

Multi-patch methods in general relativistic astrophysics – I. Hydrodynamical flows on fixed backgrounds

Burkhard Zink,^{1,2} Erik Schnetter,^{1,2} and Manuel Tiglio^{1,2,3,4}

¹Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, USA

²Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA 70803, USA

³Department of Physics, University of Maryland, College Park MD 20742, USA

⁴Center for Scientific Computation and Mathematical Modeling,
University of Maryland, College Park MD 20742, USA

Many systems of interest in general relativistic astrophysics, including neutron stars, accreting compact objects in X-ray binaries and active galactic nuclei, core collapse, and collapsars, are assumed to be approximately spherically symmetric or axisymmetric. In Newtonian or fixed-background relativistic approximations it is common practice to use spherical polar coordinates for computational grids; however, these coordinates have singularities and are difficult to use in *fully relativistic* models. We present, in this series of papers, a numerical technique which is able to use *effectively spherical* grids by employing multiple patches. We provide detailed instructions on how to implement such a scheme, and present a number of code tests for the fixed background case, including an accretion torus around a black hole.

PACS numbers: 95.30.Lz, 47.11.Df, 04.25.Dm

I. INTRODUCTION

Numerical simulations have become one of our most valuable tools in building and refining models of compact astrophysical objects and their environments, which are commonly associated with high-energy events like gamma-ray bursts, active galactic nuclei and jets, X-ray binaries and supernovae. These rather exotic systems form one of the arguably most exciting physical laboratories we know of, where general relativity, nuclear physics, transport of radiation, magnetohydrodynamics (and so on) contribute to their dynamical properties.

The rapid advance of computing performance has made it possible to simulate increasingly sophisticated problems.¹ But even with current high-performance supercomputers building general relativistic three-dimensional models poses a formidable challenge, since the complexity of solving an hyperbolic problem of dimension n scales with the linear spatial resolution h like $O(1/h)^{n+1}$. It is therefore imperative to investigate the application of advanced, efficient numerical techniques in astrophysical models.

This series of papers will focus on a particular approach in which the computational domain is understood as a manifold covered by several distinct coordinate maps called *patches*. Above all else, this approach admits to cover spheres with smooth, and in particular singularity-free, coordinates, and also allows to employ many of the advantages of spherical polar grids, like

the decoupling of radial and angular resolution and intrinsically spherical domain boundaries, without sharing their major disadvantages.

Multi-patch techniques have been used in different contexts in general relativity (see, for example, [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26] and references therein). The multi-patch infrastructure used for this paper [27] admits both overlapping patches and abutting ones (also called *blocks*), with the latter typically used when solving systems of equations with smooth solutions. In those cases the inter-block boundary information is transported by the *simultaneous approximation technique* [28], combined with high-order finite difference operators satisfying an algebraic property called *summation by parts* and associated dissipation operators. With these techniques the evolution system at the block-interfaces is decomposed into its characteristic structure and any numerical mismatch between modes at these interfaces is dissipated away in time and/or with increasing resolution in a numerically stable way (see [29, 30] for more details). It has been possible to perform very accurate simulations of scalar fields on a Kerr background [31] and fully non-linear simulations of distorted black holes [32] with this approach.

As mentioned, these multi-patch techniques are designed for systems with smooth solutions, as is usually the case when solving the Einstein equations in vacuum. To model a larger class of astrophysical objects of interest one needs to deal with the presence of matter and the development of shocks in multi-patch simulations, which is precisely the goal of this series of papers.

Here we treat *hydrodynamical flows* in the so-called test-fluid approximation, i.e., on a specified spacetime geometry. While this has interesting applications of its own, like the dynamics of non-self gravitating accretion disk models around black holes [11, 33, 34, 35] and the normal mode spectrum of isolated neutron stars [36], we

¹ For some recent results in general relativistic flow models see [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13], though this list is far from being complete.

will focus here on *test cases with well-defined error functions* (either Riemann problems or stationary solutions), to demonstrate the ability of the code to transport shock fronts across interfaces, and to keep stationary solutions near equilibrium. In later papers in the series we will consider the technique for the fully coupled evolution system and magnetized flows.

This paper is organized as follows: In Section II we give an overview of the multi-patch setup, the evolution system, and the discrete techniques we are using. In Section III, we give a specific description how to implement such a scheme, which should be helpful for practitioners who would like to adopt this approach. In Section IV we discuss a number of multi-patch systems useful for test and production simulations. In Section V we present the results from a number of code tests, including shock tubes, rotating stars and an accretion torus around a black hole. Finally, in Section VI, we summarize our results.

II. THEORY AND DISCRETE TECHNIQUES

A. The multi-patch setup in general relativity

In general relativity (GR) one starts with a spacetime manifold (M, g) , where M denotes the set of events and g the metric tensor field. Since the manifold is endowed with a differentiable structure D , an obvious choice for a continuum multi-patch model is, given an atlas $[A] \in D$, a subset $A^0 \subseteq A$ of charts $\phi : M \rightarrow \mathbb{R}^4$ which cover the domain of interest (see, e.g., [37]).

In our computational setup we will have, for practical reasons, an additional element. We will assume that we are interested in solving our system of equations in a spacetime region $M_0 \subseteq M$ which can be covered by a *single* chart ϕ_G . That is, we will assume that there is a *global coordinate system*. This requirement is not fundamental to using multi-patch techniques, but makes it easier to set up initial data, visualize results, and transport information between local patches, as discussed below. If needed, the assumption of such global system can be eliminated without any of the techniques of this paper changing. On the other hand, whenever we assume the existence of such global system, for definiteness we will typically choose it to be of “Cartesian” type.

In differential geometry language, a differentiable structure D_L can be associated with the global region of interest $\phi_G(M_0)$. A multi-patch setup, given an atlas $[A_L] \in D_L$, is then a subset of A_L covering $\phi_G(M_0)$. For any $\phi_{GL} \in A_L$, the product chart $\phi_L \equiv \phi_{GL} \circ \phi_G$ defines a *local coordinate system* on that local patch. Fig. 1 illustrates this relationship.

A point of some technical relevance is the use of local or global tensor bases. At each point in M_0 , tensorial quantities can be written in terms of global or local coordinates (in differential geometry language, the tangent space bases associated with ϕ_G or ϕ_L , respectively,

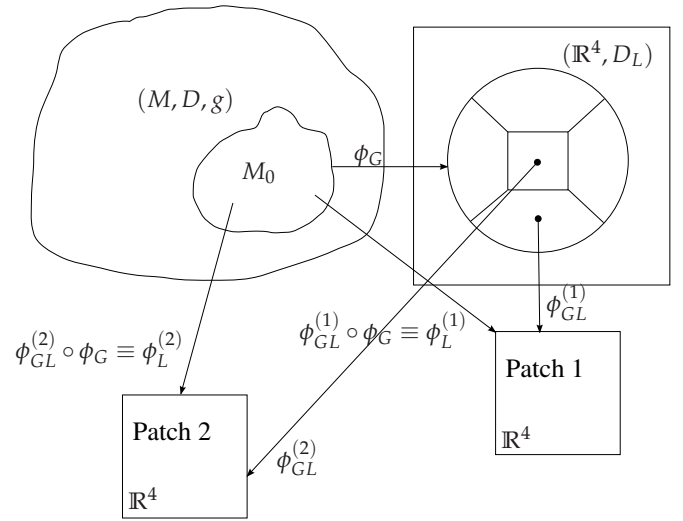


Figure 1: The general-relativistic multi-patch model used in this paper. The space-time sub-domain $M_0 \subset M$ of interest is assumed to be covered by a single *global coordinate system* ϕ_G . The target domain $\phi_G(M_0) \subset \mathbb{R}^4$ is then further covered by several patches associated with the charts ϕ_{GL}^n , which give rise to the *local coordinate systems* ϕ_L^n .

can be used). The latter appears to be the more natural choice and will be employed to represent the hydrodynamical variables in this paper to enforce conservation when this is available. A global system is still useful (though not strictly necessary) to serve as a “reference” frame. On the other hand, there is no obvious conservation law for the metric sector, and therefore there is no advantage in using local coordinates. In fact, it is easier to use global ones, and therefore when we solve for the metric we do so for its global components (see, e.g., [27, 32]). In such case, a standard finite-difference discretization delivers partial derivatives consistent with ϕ_L which are then transformed by the Jacobians associated with ϕ_{GL} .

Once the multi-patch system has been set up, the system of equations to be solved can be discretized on each patch and two types of boundary conditions appear: those associated with the outer boundaries of the whole domain M_0 , and those connecting neighboring patches. In our applications each point in M_0 will in general be associated with the *interior* of exactly one patch, and with the boundary region of several (two or more) patches (see Fig. 2). Multi-patch techniques can generally make use of the overlap region to define *ghost zones*, or they use the interfaces to establish a common surface for the communication of variables.

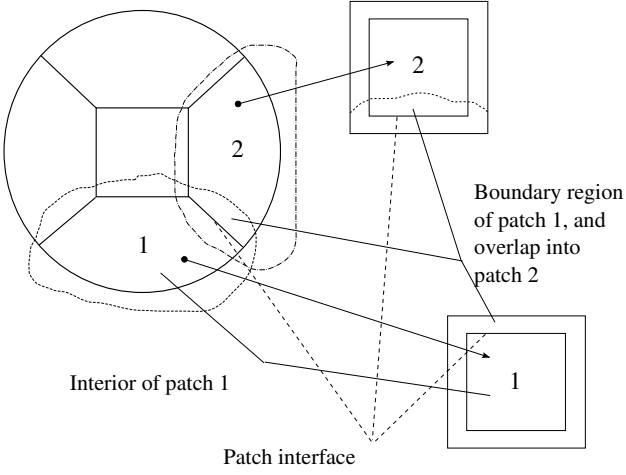


Figure 2: The notions of interior, interface and boundary region associated with a patch. The global coordinate domain $\phi_G(M_0)$ is decomposed into the patch interiors as indicated in the figure, with interfaces adjoining two interiors. The charts ϕ_L^n therefore define an interior and a boundary region in $\phi_L^n(M_0)$.

B. General relativistic hydrodynamics

1. Evolution system

We assume an energy-momentum tensor of the usual form [38]

$$T^{ab} = (\rho + u + P)u^a u^b + P g^{ab} \quad (1)$$

where ρ is the fluid's rest-frame mass density, u is the internal energy density, u^a the four-velocity, P the isotropic pressure and g^{ab} the contravariant components of the spacetime metric. Given a patch with coordinate system ϕ_L the conservation laws of mass and energy-momentum

$$\nabla_a(\rho u^a) = 0 \quad (2)$$

$$\nabla_a T^{ab} = 0 \quad (3)$$

are transformed into an evolution system by a 3+1 split (see, e.g., [39])

$$\partial_t(\sqrt{-g}\rho u^t) + \partial_i(\sqrt{-g}\rho u^i) = 0 \quad (4)$$

$$\partial_t(\sqrt{-g}T^t_a) + \partial_i(\sqrt{-g}T^i_a) = \sqrt{-g}T^c_d \Gamma^d_{ac} \quad (5)$$

where t is the time coordinate, i is a *local* space coordinate, $g \equiv \det(g_{ab})$ is the determinant of the spacetime metric, and Γ^d_{ac} are the Christoffel symbols associated with it. We therefore promote the expressions $D \equiv \sqrt{-g}\rho u^t$ and $Q_a \equiv \sqrt{-g}T^t_a$ to evolution variables, and collect them into the tuple $w = (D, Q_a)$. The equations can then be expressed in the flux conservative form

$$\partial_t w + \partial_i F^i(v(w)) = s(v(w)). \quad (6)$$

Here, the tuple $v = (\rho, u, u^x, u^y, u^z, P)$ denotes a choice of *primitive variables*, which, in general, are an implicit function of the *conserved variables* w . The *fluxes* $F^i(v)$ and *sources* $s(v)$ are expressed in terms of the primitive variables, which makes it necessary to obtain the primitive from the conserved variables at each evolution step.

For calculating the primitive variables, we make use of the “2D scheme” from [40], since it easily lends itself to the inclusion of magnetic fields. The conversion is a non-linear root-finding problem, which is being solved using a Newton-Raphson scheme, and as such is one of the most delicate parts of the implementation. The Newton-Raphson scheme only converges given a sufficiently close initial guess: We use the value of the primitive variables at the last sub-step for this purpose. Also, in some cases the conserved variables could potentially obtain values which are not compatible with any set of physical primitive variables and, finally, the round-off errors may lead to subtle problems, e.g. a Lorentz factor of $\approx 1 - 10^{-16}$. Most of this is discussed in [40] and in the accompanying source code examples.

The pressure function $P = P(\rho, u)$ is chosen according to the physical properties of the fluid, which is in general determined by the equation of state. For purposes of this publication, we will assume the pressure to be obtained from the *gamma law* $P = (\Gamma - 1)u$.

2. Discretization

On any patch, we use a standard finite-volume scheme [41] to update the evolution variables. The computational domain is broken into blocks, and each grid point is assumed to coincide with the center of a finite cell. The primitive variables associated with a grid point are naturally interpreted in terms of volume averages over the cell. The evolution system eqn. 6 can then be integrated over the cell to yield the weak form of the equations, which represents the update of the volume averages in terms of fluxes across the cell interfaces and source terms:

$$\begin{aligned} & \partial_t \int_{(a_1, b_1, c_1)}^{(a_2, b_2, c_2)} w \, da \, db \, dc \\ &= - \int_{(b_1, c_1)}^{(b_2, c_2)} \left(F^1(v)|_{a_2} - F^1(v)|_{a_1} \right) db \, dc \\ & \quad - \int_{(a_1, c_1)}^{(a_2, c_2)} \left(F^2(v)|_{b_2} - F^2(v)|_{b_1} \right) da \, dc \\ & \quad - \int_{(a_1, b_1)}^{(a_2, b_2)} \left(F^3(v)|_{c_2} - F^3(v)|_{c_1} \right) da \, db \\ & \quad + \int_{(a_1, b_1, c_1)}^{(a_2, b_2, c_2)} s(v) \, da \, db \, dc \end{aligned} \quad (7)$$

The fluxes across the cell interfaces are obtained in two steps: First, the primitive variables are extrapolated to their (left and right) interface values using a *reconstruction* algorithm, and then the thus-defined local Riemann problem is solved with an approximate method. For purposes of reconstruction, we use the MC (monotonized central) algorithm. If u_i is the value of the primitive variables at zone i , we define the reconstructed quantities v_L and v_R at the interface location $i + \frac{1}{2}$ as [42]

$$\begin{aligned} v_L &\equiv v_i + \bar{\Delta}_i/2 \\ v_R &\equiv v_{i+1} - \bar{\Delta}_{i+1}/2 \\ \bar{\Delta}_i &\equiv \begin{cases} \text{sgn}(\Delta_{i+1}) \min(2|\Delta_i|, 2|\Delta_{i+1}|, \\ |\Delta_i + \Delta_{i+1}|/2) & \Delta_i \Delta_{i+1} \geq 0 \\ 0 & \text{otherwise} \end{cases} \\ \Delta_i &\equiv v_i - v_{i-1}. \end{aligned} \quad (8)$$

The resulting local Riemann problem defined by the states v_L and v_R is approximated using the HLL (Harten, Lax, van Leer) flux formula [43]

$$\begin{aligned} F^i &\equiv (c_{\min} F^i(v_R) + c_{\max} F^i(v_L) \\ &\quad - c_{\min} c_{\max} (w_R - w_L)) (c_{\min} c_{\max})^{-1} \\ c_{\min} &\equiv -\min(0, c_R^-, c_L^-) \\ c_{\max} &\equiv \max(0, c_R^+, c_L^+) \end{aligned} \quad (9)$$

Here c^- and c^+ are the minimal and maximal characteristic speeds associated with the variables $v_{L,R}$, and $w_{L,R}$ are the conserved variables obtained from $v_{L,R}$. This flux formula has the advantage that it does not require the full characteristic decomposition of eqn. 5, and is thus well-suited to be extended to the more complicated case involving magnetic fields [44]. The characteristic speeds are obtained from the rest-frame sound speed of the fluid by a Lorentz transformation to the local coordinate frame [45].

A particular difficulty in solving general relativistic hydrodynamics problems are regions of very low density [46, 47], e.g. those outside a neutron star. In those regions, we use an artificial atmosphere of low density to make the problem tractable. This atmosphere effectively acts as a boundary condition on the stellar material.

3. Boundary treatment

To use an unmodified finite-volume scheme also at the boundaries of the interior patch domain, we follow the same approach typically used in adaptive mesh refinement implementations: we introduce *ghost zones* in the boundary region which are updated using an interpolation scheme. The results in this paper have been obtained with a first-order operator, which does

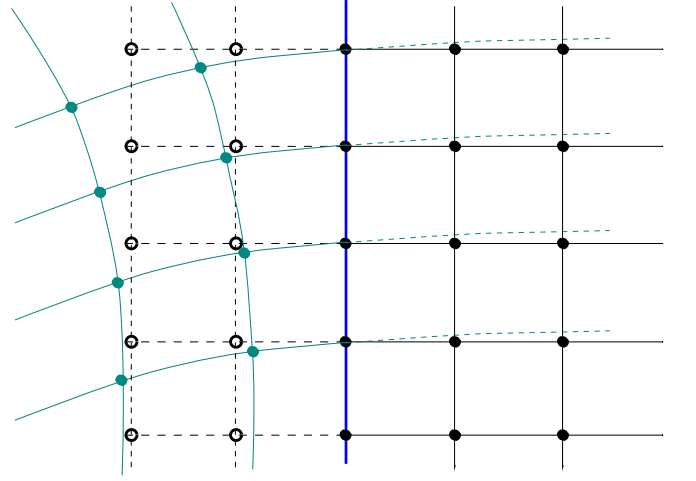


Figure 3: Illustration of the patch interface boundary treatment. The interface, here represented by the vertical thick blue line in the center, separates patch L (left) and patch R (right). The coordinate lines of both patches are represented in the local coordinates of patch R, which makes the lines of L appear curved in general. To evolve the system on patch R, *boundary ghost zones*, here represented by non-filled circles, are introduced as a linear extrapolation of patch R's coordinate lines, and, before the time update is performed, receive data from a data interpolation operation on patch L. Afterwards, the data is transformed to the tensor basis defined by patch R's coordinate system via the transformation map.

not introduce new extrema and is compatible to the (at most) second-order accurate MC reconstruction technique. The patch interface boundary treatment is illustrated in Fig. 3.

Since the patches have different local coordinate systems, we need to use the transformation maps between the patches to obtain the local representation of interpolated quantities. Internally, first a transformation to the global coordinate system is performed, followed by a separate transformation to the appropriate local system. For the case of general relativistic hydrodynamics on specified backgrounds, we interpolate the set of primitive variables for purposes of interface reconstruction. These quantities are components of tensors and therefore subject to trivial transformation laws [37].

The decomposition of each patch into sub-domains, for purposes of implementing a distributed computing model, introduces additional boundaries which are treated by introducing ghost zones. No interpolation or coordinate transformation is needed in this case, and the synchronization operation copies boundary data between these sub-domains.

The outer boundaries of the computational domain are also handled with ghost zones. For purposes of imposing an outflow boundary condition, we copy data from the first cell inside the computational domain to the ghost zones [48], though we will typically prefer to select a grid setup where material ejected from it is en-

tirely contained on the grid during the course of the evolution. In this case, the artificial atmosphere is the effective boundary for the fluid. If material should leave the computational domain, the exact form of the boundary condition is important only in those cases where there is a significant back-reaction of the material on the central object, or other regions of interest.

4. Setup of initial data

For convenience, initial data is always expressed first in terms of the *global* coordinate system, and mapped to the patch locations in this basis. Afterwards, the coordinate transformation to the appropriate local basis is performed if the quantity is not a scalar field. All initial data specifications in this paper will therefore also be written in an appropriate global system.

C. Computational framework

The code has been implemented as a module into the Cactus computational framework [49, 50], and using the Carpet driver [51, 52] to administrate the multi-patch infrastructure. In our implementation (as stated above), the Carpet driver reserves a number of *boundary ghost zones* beyond each patch boundary, which are then filled using interpolation and tensor transformation. For a first order interpolation scheme, each boundary ghost zone is located in exactly one cell in the interior of another patch. However, if the scheme was to be extended to higher order interpolation operators, a non-central stencil would need to be applied.

D. Limitations of our approach

In this section, we collect the most important known limitations of the approach followed in this paper.

The most obvious limitations are clearly those related to the physical model: the assumption of a fixed spacetime neglects motions of the gravitational field and therefore suppresses all gravitational instabilities, and is only strictly justified in code tests involving equilibrium systems. Our fluid model, besides making the assumptions common for all hydrodynamical approximations, models *one* fluid component without tangential stresses, and our choice of the pressure relation neglects micro-physical properties. This paper does not include magnetic fields either, although we will turn to full GRMHD in a subsequent publication.

On the numerical side, we have a number of restrictions. While many of these are common to many approaches in the field, we find it useful to give a detailed list here:

- The scheme that we use is at best second order accurate. In particular, it drops to local first order

near maxima and stellar surfaces. The global convergence order might be less than two in practical applications. This approach is very robust, but may be too inaccurate for very long term simulations (hundreds of dynamical timescales), or turbulent phenomena like the MRI [53]. There are higher order schemes available [54], and we are evaluating using such techniques.²

- The inter-patch boundary treatment does not trivially lend itself to higher order implementations, since we need a non-oscillatory higher order interpolation operator to fill the boundary ghost zones. A compromise would be to use Lagrangian operators, but drop to trilinear interpolation if unphysical variables are produced, or use an ENO scheme.
- The outer boundary conditions are only approximate, and may lead to unphysical variables (see [39] for an improved scheme with densitized variables). This may have practical and fundamental consequences. On a practical level, undesirable artifacts may appear at the boundaries, though in stationary systems, those are only related to artefactual flows leaving the compact support of the equilibrium object. However, if the system does show physical outflows across the boundary, those will be modeled only approximately. For supersonic or even causally disconnected flows, however, this restriction may not be dynamically relevant
- The artificial atmosphere is used to employ a unified computational scheme. However, to produce an exact discrete representation of an equilibrium star or disk, the discrete scheme would need to be modified near the surface. The addition of atmospheric fluid usually introduces error levels which are significantly smaller than errors from the internal dynamics of the star in real applications, but for equilibrium problems, which may be dominated by the surface errors, the code may not show convergence to the correct solution even when reducing the atmospheric density.
- The transformation from conserved to primitive variables works well in most cases, but can be a cause of considerable difficulty in some situations. This problem may be more severe when using a tabulated equation of state and coupled magnetic fields; in fact, the technique we currently use may

² Higher order reconstruction operators like PPM [55] do not increase the overall accuracy but may decrease local error levels. We do have a simplified implementation of PPM in our code (without flattening), but we did not find it superior in the particular test cases discussed here.

need a considerable extension to also operate in these cases [56].

- The multi-patch setup, for all its advantages mentioned above, requires a somewhat sophisticated software infrastructure, and a certain reduction in computational efficiency is introduced by the boundary communication. However, a multi-patch setup scales more favourably when compared to Cartesian mesh refinement when one considers increases in radial resolution or an increase of the computational domain. The computational cost of a multi-patch simulation scales as $O(N^2)$ and $O(N)$, respectively, in these two cases, while a mesh refinement simulation scales as $O(N^4)$ and $O(N^3)$, respectively. The difference comes from the fact that the radial resolution remains unchanged in these cases in a multi-patch setup, which is not possible in a Cartesian mesh-refinement simulation.
- The patch interface grid points from both sides must match, i.e., the grid topology introduces constraints on the possible choices of patch cell numbers. While this could be relaxed for the hydrodynamical scheme alone, the SBP/penalty techniques used in the generalized harmonic code for solving Einstein's equations [27] require these constraints, and we are ultimately interested in the full evolution system.

III. IMPLEMENTING A MULTI-PATCH SCHEME

In this section we describe how to implement a multi-patch scheme, either from scratch or by extending an existing unigrid code. Multi-patch techniques can be useful for any numerical code which evolves systems with a certain symmetry (for example, single stars or stars with accretion disks), mostly because the grid is adapted to the problem, and the angular and radial resolutions are decoupled. Because of these and other advantages, this section should be of interest to practitioners in the field of computational astrophysics, whether Newtonian or relativistic.

What amount of work can one expect, and what benefits can be gained in practical terms? We will answer these questions in comparison to the two main alternatives to multi-patch grids:

- In comparison to *spherical polar grids*, the multi-patch technique avoids the need to exclude the axis of symmetry and impose artificial boundary conditions there. This is important, in particular, for systems where outflows are generated and collimated on or near the axis. Also, the typical finite-difference or finite-volume Courant limitations to the time step near the poles of each surface $r = \text{const}$ are avoided. Three capabilities need

to be added: geometric terms which are not hard-coded into the equations, several grids instead of one, and a linear interpolation and transformation at the boundaries (see below). The finite volume scheme, Riemann solver, and local physics are unaffected.

- In comparison to *mesh refinement*, the multi-patch technique offers decoupled radial and angular resolution, which is of particular use far away from the central object (e.g. to extract gravitational radiation at large distances). However, mesh refinement is superior for processes which do not have explicit approximate symmetries, e.g. violent instabilities or binary mergers. A mesh refinement code can already handle several grids and boundary interpolation, so the only capabilities needed in addition are the coordinate transformations at the boundary and the geometric terms for the local coordinates. The finite volume scheme, Riemann solver, and local physics are again unaffected.

Given a certain finite volume/finite difference code, these steps need to be performed:

- The code needs to be able to handle several separate grids. These grids are all logically Cartesian and independent, so in many cases this might just be an additional loop statement. The local evolution scheme (e.g. flux calculation, finite differences, update) is logically unaffected.
- Each grid cell needs to store (or calculate during run-time) its location in global coordinates, typically Cartesian $x^i = (x, y, z)$, and in local coordinates, called $a^i = (a, b, c)$ below. In addition, the Jacobian $J^i_j = \partial x^i / \partial a^j$ and its inverse are needed; these can be obtained from the equations in Section IV.
- The patches need to use local coordinates in a certain range; say $[-1, +1]$ for the interior, with $a^i = \pm 1$ on the boundary, and a number of grid points extruding beyond the boundary for setting interpolation data. This depends on the stencil; for the monotonized central scheme used here, we use two ghost cells.
- Before each time update, and after the conversion from conserved to primitive variables, the boundary ghost cells need to be set by a (for example, trilinear) interpolation operation. For the patch systems presented in Section IV, it is known, for each patch, which other patch needs to be interpolated on,³ so a simple list of six entries for each patch

³ In general, this is not the case. Given the location of the ghost point

is enough to store the information. Given the target patch, we transform the location of the ghost point into *local coordinates on the target patch* using the equations in Section IV, and use traditional trilinear interpolation to get the data.

- Although boundary data has now been set, it is still expressed in the coordinate system of the target patch. Say, for relativistic hydrodynamics, we are interpolating boundary data for patch 4, and the data is coming from patch 5. After the interpolation, patch 4 has one side of its boundary stencil (the side adjoining to patch 5) filled with interpolated values for $(\rho, u, u_{(5)}^a, u_{(5)}^b, u_{(5)}^c)$. As indicated, the 3-velocity components are still expressed in the local coordinate system of patch 5. Therefore, a simple transition map $u_{(4)}^i = (\partial a_{(4)}^i / \partial x^j) (\partial x^j / \partial a_{(5)}^k) u_{(5)}^k$ is applied at each ghost cell, where $(\partial x^j / \partial a_{(5)}^k)$ is the Jacobian local \rightarrow global to transform $u_{(5)}^k$ to global coordinates, and $(\partial a_{(4)}^i / \partial x^j)$ is the Jacobian global \rightarrow local on patch 4.
- The numerical scheme needs to be able to work on a general background metric. For relativistic codes that is already contained in the covariant form, but a Newtonian code requires an addition of the geometrical terms.
- It is convenient to describe the initial data in terms of the global Cartesian coordinate system. That is, at each grid point all tensorial data can be represented in global coordinates (x, y, z) and afterwards transformed to the patch-local coordinate systems using the standard tensor transformation laws. In general relativistic hydrodynamics on fixed backgrounds, we need to transform the 3-velocity $u_{(G)}^i$ and the 4-metric $g_{\mu\nu}^{(G)}$ from global into local coordinates on patch m : $u_{(m)}^i = (\partial a_{(m)}^i / \partial x^j) u_{(G)}^j$, and $g_{0i}^{(m)} = (\partial x^j / \partial a_{(m)}^i) g_{0j}^{(G)}$, $g_{ij}^{(m)} = (\partial x^k / \partial a_{(m)}^i) (\partial x^l / \partial a_{(m)}^j) g_{kl}^{(G)}$.
- To parallelize the code, a simple domain decomposition technique can be used for each patch. In addition, it may be useful to have each patch distributed to as few processes as possible. For example, for a setup with N patches one might want to use $N \times M$ processes and assign each patch onto

M processes, in order to reduce communication overhead.⁴

IV. MULTI-PATCH SYSTEMS

In this section we specify the particular set of multi-patch setups used in this paper. A general introduction to the kind of systems useful for single stars or black holes can also be found in [29, 30].

For convenience and clarity, we will denote those 3-coordinates associated with the global coordinate system as $x^i, i = 1 \dots 3$ or (x, y, z) , and those associated with a particular local coordinate system as $a^i, i = 1 \dots 3$. If necessary, we indicate the particular patch p with a subscript, as in a_p^i .

A. The uni-patch system

We refer to a setup with one patch and a coordinate transformation with a constant diagonal Jacobian of the form

$$J_j^i \equiv \frac{\partial x^i}{\partial a^j} = c \delta_j^i \quad (10)$$

as a *uni-patch* system, where δ_j^i is the Kronecker symbol and $c \in \mathbb{R}$ a number. The associated transformation between local and global coordinates is the affine map

$$x^i = c a^i + b^i, i = 1 \dots 3 \quad (11)$$

and therefore we only need to specify the scale c and the 3-vector location of the origin b^i .

This patch system most closely matches the unigrid setups used in some three-dimensional simulations, and since it contains no patch interfaces it is used to test the performance of the code in the patch interior.⁵

B. The distorted two patches system

The *distorted two patches* system consists of two patches with coordinates

$$\text{Patch 0: } x^i = a_1^i, i = 1 \dots 3 \quad (12)$$

in global coordinates, a generic approach is then to transform the point to *all* local coordinate systems, and check whether it is in the appropriate interior range, say $[-1, +1]^3$. We perform this step once during initialization.

⁴ Our load distribution scheme is in fact more complex and can also efficiently handle patches of different sizes.

⁵ It is also easily possible to make this system toroidal by performing a topological identification between some boundaries, though we will not make use of this option in the examples below.

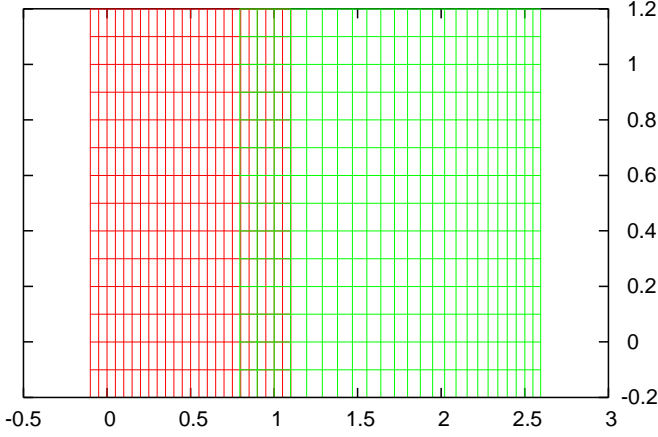


Figure 4: An illustration of the distorted two patches system. The left patch is constructed using an identity map between local and global coordinates, whereas the right patch follows from eqn. 13. The common interface is located at $x = 1$, which also coincides with the local coordinates $a_1^1 = 1$ and $a_2^1 = 1$. Note the ghost zones extending beyond the interface (cf. Fig. 3).

$$\text{Patch 1: } x^j = -\frac{1}{2}(a_2^j + 1)^2 + 3, j \in \mathbb{R} \quad (13)$$

$$x^i = a_i^j, i = 1 \dots 3, i \neq j.$$

The internal coordinate range of each patch is chosen to be $(0, 1)^3$, and the common interface is then located at $x^j = a_1^j = a_2^j = 1$. The second patch is quadratically distorted and matches the first one at the *upper* boundary of coordinate j . An example with $j = 1$ is illustrated in Fig. 4.

C. The cubed-sphere six patches system

Whereas the two multi-patch systems discussed above are useful mostly for purposes of code testing, the *cubed-sphere six patches* system [29] is a setup which can be directly used to efficiently model black holes and accretion disks around black holes. The system covers a spherical region with a central sphere cut out (*excised*). The central excised sphere can be used to exclude part of the interior of a black hole, since it is causally disconnected from the exterior region. In a fully relativistic setting, a symmetric hyperbolic formulation of Einstein's equations with only physical speeds of propagation admits to use the interior spherical surface as an outflow boundary, since the *continuum* property of causal disconnection is then also represented on the *discrete* level. We will make use of this fact in the coupled evolutions in a later paper in this series.

The domain is covered by a family of spheres with different “radial” coordinates $r = (\sum_i (x^i)^2)^{1/2}$. A simple approach would be to cover almost the entire sphere by spherical polar coordinates, which, unfortunately, also

introduces coordinate singularities. Therefore, either one opts to cover the sphere using at least to patches with properly rotated spherical polar coordinate systems (*yin-yang grid* [57]), or one uses a patch system which does not derive from spherical polar coordinates. We will use the latter approach here.

The *cubed sphere* patch system on a sphere is conceptually constructed by imagining a cube con-central with the sphere, introducing the canonical Cartesian restrictions on each surface of the cube, and projecting these coordinate lines onto the sphere. When mapping the radial coordinate via the identity, this produces six patches with coordinate transformations [29]

$$\begin{aligned} \text{Patch 0: } x &= \frac{G_0}{E_0}, y = \frac{G_0 a_0^2}{E_0}, z = \frac{G_0 a_0^1}{E_0} \\ \text{Patch 1: } x &= \frac{-G_1 a_1^2}{E_1}, y = \frac{G_1}{E_1}, z = \frac{G_1 a_1^1}{E_1} \\ \text{Patch 2: } x &= \frac{-G_2}{E_2}, y = \frac{-G_2 a_2^2}{E_2}, z = \frac{G_2 a_2^1}{E_2} \\ \text{Patch 3: } x &= \frac{G_3 a_3^2}{E_3}, y = \frac{-G_3}{E_3}, z = \frac{G_3 a_3^1}{E_3} \\ \text{Patch 4: } x &= \frac{-G_4 a_4^1}{E_4}, y = \frac{G_4 a_4^2}{E_4}, z = \frac{G_4}{E_4} \\ \text{Patch 5: } x &= \frac{G_5 a_5^1}{E_5}, y = \frac{G_5 a_5^2}{E_5}, z = \frac{-G_5}{E_5} \end{aligned} \quad (14)$$

$$E_i \equiv 1 + (a_i^1)^2 + (a_i^2)^2$$

$$G_i \equiv \frac{1}{2}[r_0(1 - a_i^3) + r_1(1 + a_i^3)]$$

$$a_i^j \in [-1, 1]$$

Here, r_0 and r_1 are free parameters which determine the location of the two outer boundary spheres in terms of global coordinates. Patches 0 to 3 are constructed in the equatorial plane counterclockwise starting from the positive x axis, and patches 4 and 5 intersect with the z axis. An illustration of this patch system in the equatorial plane $z = 0$ is given in Fig. 5.

D. The cubed-sphere seven patches system

While the six patches system is well-suited to excise the interior of black holes from the computational domain, there are also situations (for example, stellar simulations) where we would like to cover the entire interior of a sphere with regular coordinates. One approach to achieve this is to cover the central part of the computational domain with Cartesian coordinates (i.e., a uni-patch as described above), describe the outer boundary as a sphere, and decompose the region between the surface of the Cartesian patch and the outer boundary into six additional patches [30]. Then, on the outer boundary,

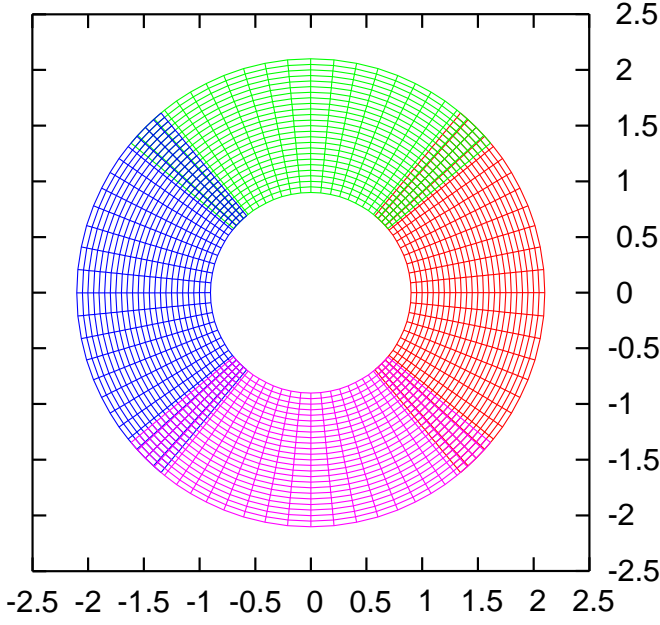


Figure 5: An illustration of the cubed-sphere six patches system. The diagram shows a cut of grid lines with the equatorial plane $z = 0$. In the notation of eqn. 15, patch 0 is to the right (in red), counterclockwise continuing with patch 1 (in green), patch 2 (in blue) and patch 3 (in violet). The inner spherical boundary can be used as an *excision* surface for modeling black holes.

spherical coordinates are fixed as in the six patches system, whereas the intermediate surfaces of constant coordinate a^3 (which are spheres in the six patches system) are deformed in a way to interpolate between a spherical section and a flat face of the central cube.

In detail, the transformations between local and global coordinates which specify the cubed-sphere seven patches system are as follows (please note that we introduce the central cube as patch 6 to retain the counting convention used in the six patches system):

$$\begin{aligned}
 \text{Patch 0: } x &= \frac{G_0}{F_0}, y = \frac{G_0 a_0^2}{F_0}, z = \frac{G_0 a_0^1}{F_0} \\
 \text{Patch 1: } x &= \frac{-G_1 a_1^2}{F_1}, y = \frac{G_1}{F_1}, z = \frac{G_1 a_1^1}{F_1} \\
 \text{Patch 2: } x &= \frac{-G_2}{F_2}, y = \frac{-G_2 a_2^2}{F_2}, z = \frac{G_2 a_2^1}{F_2} \\
 \text{Patch 3: } x &= \frac{G_3 a_3^2}{F_3}, y = \frac{-G_3}{F_3}, z = \frac{G_3 a_3^1}{F_3} \\
 \text{Patch 4: } x &= \frac{-G_4 a_4^1}{F_4}, y = \frac{G_4 a_4^2}{F_4}, z = \frac{G_4}{F_4} \\
 \text{Patch 5: } x &= \frac{G_5 a_5^1}{F_5}, y = \frac{G_5 a_5^2}{F_5}, z = \frac{-G_5}{F_5} \\
 \text{Patch 6: } x &= r_0 a_6^1, y = r_0 a_6^2, z = r_0 a_6^3
 \end{aligned} \tag{15}$$

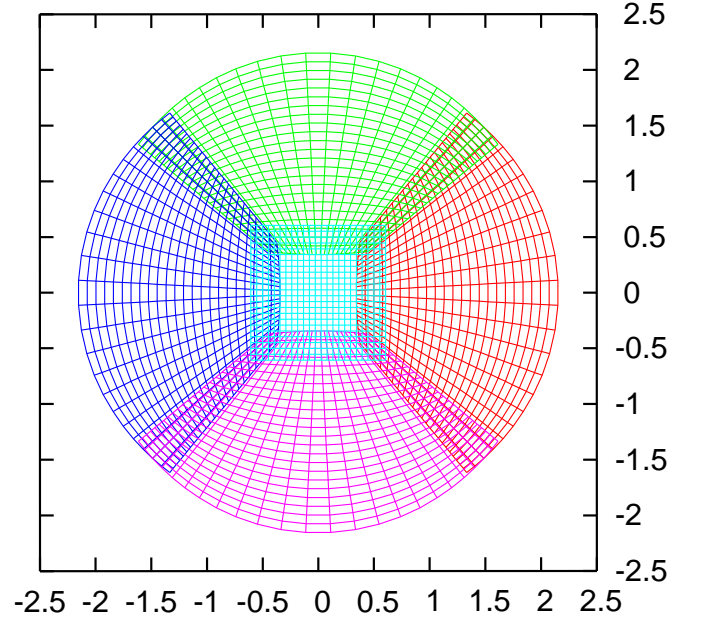


Figure 6: An illustration of the cubed-sphere seven patches system. The diagram shows a cut of grid lines with the equatorial plane $z = 0$. In the notation of eqn. 16, patch 0 is to the right (in red), counterclockwise continuing with patch 1 (in green), patch 2 (in blue) and patch 3 (in violet). The central, Cartesian patch is patch 6 (light blue). The parameters for the patch system were chosen to be $r_0 = 0.5$ and $r_1 = 2$, so the outer boundary is a sphere of radius 2 in this diagram. The outermost two grid lines, however, are ghost zones required for the MC reconstruction algorithm (see Section II B 3).

$$\begin{aligned}
 F_i &\equiv \left(\frac{(r_1 - G_i) + (G_i - r_0)E_i}{r_1 - r_0} \right)^{1/2} \\
 E_i &\equiv 1 + (a_i^1)^2 + (a_i^2)^2 \\
 G_i &\equiv \frac{1}{2}[r_0(1 - a_i^3) + r_1(1 + a_i^3)] \\
 a_i^j &\in [-1, 1]
 \end{aligned}$$

Here, r_0 and r_1 are free parameters which determine, in terms of global coordinates, the location of the outer boundary (r_1) and the extent of the central cube (r_0). An illustration of the cubed-sphere seven patch system in the equatorial plane $z = 0$ is given in Fig. 6.

E. The cubed-sphere thirteen patches system

A (minor) disadvantage of the cubed-sphere seven patches system is that, beyond the Cartesian patch around the origin in global coordinates, all surfaces of constant a_i^3 for $i = 0 \dots 5$ up to the outer boundary at $a_i^3 = 1$ are not spherical, but rather “intermediate” between spheres and cube surfaces. To cover a star with spherical surfaces, a different approach is to combine the advantages of the six and seven patches systems by covering a large region of the star with a six patches system,

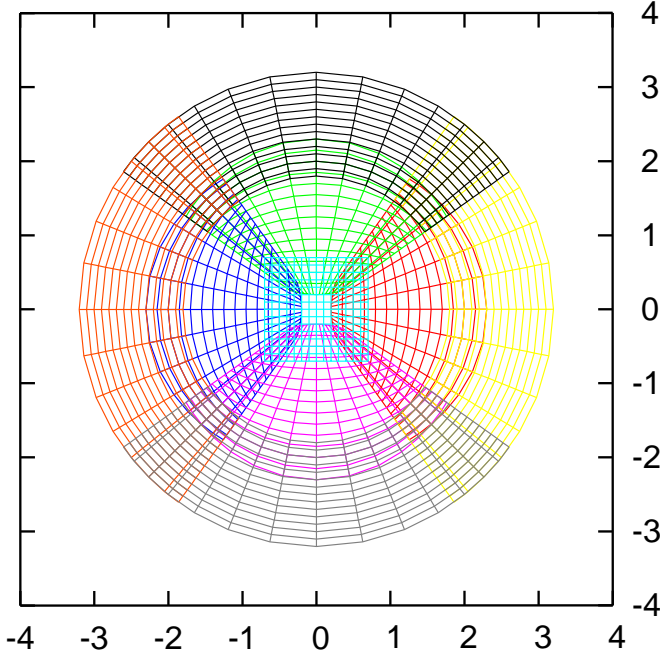


Figure 7: An illustration of the cubed-sphere thirteen patches system. This system consist of an outer six patches system, where all surfaces of constant coordinate a^3 are spheres, and an inner seven patches system matched to the inner boundary of the outer system to cover the origin in a regular manner.

and making the origin regular by introducing a seven patches system in a way that the parameters r_1 of the seven patches system and r_0 of the six patches system match. This construction can also be used, for example, to extract gravitational waves without needing to interpolate to spheres in order to decompose the solution into its different multipole components. A cut of the coordinate lines with the equatorial plane $z = 0$ is illustrated in Fig. 7

V. RESULTS

A. Shock tubes on the uni-patch system

To investigate the code's ability to evolve special relativistic Riemann problems without the additional complications of non-trivial Jacobians and inter-patch interfaces, we perform three kinds of standard tests specified by the fluid states

$$\text{Sod test: } \rho_L = 1, u_L = 1.5, u_L^i = 0 \quad (16)$$

$$\rho_R = 0.125, u_R = 0.15, u_R^i = 0$$

$$\text{"Simple" test: } \rho_L = 10, u_L = 20, u_L^i = 0 \quad (17)$$

$$\rho_R = 1, u_R = 10^{-6}, u_R^i = 0$$

$$\text{Blast wave test: } \rho_L = 1, u_L = 1.5 \cdot 10^3, u_L^i = 0 \quad (18)$$

$$\rho_R = 1, u_R = 1.5 \cdot 10^{-2}, u_R^i = 0$$

In all cases we assume an equation of state of the form $P = (\Gamma - 1)u$ with $\Gamma = 4/3$, and the metric has the Cartesian Minkowski form $g_{\mu\nu} = \eta_{\mu\nu}$. The Sod test, in particular, will also be used for shock tube experiments on several patches below.

For the uni-patch system, we choose the free parameter c in eqn. 10 to be unity, and we restrict attention to a one-dimensional problem where the initial contact surface is located at $x = 0.5$. The grids, therefore, are only specified by the number of cells in the x direction, where we choose values from 50 to 800.

To generate a measure of error, we calculate the exact solution to the the special relativistic problems using the `riemann` code [58]. The error function for each primitive variable is simply taken to be a norm over the difference function between the exact solution and the discrete result. The boundaries are set to the exact solution produced with `riemann`.

The results of evolution of the Sod, "Simple" and blast wave problem are illustrated in Figs. 8, 9 and 10. The code has some inherent dissipation due to the use of the MC limiter, but it converges to the exact solution.

B. Sod test on the distorted two patches system

We have performed the Sod test (eqn. 16) on the two distorted two patches system described in Section IV B. The first and second patch both have $n \times 1 \times 1$ cells, where $n = 50 \dots 800$, and the patch interface is located at $x = 1$ in terms of global coordinates. As before, the boundary ghost cells are set to the exact solution.

Fig. 11 demonstrates that the multi-patch approximation leads to a convergent transmission of the shock across the interface, although both patches use different local tensor bases, and a non-trivial Jacobian is associated with the second patch.

C. Sod test on the cubed sphere six patches system

To perform the Sod test (eqn. 16) on the cubed sphere six patches system described in Section IV C, we prepared a setup with $n \times n \times n$ cells per patch, where $n = 20, 40, 80$, and chose the free parameters r_0 and r_1 , which specify the inner and outer boundary radius in terms of the global coordinates, to be given by $r_0 = 1$ and $r_1 = 2$. The outer boundaries of the domain are set to the exact solution produced by `riemann`.

The evolution of the density for the case $n = 80$ is shown in Fig. 12. This plot shows the density function in the equatorial plane, and the motion of the waves resulting from the Riemann problem. Global convergence to the exact solution, which is again constructed using `riemann` and then transformed to the local coordinate systems as required, is demonstrated in Fig. 13.

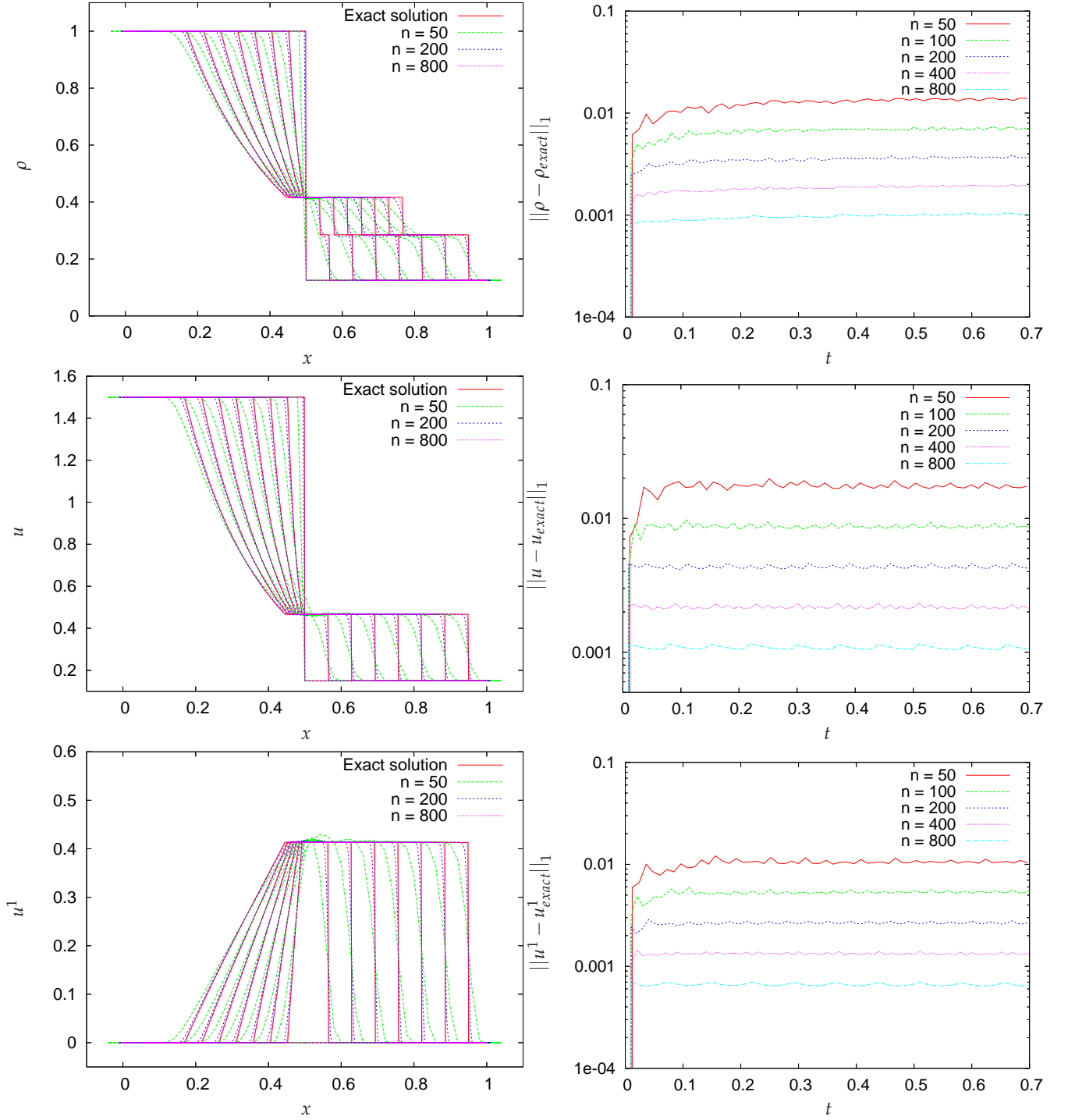


Figure 8: Sod test on the uni-patch system. For each of the primitive variables ρ , u , and u^1 , the left plots show the comparison of the numerical result with the exact solution for different times (spaced in units of $\Delta t = 0.1$), and right plots show the error function (l_1 norm of the difference between the exact solution and the discrete result) over time.

As explained in Section II B 3, the boundaries are treated by setting boundary ghost zones using an interpolation operation before each time update, followed by a suitable coordinate transformation. Fig. 14 shows a pseudo-Schlieren plot of the transmission of the shock front across an interface. The code produces a number

of minor reflections as expected, but the overall shape of the shock front remains intact.

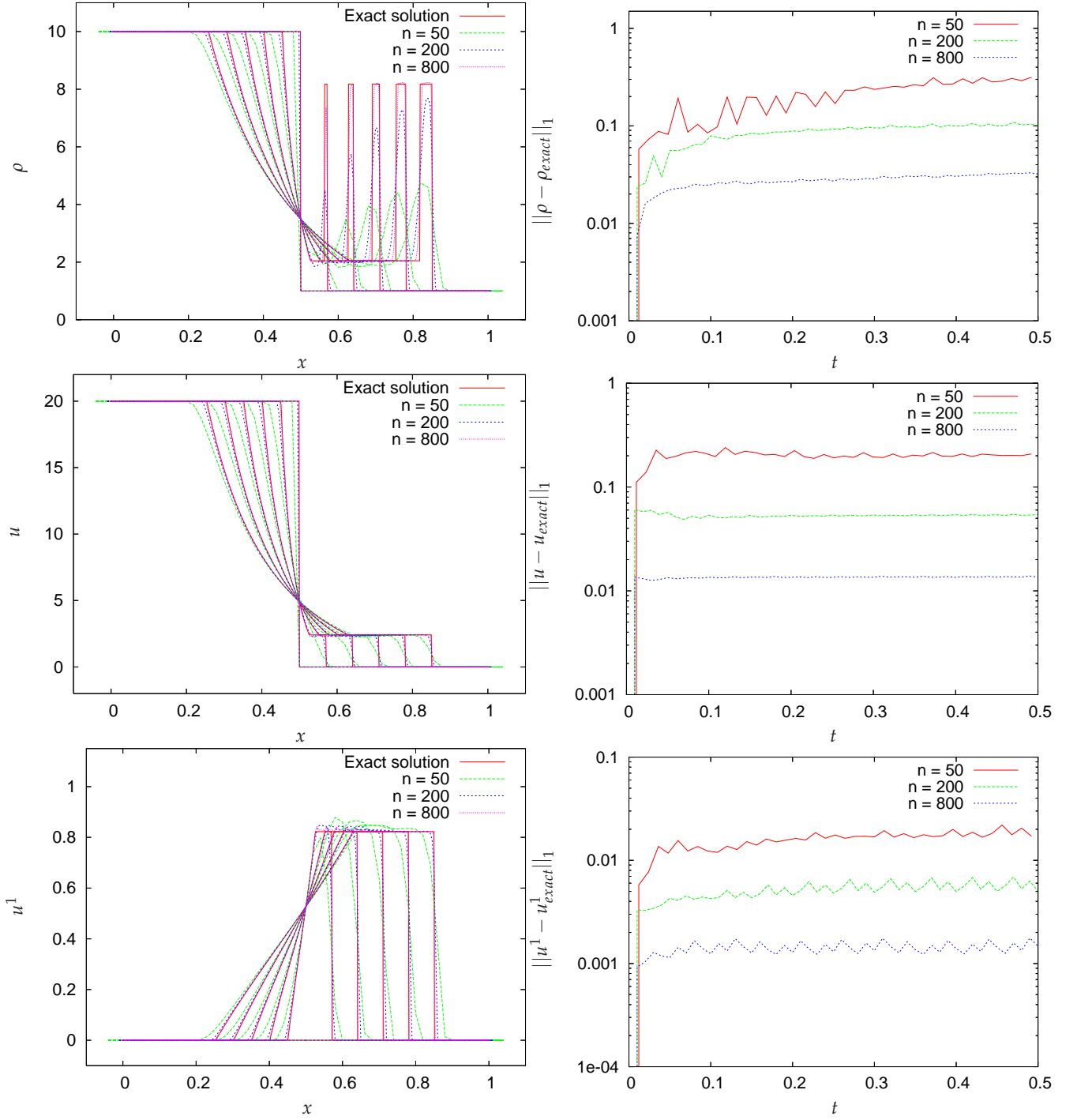


Figure 9: “Simple” shock test on the uni-patch system. Same as in Fig. 8.

D. Sod test on the cubed sphere seven patches system

The setup of this test is very similar to Section V C, but uses the cubed sphere seven patches system described in Section IV D. The free parameters r_0 and r_1 , which specify the extent of the central cube and the location of the outer spherical boundary, are set to $r_0 = 0.5$ and $r_1 = 2$. Again, we use $n \times n \times n$ cells, with $n = 20, 40, 80$,

on each patch, and set the outer boundary ghost zones to the exact solution.

As Fig. 15 shows, the evolution proceeds very similar to the six patches system and is convergent to the exact solution.

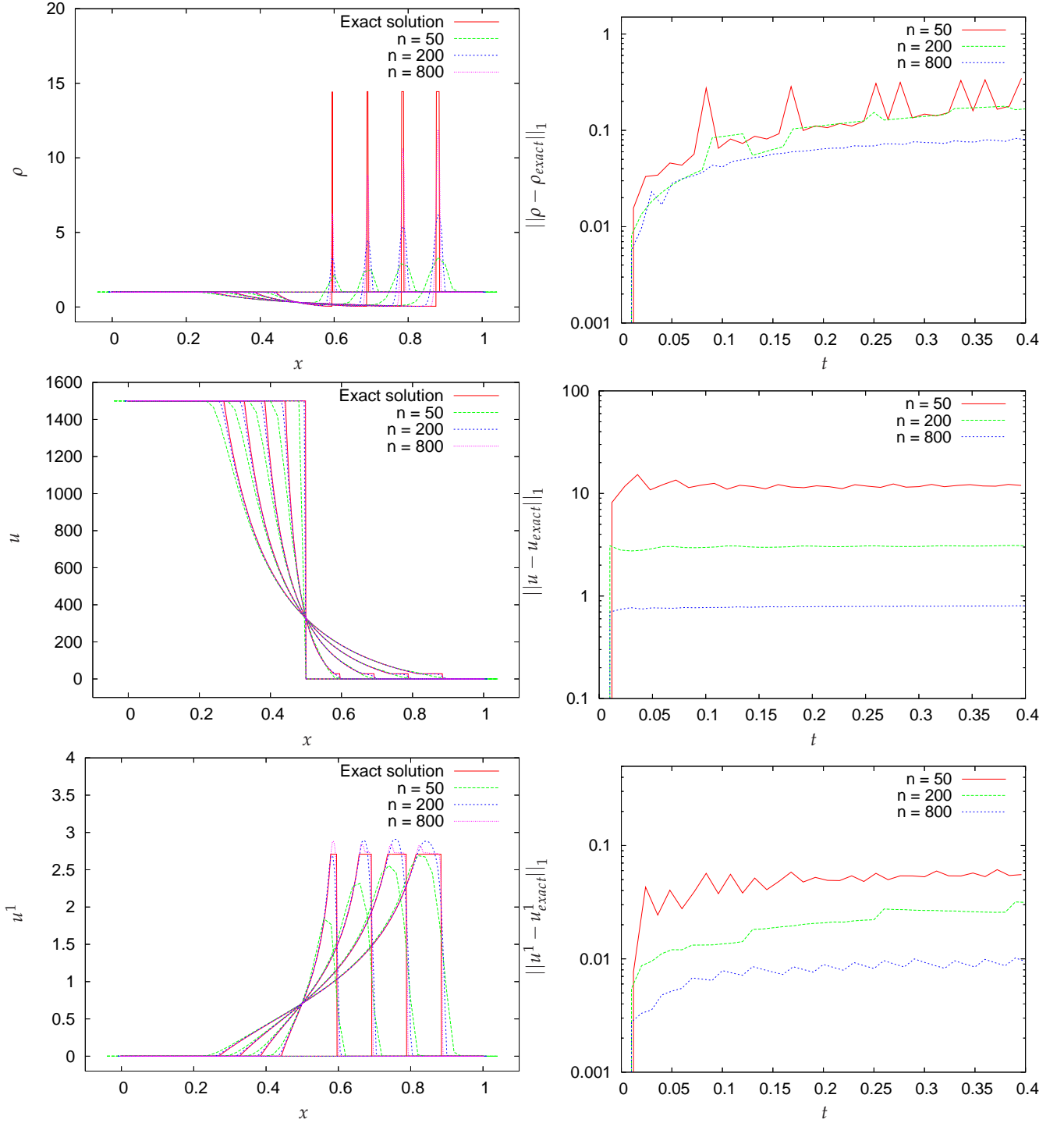


Figure 10: Blast wave test on the uni-patch system. Same as in Fig. 8.

E. Sod test on the cubed sphere thirteen patches system

In the same way as in Sections VC and VD, we perform the Sod test on the cubed sphere thirteen patches system described in Section IV E. Since this setup consists of seven inner patches matched to six outer ones, we have three free parameters available. For purposes

of this particular test, we use $r_0 = 0.5$, $r_1 = 2$ and $r_2 = 3$. The results are presented in Fig. 16, and show convergence to the exact solution.

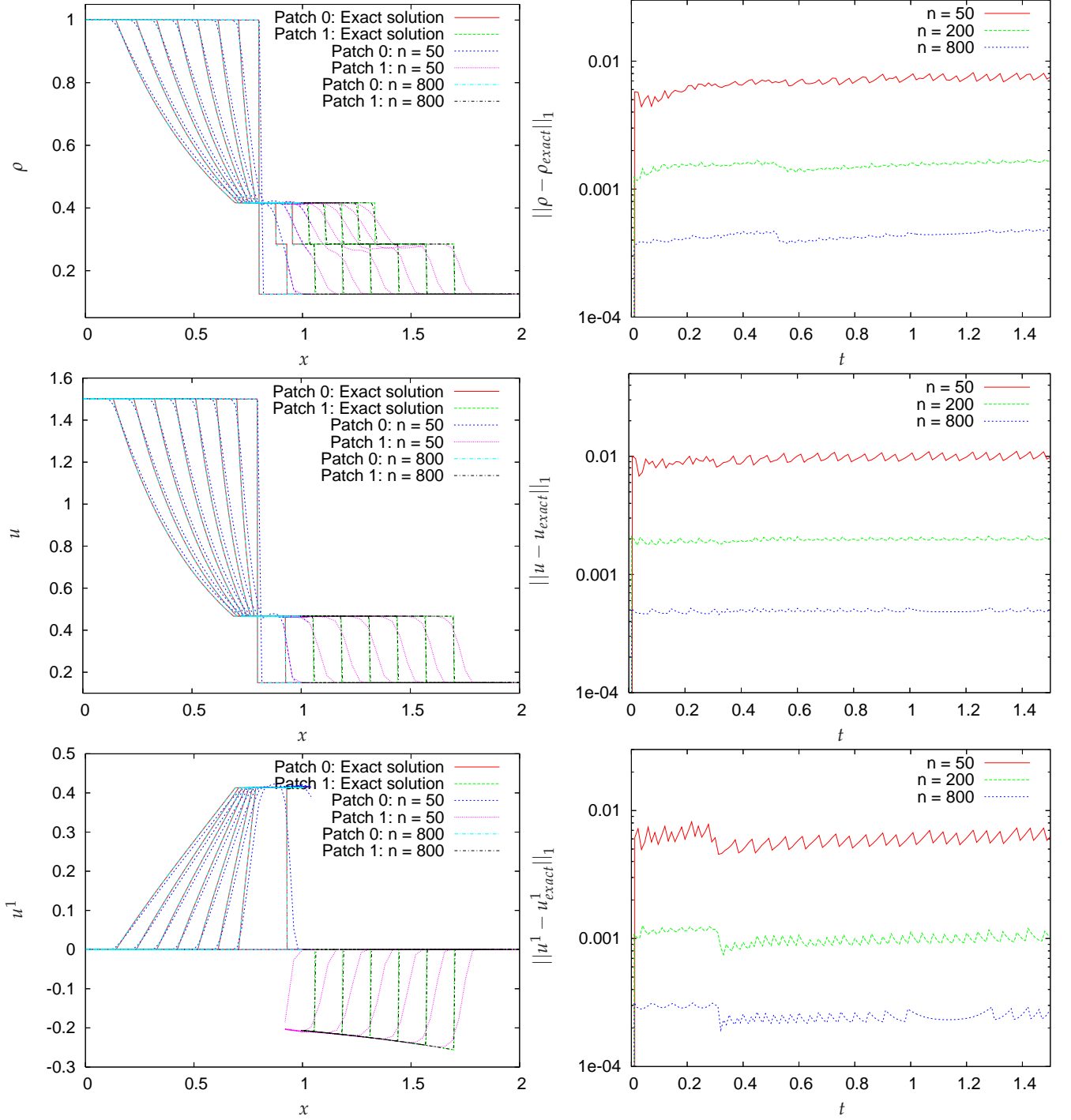


Figure 11: Sod test on the distorted two patches system. This system, defined in Section IV B, consists of a left patch (*patch 0*) which is undistorted, and a right patch (*patch 1*) which has a non-trivial Jacobian associated with the transformation from local to global coordinates. The left panel shows the primitive variables ρ , u and u^1 in dependence on the global coordinate x , for different coordinate times ($\Delta t = 0.2$). The interface between the patches is located at $x = 1$. In the lower left diagram, note how the velocity component u^1 appears discontinuous across the interface since it is represented in different tensor bases on each patch. The right panel shows global convergence with respect to the l_1 norm. (The left panel does not contain the graphs for $n = 200$ to enhance the visual clarity of the plots.)

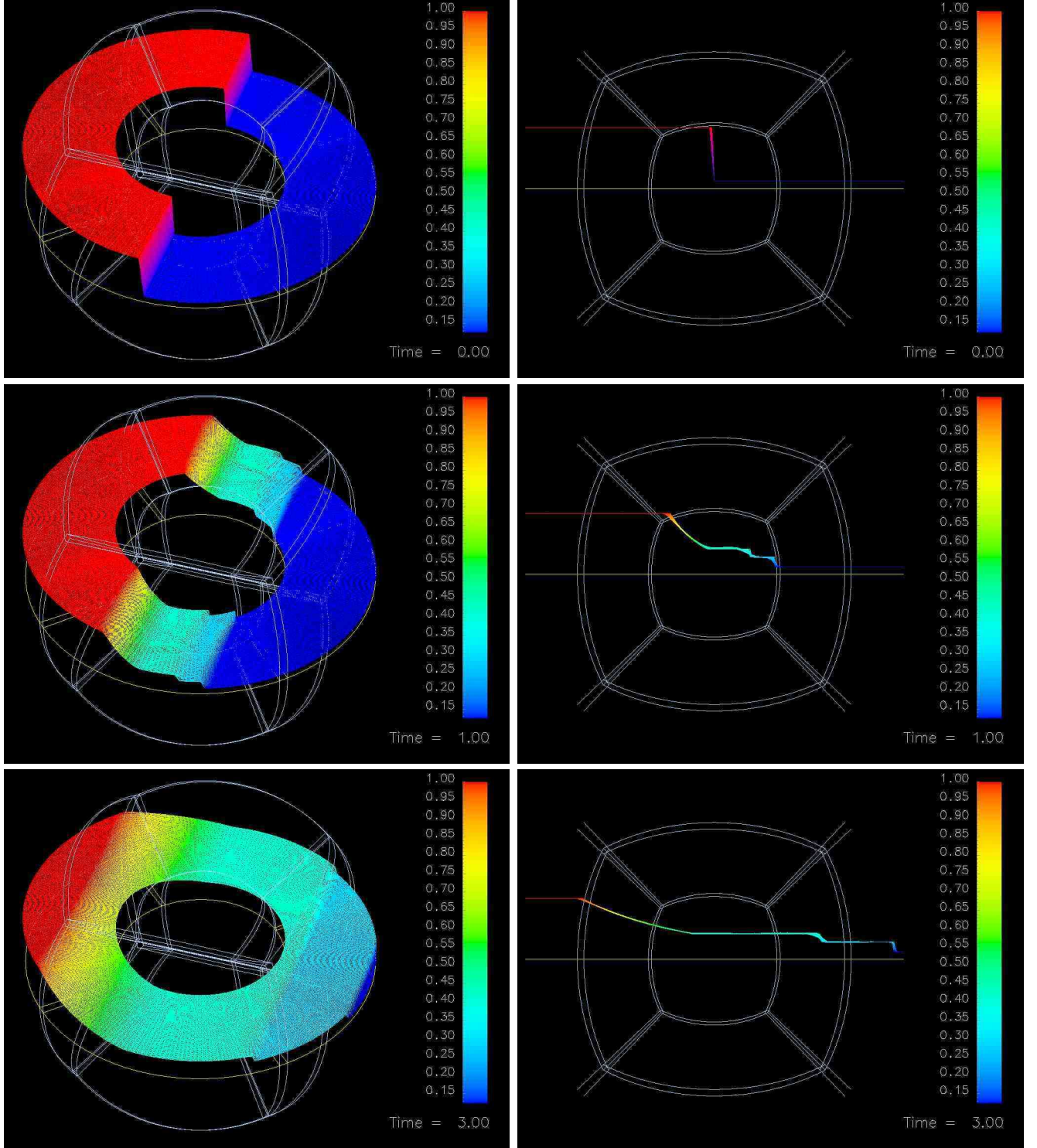


Figure 12: Sod test on the six patches system: evolution of the density for the case where each of the patches has resolution $80 \times 80 \times 80$. The white lines indicate the boundaries of the six patches which are used to cover the computational domain. The yellow surface is the cut of the plane $z = 0$ with the grid boundaries, and the density function is shown at $z = 0$. The left panel shows a perspective view of the evolution for coordinates times 0, 1 and 3, whereas the right panel shows an orthogonal projection from the negative y axis at the same times.

Polytropic scale	K	100
Polytropic index	Γ	2
Central rest-mass density	ρ_c	10^{-3}
Coordinate axis ratio	r_p/r_e	0.7
ADM mass	M	1.4906
Rest mass	M_0	1.5936
Equatorial proper radius	R_e	12.322
Equatorial inverse compactness	R_e/M	7.7321
Angular momentum	J	1.3192
Normalized angular momentum	J/M^2	0.5938
Kinetic over binding energy	$T/ W $	$7.4792 \cdot 10^{-2}$
(see caption)	Ω/Ω_K	0.7536

Table I: Parameters and integral quantities of the uniformly rotating polytrope used as a code test for the cubed sphere thirteen patches system. The quantities K , Γ , ρ_c , and r_p/r_e are parameters. The quantity Ω is the angular velocity, while Ω_K is the associated Keplerian velocity. Therefore, the mass-shedding sequence is located at $\Omega_e/\Omega_K = 1$ (cf. also [59]).

F. Uniformly rotating polytrope on the cubed sphere thirteen patches system

An important equilibrium solution for general relativistic astrophysics are rotating polytropes, since they provide an approximate model for relativistic stars. They are also particular well-suited to serve as code tests, since they involve a non-trivial spacetime geometry even in adapted coordinates [59], and their stability properties are well investigated in the case of stiff (polytropic index $\Gamma = 2$) and uniformly rotating solutions.

We constructed a uniformly rotating polytropic solution with the *rns* code [60], assuming a stratification $P = K\rho^\Gamma$ with $K = 100$ and $\Gamma = 2$. The central density is fixed to $\rho_c = 10^{-3}$, and the ratio of polar to equatorial radii (in terms of the particular gauge choice in *rns* [59]) is set to $r_p/r_e = 0.7$, which produces a rapidly rotating model (see Table I). The solution, which is represented on a two-dimensional grid in *rns*, is then mapped to every patch, and we apply the appropriate tensor coordinate transformations on all quantities (the four-metric, its derivatives, and the primitive hydrodynamical variables).

To specify a multi-patch system, we need to choose resolutions and set the free parameters r_0, r_1, r_2 (see Section IV E) which correspond to the location of the cube boundary, the spherical boundary between the seven patches and the six patches system, and the spherical outer boundary. We use $r_0 = 1$, $r_1 = 4$ and $r_2 = 14$ for these locations in the test case presented here, but we have also experimented with different values and obtained very similar results. The number of cells per patch are set to n^3 , where $n = 20, 40, 80$. During evolution, we do *not* enforce the polytropic constraint $P = K\rho^\Gamma$, but rather use the gamma law $P = (\Gamma - 1)u$.

Quantity	$20 \times 20 \times 20$	$40 \times 40 \times 40$	$80 \times 80 \times 80$
Central density	0.1%	0.04%	0.013%
Rest mass	0.45%	0.16%	0.054%
Angular momentum	1.07%	0.38%	0.13%

Table II: Rotating neutron star on the cubed sphere system of thirteen patches. This table shows average errors acquired per dynamical time in different resolutions.

Fig. 17 shows the initial data mapped to the thirteen patches system. Note that the density outside the star is not exactly zero, but set to the atmosphere value (see Section II B 2) since our techniques are unable to handle vacuum-matter interfaces. Fig. 18 shows an evolution of the polytrope for about 10 dynamical times⁶. Note that we use the *conserved* variables to show convergence: the three-velocity error is dominated by low-density, but fast material leaving the stellar surface, which does not carry a high amount of momentum, however. This reflects the approximation we make with using an artificial atmosphere in the first place. Therefore, convergence tests in the u^i are dominated by noise.

The evolution of the star exhibits artifact oscillations and a linear drift in the central density, but converges to the stationary exact solution. The average errors acquired by the star per dynamical time are listed in Table II.

G. Equilibrium accretion torus on a Schwarzschild background

This test models a thick accretion disk in the test-fluid limit around a black hole. The equilibrium structure of these solutions has been discussed in several seminal papers by Fishbone and Moncrief [61] and Abramowicz, Jaroszyński, Sikora and Kozłowski [62, 63, 64]. For the purposes of this test, we use a polytropic disk with constant specific angular momentum on a Schwarzschild background.

The details of how to construct these models can be found in the aforementioned publications, we will only give a quick overview here: Starting with a spacetime with a line element of the form

$$ds^2 = g_{tt}dt^2 + 2g_{t\phi}dtd\phi + g_{rr}dr^2 + g_{\theta\theta}d\theta^2 + g_{\phi\phi}d\phi^2, \quad (19)$$

we seek stationary solutions for a rotating fluid given by the energy-momentum tensor eqn. 1. For the four velocity of the fluid, we assume the form

⁶ We make use of the common choice $t_D = R_e\sqrt{R_e/M}$ as a measure of dynamical time, where R_e is the proper equatorial circumferential radius and M the ADM mass of the star.

$$u^\mu = (u^t, 0, 0, u^\phi)^T \quad (20)$$

and define the angular velocity $\Omega = u^\phi/u^t$ and specific angular momentum $l = -u_\phi/u_t$. From the normalization $u^\mu u_\mu = -1$ we obtain

$$(u_t)^{-2} = g^{tt} - 2g^{t\phi}l + g^{\phi\phi}l^2. \quad (21)$$

From this quantity, and using $l = \text{const}$ and the polytropic relation $P = K\rho^\Gamma$, we can calculate the specific internal energy $\epsilon \equiv u/\rho$ and the density [63]:

$$\begin{aligned} \epsilon &= \left(\frac{(u_t)_0}{u_t} - 1 \right) / \Gamma \\ \rho &= \left(\frac{\Gamma - 1}{K\epsilon} \right)^{\frac{1}{\Gamma-1}} \end{aligned} \quad (22)$$

Here, $(u_t)_0$ is a free parameter of the solution. The particular parameter values we use to construct the torus are $K = 10^{-2}$, $\Gamma = 4/3$, $l = 4.5$, and $(u_t)_0 = -0.98$. The atmospheric density is set to 10^{-12} (compared to a maximal density of $2 \cdot 10^{-2}$). In addition, we apply a radial coordinate transformation of the form $r = \exp(\bar{r})$, i.e., a logarithmic grid, to resolve the region close to the black hole better. The inner and outer boundaries are treated by extrapolation onto ghost zones as mentioned in Section II B 3.

We use a cubed sphere six patches system, with $15 \times 15 \times 40$, $30 \times 30 \times 80$ and $60 \times 60 \times 160$ cells per patch (remember that our convention in eqn. 15 implies that the third local coordinate corresponds to the location of each sphere in the global coordinate space, whereas the first two coordinates roughly correspond to the angular directions on the sphere). The free parameters specifying the boundary location are set to $r_0 = 6$ and $r_1 = 50$. This places the inner radius outside of the horizon, which is necessary since there is a coordinate singularity at the horizon in these coordinates. This is possible since the background is not evolved and since there is no matter near the horizon.

The evolution of the torus for about ten rotational times (in terms of the radius of maximal density) is displayed in Fig. 20. The lowest resolution case, $n = 15 \times 15 \times 40$ per patch, is clearly under-resolved. The error ratio between the higher resolutions is about 0.44, which is only slightly better than first order convergence (Table III collects the errors for different resolutions). The numerical techniques we are using are “at best” second-order accurate, but drop to first order near maxima and at the boundary to the atmosphere. In cases where internal turbulence is present, as is expected in realistic accretion disks [53, 65], it will likely dominate the solution errors.

Quantity	$15 \times 15 \times 40$	$30 \times 30 \times 80$	$60 \times 60 \times 160$
Central density	$\approx 4.6\%$	0.055%	0.021%
Rest mass	$\approx 0.1\%$	0.32%	0.14%
Angular momentum	$\approx 0.1\%$	0.32%	0.14%

Table III: Equilibrium accretion torus on a Schwarzschild background, using the cubed sphere six patches system. This table shows average errors acquired per rotational time in different resolutions. Note that the lowest resolution case ($n = 15 \times 15 \times 40$ is under-resolved).

VI. SUMMARY

The purpose of this paper was to demonstrate that multi-patch techniques can be employed to use quasi-spherical grids for applications in general relativistic astrophysics. In addition to a standard set of techniques for modeling relativistic flows, we use overlapping grid patches with boundary ghost zones, which are synchronized to data obtained from an interpolation operation, and a transformation associated with the transition map.

The resulting scheme is able to cleanly transport shock fronts across patch interfaces and evolve accretion disks with all the advantages of a grid based on spherical polar coordinates, but without sharing its disadvantages like artificial outer boundaries due to coordinate singularities, or small time steps imposed by the convergence of the latitudinal great circles towards the poles. In addition, spherical grids admit to choose radial and angular resolutions independently, which is mirrored in our approach, and makes it possible to use radial grids with sufficient radial and angular resolution near the black hole horizon.

The main focus of this approach will be general relativistic single star and star + accretion disk models, but there is no fundamental reason why they could not be employed in binary models as well. In particular, the inspiral phase could potentially be treated with a reduced solution error compared to Cartesian mesh refinement setups (see, for example, [66]). Another possible application is to model a source with Cartesian mesh refinement in the domain center, but use a cubed sphere seven or thirteen patches system to provide outer boundary conditions and propagate gravitational radiation. Clearly, many of these scenarios require to solve Einstein’s equations coupled to relativistic hydrodynamics or even magnetohydrodynamics, which will be a topic of this series in the future.

Acknowledgments

We greatly enjoyed the inspiring discussions we had with M. Anderson, P. Diener, H. Dimmelmeier, N. Dorband, J. Frank, I. Hawke, K. Kifonidis, L. Lehner, M.

Megevand, E. Müller, P. Motl, M. Obergaulinger, I. Olabarrieta, C. D. Ott, C. Palenzuela-Luque, J. Pullin, E. Seidel, N. Stergioulas, and J. Tohline. This research was supported in part by the National Science Foundation through grant PHY 0505761 to Louisiana State University and Teragrid allocation TG-MCA02N014. This research also employed resources of the Center for Com-

putation & Technology at LSU, which is supported by funding from the Louisiana legislature's Information Technology Initiative, and of the Louisiana Optical Network Initiative. We used the computational resources of Abe and Tungsten at the NCSA, and the workstation machines at CCT.

-
- [1] B. Zink, N. Stergioulas, I. Hawke, C. D. Ott, E. Schnetter, and E. Müller, *Phys. Rev. Lett.* **96**, 161101 (2006).
 - [2] B. Zink, N. Stergioulas, I. Hawke, C. D. Ott, E. Schnetter, and E. Müller, *Phys. Rev. D* **76**, 024019 (2007).
 - [3] C. D. Ott, H. Dimmelmeier, A. Marek, H.-T. Janka, I. Hawke, B. Zink, and E. Schnetter, *Phys. Rev. Lett.* **98**, 261101 (2007).
 - [4] M. Shibata, Y. Sekiguchi, and R. Takahashi, *Prog. Theo. Phys.* **118**, 257 (2007).
 - [5] M. Shibata and K. Uryu, *Phys. Rev. D* **74**, 121503 (R) (2006).
 - [6] B. C. Stephens, M. D. Duez, Y. T. Liu, S. L. Shapiro, and M. Shibata, *Classical and Quantum Gravity* **24**, 207 (2007).
 - [7] M. D. Duez, Y. T. Liu, S. L. Shapiro, M. Shibata, and B. C. Stephens, *Phys. Rev. Lett.* **96**, 031101 (pages 4) (2006).
 - [8] H. Dimmelmeier, C. D. Ott, H.-T. Janka, A. Marek, and E. Müller, *Phys. Rev. Lett.* **98**, 251101 (2007).
 - [9] L. Baiotti, R. D. Pietri, G. M. Manca, and L. Rezzolla, *Phys. Rev. D* **75**, 044023 (2006).
 - [10] L. Baiotti, I. Hawke, and L. Rezzolla, *Class. Quantum Grav.* **24** *in press* (2007).
 - [11] P. J. Montero, O. Zanotti, J. A. Font, and L. Rezzolla, *Mon. Not. R. Astron. Soc.* **378**, 1101 (2007).
 - [12] M. Anderson, E. W. Hirschmann, L. Lehner, S. L. Liebling, P. M. Motl, D. Neilsen, C. Palenzuela, and J. E. Tohline (2007), [arXiv:0708.2720 \[gr-q\]](https://arxiv.org/abs/0708.2720).
 - [13] R. Oechslin and H.-T. Janka, *Phys. Rev. Lett.* **99**, 121102 (2007).
 - [14] J. Thornburg, *Class. Quantum Grav.* **4**, 1119 (1987).
 - [15] J. Thornburg, Ph.D. thesis, University of British Columbia, Vancouver, British Columbia (1993).
 - [16] R. Gómez, L. Lehner, P. Papadopoulos, and J. Winicour, *Class. Quantum Grav.* **14**, 977 (1997).
 - [17] R. Gómez, L. Lehner, R. Marsa, J. Winicour, A. M. Abrahams, A. Anderson, P. Anninos, T. W. Baumgarte, N. T. Bishop, S. R. Brandt, et al., *Phys. Rev. Lett.* **80**, 3915 (1998).
 - [18] S. Bonazzola, E. Gourgoulhon, and J.-A. Marck, *J. Comp. Appl. Math.* **109**, 433 (1999).
 - [19] L. E. Kidder, M. A. Scheel, and S. A. Teukolsky, *Phys. Rev. D* **64**, 064017 (2001).
 - [20] E. Gourgoulhon, P. Grandclément, K. Taniguchi, J.-A. Marck, and S. Bonazzola, *Phys. Rev. D* **63**, 064029 (2001).
 - [21] P. Grandclément, S. Bonazzola, E. Gourgoulhon, and J.-A. Marck, *J. Comput. Phys.* **170**, 231 (2001).
 - [22] H. P. Pfeiffer, L. E. Kidder, M. A. Scheel, and S. A. Teukolsky, *Comput. Phys. Commun.* **152**, 253 (2003).
 - [23] J. Thornburg, *Class. Quantum Grav.* **21**, 3665 (2004).
 - [24] G. Calabrese and D. Neilsen, *Phys. Rev. D* **71**, 124027 (2005).
 - [25] M. A. Scheel, H. P. Pfeiffer, L. Lindblom, L. E. Kidder, O. Rinne, and S. A. Teukolsky, *Phys. Rev. D* **74**, 104006 (2006).
 - [26] H. P. Pfeiffer, D. Brown, L. E. Kidder, L. Lindblom, G. Lovelance, and M. A. Scheel, *Class. Quantum Grav.* **24**, S59 (2007).
 - [27] E. Schnetter, P. Diener, N. Dorband, and M. Tiglio, *Class. Quantum Grav.* **23**, S553 (2006).
 - [28] M. H. Carpenter, J. Nordström, and D. Gottlieb, *J. Comput. Phys.* **148**, 341 (1999).
 - [29] L. Lehner, O. Reula, and M. Tiglio, *Class. Quantum Grav.* **22**, 5283 (2005).
 - [30] P. Diener, E. N. Dorband, E. Schnetter, and M. Tiglio, *J. Sci. Comput.* **32**, 109 (2007).
 - [31] E. N. Dorband, E. Berti, P. Diener, E. Schnetter, and M. Tiglio, *Phys. Rev. D* **74**, 084028 (2006).
 - [32] E. Pazos, E. N. Dorband, A. Nagar, C. Palenzuela, E. Schnetter, and M. Tiglio, *Class. Quantum Grav.* **24**, S341 (2007).
 - [33] J. Hawley, L. Smarr, and J. Wilson, *Astrophys. J. Suppl. Ser.* **55**, 211 (1984).
 - [34] J. Hawley, L. Smarr, and J. Wilson, *Astrophys. J.* **277**, 296 (1984).
 - [35] J. P. D. Villiers and J. F. Hawley, *Astrophys. J.* **592**, 1060 (2003).
 - [36] J. A. Font, H. Dimmelmeier, A. Gupta, and N. Stergioulas, *Mon. Not. R. Astron. Soc.* (2001), *in press*, [astro-ph/0012477](https://arxiv.org/abs/astro-ph/0012477).
 - [37] R. M. Wald, *General relativity* (The University of Chicago Press, Chicago, 1984).
 - [38] C. W. Misner, K. S. Thorne, and J. A. Wheeler, *Gravitation* (W. H. Freeman, San Francisco, 1973).
 - [39] C. F. Gammie, J. C. McKinney, and G. Tóth, *Astrophys. J.* **589**, 458 (2003).
 - [40] S. C. Noble, C. F. Gammie, J. C. McKinney, and L. Del Zanna, *Astrophys. J.* **641**, 626 (2006).
 - [41] J. A. Font, *Living Rev. Relativity* **3**, 2 (2000).
 - [42] B. J. van Leer, *J. Comput. Phys.* **23**, 276 (1977).
 - [43] A. Harten, P. D. Lax, and B. van Leer, *SIAM Rev.* **25**, 35 (1983).
 - [44] L. Antón, O. Zanotti, J. A. Miralles, J. M. Martí, J. M. Ibáñez, J. A. Font, and J. A. Pons, *Astrophys. J.* **637**, 296 (2006).
 - [45] M. D. Duez, Y. T. Liu, S. L. Shapiro, and B. C. Stephens, *Phys. Rev. D* **p. 024028** (2005).
 - [46] L. Baiotti, I. Hawke, P. Montero, and L. Rezzolla, in *Computational Astrophysics in Italy: Methods and Tools*, edited by R. Capuzzo-Dolcetta (Mem. Soc. Astron. It. Suppl., Trieste, 2003), vol. 1, p. 210.
 - [47] M. D. Duez, P. Marronetti, S. L. Shapiro, and T. W. Baumgarte, *Phys. Rev. D* **67**, 024004 (2003).
 - [48] I. Hawke, F. Löffler, and A. Nerozzi, *Phys. Rev. D* **71**, 104006 (2005).
 - [49] T. Goodale, G. Allen, G. Lanfermann, J. Massó, T. Radke, E. Seidel, and J. Shalf, in *Vector and Parallel Processing –*

VECPAR'2002, 5th International Conference, Lecture Notes in Computer Science (Springer, Berlin, 2003).

- [50] URL <http://www.cactuscode.org>.
- [51] E. Schnetter, S. H. Hawley, and I. Hawke, *Class. Quantum Grav.* **21**, 1465 (2004).
- [52] URL <http://www.carpetcode.org>.
- [53] S. A. Balbus and J. F. Hawley, *Astrophys. J.* **376**, 214 (1991).
- [54] A. Tchekhovskoy, J. C. McKinney, and R. Narayan, *Mon. Not. R. Astron. Soc.* **379**, 469 (2007).
- [55] P. Woodward and P. Collela, *J. Comput. Phys.* **54**, 115 (1984).
- [56] A. Mignone and J. C. McKinney, *Mon. Not. R. Astron. Soc.* **378**, 1118 (2007).
- [57] A. Kageyama, *Geochemistry, Geophysics, Geosystems* **5**, Q09005 (2004).
- [58] J. M. Martí and E. Müller, *Living Rev. Relativity* **2**, 3 (1999).
- [59] N. Stergioulas, *Living Rev. Relativity* **1**, 8 (1998).
- [60] N. Stergioulas and J. L. Friedman, *Astrophys. J.* **444**, 306 (1995).
- [61] L. G. Fishbone and V. Moncrief, *Astrophys. J.* **207**, 962 (1976).
- [62] M. Abramowicz, M. Jaroszynski, and M. Sikora, *Astron. Astrophys.* **63**, 221 (1978).
- [63] M. Kozłowski, M. Jaroszynski, and M. A. Abramowicz, *Astrophys. J.* **63**, 209 (1978).
- [64] M. Jaroszynski, M. A. Abramowicz, and B. Paczynski, *Acta Astronomica* **30**, 1 (1980).
- [65] N. I. Shakura and R. A. Sunyaev, *Astron. Astrophys.* **24**, 337 (1973).
- [66] M. Boyle, D. A. Brown, L. E. Kidder, A. H. Mroue, H. P. Pfeiffer, M. A. Scheel, G. B. Cook, and S. A. Teukolsky (2007), [arXiv:0710.0158](https://arxiv.org/abs/0710.0158) [gr-qc].

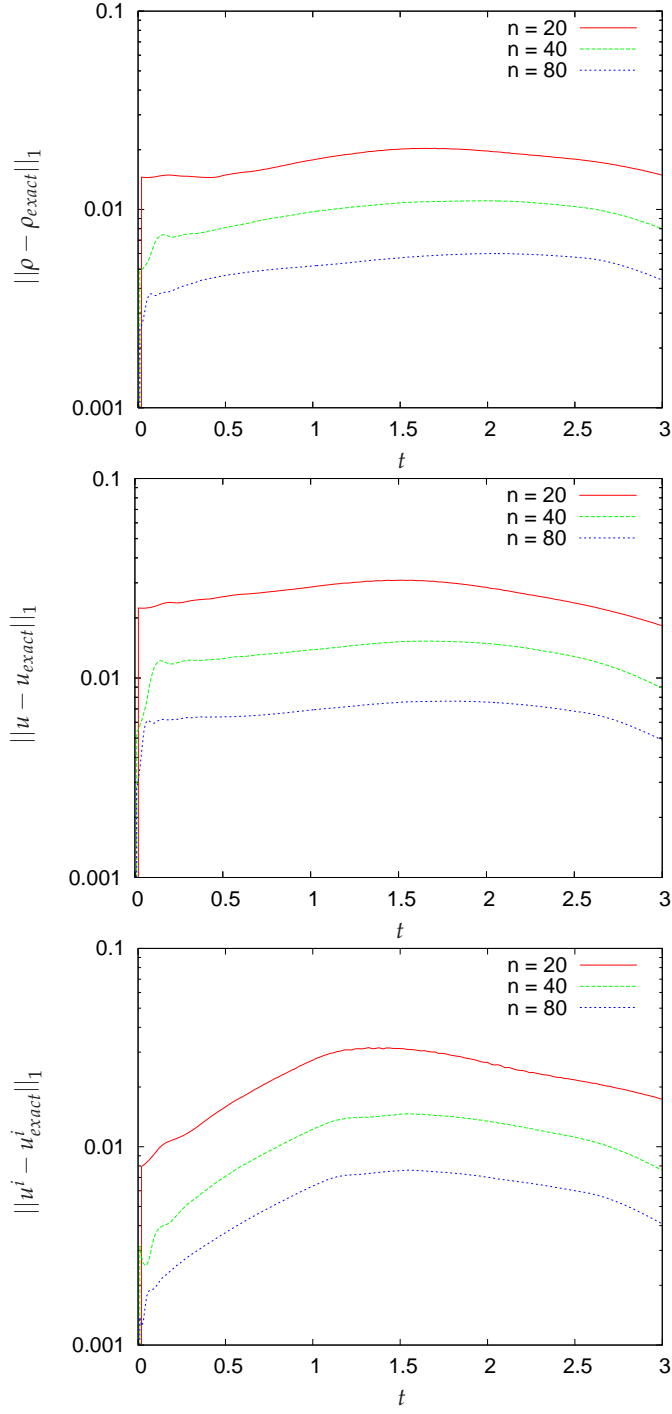


Figure 13: Sod test on the six patches system. These series of plots demonstrate the convergence of the primitive variables to the solution produced by the `riemann` code. Each of the six patches has the same resolution defined by the parameter n , which determines the number of cells by $n \times n \times n$.

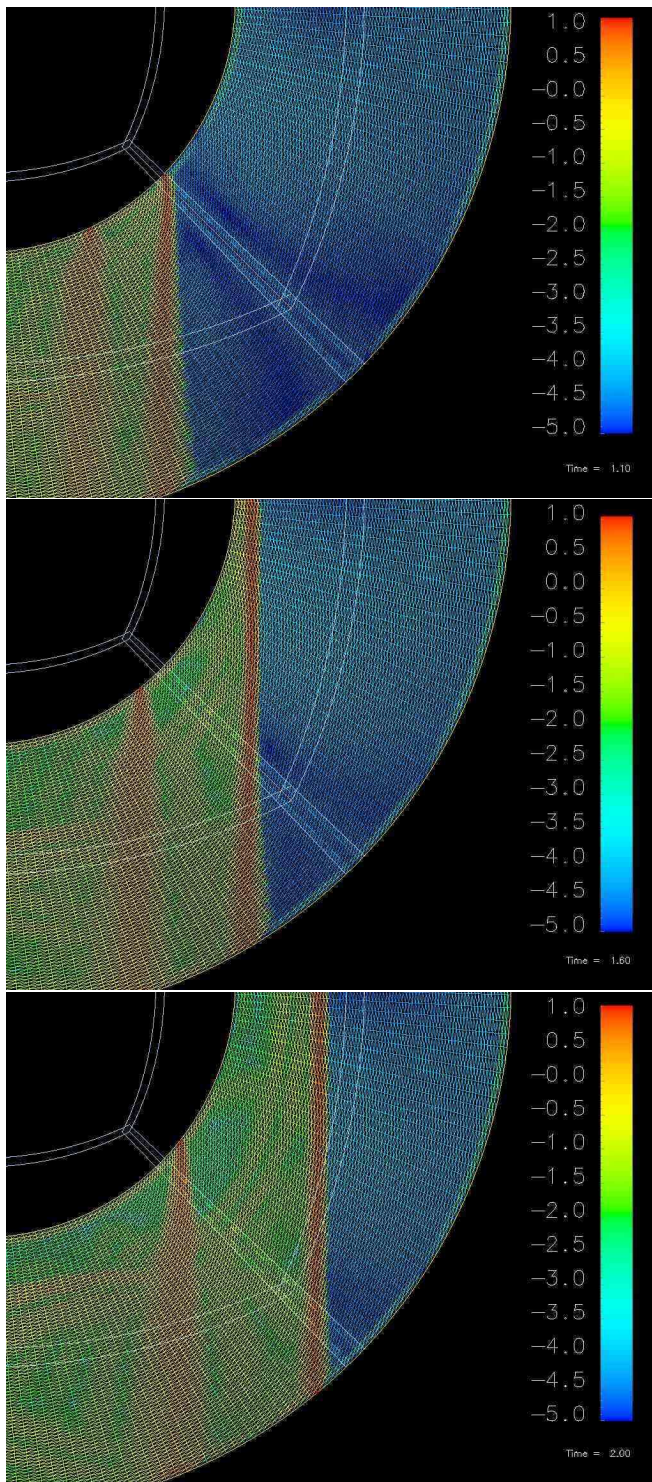


Figure 14: Sod test on the six patches system. The plots show a close-up view of the transmission of the shock front across one of the interfaces, using a *pseudo-Schlieren* plot: The function displayed is the norm of the density gradient, in logarithmic scale. The interface is between the patch boundaries indicated by white lines.

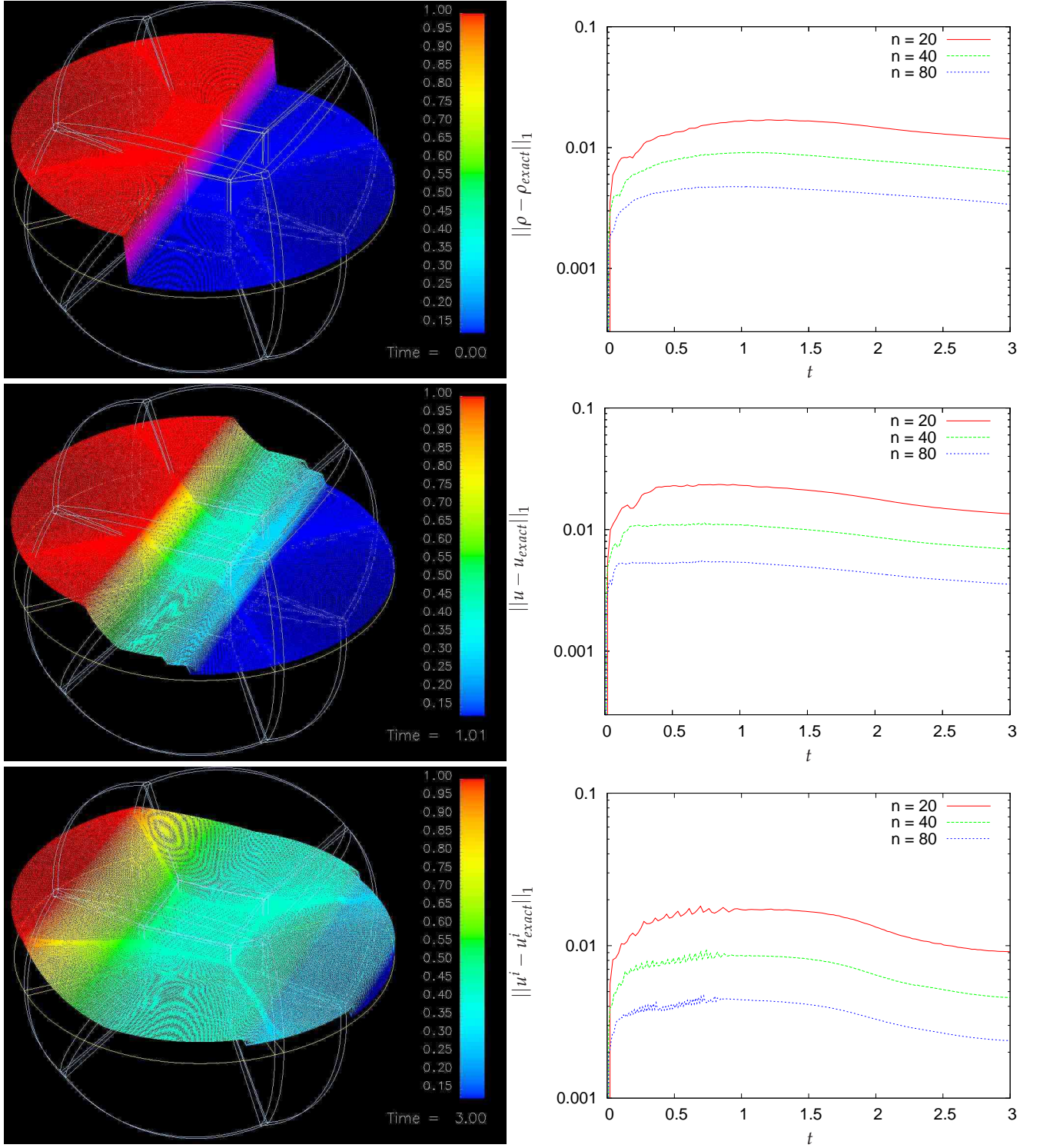


Figure 15: Sod test on the seven patches system. The left panel shows the evolution of the density for the case where each of the patches has resolution $80 \times 80 \times 80$. The white lines indicate the boundaries of the seven patches which are used to cover the computational domain. The yellow surface is the cut of the plane $z = 0$ with the grid boundaries, and the density function is shown at $z = 0$. The right panel shows convergence in the primitive variables.

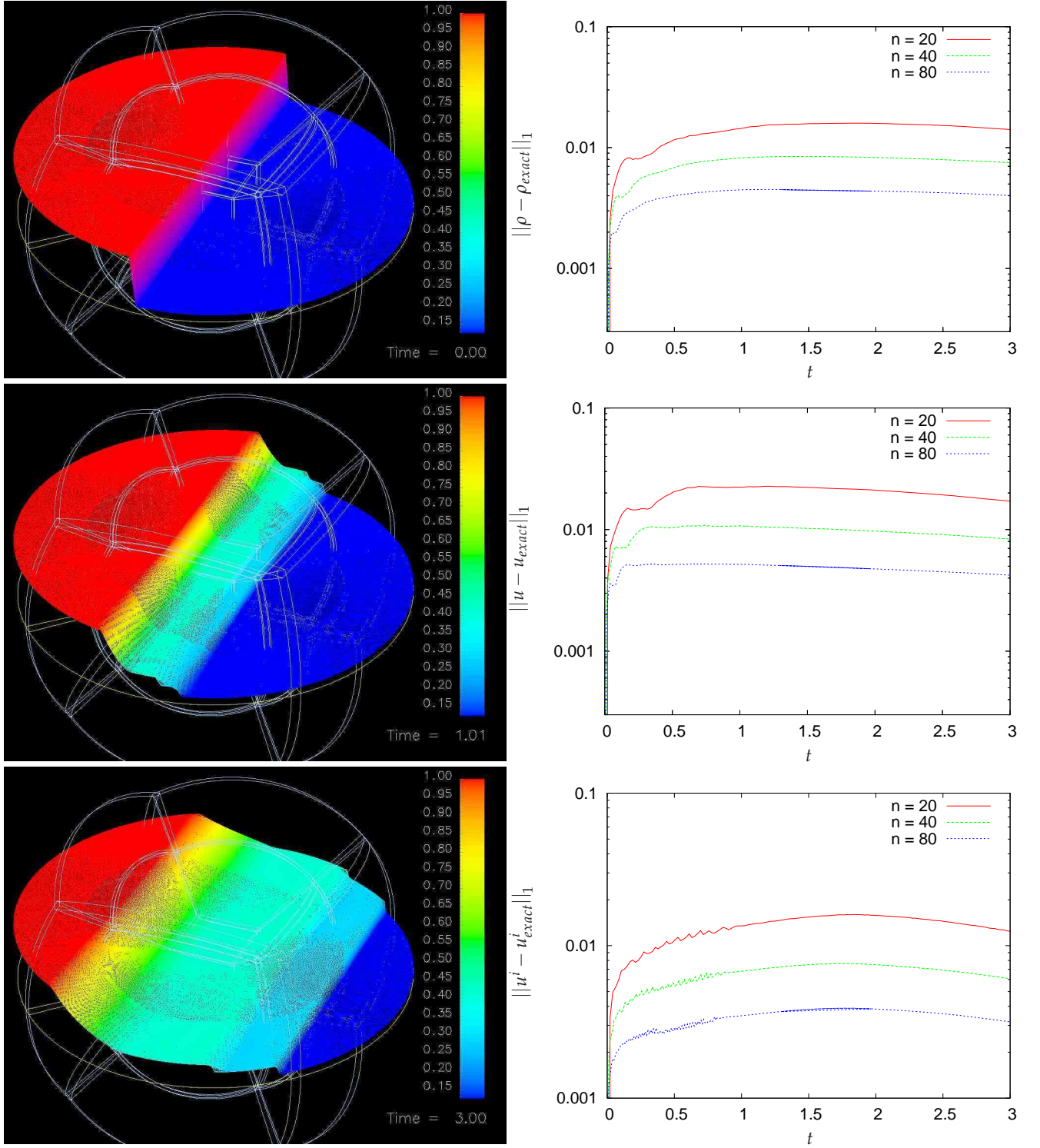


Figure 16: Sod test on the thirteen patches system. The left panel shows the evolution of the density for the case where each of the patches has resolution $80 \times 80 \times 80$. The white lines indicate the boundaries of the thirteen patches which are used to cover the computational domain. The yellow surface is the cut of the plane $z = 0$ with the grid boundaries, and the density function is shown at $z = 0$. The right panel shows convergence in the primitive variables.

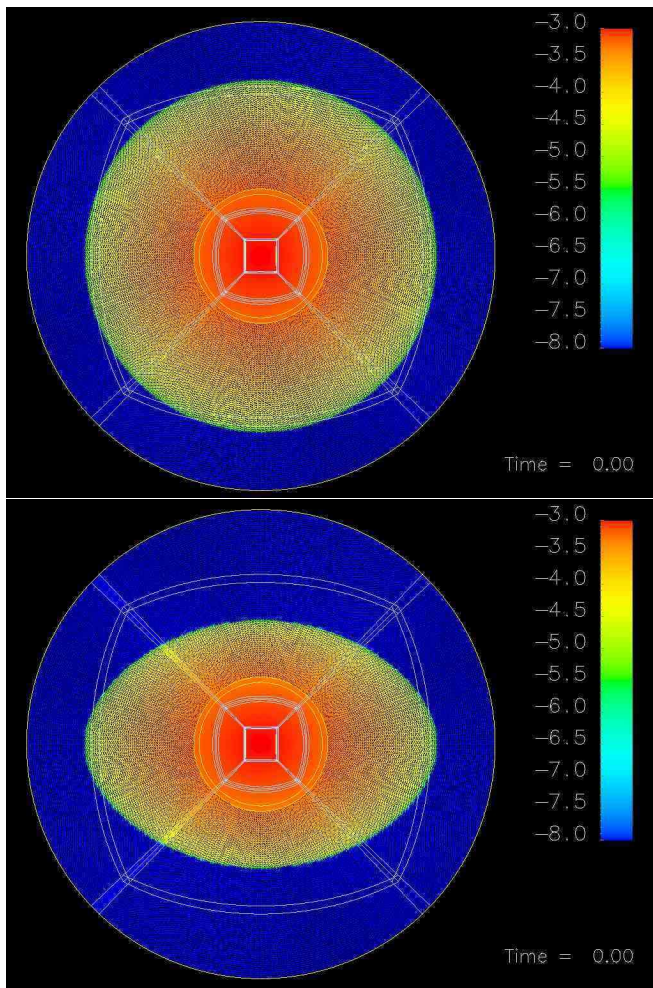


Figure 17: Rotating neutron star on the cubed sphere system of thirteen patches. The initial data is produced with the `rns` code [60] then mapped to the each patch and transformed to the local coordinate system. These plots show the logarithm of the density function in the equatorial plane $z = 0$ (left) and the plane $x = 0$ (right). For visual clarity, the density is cut below 10^{-8} ; the actual initial density in the atmosphere is 10^{-10} .

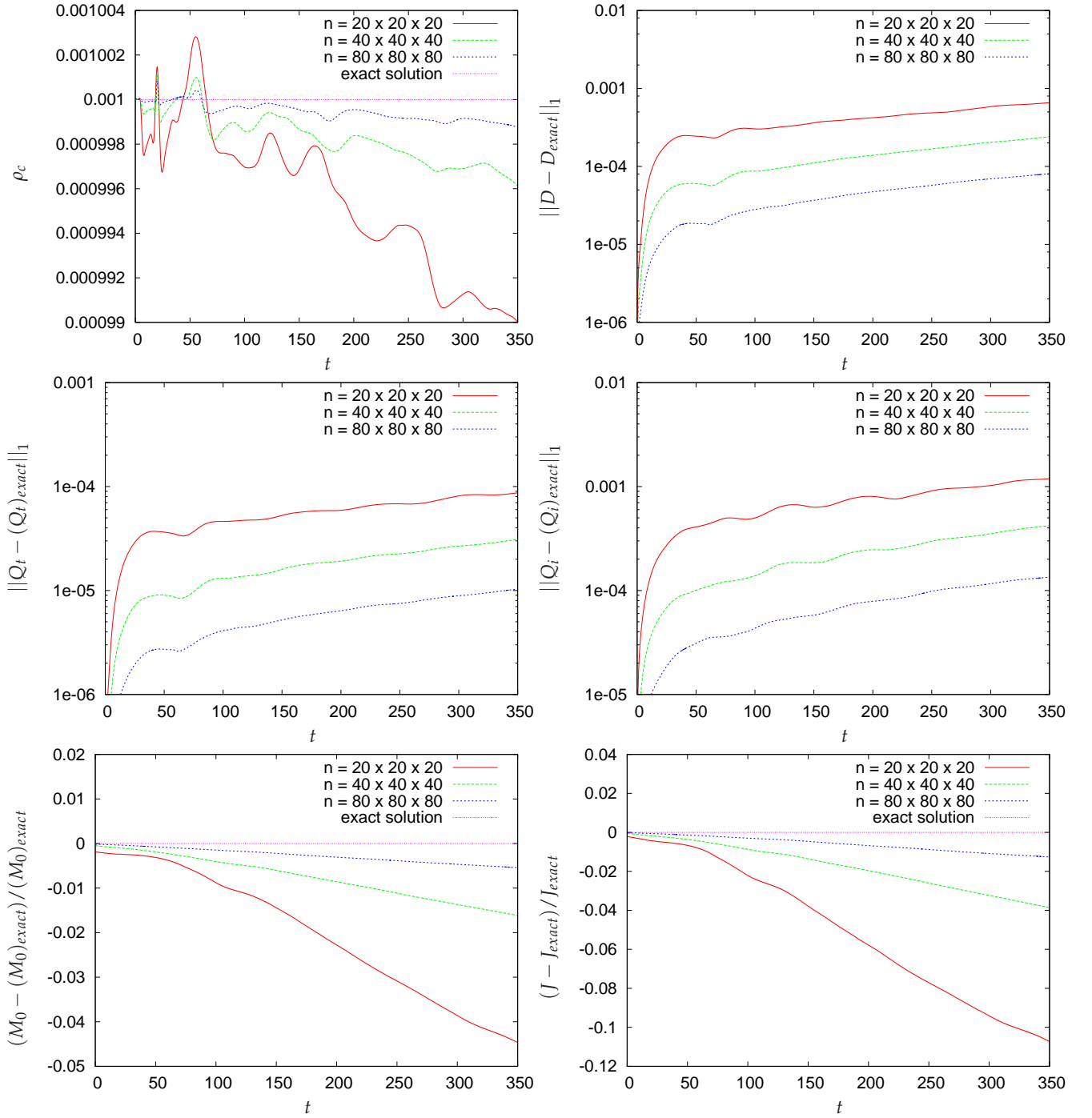


Figure 18: Evolution of a rotating neutron star on the cubed sphere system of thirteen patches with different resolutions. The star is evolved up to a coordinate time of 350, which translates into about ten dynamical times ($t_D = R_e \sqrt{R_e/M} \approx 35.428$). The top left panel shows the central density, the top right, middle left, and middle right panels show convergence of the conserved variables, and the last two plots show the errors in rest mass (bottom left panel) and angular momentum (bottom right panel). For a definition of these quantities see [59].

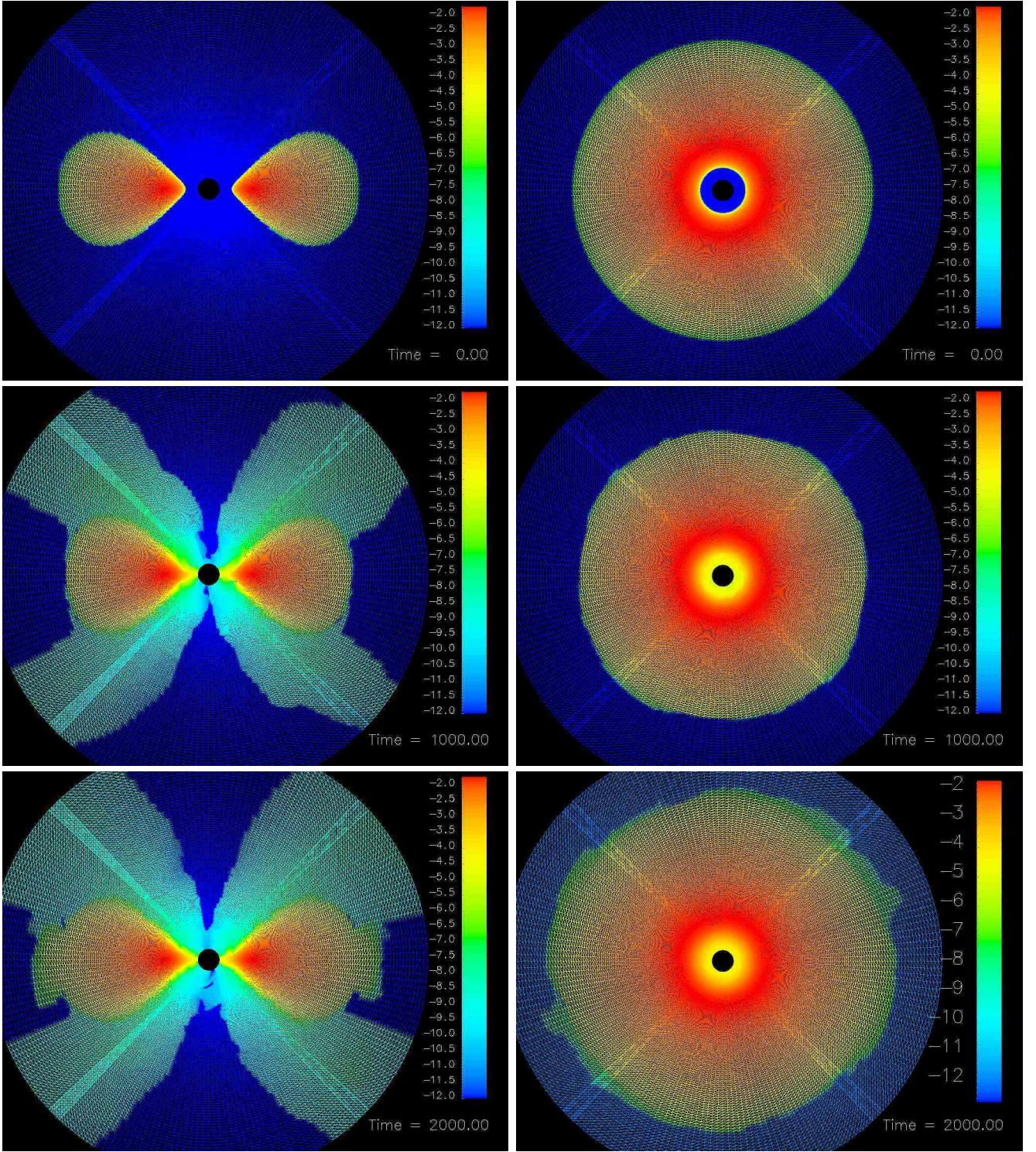


Figure 19: Equilibrium accretion torus on a Schwarzschild background, using the cubed sphere six patches system. The plots show, for the evolution with $60 \times 60 \times 160$ cells per patch, the decadic logarithm of the density for coordinate times $t = 0$, 1000 and 2000 (the rotational time of the torus at the locus of highest density is about 200, i.e., we are evolving for about 10 rotational times). The left panel shows the cross-section surface $x = 0$ in terms of the global, pseudo-Cartesian coordinates, and the right panel provides a view of the equatorial plane $z = 0$. The system is stationary, so all deviations from the initial data is an artifact due to the finite resolution, finite precision of floating point numbers, or the use of an artificial atmosphere (though small numerical errors might initiate physical instabilities in the disk). A small amount of material gets unbound and is either accreted into the black hole or forms a corona. In disks where a real physical effect like turbulence causes redistribution of angular momentum [35], similar domains are present, though with much higher densities (note that the color map covers a range of 10 orders of magnitude in density).

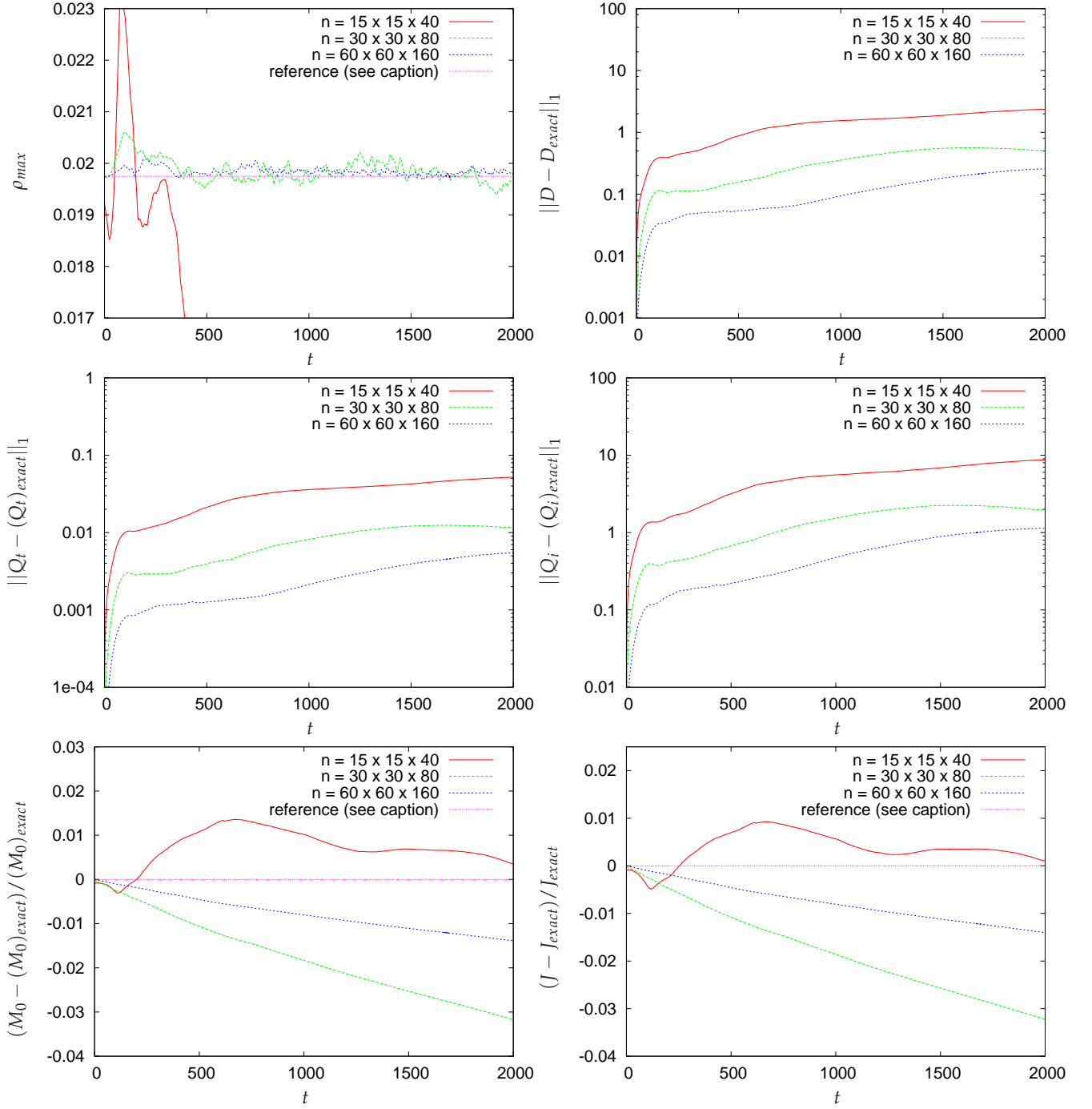


Figure 20: Equilibrium accretion torus on a Schwarzschild background, using the cubed sphere six patches system. The disk is evolved up to a coordinate time of 2000, which translates into about ten rotational times at the location of maximal density. The top left panel shows the maximal density, the top right, middle left, and middle right panels show convergence of the conserved variables, and the last two plots show the errors in rest mass (bottom left panel) and angular momentum (bottom right panel). For a definition of these quantities see [59]. (The “reference” solution mentioned in the text is the value corresponding to the initial data on grid of $120 \times 120 \times 320$ cells per patch.) Note that the case $n = 15 \times 15 \times 40$ is clearly under-resolved.