Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **498,** 3870–3887 (2020)
Advance Access publication 2020 September 2

doi:10.1093/mnras/staa2657

Hydrodynamic simulations of AGN jets: the impact of Riemann solvers and spatial reconstruction schemes on jet evolution

G. Musoke [1,2,3]★ A. J. Young [3] and M. Birkinshaw[3]

[1]*Department of Astrophysics/IMAPP, Radboud University, P.O. Box 9010, NL-6500 GL Nijmegen, The Netherlands*
[2]*Anton Pannekoek Institute for Astronomy, University of Amsterdam, Science Park 904, NL-1098 XH Amsterdam, The Netherlands*
[3]*H. H. Wills Physics Laboratory, Tyndall Avenue, Bristol BS8 1TL, UK*

Accepted 2020 August 11. Received 2020 July 21; in original form 2019 July 1

ABSTRACT

Numerical simulations play an essential role in helping us to understand the physical processes behind relativistic jets in active galactic nuclei. The large number of hydrodynamic codes available today enables a variety of different numerical algorithms to be utilized when conducting the simulations. Since many of the simulations presented in the literature use different combinations of algorithms it is important to quantify the differences in jet evolution that can arise due to the precise numerical schemes used. We conduct a series of simulations using the FLASH (magneto-)hydrodynamics code in which we vary the Riemann solver and spatial reconstruction schemes to determine their impact on the evolution and dynamics of the jets. For highly refined grids the variation in the simulation results introduced by the different combinations of spatial reconstruction scheme and Riemann solver is typically small. A high level of convergence is found for simulations using third-order spatial reconstruction with the Harten–Lax–Van-Leer with contact and Hybrid Riemann solvers.

Key words: hydrodynamics – methods: numerical – galaxies: active – galaxies: jets.

1 INTRODUCTION

Relativistic jets produced by active galactic nuclei (AGN) play an important role in the evolution of galaxies and the large-scale structure in the Universe. However, the detailed physical processes associated with the energetic interactions between jets and their environments are not fully understood. Numerical simulations are a powerful tool to investigate the interactions between jets and their surroundings, providing insights into how jets couple their energy to their surroundings, impact the evolution of galaxies and clusters, and how jet morphology and dynamics are influenced by the conditions in their local environments.

The development of early hydrodynamic and magneto-hydrodynamic (MHD) codes such as the hydrodynamics code developed by Gull (1976) and based on the Eulerian finite difference method of Gentry, Martin & Daly (1966), the 2D axisymmetric MHD code flow (Lind 1987), and the Eulerian finite difference schemes for conservative Eulerian hydrodynamics proposed by Wilson (1979) enabled early simulations of various jet processes. Numerical simulations using these codes included the production of jets from magnetised accretion discs via the Blandford–Payne mechanism (*e.g.* Lind, Meier & Payne 1994), and the kiloparsec scale evolution of jets (*e.g.* Norman et al. 1982). Early simulations conducted by Norman et al. (1982), Wilson & Scheuer (1983), Clarke, Norman & Burns (1986), Lind et al. (1989), and Martí et al. (1997) explored the nature of jet propagation and improved our understanding of the structure of radio galaxies. These simulations also provided some important constraints on the density of AGN jets – they concluded that in order to reproduce the expanding cocoon morphologies present in observations, AGN jets must be composed of material that is underdense with respect to the surrounding environment.

The simulations in Norman et al. (1982) were some of the first high-resolution hydrodynamic simulations of extragalactic jets. These 2D axisymmetric simulations recovered features at the heads of supersonic jets – the bow shock, jet shock, and contact surface, which were remarkably similar to the basic jet components in the beam model for extragalactic radio sources proposed by Blandford & Rees (1974). Improvements in computing performance and computer memory, alongside advances in multiprocessor computing and the development of computationally efficient methods for multidimensional problems made 3D simulations of extragalactic jets viable. Some of the very first 3D simulations of AGN jets were performed by Williams & Gull (1984) and Arnold & Arnett (1986) at low resolution.

Continual advancements in computational hardware, alongside developments in high-resolution shock capturing schemes and temporal and spatial discretization have lead to the increased solution accuracy of (magneto-)hydrodynamics codes, better shock handling capabilities, and the ability to run simulations at increasingly high resolutions. These advancements have lead to the generation of a variety of modern MHD codes such as FLASH (Fryxell et al. 2000), AMRVAC (Keppens et al. 2003), PLUTO (Mignone et al. 2007), ATHENA (Stone et al. 2008), and ZEUS-3D (Stone & Norman 1992a,b; Clarke, Norman & Fiedler 1994, see Martí & Müller 2015 for a detailed list of the available MHD codes). These modern codes are enabling complex jet simulations to be conducted – from fully relativistic simulations of AGN jets (O'Neill, Beckwith & Begelman 2012; Guan, Li & Li 2014;

★ E-mail: g.musoke@uva.nl

Tchekhovskoy & Bromberg 2016; English, Hardcastle & Krause 2016, 2019), to simulations exploring the disturbed morphology of (M)HD jets propagating in dynamic cluster atmospheres (e.g. Heinz et al. 2006, 2008; Mendygral, Jones & Dolag 2012; Musoke et al. 2020) alongside the interactions between jets and substructure in the surrounding ambient medium (e.g. the interactions between jets and interstellar clouds; Saxton et al. 2005; Mukherjee et al. 2016).

A major problem faced when simulating jets concerns the vastly different temporal and spatial scales associated with aspects of jet phenomena. The formation, acceleration, collimation, and large-scale propagation of jets, and the feedback and regulation processes related to AGN activity, all depend on processes occurring on vastly different scales. This leads to jet processes spanning a large dynamic range, from process occurring on spatial scales of a few Schwarzschild radii through to megaparsec scales. The formation and launching of the jet in the vicinity of the black hole and accretion disc for example concern spatial scales in the order of parsecs and smaller. The collimation and large-scale interactions of the jets as they propagate through the intracluster medium concern kiloparsec spatial scales, while the processes that govern AGN feedback can concern spatial scales in the order of megaparsecs or larger. Furthermore, specific physical processes can play important non-negligible roles for some evolutionary phases but can often be neglected in others depending on the precise nature of the system. For example, magnetic fields and general relativity pertinent to the magneto-rotational acceleration processes are clearly essential for modelling jet formation.

The vast number of hydrodynamics codes available today allow simulations to be performed using different algorithms, each of which determine how the numerical problem is evaluated. The different approaches taken in the computations, include changes in the spatial representation of the fluid states on a discretized simulation grid and how discontinuities in the fluid flow are handled, alongside differences in the underlying assumptions used in the calculations. These algorithms can provide varying degrees of spatial and temporal accuracy leading to different computational efficiencies and abilities to resolve features within the fluid flow.

Many numerical simulations of jets presented in the literature are not only conducted with a variety of different codes, but also often incorporate different combinations of numerical schemes within these codes. Different numerical algorithms applied to the same initial conditions may not produce exactly the same results.

In this work, we conduct 2D axisymmetric simulations of AGN jets, to determine the dependence of the jet evolution on the choice of numerical algorithm. We use the FLASH (Fryxell et al. 2000) code to conduct the simulations. FLASH is a Eulerian finite-volume, Godunov-based MHD code that utilizes parallel adaptive mesh refinement (AMR). The algorithms of interest in this work are the Riemann solvers used to compute the solution to the Riemann problem at the cell interfaces and the reconstruction method used to determine the spatial variation of the fluid variables across the grid cells. Of the different solvers available in FLASH, the 1D-based Harten–Lax–Van-Leer with contact (HLLC; Li 2005), Harten–Lax Van-Leer [HLL(E); Einfeldt 1988], Hybrid [use of HLLC and HLL(E) solvers] Riemann solvers, and the LLF (Lax 1954) solver are compared. For each solver, a selection of spatial reconstruction methods are tested. These reconstruction methods determine the order of spatial accuracy in reconstructing the normal components of the fluid states at the cell interfaces from the volume averaged cell states. Second- and third-order reconstruction methods are tested, which correspond to the MUSCL[1]-Hancock (MH; see VanLeervan Leer 1979) method in which the cell states are modelled as linear functions and the piece-wise parabolic method (PPM; Colella & Woodward 1984) in which the cell states are modelled as parabolic functions, respectively. We quantify how the morphology, dynamics, and energetics of the jets differ depending on the choice of Riemann solver and reconstruction scheme.

Sections 2–5 introduce key concepts concerning the different methods employed by a selection of Riemann solvers and spatial reconstruction schemes. A full description of the Riemann solvers is beyond the scope of this work and the reader is referred to Toro, Spruce & Speares (1994), Toro (2009), LeVeque (2002), and references therein for a full overview. The simulation parameters and the numerical approach taken are discussed in Section 6. In the remaining sections, we discuss and quantify the differences in the numerical solutions introduced by the various combination of algorithms by comparing the morphology, dynamics, and energetics of the simulated jets over the course of their evolution.

2 EULER EQUATIONS

In this work, we assume that the dynamics of AGN jets are dominated by the thermal plasma rather than the jet magnetic fields. We therefore take a purely hydrodynamic approach when simulating the jets.

By neglecting magnetic fields and relativistic effects alongside fluid viscosity and heat conduction, and assuming a compressible flow, the evolution of the jet flow is governed by the time-dependent non-relativistic Euler equations.

The Euler equations can be written in terms of the conserved variables; the mass density ρ, the momentum $\boldsymbol{u}$ where $\boldsymbol{u} = (\rho u, \rho v, \rho w)$, u, v, and w are the x, y, and z are components of the velocity, respectively, the velocity $\boldsymbol{v} = (u, v, w)$, and the total energy per unit volume of the fluid E. In conservative form, the Euler equations are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0, \tag{1}$$

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}\boldsymbol{v}) + \nabla P = \rho \boldsymbol{g}, \tag{2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + P)\boldsymbol{v}] = \rho \boldsymbol{v} \cdot \boldsymbol{g}, \tag{3}$$

where $\boldsymbol{g}$ is the gravitational acceleration, which can be calculated from an external gravitational potential Φ through $\boldsymbol{g} = -\nabla\Phi$.

The total energy per unit volume E is the sum of the internal energy $\rho\epsilon$ and kinetic energy and is given by

$$E = \rho \left(\epsilon + \frac{1}{2}\boldsymbol{v}^2 \right), \tag{4}$$

where ϵ is the specific internal energy.

Equations (1), (2), and (3) represent the conservation of mass, momentum, and energy, respectively. It should be noted that the above formulation of equation (3) neglects energy loss terms from radiative cooling.

We close equations (1–3) using the ideal gas equation of state:

$$P = (\gamma - 1)\rho\epsilon, \tag{5}$$

where P is the pressure and γ is the adiabatic index.

[1]Monotone upwind scheme for conservation laws.

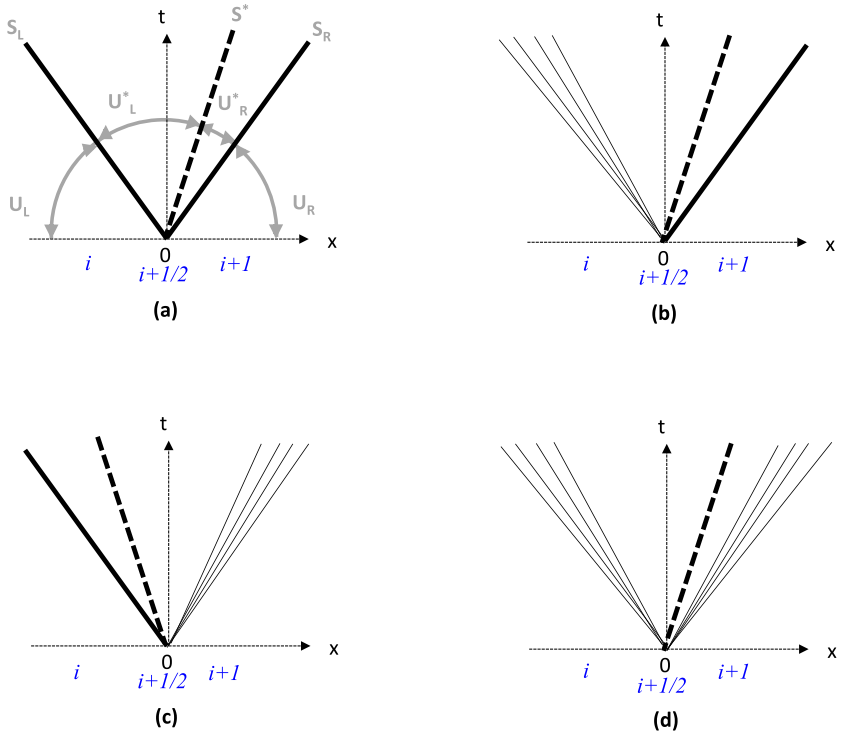

Figure 1. Four possible wave patterns in the solution to the Riemann problem for the Euler equations evaluated at the interface between cells i and $i + 1$. The solid lines represent shock waves, the dashed lines represent the contact discontinuity, and the group of four thin lines correspond to rarefaction waves. The cell index $i + 1/2$ (i.e. along the t-axis in these images) denotes the interface position between the cells. The four states $\boldsymbol{U}_{\rm L}$, $\boldsymbol{U}^*_{\rm L}$, $\boldsymbol{U}^*_{\rm R}$, and $\boldsymbol{U}_{\rm R}$ alongside the wave speeds $S_{\rm L}$, S^*, and $S_{\rm R}$ correspond to all the Riemann fans presented but are shown only in Figure (*a*) for clarity. The half-integer notation on the fluid states (and the time indices) have been dropped, (i.e. $\boldsymbol{U}_{\rm L}$ corresponds to $\mathbf{U}_{L,i+\frac{1}{2}}$, similarly for the other fluid states.).

The Euler equations can also be written in conservation law form, in three dimensions they then read

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x + \boldsymbol{G}(\boldsymbol{U})_y + \boldsymbol{H}(\boldsymbol{U})_z = \boldsymbol{S}(\boldsymbol{U}), \tag{6}$$

where each suffix represents a derivative, $\boldsymbol{U}$ is a column vector of conserved state variables, and vectors $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{H}$ represent flux vectors in the x-, y-, and z-directions, respectively. These vectors are given by

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \boldsymbol{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{bmatrix}, \boldsymbol{H} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{bmatrix}. \tag{7}$$

$\boldsymbol{S}$ is a source term and body forces such as gravity are represented in this term. The injection of mass, momentum, or energy into a system can also be included in this source term. In the absence of such terms, the right-hand side of equation (6) is simply 0.

3 RIEMANN PROBLEM

The Euler equations can be written in terms of a hyperbolic system of conservation laws. In a one dimension, in a Cartesian coordinate system and in the absence of source terms this gives

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x = 0, \tag{8}$$

where $\boldsymbol{U}(= \boldsymbol{U}(x, t))$ is a vector of the conserved fluid variables and the flux vector $\boldsymbol{F}$ is a function of the conserved variable vector $\boldsymbol{U}$ and denotes the flux along the x-direction.

The two vectors are defined as

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \boldsymbol{F} = \begin{bmatrix} \rho u \\ \rho^2 u + p \\ u(E+p) \end{bmatrix}. \tag{9}$$

The Riemann problem for the 1D Euler equations is then the initial value problem given by the following initial conditions:

$$\boldsymbol{U}(x, t = 0) = \begin{cases} \boldsymbol{U}_{\rm L} \text{ if } x < 0 \\ \boldsymbol{U}_{\rm R} \text{ if } x > 0 \end{cases}. \tag{10}$$

At the time of the initial set-up ($t = 0$), the Riemann problem consists of two states, described by the conserved variable vectors $\boldsymbol{U}_{\rm L}$ and $\boldsymbol{U}_{\rm R}$ (where the subscripts L and R refer to the left and right states, respectively), separated by a common boundary at $x = 0$. At $t = 0$ the boundary is removed and the states $\boldsymbol{U}_{\rm L}$ and $\boldsymbol{U}_{\rm R}$ are allowed to interact. The Riemann problem is a generalization of the shock tube problem in which two stationary gasses in a tube are separated by a membrane (Toro 2009). The membrane is removed and the gas states interact, generating waves that travel in both directions down the tube. Solving the Riemann problem for the Euler equations determines what the resulting waves will be.

3.1 Solutions to the Riemann problem

For purely hydrodynamic problems, the solution to the Riemann problem consists of a three-wave pattern generated once the boundary is removed and the states interact. In the x–t plane, see Fig. 1, this three-wave pattern corresponds to a *Riemann fan* bounded by two non-linear waves that are separated by a contact discontinuity

(contact wave) moving at the fluid velocity S^*. The two non-linear waves can represent either shocks or rarefaction waves. The two non-linear waves on the left and right of the contact discontinuity travel with velocities as S_L and S_R, respectively.

Four possible three-wave patterns in the solution of the Riemann problem are shown in Fig. 1. Only a selection of possible wave configurations are shown – the contact discontinuity can be on either side of the interface, or the waves can all be on one side of the interface (e.g. if all waves are supersonic). The wave pattern formed by the two non-linear waves and the contact discontinuity divides the $x-t$ plane into four regions; $\boldsymbol{U}_L$, $\boldsymbol{U}^*_L$, $\boldsymbol{U}^*_R$, and $\boldsymbol{U}_R$ [see panel (*a*) in Fig. 1], each of which define a constant state (Mignone & Bodo 2005). The region between the two non-linear waves, given by the states $\boldsymbol{U}^*_L$ and $\boldsymbol{U}^*_R$ is referred to as the star region. Within the star region, the pressure, and the velocity are constant, i.e. $p^*_L = p^*_R$ and $u^*_L = u^*_R$, while the density (and internal energy) changes discontinuously across the contact discontinuity: The density on the left of the contact discontinuity in the region $\boldsymbol{U}^*_L$ is then given by ρ^*_L, and the density on the right of the discontinuity in state $\boldsymbol{U}^*_R$ is given by ρ^*_R. The states in region $\boldsymbol{U}_L$ and $\boldsymbol{U}_R$ simply correspond to the initial left and right states of the Riemann problem as the waves have not had time to reach these regions.

Consider the initial state of the Riemann problem in the context of a discretized simulation domain – the states $\boldsymbol{U}_L$ and $\boldsymbol{U}_R$ then correspond to the fluid states immediately to the left and right of the cell interfaces and the boundary then corresponds to the interfaces between the computational grid cells. For a given cell with index i the boundary is positioned at $x_{i+\frac{1}{2}}$ and the left and right hand states at the boundary are given by $\boldsymbol{U}_{L,i+\frac{1}{2}}$ and $\boldsymbol{U}_{R,i+\frac{1}{2}}$, respectively (see Fig. 1). Each interface in the discretized grid then effectively generates a local Riemann problem, which for each interface then reads

$$\boldsymbol{U}(x, t=0) = \begin{cases} \boldsymbol{U}_{L,i+\frac{1}{2}} \text{ if } x < x_{i+\frac{1}{2}} \\ \boldsymbol{U}_{R,i+\frac{1}{2}} \text{ if } x > x_{i+\frac{1}{2}}. \end{cases} \tag{11}$$

Once the interface is removed at $t = 0$, the states are allowed to interact for some time interval Δt and the solution is advanced in time to give the new cell averaged states $\boldsymbol{U}_i^{n+1}$ using Godunov's method (Godunov 1959)

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n + \frac{\Delta t}{\Delta x}[\boldsymbol{F}_{i-\frac{1}{2}} - \boldsymbol{F}_{i+\frac{1}{2}}], \tag{12}$$

where n denotes the time $t = t^n$ and $n+1$ the time at $t = t^n + \Delta t$ and $\mathbf{U}_i^n$ is the cell averaged state at time t given by

$$\boldsymbol{U}_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \tilde{\boldsymbol{U}}(x, t_n)\mathrm{d}x\,. \tag{13}$$

Here, $\tilde{\boldsymbol{U}}$ is the reconstructed cell averaged state given by a piece-wise linear, constant, or parabolic function describing the cell state along x.

The intercell flux $\boldsymbol{F}_{i+\frac{1}{2}} = \boldsymbol{F}(\boldsymbol{U}_{i+\frac{1}{2}}(0))$ is given by the solution to the local Riemann problem at $x_{i+\frac{1}{2}}$ provided that the time-step Δt satisfies the Courant, Friedrichs, and Lewy (CFL) condition.

The CFL condition guarantees that information is not allowed to travel across more than one computational grid cell per time-step. In order to meet this constraint, the time-step Δt must be small enough the ensure that the fastest waves on the grid at a given time do not have enough time to travel a distance of more than once cell length Δx. If we denote the fastest wave speed present on grid at some time $t = t^n$ as $S^n_{\max}$, then the CFL coefficient $C_{\rm CFL}$ is defined by

$$C_{\rm CFL} = \frac{\Delta t\ S^n_{\max}}{\Delta x}, \tag{14}$$

where the time-step Δt is of a size such that

$$0 \ < \ C_{\rm CFL} \leq 1\,. \tag{15}$$

4 NUMERICAL DIFFUSION

In Eulerian-based finite-volume hydrodynamic codes, the physical fluid variables are approximated over discretized grid cells of size Δx. The change in various fluid quantities over some evolution time t is then obtained by computing the flux of the discretized variables through the cell boundaries during time intervals of size Δt. In Godunov-based numerical schemes, the approximate solutions for these discretized fluxes are obtained by solving the Riemann problem at cell interfaces, with the initial states on either side of the interface determined from cell averaged fluid variables.

As the resulting flux is then itself a discretized approximation to the physical flux exchanged at the cell boundaries during the time interval Δt, there is a truncation error associated with it. Additional or missing high order terms in the discretized approximation can appear as an effective viscosity or thermal conductivity (Robertson et al. 2010). This results in the smearing or dispersion of features in the fluid flow, i.e. the addition of numerical diffusion. Thus, the inherent discretization of Godunov-based numerical schemes mean that they are naturally associated with numerical diffusion. In order to ensure that the discretized approximation to the physical flux accurately represents the physical flux, the numerical dissipation must be kept to a minimum. Methods that add the least numerical diffusion are therefore the most accurate. The level of numerical diffusion in Godunov-based codes like FLASH varies depending on the precise numerical set-up used and is influenced by (i) the cell size Δx, (ii) the spatial reconstruction method used to determine the states on either side of the cell interfaces, (iii) the precise method used to solve the Riemann problem, (iv) the nature of the flow velocities being simulated, and (v) the choice of slope limiter (e.g. minmod, Van Leer, MC, Limited, Superbee – for a full overview of the effect of slope limiters on hydrodynamic and MHD problems, see Tóth & Odstrčil 1996).

4.1 Spatial reconstruction methods

Once the original physical, continuous fluid state is discretized, we obtain a description of the fluid in terms of volume-averaged states in each cell. However, in order to solve the Riemann problem we must represent how these states vary spatially across the entire cell in order to obtain the values of the fluid states at the cell interfaces. The method used to determine the fluid states between the cell interfaces is given by the reconstruction[2] scheme and can effect the level of numerical diffusion in the simulations.

In the standard Godunov scheme, the reconstruction method models the fluid states between the cell interfaces as constant spatial functions; the cell-averaged fluid states simply correspond to the fluid state across the entire cell. This reconstruction method is first-order accurate in space and is termed the first-order Godunov method. Due to the low order of spatial accuracy, this reconstruction scheme introduces significant amounts of diffusion to the numerical solution.

In order to improve the description of these intercell fluid states higher order reconstruction methods are used, which increase the

[2]In FLASH, the reconstruction is performed on primitive variables.

spatial accuracy of the representation and consequently introduce less numerical diffusion. These higher order methods fit linear, parabolic, and smooth high degree polynomial functions to the averaged states at the cell centres in order to reconstruct the intercell states from the average states at the cell centres and are second-, third-, and fifth-order accurate in space, respectively. The linear and parabolic functions correspond to the MH and the PPM methods, respectively. Smooth high-degree approximations can be obtained using the weighted essentially non-oscillatory (Jiang & chin Wu 1999) functions, in which the cell states are defined as the weighted average of multiple interpolating polynomials. (The multiple interpolating polynomials result from selecting different neighbouring cell states over which to interpolate).

5 RIEMANN SOLVERS

The method used to solve the Riemann problem is determined by the choice of Riemann solver, each of which has its own associated levels of dissipation. The most accurate solvers are those that are the least diffusive, however, they can also be the most susceptible to numerical pathologies. An overview of a selection of Riemann solvers is given below in order to highlight the differences between the algorithms and their associated accuracy.

In the following sections, we drop the half-integer notation $i + \frac{1}{2}$ on the fluid state vectors $\boldsymbol{U}$ used to denote the cell interface positions for simplicity (e.g. $\boldsymbol{U}_L \equiv \boldsymbol{U}_{L,i+\frac{1}{2}}$, $\boldsymbol{U}_{\rm HLL} \equiv \boldsymbol{U}_{{\rm HLL},i+\frac{1}{2}}$ $\boldsymbol{U}^*_L \equiv \boldsymbol{U}^*_{L,i+\frac{1}{2}}$ and so on). We also drop the time index n on the fluid state vectors (and the flux vectors, e.g. $\boldsymbol{F}$) by assuming the problem is evaluated at $t = t^n = 0$ for the standard Godunov scheme where the cell interface states are not centred in time prior to evaluating the Riemann problems.

5.1 HLL(E)

The HLL solver originally introduced by Harten, Lax & van Leer (1983) provides the simplest approximation to the solution of the Riemann problem. The HLL solver is based on three key assumptions as described in Toro et al. (1994):

(i) The four wave patterns present in the exact solution of the Riemann problem are represented by only a single wave pattern.
(ii) The only waves present in the solution are the two non-linear waves.
(iii) The a priori estimates of the maximum and minimum bounds of the wave speeds $S_{\rm L}$ and $S_{\rm R}$ of the non-linear waves are available.

As a result of the first approximation stated, the two non-linear waves bounding the Riemann fan are treated as shocks. The approximation listed in point (*2*) above has some important consequences; the contact discontinuity present in the exact solution of the Riemann problem is no longer present in the HLL approximation to the solution and only the two of the three waves present in the exact solution are recovered. As a result, the states describing the star region $\boldsymbol{U}^*_{\rm L}$ and $\boldsymbol{U}^*_{\rm R}$ are averaged into a single constant state $\boldsymbol{U}_{\rm HLL}$. The wave configuration then corresponds to two waves separated by three constant states, $\boldsymbol{U}_{\rm L}$, $\boldsymbol{U}_{\rm HLL}$, and $\boldsymbol{U}_{\rm R}$, and the approximate solution to the Riemann problem on the $x/t = 0$ axis (corresponding to the interface position $i + 1/2$) is therefore given by

$$\boldsymbol{U}(0,t) = \begin{cases} \boldsymbol{U}_{\rm L} \text{ if } S_{\rm L} \geqslant 0, \\ \boldsymbol{U}_{\rm HLL} \text{ if } S_{\rm L} \leqslant 0 \leqslant S_{\rm R}, \\ \boldsymbol{U}_{\rm R} \text{ if } S_{\rm R} \leqslant 0, \end{cases} \tag{16}$$

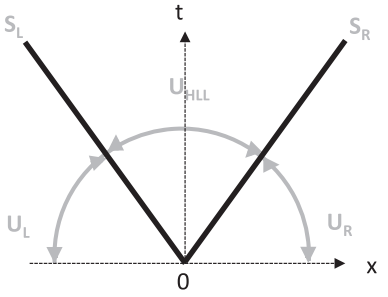

Figure 2. The Riemann fan for HLL solver. The solver recovers only two of the waves present in the solution to the Riemann problem and represents these as shocks propagating with speeds $S_{\rm L}$ and $S_{\rm R}$. The contact discontinuity is not recovered in this approximation and thus the HLL solver is a two-wave, three-state solver.

where the constant state $\boldsymbol{U}_{\rm HLL}$ can be constructed from estimates of the wave speeds $S_{\rm L}$ and $S_{\rm R}$ to give

$$\mathbf{U}_{HLL} = \frac{S_R\mathbf{U}_R - S_L\mathbf{U}_L + \mathbf{F}_L - \mathbf{F}_R}{S_R - S_L}. \tag{17}$$

It should be noted that equation (17) corresponds to the integral average of the solution of the Riemann problem over the Riemann fan (Toro 2009). The flux functions (in the x-direction) $\boldsymbol{F}_{\rm L}$ and $\boldsymbol{F}_{\rm R}$ are given by $\boldsymbol{F}_{\rm L} = \boldsymbol{F}(\boldsymbol{U}_{\rm L})$ and $\boldsymbol{F}_{\rm R} = \boldsymbol{F}(\boldsymbol{U}_{\rm R})$. The corresponding intercell flux $\boldsymbol{F}_{i+\frac{1}{2}}$ for the HLL scheme is then given by

$$\boldsymbol{F}_{i+\frac{1}{2}} = \begin{cases} \boldsymbol{F}_{\rm L} \text{ if } S_{\rm L} \geqslant 0, \\ \boldsymbol{F}_{\rm HLL} \text{ if } S_{\rm L} \leqslant 0 \leqslant S_{\rm R}, \\ \boldsymbol{F}_{\rm R} \text{ if } S_{\rm R} \leqslant 0, \end{cases} \tag{18}$$

where

$$\boldsymbol{F}_{\rm HLL} = \frac{S_{\rm R}\boldsymbol{F}_{\rm L} - S_{\rm L}\boldsymbol{F}_{\rm R} + S_{\rm R}S_{\rm L}(\boldsymbol{U}_{\rm R} - \boldsymbol{U}_{\rm L})}{S_{\rm R} - S_{\rm L}}. \tag{19}$$

The wave speed estimates are obtained using a variety of different algorithms depending on the exact formulation of the HLL solver used, for an overview of such algorithms see section 10.5 of Toro (2009). The HLL solver in FLASH is the HLL(E) solver of Einfeldt (Einfeldt 1988) that differs from the HLL solver of Harten, Lax and van Leer (Harten et al. 1983) with regards to the computation of the wave speeds $S_{\rm L}$ and $S_{\rm R}$.

The HLL scheme recovers only two of the three waves present in the pure hydrodynamic (Euler) solution to the Riemann problem; the outermost and fastest forward and backward moving waves. Thus, the Riemann fans (Fig. 1) are replaced with simpler approximation as shown in Fig. 2 by resolving/recovering only a selection of the moving discontinuities (waves) and replacing the intermediate states that are separated by the unresolved intermediate waves with an average single state $\boldsymbol{U}_{\rm HLL}$. The HLL scheme does not retain the central contact (entropy) wave as it is effectively lost in the averaging of the intermediate system state, hence the HLL solver is an *incomplete* Riemann solver as it does not recover all of the three waves present in the solution to the Euler equations.

The lack of resolution of the central contact wave means that the HLL solver does not recover all of the intermediate substructure in the Riemann fan. This introduces further numerical diffusion into the system on top of that already introduced through the averaging process inherent in the discretization. Consequently, the resolution of shocks, contact discontinuities, shear waves, and material interfaces in the HLL scheme can be very inaccurate (Toro 2009), as small-scale flow features are more effectively smeared out as the numerical diffusion increases. In order to keep the simulated discretized system

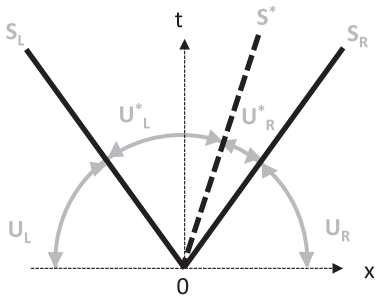

Figure 3. Riemann fan for HLLC solver. The HLLC solver restores the central contact discontinuity absent in the HLL solver, making it a three-wave, four-state solver.

behaving as closely to the real physical continuous system on which it is based, numerical diffusion is therefore required to be kept to a minimum. The accuracy of a Riemann solver is therefore tied to the level of approximation used in capturing the correct substructure in the Riemann fan (Mignone 2007). The most diffusive solvers are the least accurate, though are typically the most stable. The choice of suitable Riemann solver is therefore often determined by the solvers ability to best balance numerical accuracy and stability. Consequently, the specific choice of Riemann solver may differ depending on the nature of the problem to which it is being applied.

The problems associated with the lack of resolution of the contact discontinuity can be addressed by methods that restore this missing central contact wave, restoring some of the substructure missing in the approximation to the exact Riemann fan. This leads to schemes such as the HLLC scheme proposed by Toro et al. (1994) in which the central contact wave is restored.

5.2 HLLC

The HLLC solver improves the standard HLL formalism by restoring the central contact wave, and is therefore a three-wave, four-state solver as shown in Fig. 3. The single averaged state $\boldsymbol{U}_{\rm HLL}$ is now replaced with two approximate states $\boldsymbol{U}^*_{\rm L}$ and $\boldsymbol{U}^*_{\rm R}$ separated by the central contact wave which is assumed to have constant speed S^*. The full solution to the Riemann problem is then given by

$$\boldsymbol{U}(0,t) = \begin{cases} \boldsymbol{U}_{\rm L} \text{ if } S_{\rm L} \geqslant 0, \\ \boldsymbol{U}^*_{\rm L} \text{ if } S_{\rm L} \leqslant 0 \leqslant S^*, \\ \boldsymbol{U}^*_{\rm R} \text{ if } S^* \leqslant 0 \leqslant S_{\rm R}, \\ \boldsymbol{U}_{\rm R} \text{ if } S_{\rm R} \leqslant 0. \end{cases} \tag{20}$$

The intercell numerical fluxes are then given by

$$\boldsymbol{F}_{i+\frac{1}{2}} = \begin{cases} \boldsymbol{F}_{\rm L} \text{ if } S_{\rm L} \geqslant 0, \\ \boldsymbol{F}^*_{\rm L} \text{ if } S_{\rm L} \leqslant 0 \leqslant S^*, \\ \boldsymbol{F}^*_{\rm R} \text{ if } S^* \leqslant 0 \leqslant S_{\rm R}, \\ \boldsymbol{F}_{\rm R} \text{ if } S_{\rm R} \leqslant 0\,. \end{cases} \tag{21}$$

The fluxes in the star region $\boldsymbol{F}^*_{\rm L}$ and $\boldsymbol{F}^*_{\rm R}$ can be expressed in terms of the constant states $\boldsymbol{U}^*_{\rm L}$ and $\boldsymbol{U}^*_{\rm R}$ from the Rankine–Hugoniot jump conditions (Mignone & Bodo 2005):

$$\begin{aligned} S_{\rm L}(\boldsymbol{U}^*_{\rm L} - \boldsymbol{U}_{\rm L}) &= \boldsymbol{F}^*_{\rm L} - \boldsymbol{F}_{\rm L} \\ S_{\rm R}(\boldsymbol{U}^*_{\rm R} - \boldsymbol{U}_{\rm R}) &= \boldsymbol{F}^*_{\rm R} - \boldsymbol{F}_{\rm R}\,. \end{aligned} \tag{22}$$

As the central contact wave is restored in this scheme, the HLLC solver is a *complete* Riemann solver regarding the Euler equations. This is because the approximate wave structure in the resulting Riemann fan now contains all of the waves present in the solution to the Euler equations. The inclusion of the missing central contact wave comes at the cost of requiring additional jump conditions for the restored wave, thus increasing the number of equations and unknowns in the scheme (Mignone, Bodo & Ugliano 2012). Consequently, the HLLC solver is less computationally efficient than the standard HLL solver. Despite this there are definite advantages to using the HLLC solver at the expense of computational efficiency. The HLLC scheme goes a long way in resolving the important flow features as a result of recovering the central contact wave; the solver correctly captures contact discontinuities and shear waves, providing sharper representation of these flow features alongside small-scale structure which often lacks detail with the HLL solver (Toro 2009). The HLLC Riemann solver is shown to work robustly for both hydrodynamic and MHD flows and is found to be superior to the HLL solver producing less diffusive and more accurate results to those obtained with the HLL solver (e.g. Toro et al. 1994, Mignone & Bodo 2005, Kong 2011 and section 10.8 of Toro 2009).

5.3 Roe

The Roe solver is an approximate Riemann solver proposed by Roe (1981). Like the HLLC solver, the Roe solver recovers all of the waves present in the exact solution to the Riemann problem. The key difference between the Roe and HLLC schemes concerns the description of the wave propagation speeds and average interface state, primarily concerning how these are obtained (see Toro 2009). In the Roe formalism, the average interface states (and thus corresponding numerical fluxes) are obtained rigorously by taking account of the full characteristic structure of the equations in the computation and the linearization of the hyperbolic equations is performed in such a way that the correct wave speeds are preserved. This is in contrast to the HLL-type solvers that rely on wave speed estimates from a given algorithm. The Roe solver is the most accurate and least diffusive of the approximate solvers mentioned and provides the best resolution of shocks, contact discontinuities, and small-scale flow features. The increased accuracy of the Roe solver, however, comes at the cost of computational efficiency and stability.

The low numerical diffusion of the Roe solver means it is more unstable to numerical pathologies, and thus produces numerical artefacts far more easily than the other more diffusive solvers. The Roe solver can generate non-physical solutions such as expansion shocks in the place of what should be smooth rarefaction waves (Quirk 1994) and has been known to generate negative densities and pressures in regions containing large gradients such as those in the proximity of strong shocks. Numerical artefacts associated with the Roe scheme include the carbuncle phenomenon, a type of shock wave instability that leads to the breakdown of a discrete shock profile. The phenomenon occurs in the presence of strong shocks and causes the smooth shock profile to develop a pair of oblique shocks at the shock mid-point, leading to a pyramid or point-like feature upstream of the shock (Kemm 2018). In hydrodynamic simulations of supersonic jets, the carbuncle phenomenon unusually appears as an extended nose in front of the jet along its axis (Quirk 1994).

5.4 Hybrid

The Hybrid solver for hydrodynamics problems appropriately combines the HLLC and HLL schemes depending on the strength of the shocks (Lee 2013). The HLL solver is only used in the presence of strong shock fronts that are detected by a pair of pressure-based and velocity-based shock switches (see Lee 2013; Shen, Yan & Yuan 2016 and Balsara & Spicer 1999), while the HLLC solver

is used for all the remaining regions (Lee 2013). This has the effect of using the diffusive methods associated with the HLL scheme in regions close to, and along the lengths of strong shocks while using the less diffusive methods of the HLLC scheme in regions away from the shock front. This consequently eliminates numerical artefacts associated with low numerical diffusion in the presence of strong shocks (such as the carbuncle phenomenon) and avoids the computation of non-physical system states in regions containing large gradients.

5.5 LLF

Unlike the Roe and HLL-type solvers that solve the Riemann problem, the local Lax–Friedrichs (LLF) scheme solves systems of hyperbolic partial differential equations based on finite central differences. In this scheme, the flux at the cell interfaces is defined as the average of the fluxes calculated at the centre of cells *adjacent* to the cell of interest (Massaglia, Zurlo & Bodo 2003).

The LLF scheme is obtained from the original Lax–Friedrichs scheme proposed in Lax (1954), in which the conservative temporal update in equation (12) of the original Lax–Friedrichs scheme is given by

$$\boldsymbol{U}_i^{n+1} = \frac{1}{2}\left(\boldsymbol{U}_{i-1}^{n} + \boldsymbol{U}_{i+1}^{n}\right) + \frac{1}{2}\frac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i-1}^{n} - \boldsymbol{F}_{i+1}^{n}\right), \tag{23}$$

which can be written as

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^{n} + \frac{\Delta t}{\Delta x}\left[\boldsymbol{F}_{i-\frac{1}{2}} - \boldsymbol{F}_{i+\frac{1}{2}}\right], \tag{24}$$

and the corresponding intercell numerical flux is

$$\boldsymbol{F}_{i+\frac{1}{2}} = \frac{1}{2}\left(\boldsymbol{F}_{i}^{n} + \boldsymbol{F}_{i+1}^{n}\right) + \frac{1}{2}\frac{\Delta x}{\Delta t}\left(\boldsymbol{U}_{i}^{n} - \boldsymbol{U}_{i+1}^{n}\right), \tag{25}$$

where $\Delta x = x_{i+1/2} - x_{i-1/2}$ is the length of the cell and the time-step $\Delta t = t_{n+1} - t_n$.

The standard Lax–Friedrichs scheme is extremely diffusive and produces numerical solutions that are badly smeared unless very highly refined grids are used (LeVeque 2002). The LLF scheme introduced by Rusanov (1961) on the other hand introduces less numerical diffusion and the scheme is obtained by replacing $\Delta x/\Delta t$ with a locally (chosen at each Riemann problem) determined value of the maximum modulus of the characteristic wave speeds at adjacent cells (Massaglia et al. 2003).

Of all the solvers mentioned in the previous sections, the LLF scheme is the most diffusive and is associated with the lowest resolution of small-scale flow features. As a result of the diffusive nature of the scheme, it is the least accurate of the solvers outlined.

6 SIMULATIONS

6.1 Parameters

All the simulations presented are conducted using the FLASH (M)HD code (Fryxell et al. 2000) in its pure hydrodynamics mode. Table 1 shows the combination of Riemann solver and reconstruction scheme used in each run. The simulation set-up and parameters used as the basis of the numerical study correspond to the 2D axisymmetric simulation 'Run 2' conducted in Reynolds, Heinz & Begelman (2002), in which the active and passive evolution of a bipolar conical jet is simulated. The simulations are performed in cylindrical (r, z) coordinates. We perform 2D axisymmetric simulations in order to ensure that the jet and cocoon are well resolved during both the active and passive evolution, over the full 500 Myr evolution. A full 3D treatment spanning the dynamic range of the simulations presented here would be extremely computationally intensive for the grid resolution that we utilize in this work.

Table 1. Table outlining the different combinations of Riemann solver and reconstruction method used for each simulation. The first column gives the simulation name that follows the following format; `solver_reconstruction scheme`. The second column displays the Riemann solver used in the simulation. The third column gives the spatial reconstruction method used, where MH and PPM refer to the MUSCL-Hancock and piece-wise parabolic method, respectively (see Section 4.1). The order of spatial accuracy of the corresponding reconstruction method is given in the final column. Further simulation parameters, including those used for the jet and ambient medium, are given in Section 6.

Name	Riemann solver	Reconstruction method	Order
HLLC_02	HLLC	MH	Second
HLLC_03	HLLC	PPM	Third
HLL_02	HLL	MH	Second
HLL_03	HLL	PPM	Third
LLF_02	LLF	MH	Second
LLF_03	LLF	PPM	Third
HY_02	Hybrid	MH	Second
HY_03	Hybrid	PPM	Third

As in Reynolds et al. (2002), a bipolar jet with a total kinetic power of 5.08×10^{45} erg s^{-1} propagates through a stratified external medium. The density profile of the external medium is given by the isothermal beta model:

$$\rho = \rho_0 \left(1 + \frac{R^2}{r_c^2}\right)^{-3\beta/2}, \tag{26}$$

where ρ_0 is the core density, $r_c = 100$ kpc is the core radius, and $R = \sqrt{r^2 + z^2}$ is the distance from the origin. We use $\beta = 0.5$ as in Reynolds et al. (2002).

The gravitational acceleration $\boldsymbol{g} = -\nabla\Phi$ required to maintain hydrostatic equilibrium in the external medium is then given by

$$g = -\frac{3\beta kT}{\mu m_H r_c}\frac{R/r_c}{\left[1 + \left(R^2/r_c^2\right)\right]}. \tag{27}$$

Jet material is continually injected into the simulation domain for 50 Myr, after which injection of the jet material is stopped and the system allowed to evolve passively until $t = 500$ Myr.

The jet injection velocity is $0.3c$, which corresponds to an internal (relative to the sound speed in the jet material) Mach number of 10, and the jets are initialized with a 15 deg half opening angle on the flow. The number density at the centre of the external medium $n_0 = 0.01\text{cm}^{-3}$ and the density contrast between the external medium and the jet $\eta = \rho_j/\rho_0 = 0.01$. The jet fluid is in pressure balance with the external medium and the adiabatic index $\gamma = 5/3$. The simulations are performed in scaled simulation units. The key units are briefly noted as follows: one unit of time corresponds to 50 Myr, the unit of distance is 50 kpc, and the unit of energy is 3.68×10^{44} J. The unit of speed corresponds to the sound speed in the ambient medium $c_s = 1 \times 10^8$ cm s^{-1}. Given these units, the active phase of the evolution ends at $t = 1$ simulation unit and the simulations span a total evolution time of $t = 10$ simulation units. The density in the external medium $\rho_0 = n_0 \mu m_p = 1$, where m_p is the proton mass and $\mu = 0.6$, the jet density $\rho_{jet} = 0.01$, the pressure of the jet and external medium is given by $1/\gamma = 3/5$, the sound speed in the external medium $c_s = 1$ and the jet velocity $v_j = 100$. The simulation domain spans the region $r \in (0, 53.46)$, $z \in (-26.73, -26.73)$ simulation units. The simulations

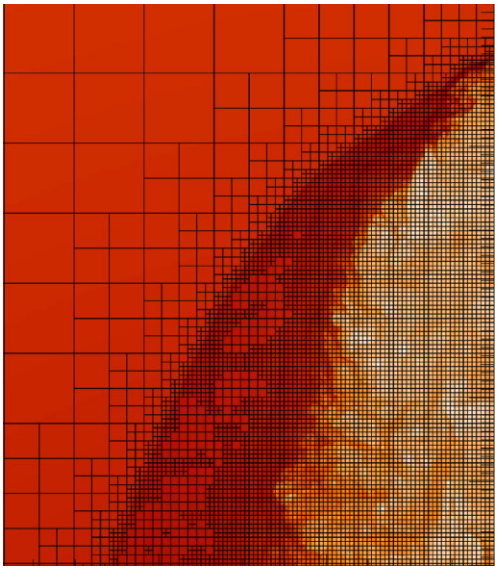

Figure 4. Figure showing different levels of grid refinement around a zoomed in region of the cocoon in run HLLC_03. Only the blocks on the AMR grid, and not the internal block cells, are shown in this image for clarity. The AMR grid is overlaid on to a log density map, where red indicates high-density regions and yellow/white corresponds to the low-density cocoon.

use AMR with a total of 11 levels of refinement. The highest refined grid cells have dimensions of 3.26×10^{-3} simulation units per side and the jet radius spans 8 of these cells. Since we only have 8 cells in the jet radius, we cannot adequately impose radial stratification in the jet inlet that will lead to a spine-sheath configuration along the jet axis downstream of the inlet. Radial stratification of the jet has been shown to impact the stability of the jet, influence the degree of entrainment of shocked ambient medium, and also impact the large-scale jet morphology (see Walg et al. 2013 for a full numerical study of jet radial stratification and references therein).

The AMR grid used in our simulations refines the grid on two criteria: (1) the second derivative of the pressure and density and (2) the passive jet tracer used to track the injected jet fluid (see Section 6.3). The highest refined grid cells are used in the locations where (1) and/or (2) are high. This ensures that high-resolution grid cells are placed within the jet, cocoon, and at the contact discontinuity and bow shock as shown in Fig. 4.

We note that the whilst high-order spatial reconstruction methods (e.g. second and third order) are available in FLASH, the order of accuracy of these methods apply only to smooth flows in which discontinuities are not present. In the vicinity of strong shocks and discontinuities, the spatial reconstruction schemes are at best first-order accurate (e.g. see ASC FLASH 2012). Nevertheless reconstructing the spatial states with higher order reconstruction schemes reduces some of the inherent numerical diffusion in the simulation and thus differences are still seen across the simulations even though their global order of accuracy is the same.

6.2 Algorithms

In this work, we compare the effect of different combinations of Riemann solver and reconstruction method on the simulation results (see Table 1). We conduct a series of simulations with the HLLC, HLL(E), and Hybrid [combination of HLLC and HLL(E)] Riemann solvers and the LLF solver. For each Riemann solver, we test two different spatial reconstruction methods – MH (second-order spatial accuracy) and the PPM (third-order spatial accuracy) methods (see Section 4.1). We note that simulations using the Roe solver did not remain stable for the problem being simulated in this work and thus it is not compared here.

In all the simulations, we use the Van Leer slope limiter, a $CFL = 0.3$ and the directionally unsplit hydrodynamics solver (see Lee 2013). The simulations are computed on the University of Bristol's BlueCrystal super computer on four nodes, each comprising 16 2.6 GHz SandyBridge cores with 4GB RAM per core.

Table 2. Computational efficiencies for algorithm comparison. Wall time for each simulation is given in seconds. The average time-step size $<dt>$ is given in simulation units where one unit of time is 50 Myr.

Run	Wall time (s)	$<dt>$
HLLC_02	2739.87	2.66 E-6
HLLC_03	3272.05	2.62 E-6
HY_02	2957.58	2.61 E-6
HY_03	3637.93	2.63 E-6
HLL_02	2478.61	2.64 E-6
HLL_03	3177.11	2.63 E-6
LLF_02	2899.34	2.63 E-6
LLF_03	3229.04	2.64 E-6

6.3 Cocoon identification

In these simulations, we assign a passive tracer to the injected jet fluid. This allows us to see how the jet fluid mixes with the surrounding ambient medium and track the cocoon region during the simulation. The volume fraction associated with the jet fluid tracer is given by f_j. For all the simulations, we define the cocoon region as the volume inside which $f_j > 10^{-3}$, where a non-zero value is chosen to take account of any mixing between the jet and shocked ambient medium.

7 COMPUTATIONAL EFFICIENCY

Table 2 displays the wall time in seconds required to complete 9000 time-steps, alongside the time-step size dt in simulation units averaged over the same time-steps. The time-steps are sampled from step number 1000–10 000 in order to omit the wind-up period from the initial small time-step $dt = 1.0\text{E}{-}10$ simulation units, while also avoiding any writing to plot files that could interfere with the wall time measurements. Additionally, during these time-steps the AMR grid does not refine, as the jet is still within the radius at which we enforce maximum refinement around the jet inflow region. The wall time presented should therefore be an accurate representation of the different computational requirements of the various algorithms. Table 2 shows that for each solver the wall time reduces with decreasing spatial reconstruction order. The second-order spatial reconstruction scheme is typically $\sim$ 10–20 per cent more efficient than third-order spatial reconstruction.

The average time-step size $<dt>$ becomes smaller as the order of accuracy in the spatial reconstruction is increased from second to third order. Thus, for the simulations using the third-order spatial reconstruction scheme more Riemann problems are solved over the course of the evolution. The computational efficiency is more dependent on the reconstruction method rather than the choice of Riemann solver. Of the combinations tested run HY_03 (Hybrid solver with third-order reconstruction) is the most inefficient. Moreover, for a

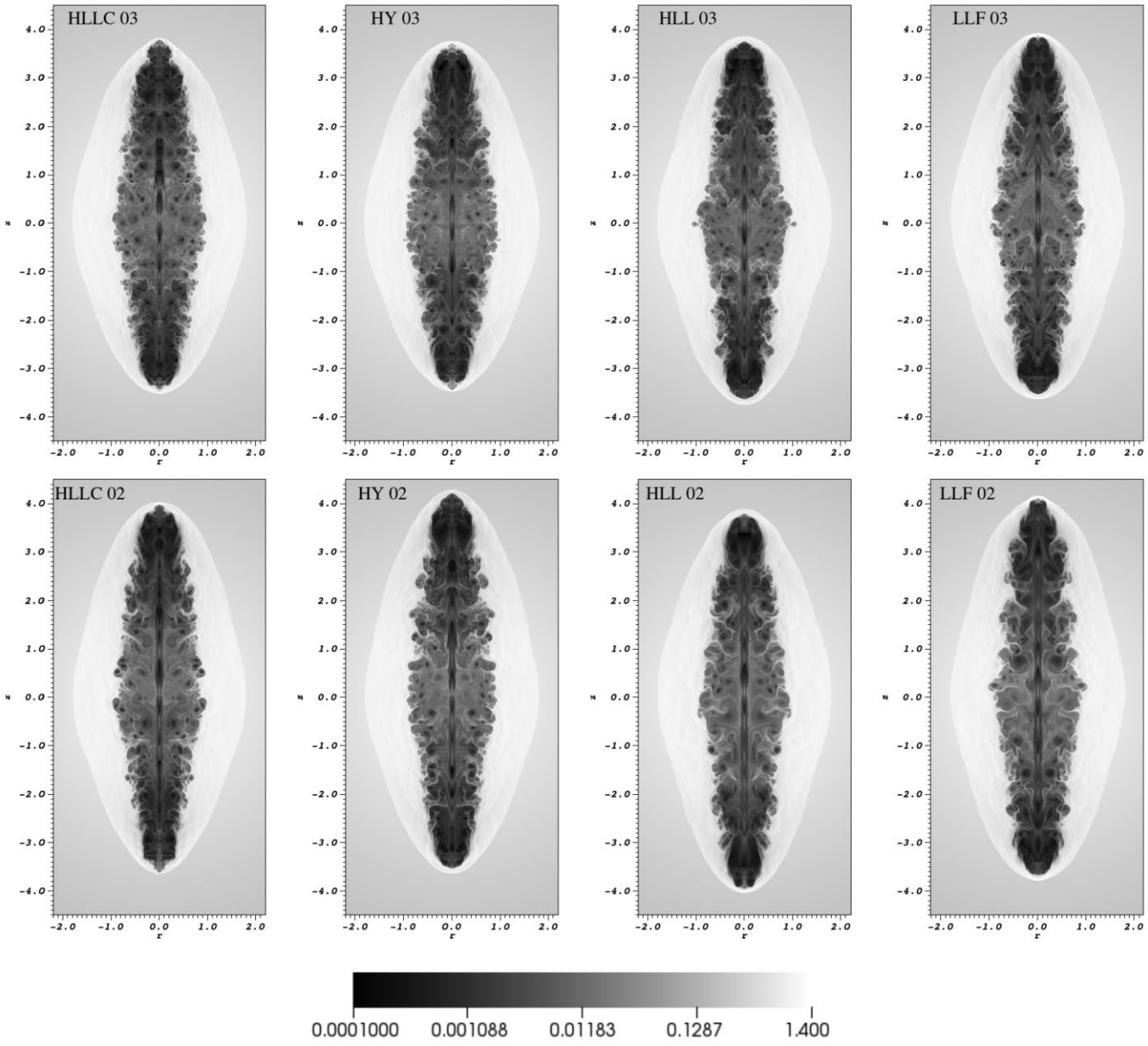

Figure 5. Log density map showing the evolution at $t = 1$ simulation unit (end of the active evolution) for the different solvers and reconstruction methods. The axes and legend are in simulation units (see Section 6). Low-density material corresponds to black and high-density material to white.

given reconstruction method the Hybrid solver is the most inefficient, requiring the longest wall times.

8 EVOLUTION OVERVIEW

8.1 Active evolution

As the jets begin to propagate into the isothermal β atmosphere, they drive a pair of upstream and downstream moving shock fronts, the bow shocks and jet shocks, respectively. The jet shock (Mach disc) marks the abrupt termination point of the jet. The jet shock acts to slow the advance of the jet by broadening the jet head, and spreading the momentum flux of the jet across a larger area. The advance speed of the jet, governed by the jet thrust, then becomes a fraction of the injection velocity. Jet material that crosses the jet termination shock is thermalized and is deflected sideways, flowing back alongside the bulk flow of the jet and inflating a cocoon of shocked jet material. Pressure feedback from the cocoon generates a series of regularly spaced oblique shocks along the bulk flow of the jet.

The bow shock compresses and heats the regions of the external medium that pass through it, creating a shell of shocked ambient medium around the cocoons. The strength of the bow shock is mostly determined by the total power of the jet (e.g. Zanni et al. 2005). The shocked ambient medium and the undisturbed ambient medium are separated by the bow shock, while the turbulent cocoon is separated from the shocked ambient medium via a contact discontinuity. This latter structure is subject to Kelvin–Helmholtz instabilities that act to destroy the contact discontinuity and mix the shocked ambient medium of the shell into the cocoon of shocked jet material. The instabilities are excited by the turbulent nature of the cocoons and increase in strength as the jet backflow increasingly decelerates away from the head of the jet. During this phase of the evolution the jet,

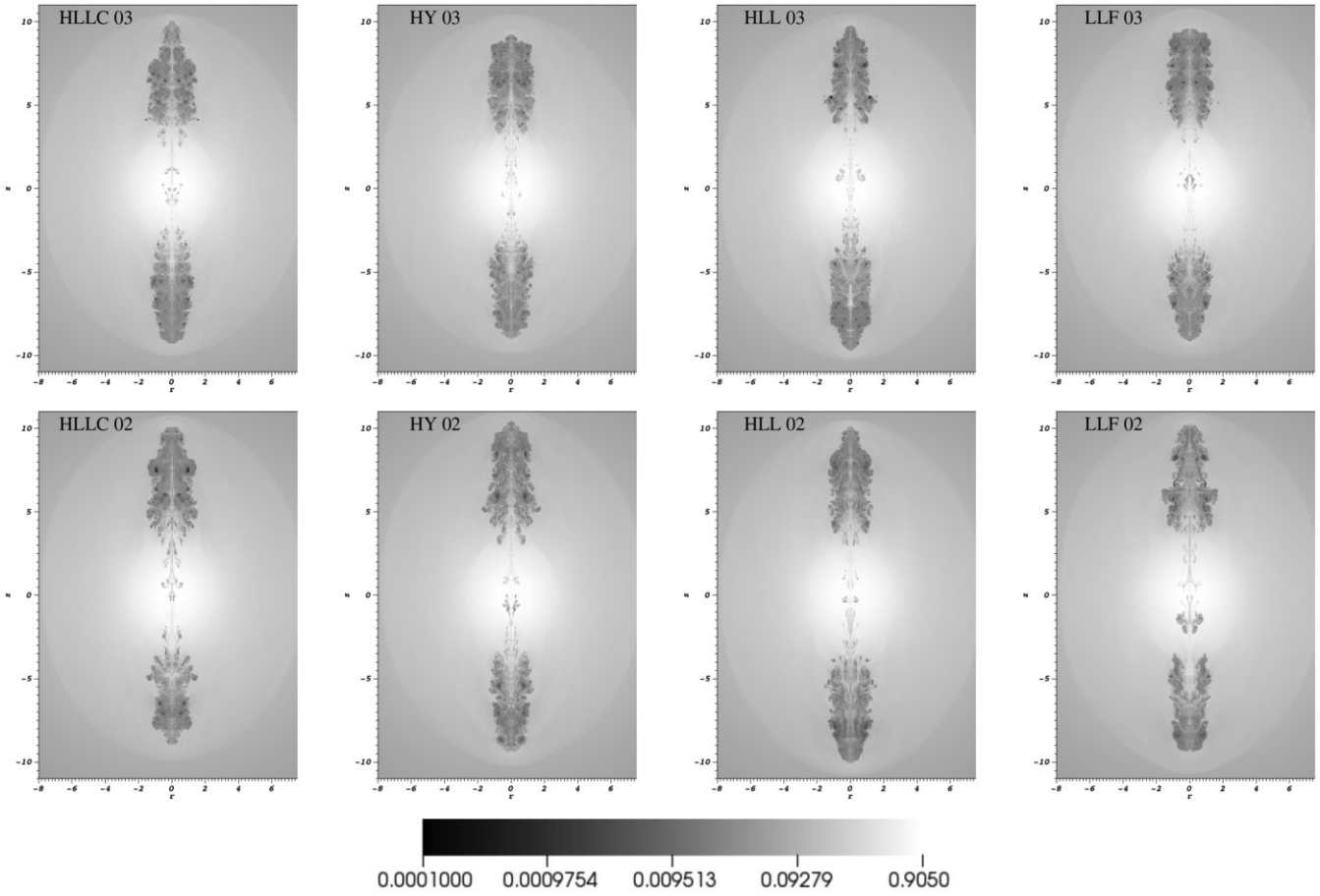

Figure 6. Log density map showing the evolution during the passive phase at $t = 6$ simulation units for the different solvers and reconstruction methods. Low-density material corresponds to black and high-density material to white. Axes and legend are in simulation units (see Section 6 for more details).

cocoon and shocked shell are in pressure equilibrium with each other, but are overpressured with respect to the ambient medium (e.g. Reynolds et al. 2002).

8.2 Passive evolution

At 50 Myr into the jet evolution ($t = 1$ simulation unit), the jets are switched off. The jet-ambient system is then allowed to evolve passively for a further 450 Myr as in Reynolds et al. (2002) in order to simulate the relics of dead radio galaxies. More recently, English et al. (2019) have used relativistic MHD simulations to model the passive evolution of jets with various kinetic powers, in a variety of different cluster environments described by the isothermal beta model (see Section 6). As described in Reynolds et al. (2002), the onset of the passive evolution immediately results in the disappearance of the jet termination shock. The channels excavated by the bulk jet flow collapse and disappear due to the absence of the oblique shocks within the bulk jet flow.

The advance speeds of the cocoon slows due to the absence of ram pressure from the jets and the velocities of the backflow reduce as the cocoons are no longer replenished by the shocked supersonic backflow created by the jet material crossing the terminal shock. The contact discontinuity between the cocoon material and shocked shell of external medium becomes unstable to Rayleigh–Taylor instabilities once the deceleration of the contact discontinuity does not exceed the local gravitational acceleration of the atmosphere (Zanni et al. 2005). Once the cocoons establish a rough pressure balance with the ambient medium this condition for stabilising the contact discontinuity against the Rayleigh–Taylor instabilities is no longer maintained and the contact discontinuity becomes Rayleigh–Taylor unstable.

The bow shock begins to weaken following the onset of the passive evolution, as it is no longer supported by the jet thrust. The profile of the bow shock becomes increasingly spherical in shape as it is no longer actively driven by the jet. The shocked shell decelerates and the bow shock eventually weakens, and begins to propagate at subsonic velocities through the ambient medium, transitioning from a shock to a sound/compression wave. This subsequently reduces the compression of the ambient medium that passes though the shock.

Following the cocoon establishing pressure balance with the ambient medium, the cocoon rises in the ambient medium as it is buoyantly unstable due to its inertia and high entropy. The surrounding shocked ambient medium begins to recover a new hydrostatic equilibrium and the dense ambient medium begins to settle back into the centre, reforming the dense core region of the atmosphere. The returning dense ambient medium squeezes the regions of the cocoon closest to the centre of the atmosphere, eventually pinching the cocoon into two separate plumes of low density and higher entropy compared to the surrounding ambient medium. Ambient medium previously entrained by the cocoons continues to be carried along by the plumes. The plumes continue to entrain ambient medium as they rise due to the cocoon vortices. These vortices drag surrounding ambient medium into the plume bases, while strong eddies produce wispy structures at the plume bases.

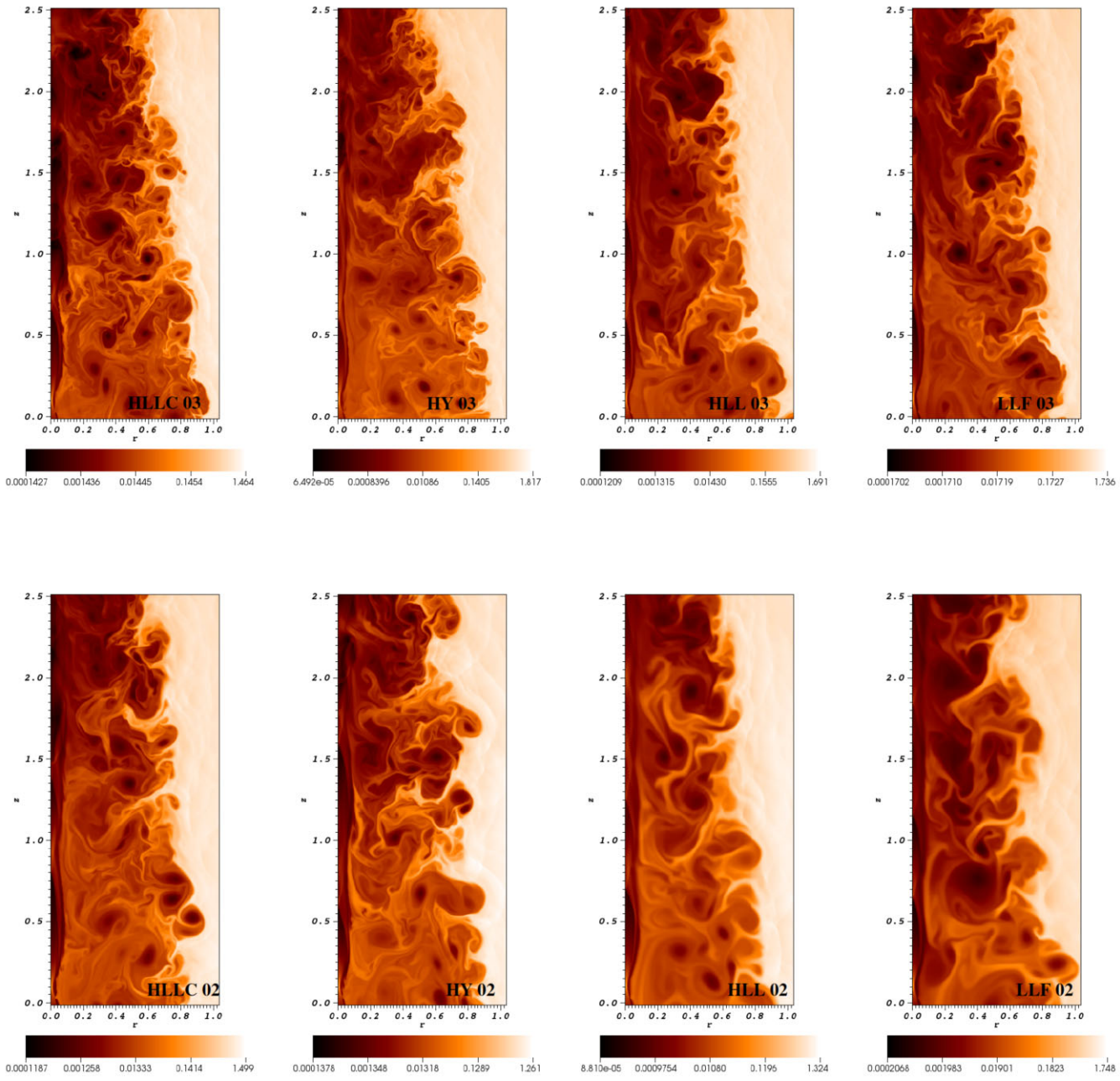

Figure 7. Log density maps showing detailed cocoon structure at $t = 1$ simulation unit (50 Myr) for different Riemann solvers and reconstruction methods. The axes and legend are in simulation units. The colour scale is in simulation units, the dark colours correspond to low density, and the light colours correspond to high density.

9 MORPHOLOGY

The log density maps in Figs 5 and 6 show the effect of changing the Riemann solver and reconstruction method on the large-scale morphology of the jets at $t = 1$ simulation unit (50 Myr, the end of the active evolution) and $t = 6$ simulation units (passive evolution). The columns in Figs 5 and 6 correspond to the different Riemann solvers tested and the solvers are roughly ordered in terms of the diffusivity of the solver, with the least diffusive solver on the left (HLLC) and the most diffusive solver on the right (LLF).

For simulations using third-order spatial reconstruction (top row), the differences in the cocoon and jet morphology are negligible for the Riemann solvers considered. For the simulations conducted with second-order spatial reconstruction, the large-scale cocoon structure is similar for the Riemann solvers tested, however, the small-scale structure within the cocoon becomes less well resolved as the diffusivity of the Riemann solver increases. The differences in small-scale structure within the cocoons are more clearly seen in Figs 7 and 8 that show a zoomed in views of the cocoon and a Schlieren plot of the density (e.g. Keppens et al. 2012 and Hughes, Miller & Duncan 2002), respectively, at $t = 1$ simulation unit (50 Myr). The small-scale turbulent features of the cocoon and the Kelvin–Helmholtz instabilities at the contact discontinuity are less sharply defined in the most diffusive simulation, LLF_02, compared to the least diffusive

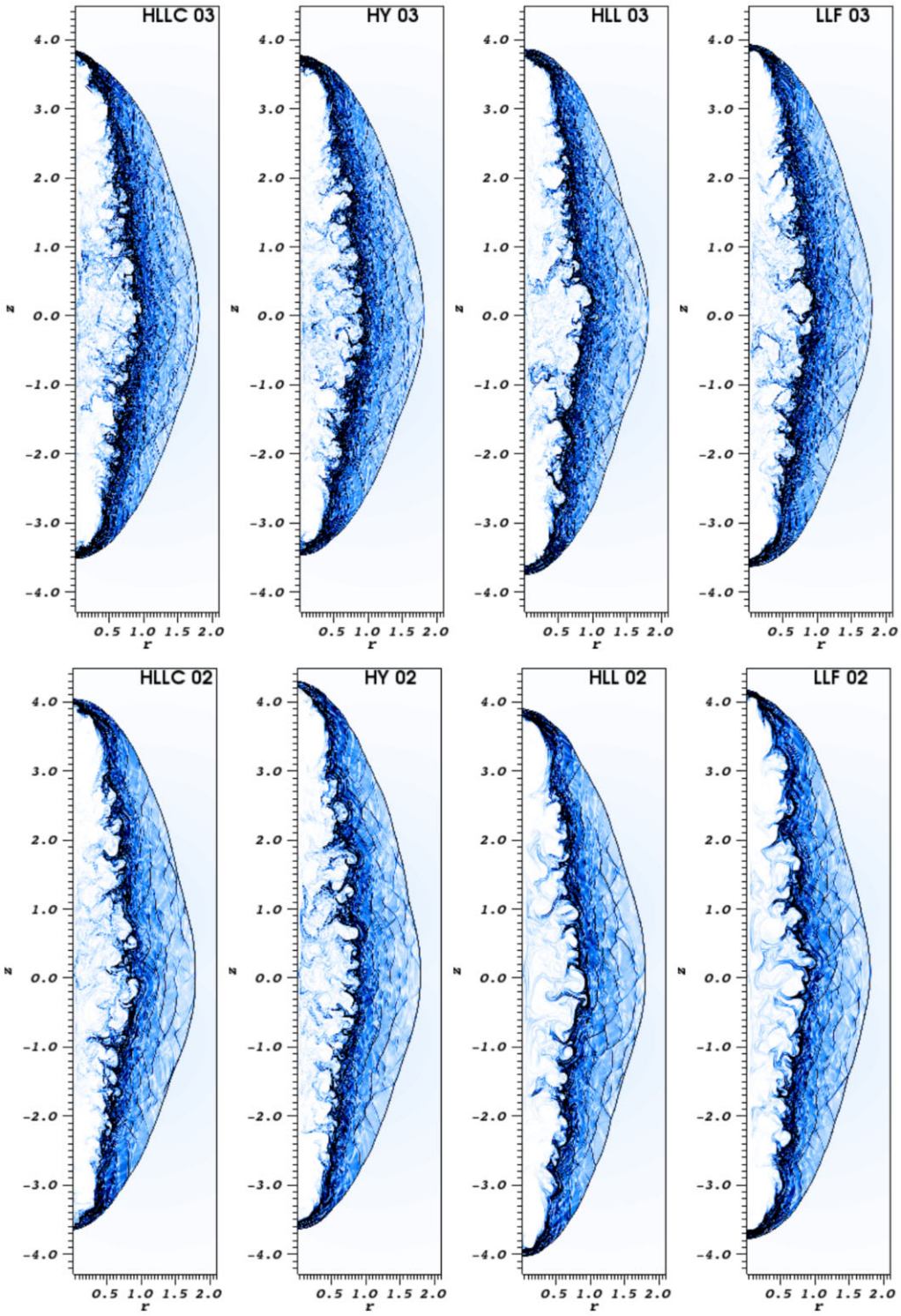

Figure 8. Schlieren plot of the density at $t = 1$ simulation unit (end of the active evolution). The figures show the small-scale density structure in the shocked shell of ambient medium, the cocoon, and at the contact discontinuity for each run. Axes are in simulation units (see Section 6).

simulation HLLC_03. This is due to the fact that the larger numerical diffusion of run LLF_02 in the subsonic regions of the fluid prevent the sharp resolution of shear and tangential waves, which results in the absence of a significant amount of the small-scale structure (e.g. see Mignone & Bodo 2005). Furthermore, fewer of the weaker shocks within the shocked shell of ambient medium are present in the most diffusive simulations (HLL_02 and LLF_02, see Fig. 8). Increasing the grid resolution of run LLF_02, for example, would better resolve the small-scale structure in the cocoons/shocked shell such that they represent those in run HLLC_03. However, increasing the resolution would be less computationally efficient compared to using a less diffusive solver with higher order spatial reconstruction.

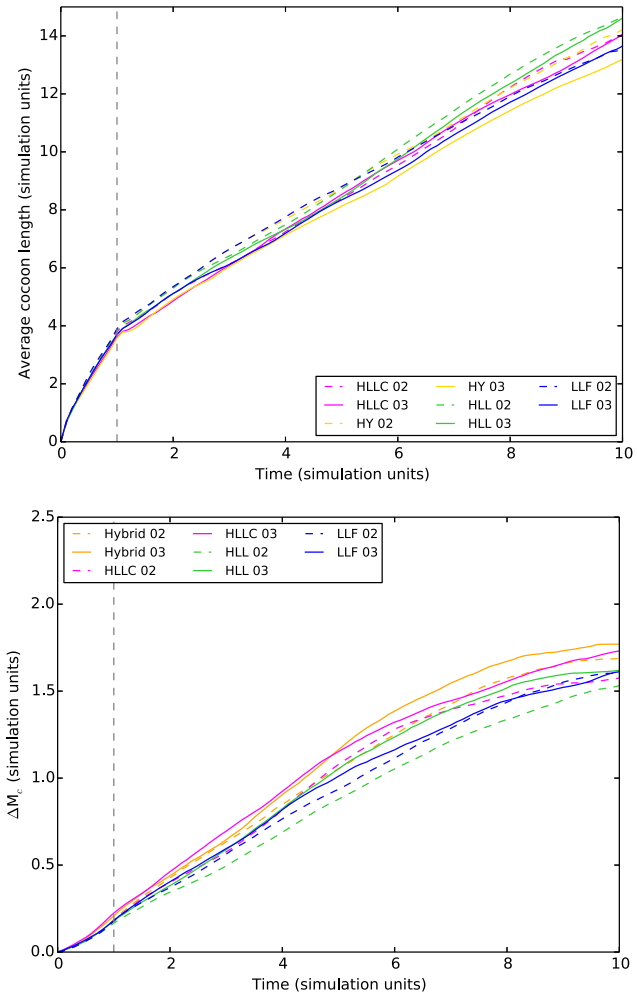

Figure 9. Top: mean cocoon length. The cocoon length is averaged over both sides of the bipolar source. Bottom: total cocoon mass. The axes of each figure are shown in simulation units (see Section 6). The grey vertical dashed line indicates the time at which the jet is switched off and begins the passive phase of its evolution.

10 JET DYNAMICS

The cocoon lengths averaged over both sides of the bipolar jet are shown in Fig. 9. During the active evolution the average differences between the propagation lengths of the cocoons for all simulations are negligible – The jets in all the simulations travel through the ambient medium with average advance speeds (averaged over both sides of the bipolar source) in the order of $0.01c$, while the bow shocks advance into the ambient medium with external Mach numbers (relative to the sound speed in the undisturbed external medium) of $\sim$3 during the active evolution of the jet.

Following the Sedov expansion phase at the very start of the passive evolution, the advance of the cocoon head slows and the bow shocks in all the simulations transition to subsonic propagation velocities in the order of $10^{-3}c$ due to the absence of ram pressure from the bulk flow of the jet. These propagation velocities remain roughly constant for the duration of the passive evolution. Though more variation in the average cocoon length (see Fig. 9) is seen between the runs towards the end of the passive evolution, all runs are in agreement at around the 10 per cent level.

11 COCOON PARAMETERS

Fig. 10 shows the volume-averaged cocoon/plume parameters. The ratio of the cocoon to ambient pressures (top left-hand panel of Fig. 10) gives a measure of the average strength of the shock driven into the external medium (Zanni et al. 2003). The shock strength is roughly the same for all combinations of Riemann solver and reconstruction method simulated. Following the initial pressure increase of the cocoon pressure at the very beginning of the active evolution, the cocoon pressure quickly adjusts to the ambient pressure as seen by the falling pressure ratios in Fig. 10. After the jets are switched off, a rough pressure balance between the cocoons and external medium begins to be established at around $t \approx 1.6$ simulation units (80 Myr).

The volume-averaged specific entropy $s = p/\rho^{5/3}$ is displayed in Fig. 10. The specific entropy presented is normalized by the entropy of the injected jet material, which has a value of 1293 simulation units. The entropy in the cocoons is increased due to shocks, and jet material that has crossed the jet termination shock can be of considerably high entropy compared to the entropy of the injected jet material, thus the cocoon entropy is largest during the active phase of the evolution. Once the jets are switched off and the terminal jet shock and internal shocks along the bulk flow dissipate and the cocoon entropy begins to drop due to the absence of the shocks and also due to the continual entrainment of lower entropy ambient medium into the cocoons/plumes (Reynolds et al. 2002). During the active phase of the evolution, the variation in specific entropy between the runs is driven by the strong shocks within the bulk flow of the jets and in the cocoon backflow.

11.1 Cocoon Volume and mass

The total cocoon volume is displayed in the bottom right-hand panel of Fig. 10. The cocoon volume rapidly increases during the active evolution due to the constant energy injection and the supersonic axial and radial expansions of the cocoon. During the active jet evolution, the cocoon volume is in excellent agreement for all simulations. Following the onset of the passive evolution, the cocoon volume begins to diverge at $t \approx 4.2$ simulation units and continues to do so for the remainder of the evolution. During the late stages of the passive evolution, the cocoon volume differs by as much as 20 per cent at $t =$ 10 simulation units for the different combinations of reconstruction schemes and Riemann solvers simulated. Simulations with second-order spatial reconstruction typically result in a smaller cocoon volume at a given evolution time towards the end of the passive evolution, and display more variation in the cocoon volume for a given Riemann solver. The simulations conducted with third-order spatial reconstruction on the other hand are in better agreement, differing by less than 10 per cent during the final stages of the passive evolution.

The cocoon mass as a function of evolution time is displayed in Fig. 9. Material in the shocked shell of ambient medium is significantly more dense than the material injected by the jet, thus its entrainment in the cocoon/plumes has a significant effect on the mass. The cocoon mass for the various runs starts to diverge as soon as the jet is switched off and begins the passive phase of its evolution. During the passive evolution the cocoon mass varies by at most $\sim$15 per cent for the different combinations of reconstruction scheme and Riemann solver used.

12 ENERGY PARTITIONING

As jets begin their active evolution into the ambient medium, the injected jet energy is converted to the internal and kinetic energies

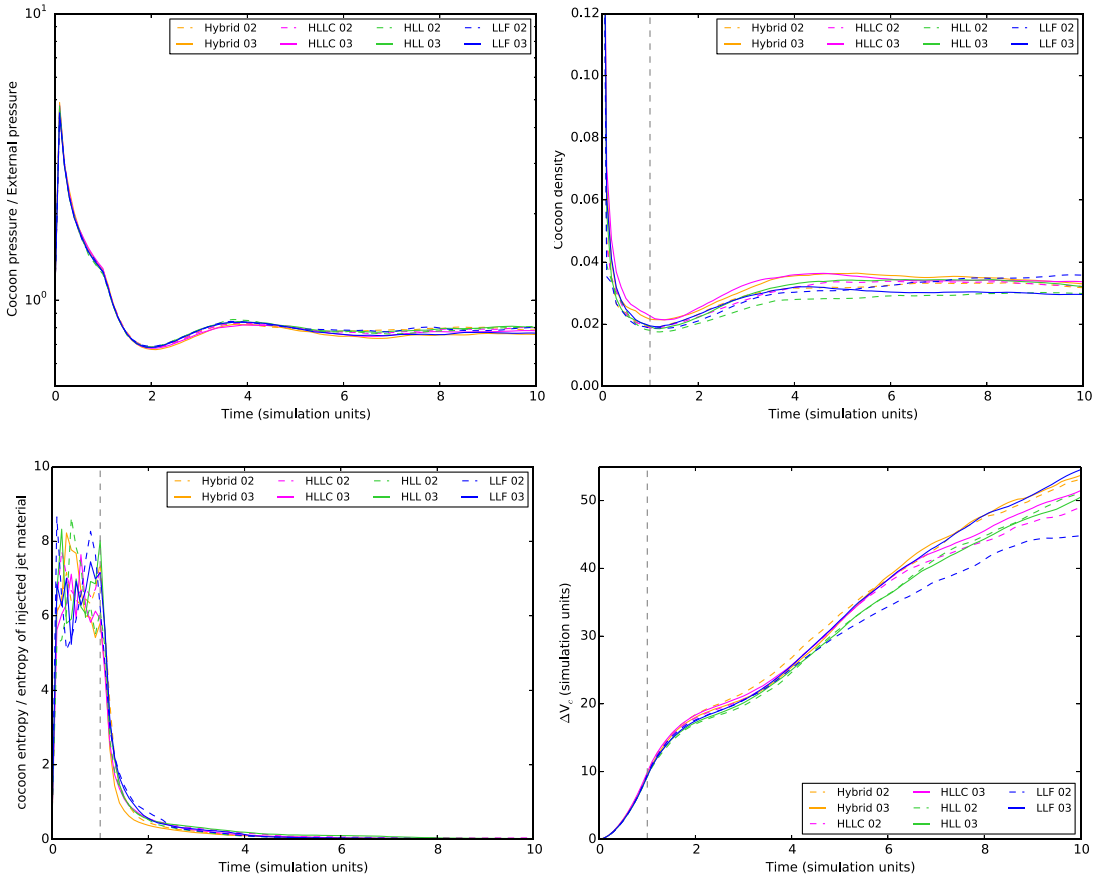

Figure 10. Volume-averaged cocoon parameters and cocoon volume (in scaled simulation units, see Section 6). Top left: ratio of the volume-averaged cocoon pressure the pressure in the external medium averaged over the corresponding cocoon volume. Top right: cocoon density. Bottom left: Volume-averaged cocoon entropy normalized by the entropy of the injected jet material. Bottom right: Cocoon volume. The grey vertical dashed line in each figure indicates the time at which the jet is switched off and begins the passive phase of its evolution.

of the external medium and cocoon and ambient medium. The internal and kinetic energy of the ambient medium increase as it is compressed and heated by the expanding bow shock. The potential energy of the external medium increases as regions of the external medium are displaced higher up in the potential well of the atmosphere by the advance of the bow shock, while the increase in the potential energy in the cocoon arises largely due to the entrainment and subsequent transport of ambient medium.

At the onset of the passive evolution, the absence of support from the jet thrust slows the advance of the bow shock, cocoon, and shocked shell of ambient medium. Consequently, the kinetic energy of the shocked shell falls sharply. For the cocoon, the absence of the large velocities associated with the active jet mean that the high backflow velocities are no longer sustained and the cocoon kinetic energy also decreases. During the Sedov expansion phase in the early passive evolution, the cocoon/lobes are still overpressured and with respect to the ambient medium and therefore continue to expand, continuing to drive the bow shock. (Reynolds et al. 2002)

Following the Sedov expansion phase, the plumes enter the buoyant phase of their evolution (Reynolds et al. 2002). The plumes slowly accelerate, rising in the atmosphere and increasing their kinetic energy. As the plumes rise they drag the surrounding ambient medium in their wake and entrain significant amounts of this material at their bases, lifting it higher into the potential well of the atmosphere and increasing the potential energy of the plumes. The bow shock weakens and the shocked shell of ambient medium expands, decreasing the pressure and consequently the internal energy of the shocked shell, as it attempts to recover hydrostatic equilibrium. The kinetic energy of the external medium begins to increase, though at a much shallower rate than that in the active evolution, as the plumes rise dragging ambient medium in their wake and the remnant bow shock broadens, increasing its spherical profile and encompassing larger regions of the surrounding ambient medium. As the ambient medium is continually lifted in the wake of the plumes the potential energy of the ambient medium also increases.

In order to study the effects of the choice of algorithm on the energetic properties of the simulations various energy components pertaining to the cocoon, ambient medium, and the entire grid are computed using as follows:

The cocoon internal energy is given by

$$E_{\rm int}(t) = \frac{1}{\gamma - 1} \int_{\rm (C,E,G)} p \, {\rm d}V. \tag{28}$$

The cocoon kinetic energy is given by

$$E_{\rm kin}(t) = \frac{1}{2} \int_{\rm (C,E,G)} \rho \upsilon^2 \, {\rm d}V, \tag{29}$$

while the gravitational potential energy of the cocoon given by

$$E_{\rm pot}(t) = - \int_{\rm (C,E,G)} \rho \Phi \, {\rm d}V. \tag{30}$$

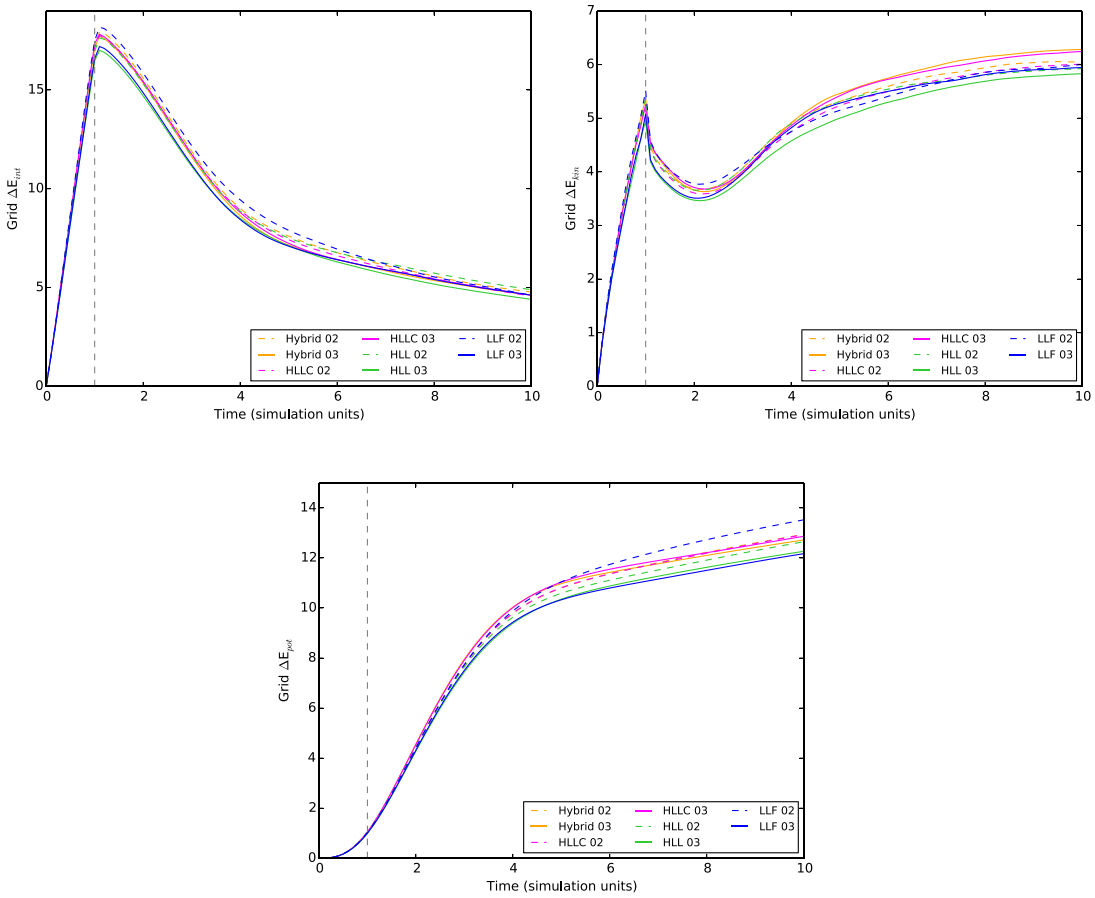

Figure 11. Clockwise from top left: Total internal, kinetic, and potential energy on the grid as a function of evolution time. The time at which the jets are switched off is indicated by the vertical dashed grey line at 1 simulation unit (50 Myr) in each of the images. Axes are in simulation units.

where C, E, and G correspond to the volumes comprising the cocoon, external medium, and entire grid (cocoon+external medium). The cocoon region is defined using the jet tracer f_j and is given by the volume inside of which $f_j > 10^{-3}$ (see Section 6.3). The external medium is defined as the regions in which $f_j \leq 10^{-3}$.

The internal $\Delta E_{\rm int}$, kinetic $\Delta E_{\rm kin}$, and potential $\Delta E_{\rm pot}$ for the cocoon, external medium, and the entire computational grid are all referenced from their values at initialisation, such that $\Delta E = E(t) - E(0)$.

The total internal kinetic and potential energies on the grid during both active and passive evolution are displayed in Fig. 11. During the active phase of the evolution, the simulations typically agree within 5 per cent for the grid internal, kinetic, and potential energy. During the passive phase, the differences in the grid energies becomes more apparent and the grid energy components can differ by ∼13 per cent.

The cocoon internal, kinetic, and potential energies are displayed as a function of evolution time in Fig. 12. The dependence of the cocoon energies on the Riemann solver is minimized for the simulations with third-order spatial reconstruction (the solid curves in Fig. 12). The jet tracer-based numerical definition used to define the cocoon region is particularly sensitive to the degree of mixing between the surrounding ambient medium and the jet material, and the level of numerical diffusion in the simulations. Thus, the simulations with second-order reconstruction typically show more variation in the cocoon energetics (as much as 30 per cent for the cocoon kinetic energy for example), particularly during the passive phase of the evolution.

The internal, kinetic, and potential energy of the ambient medium alongside the ratio of the total cocoon to total ambient medium energy as function of evolution time are shown in Fig. 13. During the active evolution, the internal energy of the ambient medium increases linearly with evolution time and is marginally larger for simulations with second-order spatial reconstruction.

The external energy components of the simulations using third-order spatial reconstruction (solid curves in Fig. 13) are typically in better agreement for the different Riemann solvers used during both the active and passive phase of the evolution. The second-order simulations with the least diffusive solvers (runs HLLC_02 and HLL_02) are in closer agreement to the simulations using third-order spatial reconstruction. For the Riemann solvers tested, more variation in the external energetics of the system is present for simulations with second-order spatial reconstruction, particularly for the internal and potential components of the external medium throughout the passive evolution.

13 SUMMARY AND CONCLUSIONS

Similar large-scale jet, cocoon, and shocked shell morphologies are produced with third- and second-order reconstruction schemes for a given Riemann solver. However, the third-order reconstruction methods provide better resolution of the small-scale features, turbulence, and shocks. Furthermore, for a given reconstruction scheme these small scale features are more sharply resolved with the least diffusive solvers; the HLLC and Hybrid Riemann solvers, though

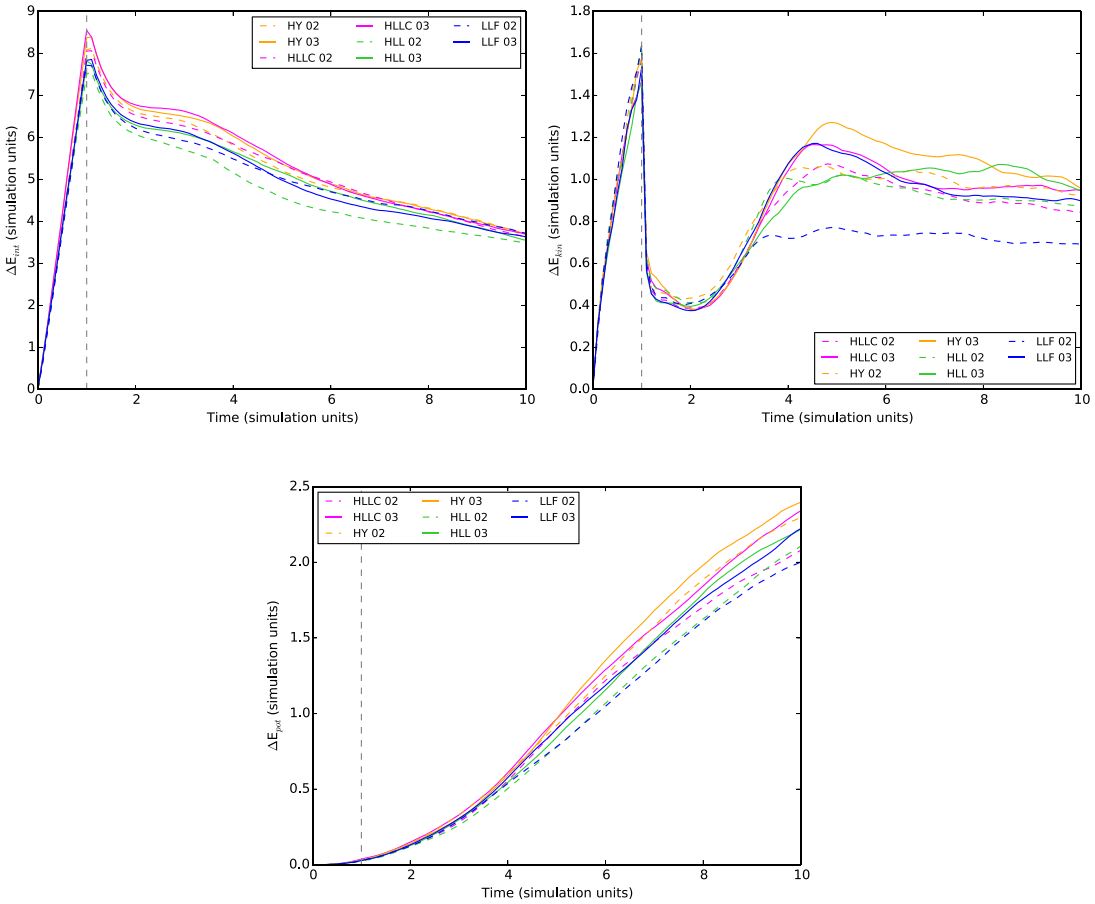

Figure 12. Cocoon energies. Clockwise from top left: cocoon internal, kinetic, and potential energy as a function of evolution time. The time at which the jets are switched off is indicated by the vertical dashed grey line at 1 simulation unit (50 Myr) in each of the images. Axes are in simulation units.

the dependence of the resolution of these features on the choice of solver is less significant than the dependence on the reconstruction method. The less diffusive solvers are therefore the most efficient for a given grid resolution in highly resolving the small scale features in the simulations and spread these flow features over fewer grid cells for a given reconstruction order. The more diffusive solvers would likely produce results similar to the HLLC and Hybrid solvers at the cost of increasing the grid resolution. The sharper representation of small scale features obtained with the HLLC and Hybrid solvers, however, comes at a cost of computational efficiency as for a given reconstruction order these solvers are the most time intensive.

The simulation results typically highlight a tendency for the temporal evolution of the parameters presented to be grouped by reconstruction order rather than favouring any clear systematic ordering in terms of solver. Hence, the solutions are more dependent on the reconstruction order used rather than the choice of solver, at least for the combinations of Riemann solver and spatial reconstruction schemes that are used here. Furthermore, the sensitivity of the solution on the reconstruction scheme is minimized as the reconstruction order is increased, and results obtained with third-order reconstruction are the most comparable across all the solvers tested. Converging behaviour between different Riemann solvers with increasing grid resolution alongside increasing convergence rates between the solvers with increasing reconstruction order has been noted by Mignone & Bodo (2005). The similarity of the solutions obtained, is typically within 10 per cent, for all quantities computed for simulations with third-order spatial reconstruction, and more-so for simulations with the HLLC and Hybrid solvers. The latter indicates a high degree of convergence between the HLLC and Hybird solvers. Thus, results obtained from suitably highly resolved simulations are likely to be produce comparable results, even with some of the more diffusive solvers.

That the least diffusive solvers obtain a sharper representation of the flow features alongside displaying both an increased intensity and a more numerous shock distribution are similar conclusions to those reached in Massaglia et al. (2003) regarding the effects of the exact Riemann solver, Roe, HLL(E), and LLF solvers with second-order reconstruction methods on simulated jets. We have extend the analysis presented in Massaglia et al. (2003) to a different set of solvers and across a selection of reconstruction methods using the FLASH code. We quantify the differences in solution for the large scale morphological and dynamical properties of the jets, and for the energetics across both a passive and active jet evolution. In agreement with Massaglia et al. (2003), we find that during the active phase for a given reconstruction method the difference in solution due to the choice of Riemann solver is typically small for many of the quantities presented. Significantly, more variation in the computed quantities is seen across the different simulations during the passive phase of the evolution. This is particularly the case for the simulations in which second-order spatial reconstruction is used, however, these differences are small, with variations typically smaller than 20 per cent for the majority of the computed quantities.

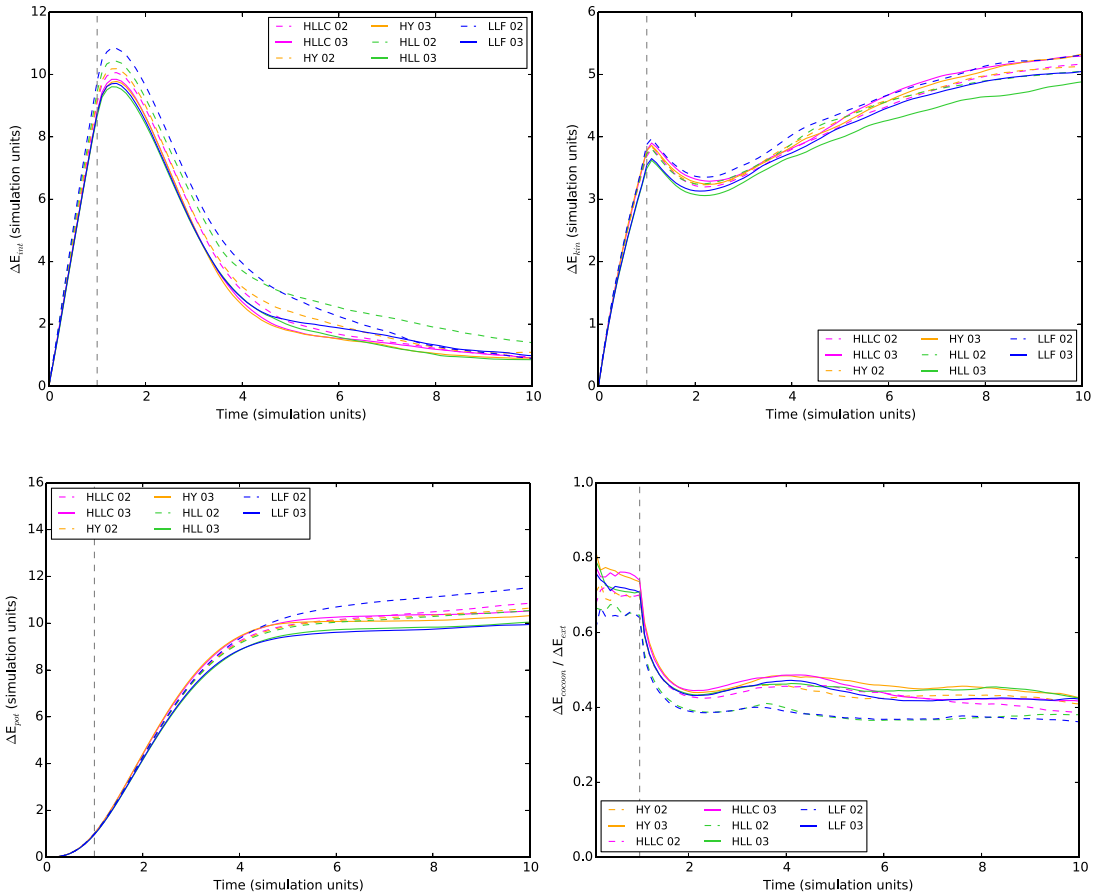

Figure 13. External medium energies. Top left: Internal energy. Top right: Kinetic energy. Bottom left: Potential energy. Bottom right: Ratio of the total cocoon energy to the total energy in the external medium. Axes are in simulation units.

We find a high level of convergence for runs HLLC_03 and HY_03. However, since we find that the Hybrid solver is the least efficient of the Riemann solvers tested we choose to use the HLLC solver with third-order spatial reconstruction for our future work. All the combinations of Riemann solver and spatial reconstruction scheme presented here are in very good agreement for both evolutionary phases and therefore the different schemes can be used interchangeably and directly compared for the type of problem simulated, provided that the grid is suitably resolved – it is important to note that the simulations presented are conducted with a high level of grid refinement. Simulations conducted at lower levels of refinement may show significantly more variation for the different combinations of Riemann solver and spatial reconstructions used.

ACKNOWLEDGEMENTS

This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol.[3] The FLASH (M)HD code used in this work was in part developed by the DOE NNSA-ASC OASCR Flash Center at the University of Chicago. GM acknowledges support from the Netherlands Organisation for Scientific Research (NWO) VICI award, grant no. 639.043.513 and a Netherlands Research School for Astronomy (NOVA), Virtual Institute of Accretion (VIA) postdoctoral fellowship. This research has used the VISIT visualization package (Childs et al. 2012).

[3]See http://www.bris.ac.uk/acrc/.

DATA AVAILABILITY

Processed data used to produce Figs 9–13 presented in this work will be made available at http://doi.org/10.5281/zenodo.3993052. The raw simulation data produced using the FLASH (M)HD code can be made available upon request – please contact the lead author.

REFERENCES

Arnold C. N., Arnett W. D., 1986, ApJ, 305, L57
ASC FLASH C., 2012, University of Chicago, Chicago
Balsara D. S., Spicer D., 1999, J. Comput. Phys., 148, 133
Blandford R. D., Rees M. J., 1974, MNRAS, 169, 395
Childs H. et al., 2012, in Bethel E. W., Childs H., Hansen C., eds, High Performance Visualization–Enabling Extreme-Scale Scientific Insight. CRC Press, Boca Raton FL, p. 357
Clarke D. A., Norman M. L., Burns J. O., 1986, ApJ, 311, L63
Clarke D. A., Norman M. L., Fiedler R. A., 1994, ZEUS-3D User Manual, National Center for Supercomputing Applications Technical Report no. 15
Colella P., Woodward P. R., 1984, J. Comput. Phys., 54, 174
Einfeldt B., 1988, SIAM J. Numer. Anal., 25, 294
English W., Hardcastle M. J., Krause M. G. H., 2016, MNRAS, 461, 2025
English W., Hardcastle M. J., Krause M. G. H., 2019, MNRAS, 490, 5807
Fryxell B. et al., 2000, ApJS, 131, 273

Gentry R. A., Martin R. E., Daly B. J., 1966, J. Comput. Phys., 1, 87
Godunov S. K., 1959, Mat. Sb., 89, 271
Guan X., Li H., Li S., 2014, ApJ, 781, 48
Gull S. F., 1976, . Ph.D. Dissertation, Univ. Cambridge
Harten A., Lax P. D., van Leer B., 1983, SIAM Rev., 25, 35
Heinz S., Brüggen M., Young A., Levesque E., 2006, MNRAS, 373, L65
Heinz S., Brüggen M., Ruszkowski M., Young A., Levesque E., 2008, in Rector T. A., De Young D. S., eds, ASP Conf. Ser. Vol. 386, Extragalactic Jets: Theory and Observation from Radio to Gamma Ray. Astron. Soc. Pac., San Fransisc, p. 327
Hughes P. A., Miller M. A., Duncan G. C., 2002, ApJ, 572, 713
Jiang G.-S., Chin Wu C., 1999, J. Comput. Phys., 150, 561
Kemm F., 2018, Appl. Math. Comput., 320, 596
Keppens R., Nool M., Tóth G., Goedbloed J. P., 2003, Comput. Phys. Commun., 153, 317
Keppens R., Meliani Z., van Marle A. J., Delmont P., Vlasis A., van der Holst B., 2012, J. Comput. Phys., 231, 718
Kong C., 2011, MSc Dissertation, Univ. Reading
Lax P. D., 1954, Commun. Pure Appl. Math., 7, 159
Lee D., 2013, J. Comput. Phys., 243, 269
LeVeque R. J., 2002, Finite Volume Methods for Hyperbolic Problems, Vol. 31. Cambridge Univ. Press, Cambridge
Li S., 2005, J. Comput. Phys., 203, 344
Lind K. R., 1987, PhD thesis, California Institute of Technology
Lind K. R., Payne D. G., Meier D. L., Blandford R. D., 1989, ApJ, 344, 89
Lind K. R., Meier D. L., Payne D. G., 1994, BAAS, 26, 922
Martí J. M., Müller E., 2015, Living Rev. Comput. Astrophys., 1, 3
Martí J. M., Müller E., Font J. A., Ibáñez J. M. Z., Marquina A., 1997, ApJ, 479, 151
Massaglia S., Zurlo N., Bodo G., 2003, Mem. Soc. Astron. Ital. Suppl., 3, 335
Mendygral P. J., Jones T. W., Dolag K., 2012, ApJ, 750, 166
Mignone A., 2007, J. Comput. Phys., 225, 1427
Mignone A., Bodo G., 2005, MNRAS, 364, 126
Mignone A., Bodo G., Massaglia S., Matsakos T., Tesileanu O., Zanni C., Ferrari A., 2007, ApJS, 170, 228
Mignone A., Bodo G., Ugliano M., 2012, in Vázquez-Cendón E., Hidalgo A., Navarro P. G., Cea L., eds, Numerical Methods for Hyperbolic Equations. CRC Press, London, p. 219
Mukherjee D., Bicknell G. V., Sutherland R., Wagner A., 2016, MNRAS, 461, 967
Musoke G., Young A. J., Molnar S. M., Birkinshaw M., 2020, MNRAS, 494, 5207
Norman M. L., Winkler K.-H. A., Smarr L., Smith M. D., 1982, A&A, 113, 285
O'Neill S. M., Beckwith K., Begelman M. C., 2012, MNRAS, 422, 1436
Quirk J. J., 1994, Int. J. Numer. Methods Fluids, 18, 555
Reynolds C. S., Heinz S., Begelman M. C., 2002, MNRAS, 332, 271
Robertson B. E., Kravtsov A. V., Gnedin N. Y., Abel T., Rudd D. H., 2010, MNRAS, 401, 2463
Roe P., 1981, J. Comput. Phys., 43, 357
Rusanov V. V., 1961, Zh. Vychisl. Mat. Mat. Fiz., 1, 267
Saxton C. J., Bicknell G. V., Sutherland R. S., Midgley S., 2005, MNRAS, 359, 781
Shen Z., Yan W., Yuan G., 2016, J. Comput. Phys., 309, 185
Stone J. M., Norman M. L., 1992a, ApJS, 80, 753
Stone J. M., Norman M. L., 1992b, ApJS, 80, 791
Stone J. M., Gardiner T. A., Teuben P., Hawley J. F., Simon J. B., 2008, ApJS, 178, 137
Tchekhovskoy A., Bromberg O., 2016, MNRAS, 461, L46
Toro E. F., 2009, The HLL and HLLC Riemann Solvers. Springer Berlin Heidelberg, Berlin, Heidelberg, p. 315
Toro E. F., Spruce M., Speares W., 1994, Shock Waves, 4, 25
Tóth G., Odstrčil D., 1996, J. Comput. Phys., 128, 82
VanLeervan Leer B., 1979, J. Comput. Phys., 32, 101
Walg S., Achterberg A., Markoff S., Keppens R., Meliani Z., 2013, MNRAS, 433, 1453
Williams A. G., Gull S. F., 1984, Nature, 310, 33
Wilson J. R., 1979, in Smarr L., ed., Sources of Gravitational Radiation. Cambridge University Press, Cambridge, p. 423
Wilson M. J., Scheuer P. A. G., 1983, MNRAS, 205, 449
Zanni C., Massaglia S., Bodo G., Rossi P., Capetti A., Ferrari A., 2003, Mem. Soc. Astron. Ital. Suppl., 1, 155
Zanni C., Murante G., Bodo G., Massaglia S., Rossi P., Ferrari A., 2005, A&A, 429, 399

This paper has been typeset from a TEX/LATEX file prepared by the author.