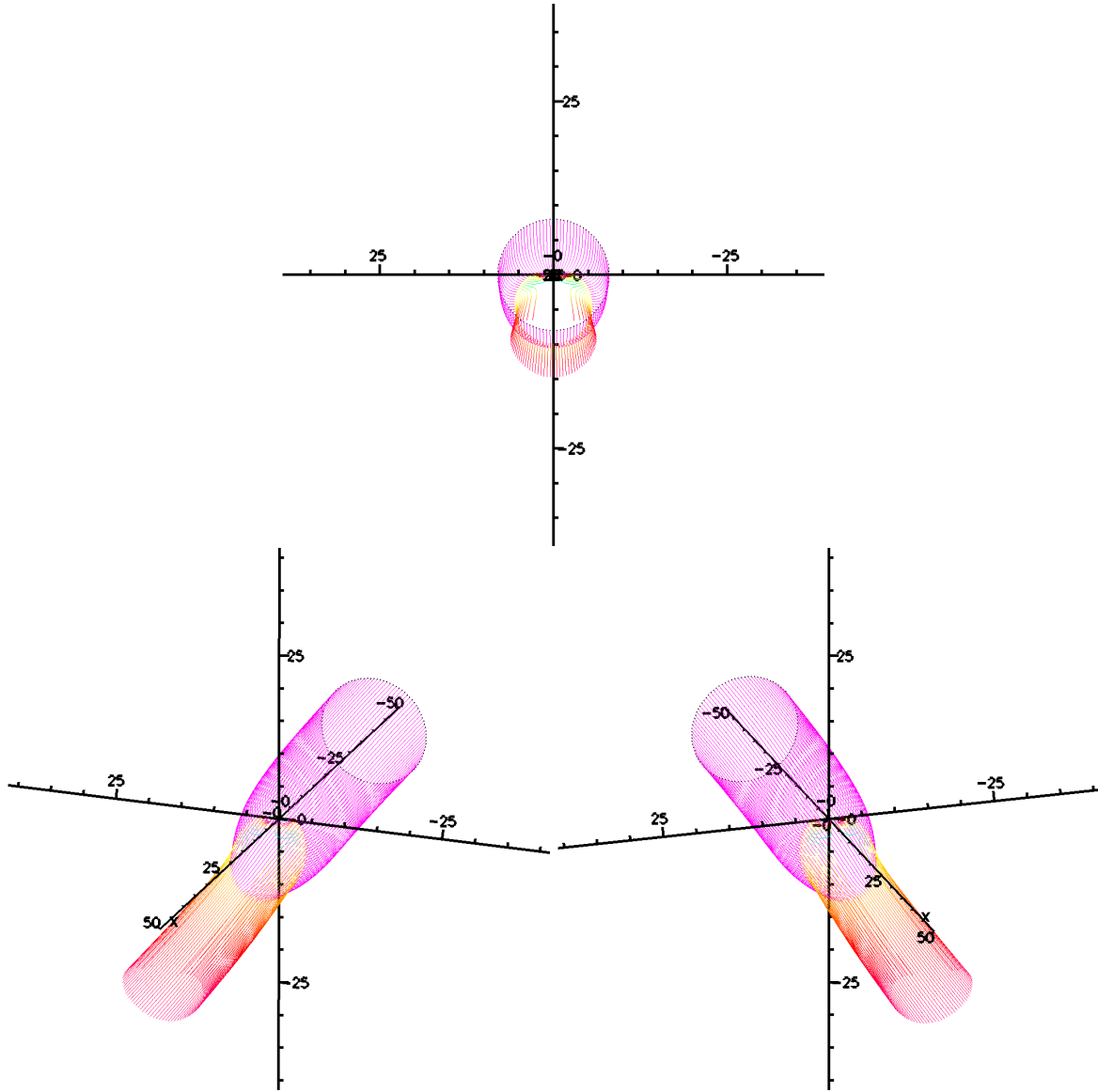


Figure 8.24: Plot of streamlines for a uniform flow parallel to the equatorial plane onto a Kerr black hole with spin $a = 0.9$.

the black hole before it has lost enough momentum to fall in.

From the plots of the accretion region (Figure 8.35), which moves down as the perturbation increases, we see that some fluid from the upper-half domain is deflected by the higher density fluid to such an extent that it does not accrete onto the black hole. The streamlines seen in Figure 8.26 also demonstrate this, since the fluid is seen to be deflected upwards by the black hole, more so for the larger perturbation. The streamlines also show the fluid forming a pair of vortices as it moves downstream.

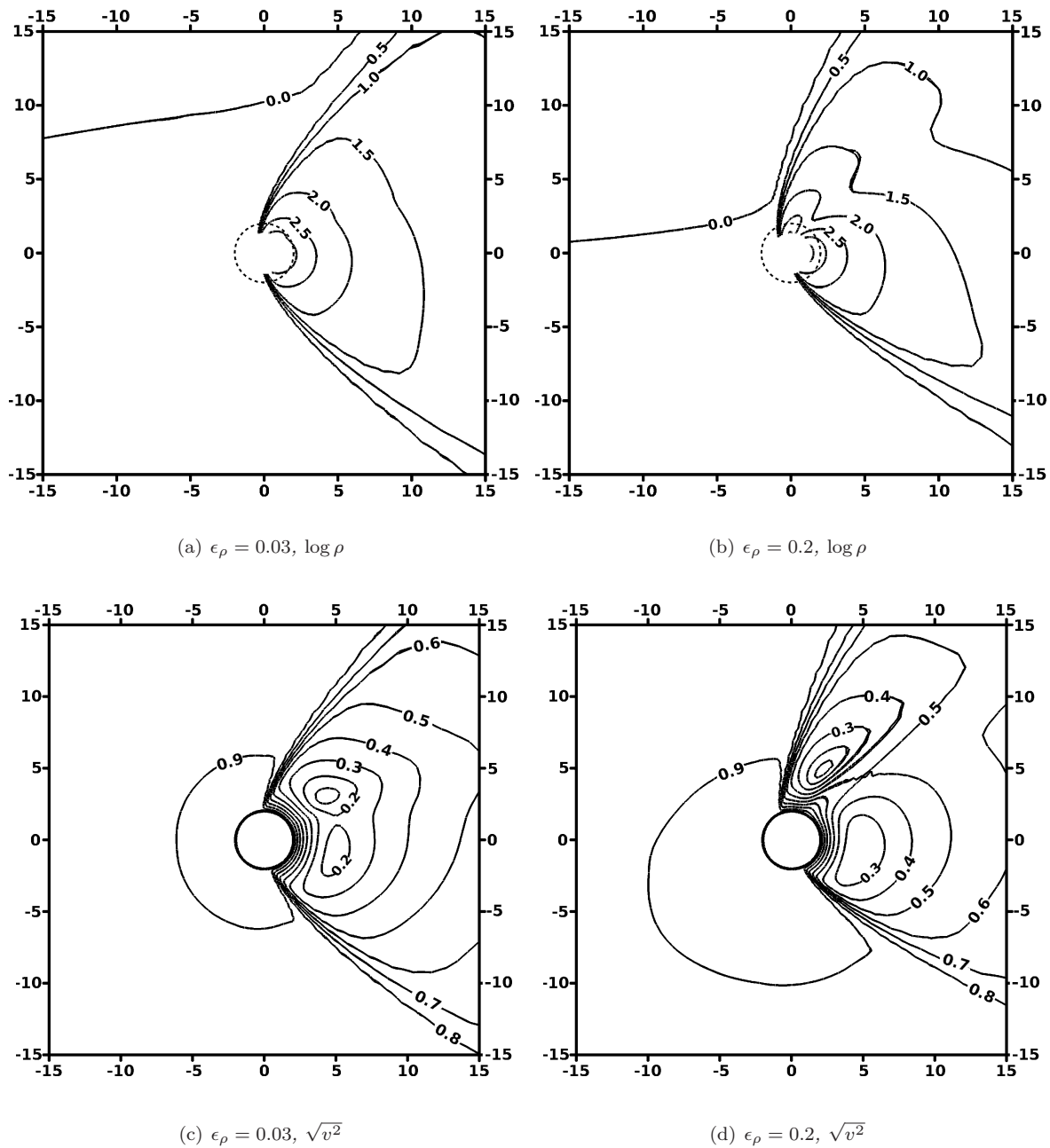


Figure 8.25: *Effect of density perturbation on accretion in the equatorial plane of a non-spinning Kerr black hole, with density increasing down the page. We plot the log of the density in Kerr-Schild coordinates, and the velocity in Boyer-Lindquist coordinates.*

Perturbing the density for a spinning black hole

If we add spin to the most perturbed case, we obtain the results shown in Figure 8.27. Now we expect to see a combination of the effects explored in the previous two sections.

For $\epsilon_\rho = 0.2$, the perturbation takes the point where the shock-cone attaches to the horizon round to

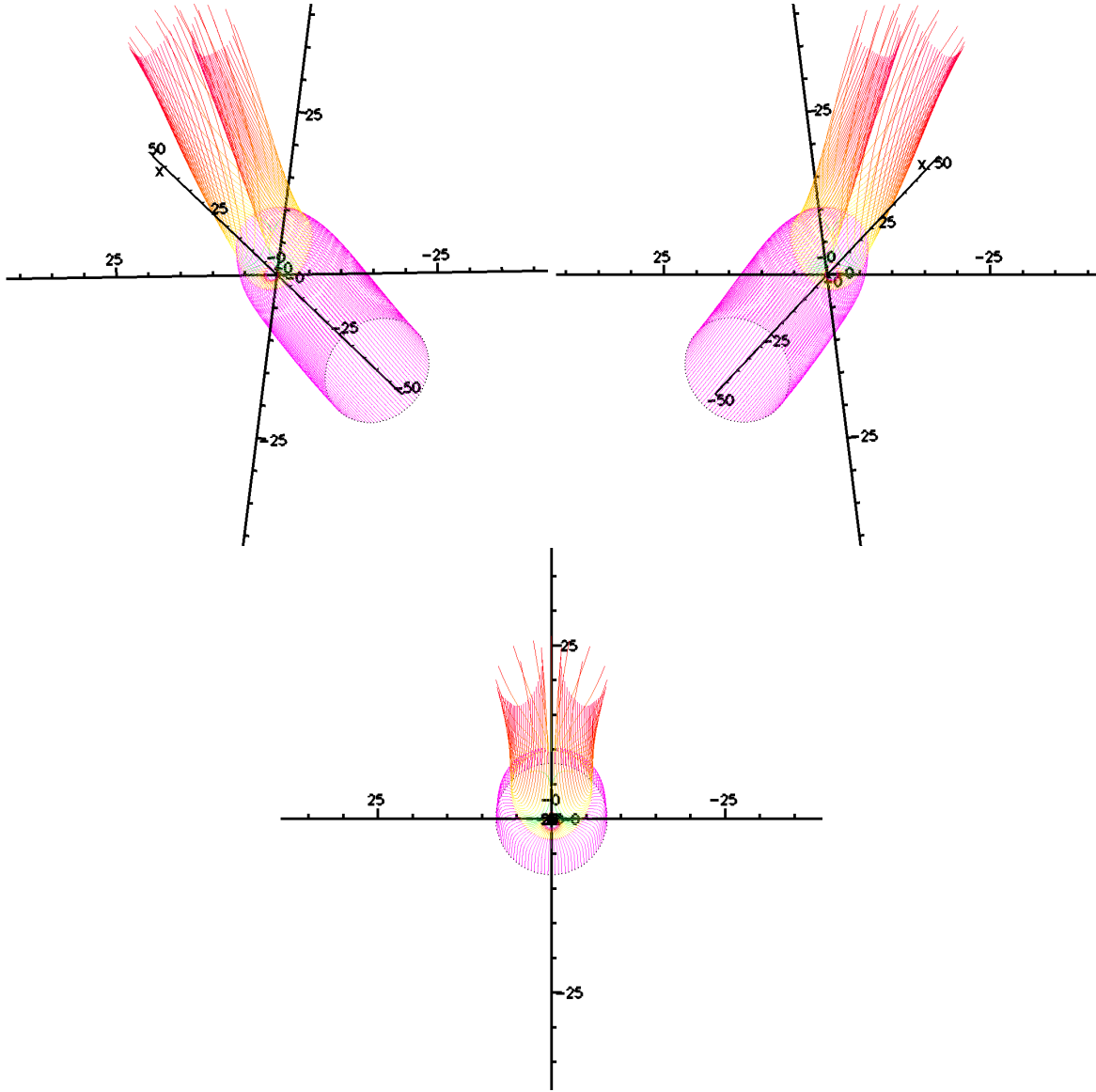


Figure 8.26: *Plot of streamlines for flow with density perturbation $\epsilon_\rho = 0.2$ onto a non-spinning Kerr black hole.*

the upstream region of the flow. A large distance downstream of the black hole, this effect is still present as we would expect from the non-spinning case above and, at the horizon, the shock-cone has wrapped further around in the anti-clockwise direction due to the added effect of the spin.

For $\epsilon_\rho = -0.2$, the effect of the perturbation a long way downstream of the black hole is the reverse of that for the non-spinning positive perturbation. At the horizon, we expect the spin to take the point where the shock-cone attaches round in the anti-clockwise direction, and the negative perturbation to take it in the clockwise direction. We see that the effect of the spin is dominant, in that the point of attachment has overall moved in an anti-clockwise direction.

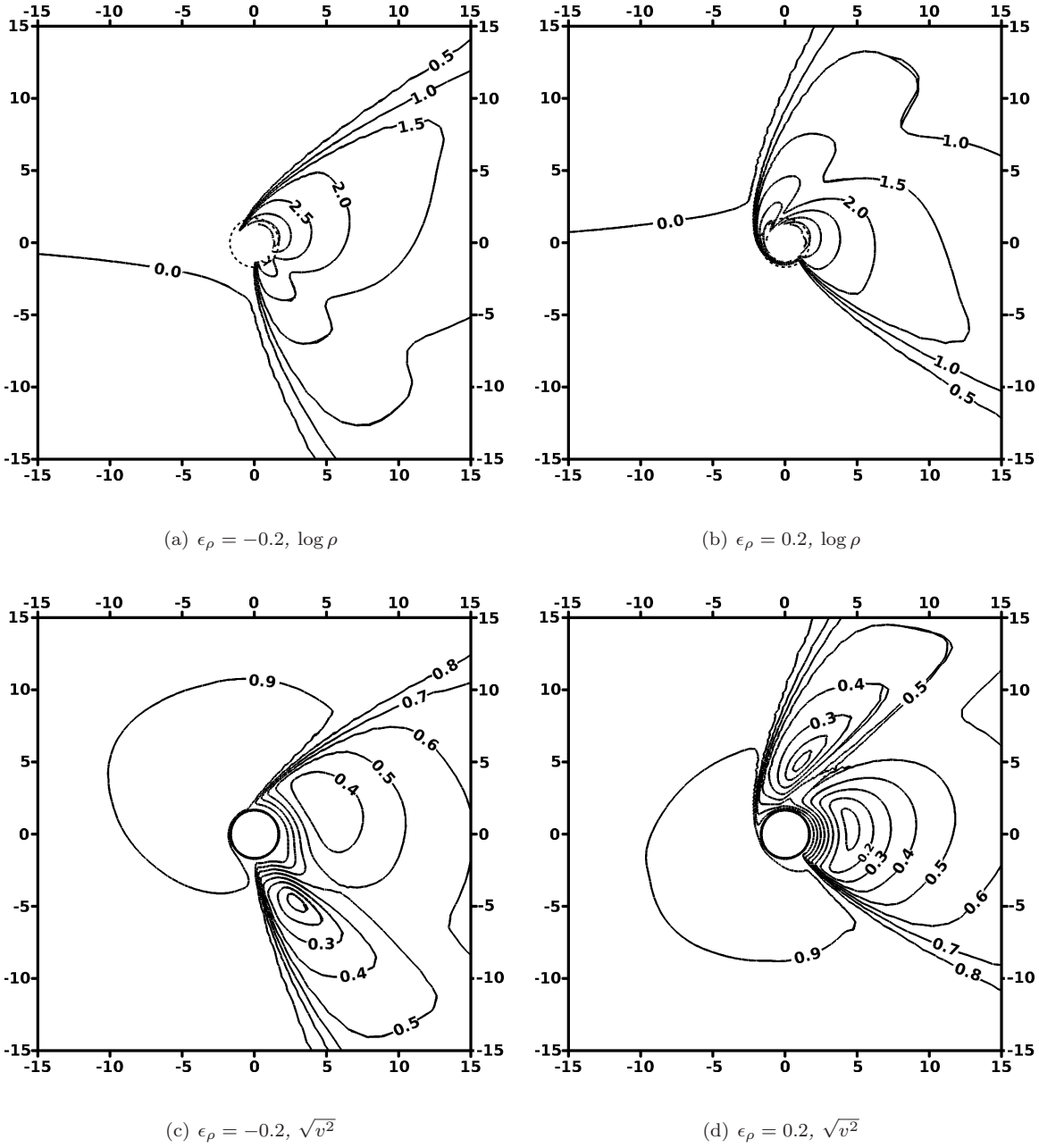


Figure 8.27: *Effect of density perturbation on accretion in the equatorial plane of a Kerr black hole with spin $a = 0.9$. We plot the log of the density in Kerr-Schild coordinates and the velocity in Boyer-Lindquist coordinates. The black hole spin is in the positive direction.*

For both cases, the divided shock-cone is still visible. For the $\epsilon_\rho = 0.2$ case, the effect has increased, so that the contour lines extend further towards the horizon, demonstrating a deepening of the division. For $\epsilon_\rho = -0.2$, the effect has reduced, showing a shallower division in the cone.

For the perturbation $\epsilon_\rho = -0.2$, the flow streamlines in Figure 8.28 appear similar to that for the

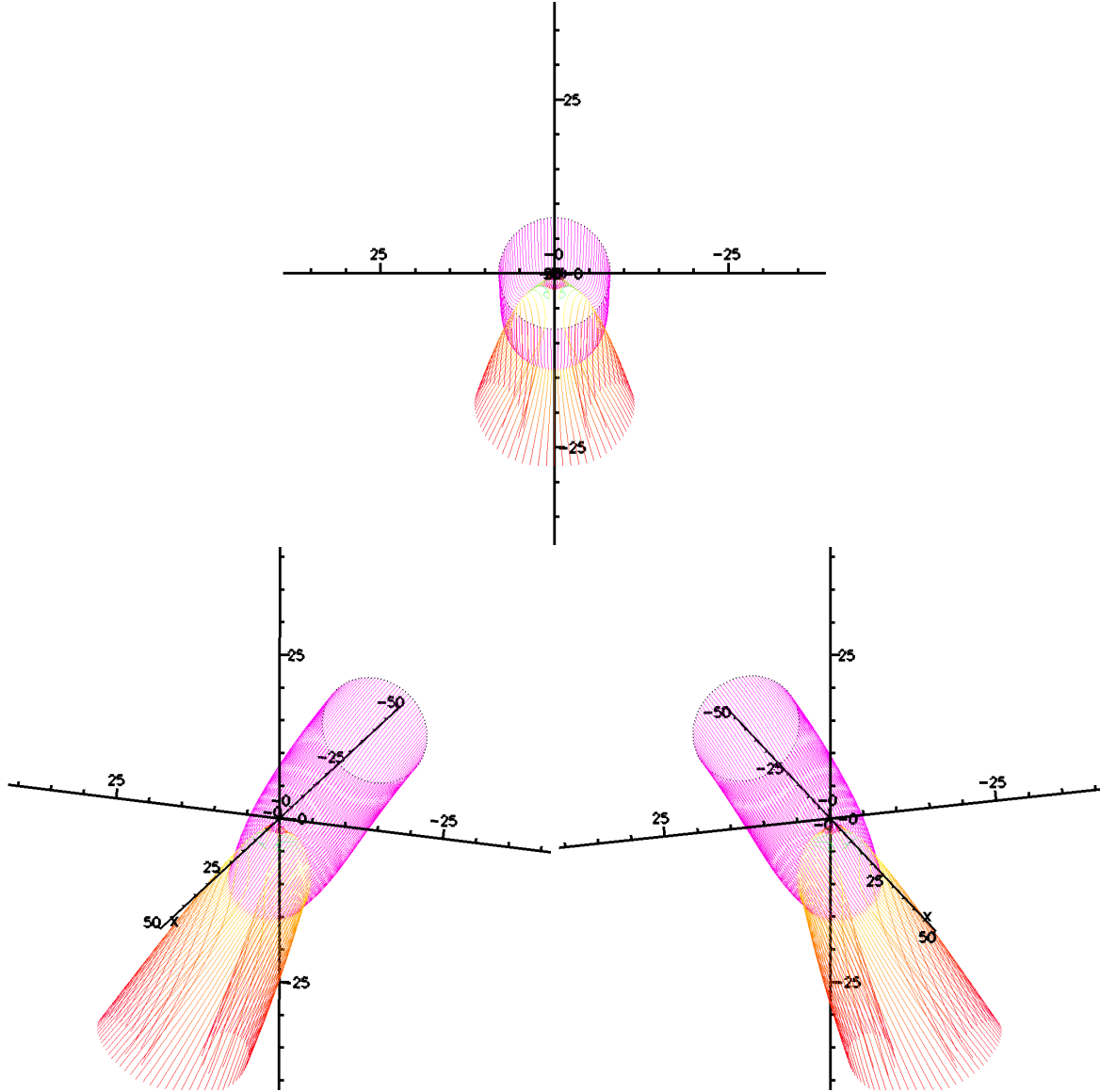


Figure 8.28: *Plot of streamlines for a perturbed flow with $\epsilon_\rho = -0.2$ onto a Kerr black hole with spin $a = 0.9$.*

non-spinning case, although the two vortices have now been moved apart from each other to some extent. For $\epsilon_\rho = 0.2$, we now see a qualitatively different flow (Figure 8.29), in that no vortices are visible, but the ends of the streamlines form a tear-drop shape downstream of the black hole. From an examination of the streamlines for the intermediate case of spin $a = 0.5$ and perturbation $\epsilon_\rho = 0.2$ in Figure 8.30, it appears that the effect of the positive spin is to elongate the two vortices in the vertical direction, at the same time moving them closer together, so that the tear-drop is composed of the inner sides of the original vortices. Starting the streamline tracers at a larger radius would then reveal the outer sides of the vortices.

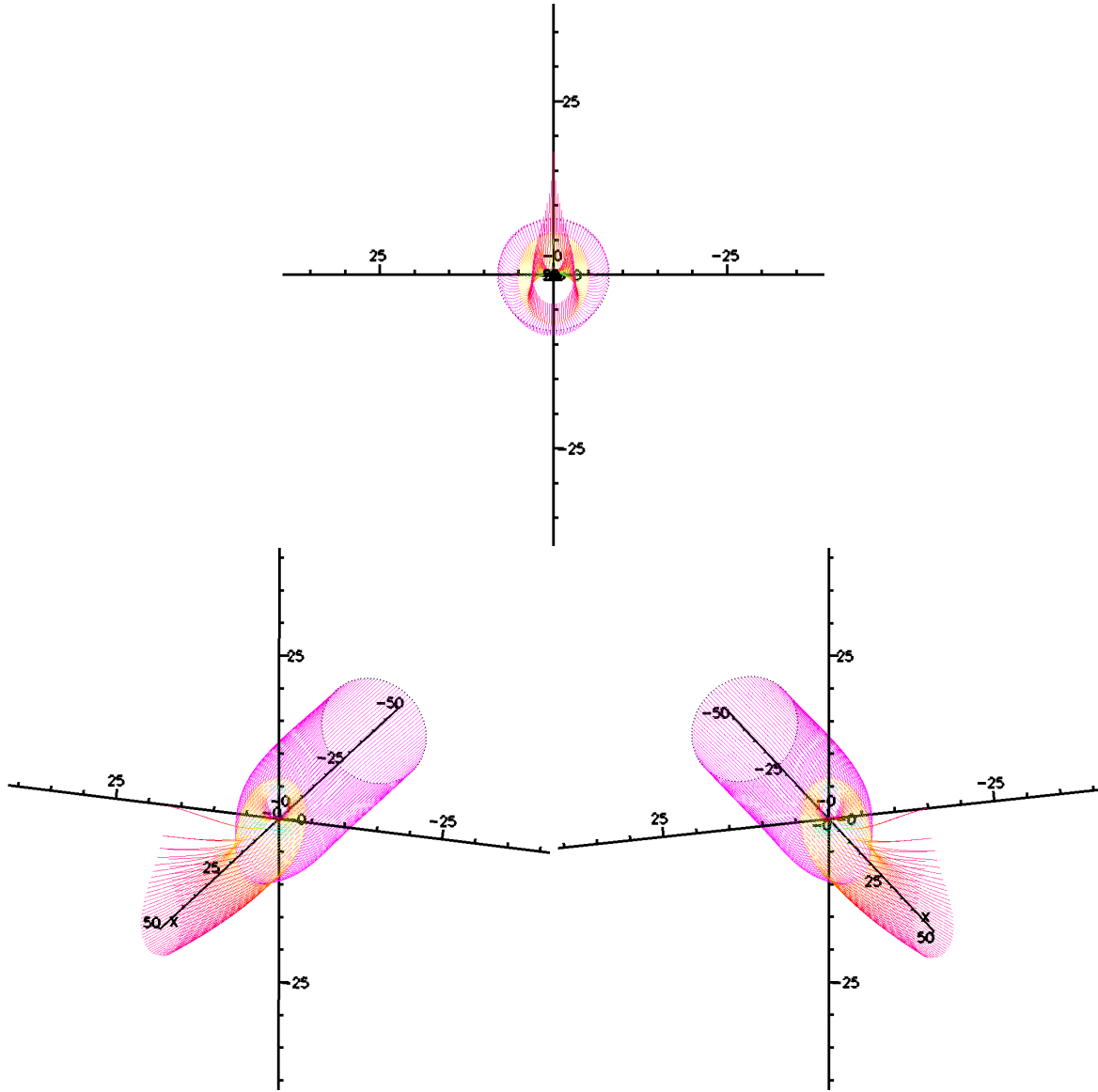


Figure 8.29: *Plot of streamlines for a perturbed flow with $\epsilon_\rho = 0.2$ onto a Kerr black hole with spin $a = 0.9$.*

The effect on the accretion region in the case of $\epsilon_\rho = 0.2$ as compared to $\epsilon_\rho = 0$, as seen in Figure 8.37, is to move it back towards the centre from its position for the unperturbed case. It has now also deformed into a kidney shape. For the case $\epsilon_\rho = -0.2$, the accretion region has now moved significantly above the centre, and has a slight tear-drop shape.

Increasing the spin for flow along the spin axis

We now orient the flow so that it comes in along the spin-axis of the black hole. The result of increasing the spin in this case is shown in Figure 8.31. In this case we hardly see any change to the qualitative

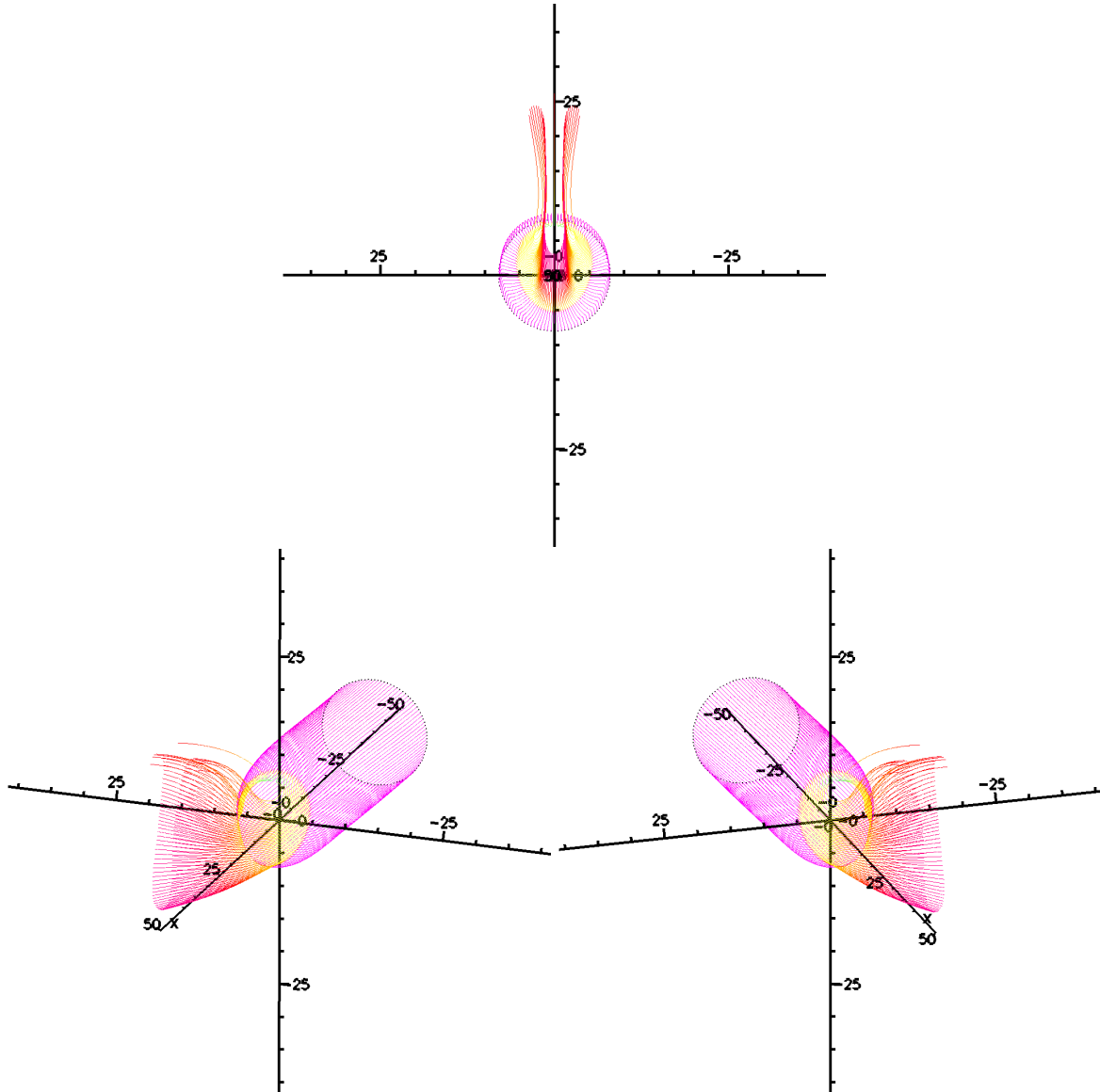


Figure 8.30: *Plot of streamlines for a perturbed flow with $\epsilon_\rho = 0.2$ onto a Kerr black hole with spin $a = 0.5$.*

features of the flow. The maximum density in this plane is less than that for accretion parallel to the equatorial plane, but still increases with spin. There is no apparent change in the streamlines (shown in Figure 8.32) from the non-spinning case, although the area of the accretion region reduces slightly as the spin is increased.

The relative lack of change is due to the fact that the direction of spin is perpendicular to the streamlines. There is some evidence from the streamlines that the flow has been pulled round by the black hole but, again, the effect is only very slight, due to the high velocity of the flow.

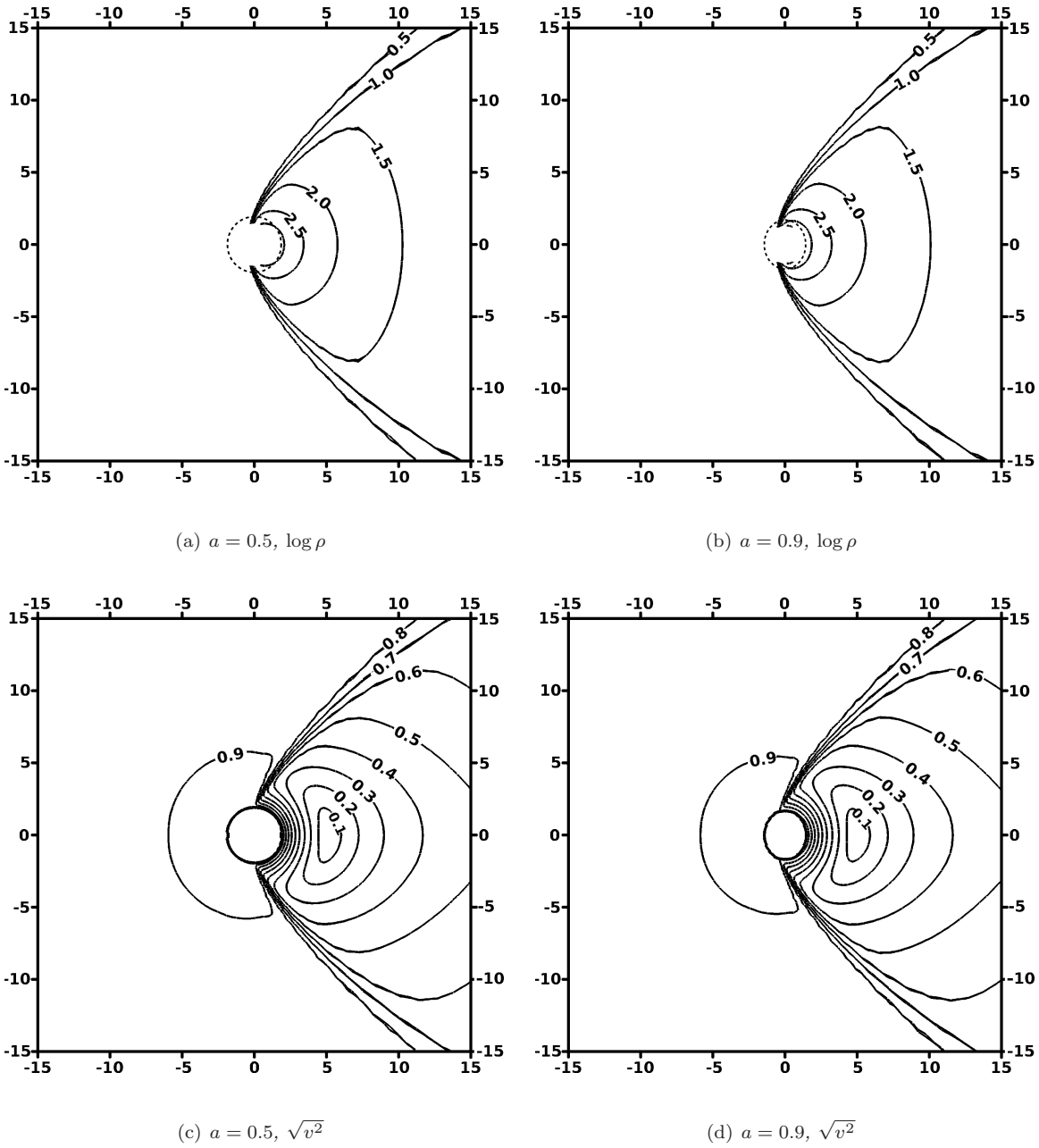
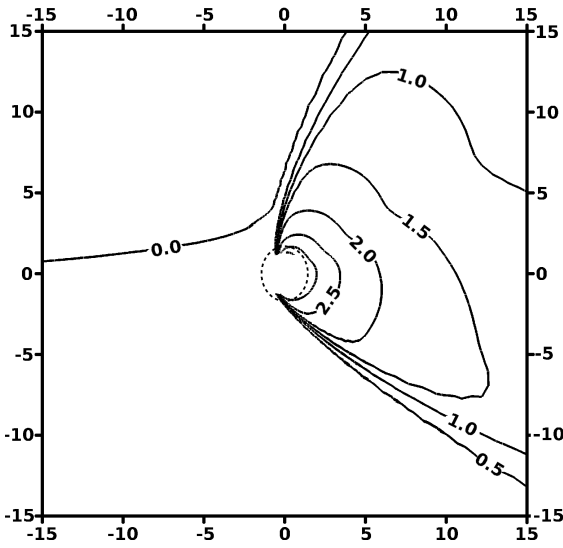


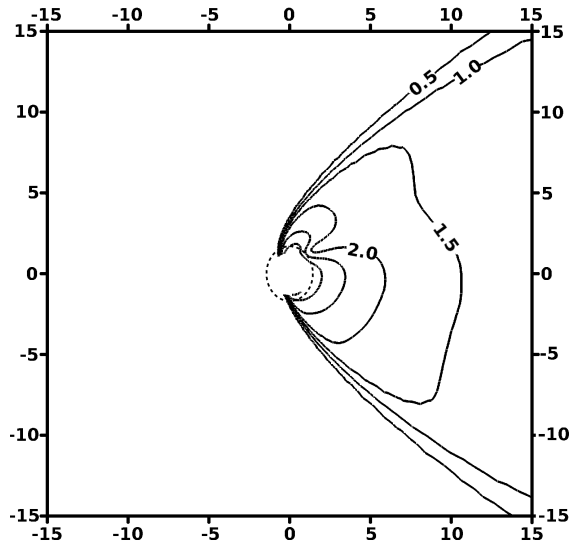
Figure 8.31: *Effect of increasing black hole spin on uniform wind accretion along the spin axis of a spinning Kerr black hole. The axis of spin is the horizontal axis, and the spin direction goes into the page in the upper half of the plot. We plot the log of the density in Kerr-Schild coordinates, and the velocity in Boyer-Lindquist coordinates.*

Perturbing the density for flow along the spin axis

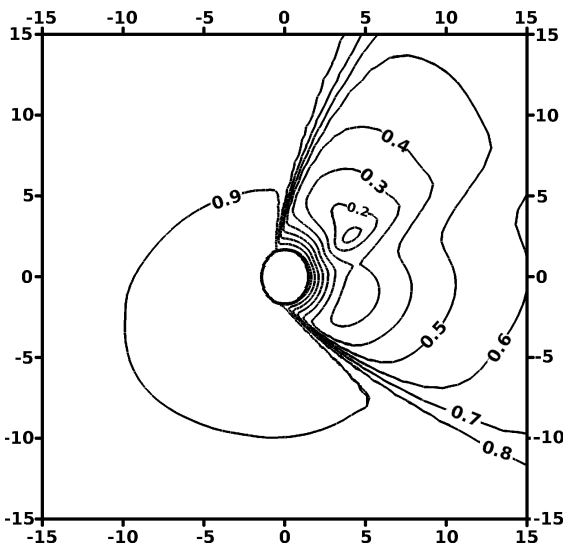
The effect of perturbing the upstream density of flow by $\epsilon_\rho = 0.2$ along the spin axis of a black hole with spin $a = 0.9$ is shown in Figure 8.33. The shock-cone is still tilted away from the symmetric



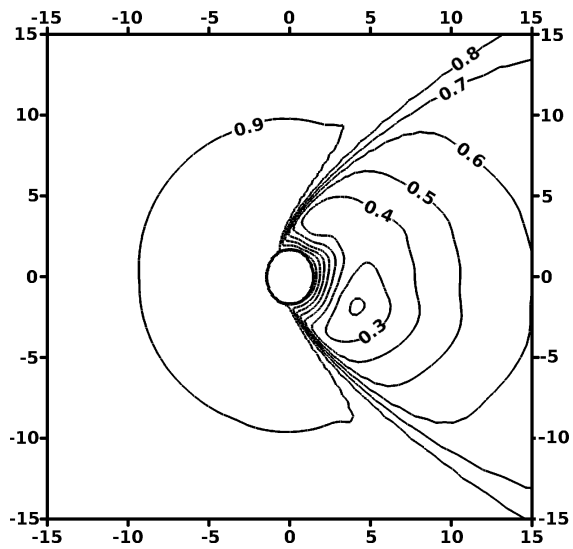
(a) $x = 0$ plane, $\log \rho$



(b) $y = 0$ plane, $\log \rho$



(c) $x = 0$ plane, $\sqrt{v^2}$



(d) $y = 0$ plane, $\sqrt{v^2}$

Figure 8.33: *Effect of density perturbation $\epsilon_\rho = 0.2$ on wind accretion along the spin axis of a Kerr black hole with spin $a = 0.9$. The axis of spin is the horizontal axis, and the spin direction goes into the page in the upper half of the plots.*

the domain.

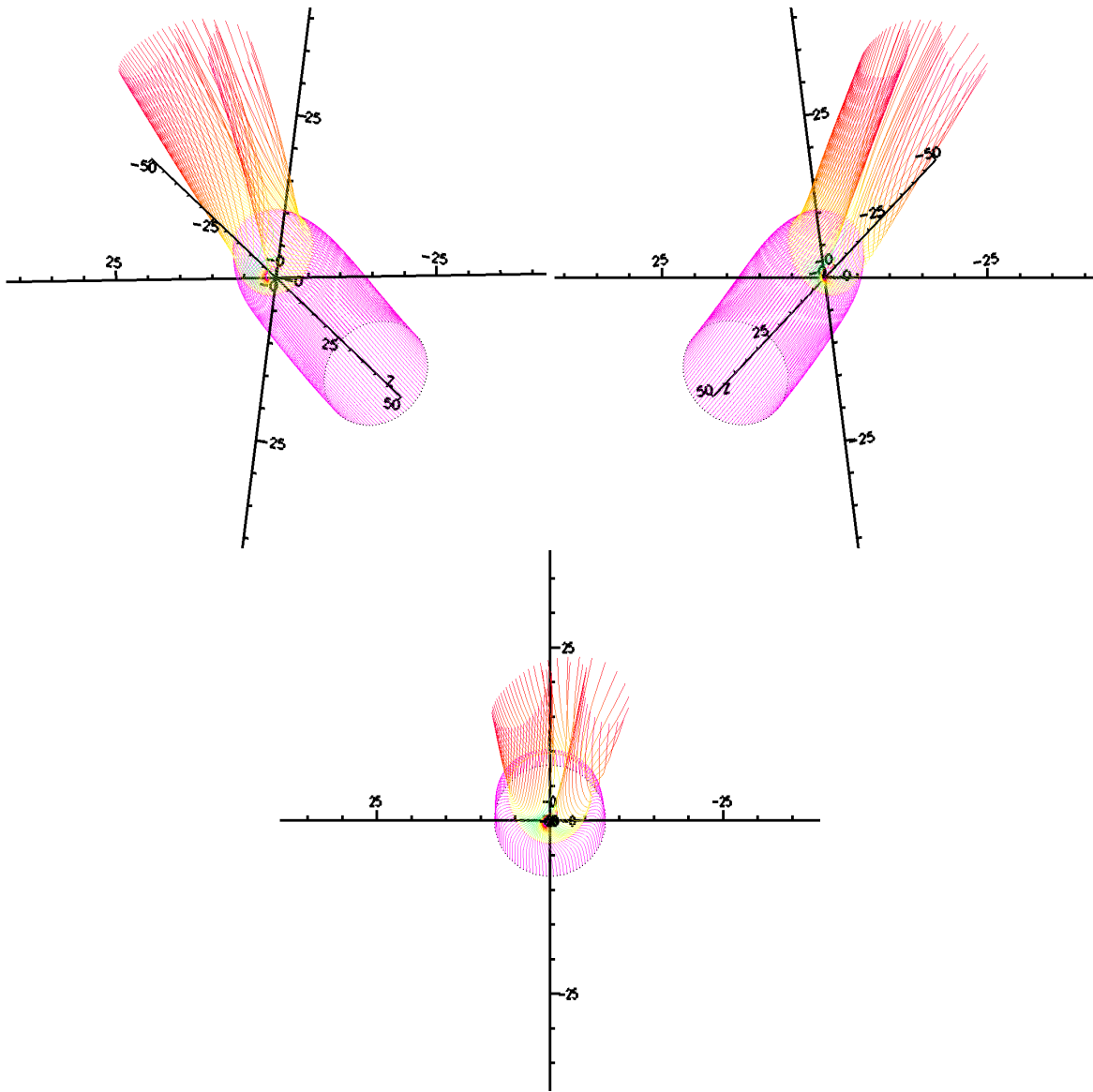


Figure 8.34: *Plot of streamlines for a perturbed flow with $\epsilon_\rho = 0.2$ along the spin-axis of a Kerr black hole with spin $a = 0.9$.*

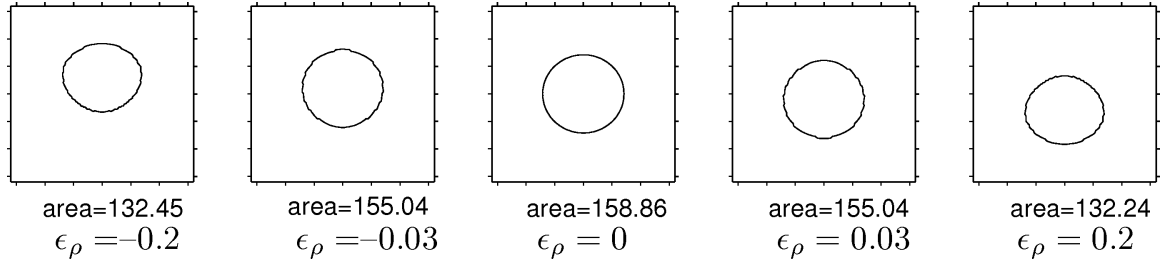


Figure 8.35: *These plots show how changing the density perturbation via ϵ_ρ affects the accretion volume for a non-spinning black-hole. The axis tick-marks are at $-15M, -10M, \dots, 10M, 15M$.*

8.4.2 Processed results

We now proceed to show the processed results, with the effect of the density perturbation and black hole spin on the overall flow characteristics.

Accretion regions

The accretion region plots are shown in Figures 8.35–8.37. The effect of each change is only very slight, but the cumulative effect of each change can be seen by comparing, for example, the plots for $\theta_w = 0$ and $\theta_w = \pi/2$ or $\epsilon_\rho = -0.2$ and $\epsilon_\rho = 0.2$.

The effect of increasing θ_w is only very slight, but the trend is that the accretion region decreases in area for $\epsilon_\rho < 0$, and shifts in the negative y -direction. As we would expect from symmetric arguments, the plot for $\theta_w = \pi/2, \epsilon_\rho = 0$ is circular in aspect. As ϵ_ρ increases, the accretion region moves appreciably in the negative y direction, and increasing θ_w brings about an increase in area.

Mass accretion rate

In Figures 8.38 and 8.39 we show the effect of varying the density perturbation parameter on the mass accretion rate for spins of $a = 0, 0.5, 0.9$. The overall effect on the mass accretion rate is fairly small, at less than 10%.

When the wind direction is along the spin axis ($\theta_w = \pi/2$), the direction of the perturbation does not affect the mass accretion rate, as we would expect from symmetry considerations. The greatest effect on the mass accretion rate occurs when the wind is parallel to the equatorial plane of the black hole.

Increasing the (positive) spin of the black hole moves the accretion region in the positive y -direction, as material in the $y > 0$ half-space is decelerated more strongly, so that the radius inside of which material finally ends up inside the black hole increases for positive y . Similarly, the acceleration of the flow in the $y < 0$ half-space caused by the spin means that the accretion radius decreases for negative y .

The upshot of this is that the accretion region moves in the positive y -direction for increased spin,

as we see from Figures 8.35–8.37. For negative ϵ_ρ , this means that the accretion region will contain more material as a result of the density being higher there, and the mass accretion rate will therefore increase. For positive ϵ_ρ , the accretion region will contain less material, and so the mass accretion rate will decrease. There is also the additional effect that the accretion region changes in size as a result of the density perturbation altering the flow.

From Figure 8.39, we see that for flow parallel to the equatorial plane, the mass accretion rate is reduced more strongly by increased perturbation than for flow along the spin axis, since the area of the accretion region has a wider variation for $\theta_w = 0$ than for $\theta_w = \pi/2$. The effect of the change with perturbation is once again the result of two combined effects: change in location and size of the accretion region and change in available material in that region due to density perturbation.

Shock angle

In Figures 8.40 and 8.41 we show the effect of spin and density perturbation on the asymptotic angle of the shock cone. As a specific example, we note that the shock-cone opening angle for zero spin and no perturbation is found to be 38° , which is less than the analytical value from $\sin^{-1}(1/\mathcal{M}) = 41.8^\circ$, but greater than that found by Font and Ibáñez, $\theta = 35^\circ$.

We have seen earlier that both the spin and the perturbation affect the innermost regions of the flow very strongly, but that the only major effect on the shock cone a long way from the black hole is the density perturbation. This is confirmed by Figure 8.40.

Further, we now see from Figure 8.41 that varying the angle between the equatorial plane and the wind direction has no effect on the angle between the shock-cone edges and the wind direction, within numerical and shock location errors. The spin of the black hole does not appear to have any effect on the shock-cone angle either.

However, the effect of the perturbation is very noticeable, varying the direction of the shock-cone by nearly 30° overall. The total opening angle of the shock-cone is unaffected by the perturbation.

Angles of Drag

In Figures 8.42 and 8.43 we show the angle of drag on the black hole in the plane containing the y -axis and the wind direction. These were calculated from momentum accretion rates at $r = 5M$. We have not plotted the drag angles in the yz -plane as these were smaller, and would not contribute as large an effect.

As spin increases for a uniform flow, Figure 8.42 shows that the black hole is dragged at a progressively steeper angle in the negative y -direction. This is in the same direction as would occur from the Magnus effect, although clearly the mechanism is entirely different. The relative dependence of the drag angle on perturbation is essentially independent of the spin, as can be seen from the similar shape of all of

the lines in Figure 8.43. A negative perturbation causes the drag angle to become more negative, and a positive perturbation causes it to become more positive. In fact, for a perturbation corresponding to $\epsilon_\rho > 0$, the drag angle becomes positive, and the black hole is dragged in the positive y -direction for zero spin, i.e. into the less dense part of the medium.

By symmetry, the drag angle is zero for $\epsilon_\rho = 0$ when the wind is directed along the spin-axis $\theta_w = \pi/2$, but as the wind direction moves round to lie parallel to the equatorial plane, the angle of drag becomes more negative. When the wind is directed along the spin-axis, the spin apparently causes a slight counterbalance to the perturbation effect as the drag angle is slightly lower for higher spin (quantitatively, the drag angle is reduced by about 7% from $a = 0$ to $a = 0.9$).

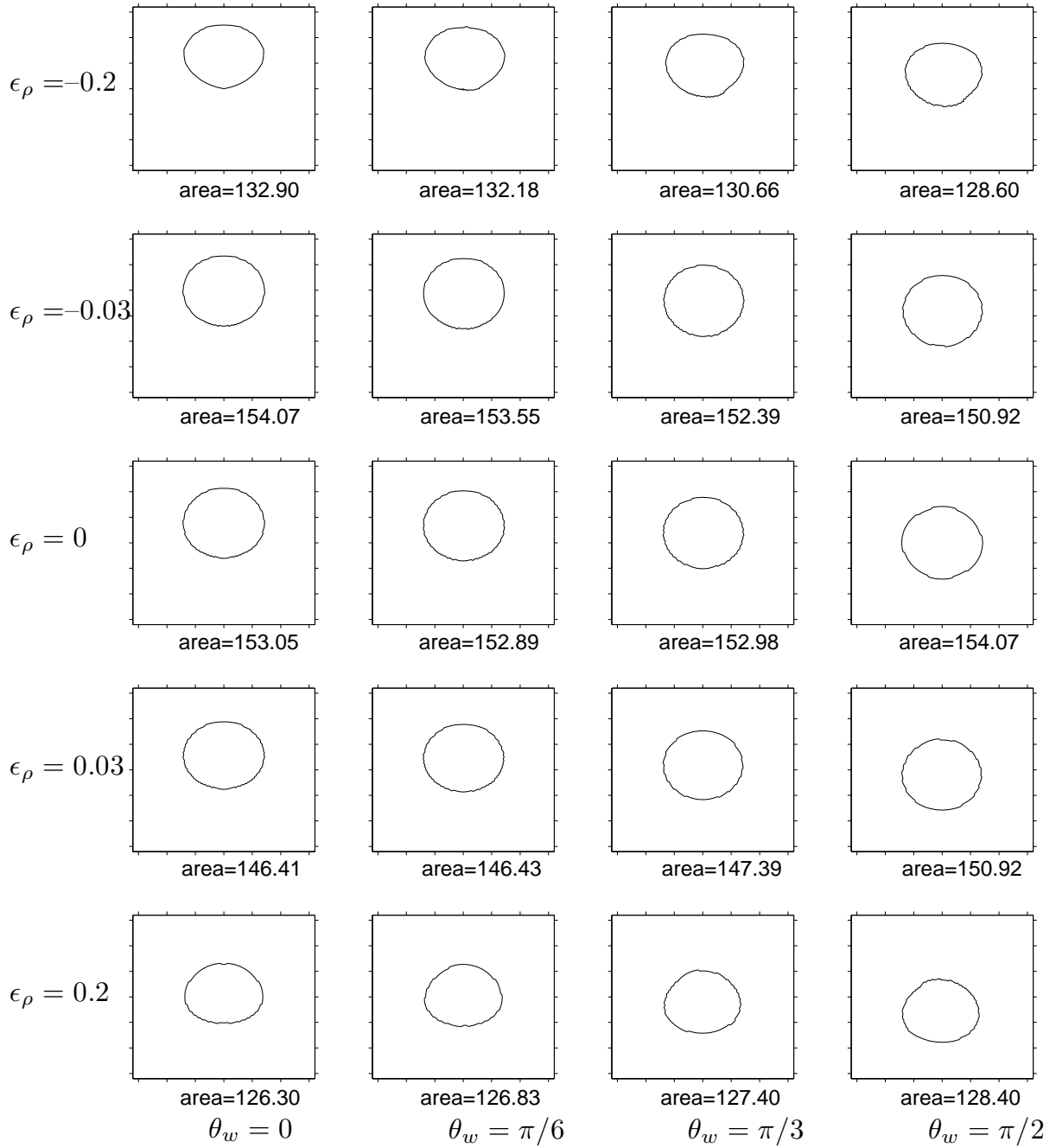


Figure 8.36: These plots show how changing the wind direction via θ_w and the density perturbation via ϵ_ρ affect the accretion region for a black hole with spin $a = 0.5$. The axis tick marks are at $-15M, -10M, \dots, 10M, 15M$.

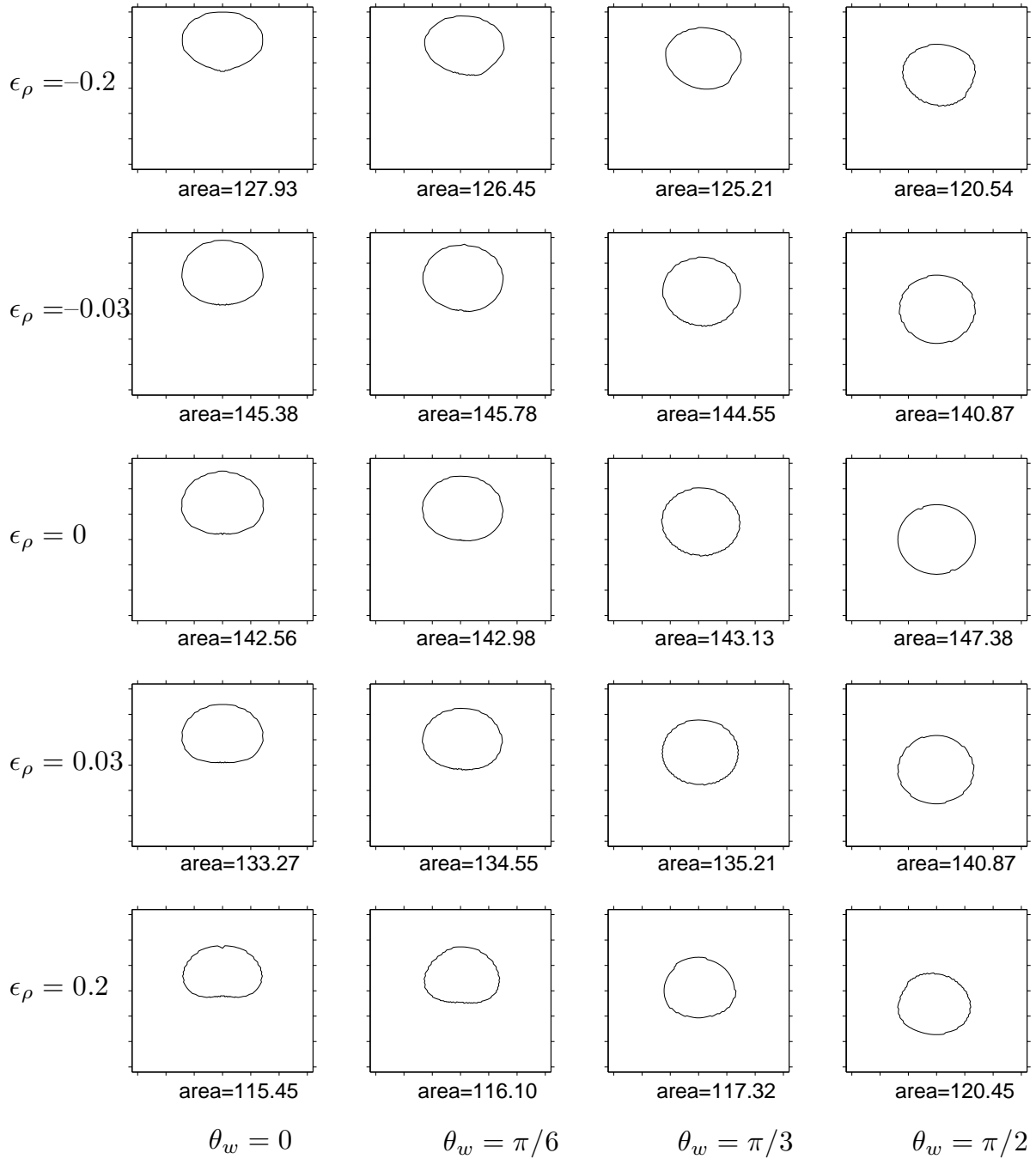


Figure 8.37: These plots show how changing the wind direction via θ_w and the density perturbation via ϵ_ρ affect the accretion region for a black hole with spin $a = 0.9$. The axis tick marks are at $-15M, -10M, \dots, 10M, 15M$.

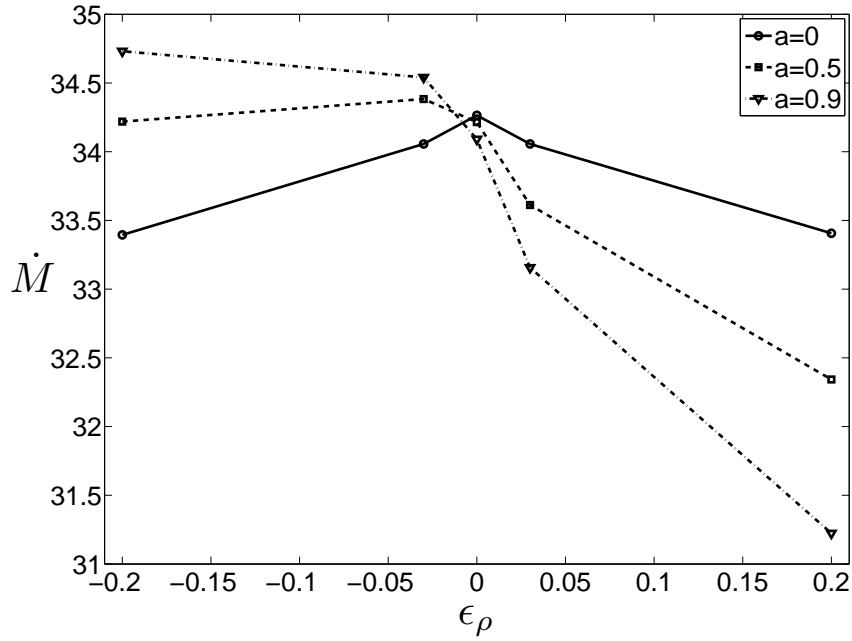


Figure 8.38: Plot of mass accretion rate against density perturbation for different spins. The wind direction is parallel to the equatorial plane of the black hole in all cases.

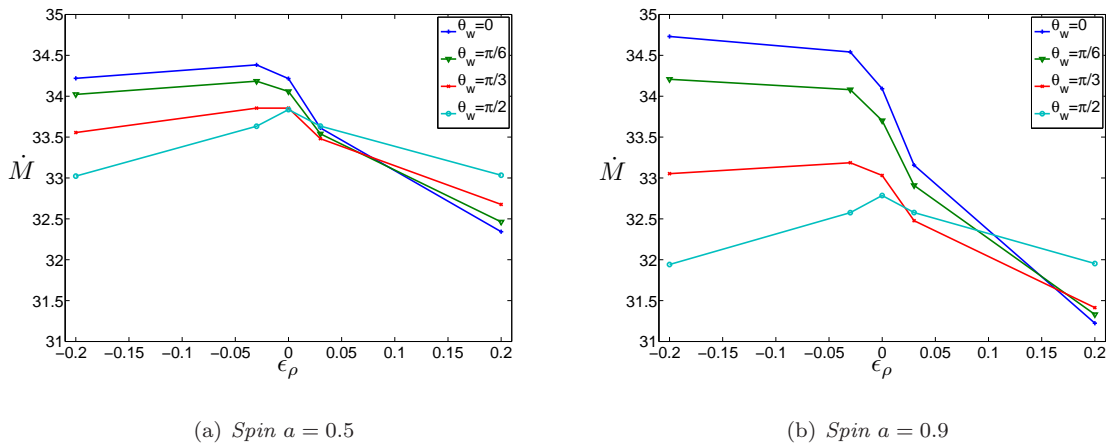


Figure 8.39: Plot of mass accretion rates against density perturbation. The labels refer to the angle between the equatorial plane and the wind direction.

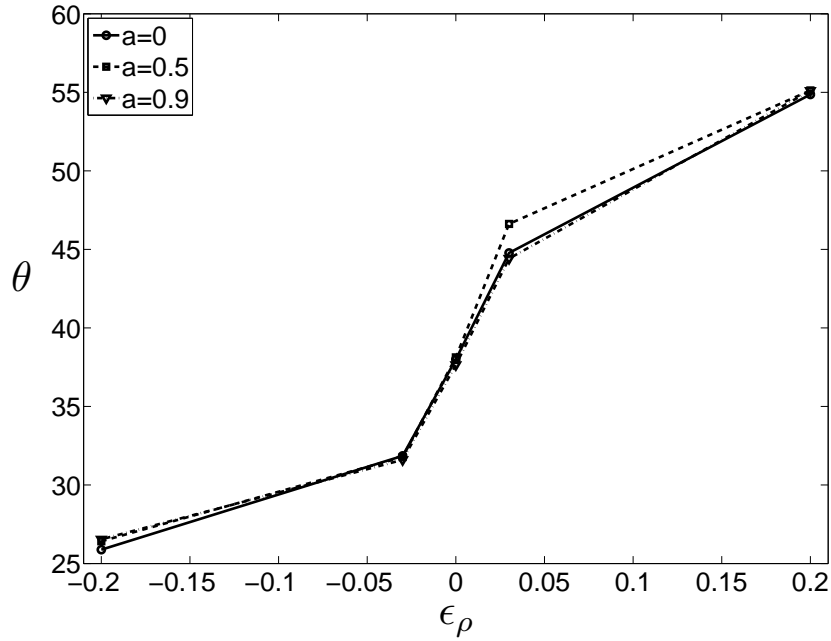


Figure 8.40: Plot of shock angles against density perturbation, varying the spin of the black hole. The wind direction is in the equatorial plane of the black hole in all cases.

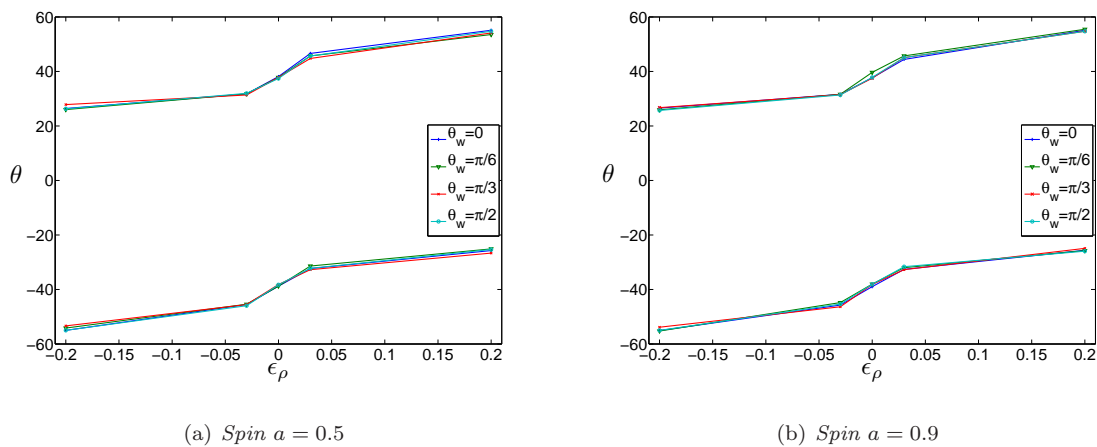


Figure 8.41: Plot of shock-cone angles against density perturbation. The angles are all taken in the plane containing the wind direction and the y -axis. The data points above and below the axis correspond to the shock-angles above and below the wind-direction axis. The labels correspond to the angle between the equatorial plane of the black hole and the wind direction.

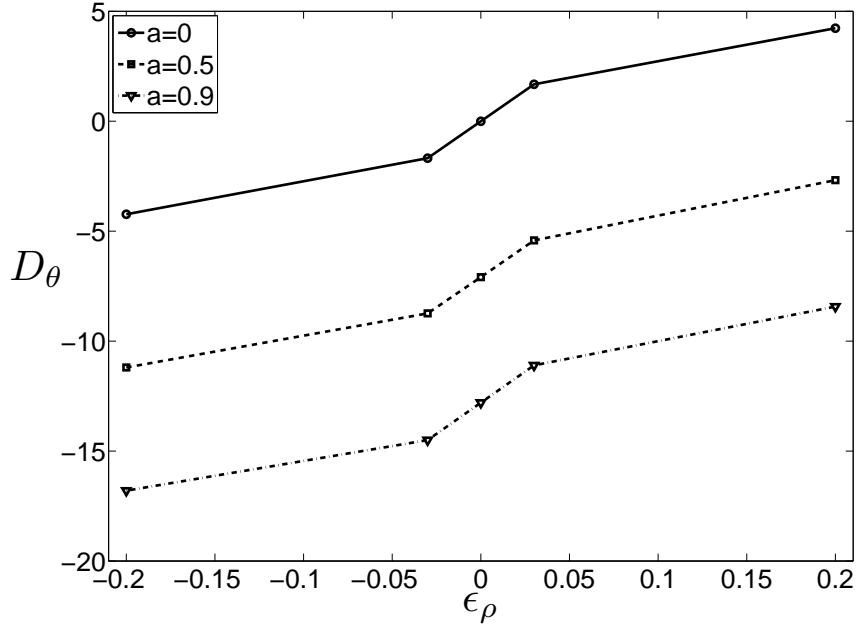


Figure 8.42: Plot of drag angle in the xy -plane against density perturbation, varying the spin of the black hole. The wind direction is in the equatorial plane of the black hole in all cases, i.e. along the x -axis with $\theta_w = 0$.

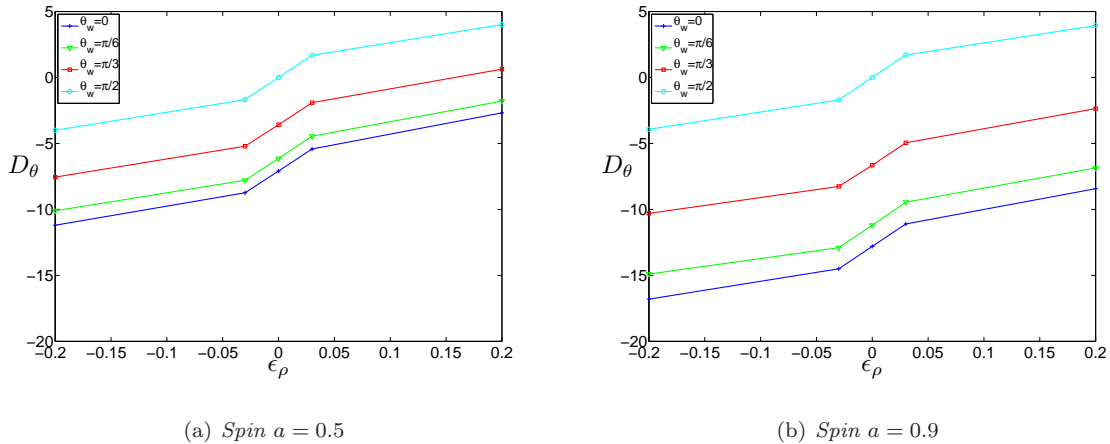


Figure 8.43: Plot of drag angles against density perturbation. The angle is measured in the plane containing the y -axis and the wind direction. The labels correspond to the angle between the equatorial plane of the black hole and the wind direction.

8.4.3 Independence of initial conditions

Although we have consistently found steady final states of the flows that we have simulated, all of these have derived from the same initial conditions. In order to confirm the universality of our results, we should test whether changing the setup of the model part way through a simulation affects its final state.

We examine two scenarios. Firstly, we test whether a tilted shock-cone, formed as the result of a perturbed flow, is resilient to a change of the upwind flow to a uniform state. Secondly, we test the effect on the flow of giving the black hole an extra kick of spin.

Perturbed flow reverting to a uniform flow

The scenario we test here is that of a perturbed flow that has settled down to steady-state having its upwind inflow conditions reverted to uniform flow. We use as initial conditions for our test the steady-state solution given by a black hole with spin parameter $a = 0.9$ and density perturbation $\epsilon_\rho = 0.2$, with $\theta_w = 0$, and change the inflow boundary condition to be that for an unperturbed flow, $\epsilon_\rho = 0$.

It turns out that the final state of the simulation is identical to the state that resulted from specifying $\epsilon_\rho = 0$ initially. The plots are identical to Figure 8.23b,d and therefore we do not show them. The way the flow pattern changes in time is also unremarkable, in that the shock-cone smoothly changes to its final position, and the two parts of the divided shock-cone merge smoothly to form a single one.

In Figure 8.44 we show the rate of mass accretion resulting from this simulation. Up to time $t = 300M$ the density is perturbed by $\epsilon_\rho = 0.2$, and after that it reverts to $\epsilon_\rho = 0$. The solid line shows the final mass accretion rate from the simulation where the density was not initially perturbed. The kink in the rate at $t \approx 350M$ corresponds to the time when the effect of the changed upstream conditions reaches the black hole (recall that the outer boundary is at $r = 50M$). Following that, the accretion rate then climbs monotonically to its final value.

Spinning-up a previously non-spinning black hole

In the close encounter of two black holes, it is possible that they will interact so as to result in a black hole with a spin somewhat larger than before the interaction. As a simple model of this we examine what happens to the steady-state accretion flow when the spin of the black hole is instantaneously increased.

The specific setup we use is that of the case where a greatly perturbed fluid with $\epsilon_\rho = 0.2$ is accreting onto a non-spinning black hole, and the flow is in steady state. We then change the black hole to have spin $a = 0.9$ and allow the simulation to again reach a steady state. At the change-over point, the fluid variables were converted to their primitive form in the original metric, and then converted to conserved form in the new metric with spin.

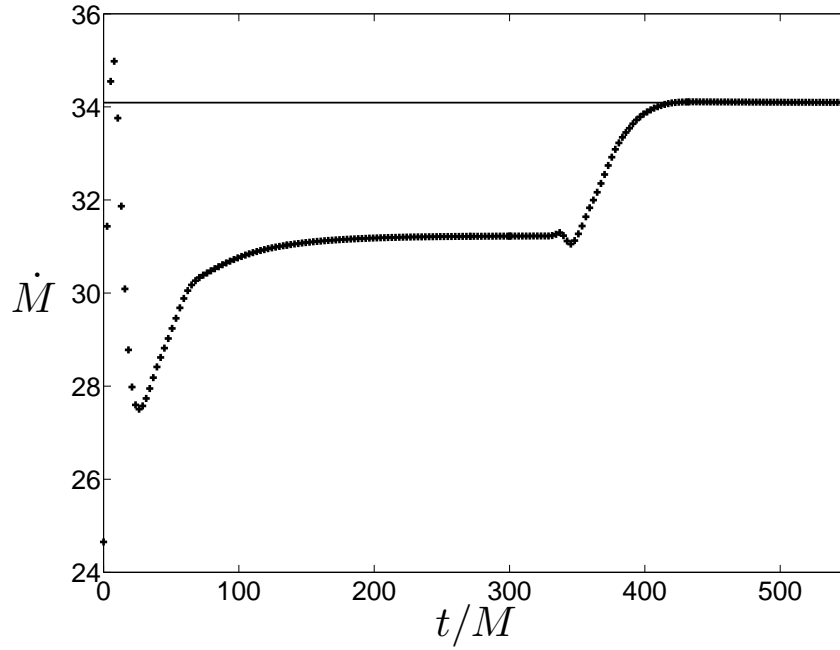


Figure 8.44: *Mass accretion rate plot for an initially perturbed model with $\epsilon_\rho = 0.2$, but reverting to $\epsilon_\rho = 0$ upstream at time $t = 300M$. The black hole has spin $a = 0.9$ and the wind is in the equatorial plane. The solid line shows the final mass accretion rate from the equivalent flow with $\epsilon_\rho = 0$ throughout.*

As with the previous example, the final state of the simulation is identical to that resulting from the simulation that maintained $a = 0.9$ throughout, as seen in Figure 8.27b,d. We show the results for the mass accretion rate in Figure 8.45. We see a discontinuity in the mass accretion rate at the time when the black hole is “kicked” ($t = 300M$). The accretion rate falls slightly below its final value, then rises above it and oscillates slightly before quickly reaching a steady state.

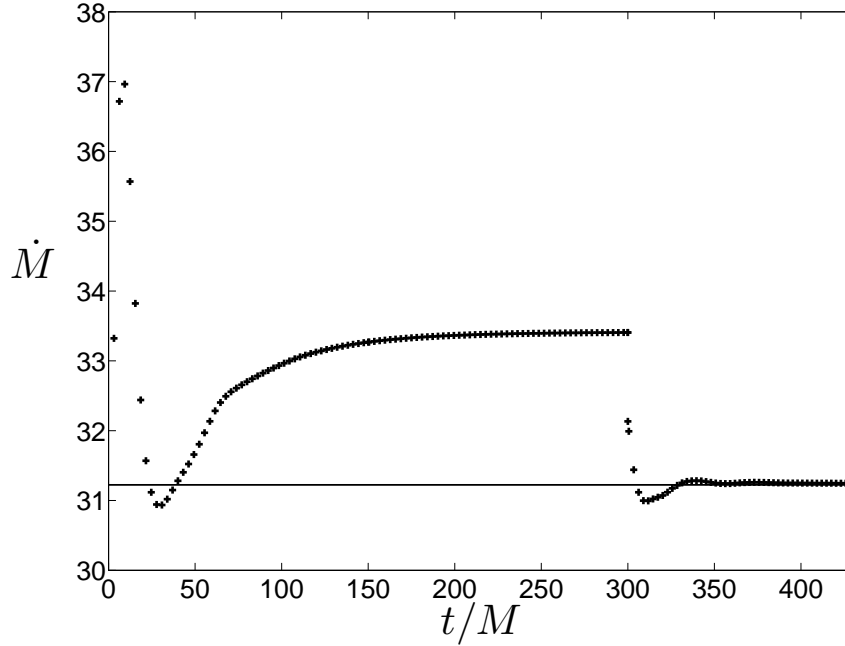


Figure 8.45: *Mass accretion rate plot for a perturbed density flow onto a non-spinning black hole which receives a kick at time $t = 300M$, increasing its spin to $a = 0.9$. The solid line shows the final mass accretion rate from the equivalent flow with $a = 0.9$ throughout.*

8.5 Conclusions

In this chapter, we have demonstrated the effects of introducing spin and a density perturbation into Bondi-Hoyle-Lyttleton accretion simulations. We have investigated the effect of these on both subsonic and supersonic flows, before applying an extensive parametric study to one supersonic base model.

We have seen that the qualitative effects of adding spin and having a perturbed density upstream of the black hole are independent of the velocity, Mach number, and adiabatic index of the fluid. In all cases presented in §8.2 we saw that the flow was wrapped around the black hole in the spin direction and that the fluid developed regions of lower density, as well as being deflected by the black hole. The extent to which these effects were seen was, however, dependent on the flow properties. In particular, only the subsonic models demonstrated any large change in the flow upstream of the black hole, as we would expect.

Investigating the supersonic model UB1 more extensively, we discovered how the separate effects of spin and density perturbation affected the flow structure.

- For a uniform flow, the effect of the spin on the flow structure is mostly restricted to the region close to the black hole, where the points at which the shock-cone meets the horizon are pulled round in

the spin direction. The mass accretion rate reduces slightly as the spin rate increases, but not to any great extent.

- Moving the flow direction from being in the equatorial plane to being along the spin axis removes any deflection of the flow by the black hole, and also slightly reduces the mass accretion rate.
- Perturbing the density changes the flow most dramatically. The flow is appreciably deflected from its original direction, and this effect persists to large distances downstream of the black hole. Vortices develop in the flow, on either side of the plane containing the wind and perturbation directions. The mass accretion rate changes by as much as 10%, although the sign of the change and its magnitude depends strongly on the spin of the black hole, and the angle of the incoming wind to the spin axis.
- The angle of the shock-cone to the wind-direction, at large distances downstream of the black hole, is independent of the spin of the black hole and the angle of the wind direction to the spin-axis. However, it is strongly dependent on the extent to which the density is perturbed, varying by about 30° over the range of ϵ_ρ that we use.
- The angle at which the black hole is dragged, relative to the wind direction, depends strongly on the spin, density perturbation, and angle of wind to the spin-axis. The drag angle in general increases with spin, and is largest when the wind direction is perpendicular to the spin-axis. The relative dependence of the drag angle on the density perturbation is independent of the spin and wind direction.
- All the flows we have tested are steady, although this was expected, as none of them approached the region which Ruffert [38] has identified as possibly containing unstable flows.
- The final flow structures are apparently independent of any past history of the flow. This was demonstrated by changing firstly the perturbation and secondly the spin of a flow after it had settled into its steady state, and seeing that the flow rapidly became identical to the flow that would have developed had the parameters not begun with separate values.

8.6 Wind Accretion onto a Binary Black Hole system

As a further test to demonstrate the full range of our algorithm’s capabilities, we perform a test of wind accretion onto a binary black hole system. We use the Brill-Lindquist data from §3.9, and keep it stationary. This is not physically accurate, as we should be evolving the metric to allow the black holes to orbit each other. However, since our algorithm is not capable of evolving the metric sufficiently accurately to perform this, we leave it stationary.

8.6.1 Simulation setup

As our flow parameters, we use adiabatic index $\Gamma = 4/3$, Mach number $\mathcal{M} = 1.5$, and asymptotic upstream sound speed $c_\infty = 0.1$. Font and Ibáñez [41] show that when accreting onto a single black hole, this flow results in a wide shock-cone which is almost detached from the upwind side of the horizon.

We set our initial metric from equation (3.73), where we specify the black holes to have unit mass and be positioned at $(0, \pm 5, 0)$, with the flow coming along the x -axis. We evolve the flow to a time $t = 2000M$, and note that the time to merger of a pair of equal mass black holes at this separation is of the same order as the simulation time, so there are not any grounds for suggesting that such a flow might develop on a time scale much shorter than it takes the black holes to move appreciably.

The simplest grid structure we can use for this problem is to have a pair of grids covering a hollow sphere for one of the black holes, with its outer boundary at a radius of $50M$, and then a second pair of grids for the second black hole, with a smaller outer radius so that they lie entirely inside the first pair of grids. A cross-section of this grid is shown in Figure 8.46. Grid lines from all four grids can be seen. Portions of the blue grid have been removed in order to allow the red and purple grids to take precedence, so that the upper black hole can be excised. The cell-sizes where the grids overlap are not necessarily identical, due to the radial scaling. This is an issue particularly around the lower black hole, where the green grid cells are very small, but are interpolating with the relatively large red cells. However, the cells to either side, at $(\pm 8, 0)$, are of similar size where they overlap.

However, this grid structure is not symmetric with respect to the black holes in two ways. Firstly, the cell-sizes where the grids interpolate from each other do not match, so that the interpolation may not be as accurate as it could be. Secondly, the spherical outer boundary is centred on the first black hole, so that the boundary conditions are not symmetric with respect to the second black hole. If we had an infinite spatial domain, this latter point would not matter, but for a finite computational domain, it could be important.

In practice, we found that evolving the flow on this grid system was successful up to about $t = 1000M$. However, the flow then became unstable, in an asymmetric way, so that we deduce that the reason for this was the asymmetric grid setup.

We therefore turned to the more complex, but symmetric, grid setup shown in Figure 8.47. Here, we have used two small pairs of grids that do not overlap to excise the two black holes, covered them with a Cartesian grid, and finally encased the Cartesian grid in a third pair of spherical grids, centred at the origin, and extending out to $50M$. Portions of the (green) Cartesian grid have been removed in order to allow the two excision spheres to be captured smoothly. The outer pair of spherical grids extend into the inner black hole grids; this should be avoided in general, but removing this would require the Cartesian grid to be made much larger, resulting in a larger number of grid cells.

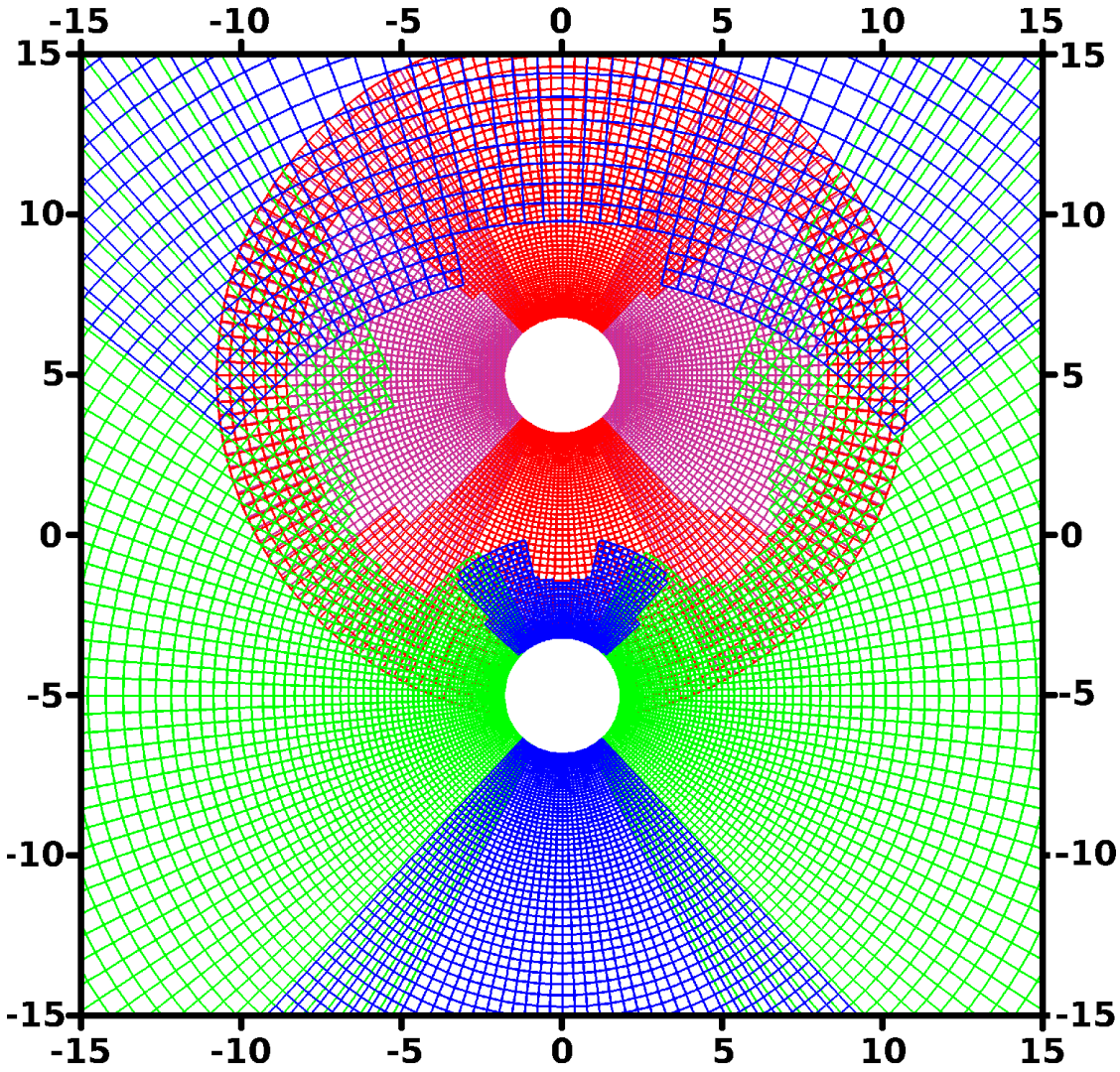


Figure 8.46: *Cross-section of the first grid structure used for the binary black hole simulation, in the $z = 0$ plane. The two black holes are centred at $(0, \pm 5, 0)$. One pair of grids (green and blue) cover a sphere that captures both the inner excision for one black hole and the outer spatial boundaries, and the second pair of grids (red and purple) capture the inner excision boundary for the second black hole. Total number of grid points: 2 457 504, total number of interpolation points: 244 587.*

The two pairs of grids covering the black holes have inner radii $1.4M$ and outer radii $4M$ so that they do not overlap. The Cartesian grid covers the region $-10M \leq x, y, z \leq 10M$, and the outer pair of grids have inner radius $6M$ and outer radius $50M$. This grid setup is now symmetric with respect to the black holes. However, we note that it requires 70% more grid points compared to, and twice as many interpolation points as, the first grid structure.

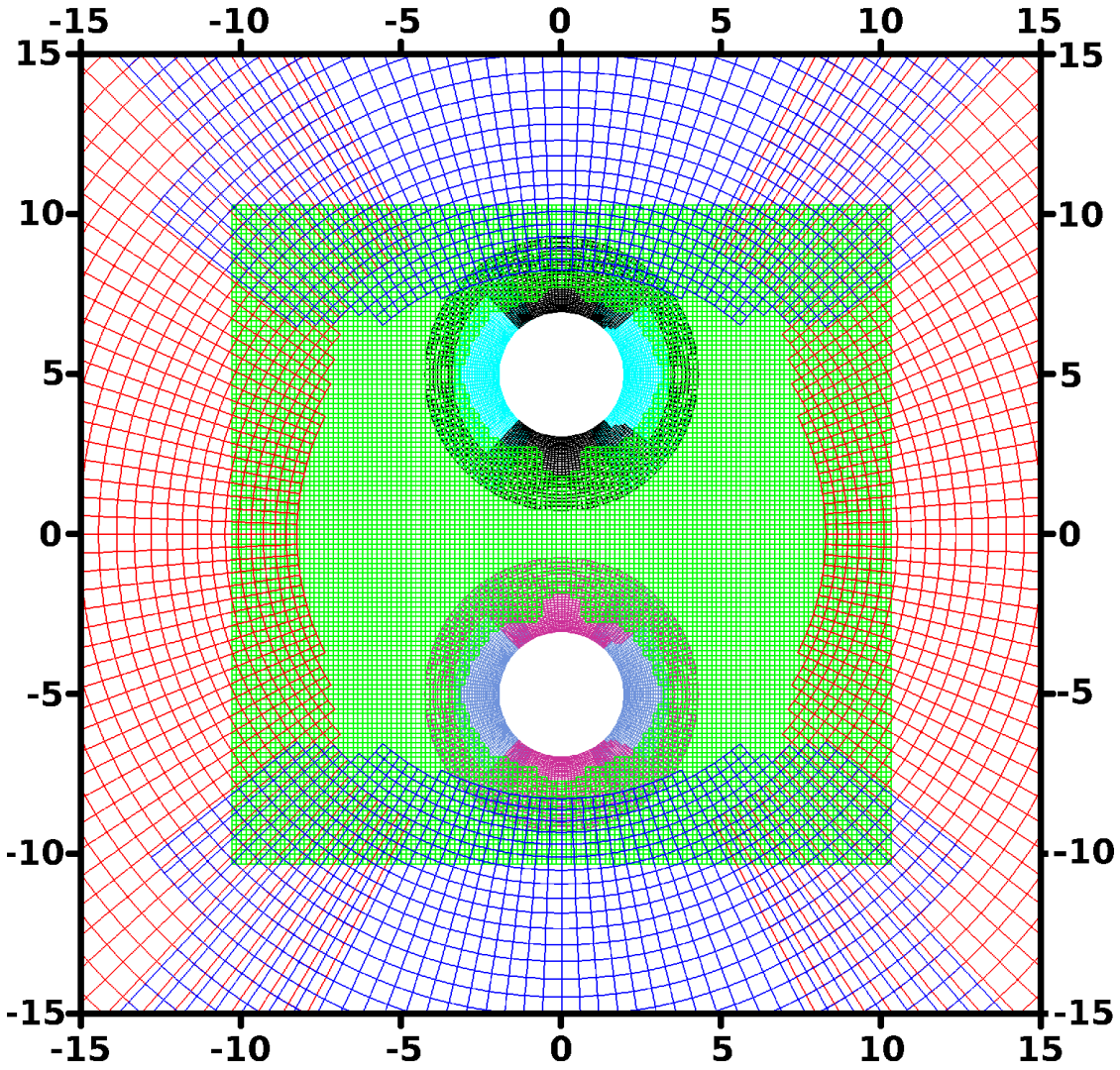


Figure 8.47: *Cross-section of the second grid structure used for the binary black hole simulation, in the $z = 0$ plane. The two black holes are centred at $(0, \pm 5, 0)$. Two pairs of spherical grids (black and cyan, and blue and magenta) are used to capture the excision boundaries of the black holes, a Cartesian grid (green) surrounds them, and a third pair of spherical grids (blue and red) is used to capture the outer spatial boundary at $50M$. Total number of grid points: 4 195 765, total number of interpolation points: 500 516.*

8.6.2 Results

A three-dimensional view of the resulting flow at $t = 2000M$ is shown in Figure 8.48. We see that a wide shock-cone has developed, although it is now detached from both of the black hole horizons. There are regions of high density to either side of the gap between the holes, as well as some regions of lower

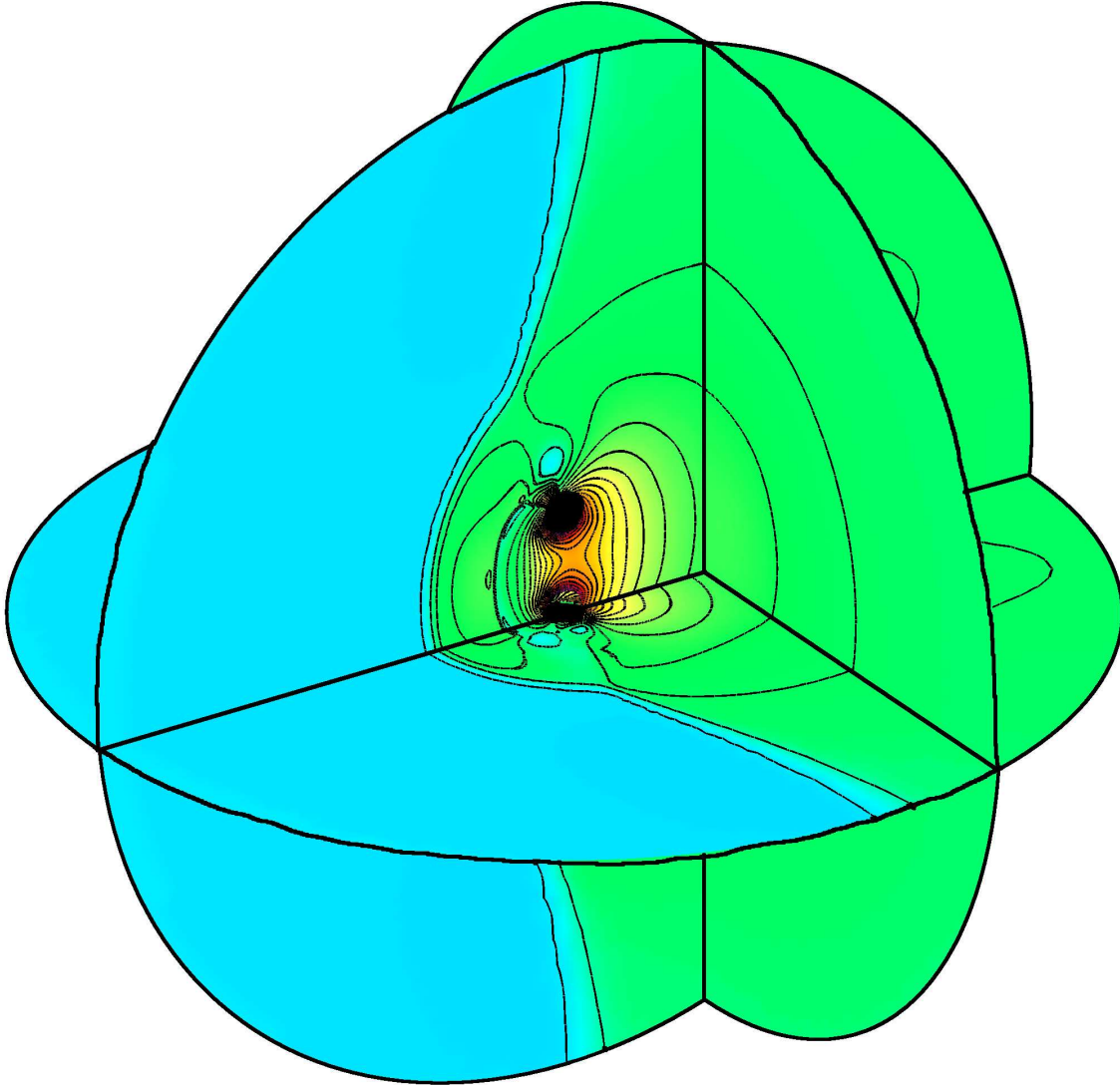


Figure 8.48: *Result of a fluid flowing onto a binary black hole system, with the metric kept stationary. There are two black holes with unit mass at $(0, \pm 5, 0)$ in the Brill-Lindquist metric. The fluid upstream of the binary system has adiabatic index $\Gamma = 4/3$, Mach number $\mathcal{M} = 1.5$, and asymptotic sound speed $c_\infty = 0.1$. We show the density, with contours evenly spaced between $\rho = 0$ and $\rho = 21$.*

density on the outside edges of the horizons.

Some more qualitative results are shown in Figure 8.49. The density plot shows the shock cone forming upstream of the black holes, with density increasing towards them. The density also increases as the black holes are approached from the downstream direction, and the region of highest density is located symmetrically between the black holes, slightly downstream of their centres. The velocity plot also shows the shock-cone, and that the flow decelerates across it. We see regions of low velocity both

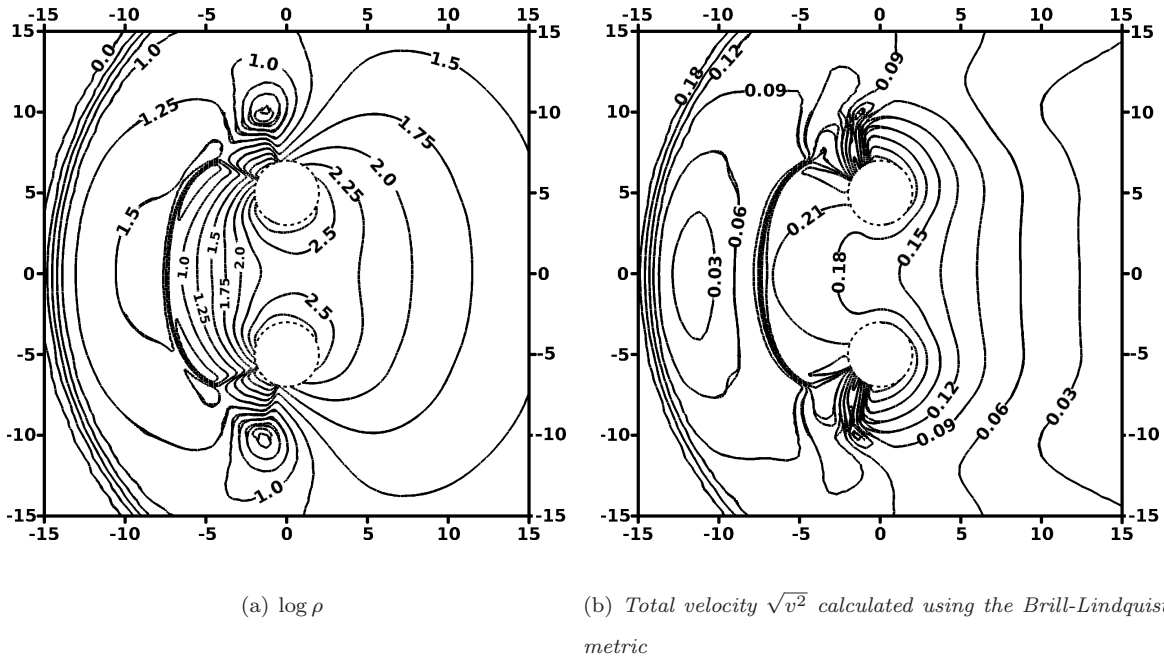


Figure 8.49: Contour plots for accretion onto a system of two unit-mass black holes at $(0, \pm 5, 0)$ using the Brill-Lindquist metric. The fluid state upstream is given by $\mathcal{M}_\infty = 1.5$, $c_\infty = 0.1$, and $\Gamma = 4/3$.

downstream and upstream of the black holes.

Similar movies to those shown for single black hole evolutions are presented on the DVD. The streamlines are calculated using the velocity vectors v^i in the Brill-Lindquist metric.

In Figure 8.50 we show the rate at which mass accretes onto the two black holes, evaluated on a sphere of radius $r = 10M$, i.e. enclosing both black holes. The simulation has clearly not reached a steady state even at $t = 2000M$.

Although the contour plots Figure 8.49 are (virtually) symmetric about the horizontal axis, the streamline plots in Figure 8.52 show that there is some problem with maintaining a symmetric flow overall as the streamlines are not symmetric about the vertical axis. The reason for this is not immediately apparent, but could either be due to asymmetries in the numerical solution not visible in the contour plots, or in the way the velocities for the streamlines are interpolated from the final numerical solution.

So, although some of the features shown in Figure 8.52 are perhaps not reliable, we can see that the fluid emerges from the binary system having been compressed in the direction perpendicular to the line joining the black holes, and forming two regions of outflow that can be interpreted as the combination of two shock-cones, one from each black hole. The accretion region shown in Figure 8.51 is only slightly distorted from being circular, having a minor axis length of $22.6M$ and a major axis length of $23.9M$.

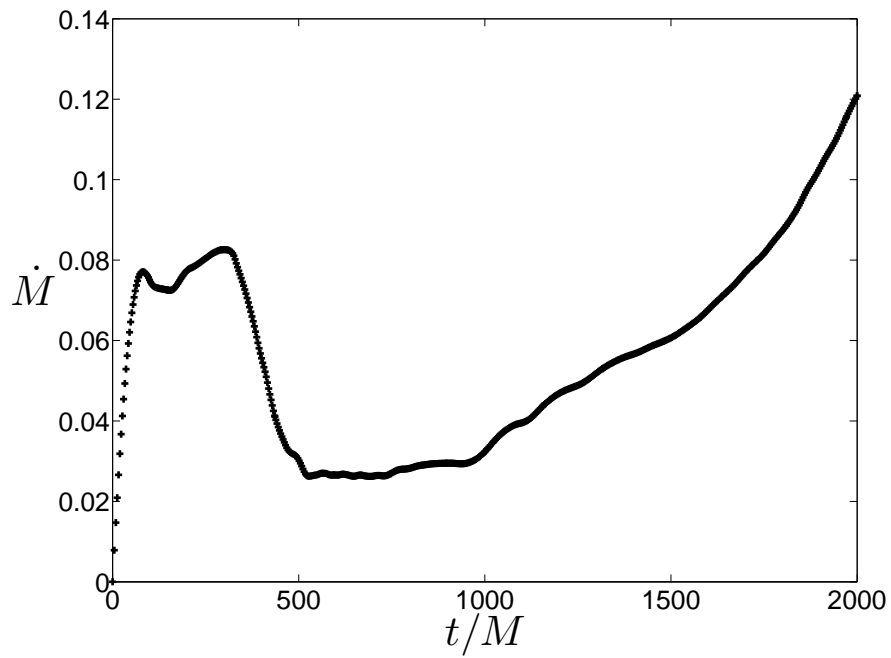


Figure 8.50: *Mass accretion rate for the binary black hole simulation. This was calculated on a sphere radius $10M$.*

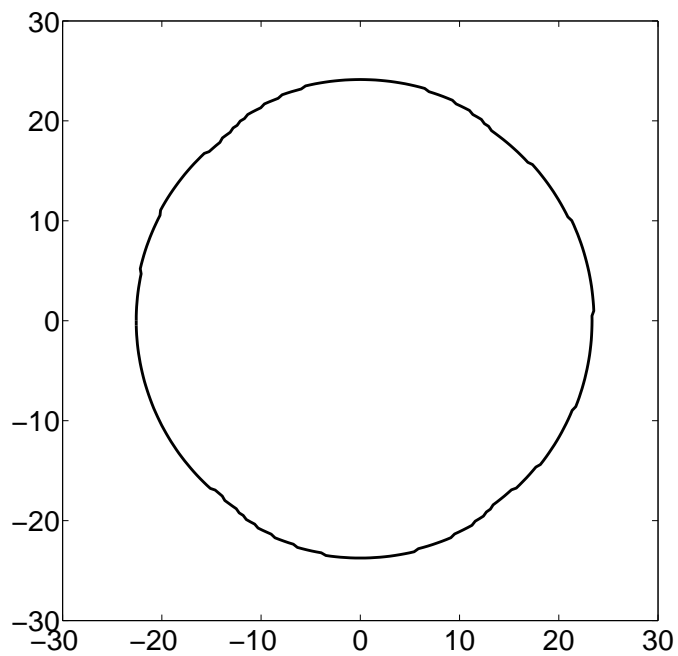


Figure 8.51: *Cross-section of the accretion region for the binary black hole simulation.*

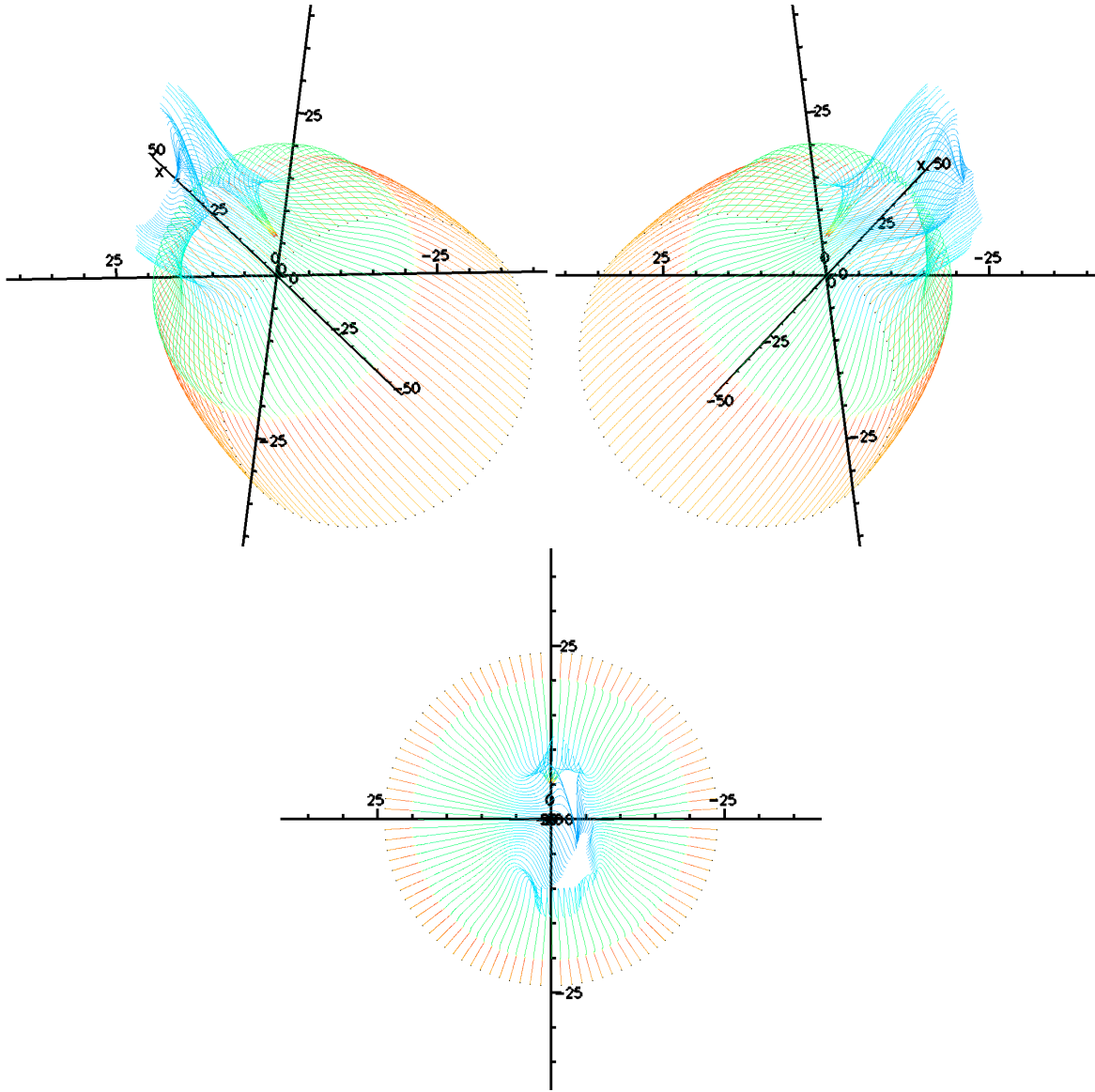


Figure 8.52: *Plot of streamlines for uniform flow onto a binary black hole system with Brill-Lindquist metric. The tracers were started on a circle of radius $24M$ at a distance $40M$ upstream of the black holes. Note that although the problem setup is symmetric in the vertical plane, the streamlines are not. This may be due to problems in the numerical solution itself, or the way in which velocities for the streamlines were interpolated from the solution data.*

8.6.3 Comments

We have shown that our methodology is capable of being applied to a problem where the grid setup is not trivial, and requires the features of Overture to generate the interpolation correctly. We have also seen that some care may be required in ensuring that the grid structure is suited to the problem being evolved. If we were to apply our algorithm to the case of inspiralling black holes, we would need to generate two moving pairs of spherical grids for the two black holes, and then a Cartesian grid that adapted itself to only just contain the two small spherical grids. The outer pair of grids going to spatial infinity could, however, remain fixed.

Chapter 9

Conclusions

In this thesis, we have demonstrated the development of a numerical algorithm that is capable of evolving a relativistic fluid on a general system of curvilinear grids on a general space-time metric. The algorithm can also be applied to the evolution of the space-time metric on similar grid systems, but the formulation we chose to perform the evolution is not sufficiently accurate or stable to compete with existing well established formulations such as those that are capable of evolving BBH systems.

As a conclusion, therefore, we provide an overview of the new work that has been performed in this thesis, and how it will be relevant to the wider NR community.

9.1 Computational infrastructure and numerical methodologies

The algorithm that has been developed through this thesis is, as far as we are aware, the first to be capable of evolving general relativistic hydrodynamical problems on general curvi-linear overlapping grids, with Adaptive Mesh Refinement. It is also capable of evolving the space-time metric, although the formulation used is not robust enough for general use.

The features of the algorithm that will be generally beneficial to Numerical Relativity are as follows:

- Curvilinear grids enable both the local and global symmetries of an evolution to be captured accurately. For example, a pair of neutron stars could be evolved with each star on its own locally spherical grid patch, and with the global domain being another sphere, capable of being extended to large radii with only a modest number of grid points, and therefore low computational expense.
- AMR will be similarly beneficial as it will enable the focusing of computational resources on areas of interest, or areas where the numerical error might otherwise be large. We note that some NR codes have the ability to use AMR, but only in the context of Cartesian grids.

- We have also demonstrated that it is possible to obtain accurate evolutions without using a Riemann solver based method. Again, this observation will be of benefit as Riemann solvers tend to be the most computationally expensive parts of a code.
- Also, in the use of Overture, we have shown that there are already available general codes that have the above features, and can be built upon to produce a mature NR code, for substantially less effort than would be needed to develop a new code from scratch.

We have documented and explained these features throughout this thesis, with the intention that other researchers will be able to short-cut some of the problems that we had to overcome in developing our algorithm and implementing it in software.

9.2 Numerical schemes

We have demonstrated the accuracy and suitability of centred finite-volume schemes such as slope-limited GFORCE for evolving strongly relativistic fluid dynamical problems. The methods have been validated on standard test problems, and shown to be of similar accuracy to other, more complex, numerical schemes.

We have also elaborated some considerations that need to be made when developing an algorithm for evolving GRHD on curvi-linear grids, and made explicit the generalisations that have to be made in this case.

Further, we have used the same schemes to evolve the space-time metric and, on application to some standard test problems, we have shown that they perform to a similar accuracy as other more mature and robust formulations, although more they would need more work before they could be applied to problems such as the merger of two black holes.

9.3 Bondi-Hoyle-Lyttleton accretion

We have applied our algorithm to the problem of perturbed Bondi-Hoyle-Lyttleton accretion onto a spinning black hole. This is the first time that such simulations have been done for full general relativity in three dimensions. We have demonstrated that we can reproduce previous results obtained in axisymmetry, and also an exact solution, to a reasonable degree of accuracy. We note that our numerical solution does not converge to the exact solution, so that more work may be required to track down this problem. However, the results are sufficiently close that we can draw valid conclusions from our other simulations.

Following two previous sets of papers on Bondi-Hoyle-Lyttleton accretion, we have combined their approaches. We first duplicated some test-cases from one of the papers, giving an overview of the types of flow structure that could develop in both subsonic and supersonic flow. We have demonstrated the effect of a spinning black hole and a perturbed upstream density on the flows, showing effects similar to those obtained by Ruffert.

We have performed a full parametric study of a supersonic flow, demonstrating the dependence of the flow structure on the fluid and black hole parameters. We have seen the development of flow features very similar to those found in the case of Newtonian Bondi-Hoyle-Lyttleton accretion, and explained the effects in terms of the flow parameters. We have done this by considering the effects of the flow parameters both separately and in combination, examining the fluid density and velocity via contour plots and streamlines. This has led us to discover some interesting effects that were not apparent in previous investigations of accretion.

We have also shown an extensive quantitative analysis of the flow, aiming to discover effects that might be observable in distant accretion flow using current technology, although we have not found any conclusive results for this.

Finally, we have demonstrated an accretion flow onto a binary black hole system, showing that our methodology is easily applied to the solution of accretion flow on geometries more complex than a single, stationary black hole. However, this has also shown that some work may be needed in fine-tuning the grid structure used for these flows.

Chapter 10

Future Work

The implementation of the algorithm that we have written for this thesis is so far only a demonstration of the full potential of the methodologies included in the algorithm. However, our choice to build on the Overture infrastructure was with a more long-term purpose in mind. Having demonstrated that the principles are sound, we can now continue to develop a mature code for evolving general space-times containing various different fluid types.

10.1 Code structure

Now that we have determined which numerical features are the correct ones to use, and how we should implement them, we should reimplement the algorithm from scratch. This could be done in a modular way, in a similar fashion to Cactus [5], so that the incorporation of new numerical methods, formulations, or equations into the algorithm would be relatively straightforward, and could be done independently of the adaptations required to use curvi-linear overlapping grids.

It would be possible, within this framework, to implement the most up-to-date formulations of Einstein's equations, with various choices for gauge conditions. Various different fluid types could also be included in the algorithm. These would require very little effort to include in such a modular code, if we do not require Riemann solvers to be implemented for each new system of equations, something we have demonstrated to be feasible and to produce good results.

10.2 Numerical Methods

Similarly, more up-to-date numerical methods could be implemented, again in a modular fashion, so that we can take advantage of the latest developments in this field. Further, we could introduce magnetic

fields into our algorithm, although this would require some work, such as the inclusion of constrained transport, to ensure that the magnetic field constraints were preserved.

Features that are implemented by Overture that we envisage could be useful for NR include moving grids and automatic grid generation. These would permit a general space-time to be evolved, with a suitable grid being generated that would take into account, for example, the location of any apparent horizons, placing an excision boundary just inside the horizon, and creating a grid that locally was of similar shape to that horizon. This would be of use, for example, in the simulation of a fluid in a BBH system. The advantage of a completely general grid system as made possible by Overture would be to remove the necessity of developing either specially designed grid systems for each space-time one wished to evolve, or special coordinate systems that keep the predicted features of the space-time aligned with a static grid. In a conceptually similar way to AMR, moving grids would allow the grid to adapt to the movement of singularities through the space-time.

Numerical techniques that could be included in such an algorithm, beyond those we have already demonstrated in this thesis, would be proper flux correction at AMR boundaries, and the ability for more general AMR setups such as varying refinement ratios, and different ways of interpolating sub-cycled grid levels' boundaries.

Given the potential for the use of a spherical grid at large radii, more appropriate boundary conditions could be implemented, and better extraction of gravitational waves could be performed as compared to evolutions carried out on Cartesian grids. Further, given that Overture is capable of solving elliptic equations on the general grid structure that it uses, via PETSc [14], the constraint equations could be solved, either just initially, or after every few time-steps.

Also, it might be possible to derive flux-source balanced schemes for evolving a fluid on a general metric on a general curvi-linear grid, so that a static solution would remain static. However, as well as requiring large amounts of effort to derive such a scheme, any resulting algorithm would quite likely introduce a large amount of computational expense into the scheme. Throughout this thesis, we have not seen the necessity for such an approach, although there might be some cases where non-balancing effects became important.

10.3 Bondi-Hoyle-Lyttleton Accretion

We have only simulated a small part of the parameter space of Bondi-Hoyle-Lyttleton accretion. As suggested by our problem setup, we envisage testing the effect of velocity perturbations. We can also vary the direction in which the perturbations occur, as allowed for by equation (7.6).

As suggested by Foglizzo *et al.* [37], it would be interesting to investigate the stability of accretion

onto a black hole, and whether relativistic effects stabilise such a flow, or whether the existence of the instability is affected by the spin of the black hole.

10.4 Final remarks

In conclusion, therefore, we believe that we have demonstrated that an algorithm making use of general overlapping grids with AMR has many advantages for solving problems of General Relativity, and that it does not require prohibitively large amounts of effort to implement such an algorithm, particularly if use is made of mature tools and software libraries widely available in the CFD community.

Bibliography

- [1] Miguel Alcubierre, *The Status of Numerical Relativity*, General Relativity and Gravitation (P. Florides, B. Nolan, and A. Ottewill, eds.), 2005.
- [2] Miguel Alcubierre, Gabrielle Allen, Carles Bona, David Fiske, Tom Goodale, F Siddhartha Guzmán, Ian Hawke, Scott H Hawley, Sascha Husa, Michael Koppitz, Christiane Lechner, Denis Pollney, David Rideout, Marcelo Salgado, Erik Schnetter, Edward Seidel, Hisa-aki Shinkai, Deirdre Shoemaker, Béla Szilágyi, Ryoji Takahashi, and Jeff Winicour, *Towards standard testbeds for numerical relativity*, Classical and Quantum Gravity **21** (2004), no. 2, 589–613.
- [3] Miguel Alcubierre, Bernd Brügmann, Peter Diener, Michael Koppitz, Denis Pollney, Edward Seidel, and Ryoji Takahashi, *Gauge conditions for long-term numerical black hole evolutions without excision*, Physical Review D (Particles, Fields, Gravitation, and Cosmology) **67** (2003), no. 8, 084023.
- [4] Miguel Alcubierre, Bernd Brügmann, Thomas Dramlitsch, José A Font, Philippos Papadopoulos, Edward Seidel, Nikolaos Stergioulas, and Ryoji Takahashi, *Towards a stable numerical evolution of strongly gravitating systems in general relativity: The conformal treatments*, Physical Review D **62** (2000), 044034.
- [5] Gabrielle Allen, Yaakoub El-Khamra, Tom Goodale, Thomas Radke, David Rideout, Erik Schnetter, Jian Tao, and Elena Caraba, *Cactus Code web-site*, www.CactusCode.org, 2009, Accessed 6th May 2009.
- [6] Matthew Anderson, Eric W. Hirschmann, Luis Lehner, Steven L. Liebling, Patrick M. Motl, David Neilsen, Carlos Palenzuela, and Joel E. Tohline, *Magnetized Neutron-Star Mergers and Gravitational-Wave Signals*, Phys. Rev. Lett. **100** (2008), no. 19, 191101.
- [7] ———, *Simulating binary neutron stars: Dynamics and gravitational waves*, Physical Review D **77** (2008), no. 2, 024006.

- [8] Peter Anninos, P. Chris Fragile, and Jay D. Salmonson, *Cosmos++: Relativistic Magnetohydrodynamics on Unstructured Grids with Local Adaptive Refinement*, *Astrophysical Journal* (2005), no. 635, 723–740.
- [9] K A Arnaud, S Babak, J G Baker, M J Benacquista, N J Cornish, C Cutler, L S Finn, S L Larson, T Littenberg, E K Porter, M Vallisneri, A Vecchio, and J-Y Vinet (The Mock LISA Data Challenge Task Force), *An overview of the second round of the Mock LISA Data Challenges*, *Classical and Quantum Gravity* **24** (2007), no. 19, S551–S564.
- [10] E. Babichev, S. Chernov, V. Dokuchaev, and Yu. Eroshenko, *Ultra-hard fluid and scalar field in the Kerr-Newman metric*, *Physical Review D* **78** (2008), no. 10, 104027.
- [11] M. C. Babiuc, S. Husa, D. Alic, I. Hinder, C. Lechner, E. Schnetter, B. Szilagyi, Y. Zlochower, N. Dorband, D. Pollney, and J. Winicour, *Implementation of standard testbeds for numerical relativity*, *Classical and Quantum Gravity* **25** (2008), no. 12, 125012.
- [12] L. Baiotti, I. Hawke, P. J. Montero, and L. Rezzolla, *A new three-dimensional general-relativistic hydrodynamics code*, *Memorie della Societa Astronomica Italiana Supplement* **1** (2003), 210.
- [13] Luca Baiotti, Bruno Giacomazzo, and Luciano Rezzolla, *Accurate evolutions of inspiralling neutron-star binaries: Prompt and delayed collapse to a black hole*, *Physical Review D* **78** (2008), no. 8, 084033.
- [14] Satish Balay, William D. Gropp, Lois C. McInnes, and Barry F. Smith, *PETSc: Portable, Extensible Toolkit for Scientific Computation*, <http://www.mcs.anl.gov/petsc>, 2008.
- [15] Enrico Barausse and Luciano Rezzolla, *Influence of the hydrodynamic drag from an accretion torus on extreme mass-ratio inspirals*, *Physical Review D* **77** (2008), 104027.
- [16] A. P. Barnes, P. G. Lefloch, B. G. Schmidt, and J. M. Stewart, *The Glimm scheme for perfect fluids on plane-symmetric Gowdy spacetimes*, *Classical and Quantum Gravity* **21** (2004), no. 22, 5043–5074.
- [17] Anita P. Barnes, *Numerical Relativistic Hydrodynamics in Planar and Axisymmetric Spacetimes*, Ph.D. thesis, Department of Applied Mathematics and Theoretical Physics, Cambridge, U.K., 2004.
- [18] Thomas W. Baumgarte and Stuart L. Shapiro, *Numerical integration of Einstein’s field equations*, *Physical Review D* **59** (1998), no. 2, 024007.
- [19] Marsha J. Berger and Joseph Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations*, *Journal of Computational Physics* **53** (1984), no. 3, 484–512.

- [20] S.J. Billett and E.F. Toro, *On WAF-Type schemes for Multidimensional Hyperbolic Conservation Laws*, Journal of Computational Physics **130** (1997), no. 1, 1–24.
- [21] C. Bona, J. Massó, E. Seidel, and J. Stela, *First order hyperbolic formalism for numerical relativity*, Physical Review D **56** (1997), no. 6, 3405–3415.
- [22] Carles Bona, Joan Masso, Edward Seidel, and Paul Walker, *Three Dimensional Numerical Relativity with a Hyperbolic Formulation*, arXiv.org:gr-qc/9804052, 1998.
- [23] Silvano Bonazzola, Ericourgoulhon, Philippe Grandclément, and Jérôme Novak, *Constrained scheme for the Einstein equations based on the Dirac gauge and spherical coordinates*, Physical Review D **70** (2004), no. 10, 104007.
- [24] Robert H. Boyer and Richard W. Lindquist, *Maximal Analytic Extension of the Kerr Metric*, Journal of Mathematical Physics **8** (1967), no. 2, 265–281.
- [25] Steven Brandt and Bernd Brügmann, *A Simple Construction of Initial Data for Multiple Black Holes*, Physical Review Letters **78** (1997), no. 19, 3606–3609.
- [26] Gioel Calabrese and David Neilsen, *Spherical excision for moving black holes and summation by parts for axisymmetric systems*, Physical Review D (Particles, Fields, Gravitation, and Cosmology) **69** (2004), no. 4, 044020.
- [27] Tony Chu, Harald P. Pfeiffer, and Mark A. Scheel, *High accuracy simulations of black hole binaries: spins anti-aligned with the orbital angular momentum*, arXiv.org:gr-qc/0909.1313, 2009.
- [28] LIGO Scientific Collaboration, *LIGO – Laser Interferometer Gravitational Wave Observatory*, <http://www.ligo.org>, 2009, Accessed 6th May 2009.
- [29] Gregory B. Cook, *Initial data for numerical relativity*, Living Rev. Relativity **3** (2001), no. 5, Cited: 31st August 2006.
- [30] Matthew D. Duez, Francois Foucart, Lawrence E. Kidder, Harald P. Pfeiffer, Mark A. Scheel, and Saul A. Teukolsky, *Evolving black hole-neutron star binaries in general relativity using pseudospectral and finite difference methods*, Physical Review D **78** (2008), no. 10, 104015.
- [31] Matthew D. Duez, Stuart L. Shapiro, and Hwei-Jang Yo, *Relativistic hydrodynamic evolutions with black hole excision*, Physical Review D **69** (2004), no. 10, 104016.
- [32] M.D. Duez, F. Foucart, L.E. Kidder, C.D. Ott, and S.A. Teukolsky, *Equation of state effects in black hole-neutron star mergers*, arXiv.org:0912.3528, 2009.

- [33] Michael Dumbser and Olindo Zanotti, *Very high order PNP schemes on unstructured meshes for the resistive relativistic MHD equations*, Journal of Computational Physics **228** (2009), no. 18, 6991–7006.
- [34] Richard Edgar, *A review of Bondi-Hoyle-Lyttleton accretion*, New Astronomy Reviews **48** (2004), no. 10, 843–859.
- [35] Benjamin Aylott et al., *Testing gravitational-wave searches with numerical relativity waveforms: results from the first Numerical INjection Analysis (NINJA) project*, Classical and Quantum Gravity **26** (2009), no. 16, 165008 (51pp).
- [36] Frits Eulerink and Garrelt Mellema, *General relativistic hydrodynamics with a Roe solver*, Astronomy and Astrophysics Supplement **110** (1995), 587–623.
- [37] T. Foglizzo, P. Galletti, and M. Ruffert, *A fresh look at the unstable simulations of Bondi-Hoyle-Lyttleton accretion*, Astronomy and Astrophysics **435** (2005), no. 2, 397–411.
- [38] T. Foglizzo and M. Ruffert, *An analytic study of Bondi-Hoyle-Lyttleton accretion. II. Local stability analysis*, Astronomy and Astrophysics **347** (1999), 901–914.
- [39] José A. Font, *Numerical Hydrodynamics and Magnetohydrodynamics in General Relativity*, Living Reviews in Relativity **11** (2008), no. 7, Cited: 6th May 2009.
- [40] José A. Font and José M. Ibáñez, *Non-axisymmetric relativistic Bondi-Hoyle accretion onto a Schwarzschild black hole*, Monthly Notices of the Royal Astronomical Society **298** (1998), no. 3, 835–846.
- [41] ———, *A Numerical Study of Relativistic Bondi-Hoyle Accretion onto a Moving Black Hole: Axisymmetric Computations in a Schwarzschild Background*, Astrophysical Journal **494** (1998), no. 1, 297–316.
- [42] José A. Font, José Maria Ibáñez, and Philippos Papadopoulos, *Non-axisymmetric relativistic Bondi-Hoyle accretion on to a Kerr black hole*, Monthly Notices of the Royal Astronomical Society **305** (1999), no. 4, 920–936.
- [43] Max-Planck-Institut für Gravitationsphysik, *GEO600 - The German-British Gravitational Wave Detector*, <http://geo600.aei.mpg.de/>, 2008, Accessed 6th May 2009.
- [44] Max Planck Society for the Advancement of Science, *Apples with Apples - Numerical Relativity Comparisons and Tests*, <http://www.appleswithapples.org>, Accessed 6th May 2009.

- [45] Francois Foucart, Lawrence E. Kidder, Harald P. Pfeiffer, and Saul A. Teukolsky, *Initial data for black hole–neutron star binaries: A flexible, high-accuracy spectral method*, Physical Review D **77** (2008), no. 12, 124051.
- [46] P. Chris Fragile, Christopher C. Linder, Peter Anninos, and Jay D. Salmonson, *Application of the cubed-sphere grid to tilted black hole accretion disks*, Astrophysical Journal **691** (2009), no. 1, 482–494.
- [47] S. Frittelli and R. Gomez, *Ill-posedness in the Einstein equations*, Journal of Mathematical Physics **41** (2000), no. 8, 5535–49.
- [48] Charles F. Gammie, Jonathan C. McKinney, and Gábor Tóth, *HARM: A Numerical Scheme for General Relativistic Magnetohydrodynamics*, Astrophysical Journal **589** (2003), no. 1, 444–457.
- [49] Bruno Giacomazzo and Luciano Rezzolla, *WhiskyMHD: a new numerical code for general relativistic magnetohydrodynamics*, Classical and Quantum Gravity **24** (2007), no. 12, S235–S258.
- [50] S.K. Godunov, *A difference scheme for numerical computation of discontinuous solutions of hydrodynamics equations*, Math. Sbornik **47** (1959), 271–306.
- [51] David Goldberg, *What every computer scientist should know about floating-point arithmetic*, ACM Computing Surveys **23** (1991), no. 1, 5–48.
- [52] Ericourgoulhon, *Construction of initial data for 3+1 numerical relativity*, Journal of Physics: Conference Series **91** (2007), 012001 (28pp).
- [53] Carsten Gundlach, Gioel Calabrese, Ian Hinder, and Jose M Martin-Garcia, *Constraint damping in the Z4 formulation and harmonic gauge*, Classical and Quantum Gravity **22** (2005), no. 17, 3767–3773.
- [54] Carsten Gundlach and Paul Walker, *Causal differencing of flux-conservative equations applied to black hole spacetimes*, Classical and Quantum Gravity **16** (1999), no. 3, 991–1010.
- [55] Mark Hannam, Sascha Husa, John G. Baker, Michael Boyle, Bernd Brügmann, Tony Chu, Nils Dorband, Frank Herrmann, Ian Hinder, Bernard J. Kelly, Lawrence E. Kidder, Pablo Laguna, Keith D. Matthews, James R. van Meter, Harald P. Pfeiffer, Denis Pollney, Christian Reisswig, Mark A. Scheel, and Deirdre Shoemaker, *Samurai project: Verifying the consistency of black-hole-binary waveforms for gravitational-wave detection*, Physical Review D **79** (2009), no. 8, 084025.
- [56] Amiram Harten, Peter D. Lax, and Bram van Leer, *On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws*, SIAM Review **25** (1983), no. 1, 35–61.

- [57] William Henshaw, Kyle Chand, Petri Fast, and Dan Quinlan, *Overture – Object-oriented Tools for solving PDEs in Complex Geometries*, <http://www.llnl.gov/CASC/Overture/>, 2009, Accessed 6th May 2009.
- [58] Ian Hinder, *The Current Status of Binary Black Hole Simulations in Numerical Relativity*, arXiv.org:gr-qc/1001.5161, 2010.
- [59] Michael Holst, Lee Lindblom, Robert Owen, Harald P. Pfeiffer, Mark A. Scheel, and Lawrence E. Kidder, *Optimal constraint projection for hyperbolic evolution systems*, Physical Review D **70** (2004), no. 8, 084017.
- [60] M.E. Hubbard and P. Garcia-Navarro, *Flux Difference Splitting and the Balancing of Source Terms and Flux Gradients*, Journal of Computational Physics **165** (2000), no. 1, 89–125.
- [61] Piotr Jaranowski and Gerhard Schäfer, *Lapse function for maximally sliced Brill-Lindquist initial data*, Physical Review D **65** (2002), no. 12, 127501.
- [62] Akira Kageyama, *Dissection of a Sphere and Yin-Yang Grids*, Journal of the Earth Simulator **3** (2005), 20–28.
- [63] Bernard Kelly, *Exact Solutions in the 3+1 Split*, www.dorkly.net/beany/physics/documentation/exact_solutions.pdf, January 2007.
- [64] Donald E. Knuth, *Structured programming with go to statements*, ACM Computing Surveys **6** (1974), no. 4, 261–301.
- [65] H-O Kreiss, O Reula, O Sarbach, and J Winicour, *Well-posed initial-boundary value problem for the harmonic Einstein equations using energy estimates*, Classical and Quantum Gravity **24** (2007), no. 23, 5973–5984.
- [66] H-O Kreiss and J Winicour, *Problems which are well posed in a generalized sense with applications to the Einstein equations*, Classical and Quantum Gravity **23** (2006), no. 16, S405–S420.
- [67] C.B. Laney, *Computational Gasdynamics*, Cambridge University Press, 1998.
- [68] Luis Lehner, *Numerical relativity: a review*, Classical and Quantum Gravity **18** (2001), no. 17, R25–R86.
- [69] Randall J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2002.

- [70] *LISA - Laser Interferometer Space Antenna*, <http://lisa.nasa.gov>, 2007, Accessed 6th May 2009.
- [71] A. Lucas-Serrano, J.A. Font, J.M. Ibáñez, and J.M. Martí, *Assessment of a high-resolution central scheme for the solution of the relativistic hydrodynamics equations*, *Astronomy and Astrophysics* **428** (2004), 703–715.
- [72] Pedro Marronetti, *Hamiltonian relaxation*, *Classical and Quantum Gravity* **22** (2005), no. 12, 2433–2451.
- [73] ———, *Momentum constraint relaxation*, *Classical and Quantum Gravity* **23** (2006), no. 7, 2681–2695.
- [74] José Maria Martí and Ewald Müller, *Numerical Hydrodynamics in Special Relativity*, *Living Reviews in Relativity* **6** (2003), no. 7, Cited: 6th May 2009.
- [75] Charles W. Misner, Kip S. Thorne, and John A. Wheeler, *Gravitation*, W.H. Freeman & Co. Ltd, 1973.
- [76] Pedro J. Montero, José A. Font, and Massaru Shibata, *Nada: A new code for studying self-gravitating tori around black holes*, *Physical Review D* **78** (2008), no. 6, 064037.
- [77] Claudia Moreno, Darío Núñez, and Olivier Sarbach, *Kerr-Schild-type initial data for black holes with angular momenta*, *Classical and Quantum Gravity* **19** (2002), 6059.
- [78] David W. Neilsen, *Extremely Relativistic Fluids in Strong-Field Gravity*, Ph.D. thesis, University of Texas at Austin, 1999, <http://laplace.physics.ubc.ca/~matt/Doc/Theses/Phd/neilsen.pdf>.
- [79] Nicholas Nethercote and Julian Seward, *Valgrind: a framework for heavyweight dynamic binary instrumentation*, *ACM SIGPLAN Notices* **42** (2007), no. 6, 89–100.
- [80] European Gravitational Observatory, *VIRGO - A Gravitational Waves Antenna*, <http://www.virgo.infn.it/>, Accessed 6th May 2009.
- [81] Enrique Pazos, Manuel Tiglio, Matthew D. Duez, Lawrence E. Kidder, and Saul A. Teukolsky, *Orbiting binary black hole evolutions with a multipatch high order finite-difference approach*, *Physical Review D* **80** (2009), no. 2, 024027.
- [82] L. I. Petrich, S. L. Shapiro, R. F. Stark, and S. A. Teukolsky, *Accretion onto a moving black hole - A fully relativistic treatment*, *Astrophysical Journal* **336** (1989), 313–349.

- [83] Loren I. Petrich, Stuart L. Shapiro, and Saul A. Teukolsky, *Accretion onto a moving black hole: An exact solution*, Physical Review Letters **60** (1988), no. 18, 1781–1784.
- [84] José A. Pons, José Ma Martí, and Ewald Müller, *The exact solution of the Riemann problem with non-zero tangential velocities in relativistic hydrodynamics*, Journal of Fluid Mechanics **422** (2000), no. -1, 125–139.
- [85] F. Pretorius, *Evolution of Binary Black-Hole Spacetimes*, Physical Review Letters **95** (2005), no. 12, 121101.
- [86] Dimitrios Psaltis, *Probes and Tests of Strong-Field Gravity with Observations in the Electromagnetic Spectrum*, Living Reviews in Relativity **11** (2008), no. 9.
- [87] M. Rampp and H.-T. Janka, *Radiation hydrodynamics with neutrinos*, A&A **396** (2002), no. 1, 361–392.
- [88] K.F. Riley, M.P. Hobson, and S.J. Bence, *Mathematical Methods for Physics and Engineering*, 3rd ed., Cambridge University Press, 2006.
- [89] Oliver Rinne, *Axisymmetric numerical relativity*, Ph.D. thesis, DAMTP, University of Cambridge, 2005.
- [90] James A. Rossmanith, Derek S. Bale, and Randall J. LeVeque, *A wave propagation algorithm for hyperbolic systems on curved manifolds*, Journal of Computational Physics **199** (2004), no. 2, 631 – 662.
- [91] Sheila Rowan and Jim Hough, *Gravitational Wave Detection by Interferometry (Ground and Space)*, Living Reviews in Relativity **3** (2000), no. 3, Cited on 6th May 2009.
- [92] M. Ruffert, *Three-dimensional hydrodynamic Bondi-Hoyle accretion. I. Code validation and Stationary accretors*, Astrophysical Journal **427** (1994), no. 1, 342.
- [93] ———, *Three-dimensional hydrodynamic Bondi-Hoyle accretion. IV. Specific heat ratio 4/3*, Astronomy and Astrophysics Supplement Series **113** (1995), 133.
- [94] ———, *Non-axisymmetric wind-accretion simulations I. Velocity gradients of 3% and 20% over one accretion radius*, Astronomy and Astrophysics **317** (1997), 793.
- [95] ———, *Non-axisymmetric wind-accretion simulations II. Density gradients*, Astronomy and Astrophysics **346** (1999), 861.

- [96] V. Schneider, V. Katscher, D.H. Rischke, B. Waldhauser, J.A. Marhun, and C.-D. Munz, *New Algorithms for Ultra-relativistic Numerical Hydrodynamics*, Journal of Computational Physics **105** (1993), no. 1, 92–107.
- [97] Erik Schnetter, *Gauge fixing for the simulation of black hole spacetimes*, Ph.D. thesis, Eberhard-Karls-Universität zu Tübingen, 2003.
- [98] Erik Schnetter, Peter Diener, Ernst Nils Dorband, and Manuel Tiglio, *A multi-block infrastructure for three-dimensional time-dependent numerical relativity*, Classical and Quantum Gravity **23** (2006), no. 16, S553.
- [99] Erik Schnetter, Scott H Hawley, and Ian Hawke, *Evolutions in 3D numerical relativity using fixed mesh refinement*, Classical and Quantum Gravity **21** (2004), no. 6, 1465–1488.
- [100] S.L. Shapiro, *Accretion onto black holes: The emergent radiation spectrum. III. Rotating (Kerr) black holes*, Astrophysical Journal **189** (1973), 343–351.
- [101] S.L. Shapiro and S.A. Teukolsky, *Black holes, white dwarfs and neutron stars: The physics of compact objects*, Wiley-Interscience, 1983.
- [102] Stuart L. Shapiro, *Potential flows in general relativity: Some exact solutions*, Physical Review D **39** (1989), no. 10, 2839.
- [103] Béla Szilágyi, Lee Lindblom, and Mark A. Scheel, *Simulations of Binary Black Hole Mergers Using Spectral Methods*, arxiv.org/0909.3557, 2009.
- [104] A. Tchekhovskoy, J. C. McKinney, and Ramesh Narayan, *WHAM: a WENO-based general relativistic numerical scheme - I. Hydrodynamics*, Monthly Notices of the Royal Astronomical Society **379** (2007), no. 2, 469–497.
- [105] Saul A. Teukolsky, *Stability of the iterated Crank-Nicholson method in numerical relativity*, Physical Review D **61** (2000), no. 8, 087501.
- [106] Jonathan Thornburg, *Black hole excision with multiple grid patches*, Classical and Quantum Gravity **21** (2004), 3665.
- [107] ———, *Event and Apparent Horizon Finders for 3+1 Numerical Relativity*, Living Reviews in Relativity **10** (2007), no. 3, Cited: 6th May 2009.
- [108] M. Tiglio, *Dynamical control of the constraints growth in free evolutions of Einstein’s equations*, arxiv:gr-qc/0304062, 2003.

- [109] E.F. Toro, *Riemann solvers and numerical methods for fluid dynamics*, 2nd ed., Springer-Verlag, 1999.
- [110] ———, *Multi-stage predictor-corrector fluxes for hyperbolic equations*, <http://www.newton.cam.ac.uk/preprints/NI03037.pdf>, 2003.
- [111] E.F. Toro and W. Hu, *Centred unsplit finite volume schemes for multidimensional hyperbolic conservation laws*, Godunov Methods (E.F. Toro, ed.), Kluwer Academic/Plenum Publishers, 2001, pp. 899–906.
- [112] E.F. Toro and V.A. Titarev, *MUSTA fluxes for systems of conservation laws*, Journal of Computational Physics **216** (2006), no. 2, 403.
- [113] Clifford M. Will, *The Confrontation between General Relativity and Experiment*, Living Reviews in Relativity **9** (2006), no. 3, Cited: 6th May 2009.
- [114] Y. Xing and C.W. Shu, *High order finite difference WENO schemes with the exact conservation property for the shallow water equations*, Journal of Computational Physics **208** (2005), no. 1, 206–227.
- [115] Tetsuro Yamamoto, Masaru Shibata, and Keisuke Taniguchi, *Simulating coalescing compact binaries by a new code (SACRA)*, Physical Review D **78** (2008), no. 6, 064054.
- [116] J. W. York, Jr., *Kinematics and dynamics of general relativity*, Sources of Gravitational Radiation (L. L. Smarr, ed.), 1979, pp. 83–126.
- [117] L. Del Zanna and N. Bucciantini, *An efficient shock-capturing central-type scheme for multidimensional relativistic flows I. Hydrodynamics*, Astronomy and Astrophysics **390** (2002), 1177.
- [118] Burkhard Zink, Erik Schnetter, and Manuel Tiglio, *Multipatch methods in general relativistic astrophysics: Hydrodynamical flows on fixed backgrounds*, Physical Review D **77** (2008), no. 10, 103015.