

---

---

# Discontinuous Galerkin Methods for General Relativistic Hydrodynamics

---

---

## DISSERTATION

zur Erlangung des akademischen Grades  
doctor rerum naturalium (Dr. rer. nat.)



---

seit 1558

vorgelegt dem Rat der  
PHYSIKALISCH-ASTRONOMISCHEN FAKULTÄT  
der  
FRIEDRICH-SCHILLER-UNIVERSITÄT JENA  
von

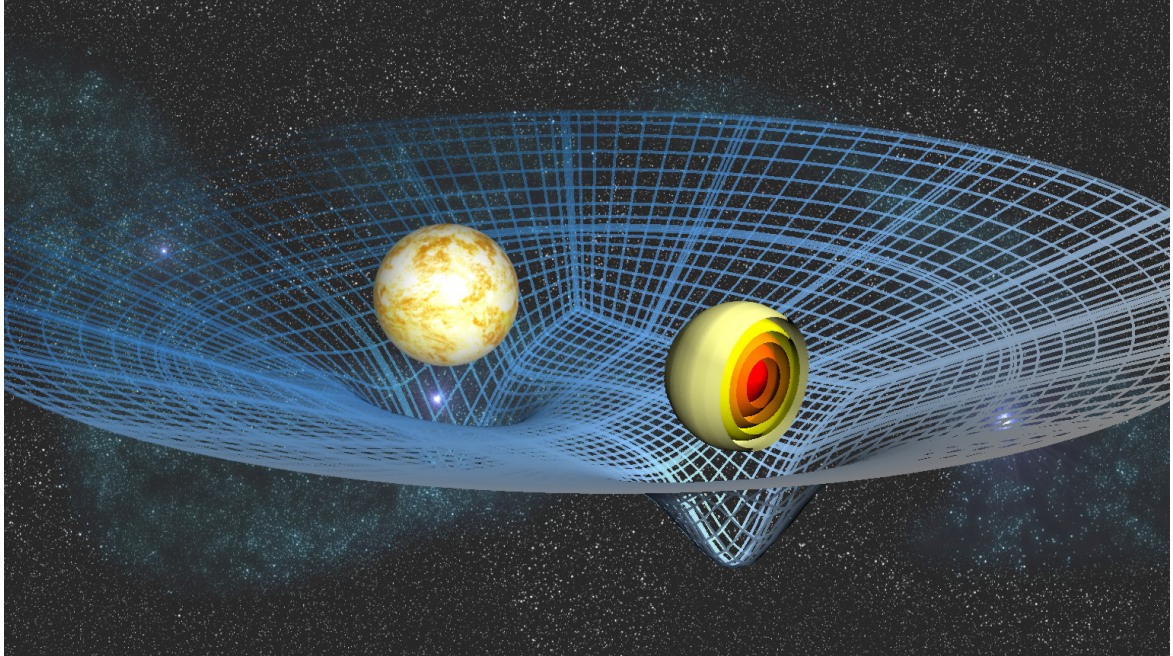
**Dipl.-Phys. Marcus Bugner**

geboren am 13.09.1989 in Köthen (Anhalt)

**GUTACHER:**

1. PROF. DR. BERND BRÜGMANN (FRIEDRICH-SCHILLER-UNIVERSITÄT JENA)
2. PROF. DR. GERHARD ZUMBUSCH (FRIEDRICH-SCHILLER-UNIVERSITÄT JENA)
3. PROF. DR. WOLFGANG TICHY (FLORIDA ATLANTIC UNIVERSITY)

**TAG DER DISPUTATION: 02.11.2017**



*Two neutron stars on the bamps cubed-ball grid.  
Artistic impression, density profile and lapse*





# Abbreviations and Notation

Throughout this work geometrized units are employed with  $c = G = M_{\odot} = 1$ . These units correlate to physical units of time, length, mass and mass density as follows:

$$\begin{aligned}\Delta t = 1 &\equiv 4.9266 \mu s, & m = 1 &\equiv 1.9889 \cdot 10^{30} \text{ kg} \\ \Delta s = 1 &\equiv 1.4769 \text{ km}, & \rho = 1 &\equiv 6.1733 \cdot 10^{17} \frac{\text{g}}{\text{cm}^3}\end{aligned}$$

We use Greek letters for four-dimensional indices running from 0 to 3 and Latin letters for three-dimensional indices running from 1 to 3.

We refer to most references using the first authors and the year of publication, except for references to our own publications for which we use Arabic numerals.

The following abbreviations are used throughout the thesis, in most cases these abbreviations are also introduced in the text at their first appearance:

ADM	Arnowitt-Deser-Misner
BNS	Binary neutron star
BSSN	Baumgarte-Shapiro-Shibata-Nakamura
DF	Dual foliation
DG	Discontinuous Galerkin
EOS	Equation of state
GHG	Generalized harmonic gauge
GR	General relativity
GRHD	General relativistic hydrodynamics
HRSC	High resolution shock capturing
NR	Numerical relativity
PDE	Partial differential equation
RHS	Right hand side
RNS	Rotating neutron star
SRHD	Special relativistic hydrodynamics
SV	Spectral volume
TOV	Tolman–Oppenheimer–Volkoff
WENO	Weighted essentially non-oscillatory



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Numerical treatment of general relativistic hydrodynamics</b>	<b>7</b>
2.1	3+1 decomposition in numerical relativity . . . . .	8
2.2	Recasting the matter equations . . . . .	10
2.3	Recasting the Einstein field equations . . . . .	15
2.4	The dual foliation formalism . . . . .	17
<b>3</b>	<b>Numerical methods</b>	<b>21</b>
3.1	Pseudospectral methods and the bamps code . . . . .	21
3.2	The Runge-Kutta Discontinuous Galerkin method . . . . .	29
3.3	High resolution shock capturing methods . . . . .	36
3.3.1	Weighted-essentially-non-oscillatory methods . . . . .	37
3.3.2	The spectral volumes method . . . . .	44
<b>4</b>	<b>Code tests</b>	<b>51</b>
4.1	Tests in flat spacetime . . . . .	51
4.2	Spherical accretion . . . . .	63
4.3	Stable neutron star . . . . .	68
4.4	Unstable neutron star . . . . .	77
4.5	Rotating neutron star . . . . .	79
<b>5</b>	<b>Neutron star head-on collision</b>	<b>81</b>
<b>6</b>	<b>Numerical experiments with a dual foliation</b>	<b>85</b>
<b>7</b>	<b>Conclusion</b>	<b>95</b>
	<b>Appendix</b>	<b>97</b>
A	Recovery of the primitive variables . . . . .	99
B	Derivation of the GHG representation of the Ricci tensor . . . . .	100
C	Derivation of the explicit Gauss-Legendre-Lobatto mass matrix . . . . .	102

<b>Bibliography</b>	<b>102</b>
<b>Publications</b>	<b>111</b>
<b>Acknowledgement</b>	<b>113</b>
<b>Additional Material</b>	<b>115</b>

# Chapter 1

## Introduction

Almost exactly 100 years after Albert Einstein published his famous series of papers on general relativity [Einstein, 1915b; Einstein, 1915a], the first gravitational wave signal was observed on September 14, 2015 by the Laser Interferometer Gravitational-Wave Observatory (LIGO) [Abbott et al., 2016b]. Although there had already been strong evidence for the existence of gravitational waves through indirect observations [Hulse and Taylor, 1975], the direct observation represents the longest pending confirmation of general relativity. It also marks the birth of gravitational wave astronomy, which will most certainly enhance our knowledge of exotic objects in the universe. During the construction of the ground based detectors [LIGO; VIRGO; GEO], expertise from several fields of physics was combined to push the detector sensitivity to the desired level. With the future earthbound [KAGRA; IND; ET] and space facilities [LISA], a global network of detectors will be available. However, due to the diverse sources of noise it will remain challenging to extract the gravitational wave signal from raw data containing noise. Therefore, a complex data analysis machinery has been developed to compare wave templates against the detector signal [LAL]. These gravitational wave templates are usually generated from post-Newtonian considerations [Blanchet, 2006], the effective one body (EOB) approach [Buonanno and Damour, 1999; Damour, 2001], or from the full numerical solution of the field equations of general relativity. In most cases, numerical simulations are needed to feed and tune analytical methods [Ajith et al., 2011; Taracchini et al., 2014] for waveform modeling. Therefore, efficient and accurate general relativistic simulations will be crucial for future gravitational wave astronomy.

The most promising source of gravitational waves are binary systems of fast moving, compact objects such as black holes and neutron stars. Although all currently observed gravitational wave events originated from black hole coalescences [Abbott et al., 2016b; Abbott et al., 2016a; Abbott et al., 2017], binary neutron stars (BNSs) are a promising source for upcoming detectors. In [Abbott et al., 2013], a BNS event

rate of 0.4 – 400 per year is expected for future detectors with a range of 200 Mpc. In contrast to binary black hole mergers, the presence of matter leads to additional physical phenomena [Baiotti and Rezzolla, 2016]: The initiation of short gamma-ray bursts [Eichler et al., 1989; Narayan et al., 1992], the creation of heavy nuclei through the r-process [Freiburghaus et al., 1999] or neutrino emission [Waxman, 2004; Dessart et al., 2009]. From a gravitational wave signal of coalescing neutron stars, unique information about their internal structure, i.e. the so far unknown equation of state may be extracted [Read et al., 2009].

Since the first simulation of BNSs [Shibata and Uryū, 2000], a variety of numerical codes for general relativistic hydrodynamics (GRHD) simulations have been developed [Font et al., 2000; Baiotti et al., 2005; Liu et al., 2008; Yamamoto et al., 2008; Thierfelder et al., 2011; Radice and Rezzolla, 2012]. All these codes are based on finite volume or finite difference methods. Parallelization is mostly realized by a domain decomposition and a shared memory/message passing paradigm [MPI]. For this purpose, some part of the grid has to be sent to other processors. Typically, the amount of communication increases with the order of the scheme. This makes efficient parallelization of these codes particularly hard. Spectral methods have the potential to solve this issue and are therefore very popular candidates for modern numerical relativity codes [SpEC; Tichy, 2006; Hilditch et al., 2016]. They allowed for impressively efficient and parallel simulations of vacuum spacetimes, e.g. gravitational wave collapse or binary black hole systems. Since spectral methods are designed for smooth solutions and matter fields typically contain shocks and discontinuities, it is particularly hard to treat both spacetime and matter spectrally. This is why production codes still rely on finite volume/difference methods for the treatment of hydrodynamical fields. Also hybrid codes like [Duez et al., 2008] have been developed.

The main objective of this thesis is to implement and investigate a novel class of numerical methods for the matter treatment in GRHD codes: The Discontinuous Galerkin (DG) methods. Since its first application [Reed and Hill, 1973], DG methods have emerged as a successful general purpose paradigm in the recent years [Canuto et al., 2006; Hesthaven and Warburton, 2008; Kopriva, 2009]. Combining the key advantages of finite volume and traditional finite element methods, e.g. element locality and *hp*-adaptivity [Karniadakis and Sherwin, 2005], DG methods offer the treatment of complex grid geometries and simple parallelization. Actually, DG methods require only communication of the shared interface layer of non-overlapping elements, independent of the order of the scheme. With its application on hyperbolic conservation laws [Cockburn and Shu, 1989; Cockburn and Shu, 1998], DG methods found its way into first general relativity applications [Zumbusch, 2009; Field et al., 2009; Radice and Rezzolla, 2011]. With the present work, we want to explore the

features and usability of DG methods in the context of GRHD.

Apart from almost trivial test cases, we expect that occurring shocks are still a challenging problem, also in the context of DG methods. Therefore, focus is also put on high resolution shock capturing (HRSC) methods as an extension to the plain DG method. Concerning this combination, two main strategies have seen remarkable development in the past years: i) a generalization of the weighted essentially non-oscillatory (WENO) reconstruction [Liu and Osher, 1994] for DG methods [Qiu and Shu, 2005; Zhong and Shu, 2013; Zhao and Tang, 2013]; ii) a subdivision strategy to divide the DG element in sub-volumes, which are then treated by another more robust (but possibly lower order) method. In [Wang, 2002; Radice and Rezzolla, 2011], this approach leads into the “spectral volumes” (SV) idea, while in the comprehensive works [Dumbser et al., 2013; Zanotti and Dumbser, 2014; Dumbser et al., 2014], “subcells” were developed as the key ingredient for complex adaptive mesh refinement schemes.

Besides the evolution algorithm, the actual extraction of the gravitational wave signal from the numerical data is a problem on its own. For an unambiguous and coordinate independent wave extraction, future null infinity should be included in the numerical grid. In most numerical relativity codes, gravitational waves are simply extracted at a sufficiently big (but finite) radius [Bishop and Rezzolla, 2016]. Another approach is the interpolation of numerical data from spatial slices to null slices, followed by extrapolation to infinity [Pollney et al., 2009; Boyle and Mroué, 2009]. As a last alternative strategy, Cauchy-characteristic matching [Winicour, 1998] should be named. Here, a global solution is obtained by matching the outer boundary of spatial (Cauchy) slices with the inner boundary of characteristic (null) slices that extend to future null infinity. Within this thesis, we present the first numerical test of a new approach, based on the dual foliation (DF) formalism [Hilditch, 2015]. The plan is to choose hyperboloidal coordinates as a second coordinate frame and to extend the outer parts of our numerical domain to future null infinity. Although these first tests only consider the flat wave equation [Bug2], the procedure is a future candidate for highly accurate gravitational wave extraction.

The structure of the thesis is as follows: We introduce the main concepts of numerical relativity in chapter 2. Starting from Einstein’s field equations, we give a rough guideline on the derivation of equations which are used in our code. Besides the GHG and “Valencia” formulations, we also introduce the novel DF formalism. In chapter 3, we concentrate on the implementation of the formerly derived equations. A short summary on the existing `bamps` code is given. This is followed by a detailed description on DG and HRSC methods, that we plan to use for the matter treatment in our code. In chapter 4, we present and analyze all test simulations for our new implementation. The behavior of several building blocks of our algorithm and its

convergence are investigated for a set of test cases with increasing complexity. In chapter 5, we show that our code can actually be used for simulating the dynamics of a BNS system and to extract the corresponding gravitational waveform. First tests of the DF implementation are shown in chapter 6. In chapter 7, we summarize and a final comment on DG methods in the context of GRHD is given. Additionally, some remarks on potentially interesting future developments are given.



# Chapter 2

## Numerical treatment of general relativistic hydrodynamics

The objective of this thesis is the numerical evolution of general relativistic neutron stars. This simple statement comprises several non-trivial facts. Evolving a “star” means to evolve matter according to the conservation of rest mass and energy-momentum

$$\nabla_{\mu} j^{\mu} = 0, \quad (2.1a)$$

$$\nabla_{\mu} T^{\mu\nu} = 0. \quad (2.1b)$$

The term “general relativistic” imposes that Einstein’s field equations have to be solved simultaneously:

$$R_{\mu\nu} = 8\pi \left( T_{\mu\nu} - \frac{1}{2} T g_{\mu\nu} \right). \quad (2.1c)$$

The remarkable and defining feature of covariant theories is the invariance of equations under tensor basis changes. Equations (2.1) satisfy this requirement, so that all possible coordinates, i.e. spatial coordinates and time, are treated equally. In order to “evolve” a state, the underlying problem has to be recast as an initial value problem. It is achieved by reverting to a formulation, in which space and time are separated. This so called *3+1 decomposition* was originally proposed in the works of Darmois, Lichnerowicz and Choquet-Bruhat [Lichnerowicz, 1944; Bruhat, 1952]. Later, it was used very successfully by Arnowitt, Deser and Misner [Arnowitt et al., 1962; York, 1979]. We collect the essential parts of this formalism in Sec. 2.1 and continue with its application on eqs. (2.1a,b) in Sec. 2.2. Detailed calculations on decomposing covariant equations in 3+1 form can be found in [Gourgoulhon, 2012; Baumgarte and Shapiro, 2010; Alcubierre, 2008]. In Sec. 2.3 we describe our coordinate choice - or gauge - for (2.1), the *Generalized harmonic gauge* (GHG) [Lindblom et al., 2006]. In Sec. 2.4 the main ideas and features of the *dual foliation formalism* [Hilditch,

2015],[Bug2], i.e. the usage of two different foliations and/or gauges, are presented.

## 2.1 3+1 decomposition in numerical relativity

A wide class of spacetime manifolds  $\mathcal{M}$  is globally hyperbolic, meaning that there is a Cauchy surface  $\Sigma$  such that the topology of  $\mathcal{M}$  is necessarily  $\Sigma \times \mathbb{R}$  [Gourgoulhon, 2012]. Exploiting this property, one introduces a smooth and regular scalar field  $t$  and foliates the four-dimensional spacetime  $(\mathcal{M}, g_{\mu\nu})$ , where  $g_{\mu\nu}$  is a metric with signature  $(-, +, +, +)$ , into three-dimensional spacelike level surfaces of  $t$ . Given the data on slice  $\Sigma_t$ , an “evolution” then refers to the process of finding compatible data at slice  $\Sigma_{t+\delta t}$ . The required equations originate from a split of the fully covariant tensor equations along and orthogonal to the slices  $\Sigma_t$ . The orthogonal direction to  $\Sigma_t$  is given by the normalized 1-form

$$\Omega_\mu = \alpha \nabla_\mu t, \quad \alpha = [-(\nabla_\mu t)(\nabla^\mu t)]^{-\frac{1}{2}}, \quad (2.3a)$$

where the normalization  $\alpha$  is called *lapse* [Wheeler, 1964], and the corresponding *normal vector*

$$n^\mu = -g^{\mu\nu} \Omega_\nu. \quad (2.3b)$$

By Taylor expansion it is easy to verify, that  $n^\mu$  points in a direction of increasing  $t$  (since  $\alpha > 0$ ). Since  $n^\mu$  is normalized and timelike, one can think of it as the four-velocity field of a family of observers. Locally,  $\Sigma_t$  can be seen as the set of simultaneous events for this *Eulerian observer* [Gourgoulhon, 2012]. In contrast to  $n^\mu$ , the vector  $\alpha n^\mu$  has the property

$$t(x^\mu + \delta t \cdot \alpha n^\mu) = \hat{t} + \delta t, \quad (2.5)$$

where  $x \in \Sigma_{\hat{t}}$ , i.e. it maps a slice to its infinitesimally displaced neighbor (see Fig. 2.1). Introducing coordinates  $x^\mu = (t, x^i)$ , another vector with this feature can be constructed by the displacement  $(t, x^i) \mapsto (t + \delta t, x^i)$ , i.e. one that conserves the spatial coordinates. This vector is typically called *time vector*  $t^\mu$ . The difference between  $\alpha n^\mu$  and  $t^\mu$  is tangent to the slice  $\Sigma_t$  and called the *shift vector*  $\beta^\mu$ . All defined vectors are therefore related by the fundamental relation

$$t^\mu = \alpha n^\mu + \beta^\mu, \quad (2.6)$$

which is depicted in Fig. 2.1.

As mentioned earlier, we want to project tensor equations into and orthogonal to  $\Sigma_t$ . To do so, we define the projector  $\gamma_\mu^\nu = \delta_\mu^\nu + n_\mu n^\nu$ . By lowering one index, it is

clear that

$$\gamma_{\mu\nu} = g_{\mu\nu} + n_\mu n_\nu \quad (2.7)$$

is identical to the full metric  $g_{\mu\nu}$  for vectors tangent to  $\Sigma_t$ , which is why  $\gamma_{\mu\nu}$  is called *induced metric*. At this point, we already have all ingredients in place to express the full metric in terms of the new functions  $\alpha$ ,  $\beta^i$  and  $\gamma_{ij}$ . We choose a basis adapted to the coordinates  $(t, x^i)$ , so that  $t^\mu = (1, 0, 0, 0)$  and with (2.3)  $n_\mu = (-\alpha, 0, 0, 0)$ . Since  $\beta^\mu$  is tangent to  $\Sigma_t$ ,  $\beta^\mu = (0, \beta^i)$  (more generally, all zeroth components of spatial contravariant tensors vanish [Baumgarte and Shapiro, 2010]). With the normalization of  $n^\mu$  and (2.6) one finds the components of  $n^\mu = \left(\frac{1}{\alpha}, -\frac{\beta^i}{\alpha}\right)$ . The inverse metric is then directly derivable as

$$g^{\mu\nu} = \begin{pmatrix} -\frac{1}{\alpha^2} & \frac{\beta^i}{\alpha^2} \\ \frac{\beta^j}{\alpha^2} & \gamma^{ij} - \frac{\beta^i \beta^j}{\alpha^2} \end{pmatrix}, \quad (2.8)$$

and it follows by inversion that

$$g_{\mu\nu} = \begin{pmatrix} -\alpha^2 + \beta^k \beta_k & \beta_i \\ \beta_j & \gamma_{ij} \end{pmatrix}. \quad (2.9)$$

One usually introduces further objects and abbreviations in the 3+1 context. In the following, those which are used in the thesis are defined and briefly described.

The *Three-dimensional covariant derivative*  $D$  is uniquely defined by the compatibility property [Baumgarte and Shapiro, 2010]

$$D_\mu \gamma_{\alpha\beta} = 0, \quad (2.10)$$

which is fulfilled if  $D$  is constructed as the fully projected four-dimensional covariant derivative

$$D_\mu T_{\beta_1 \dots \beta_m}^{\alpha_1 \dots \alpha_n} = \gamma_\mu^\nu \gamma_{\delta_1}^{\alpha_1} \dots \gamma_{\delta_n}^{\alpha_n} \gamma_{\beta_1}^{\varepsilon_1} \dots \gamma_{\beta_m}^{\varepsilon_m} \nabla_\nu T_{\varepsilon_1 \dots \varepsilon_m}^{\delta_1 \dots \delta_n}, \quad (2.11)$$

for spatial tensors  $T_{\beta_1 \dots \beta_m}^{\alpha_1 \dots \alpha_n}$ . As the induced metric defines a derivative on the slice, it is possible to measure the intrinsic curvature on  $\Sigma_t$ . The appropriate spatial curvature quantities are defined in the same way as for the manifold  $\mathcal{M}$ , except for the substitutions  $g_{\mu\nu} \rightarrow \gamma_{\mu\nu}$  and  $\nabla \rightarrow D$ .

Besides their intrinsic curvature, the slices can bend inside the ambient manifold. This is reflected by the *extrinsic curvature*, which measures the change of the normal vector along a path in  $\Sigma_t$ :

$$K_{\mu\nu} = -\gamma_\mu^\alpha \nabla_\alpha n_\nu. \quad (2.12)$$

Using (2.7), the metric compatibility of  $\nabla$  and  $n_\mu \nabla_\nu n^\mu = 0$ , it is straightforward to

show that

$$-2K_{\mu\nu} = \mathcal{L}_n \gamma_{\mu\nu}, \quad (2.13)$$

which indicates that  $K_{\mu\nu}$  carries information about the temporal change of the induced metric. Actually, the extrinsic curvature is a crucial part of the conjugate momentum tensor in the Hamiltonian formulation of general relativity [Arnowitt et al., 1962].

Commonly, the *acceleration of the Eulerian observer*  $a_\mu$  is defined as

$$a_\mu = n^\nu \nabla_\nu n_\mu, \quad (2.14)$$

and vanishes if the observer moves along a geodesic.  $a_\mu$  is tangential to  $\Sigma_t$  and can alternatively be expressed as a spatial derivative:

$$a_\mu = D_\mu \ln \alpha. \quad (2.15)$$

As a last remark in this section, we want to comment on Lie and time derivatives. The latter are required to evolve data from  $\Sigma_t$  to  $\Sigma_{t+\delta t}$ . For scalar data  $u$ , the temporal derivative can easily be recovered as

$$\partial_t u = t^\mu \partial_\mu u = \alpha n^\mu \nabla_\mu u + \beta^i D_i u. \quad (2.16)$$

For a spatial covector  $w_i$  we use the properties of the Lie derivative in adapted coordinates to find

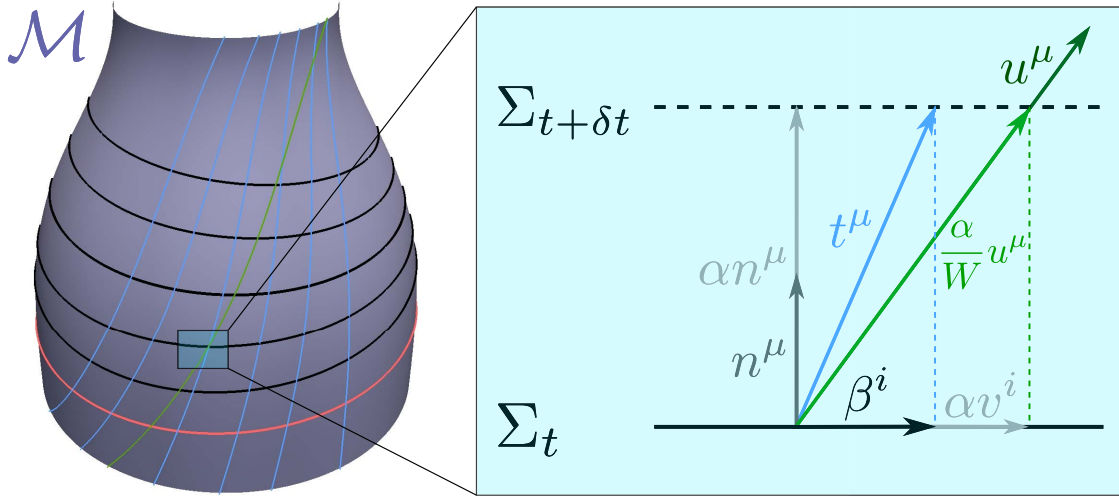
$$\begin{aligned} \partial_t w_i &= \mathcal{L}_t w_i = \mathcal{L}_{\alpha n} w_i + \mathcal{L}_\beta w_i = \gamma_i^j \mathcal{L}_{\alpha n} w_j + \mathcal{L}_\beta w_i \\ &= \alpha (n^\mu \gamma_i^j \nabla_\mu w_j - w_i K_j^i) + \mathcal{L}_\beta w_i, \end{aligned} \quad (2.17)$$

which relates its covariant derivative along  $n^\mu$  and its time derivative. The  $\mathcal{L}_\beta$  term in (2.17) can either be expressed in terms of the coordinate derivatives  $\partial_i$  or the covariant derivatives  $D_i$ .

## 2.2 Recasting the matter equations

We want to apply the 3+1 decomposition on eqs. (2.1a,b). This step is covered in many textbooks which are already mentioned in the beginning of this chapter. Nevertheless, we want to point out that in particular [Rezzolla and Zanotti, 2013] focuses on relativistic fluids and its numerical treatment. The particular set of equations that will be derived in this chapter is called the *Valencia formulation of relativistic hydrodynamics* [Martí et al., 1991; Font et al., 1994; Banyuls et al., 1997].

We start from modeling the fluid as a perfect fluid, by defining  $u^\mu$  as the fluid four-velocity field and the scalar fields  $\rho, p, \varepsilon$  as the fluid rest mass density, pressure



**Figure 2.1:** 3+1 decomposition: The manifold  $\mathcal{M}$  (purple) is sliced by a scalar function  $t$ . The slices  $\Sigma_t$  are hypersurfaces where  $t = \text{const}$  (black). Blue lines are lines of constant spatial coordinates. The initial slice is colored red. A fluid particle worldline is sketched in green. The infinitesimal vicinity of a point in  $\Sigma_t$  is shown on the right. The meaning of the important 3+1 quantities lapse  $\alpha$ , shift vector  $\beta^i$ , normal vector  $n^\mu$  and time vector  $t^\mu$  is depicted. Additionally, the four-velocity field  $u^\mu$  and the three-velocity field  $v^i$  of the fluid are illustrated.

and specific internal energy field. These fields are related to each other by an *equation of state* (EOS), i.e.  $p = p(\rho, \varepsilon)$ , which represents the microscopic characteristics of the matter. In this work, we use either the *polytropic EOS*

$$p = K\rho^\Gamma, \quad (2.18)$$

or the *ideal gas EOS*

$$p = \rho\varepsilon(\Gamma - 1). \quad (2.19)$$

The rest mass four-current and the energy-momentum tensor for this matter model are defined as

$$j^\mu = \rho u^\mu, \quad (2.20)$$

$$T^{\mu\nu} = \rho h u^\mu u^\nu + p g^{\mu\nu}, \quad (2.21)$$

where  $h = 1 + \varepsilon + \frac{p}{\rho}$  is the specific enthalpy of the fluid. As illustrated in Fig. 2.1,  $u^\mu$  can be scaled by a factor  $\frac{\alpha}{W}$ ,  $W = -u^\mu n_\mu$ , so that it shares the property (2.5), i.e. it maps points from  $\Sigma_t$  to  $\Sigma_{t+dt}$ . By comparing the proper times elapsing for the Eulerian observer  $-\delta\tau^2 = g_{\mu\nu}(\alpha n^\mu \delta t)(\alpha n^\nu \delta t)$  and the *Lagrangian observer* (i.e. an observer moving with the fluid)  $-\delta\tau_0^2 = g_{\mu\nu}(\frac{\alpha}{W} u^\mu \delta t)(\frac{\alpha}{W} u^\nu \delta t)$  in between the two slices  $\Sigma_t, \Sigma_{t+dt}$ , one finds

$$\delta\tau = W\delta\tau_0 = \alpha\delta t. \quad (2.22)$$

Because of this relation,  $W$  is called *Lorentz factor* referring to special relativity. It also explains the name of the lapse, since  $\alpha$  relates the lapse of proper time measured by the Eulerian observer and the coordinate time difference  $\delta t$ . It is also evident from Fig. 2.1, that the fluid displacement relative to this observer is given by the projection  $\gamma_{\nu}^{\mu} \frac{\alpha}{W} u^{\nu} \delta t$ . The observer dependent three-velocity of the fluid  $v^{\mu}$  is given by this displacement divided by the elapsed proper time, so that by means of (2.22)

$$v^{\mu} = \gamma_{\nu}^{\mu} \frac{1}{W} u^{\nu}. \quad (2.23)$$

Overall, we find the 3+1 decomposition of the fluid four-velocity:

$$u^{\mu} = W (n^{\mu} + v^{\mu}). \quad (2.24)$$

Hence, we reformulate the rest mass conservation law (2.1a)

$$\begin{aligned} \alpha \nabla_{\mu} j^{\mu} &= \alpha (\gamma_{\mu}^{\nu} - n_{\mu} n^{\nu}) \nabla_{\nu} \rho W (n^{\mu} + v^{\mu}) \\ &= \partial_t (\rho W) + D_i (\rho W [\alpha v^i - \beta^i]) + \rho W (D_i \beta^i - \alpha K) = 0, \end{aligned} \quad (2.25)$$

where we used eqs. (2.11-16). Although this is already a 3+1 decomposed version of the equation, it can be re-written such that its conservation properties are apparent again. With  $\partial_t \gamma_{ij} = \mathcal{L}_t \gamma_{ij}$  and (2.13), it is

$$D_i \beta^i - \alpha K = \frac{1}{2} \gamma^{ij} \partial_t \gamma_{ij}. \quad (2.26)$$

The following two textbook results can be derived by expanding the determinant and using the definition of the Christoffel symbol:

$$\frac{1}{\sqrt{\gamma}} \partial_t \sqrt{\gamma} = \frac{1}{2} \gamma^{ij} \partial_t \gamma_{ij}, \quad (2.27)$$

$$D_i v^i = \partial_i v^i + \frac{1}{2} v^i \gamma^{kl} \partial_i \gamma_{kl} = \frac{1}{\sqrt{\gamma}} \partial_i (\sqrt{\gamma} v^i). \quad (2.28)$$

Applying these building blocks on (2.25), we gain its conservation law form:

$$\partial_t (\sqrt{\gamma} \rho W) + \partial_i (\sqrt{\gamma} \rho W [\alpha v^i - \beta^i]) = \alpha \sqrt{\gamma} \nabla_{\mu} j^{\mu} = 0. \quad (2.29)$$

The conservative nature of this equation deserves much more attention. We return to this feature at the end of this section.

For the treatment of the energy-momentum conservation law (2.1b), we first

introduce the 3+1 decomposition of the energy-momentum tensor,

$$E = n^\mu n^\nu T_{\mu\nu} = \rho h W^2 - p, \quad (2.30a)$$

$$p_\alpha = -\gamma_\alpha^\mu n^\nu T_{\mu\nu} = \rho h W^2 v_\alpha, \quad (2.30b)$$

$$S_{\alpha\beta} = \gamma_\alpha^\mu \gamma_\beta^\nu T_{\mu\nu} = \rho h W^2 v_\alpha v_\beta + p \gamma_{\alpha\beta}, \quad (2.30c)$$

$$T_{\mu\nu} = E n_\mu n_\nu + p_\mu n_\nu + n_\mu p_\nu + S_{\mu\nu}, \quad (2.30d)$$

with  $E, p_\alpha$  and  $S_{\alpha\beta}$  being the energy density, momentum density and stress tensor observed by the Eulerian observer. Equation (2.1b) has a free index, so that it can be projected along and orthogonal to  $\Sigma_t$ . In fact, the former projection will lead to momentum conservation, while the latter results in energy conservation. Projecting onto  $\Sigma_t$  and using again the preliminary eqs. (2.11-15), (2.17) and (2.30) gives

$$\begin{aligned} \alpha \gamma_i^\nu \nabla_\mu T_\nu^\mu &= \alpha \gamma_i^\nu (\gamma_\mu^\beta - n_\mu n^\beta) \nabla_\beta (E n_\mu n_\nu + p_\mu n_\nu + n_\mu p_\nu + S_{\mu\nu}) \\ &= \partial_t p_i - \mathcal{L}_\beta p_i + D_j (\alpha S_i^j) + E D_i \alpha - \alpha p_i K \\ &= \partial_t p_i + D_j (\alpha S_i^j - p_i \beta^j) + p_i (D_j \beta^j - \alpha K) - p_j D_i \beta^j + E D_i \alpha \\ &= 0. \end{aligned} \quad (2.31)$$

As for the previous derivation, we use the trick (2.26) and the textbook relation

$$D_j S_i^j = \partial_j S_i^j + \frac{1}{\sqrt{\gamma}} \partial_j \sqrt{\gamma} S_i^j - \Gamma^k_{ij} S_k^j, \quad (2.32)$$

to find the balance law result for momentum conservation:

$$\alpha \sqrt{\gamma} \gamma_i^\nu \nabla_\mu T_\nu^\mu = \partial_t (\sqrt{\gamma} p_i) + \partial_j (\sqrt{\gamma} [\alpha S_i^j - p_i \beta^j]) - \frac{1}{2} S^{jk} \partial_i \gamma_{jk} - p_j \partial_i \beta^j + E \partial_i \alpha = 0. \quad (2.33)$$

Finally, we project (2.1b) onto  $n^\mu$  and by the same means as before, we derive

$$\begin{aligned} \alpha \sqrt{\gamma} n^\nu \nabla_\mu T_\nu^\mu &= \alpha \sqrt{\gamma} n^\nu \nabla_\mu (\gamma_\mu^\beta - n_\mu n^\beta) \nabla_\beta (E n_\mu n_\nu + p_\mu n_\nu + n_\mu p_\nu + S_{\mu\nu}) \\ &= \sqrt{\gamma} (\partial_t E + E [D_i \beta^i - \alpha K] + D_i [\alpha p^i - E \beta^i] + p^i D_i \alpha - \alpha S^{ij} K_{ij}) \\ &= \partial_t (\sqrt{\gamma} E) + \partial_i (\sqrt{\gamma} [\alpha p^i - E \beta^i]) + \sqrt{\gamma} p^i \partial_i \alpha + \alpha \sqrt{\gamma} S^{ij} K_{ij}. \end{aligned} \quad (2.34)$$

In this section, we obtained a system of the five equations (2.29), (2.33), (2.34) for the matter variables. As in [Banyuls et al., 1997], we can summarize these results as a fundamental balance law

$$\partial_t u + \partial_i f^i(u) = s(u), \quad (2.35)$$

which has a deeper physical meaning. Integrating this equation over a certain spatial

volume  $\Omega$ , the average of  $u$  in  $\Omega$  changes in time only due to out/inflow through  $\partial\Omega$  and due to sources and sinks in  $\Omega$ . Although  $u$  is only conserved if  $s(u) = 0$ , the components of  $u$  are usually called *conserved variables*. In contrast, the formerly introduced fluid variables  $\rho$ ,  $\varepsilon$ ,  $p$  and  $v^i$  are called *primitive variables*. The conserved variables can be expressed in terms of the primitive variables (see eqs. (2.30a-c)). Since there is no analytical inversion of this mapping, the primitives are usually recovered by a Newton-Raphson procedure. We give a more detailed explanation on this recovery in Appendix A. Note, that  $W$  is not an independent variable. From  $W = -u^\mu n_\mu$ , (2.24) and the normalization of  $n^\mu$  and  $u^\mu$  we conclude  $W = (1 - v^i v_i)^{-\frac{1}{2}}$ . The numerical methods that are explained in chapter 3 intensively use this specific form of (2.35). They are designed in a way, such that conserved variables are also conserved on a discrete numerical grid. In (2.35), we call  $f^i(u)$  *fluxes* and  $s(u)$  *sources*. In the case of GRHD, the sources also depend on  $\alpha$ ,  $\gamma_{ij}$  or  $K_{ij}$  besides  $u$ . For what we derived in this section, we can combine

$$u = \sqrt{\gamma} \begin{pmatrix} D \\ p_j \\ \tau \end{pmatrix} = \sqrt{\gamma} \begin{pmatrix} \rho W \\ \rho h W^2 v_j \\ \rho h W^2 - p - D \end{pmatrix}, \quad f^i = \sqrt{\gamma} \begin{pmatrix} D(\alpha v^i - \beta^i) \\ p_j(\alpha v^i - \beta^i) + p \delta_j^i \\ \tau(\alpha v^i - \beta^i) + p v^i \end{pmatrix}, \quad (2.36)$$

where we use  $E - D$  as the fifth conserved variable instead of  $E$  alone. This is motivated by numerical advantages [Rezzolla and Zanotti, 2013] and was originally proposed in [Banyuls et al., 1997]. The sources can be solely expressed in terms of four-tensors and the lapse:

$$s = \sqrt{-g} \begin{pmatrix} 0 \\ T^{\mu\nu} (\partial_\mu g_{\nu j} - \Gamma^\delta_{\mu\nu} g_{\delta j}) \\ \alpha (T^{\mu 0} \partial_\mu \ln \alpha - T^{\mu\nu} \Gamma^0_{\mu\nu}) \end{pmatrix}. \quad (2.37)$$

The derivation of these source terms from those in eqs. (2.31,34) is not obvious. We followed two major steps: (a) Expressing  $E$ ,  $p_i$  and  $S_{ij}$  as tensor components of the energy-momentum tensor, e.g.  $E = \alpha^2 T^{00}$ ; (b) Expressing  $\alpha$ ,  $\beta^i$  and  $\gamma_{ij}$  as components of  $g_{\mu\nu}$ , using (2.9).

The system of eqs. (2.35-37) is hyperbolic, i.e. the dynamics of the corresponding quasi-linear system can be decomposed in advective transport processes with *characteristic speeds*. These provide very useful information, e.g. the maximum propagation speed of the fields along a given direction. We will use these information to construct numerical fluxes for our schemes (see Sec. 3.2). The characteristic speeds are the eigenvalues of the matrix  $\frac{\partial(s_i f^i)}{\partial u}$  where the vector  $s^i$  specifies the direction of



propagation. These eigenvalues are given by [Banyuls et al., 1997]:

$$\begin{aligned}\lambda_0 &= \alpha s_i v^i - s_i \beta^i \quad (\text{triple}), \\ \lambda_{\pm} &= \frac{\alpha}{1 - v^2 c_s^2} \left[ s_i v^i (1 - c_s^2) \pm c_s \sqrt{(1 - v^2) [s^2 (1 - v^2 c_s^2) - (s_i v^i)^2 (1 - c_s^2)]} \right] - s_i \beta^i.\end{aligned}\tag{2.38}$$

Here,  $c_s$  denotes the *speed of sound* defined by  $c_s^2 = \left( \frac{\partial p}{\partial \rho} + \frac{p}{\rho^2} \frac{\partial p}{\partial \varepsilon} \right)$ .

## 2.3 Recasting the Einstein field equations

After the treatment of the matter equations in Sec. 2.2, we want to discuss the field equations of GR (2.1c). In principle, we could simply proceed as in the previous section and apply the 3+1 formalism. This leads to the ADM equations [Arnowitt et al., 1962]. By further modifications (e.g. introducing a conformal decomposition of the spatial metric) the BSSN system [Baumgarte and Shapiro, 1998; Shibata and Nakamura, 1995] can be derived. Together with a finite-differencing scheme it yields the standard framework used in a majority of numerical relativity codes. In this work, we follow another strategy which is closely linked to the Ph.D. project of Andreas Weyhausen [Weyhausen, 2014; Hilditch et al., 2016]. Instead of BSSN, we use the GHG system as presented in [Lindblom et al., 2006] to evolve the spacetime quantities. Generalized harmonic coordinates proved to be very useful in numerical simulations. In particular, they allow a reliable binary black hole inspiral evolution up to the merger [Pretorius, 2005; Pretorius, 2006]. In this section, we summarize the main derivation steps of the GHG system and its properties. We again refer the reader to [Lindblom et al., 2006; Weyhausen, 2014] for more details.

The foundation of the GHG system is a certain representation of the Ricci tensor, that is

$$R_{\mu\nu} = -\frac{1}{2} g^{\alpha\beta} \partial_\alpha \partial_\beta g_{\mu\nu} + \nabla_{(\mu} \Gamma_{\nu)} + g^{\alpha\beta} g^{\delta\epsilon} ([\partial_\delta g_{\alpha\mu}] [\partial_\epsilon g_{\beta\nu}] - \Gamma_{\mu\alpha\delta} \Gamma_{\nu\beta\epsilon}), \tag{2.39}$$

with  $\Gamma_{\alpha\mu\nu} = g_{\alpha\beta} \Gamma^{\beta}_{\mu\nu}$  and  $\Gamma_\alpha = g^{\mu\nu} \Gamma_{\alpha\mu\nu}$ . Although (2.39) is mentioned and discussed throughout the GHG literature, we give a detailed derivation in Appendix B. From this formulation it is obvious, that if we choose coordinates satisfying

$$g_{\mu\nu} \nabla^\alpha \nabla_\alpha x^\nu = -\Gamma_\mu = H_\mu(x, g), \tag{2.40}$$

where  $H$  is a function of the coordinates and the metric (but not its derivatives), the Einstein equations are manifestly hyperbolic. Hence, the principal part of (2.39) is just that of a scalar wave equation. The given function  $H_\mu$  is usually called *Gauge source function*. With a 3+1 decomposition,  $H_\mu$  can be related to the derivatives of

lapse and shift. For all our tests, we chose the Gauge source function constant in time, such that  $\partial_t \alpha = \partial_t \beta^i = 0$  is fulfilled on the initial slice. Although this is sufficient for our purposes, it should be mentioned that there are more sophisticated ways to chose  $H_\mu$ , see e.g. [Lindblom and Szilágyi, 2009]. In numerical evolutions (2.40) can be violated, which is why we view it as the *Harmonic constraint*  $\mathcal{C}_\mu = H_\mu + \Gamma_\mu$ . Furthermore, the Einstein equations (2.1c) are modified:

$$R_{\mu\nu} - \nabla_{(\mu} \mathcal{C}_{\nu)} + \gamma_4 \Gamma_{\mu\nu}^\alpha \mathcal{C}_\alpha - \frac{1}{2} \gamma_5 g_{\mu\nu} \Gamma_\alpha C^\alpha + \gamma_0 \left( n_{(\mu} \mathcal{C}_{\nu)} - \frac{1}{2} g_{\mu\nu} n^\alpha \mathcal{C}_\alpha \right) = 8\pi \left( T_{\mu\nu} - \frac{1}{2} T g_{\mu\nu} \right). \quad (2.41)$$

This system is still hyperbolic and solutions satisfying  $\mathcal{C}_\mu = 0$  will be solutions of the original field equations of GR. An analysis of the constraint evolution system gives two additional insights: 1) The evolution system is self-consistent, i.e. if  $\mathcal{C}_\mu = \partial_t \mathcal{C}_\mu = 0$  in a domain  $\Omega$ , the constraints also vanish in the domain of dependence of  $\Omega$  [Lindblom et al., 2006]; 2) The term proportional to  $\gamma_0$  causes a damping of high frequency constraint violations. Although this has been proved in [Gundlach et al., 2005] (performing a mode analysis on a linearization of GR), it is not obvious that these arguments still hold for a general spacetime containing matter. However, in practice  $\gamma_0 > 0$  leads to constraint damping in all our tests. The terms including  $\gamma_4$  and  $\gamma_5$  were incorporated into the GHG formulation in [Hilditch et al., 2016] to simplify the constraint subsystem. Unless otherwise stated, we use  $\gamma_4 = \gamma_5 = \frac{1}{2}$  in this work.

The second order PDE system (2.41) can be reduced to be first order in time and space by introducing the reduction variables

$$\Pi_{\mu\nu} = -n^\alpha \partial_\alpha g_{\mu\nu}, \quad \Phi_{i\mu\nu} = \partial_i g_{\mu\nu}. \quad (2.42)$$

However, this adds two new constraints to the system, namely the *Reduction constraint*  $\mathcal{C}_{i\mu\nu} = \Phi_{i\mu\nu} - \partial_i g_{\mu\nu}$ , and the *Ordering constraint*  $\mathcal{C}_{ij\mu\nu} = \partial_{[i} \Phi_{j]\mu\nu}$ . We do not want to give all details of the first order reduction calculation of (2.41). The main step is expressing the second order part  $g^{\alpha\beta} \partial_\alpha \partial_\beta g_{\mu\nu}$  in terms of the reduction variables. Again using the 3+1 formalism (2.9), we can verify the intermediate results

$$\begin{aligned} \alpha n^\mu n^\nu \Phi_{i\mu\nu} &= -2\partial_i \alpha, & \alpha^2 n^\mu n^\nu \Pi_{\mu\nu} &= 2(\partial_t \alpha - \beta^i \partial_i \alpha) \\ \alpha n^\mu \gamma^{j\nu} \Phi_{i\mu\nu} &= \partial_i \beta^j, & \alpha^2 n^\mu \gamma^{j\nu} \Pi_{\mu\nu} &= -(\partial_t \beta^j - \beta^i \partial_i \beta^j). \end{aligned} \quad (2.43)$$

With (2.6)  $\Pi_{\mu\nu}$  can be linked to the time derivative of the metric. Finally, one finds

the first order GHG evolution system as implemented in our code:

$$\partial_t g_{\mu\nu} = \beta^i \partial_i g_{\mu\nu} - \alpha \Pi_{\mu\nu} + \gamma_1 \beta^i \mathcal{C}_{i\mu\nu} \quad (2.44a)$$

$$\partial_t \Phi_{i\mu\nu} = \beta^j \partial_j \Phi_{i\mu\nu} - \alpha \partial_i \Pi_{\mu\nu} + \gamma_2 \alpha \mathcal{C}_{i\mu\nu} + \frac{1}{2} \alpha n^\alpha n^\beta \Phi_{i\alpha\beta} \Pi_{\mu\nu} + \alpha \gamma^{jk} n^\alpha \Phi_{ij\alpha} \Phi_{k\mu\nu} \quad (2.44b)$$

$$\begin{aligned} \partial_t \Pi_{\mu\nu} = & \beta^i \partial_i \Pi_{\mu\nu} - \alpha \gamma^{ij} \partial_i \Phi_{j\mu\nu} + \gamma_1 \gamma_2 \beta^i \mathcal{C}_{i\mu\nu} - \frac{1}{2} \alpha n^\alpha n^\beta \Pi_{\alpha\beta} \Pi_{\mu\nu} - \alpha n^\alpha \gamma^{ij} \Pi_{\alpha i} \Phi_{j\mu\nu} \\ & + 2\alpha g^{\alpha\beta} \left( \gamma^{ij} \Phi_{i\alpha\mu} \Phi_{j\beta\nu} - \Pi_{\alpha\mu} \Pi_{\beta\nu} - g^{\delta\epsilon} \Gamma_{\mu\alpha\delta} \Gamma_{\nu\beta\epsilon} \right) + \alpha \gamma_0 \left( 2n_{(\mu} \mathcal{C}_{\nu)} - g_{\mu\nu} n^\alpha \mathcal{C}_\alpha \right) \\ & - 2\alpha \left( 8\pi \left[ T_{\mu\nu} - \frac{1}{2} T g_{\mu\nu} \right] + \nabla_{(\mu} H_{\nu)} + \gamma_4 \Gamma^\alpha{}_{\mu\nu} \mathcal{C}_\alpha - \frac{1}{2} \gamma_5 g_{\mu\nu} \Gamma^\alpha \mathcal{C}_\alpha \right), \quad (2.44c) \end{aligned}$$

where extra terms with the coefficients  $\gamma_1 = -1$ ,  $\gamma_2 > 0$  have been added in order to damp the reduction constraint  $\mathcal{C}_{i\mu\nu}$  and to make the system linearly degenerate [Lindblom et al., 2006]. In this first order version, the Christoffel symbols have to be viewed as combinations of the reduction variables

$$\Gamma_{\mu\nu\alpha} = \gamma^i{}_{(\nu} \Phi_{i|\alpha)\mu} - \frac{1}{2} \gamma_\mu^i \Phi_{i\nu\alpha} + n_{(\nu} \Pi_{\alpha)\mu} - \frac{1}{2} n_\mu \Pi_{\nu\alpha}. \quad (2.45)$$

Besides to the evolution system, reasonable outer boundary conditions are needed for a reliable numerical evolution. This topic could be a whole lecture on its own, and we refer the reader to [Hilditch et al., 2016; Rinne et al., 2007] for a more detailed description of our implementation. The fundamental idea is to impose boundary conditions on the derivatives  $s^i \partial_i u_{\mu\nu}^{\hat{\alpha}}$ , where  $s^i$  is normal to the boundary and  $u_{\mu\nu}^{\hat{\alpha}}$  is a characteristic variable. Just as for the evolved variables, a characteristic decomposition of the constraint evolution system can be derived. It turns out, that the characteristic fields for the constraint evolution system are linked with the former mentioned derivatives  $s^i \partial_i u_{\mu\nu}^{\hat{\alpha}}$  [Lindblom et al., 2006]. Exploiting this fact, it is possible to find boundary conditions, such that the incoming characteristic constraint fields vanish, i.e. no constraint violations originate from the outer boundary. However, additionally to the constraint fields two physical and four gauge degrees of freedom have to be fixed at the boundary. For the former, this is done by imposing the gravitational wave scalar  $\Psi_4$ . This is defined as a contraction of the Riemann tensor, which in turn contains combinations of  $\partial_i u_{\mu\nu}^{\hat{\alpha}}$ . The gauge degrees of freedom are treated according to [Rinne et al., 2007; Hilditch and Ruiz, 2016]. We implemented all these boundary conditions using the Bjørhus method [Bjørhus, 1995].

## 2.4 The dual foliation formalism

One key feature of the theory of general relativity is its invariance under coordinate transformations. In various situations, it can be desirable to employ two different coordinate systems  $x^\mu = (t, x^i)$ ,  $X^\mu = (T, X^i)$  in order to exploit the advantages

**Table 2.1:** Two foliations and the corresponding 3+1 quantities

coordinates	slices	normal vector	lapse	shift	time vector	induced metric
$(t, x^i)$	$t = \text{const}$	$n^\mu$	$\alpha$	$\beta^i$	$t^\mu$	$\gamma_{ij}$
$(T, X^i)$	$T = \text{const}$	$N^\mu$	$A$	$B^i$	$T^\mu$	${}^{(N)}\gamma_{ij}$

of both. This means that two foliations  $\Sigma_t, \Sigma_T$  are introduced with  $t = \text{const}$  and  $T = \text{const}$ , respectively. The 3+1 quantities as explained in Sec. 2.1 can be defined for both foliations (see table 2.1).

In what follows, we will refer to these two frames as the *lower case* and the *upper case* frame. In [Hilditch, 2015], which is the pioneer work on the DF formalism, the connection between these sets of variables was figured out. Here we will follow the route of this paper to derive some of its main results.

In sec 2.2 we introduced the Lagrangian observer traveling along with the fluid. This situation is identical to what we consider now, except that  $N^\mu$  is the four-velocity of an arbitrary family of observers. In perfect analogy, we define the Lorentz factor  $W = -N^\mu n_\mu$ , so that the proper times of the two observers  $\delta\tau$  and  ${}^{(N)}\delta\tau$  can be related in the fashion of eq. (2.22), yielding  $\delta\tau = W {}^{(N)}\delta\tau$ . This leads to the insight, that  $W$  links the two lapse functions  $\alpha, A$ :

$$\alpha\delta t = \delta\tau = W {}^{(N)}\delta\tau = W A\delta t \quad \Rightarrow \quad \alpha = W A. \quad (2.46)$$

We again find the relation

$$N^\mu = W (n^\mu + v^\mu). \quad (2.47a)$$

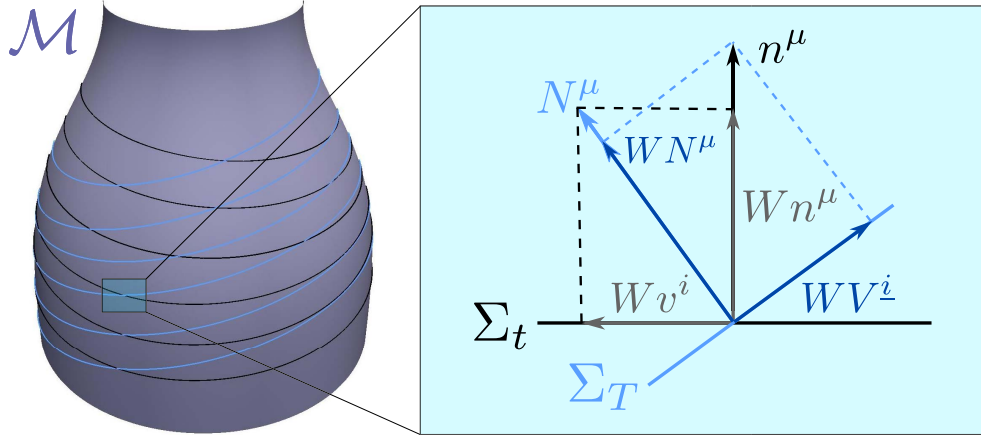
Here  $v^\mu$  has no longer the meaning of a fluid velocity, but instead describes the relative velocity of the two observers. Vice versa, switching the point of view to the other observer, we find

$$n^\mu = W (N^\mu + V^\mu). \quad (2.47b)$$

In the above equations, we call  $v^\mu$  the lower case and  $V^\mu$  the upper case *boost vector*. This reciprocity of the two coordinate systems is also evident from Fig 2.2. The boost vectors can be used to recover an uppercase spatial vector  $S^\mu N_\mu = 0$  from only its projection onto the lower case slice  $\Sigma_t$ :

$$S^\mu = \gamma_\nu^\alpha S^\nu (\delta_\alpha^\mu + n^\mu v_\alpha). \quad (2.49)$$

It is straightforward to generalize this result for arbitrary uppercase spatial tensors. Furthermore, we can now 3+1 decompose the Jacobian which is used to transform



**Figure 2.2:** Dual foliation formalism: The manifold  $\mathcal{M}$  (purple) is now sliced by two sets of hypersurfaces  $\Sigma_t, \Sigma_T$  where  $t = \text{const}$  (black) and  $T = \text{const}$  (blue). The infinitesimal vicinity of a point in  $\Sigma_t \cap \Sigma_T$  is shown on the right. The meaning of the DF quantities Lorentz factor  $W$  and boost vectors  $v^i, V^i$  is depicted.

between the two coordinate systems. With the insights of Sec. 2.1 and (2.47) we find

$$J^0_{\underline{\mu}} = \nabla_{\underline{\mu}} t = -\frac{1}{\alpha} n_{\underline{\mu}} = -\frac{W}{\alpha} (N_{\underline{\mu}} + V_{\underline{\mu}}) \quad (2.50)$$

$$J^{\underline{\mu}}_{\underline{\nu}} = T^{\underline{\nu}} J^{\underline{\mu}}_{\underline{\nu}} = AN^{\underline{\mu}} + B^{\underline{\nu}} J^{\underline{\mu}}_{\underline{\nu}} = AW(n^{\underline{\mu}} + v^{\underline{\mu}}) + B^{\underline{\nu}} J^{\underline{\mu}}_{\underline{\nu}}. \quad (2.51)$$

Introducing a symbol for the spatial part of the Jacobian  $\Phi^i_{\underline{j}} \equiv J^i_{\underline{j}}$  and using that  $V_{\underline{0}} = B^i V_{\underline{i}}$  because  $V^{\underline{\mu}}$  is spatial with respect to the upper case foliation, a concluding representation of the Jacobian in terms of 3+1 quantities can be found:

$$J^{\underline{\mu}}_{\underline{\nu}} = \begin{pmatrix} \frac{W}{\alpha} (A - B^i V_{\underline{i}}) & -\frac{W}{\alpha} V_{\underline{j}} \\ AW \left( v^i - \frac{\beta^i}{\alpha} \right) + B^j \Phi^i_{\underline{j}} & \Phi^i_{\underline{j}} \end{pmatrix}. \quad (2.52)$$

A major result of the first DF work [Hilditch, 2015] is a transformation rule for first order PDE systems to switch between the foliations. Assuming, that using the upper case foliation yields a system of the form

$$\partial_T \mathbf{u} = (\mathbf{A} \mathbf{A}^i + B^i \mathbf{1}) \partial_i \mathbf{u} + \mathbf{A} \mathbf{S}, \quad (2.53)$$

we can use the Jacobian (2.52) and the lapse relation (2.46) to find the system in the lowercase coordinates:

$$\partial_T \mathbf{u} = J^{\underline{\mu}}_{\underline{0}} \partial_{\underline{\mu}} \mathbf{u} = \frac{W}{\alpha} (A - B^i V_{\underline{i}}) \partial_t \mathbf{u} + \left( AW \left( v^i - \frac{\beta^i}{\alpha} \right) + B^j \Phi^i_{\underline{j}} \right) \partial_i \mathbf{u}, \quad (2.54)$$

$$\partial_i \mathbf{u} = J^{\underline{\mu}}_{\underline{i}} \partial_{\underline{\mu}} \mathbf{u} = -\frac{W}{\alpha} V_{\underline{i}} \partial_t \mathbf{u} + \Phi^j_{\underline{i}} \partial_j \mathbf{u}, \quad (2.55)$$

$$\Rightarrow (1 + \mathbf{A}^i V_{\underline{i}}) \partial_t \mathbf{u} = \frac{\alpha}{W} \left( \mathbf{A}^i \Phi^j_{\underline{i}} - W \left( v^j - \frac{\beta^j}{\alpha} \right) \right) \partial_j \mathbf{u} + \frac{\alpha}{W} \mathbf{S}. \quad (2.56)$$

Note, that in the final transformed PDE system (2.56), the state vector  $\mathbf{u}$  was not affected by the transformation. In particular,  $\mathbf{u}$  may contain tensor components with respect to the upper case basis, even though lower case coordinates are used for the evolution. This decoupling of coordinates and tensor basis is the unique feature of the DF formalism compared to standard methods in NR. In the original work, (2.56) is used to analyze certain properties of the PDE system, such as symmetric hyperbolicity. In [Bug2], we applied the transformation to the wave equation in Minkowski spacetime and introduced compactified hyperboloidal slices as a second foliation. The tests and results of this certain work are the topic of chapter 6.

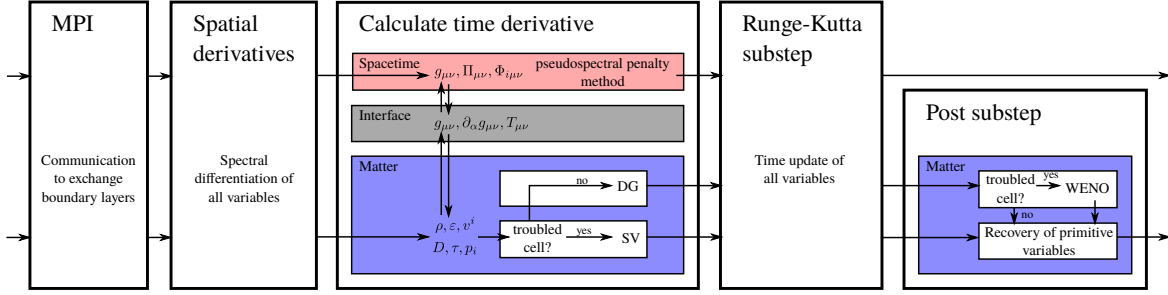
# Chapter 3

## Numerical methods

This chapter gives an overview on the numerical methods that are relevant for this work. We first introduce the existing infrastructure of the `bamps` code, its grid layout and symmetry treatment in section 3.1. Moreover, a summary is given on pseudospectral and penalty methods, which are used for the numerical solution of the GHG system (2.44) as presented in [Weyhausen, 2014; Hilditch et al., 2016]. Section 3.2 focuses on the key aspect of this work, the Runge-Kutta Discontinuous Galerkin method [Cockburn and Shu, 1989] as applied to the GRHD eqs. (2.35-37). We give a detailed description of the method and explain its application on three-dimensional curvilinear grids. Since shocks are expected in a general setup, the code has to be equipped with a shock capturing technique. In the scope of this work we followed two different shock capturing strategies, the weighted-essentially-non-oscillatory (WENO) methodology [Liu and Osher, 1994; Jiang and Shu, 1996; Qiu and Shu, 2005; Zhao and Tang, 2013] [Bug1] and the local finite volume (or spectral volume) treatment [Wang, 2002; Radice and Rezzolla, 2011]. Both shock capturing methods are subject of section 3.3. Figure 3.1 gives a schematic overview on a single time evolution substep (i.e. a Runge-Kutta substep) as implemented in `bamps`. The different constituents are subject of this chapter.

### 3.1 Pseudospectral methods and the `bamps` code

The usage of spectral decompositions dates back to the days of Fourier’s analytical investigations [Fourier, 1822]. With the development of the fast Fourier transformation [Cooley and Tukey, 1965], the popularity of spectral methods as a numerical approach to solve PDEs significantly increased in the 1970s. Since then, spectral methods were very successfully applied in many fields of physics [Boyd, 1989]. Of particular interest regarding this work was the numerical solution of the GHG system with a pseudospectral (or collocation) method implemented with the `SpEC` code [SpEC]. Just as `SpEC`, the `bamps` code [Hilditch et al., 2016] in its



**Figure 3.1:** A time evolution substep in the `bamps` code. MPI [MPI] is used to communicate grid data from one parallel process to another. The spatial derivatives are calculated via a spectral approximation (see Sec. 3.1). For the spacetime variables, the time derivatives are computed from (2.44) with a pseudospectral method (see Sec. 3.1). The matter balance law (2.35) is usually evolved by a DG method (see Sec. 3.2). Both sets of variables are stored in an interface array, whose components have a globally defined meaning. This concept was mainly developed by Hannes Rüter. If the matter variables undercut a threshold, the cell is marked as troubled and the spectral volume method (see Sec. 3.3.2) is employed instead of DG. After the actual substep, the solution is checked by another troubled cell indicator. If necessary, a WENO limiting procedure is applied (see Sec. 3.3.1). In a final step, the primitive variables are reconstructed from the conservative variables (see Appendix A).

initial configuration is based on a pseudospectral method. The code is written in the C programming language, making intensive use of structures towards an object-oriented fashion. The technical branch of the code, including e.g. grid management, input and output, parallel processing [MPI] and fundamental algebra is separated from the actual physics project modules. The various physics projects each contain their respective first order evolution equations, boundary conditions and formulas for miscellaneous analysis quantities. To simplify the input of these equations, a Mathematica script is used to convert tensor expressions into C code. `bamps` has been used for gravitational wave collapse simulations, giving impressive results both in accuracy and efficiency [Hilditch et al., 2017].

### Grid structure

Now focusing on the term “pseudospectral”, we want to emphasize that all types of polynomial spectral methods (e.g. pseudospectral, Galerkin, Tau) rest on the same fundamental considerations [Hesthaven et al., 2007]: Starting from

$$\partial_t u = Lu \quad (3.1)$$

where  $L$  is a differential operator acting on the coordinates  $\xi$ , we approximate the solution as an expansion of polynomials  $\phi_i$ :

$$u(\xi, t) \approx u_n(\xi, t) = \sum_{k=0}^N a_k(t) \phi_k(\xi). \quad (3.2)$$



Furthermore, the residual

$$R_n = \partial_t u_n - Lu_n \quad (3.3)$$

is defined and should of course vanish in the continuum limit. It is now the specific implementation of this condition for  $R_n$ , that distinguishes between the several spectral methods. In the case of a *pseudospectral method*, the residual is enforced to vanish exactly at certain points in the domain,

$$\partial_t u_n(\xi_i, t) = Lu_n(\xi_i, t), \quad (3.4)$$

where  $\xi_i$  are the so called *collocation points* or *nodes*. In bamps as used for this work, these  $N + 1$  nodes are given by the Legendre-Gauss-Lobatto points,

$$\xi_i \in \left\{ \xi \mid (1 - \xi^2) \frac{dP_N}{d\xi}(\xi) = 0 \right\}, \quad (3.5)$$

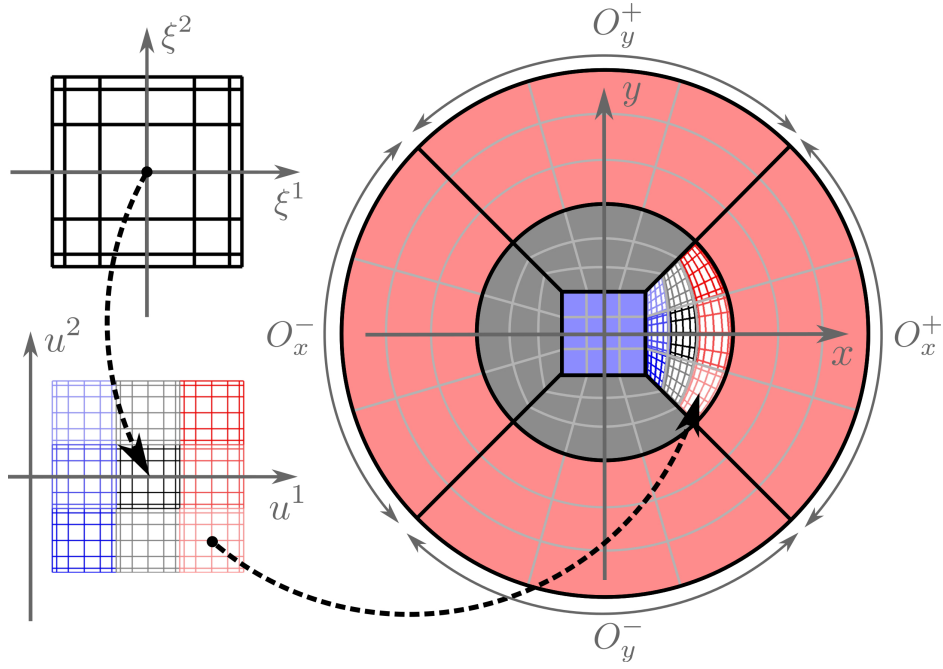
where  $P_N$  is the  $N$ -th Legendre polynomial. Note, that the lower index in  $\xi_i$  enumerates the nodes and should not be confused with the upper index in  $\xi^i$ , which is a coordinate index. Using these points in each dimension spans the cubic interval  $[-1, 1]^3$ , which is the fundamental reference grid in bamps (see top left part of Fig. 3.2). From this reference grid, a rectangular box patch covering the interval  $[u_{\min}^1, u_{\max}^1] \times [u_{\min}^2, u_{\max}^2] \times [u_{\min}^3, u_{\max}^3]$  with local patch coordinates  $u^i$  is constructed. To increase resolution, these patches can be divided into  $N_1 \times N_2 \times N_3$  subpatches. Each subpatch is a simple linear mapping of the fundamental reference grid (see bottom left part of Fig. 3.2):

$$u^i = \frac{\bar{u}_{\max}^i - \bar{u}_{\min}^i}{2} \xi^i + \frac{\bar{u}_{\max}^i + \bar{u}_{\min}^i}{2},$$

$$\bar{u}_{\max}^i = u_{\min}^i + \frac{k_i}{N_i} (u_{\max}^i - u_{\min}^i), \quad \bar{u}_{\min}^i = u_{\min}^i + \frac{k_i - 1}{N_i} (u_{\max}^i - u_{\min}^i), \quad k_i \in [1, N_i]. \quad (3.6)$$

From the patches, the grid is built in global Cartesian coordinates  $x^i = (x, y, z)$ . In bamps, the grid layout is a *cubed-ball* grid, i.e. it consists of three types of coordinate patches: i) A cube which is centered around the origin; ii) Transition shells that transfer the inner cube to a spherical shell; iii) Outer shells, that extend the grid radially to its outer boundary. The full grid configuration is depicted on the right part of Fig. 3.2. It includes the origin as a perfectly regular point and has a spherical outer boundary, which is desirable e.g. for the formulation of boundary conditions and for gravitational wave extraction.

For the central cube, the box patch is just mapped one-to-one into Cartesian



**Figure 3.2:** The bumps grid layout for the  $x$ - $y$ -plane. Top left: The fundamental reference grid consisting of Legendre-Gauss-Lobatto grid points  $\xi^i \in [-1, 1]$ . Bottom left: A grid patch consisting of several subpatches. Each subpatch is a mapping from the fundamental reference grid. Right: The cubed-ball grid layout, built from a central cube (blue), transition shells (gray) and outer shells (red). Each part of this grid is a mapping from the cubic grid patch. The shell regions are first mapped to a master patch, which is oriented in positive  $x$  direction ( $O_x^+$ ). All other orientations, as indicated in the figure, are achieved by 90 degree rotations of this master patch.

coordinates:

$$x = u^1, \quad y = u^2, \quad z = u^3. \quad (3.7)$$

The mapping of a box patch to a spherical shell in bumps is more complicated and relies on the cubed-sphere transformation first presented in [Ronchi et al., 1996] and used in a multipatch fashion in [Lehner et al., 2005; Pollney et al., 2011]. The concrete coordinate transformation is given by

$$\bar{x} = \frac{u^1}{s}, \quad \bar{y} = \frac{u^1 u^2}{s}, \quad \bar{z} = \frac{u^1 u^3}{s}, \quad (3.8)$$

where for the outer shells

$$s = \sqrt{1 + (u^2)^2 + (u^3)^2} \quad (3.9)$$

and for the transition shells

$$s = \sqrt{\frac{1 + 2\lambda}{1 + \lambda((u^2)^2 + (u^3)^2)}}, \quad \lambda = \frac{(u^1)^2 - (u_{\min}^1)^2}{(u_{\max}^1)^2 - (u_{\min}^1)^2}. \quad (3.10)$$

With this mapping, the whole domain is not yet covered. Solely the patches with positive  $x$  orientation  $O_x^+$  (see Fig 3.2) are reached by setting  $x = \bar{x}, y = \bar{y}, z = \bar{z}$ . All other patches can be generated by 90 degree rotations of these  $O_x^+$  patches:

$$\begin{aligned} O_x^- : x &= -\bar{x}, & y &= -\bar{y}, & z &= +\bar{z} \\ O_y^+ : x &= +\bar{z}, & y &= +\bar{x}, & z &= +\bar{y}, & O_y^- : x &= +\bar{z}, & y &= -\bar{x}, & z &= -\bar{y} \\ O_z^+ : x &= +\bar{y}, & y &= +\bar{z}, & z &= +\bar{x}, & O_z^- : x &= -\bar{y}, & y &= +\bar{z}, & z &= -\bar{x}. \end{aligned} \quad (3.11)$$

This last building block completes the coordinate transformation as used in *bamps*. Accordingly, the Jacobian for this mapping consists of three parts:

$$\frac{\partial x^i}{\partial \xi^j} = \frac{\partial x^i}{\partial \bar{x}^l} \frac{\partial \bar{x}^l}{\partial u^k} \frac{\partial u^k}{\partial \xi^j}. \quad (3.12)$$

The right part is the scaling from the fundamental reference grid to the box patch, the middle part is the complicated mapping from the box patch to a curved patch and the left part is the final permutation (and eventually negation) of Cartesian coordinates, in order to rotate the patch in place.

### Numerical differentiation

To build a numerical scheme from (3.4), an approximation for spatial derivatives of  $u$  is needed. A simple way to derive this, is exploiting the properties of the *Lagrange polynomials*

$$\ell_k(\xi) = \prod_{\substack{l=0 \\ l \neq k}}^N \frac{\xi - \xi_l}{\xi_k - \xi_l}, \quad (3.13)$$

used for the expansion (3.2), i.e.  $\phi_k \equiv \ell_k$ . For each collocation point  $\xi_i$ , one corresponding Lagrange polynomial  $\ell_i$  can be defined with the properties  $\ell_i(\xi_i) = 1$  and  $\ell_i(\xi_j) = 0$  for  $i \neq j$ . This trivially leads to the fact, that the expansion coefficients  $a_k$  in (3.2) are identical with the values at the nodes:

$$u_n(\xi, t) = \sum_{k=0}^N u_n(\xi_k, t) \ell_k(\xi). \quad (3.14)$$

Using (3.13), it is straightforward to verify for the derivatives of  $\ell_k$

$$D_{ik} := \frac{\partial \ell_k}{\partial \xi}(\xi_i) = \begin{cases} \sum_{\substack{m=0 \\ m \neq i}}^N (\xi_i - \xi_m)^{-1}, & \text{for } i = k \\ \left( \prod_{\substack{j=0 \\ j \neq k}}^N (\xi_k - \xi_j) \right)^{-1} \prod_{\substack{m=0 \\ m \neq k \\ m \neq i}}^N (\xi_i - \xi_m), & \text{for } i \neq k \end{cases}. \quad (3.15)$$

With the spectral ansatz (3.2) and the property (3.14), we see that  $D_{ik}$  is a derivative matrix, that maps function values of the approximation at the nodes to its spatial derivative at the same nodes:

$$\frac{\partial u_n}{\partial \xi}(\xi_i) = \sum_{k=0}^N u_n(\xi_k, t) \frac{\partial \ell_k}{\partial \xi}(\xi_i) = \sum_{k=0}^N D_{ik} u_n(\xi_k, t). \quad (3.16)$$

In the pseudospectral code, the numerical differentiation of a function is nothing more than a matrix multiplication, which can be performed by optimized libraries, as BLAS [BLAS]. In *bamps*, a subtlety in the calculation of  $D_{ik}$  is used to make the code less sensitive to rounding errors. Instead of using the  $D_{ii}$  result from (3.15) directly, it is constructed as the negative sum of the off-diagonal elements [Baltensperger and Trummer, 2003]

$$D_{ii} = - \sum_{\substack{k=0 \\ k \neq i}}^N D_{ik}, \quad (3.17)$$

such that the derivative of a constant function is numerically exactly zero. This differentiation is always done on the fundamental reference grid. In the multidimensional case, the derivative matrix is applied to each direction, i.e.

$$\frac{\partial u_n}{\partial \xi^2}(\xi_i^1, \xi_j^2, \xi_k^3, t) = \sum_{l=0}^N D_{jl} u_n(\xi_i^1, \xi_l^2, \xi_k^3, t), \quad (3.18)$$

for derivatives with respect to  $\xi^2$ . To get the derivatives with respect to Cartesian coordinates, which usually appear in the evolution equations, we have to multiply with the appropriate Jacobian (3.12).

### Time stepping

Now that spatial derivatives are available, the right hand side of (3.4) can be built and therefore the time derivatives are known. To integrate forward in time, a standard 4th order Runge-Kutta scheme is used. Unless otherwise stated, we use a time-step  $\Delta t = 0.25\Delta x$ , where  $\Delta x$  is the minimal Cartesian distance of two points in the grid.

### Penalty method for patching boundaries

All ingredients for the numerical evolution of a single grid patch with the pseudospectral method have been presented. Still, it is a central question, how the patches interact with each other. In *bamps*, the penalty method [Taylor et al., 2010; Hesthaven et al., 2007] is employed to connect the grid patches properly. This patching method consists of two fundamental ideas: i) Boundary conditions are imposed for incoming characteristic variables  $u_-$ . Therefore, the evolution system is transformed to the

characteristic fields; ii) Boundary conditions are not directly imposed as  $u_- = u_-^{BC}$  at boundary points. Instead, the right hand side of the evolution equation is modified to be

$$\frac{\partial u_-}{\partial t} \hat{=} \dots + c(u_-^{BC} - u_-) \quad (3.19)$$

at the boundary. Denoting two adjacent grid patches as  $I$  and  $II$ , we intuitively set the incoming characteristics of  $I$  being equal to the outgoing characteristics of  $II$  and vice versa,

$$u_{-,II}^{BC} = u_{+,I}, \quad u_{-,I}^{BC} = u_{+,II}. \quad (3.20)$$

A back transformation to the original fields is performed to obtain the evolution system with penalty terms. The penalty parameter  $c$  can be determined from a semi-discrete energy analysis. This has been carried out for a general symmetric hyperbolic system of PDEs in [Hilditch et al., 2016]. In this analysis, the total energy on the two adjacent patches is expressed as an numerical integral of the fields. For our choice of collocation points, this is done by means of Legendre-Gauss-Lobatto integration. Demanding that the change of this energy due to the boundary patching should not be positive, an estimate for the penalty parameter can be found. Typically, this parameter depends on the integration weights and the Jacobian at the boundary point. In our case, the penalty parameters for a characteristic field with speed  $\Lambda^s$  in  $s$ -direction, e.g. at the  $\xi^1 = \text{const}$ -surfaces, can be expressed as

$$c_{0jk} = \frac{\Lambda^s N(N-1)}{2l} (\xi_0, \xi_j, \xi_k), \quad l = \sqrt{\gamma^{ij} \frac{\partial \xi^1}{\partial x^i} \frac{\partial \xi^1}{\partial x^j}}, \quad (3.21)$$

where  $N$  is the number of collocation points in  $\xi^1$ -direction.

### The cartoon method

The *bamps* code allows us to exploit the axisymmetry or the spherical symmetry of a physical system. In the first case, the numerical domain is reduced to the  $y = 0, x > 0$ -half plane. In the latter case, only the positive  $x$ -axis is used for numerical evolution. It is obvious, that this approach decreases the computational costs significantly. To account for these two types of symmetries, we use the *cartoon method* [Alcubierre et al., 2001; Pretorius, 2005] in Cartesian coordinates and with Cartesian tensor components. In the case of axisymmetry, e.g. for rotations along the vector field  $\phi = (-y, x, 0)^T$ , the Lie derivative of an arbitrary tensor field  $T$  along  $\phi$  should vanish:

$$\mathcal{L}_\phi T = 0. \quad (3.22)$$

Evaluating this condition at  $y = 0$  yields analytic expressions for the unknown  $y$ -derivative. We have

$$\begin{aligned} \mathcal{L}_\phi T_{\underline{j}_1 \dots \underline{j}_m}^{i_1 \dots i_n} &= \phi^l \partial_l T_{\underline{j}_1 \dots \underline{j}_m}^{i_1 \dots i_n} - \sum_{k=1}^n T_{\underline{j}_1 \dots \underline{j}_m}^{i_1 \dots l \dots i_n} \partial_l \phi^{i_k} + \sum_{k=1}^n T_{\underline{j}_1 \dots \underline{j}_m}^{i_1 \dots i_n} \partial_{j_k} \phi^l \\ &= x \partial_y T_{\underline{j}_1 \dots \underline{j}_m}^{i_1 \dots i_n} - \sum_{k=1}^n T_{\underline{j}_1 \dots \underline{j}_m}^{i_1 \dots l \dots i_n} A_l^{i_k} + \sum_{k=1}^n T_{\underline{j}_1 \dots \underline{j}_m}^{i_1 \dots i_n} A_{j_k}^l = 0, \end{aligned} \quad (3.23)$$

with the simple matrices

$$A_i^j = \partial_i \phi^j = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (3.24)$$

For all  $x \neq 0$ , (3.23) further simplifies to

$$\partial_y u(x, 0, z) = 0 \quad (3.25)$$

for a scalar field  $u$ , and to

$$\partial_y v^x(x, 0, z) = -\frac{v^y}{x}(x, 0, z), \quad \partial_y v^y(x, 0, z) = \frac{v^x}{x}(x, 0, z) \quad (3.26)$$

for a vector field  $v^i$ . For all higher rank tensor valences, the general formula (3.23) is applied in the code. In the case of spherical symmetry, another Killing vector field  $\tilde{\phi} = (-z, 0, x)^T$  exists and additionally to (3.22),  $\mathcal{L}_{\tilde{\phi}} T = 0$  has to be imposed. In perfect analogy to (3.23), a formula for the unknown derivatives in  $z$ -direction at  $z = 0$  can be derived, e.g.  $\partial_z v^x(x, y, 0) = -\frac{v^z}{x}(x, y, 0)$ . To cover the limit  $x \rightarrow 0$ , we use the L'Hôpital's rule if necessary, e.g. as  $v^y \rightarrow 0$  for  $x \rightarrow 0$ , we get the relation

$$\partial_y v^y(0, 0, z) = \partial_x v^x(0, 0, z). \quad (3.27)$$

Note, that in these symmetry cases we first apply the above cartoon relations on the PDE and then solve the transformed PDE with our numerical methods. E.g. for the rest mass conservation eqs. (2.35,36), this procedure converts

$$\partial_t (\sqrt{\gamma} D) + \partial_x (\sqrt{\gamma} D [\alpha v^x - \beta^x]) + \partial_y (\sqrt{\gamma} D [\alpha v^y - \beta^y]) + \partial_z (\sqrt{\gamma} D [\alpha v^z - \beta^z]) = 0 \quad (3.28)$$

into

$$\begin{aligned} \partial_t (\sqrt{\gamma} D) + \partial_x (\sqrt{\gamma} D [\alpha v^x - \beta^x]) + \partial_z (\sqrt{\gamma} D [\alpha v^z - \beta^z]) = \\ - \begin{cases} \partial_x (\sqrt{\gamma} D [\alpha v^x - \beta^x]), & x = 0 \\ \frac{\sqrt{\gamma} D (\alpha v^x - \beta^x)}{x}, & \text{else} \end{cases} \end{aligned} \quad (3.29)$$

for the axisymmetric case, and into

$$\partial_t (\sqrt{\gamma} D) + \partial_x (\sqrt{\gamma} D [\alpha v^x - \beta^x]) = -2 \begin{cases} \partial_x (\sqrt{\gamma} D [\alpha v^x - \beta^x]), & \text{for } x = 0 \\ \frac{\sqrt{\gamma} D (\alpha v^x - \beta^x)}{x}, & \text{else} \end{cases} \quad (3.30)$$

for spherical symmetry. Note, that this procedure converts flux terms into source terms for balance law type PDEs and principal part terms in non-principal part terms in general.

## 3.2 The Runge-Kutta Discontinuous Galerkin method

In the last section we described our usage of pseudospectral methods for the spacetime evolution. To simultaneously solve the GRHD equations, we employ a DG method to compute spatial derivatives. Coupling this approach to a Runge-Kutta discretisation in time, its application on hyperbolic problems [Cockburn and Shu, 1989; Cockburn et al., 1990; Cockburn and Shu, 1998] turned out to be very successful and somehow natural in the case of balance law equations. The main advantage of a DG scheme is that it combines appealing properties of finite volume and finite difference methods: The covering of complex geometries, hp-adaptivity and locality. A detailed discussion on the relation of DG methods to other numerical schemes is given in [Hesthaven et al., 2007].

### Weak formulation of PDEs

In contrast to the pseudospectral method, the analytical limit of the residual is treated differently in a *Galerkin method*. Instead of enforcing the residual  $R_n$  to vanish at certain points (see eq. (3.4)), its projection on the space of approximating polynomials is set to zero:

$$\int_{\Omega} [\partial_t u_n(\xi, t) - Lu_n(\xi, t)] \phi_i(\xi) d\xi = 0. \quad (3.31)$$

In a one-dimensional pseudospectral method,  $N + 1$  collocation points are used to construct equations for the  $N + 1$  degrees of freedom. Now, the projection of  $R_n$  onto the  $N + 1$  basis polynomials gives the desired analogon in a one-dimensional Galerkin method. Note, that (3.31) approximates the weak form of the original PDE, in which  $\phi_i$  would be a more general test function. It would be natural to choose the

approximation space (and therefore also the test function space) to be polynomials or at least continuous functions in  $\Omega$ . This is exactly the starting point of finite element, or continuous Galerkin methods. However, this ansatz breaks the locality of the scheme, i.e. if the domain is subdivided into elements, numerical operators will act globally and not element-wise. This is a clear disadvantage regarding grid decomposition and parallelization of the algorithm.

In a *Discontinuous Galerkin method*, this issue is overcome by weakening the constraints on the test and basis functions and accepting discontinuities at element interfaces. More precisely, we consider  $\phi_i$  to be piecewise polynomial on  $\Omega$ . On the one hand, this increases the degrees of freedom, as we have multiple values at each element boundary. Furthermore, the connection of the elements, which is enforced by the continuity of the basis functions in a finite element method, needs special care in a DG method. On the other hand, we gain back element local numerical operators and a scheme, which is easy to parallelize. We will comment on these points later again, as we derive the DG scheme for general balance law systems in three spatial dimensions.

As discussed in section 3.1, we divide the numerical domain in subpatches. Each subpatch is now considered as an element  $E$ . We choose our basis functions to be polynomials on the reference grid  $B$  which is mapped to  $E$ :

$$\phi_k(\xi^1, \xi^2, \xi^3) = \ell_{k_1}(\xi^1)\ell_{k_2}(\xi^2)\ell_{k_3}(\xi^3), \quad (3.32)$$

where  $k$  has the meaning of a multiindex  $k = (k_1, k_2, k_3)$ . The  $\ell_i$  are again the Lagrange polynomials introduced in (3.13), defined with respect to the corresponding Legendre-Gauss-Lobatto collocation nodes  $\xi_i \in B$ . Just as the pseudospectral method considered earlier, the final DG scheme will also map nodal values of the solution to its time derivative because of this choice of basis polynomials. The method we employ is therefore categorized as a *nodal* DG scheme. Since we choose the very same collocation points as in section 3.1, the DG scheme operates on the same grid points as the pseudospectral method, so that no interpolation step is necessary. We start from the original balance law

$$\partial_t u + \partial_i f^i(u) = s(u), \quad (3.33)$$

and integrate element-wise against the test functions for a weak formulation:

$$\partial_t \int_E \phi_k u \, d^3x + \int_E \phi_k \partial_i f^i \, d^3x = \int_E \phi_k s \, d^3x. \quad (3.34)$$



To evaluate these integrals, we need a transformation to the reference grid:

$$\partial_t \int_B \phi_k u \det J d^3 \xi + \int_B \phi_k (J^{-1})^{j_{\underline{i}}} (\partial_j f^{\underline{i}}) \det J d^3 \xi = \int_B \phi_k s \det J d^3 \xi, \quad (3.35)$$

making use of the Jacobian (3.12).

### Metric identities

The second integral on the left hand side of (3.35) can be further modified by exploiting the *metric identities* [Kopriva, 2006]

$$\partial_j \left( (J^{-1})^{j_{\underline{i}}} \det J \right) = 0. \quad (3.36)$$

These identities can be verified in several ways. We will derive them in a straightforward manner. Using the adjoint method for the Jacobian inverse we write

$$(J^{-1})^{j_{\underline{i}}} \det J = \frac{1}{2} \varepsilon_{\underline{i} \underline{i}_1 \underline{i}_2} \varepsilon^{j j_1 j_2} \frac{\partial x^{\underline{i}_1}}{\partial \xi^{j_1}} \frac{\partial x^{\underline{i}_2}}{\partial \xi^{j_2}} = \frac{1}{2} \varepsilon_{\underline{i} \underline{i}_1 \underline{i}_2} \varepsilon^{j j_1 j_2} \frac{\partial}{\partial \xi^{j_1}} \left( x^{\underline{i}_1} \frac{\partial x^{\underline{i}_2}}{\partial \xi^{j_2}} \right), \quad (3.37)$$

where the last equality holds because  $\varepsilon^{j j_1 j_2}$  is antisymmetric, whereas  $\partial_{j_1} \partial_{j_2} x^{\underline{i}_2}$  is symmetric and therefore the second derivative terms cancel. By the same argument, we immediately see that the metric identities must hold, because  $\varepsilon^{j j_1 j_2}$  is antisymmetric, whereas  $\partial_j \partial_{j_1} (\dots)$  is symmetric. In vector calculus, this is a basic principle of the vanishing divergence of the curl. In numerics, the metric identities can be violated due to numerical differentiation errors. However, eq. (3.37) can be used to fulfill the metric identities numerically. This is a very important feature of a numerical scheme acting on curvilinear grids, if we want to guarantee free stream preservation. Although the identities hold in the continuum limit, the numerical divergence of  $s_{\underline{i}}^j := (J^{-1})^{j_{\underline{i}}} \det J$ ,

$$\sum_{l=1}^N \left( D_{il} s_{\underline{i}}^1(\xi_l^1, \xi_j^2, \xi_k^3) + D_{jl} s_{\underline{i}}^2(\xi_i^1, \xi_l^2, \xi_k^3) + D_{kl} s_{\underline{i}}^3(\xi_i^1, \xi_j^2, \xi_l^3) \right), \quad (3.38)$$

can differ from zero. To cure this issue, we do not use the analytical expression for  $s_{\underline{i}}^j$  in the code, but rather calculate an  $N$ -th order approximation using (3.37) and replacing the partial derivative by a numerical derivative:

$$2(J^{-1})^{j_{\underline{i}}} \det J \approx \varepsilon_{\underline{i} \underline{i}_1 \underline{i}_2} \sum_{l=1}^N \left( \varepsilon^{1 j_2 j} D_{il} \left( x^{\underline{i}_1} \frac{\partial x^{\underline{i}_2}}{\partial \xi^{j_2}} \right) (\xi_l^1, \xi_j^2, \xi_k^3) + \varepsilon^{2 j_2 j} D_{jl} \left( x^{\underline{i}_1} \frac{\partial x^{\underline{i}_2}}{\partial \xi^{j_2}} \right) (\xi_i^1, \xi_l^2, \xi_k^3) + \varepsilon^{3 j_2 j} \dots \right). \quad (3.39)$$

With this modification, the expression (3.38) is exactly zero in the numerics. In Sec. 4.1 we investigate the effect of using (3.39) on the mass conservation in fluid simulations.

### Discretization of the integrals

Making use of (3.36), the flux term in (3.35) can be changed to

$$\int_B \phi_k (J^{-1})^{j_{\underline{i}}} (\partial_j f^{\underline{i}}) \det J d^3 \xi = \int_B \phi_k \partial_j ((J^{-1})^{j_{\underline{i}}} f^{\underline{i}} \det J) d^3 \xi. \quad (3.40)$$

For the treatment of the integrals, we expand all spatial variables along  $\phi_l = \ell_{l_1} \ell_{l_2} \ell_{l_3}$ , e.g. we approximate  $u \det J \approx (u \det J)_n^l \phi_l$  and get

$$\begin{aligned} \partial_t \int_B \phi_k (u \det J) d^3 \xi &\approx \partial_t (u \det J)_n^l \int_B \phi_k \phi_l d^3 \xi \\ &= (\partial_t u \det J)_n^{l_1 l_2 l_3} M_{l_1 k_1} M_{l_2 k_2} M_{l_3 k_3}, \end{aligned} \quad (3.41)$$

where we defined the *mass matrix* as

$$M_{ab} = \int_{-1}^1 \ell_a(\xi) \ell_b(\xi) d\xi. \quad (3.42)$$

Note, that the source integral in (3.35) is of the same type and we approximate

$$\int_B \phi_k s \det J d^3 \xi \approx (s \det J)_n^{l_1 l_2 l_3} M_{l_1 k_1} M_{l_2 k_2} M_{l_3 k_3}. \quad (3.43)$$

Instead of expanding the product of the solution  $u$  and the Jacobian determinant, one might expand only  $u$  and compute the full integral, including the analytical Jacobian determinant. However, this choice would make the mass matrix element dependent, whereas in our approach one single mass matrix is valid on all elements with the same order of approximation. As for the flux integral (3.40), we introduce the surface normal covector in the  $j$ -direction  $s_{\underline{i}}^j = (J^{-1})^{j_{\underline{i}}} \det J$  and apply the fundamental theorem of calculus:

$$\begin{aligned} &\int_B \phi_k \partial_j ((J^{-1})^{j_{\underline{i}}} f^{\underline{i}} \det J) d^3 \xi \\ &= \iiint_{-1}^1 \phi_{k_1} \phi_{k_2} \phi_{k_3} (\partial_1 (s_{\underline{i}}^1 f^{\underline{i}}) + \partial_2 (s_{\underline{i}}^2 f^{\underline{i}}) + \partial_3 (s_{\underline{i}}^3 f^{\underline{i}})) d\xi^1 d\xi^2 d\xi^3 \\ &= \iint_{-1}^1 \left( [\phi_{k_1} s_{\underline{i}}^1 f^{\underline{i}}]_{-1}^1 - \int_{-1}^1 s_{\underline{i}}^1 f^{\underline{i}} \partial_1 \phi_{k_1} d\xi^1 \right) \phi_{k_2} \phi_{k_3} d\xi^2 d\xi^3 + \\ &\quad \iint_{-1}^1 \left( [\phi_{k_2} s_{\underline{i}}^2 f^{\underline{i}}]_{-1}^1 - \int_{-1}^1 s_{\underline{i}}^2 f^{\underline{i}} \partial_2 \phi_{k_2} d\xi^2 \right) \phi_{k_1} \phi_{k_3} d\xi^1 d\xi^3 + \\ &\quad \iint_{-1}^1 \left( [\phi_{k_3} s_{\underline{i}}^3 f^{\underline{i}}]_{-1}^1 - \int_{-1}^1 s_{\underline{i}}^3 f^{\underline{i}} \partial_3 \phi_{k_3} d\xi^3 \right) \phi_{k_1} \phi_{k_2} d\xi^1 d\xi^2. \end{aligned} \quad (3.44)$$

Note, that the first terms on each line are surface integrals over the elements boundary surface. Although we will use (3.44) for our computations, the result can be summarized in a very elegant way with the divergence theorem and multidimensional partial integration:

$$\iiint_E \phi_k \nabla \mathbf{f} \, d^3x = \iint_{\partial E} \phi_k \mathbf{f} \cdot d\mathbf{S} - \iiint_E \mathbf{f} \cdot \nabla \phi_k \, d^3x. \quad (3.45)$$

To treat the remaining integrals in (3.44), we again have to approximate the spatial variables, i.e.  $s_i^j f^i \approx (s_i^j f^i)_n^l \phi_l$ . With this expansion, we get the final expressions for our DG scheme, representing the flux term of the balance law:

$$\begin{aligned} & \int_B \phi_k \partial_j ((J^{-1})^j_i f^i \det J) \, d^3\xi \\ & \approx \left( [\delta_{k_1 N} (s_i^1 f^{*i})_n^{Nl_2l_3} - \delta_{k_1 0} (s_i^1 f^{*i})_n^{0l_2l_3}] - (s_i^1 f^i)_n^{l_1l_2l_3} S_{k_1l_1} \right) M_{k_2l_2} M_{k_3l_3} + \\ & \quad \left( [\delta_{k_2 N} (s_i^2 f^{*i})_n^{Nl_1l_3} - \delta_{k_2 0} (s_i^2 f^{*i})_n^{0l_1l_3}] - (s_i^2 f^i)_n^{l_1l_2l_3} S_{k_2l_2} \right) M_{k_1l_1} M_{k_3l_3} + \\ & \quad \left( [\delta_{k_3 N} (s_i^3 f^{*i})_n^{Nl_1l_2} - \delta_{k_3 0} (s_i^3 f^{*i})_n^{0l_1l_2}] - (s_i^3 f^i)_n^{l_1l_2l_3} S_{k_3l_3} \right) M_{k_1l_1} M_{k_2l_2}, \end{aligned} \quad (3.46)$$

where we defined  $k = (k_1, k_2, k_3)$  as a multiindex and the *stiffness matrix* as

$$S_{ab} = \int_{-1}^1 \ell_b(\xi) \partial_\xi \ell_a(\xi) \, d\xi. \quad (3.47)$$

The numerical flux function  $f^{*i}$  will be explained momentarily.

### Numerical flux

Another inconspicuous but very important change, comparing the analytic form (3.44) and its numerical approximation (3.46), is the substitution of the physical flux  $f$  by a *numerical flux*  $f^*$  in the surface integral terms. As mentioned in the beginning of this section, the numerical solution is allowed to be discontinuous at element interfaces. The value of  $u_n$  is therefore not uniquely defined on the element surfaces and neither is  $f^i$ . The idea to overcome this issue by introducing the numerical flux is borrowed from finite volume methods. In both methodologies, DG and finite volume,  $f^*$  is used to connect the elements and constitutes a key feature of the numerical scheme. It maps the two solutions  $u_L, u_R$  from the adjacent elements to a unique value  $f^{*i}(u_L, u_R)$  that is used as the surface flux for both grids. A natural requirement on  $f^*$  is its reduction to the physical flux in the continuous solution case  $f^{*i}(u, u) = f^i(u)$ . Usually, knowledge of the dynamics of the system enters the numerical flux construction. There are many different choices for  $f^*$ , we just want to name the Roe flux [Roe, 1981], Harten, Lax and van Leer (HLL) flux [Harten et al., 1983] and the local Lax-Friedrich (LLF) flux [Shu and Osher, 1989] as some of the

most widely used. We decided to embed the HLL flux

$$s_i f^{*i}(u_L, u_R) = \frac{\lambda_+ s_i f^i(u_L) - \lambda_- s_i f^i(u_R) + \lambda_- \lambda_+ (u_R - u_L)}{\lambda_+ - \lambda_-}, \quad (3.48)$$

with the two characteristic speeds  $\lambda_+, \lambda_-$ , into our scheme. We want to cite a short calculation from [Rezzolla and Zanotti, 2013] to motivate (3.48) and to understand the approximations made. The main assumption of the HLL flux is that the initial discontinuity (i.e. the Riemann problem with initial states  $u_L, u_R$ ) decays into three constant states  $u_L, u_{\text{HLL}}, u_R$  and just two waves traveling in opposite directions with speeds  $\lambda_+, \lambda_-$ :

$$u(x^s, t) = \begin{cases} u_L, & \text{if } x^s < \lambda_- t \\ u_R, & \text{if } x^s > \lambda_+ t \\ u_{\text{HLL}}, & \text{else} \end{cases}, \quad (3.49)$$

where  $x^s = s_i x^i$ . To evaluate the intermediate state at the boundary  $u_{\text{HLL}}$ , a perfect conservation law for  $u$  is assumed (i.e. the source terms are neglected) and integrated over a spacetime control volume  $[0, T] \times [-S, S]$ :

$$\int_0^T \int_{-S}^S \partial_t u(x^s, t) dx^s dt + \int_0^T \int_{-S}^S \partial_{x^s} (s_i f^i(x^s, t)) dx^s dt = 0. \quad (3.50)$$

On the one hand, this leads to

$$\begin{aligned} \int_{-S}^S u(x^s, T) dx^s &= \int_{-S}^S u(x^s, 0) dx^s + \int_0^T s_i f^i(-S, t) dt - \int_0^T s_i f^i(S, t) dt \\ &= S(u_L + u_R) + T(s_i f_L^i - s_i f_R^i), \end{aligned} \quad (3.51)$$

on the other hand it is

$$\begin{aligned} \int_{-S}^S u(x^s, T) dx^s &= \int_{-S}^{T\lambda_-} u_L dx^s + \int_{T\lambda_-}^{T\lambda_+} u_{\text{HLL}} dx^s + \int_{T\lambda_+}^S u_R dx^s \\ &= u_L(T\lambda_- + S) + u_{\text{HLL}}T(\lambda_+ - \lambda_-) + u_R(S - T\lambda_+). \end{aligned} \quad (3.52)$$

Combining eqs. (3.51,52), the intermediate state can be expressed as

$$u_{\text{HLL}} = \frac{\lambda_+ u_R - \lambda_- u_L + s_i f_L^i - s_i f_R^i}{\lambda_+ - \lambda_-}, \quad (3.53)$$

and from the Rankine-Hugoniot conditions  $s_i f^{*i} = s_i f_L^i + \lambda_- (u_{\text{HLL}} - u_L)$  the HLL numerical flux (3.48) follows immediately. Finally, we choose  $\lambda_+, \lambda_-$  to be the maximum propagation speeds in positive and negative  $s_i$  direction, making use of the

characteristic eigenvalues of the GRHD system (2.38):

$$\lambda_{\pm} = \max(0, \pm\lambda_{\pm}(u_R), \pm\lambda_{\pm}(u_L)). \quad (3.54)$$

### Final DG scheme

Now, combining the weak form of the PDE (3.35) and the integral approximations (3.41,43,46), one derives the final DG scheme as used in bamps:

$$\begin{aligned} (\partial_t u)_n^{l_1 l_2 l_3} &= (s)_n^{l_1 l_2 l_3} + \\ &\frac{1}{(\det J)_n^{l_1 l_2 l_3}} \left( (M^{-1})^{0l_1} (s_i^1 f^{*i})_n^{0l_2 l_3} - (M^{-1})^{Nl_1} (s_i^1 f^{*i})_n^{Nl_2 l_3} + (M^{-1})^{ml_1} S_{mk_1} (s_i^1 f^i)_n^{k_1 l_2 l_3} \right. \\ &\quad + (M^{-1})^{0l_2} (s_i^2 f^{*i})_n^{l_1 0l_3} - (M^{-1})^{Nl_2} (s_i^2 f^{*i})_n^{l_1 Nl_3} + (M^{-1})^{ml_2} S_{mk_2} (s_i^2 f^i)_n^{l_1 k_2 l_3} \\ &\quad \left. + (M^{-1})^{0l_3} (s_i^3 f^{*i})_n^{l_1 l_2 0} - (M^{-1})^{Nl_3} (s_i^3 f^{*i})_n^{l_1 l_2 N} + (M^{-1})^{ml_3} S_{mk_3} (s_i^3 f^i)_n^{l_1 l_2 k_3} \right) \end{aligned} \quad (3.55)$$

with the precomputed geometric quantities  $\det J$  and  $s_i^j$  from the static curvilinear grid structure, and the state dependent fluxes  $f^i(u)$ ,  $f^{*i}(u)$  and sources  $s(u)$ . The time derivatives of  $u$  are again fed into the time integrator, just as the spacetime evolution variables in the pseudospectral scheme. Note, that by means of (3.55), only an evolution scheme for the conserved variables is available. However, for the flux computation also the primitive variables are needed. Therefore, after each Runge-Kutta substep a recovery of primitive variables has to be performed. See Appendix A for details. For the sake of simplicity, we assume the use of the same number of collocation points (i.e. the same polynomial approximation order) in each direction. Obviously, the DG scheme is not restricted to that case, as one calculates one mass and stiffness matrix for each direction. The stiffness matrix is easily available, using Gauss-Legendre-Lobatto integration:

$$S_{ab} = \int_{-1}^1 \ell_b(\xi) \partial_{\xi} \ell_a(\xi) d\xi = \sum_{k=1}^N \ell_b(\xi_k) \partial_{\xi} \ell_a(\xi_k) \omega_k = \partial_{\xi} \ell_a(\xi_b) \omega_b \equiv D_{ba} \omega_b. \quad (3.56)$$

Note, that the second equality in the above equation is exact, because the integrand is of order  $2N - 1$  and so is the highest order of approximation with  $N + 1$  Gauss-Legendre-Lobatto points. For the same reason, the mass matrix can not be determined exactly in the same simple manner. However, this *mass lumped* approximation

$$M_{ab} = \int_{-1}^1 \ell_a(\xi) \ell_b(\xi) d\xi \approx \sum_{k=1}^N \ell_a(\xi_k) \ell_b(\xi_k) \omega_k = \delta_{ab} \omega_a. \quad (3.57)$$

is often used in practice. It comes with the effect of a modal filter, decreasing the highest mode by a factor  $\frac{N}{2N+1}$  [Gassner and Kopriva, 2011]. The diagonal form of

the mass matrix further simplifies the DG scheme, so that all numerical flux terms only enter the right hand sides, that correspond to boundary points. In the bumps code, we calculate the mass matrix exactly and find that this decreases numerical errors significantly as compared to the mass lumped version (see Sec. 4.1, smooth 1D SRHD tests). In [Bug1] we found independently from [Teukolsky, 2015b] the following simple relation for the Gauss-Legendre-Lobatto mass matrix:

$$(M^{-1})^{ij} = \frac{1}{\omega_j} \delta^{ij} + \frac{N+1}{2} P_N(\xi_i) P_N(\xi_j), \quad (3.58)$$

with  $P_N$  again denoting the  $N$ -th Legendre polynomial. We give a derivation of this result in Appendix C. As a closing statement of this section, we want to revisit locality of the scheme. The most costly part of the DG scheme (3.55) is a matrix multiplication acting on the element local flux field. The only interaction with neighboring elements is implemented through the numerical flux  $f^{*i}$ . For the computation of  $f^{*i}$ , only the  $(d-1)$ -dimensional surface data of the  $d$ -dimensional neighbor element is needed. Since we use Gauss-Legendre-Lobatto points, so that the element boundaries are actually covered by nodes, this reduces the communication between two elements to a simple exchange of the data fields at the shared boundary. This is a neat feature of the scheme and simplifies its parallel implementation significantly.

### 3.3 High resolution shock capturing methods

As we present in the tests section of this work (Chapter 4), the DG scheme as derived in Sec. 3.2 almost perfectly applies on smooth solutions without further modification. However, in the case of strong discontinuities or shocks, the method tends to generate spurious oscillations which ultimately lead to instability. In this section, a variety of methods is presented, which target the prevention of such artificial oscillations and unphysical behavior caused by the Gibbs phenomenon. For this purpose, we locate discontinuities and oscillations with a troubled cell indicator and apply limiting procedures on these elements. In the scope of this work, we investigate the traditional WENO limiting approach in the context of DG methods [Qiu and Shu, 2005; Zhao and Tang, 2013], as well as a subcell based strategy [Radice and Rezzolla, 2011; Dumbser et al., 2014], where the DG elements are subdivided into control volumes and the solution is obtained by means of a finite volume scheme, in Sec. 3.3.2. In [Bug1], we also investigated the WENO-Z [Borges et al., 2008] and the simple WENO [Zhong and Shu, 2013] approach. The former method (which is similar to what we introduce as the WENO-5 method below) was not able to resolve shocks or star surfaces accurately enough in the multidimensional case. The latter method underlies a significant overhead for the parallelization, as the

whole polynomial information of neighboring cells has to be communicated. Because of these issues, we do not consider both methods in full detail here.

### 3.3.1 Weighted-essentially-non-oscillatory methods

In this section, the WENO limiting methodology and the detection of elements containing shocks and discontinuities is explained. For both, we first consider one-dimensional problems for simplicity and subsequently give a generalization to multi-dimensional problems. Since the bumps code has a magnificent parallel performance, a parallel WENO adaption is discussed. We also analyze a key ingredient of the WENO scheme, involving the solution of an overdetermined linear system of equations, in terms of solvability.

#### Troubled cell indication

Given the coefficients of the numerical solution  $u_n(x, t)$  at time  $t$ , we can calculate its average on the elements  $E_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ :

$$\bar{u}_j(t) := \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u_n(x, t) dx = \frac{1}{2} \int_{-1}^1 u_n^i(t) \ell_i(\xi) d\xi, \quad (3.59)$$

We further denote the boundary values of  $u_n$  as

$$u_j^- := u_n(x_{j-\frac{1}{2}}), \quad u_j^+ := u_n(x_{j+\frac{1}{2}}), \quad (3.60)$$

and define the four differences:

$$\begin{aligned} \tilde{u}_j^- &:= \bar{u}_j - u_j^-, & \tilde{u}_j^+ &:= u_j^+ - \bar{u}_j \\ \Delta_- u &:= \bar{u}_j - u_{j-1}, & \Delta_+ u &:= u_{j+1} - \bar{u}_j \end{aligned} \quad (3.61)$$

We also introduce the minmod function

$$\text{minmod}(a_1, a_2, \dots, a_n) = \begin{cases} s \cdot \min_{1 \leq j \leq n} |a_j| & \text{if } \text{sign}(a_1) = \dots = \text{sign}(a_n) =: s \\ 0 & \text{otherwise} \end{cases} \quad (3.62)$$

and the modified minmod function

$$\text{minmod}_M(a_1, a_2, \dots, a_n) = \begin{cases} a_1 & \text{if } |a_1| \leq M (\max_j \Delta x_j)^2 \\ \text{minmod}(a_1, a_2, \dots, a_n) & \text{otherwise} \end{cases}. \quad (3.63)$$

In practice, the constant  $M > 0$  acts as a threshold for the troubled cell indication. The lower  $M$ , the more elements will be marked as troubled. The particular choice of

$M$  is problem-dependent. Our troubled cell indicator marks an element as troubled, if

$$\begin{aligned} \min\text{mod}_M(\tilde{u}_j^-, \Delta_- u, \Delta_+ u) &\neq (\tilde{u}_j^-) \quad \text{or} \\ \min\text{mod}_M(\tilde{u}_j^+, \Delta_- u, \Delta_+ u) &\neq (\tilde{u}_j^+). \end{aligned} \quad (3.64)$$

This is exemplary for a situation, in which  $u_n$  is not monotonous (because the arguments of  $\min\text{mod}$  differ in sign) or its gradient inside a patch is larger than that of the neighboring patches (shock inside the element). In the case of a system of equations, we perform the check (3.64) on each component of  $u_n$ . If one of the components is marked as troubled, the element is marked as troubled.

In the case of multiple dimensions, we perform the 1D troubled cell indication in every coordinate direction. An element is marked as troubled, if at least one of these indications results in a troubled state. To apply the 1D algorithm, the boundary values used in (3.60) have to be modified, since the element boundaries are no longer single points, but lines or surfaces. Therefore, we redefine  $u_j^\pm$  by the boundary averages, i.e. for a 3D rectilinear element  $E_{jkl} = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \times [y_{k-\frac{1}{2}}, y_{k+\frac{1}{2}}] \times [z_{l-\frac{1}{2}}, z_{l+\frac{1}{2}}]$  in  $x$ -direction

$$\begin{aligned} u_{jkl}^- &:= \frac{1}{\Delta y \Delta z} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} \int_{z_{l-\frac{1}{2}}}^{z_{l+\frac{1}{2}}} u_n(x_{j-\frac{1}{2}}, y, z, t) dy dz, \\ u_{jkl}^+ &:= \frac{1}{\Delta y \Delta z} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} \int_{z_{l-\frac{1}{2}}}^{z_{l+\frac{1}{2}}} u_n(x_{j+\frac{1}{2}}, y, z, t) dy dz. \end{aligned} \quad (3.65)$$

### Traditional WENO limiting

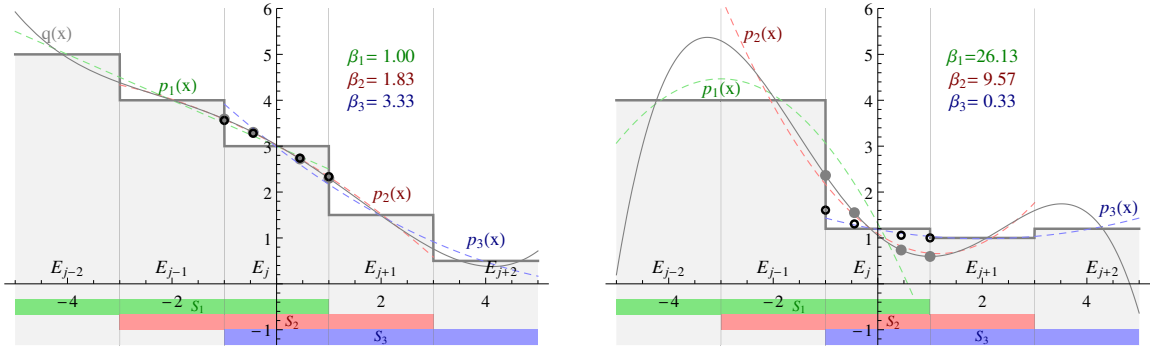
In the standard WENO method of order  $2w + 1$ , one constructs  $w + 1$  stencils  $S_i$  around  $E_j$ , each as an aggregation of  $w + 1$  elements:  $S_i = (E_{j-w+i}, E_{j-w+i+1}, \dots, E_{j+i})$ ,  $0 \leq i \leq w$ . In Fig. 3.3 this partitioning is shown for  $w = 2$ . For each stencil, we construct a  $w$ -th order polynomial  $p_i$ , which has the same average as the numerical solution  $u_n$  over each element in the stencil. That means solving the system

$$\bar{u}_k = \frac{1}{\Delta x} \int_{E_k} p_i(x) dx, \quad \text{for all } E_k \in S_i \quad (3.66)$$

for the  $w + 1$  coefficients of  $p_i$ . Similarly, we construct a  $2w$ -th order polynomial  $q$  fulfilling

$$\bar{u}_k = \frac{1}{\Delta x} \int_{E_k} q(x) dx, \quad \text{for all } E_k \in S, \quad (3.67)$$





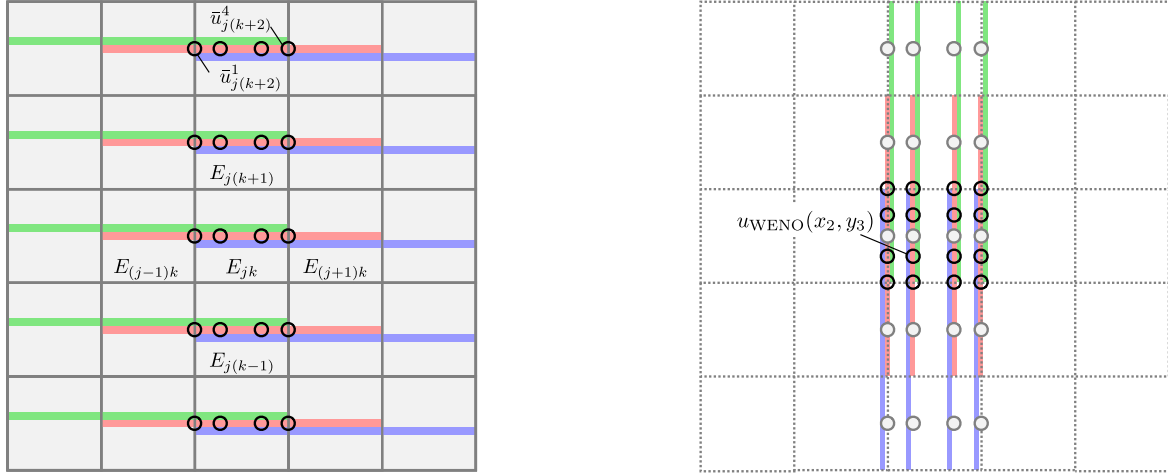
**Figure 3.3:** The WENO-5 methodology applied in a smooth case (left figure) and a shock case (right figure). The values in the interval  $x \in E_j = [-1, 1]$  are to be reconstructed from the five element averages  $\bar{u}_{j-2}, \bar{u}_{j-1}, \bar{u}_j, \bar{u}_{j+1}, \bar{u}_{j+2}$ . The three stencils  $S_1, S_2, S_3$  are created as a clustering of three elements each with the corresponding approximating polynomial  $p_1(x), p_2(x), p_3(x)$ . Another higher order polynomial  $q(x)$  can be found from employing all five averages. Following the strategy as described in the text, the smoothness indicators  $\beta_i$  are calculated for each stencil. A large  $\beta_i$  indicates non-smoothness of the corresponding polynomial  $p_i$ , which leads to a minor contribution of the stencil  $S_i$  for the reconstruction. In the shock case, the reconstructed point values (empty black circles) lie very close to the smoothest polynomial  $p_3$ , whereas in the smooth case all three approximating polynomials are taken into account almost equally, so that the reconstruction is very close to the 5th order polynomial  $q$  (filled grey circles). Adapted from [Bug1].

with  $S := \cup_i S_i$  being the large stencil over all  $2w + 1$  elements. The fundamental concept is to approximate the solution in  $E_j$  as a linear combination of the  $p_i$ , which should give the same result as the higher order approximation  $q$  in smooth regions. This condition defines the linear (or ideal) weights  $\gamma_i$  satisfying

$$q(x) = \sum_{i=1}^{w+1} \gamma_i(x) p_i(x). \quad (3.68)$$

We emphasize that the  $\gamma_i$  depend on the point  $x$  where the approximation should hold. It is remarkable that although both sides of Eq. (3.68) depend intrinsically on the  $2w + 1$  averages  $\bar{u}_k$ , the system is soluble for almost every  $x \in E_j$ , although it is overdetermined (only  $w + 1$  variables). Details on the solvability of (3.68) are given in the following discussion. In regions where the solution is not smooth, the weights should be chosen such that the smoothest polynomial in  $\{p_i\}$  is preferred. For this purpose, we use a smoothness indicator as suggested in [Jiang and Shu, 1996]:

$$\beta_i = \sum_{l=1}^w \int_{E_j} \Delta x^{2l-1} \left( \frac{d^l}{dx^l} p_i(x) \right)^2 dx. \quad (3.69)$$



**Figure 3.4:** Direction-by-direction paradigm for the WENO-5 reconstruction of element  $E_{jk}$  in a 2D rectilinear grid. In accordance with Fig. 3.3, input averages for the reconstruction are colored gray, the 3 stencils are colored green, red and blue and the reconstructed quantities are marked by empty black circles. On the left (right), the first (second) reconstruction step, as explained in the text, is depicted.

Because  $\beta_i$  is large for non-smooth  $p_i$ , the weights are chosen indirect proportional to  $\beta_i$ . We use the traditional WENO choice

$$\tilde{\omega}_i(x) = \frac{\gamma_i(x)}{(10^{-6} + \beta_i)^2}, \quad (3.70)$$

and normalize the result:

$$\omega_i(x) = \frac{\tilde{\omega}_i(x)}{\sum_{l=1}^{w+1} \tilde{\omega}_l(x)}, \quad (3.71)$$

where  $\omega_i(x)$  are the final reconstruction weights. The reconstructed solution is then given by:

$$u_{\text{WENO}}(x) = \sum_{i=1}^{w+1} \omega_i(x) p_i(x). \quad (3.72)$$

In the case of systems, we apply this reconstruction method on each component of  $u_n$ .

To generalize this reconstruction mechanism to multidimensional problems, we use the procedure described in [Zhao and Tang, 2013]. For simplicity, we assume a rectilinear 2D grid structure with  $N + 1$  grid points  $x_p$  per element and direction. To reduce the full reconstruction of the element  $E_{jk}$  to the 1D case, we decouple the different directions as suggested in Fig. 3.4. First we perform  $2w + 1$  1D WENO

reconstructions in the  $x$  direction with input data

$$\{\bar{u}_{j-w,\tilde{k}}, \bar{u}_{j-w+1,\tilde{k}}, \dots, \bar{u}_{j+w,\tilde{k}}\}, \quad k-w \leq \tilde{k} \leq k+w \quad (3.73)$$

to reconstruct the  $N+1$  line averages per element at  $x = x_p$ , denoted  $\bar{u}_{j,\tilde{k}}^p$ . Then, we can apply a second 1D WENO reconstruction based on the 1D averages in  $y$  direction with the input data

$$\{\bar{u}_{j,k-w}^p, \bar{u}_{j,k-w+1}^p, \dots, \bar{u}_{j,k+w}^p\}, \quad 1 \leq p \leq N+1 \quad (3.74)$$

to get the 2D reconstructed values inside the element  $E_{jk}$ :

$$u_{\text{WENO}}(x_p, y_q), \quad 1 \leq p, q \leq N+1 \quad (3.75)$$

This direction-by-direction fashion of the multidimensional limiting procedure is only possible for rectilinear grids, where the ansatz  $p_i(x, y) = p_i^1(x)p_i^2(y)$  for all reconstruction polynomials can be exploited. On unstructured, curvilinear grids, a general limiting algorithm would be much more costly. In this work, we therefore restrict WENO reconstruction to box grid patches, in particular to the central cube of the bumps grid.

### Parallelization of the WENO method

For both the troubled cell indication and the polynomial reconstruction, averages from the neighboring elements are needed. It is therefore obvious that the WENO step is non-local and additional communication is needed in a distributed memory parallelization. Referring to Fig. 3.1, this additional MPI communication block is inserted in between the Runge-Kutta substep and the post substep block. This is very unfavorable for the parallel performance of the code, because for each communication step, synchronization of the processes is needed. While slower processes are still working, faster processes are idling until synchronization is achieved. Another problem is the big stencil of WENO methods. In the case of two-dimensional WENO-5 reconstruction, for example  $5 \times 5$  cell averages are necessary for the computation. This goes far beyond the communication that is usually required in bumps, where data exchange occurs only between grids that share a common face. Even for the WENO-3 reconstruction, this communication is not sufficient. Also average data from the cells that share a common corner point is needed in this case. Altogether, the WENO reconstruction is therefore expected to substantially diminish the parallel performance of the code.

### Solubility of (3.68)

The determination of the optimal weights  $\gamma_j$  is the crucial ingredient in a WENO method, since it guarantees high order convergence for smooth problems. As discussed, the ideal weights are the solutions of an overdetermined linear system of equations. We first want to substantiate this statement, before we actually analyze the solvability of the system. Although this issue has been recognized in some works on WENO methods [Qiu and Shu, 2005; Shi et al., 2002], no analysis and no systematic investigation has been done on the solvability of system (3.68), as far as we know. Intensively using the ideas of [Shu, 1997] one starts by asking for a polynomial  $Q(x)$  which takes certain values at the cell boundary points:

$$Q(x_{j-w-\frac{1}{2}}) = 0, \quad Q(x_{j-w+\frac{1}{2}+k}) = \sum_{n=0}^k \bar{u}_{j-w+n} \Delta x_{j-w+n}, \quad 0 \leq k \leq 2w. \quad (3.76)$$

By explicit integration it is simple to see that  $q(x) = \frac{d}{dx}Q(x)$  fulfills condition (3.67) and therefore represents the interpolating function  $q$  we are looking for. To build the necessary polynomial  $Q$  with the properties (3.76), a standard Lagrange interpolation is performed:

$$Q(x) = \sum_{m=-1}^{2w} \left( \sum_{n=0}^m \bar{u}_{j-w+n} \Delta x_{j-w+n} \right) \prod_{\substack{l=-1 \\ l \neq m}}^{2w} \frac{x - x_{j-w+l+\frac{1}{2}}}{x_{j-w+m+\frac{1}{2}} - x_{j-w+l+\frac{1}{2}}}. \quad (3.77)$$

We can express the dependence of  $q$  from the averages  $\bar{u}$  as a scalar product

$$q(x) = \sum_{n=0}^{2w} \bar{u}_{j-w+n} q_n(x) \quad \text{with}$$

$$q_n(x) := \Delta x_{j-w+n} \frac{d}{dx} \sum_{m=n}^{2w} \prod_{\substack{l=-1 \\ l \neq m}}^{2w} \frac{x - x_{j-w+l+\frac{1}{2}}}{x_{j-w+m+\frac{1}{2}} - x_{j-w+l+\frac{1}{2}}}, \quad 0 \leq n \leq 2w. \quad (3.78)$$

In a similar manner, we can find the lower order polynomials  $p_i$  by fixing the interpolating values only at the cell boundary points of the  $i$ -th stencil. Here, we may express the dependence of  $p_i$  from the averages  $\bar{u}$  as a matrix-vector multiplication:

$$p_i(x) = \sum_{n=0}^w \bar{u}_{j-w+i+n} p_{i,n}(x), \quad 0 \leq i \leq w \quad \text{with}$$

$$p_{i,n}(x) := \Delta x_{j-w+i+n} \frac{d}{dx} \sum_{m=n}^w \prod_{\substack{l=-1 \\ l \neq m}}^w \frac{x - x_{j-w+i+l+\frac{1}{2}}}{x_{j-w+i+m+\frac{1}{2}} - x_{j-w+i+l+\frac{1}{2}}}, \quad 0 \leq n \leq w. \quad (3.79)$$

**Lemma.**  $\forall p_{i,n}(x)$  there exists at most one  $x \in E_j$  with  $p_{i,n}(x) = 0$ .

**Proof.** Consider again the antiderivative of  $p_{i,n}(x) =: \frac{d}{dx}P_{i,n}(x)$ . From its construction and the interpolating property of the Lagrange polynomials, we can read the values of  $P$  at the  $w + 2$  cell boundaries in the stencil:

$$\begin{aligned} P_{i,n}(x_{j-w+i-\frac{1}{2}}) &= \cdots = P_{i,n}(x_{j-w+i+n-\frac{1}{2}}) = 0, \\ P_{i,n}(x_{j-w+i+n+\frac{1}{2}}) &= \cdots = P_{i,n}(x_{j+i+\frac{1}{2}}) = \Delta x_{j-w+i+n} \end{aligned} \quad (3.80)$$

It follows by the mean value theorem, that  $\frac{d}{dx}P_{i,n}$  has at least one root in each of the  $w$  intervals  $E_{j-w+i}, \dots, E_{j-w+i+n-1}, E_{j-w+i+n+1}, \dots, E_{j+i}$ . Since  $p_{i,n} \in \mathbb{P}^w$  has a maximum of  $w$  roots, there is at most one root in  $E_j$ .

Having evaluated the interpolating polynomials, the next step is to find linear weights consistent with (3.68). We can rewrite (3.68) in matrix vector form:

$$\begin{pmatrix} \bar{u}_{j-w} \\ \bar{u}_{j-w+1} \\ \vdots \\ \bar{u}_{j+w} \end{pmatrix}^T \underbrace{\begin{pmatrix} p_{0,0} & 0 & \cdots & 0 \\ p_{0,1} & p_{1,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{0,w} & p_{1,w-1} & \cdots & p_{w,0} \\ 0 & p_{1,w} & \cdots & p_{w,1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{w,w} \end{pmatrix}}_{=: \mathbf{A}} \underbrace{\begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_w \end{pmatrix}}_{=: \boldsymbol{\gamma}} = \begin{pmatrix} \bar{u}_{j-w} \\ \bar{u}_{j-w+1} \\ \vdots \\ \bar{u}_{j+w} \end{pmatrix}^T \underbrace{\begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{2w} \end{pmatrix}}_{=: \mathbf{q}}, \quad (3.81)$$

or with shorthands and valid for all combinations of averages  $\bar{u}$

$$\mathbf{A}\boldsymbol{\gamma} = \mathbf{q}. \quad (3.82)$$

Here,  $\mathbf{A}$  is a matrix with  $2w + 1$  rows and  $w + 1$  columns.  $\boldsymbol{\gamma}$  and  $\mathbf{q}$  are vectors with  $w + 1$  and  $2w + 1$  entries, respectively. We want to emphasize, that both  $\mathbf{A}$  and  $\mathbf{q}$  depend on  $x$ . The linear system of equations (3.82) can therefore be overdetermined and has to be investigated for all points  $x \in E_j$ . We analyze the augmented matrix

$$(\mathbf{A}|\mathbf{q}) = \left( \begin{array}{cccc|c} p_{0,0} & 0 & \cdots & 0 & q_0 \\ p_{0,1} & p_{1,0} & \cdots & 0 & q_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{0,w} & p_{1,w-1} & \cdots & p_{w,0} & q_w \\ 0 & p_{1,w} & \cdots & p_{w,1} & q_{w+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & p_{w,w} & q_{2w} \end{array} \right). \quad (3.83)$$

**Lemma.**  $p_{i,w} \neq 0, 0 \leq i \leq w \Rightarrow w + 1 \leq \text{rank}(\mathbf{A}), w + 1 \leq \text{rank}(\mathbf{A}|\mathbf{q})$

**Proof.** If the condition  $p_{i,w} \neq 0$  is met for all  $0 \leq i \leq w$ , the  $w + 1$  columns of  $\mathbf{A}$  (and therefore the  $w + 1$  left columns of  $(\mathbf{A}|\mathbf{q})$ ) are linearly independent. The rank of  $\mathbf{A}$  and  $(\mathbf{A}|\mathbf{q})$  must be at least  $w + 1$  in this case.

**Lemma.**  $\text{rank}(\mathbf{A}(x_0)) \leq w + 1$  and  $\text{rank}((\mathbf{A}|\mathbf{q})(x_0)) \leq w + 1$  for every  $x_0 \in E_j$ .

**Proof.**  $\text{rank}(\mathbf{A}) \leq w + 1$  is trivial, since  $\mathbf{A}$  has  $w + 1$  columns. At  $x_0$  fixed, we consider  $(\mathbf{A}|\mathbf{q})$  to be the representation of a linear map  $V = \mathbb{R}^{2w+1} \rightarrow W = \mathbb{R}^{w+2}$ . To show the second inequality, we use the rank-nullity theorem  $\dim(V) = 2w + 1 = \text{rank}(\mathbf{A}|\mathbf{q}) + \ker(\mathbf{A}|\mathbf{q})$  and show that  $\ker(\mathbf{A}|\mathbf{q}) \geq w$ . To prove this, we construct  $w$  polynomials  $n_i(x)$  of order  $w$  with the following properties:

$$\begin{aligned} n_1(x_0) = 0, \int_{E_{j+1}} n_1(x) dx = 1, \int_{E_{j+2}} n_1(x) dx = 0, \dots, \int_{E_{j+w}} n_1(x) dx = 0 \\ n_2(x_0) = 0, \int_{E_{j+1}} n_2(x) dx = 0, \int_{E_{j+2}} n_2(x) dx = 1, \dots, \int_{E_{j+w}} n_2(x) dx = 0 \\ \dots \\ n_w(x_0) = 0, \int_{E_{j+1}} n_w(x) dx = 0, \int_{E_{j+2}} n_w(x) dx = 0, \dots, \int_{E_{j+w}} n_w(x) dx = 1 \end{aligned} \quad (3.84)$$

i.e. all polynomials have a zero at  $x_0 \in E_j$  and have a zero average in all intervals  $E_{j+1}, \dots, E_{j+w}$  except for one. If we would require all of the above averages to be zero, all polynomials would be identically zero (again by the mean-value-theorem). But in the case (3.84) all  $n_i$  can be constructed with the desired properties. On the one hand, the  $w$  vectors containing the corresponding polynomial averages  $(\bar{u}_{j-w}, \dots, \bar{u}_{j+w})$  are linearly independent by construction. On the other hand, since the approximation by both the  $w$ -th order polynomials  $p_i$  and the  $2w$ -th order polynomial  $q$  is exact,  $(\mathbf{A}|\mathbf{q})$  maps all these vectors to zero, because  $n_i(x_0) = 0$  by construction. Therefore the kernel dimension of  $(\mathbf{A}|\mathbf{q})$  is at least  $w$ .

Combining these insights, the best statement we can give on the solubility of (3.68) is the following: If all  $p_{i,w}(x_0) \neq 0$  for  $x_0 \in E_j$ , then (3.68) is uniquely solvable at  $x_0$ , because  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}|\mathbf{q}) = w + 1$ . At the same time, each  $p_{i,w}$  has at most one root in  $E_j$ . Therefore there exist at most  $w + 1$  points in  $E_j$ , where one of the  $p_{i,w}$  is zero and solubility can not be guaranteed.

### 3.3.2 The spectral volumes method

In this section we discuss a different shock capturing approach based on a reliable, low order finite volume method. A very similar idea has been used to stabilize DG methods for spherically symmetric neutron star simulations containing an atmosphere

in [Radice and Rezzolla, 2011]. The purpose of this method is to stabilize the scheme in low density regions of star simulations. We mark a cell as troubled, if a specified variable  $u$  (in most cases the fluid density) falls below a certain threshold  $u_{SV}$  at one or more collocation points in the cell. This check is performed before each time step. If a cell is flagged troubled, the  $N$ -th order polynomial from the DG ansatz is converted to  $N + 1$  subcell averages. A finite volume method is then used to evolve these averages forward in time. We first give details on this method in one dimension and explain its interaction with the DG method. Afterwards, a possible implementation and issues of the higher dimensional method are explained.

### Troubled cell evolution

If a one-dimensional  $N$ -th order DG element  $E_j = [a, b]$  is flagged as troubled, we divide this element into  $N + 1$  *spectral volumes* (SV) [Wang, 2002] of equal size  $\Delta x = \frac{b-a}{N+1}$ :

$$S_k = \left[ \underbrace{a + k\Delta x}_{=:x_k^B}, \underbrace{a + (k+1)\Delta x}_{=:x_{k+1}^B} \right], \quad 0 \leq k \leq N. \quad (3.85)$$

For these elements, the  $N + 1$  degrees of freedom  $u^i$  in the DG approximation  $u(x) = u^i \ell_i(x)$  are converted to  $N + 1$  SV averages  $\bar{u}_k$ :

$$\bar{u}_k = \frac{1}{\Delta x} \int_{S_k} u^i \ell_i(x) dx = u^i \ell_i(x_k^C) + \mathcal{O}(\Delta x^2), \quad (3.86)$$

where we used the midpoint rule approximation of the integral with the midpoint  $x_k^C := \frac{x_{k+1}^B + x_k^B}{2}$ . Although the integral after the first equality could be easily calculated in order to determine  $\bar{u}_k$  exactly, the second equality is employed in the code. Interpolation routines and weights were already available in `bamps` and the induced second order error does not impair the first order accurate SV scheme.

Once the SV averages have been determined, a standard finite volume scheme can be applied. We integrate the balance law PDE (2.35) over a SV  $S_k$  to gain:

$$\begin{aligned} \frac{1}{\Delta x} \int_{S_k} \partial_t u(x) dx + \frac{1}{\Delta x} \int_{S_k} \partial_x f(u) dx &= \frac{1}{\Delta x} \int_{S_k} s(u) dx \\ \partial_t \bar{u}_k + \frac{f^*(x_{k+1}^B) - f^*(x_k^B)}{\Delta x} &= \bar{s}_k \end{aligned} \quad (3.87)$$

Just as in the DG method, a numerical flux function  $f^*$  has to be specified as an approximation of the physical flux at the interface between two SVs. For this purpose, the HLL flux (3.48) is again used. However, a major difference to the DG method is that the left and right state  $u_L$  and  $u_R$  are not immediately accessible, because the solution is stored in terms of SV averages. Hence, a finite volume method is characterized by a certain reconstruction of  $u_L$  and  $u_R$  from the averages  $\bar{u}_k$ . For our

implementation, we chose linear reconstruction with a minmod slope limiter, i.e.

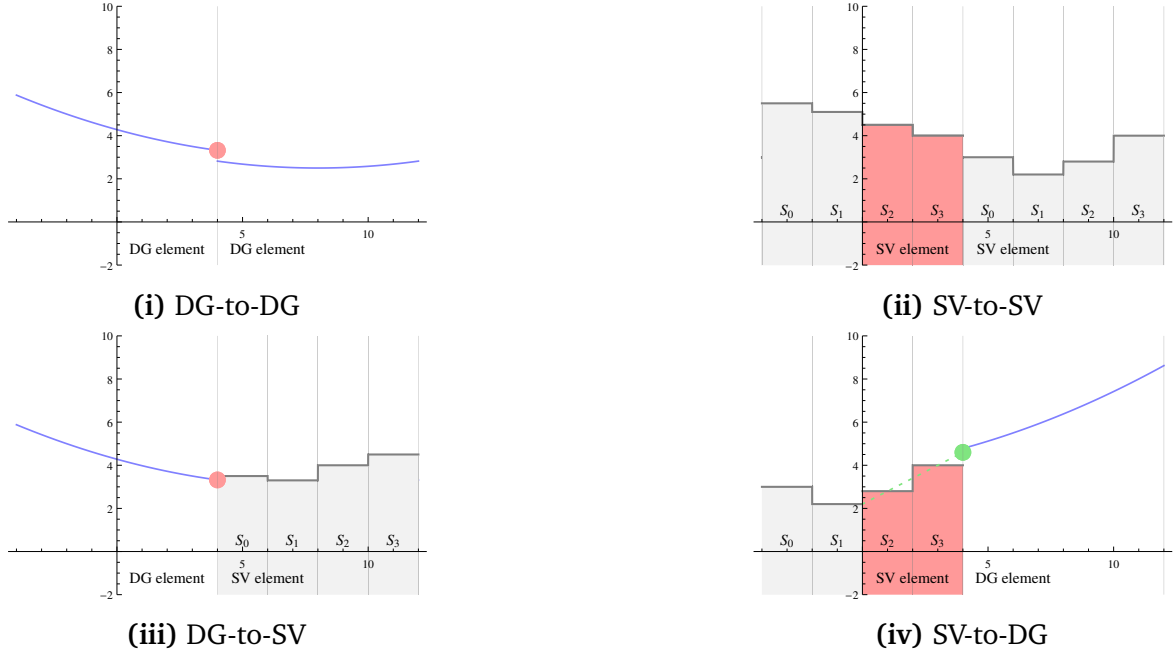
$$\begin{aligned} u_R(x_k^B) &= \bar{u}_k - \frac{1}{2} \text{minmod}(\bar{u}_{k+1} - \bar{u}_k, \bar{u}_k - \bar{u}_{k-1}) \\ u_L(x_k^B) &= \bar{u}_{k-1} + \frac{1}{2} \text{minmod}(\bar{u}_k - \bar{u}_{k-1}, \bar{u}_{k-1} - \bar{u}_{k-2}), \end{aligned} \quad (3.88)$$

where the minmod function (3.62) was used. Although this results only in a first order approximation of the boundary states, this method performs very reliably and is robust in the presence of shocks. In the case of the GRHD eqs. (2.35-37), we apply this limiting procedure on the primitive variables and subsequently calculate the appropriate conserved quantities. With these ingredients, the temporal change of the averages  $\bar{u}_k$  is computable by (3.87). A Runge-Kutta method is again used for the time integration. If an element changes its state to untroubled during the evolution, the DG polynomial has to be recovered from the SV averages. This is done by a linear interpolation to the collocation points, again combined with minmod slope limiting. Obviously, the influence of the SV evolution on the overall convergence order in a hybrid DG-SV scheme is questionable. In Chapter 4, a closer investigation is shown for several example setups.

### Patching DG and SV cells

As mentioned above, the SV scheme is only used in some elements, while in the main part of the domain a DG method is employed for the evolution. In practice we therefore encounter four possible types of communicating boundary data for neighboring elements, which are also depicted in Fig 3.5: i) DG element sends, DG element receives; ii) SV element sends, SV element receives; iii) DG element sends, SV element receives; iv) SV element sends, DG element receives. Case i) is the standard case present in a DG method, which we already discussed in Sec. 3.2. In case ii), two SV elements exchange boundary data. As seen from (3.88) for  $k = 0$ , the next two left averages  $\bar{u}_{-1}$  and  $\bar{u}_{-2}$  are necessary for the reconstruction of the leftmost boundary value. Therefore, communicating the boundary layer only is no longer sufficient for SV elements. Instead, the last two average layers have to be sent to the neighboring SV element. In case iii), the DG element sends only its boundary layer as usual, which is then directly interpreted as  $u_L(a)$  ( $u_R(b)$ ) if the DG element is the left (right) neighbor of the SV element. Since the average  $\bar{u}_{-1}$  ( $\bar{u}_{N+1}$ ) is missing for the reconstruction of  $u_R(a)$  ( $u_L(b)$ ), we do not limit the slope for this last point in the SV cell. This is not expected to cause additional problems, as shocks should never appear close to DG cells. Finally in case iv), the SV cell sends its last two average layers to the neighboring DG element. From these two averages, a boundary value is post-processed by linear reconstruction.



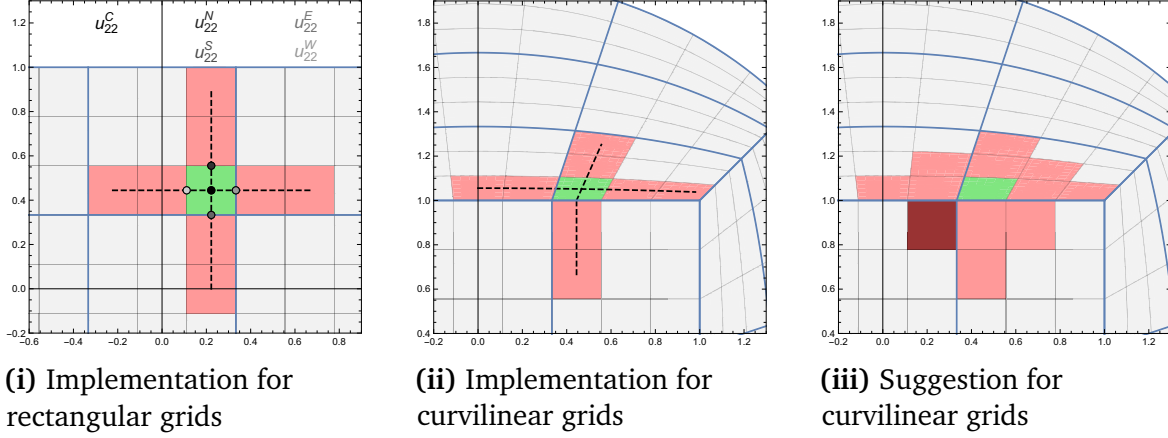


**Figure 3.5:** Possible types of boundary communication for adjacent grids. Polynomial DG data is colored blue, while SV averages are depicted as gray areas. Sent data are marked red and post-processing steps are shown in green. More details are given in the text.

### Multidimensional case

For a multidimensional generalization of the SV method, we simply divide the  $d$ -dimensional element into  $(N + 1)^d$   $d$ -dimensional spectral volumes. In Fig. 3.6, this division is shown for  $N + 1 = 3$  spectral volumes per element. In order to gain the SV averages from polynomial DG data, we use again the interpolation to the cell center, as suggested in (3.86). This interpolation is possible for general curvilinear elements without further modification. The local patch coordinates  $u^i$  are used for the interpolation on these grid patches. In what follows, we will extensively use the SV  $S_{kl}$  cell center point  $u^C$  and boundary center points  $u^N, u^E, u^S, u^W$ , which we define in local patch coordinates for the patch  $[\bar{u}_{\min}^1, \bar{u}_{\min}^1 + \Delta\bar{u}^1] \times [\bar{u}_{\min}^2, \bar{u}_{\min}^2 + \Delta\bar{u}^2]$ :

$$\begin{aligned}
 u_{kl}^C &:= \left( \bar{u}_{\min}^1 + \left(k + \frac{1}{2}\right) \frac{\Delta\bar{u}^1}{N+1}, \bar{u}_{\min}^2 + \left(l + \frac{1}{2}\right) \frac{\Delta\bar{u}^2}{N+1} \right) \\
 u_{kl}^N &:= \left( \bar{u}_{\min}^1 + \left(k + \frac{1}{2}\right) \frac{\Delta\bar{u}^1}{N+1}, \bar{u}_{\min}^2 + (l+1) \frac{\Delta\bar{u}^2}{N+1} \right) \\
 u_{kl}^E &:= \left( \bar{u}_{\min}^1 + (k+1) \frac{\Delta\bar{u}^1}{N+1}, \bar{u}_{\min}^2 + \left(l + \frac{1}{2}\right) \frac{\Delta\bar{u}^2}{N+1} \right) \\
 u_{kl}^S &:= \left( \bar{u}_{\min}^1 + \left(k + \frac{1}{2}\right) \frac{\Delta\bar{u}^1}{N+1}, \bar{u}_{\min}^2 + l \frac{\Delta\bar{u}^2}{N+1} \right) \\
 u_{kl}^W &:= \left( \bar{u}_{\min}^1 + k \frac{\Delta\bar{u}^1}{N+1}, \bar{u}_{\min}^2 + \left(l + \frac{1}{2}\right) \frac{\Delta\bar{u}^2}{N+1} \right), \quad 0 \leq k, l \leq N. \quad (3.89)
 \end{aligned}$$



**Figure 3.6:** Strategies for the two-dimensional SV method with three spectral volumes per element and direction ( $N = 2$ ). Spectral volumes are colored gray, while element boundaries are marked by blue lines. Assuming that (3.91) has to be evaluated for the green SV, the corresponding stencil of necessary average data is depicted in light red. Dashed black lines are curves along which linear reconstruction and limiting is performed. In (i), the SV center and SV boundary points are denoted. The dark red SV average in (iii) would not be accessible with our current parallelization strategy, if we implemented an improved version of the multidimensional SV scheme.

These points are also depicted in Fig. 3.6 (i). The two-dimensional counterpart of (3.87) can be derived by again integrating the balance law (2.35) and using the midpoint rule in local patch coordinates,

$$\begin{aligned}
 |S_{kl}| &:= \iint_{S_{kl}} dx dy = \det J(u_{kl}^C) \Delta \bar{u}^1 \Delta \bar{u}^2 + \mathcal{O}([\Delta \bar{u}^1 \Delta \bar{u}^2]^2) \\
 \bar{u}_{kl} &:= \frac{1}{|S_{kl}|} \iint_{S_{kl}} u(x, y) dx dy = u(u_{kl}^C) + \mathcal{O}(\Delta \bar{u}^1 \Delta \bar{u}^2) \\
 \iint_{S_{kl}} \partial_i f^i(x, y) dx dy &= [s_i^1 f^i(u_{kl}^E) - s_i^1 f^i(u_{kl}^W)] \Delta \bar{u}^2 \\
 &\quad + [s_i^2 f^i(u_{kl}^N) - s_i^2 f^i(u_{kl}^S)] \Delta \bar{u}^1 + \mathcal{O}([\Delta \bar{u}^1]^3, [\Delta \bar{u}^2]^3),
 \end{aligned} \tag{3.90}$$

where  $J_{ij}^i = \frac{\partial x^i}{\partial u^j}$  and  $s_i^j = (J^{-1})^j_i \det J$  as in Sec. 3.2. The final outcome is the generalization of (3.87) for general two-dimensional curvilinear grid patches:

$$\partial_t \bar{u}_{kl} + \frac{s_i^1 f^{*i}(u_{kl}^E) - s_i^1 f^{*i}(u_{kl}^W)}{\det J(u_{kl}^C) \Delta \bar{u}^1} + \frac{s_i^2 f^{*i}(u_{kl}^N) - s_i^2 f^{*i}(u_{kl}^S)}{\det J(u_{kl}^C) \Delta \bar{u}^2} = \bar{s}_{kl} \tag{3.91}$$

To feed the numerical flux function  $f^*$ , a reconstruction of the solution at the boundary center points has to be done. For this, a limiter should be applied as in the one-dimensional case. For rectangular elements, this procedure can be reduced to the one-dimensional reconstruction and limiting, which is applied in a direction-by-

direction fashion (see Fig. 3.6 (i)). However, a generalization to curved elements is not straightforward. What we actually implemented is a reconstruction along the  $u^i$  coordinate lines as shown in Fig. 3.6 (ii). However, at patch boundaries, these coordinate lines are non-smooth. Although we can not provide a clear proof, we expect that this causes additional errors. Unfortunately, the literature on finite volume methods for patched, curvilinear grids is sparse. Either patched quadrilateral grids are discussed as in [Barth and Jespersen, 1989], or curvilinear grids with a global coordinate transformation are considered [Bonnement, A. et al., 2011]. We expect however, that the reconstruction procedure from [Barth and Jespersen, 1989] is suitable for our grid setup. Here, a full two-dimensional reconstruction from all adjacent cells (not only along a coordinate line) is suggested. This however significantly increases the stencil of involved SV averages, as depicted in Fig. 3.6 (iii). Even more problematic, it breaks the parallelization strategy of the `bamps` code, as information of directly adjacent cells is not sufficient. In Fig. 3.6 (iii), the dark red colored SV average would not be part of the data exchange area of the green SV cell. Therefore, an additional communication step has to be included into `bamps`, possibly right after the first MPI exchange (see Fig. 3.1). Although we did not consider this development within this thesis, we definitely recommend its implementation in future iterations of the code.

An even more concerning issue of higher dimensionality is the non-trivial patching of DG and SV cells. The one-dimensional case which we discussed earlier, is special in that the shared interface is only one point. At this point, the numerical flux is uniquely determined. In higher dimensions, the interface is a line or a two-dimensional surface. In this case, data from the adjacent DG and SV cell is not directly available at a unique set of points on this surface. In our current implementation, we simply “average” data sent from the DG cell to make it consistent with the SV averages. On the other hand, we reconstruct the communicated SV data, to get collocation point data for the DG element. Although this approach yields the expected order of convergence, we encounter instabilities in multidimensional hybrid DG/SV simulations (see Sec. 4.3). In [Choi, 2015], this problem is addressed for a very similar numerical setup. According to [Choi, 2015], instabilities can arise from breaking conservation and outflow conditions at the interface. As a solution, the mortar method [Kopriva, 1996] is presented. This is based on the high order representation of both conservative fields from the adjacent grids and the unique numerical fluxes on the interface. A least squares projection is used to gain these high order interface polynomials and to transfer back the unique numerical flux to DG and SV cell. We think that a future implementation of the mortar method as discussed in [Choi, 2015] is needed to allow reliable, long-term hybrid DG/SV simulations with `bamps`.



# Chapter 4

## Code tests

In this chapter, we want to present tests and results of our implementation. Before we actually treat a binary system of neutron stars in Chapter 5, we check the functionality of our methods with a variety of testbeds of increasing complexity: Starting from very simple PDEs in flat spacetime, we check the fundamental convergence behavior of all newly implemented methods. Continuing with the equations of special relativistic hydrodynamics, we perform further convergence analysis on smooth problems and test the shock capturing ability of our implementation in shock tube simulations. As a next step toward GRHD, we consider the Michel-Bondi accretion [Michel, 1972] as a setup in a curved spacetime without shocks or an artificial atmosphere. In a series of TOV star evolutions, we investigate the code’s behavior in a “real life” application and discuss the influence of the artificial atmosphere. We finish the testing section with rotating neutron star tests, still with a static spacetime, and the migration test of an unstable TOV star, which comprises a truly dynamic, general relativistic testbed for our code.

### 4.1 Tests in flat spacetime

#### Advection equation

As a first test for the DG algorithm and the shock capturing methods, we consider the advection equation

$$\partial_t u + \partial_{\underline{i}} (v^{\underline{i}} u) = 0, \quad (4.1)$$

on the interval  $x \in [-3, 3]$  for an effectively one-dimensional Gaussian peak

$$u(x, 0) = A e^{(-x^2/\sigma^2)}, \quad (4.2)$$

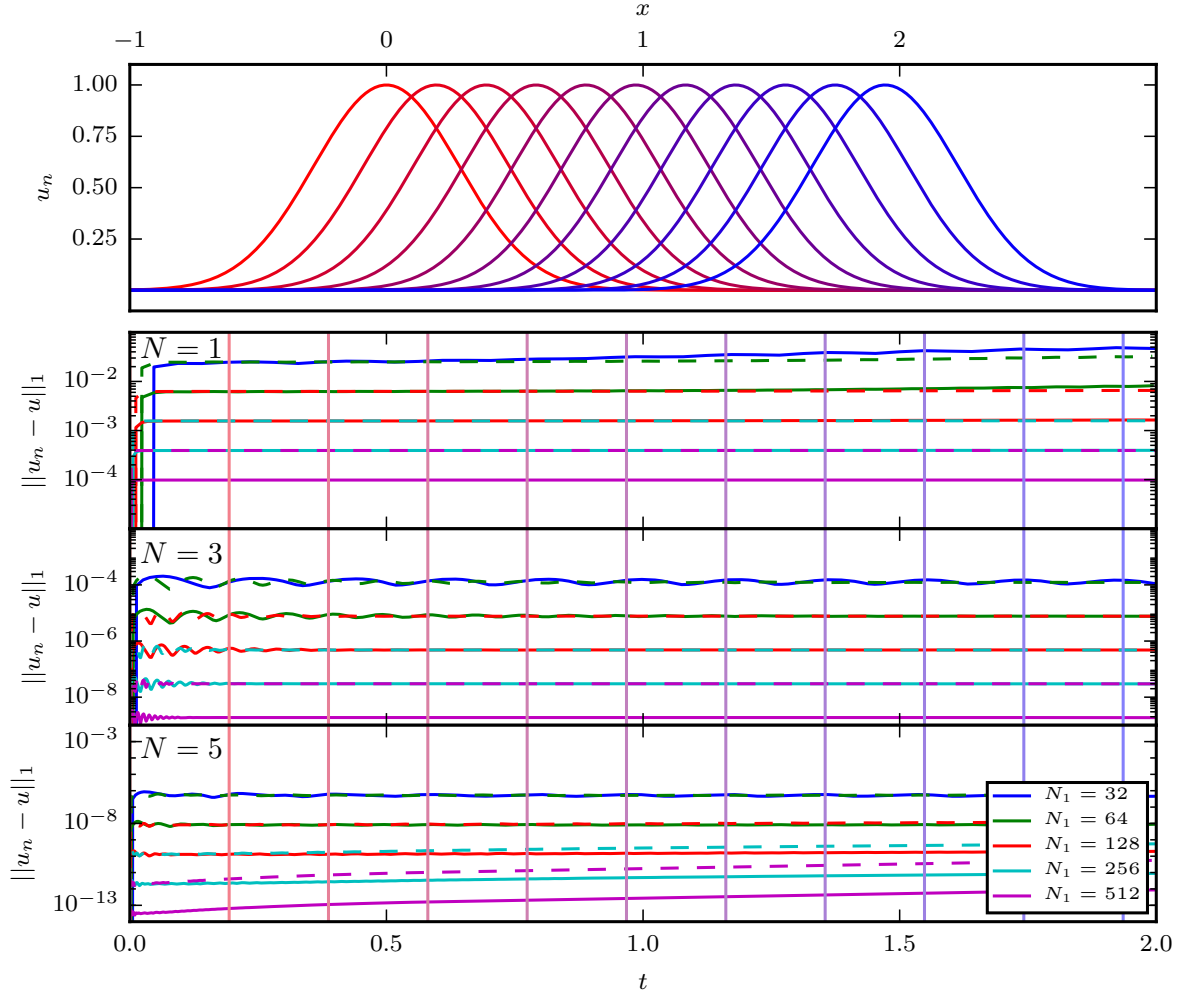
**Table 4.1:** Comparison of numerical methods introduced in Chapter 3: Errors and convergence orders for the advection equation (4.1) (initial state (4.2)) at  $t = 2.0$  for different polynomial order  $N$  and numbers of grid patches  $N_1$ . We chose  $M = 1$  for the WENO tests and  $u_{SV} = 0.1$  for the DG+SV run.

$N_1$	$N$	DG		DG + WENO-3		DG + WENO-5		DG + SV		SV only	
		$L_1$ error	order	$L_1$ error	order	$L_1$ error	order	$L_1$ error	order	$L_1$ error	order
16	1	$1.33 \cdot 10^{-1}$	-	$2.81 \cdot 10^{-1}$	-	$2.60 \cdot 10^{-1}$	-	$1.09 \cdot 10^{-1}$	-	$2.03 \cdot 10^{-1}$	-
32	1	$3.07 \cdot 10^{-2}$	2.11	$1.11 \cdot 10^{-1}$	1.33	$1.55 \cdot 10^{-1}$	0.74	$3.26 \cdot 10^{-2}$	1.74	$7.29 \cdot 10^{-2}$	1.48
64	1	$6.39 \cdot 10^{-3}$	2.26	$3.57 \cdot 10^{-2}$	1.64	$4.72 \cdot 10^{-2}$	1.71	$1.07 \cdot 10^{-2}$	1.61	$2.75 \cdot 10^{-2}$	1.40
128	1	$1.58 \cdot 10^{-3}$	2.01	$8.25 \cdot 10^{-3}$	2.11	$1.05 \cdot 10^{-2}$	2.16	$3.02 \cdot 10^{-3}$	1.82	$9.74 \cdot 10^{-3}$	1.49
256	1	$3.93 \cdot 10^{-4}$	2.00	$1.66 \cdot 10^{-3}$	2.30	$2.00 \cdot 10^{-3}$	2.39	$8.14 \cdot 10^{-4}$	1.89	$2.67 \cdot 10^{-3}$	1.86
512	1	$9.81 \cdot 10^{-5}$	2.00	$3.19 \cdot 10^{-4}$	2.38	$4.05 \cdot 10^{-4}$	2.30	$2.14 \cdot 10^{-4}$	1.92	$7.26 \cdot 10^{-4}$	1.88
16	3	$2.02 \cdot 10^{-3}$	-	$3.20 \cdot 10^{-1}$	-	$2.29 \cdot 10^{-1}$	-	$3.12 \cdot 10^{-2}$	-	$6.83 \cdot 10^{-2}$	-
32	3	$1.49 \cdot 10^{-4}$	3.75	$1.67 \cdot 10^{-1}$	0.93	$1.55 \cdot 10^{-2}$	3.88	$9.30 \cdot 10^{-3}$	1.74	$2.64 \cdot 10^{-2}$	1.37
64	3	$7.78 \cdot 10^{-6}$	4.26	$4.92 \cdot 10^{-2}$	1.76	$3.45 \cdot 10^{-4}$	5.49	$2.54 \cdot 10^{-3}$	1.86	$9.39 \cdot 10^{-3}$	1.49
128	3	$4.78 \cdot 10^{-7}$	4.02	$1.29 \cdot 10^{-2}$	1.93	$7.33 \cdot 10^{-6}$	5.55	$7.06 \cdot 10^{-4}$	1.85	$2.63 \cdot 10^{-3}$	1.83
256	3	$3.00 \cdot 10^{-8}$	3.99	$2.89 \cdot 10^{-3}$	2.15	$1.33 \cdot 10^{-7}$	5.77	$1.88 \cdot 10^{-4}$	1.90	$7.28 \cdot 10^{-4}$	1.85
512	3	$1.87 \cdot 10^{-9}$	3.99	$3.66 \cdot 10^{-4}$	2.98	$3.22 \cdot 10^{-9}$	5.37	$5.34 \cdot 10^{-5}$	1.82	$1.93 \cdot 10^{-4}$	1.91
16	5	$5.11 \cdot 10^{-5}$	-	$3.29 \cdot 10^{-1}$	-	$2.56 \cdot 10^{-1}$	-	$1.53 \cdot 10^{-2}$	-	$4.13 \cdot 10^{-2}$	-
32	5	$5.85 \cdot 10^{-7}$	6.44	$1.67 \cdot 10^{-1}$	0.97	$2.56 \cdot 10^{-2}$	3.32	$4.17 \cdot 10^{-3}$	1.88	$1.48 \cdot 10^{-2}$	1.48
64	5	$8.05 \cdot 10^{-9}$	6.18	$5.02 \cdot 10^{-2}$	1.73	$1.96 \cdot 10^{-3}$	3.70	$1.07 \cdot 10^{-3}$	1.95	$4.41 \cdot 10^{-3}$	1.74
128	5	$1.32 \cdot 10^{-10}$	5.92	$1.34 \cdot 10^{-2}$	1.90	$4.62 \cdot 10^{-5}$	5.40	$2.80 \cdot 10^{-4}$	1.93	$1.24 \cdot 10^{-3}$	1.82
256	5	$2.00 \cdot 10^{-12}$	6.04	$3.00 \cdot 10^{-3}$	2.16	$7.93 \cdot 10^{-7}$	5.86	$7.31 \cdot 10^{-5}$	1.94	$3.32 \cdot 10^{-4}$	1.89
512	5	$3.30 \cdot 10^{-13}$	2.60	$3.96 \cdot 10^{-4}$	2.92	$1.07 \cdot 10^{-8}$	6.19	$1.87 \cdot 10^{-5}$	1.96	$8.86 \cdot 10^{-5}$	1.90

and a rectangular pulse (non-smooth initial data)

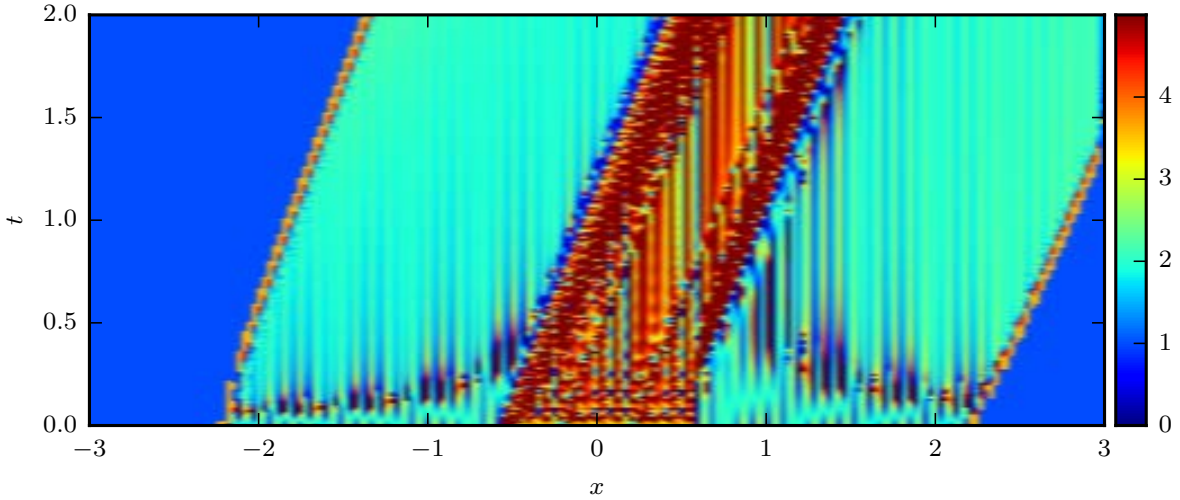
$$u(x, 0) = \begin{cases} 1 & \text{if } |x - x_0| < 0.5 \\ 0 & \text{else} \end{cases} \quad (4.3)$$

We first employ the DG method without any additional shock capturing methods. In Fig. 4.1, the time evolution, as well as the absolute difference between the numerical solution  $u_n$  and the available analytical solution  $u$  is shown for the first test case (4.2) ( $A = 1$ ,  $\sigma = 0.4$ ). From here on, all error integrals over the computational domain  $\|\Delta\|_1 = \int_{\Omega} |u| dx$  are determined by a Legendre-Gauss-Lobatto quadrature. Due to our choice of collocation points, we can directly use the function values for this integration. The expected convergence order  $N + 1$ , where  $N$  is the maximal order of the DG polynomial, is precisely observed. However, in an error regime beyond  $10^{-11}$ , we find a further drop in the convergence rates, because of the growing influence of truncation errors. When we employ shock capturing methods, as discussed in Sec. 3.3, the final convergence rate is influenced by several effects; see Table. 4.1. Applying the traditional WENO reconstruction procedure, we observe that the order of the full scheme is the result of the DG order  $N + 1$  and the order of the WENO method,  $2w + 1$ . Focusing on the  $N = 1$ -WENO-3 case, the convergence order is determined



**Figure 4.1:** Numerical solution and convergence test for the advection equation (4.1) with the DG method. The test is performed in one spatial dimension with  $v^i = (0.5, 0, 0)^T$  and outgoing boundary conditions. As initial state, a Gaussian pulse (4.2) ( $A = 1$ ,  $\sigma = 0.4$ ) is chosen. Top: Time evolution of the pulse for  $t \in [0, 1.94]$ . The numerical solution  $u_n$  is shown at 11 evenly distributed times. Bottom: Numerical errors and convergence for different polynomial order  $N$  and number of grid patches  $N_1$ . The solid line is the absolute difference of numerical and analytical solution, integrated over the domain ( $L_1$  norm). The dashed lines are scaled according to  $(N + 1)$ -th order convergence. Vertical lines are color coded in accordance with the top panel, indicating the times at which the numerical solution is shown.

predominantly by the second order DG scheme, while in the  $N = 3$ -WENO-3 case, the 3rd order reconstruction limits the order of the scheme. Choosing a WENO order higher than that of the DG scheme, we observe that convergence for small numbers of  $N_1$  is slower, but finally shows convergence above  $N + 1$ -th order. This can be explained by the decreasing influence of the WENO procedure for increasing  $N_1$ . The cell indicator only marks the cells around the maximum of the Gaussian peak as troubled, so the effective area in which the WENO reconstruction takes place decreases. Since the reconstruction has a strong smoothing effect with a local reconstruction of order  $w$ , the numerical results significantly differ from the analytic



**Figure 4.2:** Pointwise convergence order for the DG + SV solution of the advection equation (4.1) (initial state (4.2)) over time for polynomial order  $N = 3$  and numbers of grid patches  $N_{1,\text{low}} = 128$ ,  $N_{1,\text{high}} = 256$ . We chose  $u_{\text{SV}} = 0.1$ .

solution for small  $N_1$  and tend to the pure DG solution for large  $N_1$ . Applying the spectral volumes approach, second order convergence is observed throughout all runs. This is consistent with the underlying first order finite volume scheme. The absolute errors in the mixed DG+SV run are a factor of 3 – 5 smaller compared to the pure SV run. However, we do not observe the higher order DG convergence in the mixed run which leads to the conclusion that the inaccuracies of the SV scheme are dominant. A natural question is whether or not the error introduced by the lower order SV solution propagates over the grid and spoils the results in the higher order DG region. To investigate this, we tested pointwise convergence over time for the  $N = 3$ -DG+SV setup and show the results in Fig. 4.2. Since we chose  $u_{\text{SV}} = 0.1$  and the SV scheme is only activated for  $u_n$  below this threshold, the high order DG region “travels” with the Gaussian pulse. The two areas - one with second order convergence, where  $u_n < u_{\text{SV}}$  (SV scheme) and one with fourth order convergence, where  $u_n > u_{\text{SV}}$  (DG scheme) - are clearly visible. Moreover, this structure does not change during time evolution. It is remarkable that cells on the right side of the pulse show high order convergence once the DG scheme is employed, although second order errors were present in an earlier stage of the simulation. This could however also be an artifact of the simple characteristic structure of the advection equation, where all information travels with the same speed. From this test it is not clear that this structured convergence behavior is also conserved over time for a more complicated set of PDEs.

For the non-smooth test case Eq. (4.3), which we just want to summarize briefly, we observe larger total errors than for the smooth problem discussed above. Again the pure DG method errors are below the corresponding errors for the DG + shock capturing methods. Independent of the scheme we observe first order convergence, which is



consistent with the expectation for a non-smooth problem containing discontinuities.

In these and some of the following tests the DG method gives accurate results even when not combined with a shock capturing method, and is clearly also the most efficient method. More demanding situations involving shocks typically require methods to handle non-smoothness explicitly.

### Burgers equation

The Burgers equation

$$\partial_t u + \partial_x \left( \frac{1}{2} u^2 \right) = 0 \quad (4.4)$$

allows the formation of shocks from smooth initial data  $u_0$ . After the time

$$t_{\text{shock}} = - \left( \min \frac{\partial u_0}{\partial x} \right)^{-1} \quad (4.5)$$

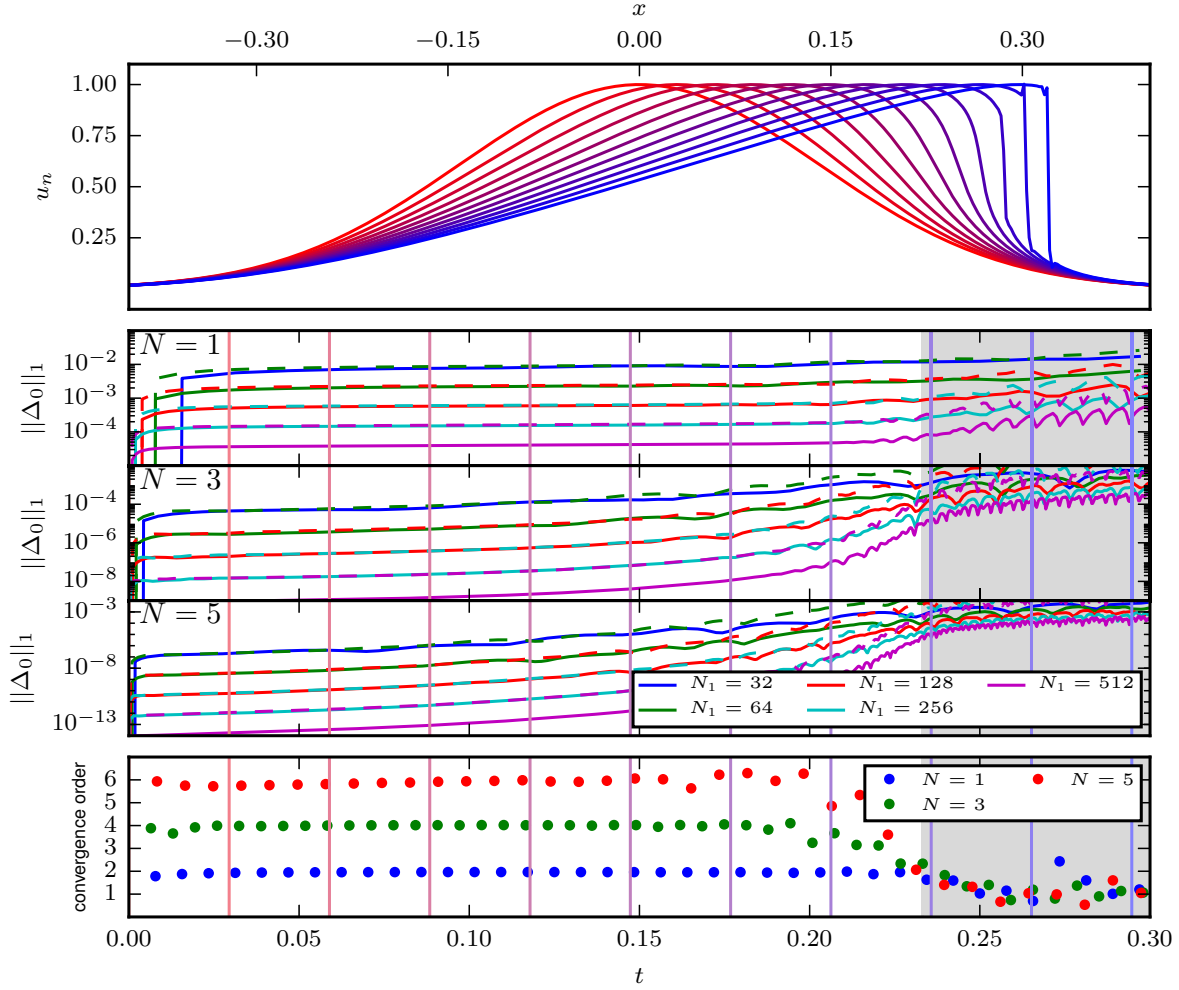
shocks will appear during the evolution. We use this as a testbed for our code and evolve the initial Gaussian peak (4.2) with  $A = 1$  and  $\sigma = 0.2$ . For this initial conditions, a shock forms at  $t_{\text{shock}} \approx 0.23316$ . Contrary to the advection problem, we have no immediate analytic solution  $u(x, t)$  at hand. To determine an error measure, we consider the following: Since the Burgers equation can be rewritten as  $(\partial_t + u\partial_x)u = 0$ , the solution  $u(x(t), t) = u(x_0, 0)$  is constant along the spacetime curve  $x(t) = x_0 + u(x_0, 0)t$ . With this insight, we evolve the numerical solution back in time to get a consistent counterpart of the numerical solution at  $t = 0$ :

$$u_n(x - u_n(x, t)t, 0) = u_n(x, t). \quad (4.6)$$

One can then use this back evolved solution to compare with the analytically given initial data and to give a meaningful error estimate:

$$\Delta_0 = |u(x - u_n(x, t)t, 0) - u_n(x, t)|. \quad (4.7)$$

As we would expect from our smooth and non-smooth advection equation results, we observe that the convergence rate decreases after  $t_{\text{shock}}$ ; see Fig. 4.3. Again, we find the expected convergence order of  $N + 1$  up to the shock formation time  $t_{\text{shock}}$ . Shortly before that time, convergence starts to drop for all  $N$  (gray shaded region) to approximately first order convergence. Although not shown in Fig. 4.3, the developing shock was captured by the troubled cell indicator when WENO was active. Both, the WENO and the SV method successfully avoided additional oscillations close to the discontinuity. More convincing results on the shock capturing performance are given in the SRHD shock tube test subsection.



**Figure 4.3:** Numerical solution and convergence test for the burgers equation (4.4). The test is performed in one spatial dimension and outgoing boundary conditions. As initial state, a Gaussian pulse (4.2) ( $A = 1$ ,  $\sigma = 0.2$ ) is chosen. Top: Time evolution of the pulse for  $t \in [0, 0.29]$ . The numerical solution  $u_n$  is shown at 11 evenly distributed times. Middle: Numerical errors for different polynomial order  $N$  and number of grid patches  $N_1$ . The solid line is the absolute difference of the back evolved numerical solution and the initial state (see text), integrated over the domain ( $L_1$  norm). The dashed lines are scaled according to  $(N + 1)$ -th order convergence. Vertical lines are color coded in accordance with the top panel, indicating the times at which the numerical solution is shown. Bottom: Convergence order calculated from the numerical errors of the  $N_1 = 128, 256$  runs for different polynomial order  $N$ . The gray shaded region indicates the time after shock formation.

In addition, we prepared the initial conditions

$$u(x, 0) = 0.5 + \sin(x\pi) \quad (4.8)$$

with periodic boundary conditions and check convergence at  $t = 0.5/\pi$  to compare with the results of [Qiu and Shu, 2005], Tab. 4.2 summarizes the results. Because  $t < t_{\text{shock}}$  in this example, the solution is still smooth and we observe again  $(N + 1)$ -th order convergence for the DG methods. As for the advection equation, we find higher

**Table 4.2:** Comparison of numerical methods introduced in Chapter 3: Errors and convergence orders for the Burgers equation (4.4) (initial state (4.8)) at  $t = 0.5/\pi$  for different polynomial order  $N$  and numbers of grid patches  $N_1$ . We chose  $M = 0.01$  for the WENO tests and  $u_{SV} = 0.1$  for the DG+SV run

$N_1$	$N$	DG		DG + WENO-3		DG + WENO-5		DG + SV		SV only	
		$L_1$ error	order	$L_1$ error	order	$L_1$ error	order	$L_1$ error	order	$L_1$ error	order
16	2	$3.33 \cdot 10^{-2}$	-	$6.88 \cdot 10^{-2}$	-	$5.76 \cdot 10^{-2}$	-	$3.33 \cdot 10^{-2}$	-	$1.54 \cdot 10^{-2}$	-
32	2	$1.03 \cdot 10^{-2}$	1.69	$1.90 \cdot 10^{-2}$	1.85	$1.64 \cdot 10^{-2}$	1.80	$6.89 \cdot 10^{-3}$	2.27	$4.96 \cdot 10^{-3}$	1.63
64	2	$3.17 \cdot 10^{-3}$	1.69	$5.02 \cdot 10^{-3}$	1.92	$4.73 \cdot 10^{-3}$	1.80	$1.91 \cdot 10^{-3}$	1.85	$1.28 \cdot 10^{-3}$	1.94
128	2	$9.44 \cdot 10^{-4}$	1.75	$1.40 \cdot 10^{-3}$	1.84	$1.29 \cdot 10^{-3}$	1.87	$5.06 \cdot 10^{-4}$	1.91	$3.31 \cdot 10^{-4}$	1.95
256	2	$2.73 \cdot 10^{-4}$	1.78	$3.53 \cdot 10^{-4}$	1.98	$3.44 \cdot 10^{-4}$	1.90	$1.31 \cdot 10^{-4}$	1.94	$8.43 \cdot 10^{-5}$	1.97
512	2	$7.78 \cdot 10^{-5}$	1.81	$9.32 \cdot 10^{-5}$	1.92	$9.20 \cdot 10^{-5}$	1.90	$3.37 \cdot 10^{-5}$	1.95	$2.15 \cdot 10^{-5}$	1.96
16	4	$2.26 \cdot 10^{-4}$	-	$4.40 \cdot 10^{-2}$	-	$1.88 \cdot 10^{-3}$	-	$7.83 \cdot 10^{-4}$	-	$4.99 \cdot 10^{-3}$	-
32	4	$1.70 \cdot 10^{-5}$	3.73	$1.05 \cdot 10^{-2}$	2.06	$9.66 \cdot 10^{-5}$	4.28	$1.51 \cdot 10^{-4}$	2.37	$1.30 \cdot 10^{-3}$	1.93
64	4	$1.22 \cdot 10^{-6}$	3.79	$2.34 \cdot 10^{-3}$	2.17	$4.55 \cdot 10^{-6}$	4.40	$2.60 \cdot 10^{-5}$	2.53	$3.55 \cdot 10^{-4}$	1.87
128	4	$8.29 \cdot 10^{-8}$	3.88	$4.27 \cdot 10^{-4}$	2.45	$2.07 \cdot 10^{-7}$	4.45	$6.32 \cdot 10^{-6}$	2.03	$9.30 \cdot 10^{-5}$	1.93
256	4	$5.44 \cdot 10^{-9}$	3.92	$2.26 \cdot 10^{-5}$	4.24	$8.02 \cdot 10^{-9}$	4.69	$1.45 \cdot 10^{-6}$	2.11	$2.45 \cdot 10^{-5}$	1.92
512	4	$3.57 \cdot 10^{-10}$	3.93	$9.76 \cdot 10^{-8}$	7.85	$3.70 \cdot 10^{-10}$	4.43	$3.35 \cdot 10^{-7}$	2.12	$6.31 \cdot 10^{-6}$	1.96
16	6	$3.01 \cdot 10^{-6}$	-	$4.44 \cdot 10^{-2}$	-	$2.65 \cdot 10^{-3}$	-	$2.60 \cdot 10^{-4}$	-	$2.02 \cdot 10^{-3}$	-
32	6	$6.14 \cdot 10^{-8}$	5.61	$1.01 \cdot 10^{-2}$	2.13	$1.72 \cdot 10^{-4}$	3.94	$4.92 \cdot 10^{-5}$	2.40	$4.79 \cdot 10^{-4}$	2.07
64	6	$1.16 \cdot 10^{-9}$	5.72	$2.27 \cdot 10^{-3}$	2.15	$9.89 \cdot 10^{-6}$	4.12	$9.38 \cdot 10^{-6}$	2.39	$1.34 \cdot 10^{-4}$	1.83
128	6	$2.42 \cdot 10^{-11}$	5.58	$4.21 \cdot 10^{-4}$	2.43	$5.30 \cdot 10^{-7}$	4.22	$2.28 \cdot 10^{-6}$	2.03	$3.62 \cdot 10^{-5}$	1.89
256	6	$5.03 \cdot 10^{-13}$	5.58	$2.67 \cdot 10^{-5}$	3.97	$1.27 \cdot 10^{-8}$	5.38	$5.30 \cdot 10^{-7}$	2.10	$9.38 \cdot 10^{-6}$	1.95
512	6	$2.56 \cdot 10^{-14}$	4.29	$8.23 \cdot 10^{-8}$	8.34	$7.40 \cdot 10^{-11}$	7.42	$1.19 \cdot 10^{-7}$	2.14	$2.40 \cdot 10^{-6}$	1.96

than the expected rate of convergence when the WENO reconstruction is applied. This is again caused by the fact that a larger number of cells decrease the effective area, in which the reconstruction is performed. Although our errors are slightly higher than in [Qiu and Shu, 2005], the same effect is present in their results. For the SV method we find again second order convergence, as expected.

### Smooth 1D SRHD

We now consider the GRHD conservation law eqs. (2.35-37) without source terms,  $\alpha = 1$ ,  $\beta^i = 0$  and  $\gamma_{ij} = \delta_{ij}$ , for simulating the special relativistic test case, i.e. flat spacetime. Regarding the EOS, we chose an ideal gas EOS and set  $\Gamma = \frac{5}{3}$ . As a first test, we consider a smooth sine wave propagating with constant speed:

$$\begin{aligned}
 \rho(x, t) &= 1 + 0.2 \sin(2\pi(x - v_x t)) \\
 v_x(x, t) &= 0.2 \\
 p(x, t) &= 1
 \end{aligned} \tag{4.9}$$

inside the periodic 1D domain  $x \in [-1, 1]$  divided into  $N_1$  uniform grid patches. Viewing the summed  $L_1$  errors for all primitive variables and convergence rates (Tab. 4.3), we again find the convergence rate of the DG scheme to be  $N + 1$ . We

**Table 4.3:** Comparison of numerical methods introduced in Chapter 3: Errors and convergence orders for the smooth special relativistic setup (4.9) for different polynomial order  $N$  and numbers of grid patches  $N_1$ . We chose  $M = 5$  for the WENO tests and  $u_{SV} = 0.9$  for the DG+SV run. In DG setup (a) the analytic mass matrix (3.58) was used, while in DG setup (b) the mass lumped version (3.57) is employed.

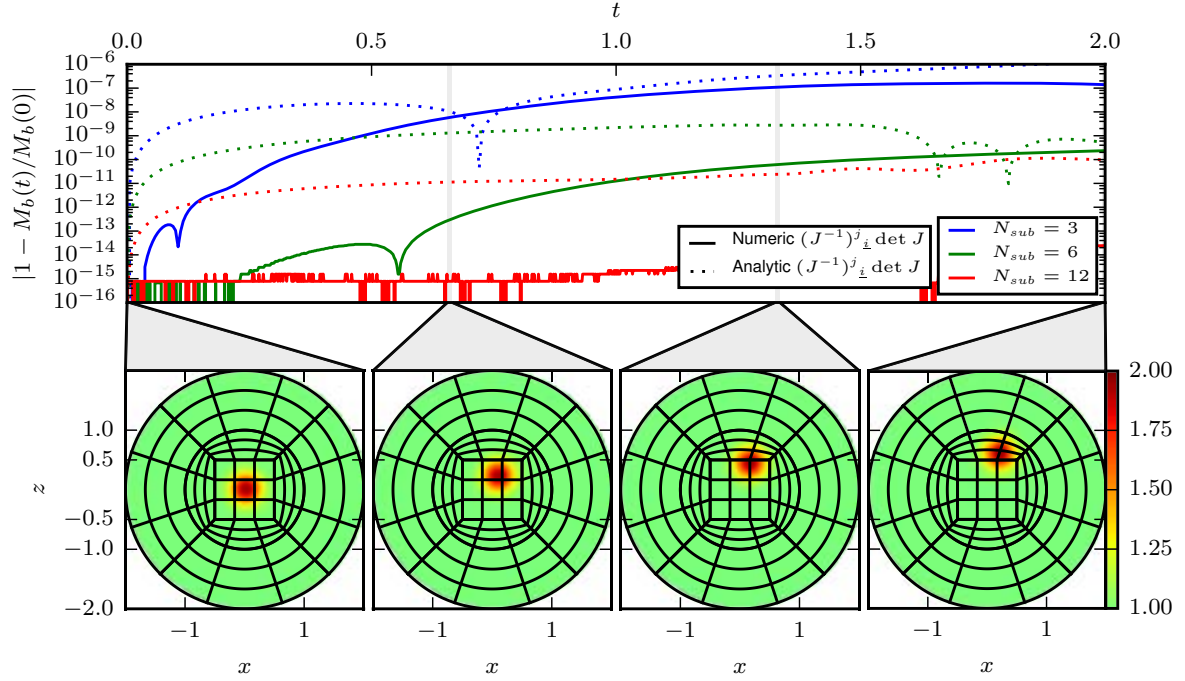
$N_1$	$N$	DG (a)		DG (b)		DG + WENO-3		DG + WENO-5		DG + SV		SV only	
		$L_1$ error	order	$L_1$ error	order	$L_1$ error	order	$L_1$ error	order	$L_1$ error	order	$L_1$ error	order
16	1	$4.64 \cdot 10^{-3}$	-	$5.86 \cdot 10^{-2}$	-	$2.11 \cdot 10^{-2}$	-	$2.71 \cdot 10^{-2}$	-	$7.09 \cdot 10^{-2}$	-	$7.85 \cdot 10^{-2}$	-
32	1	$9.79 \cdot 10^{-4}$	2.24	$1.54 \cdot 10^{-2}$	1.92	$7.71 \cdot 10^{-3}$	1.45	$9.05 \cdot 10^{-3}$	1.58	$2.15 \cdot 10^{-2}$	1.72	$2.73 \cdot 10^{-2}$	1.52
64	1	$2.38 \cdot 10^{-4}$	2.03	$3.90 \cdot 10^{-3}$	1.98	$1.52 \cdot 10^{-3}$	2.34	$1.95 \cdot 10^{-3}$	2.21	$6.86 \cdot 10^{-3}$	1.64	$1.03 \cdot 10^{-2}$	1.40
128	1	$5.93 \cdot 10^{-5}$	2.00	$1.01 \cdot 10^{-3}$	1.93	$2.58 \cdot 10^{-4}$	2.55	$3.45 \cdot 10^{-4}$	2.49	$1.95 \cdot 10^{-3}$	1.81	$2.99 \cdot 10^{-3}$	1.78
256	1	$1.48 \cdot 10^{-5}$	2.00	$2.59 \cdot 10^{-4}$	1.97	$5.48 \cdot 10^{-5}$	2.23	$5.82 \cdot 10^{-5}$	2.56	$5.45 \cdot 10^{-4}$	1.84	$8.24 \cdot 10^{-4}$	1.86
512	1	$3.70 \cdot 10^{-6}$	2.00	$6.49 \cdot 10^{-5}$	1.99	$1.15 \cdot 10^{-5}$	2.24	$1.04 \cdot 10^{-5}$	2.47	$2.04 \cdot 10^{-4}$	1.41	$2.22 \cdot 10^{-4}$	1.89
16	3	$2.27 \cdot 10^{-5}$	-	$5.75 \cdot 10^{-5}$	-	$3.80 \cdot 10^{-2}$	-	$2.72 \cdot 10^{-3}$	-	$2.14 \cdot 10^{-2}$	-	$2.66 \cdot 10^{-2}$	-
32	3	$1.71 \cdot 10^{-6}$	3.73	$2.94 \cdot 10^{-6}$	4.28	$1.59 \cdot 10^{-2}$	1.25	$6.41 \cdot 10^{-5}$	5.40	$7.10 \cdot 10^{-3}$	1.59	$1.06 \cdot 10^{-2}$	1.32
64	3	$7.29 \cdot 10^{-8}$	4.55	$1.81 \cdot 10^{-7}$	4.02	$3.12 \cdot 10^{-3}$	2.35	$1.16 \cdot 10^{-6}$	5.78	$2.00 \cdot 10^{-3}$	1.82	$3.03 \cdot 10^{-3}$	1.80
128	3	$4.71 \cdot 10^{-9}$	3.95	$1.10 \cdot 10^{-8}$	4.03	$4.52 \cdot 10^{-4}$	2.78	$2.00 \cdot 10^{-8}$	5.85	$5.28 \cdot 10^{-4}$	1.92	$8.23 \cdot 10^{-4}$	1.88
256	3	$2.93 \cdot 10^{-10}$	4.00	$6.85 \cdot 10^{-10}$	4.00	$1.62 \cdot 10^{-5}$	4.80	$5.62 \cdot 10^{-10}$	5.15	$1.43 \cdot 10^{-4}$	1.87	$2.23 \cdot 10^{-4}$	1.88
512	3	$1.85 \cdot 10^{-11}$	3.98	$4.30 \cdot 10^{-11}$	3.99	$1.33 \cdot 10^{-7}$	6.92	$2.98 \cdot 10^{-11}$	4.23	$3.79 \cdot 10^{-5}$	1.91	$5.92 \cdot 10^{-5}$	1.91
16	5	$2.87 \cdot 10^{-8}$	-	$5.52 \cdot 10^{-8}$	-	$4.45 \cdot 10^{-2}$	-	$7.53 \cdot 10^{-3}$	-	$1.20 \cdot 10^{-2}$	-	$1.67 \cdot 10^{-2}$	-
32	5	$4.61 \cdot 10^{-10}$	5.96	$9.84 \cdot 10^{-10}$	5.81	$1.76 \cdot 10^{-2}$	1.33	$3.63 \cdot 10^{-4}$	4.37	$3.37 \cdot 10^{-3}$	1.83	$5.15 \cdot 10^{-3}$	1.69
64	5	$6.35 \cdot 10^{-12}$	6.18	$1.52 \cdot 10^{-11}$	6.01	$4.61 \cdot 10^{-3}$	1.93	$7.11 \cdot 10^{-6}$	5.67	$9.19 \cdot 10^{-4}$	1.87	$1.39 \cdot 10^{-3}$	1.88
128	5	$3.65 \cdot 10^{-13}$	4.12	$4.55 \cdot 10^{-13}$	5.06	$7.21 \cdot 10^{-4}$	2.67	$1.02 \cdot 10^{-7}$	6.11	$2.44 \cdot 10^{-4}$	1.91	$3.81 \cdot 10^{-4}$	1.87
256	5	$4.59 \cdot 10^{-13}$	-	$3.90 \cdot 10^{-13}$	0.22	$2.48 \cdot 10^{-5}$	4.86	$5.87 \cdot 10^{-10}$	7.44	$6.58 \cdot 10^{-5}$	1.89	$1.01 \cdot 10^{-4}$	1.90
512	5	$5.43 \cdot 10^{-13}$	-	$5.39 \cdot 10^{-13}$	-	$1.12 \cdot 10^{-7}$	7.78	$1.26 \cdot 10^{-12}$	8.85	$1.72 \cdot 10^{-5}$	1.93	$2.68 \cdot 10^{-5}$	1.92

collaborated with the authors of [Kidder et al., 2016] who published SpECTRE, a task-based DG code for relativistic astrophysics. We compare numerical results from both codes for this specific example, which is an important code validation. Looking at Tab. B.1 in [Kidder et al., 2016] and Tab. IV in [Bug1], we find excellent agreement between the two codes. Comparing actual numbers, the difference is not larger than  $\approx 5 \cdot 10^{-12}$ . During this code comparison, we realized that using the analytic mass matrix leads to smaller errors than those from the mass lumped (diagonal) matrix scheme, as expected. However, the difference is comparatively small for higher order  $N$ . This is evident from columns DG (a) and DG (b) in Tab. 4.3. Looking at all other columns of this table, the observations from the former scalar PDE tests are confirmed for the case of a system.

The smooth flow SRHD setup is also a convenient testbed for the mass conservation properties of the scheme. In Sec. 3.2, we also discussed the importance of numerically fulfilling the metric identities. The following test substantiates this hypothesis. We again prepare smooth initial data

$$\begin{aligned}
 \rho(x, z, t = 0) &= 1 + \exp \left[ -\frac{x^2 + z^2}{0.25^2} \right] \\
 v(x, z, t = 0) &= (0.1, 0.0, 0.3) \\
 p(x, z, t = 0) &= 1,
 \end{aligned} \tag{4.10}$$

that propagates with constant speed over the 2D domain  $x^2 + z^2 < 4$ . This domain is covered by the curvilinear bumps grid. Revisiting the GRHD equations eqs. (2.35-37),



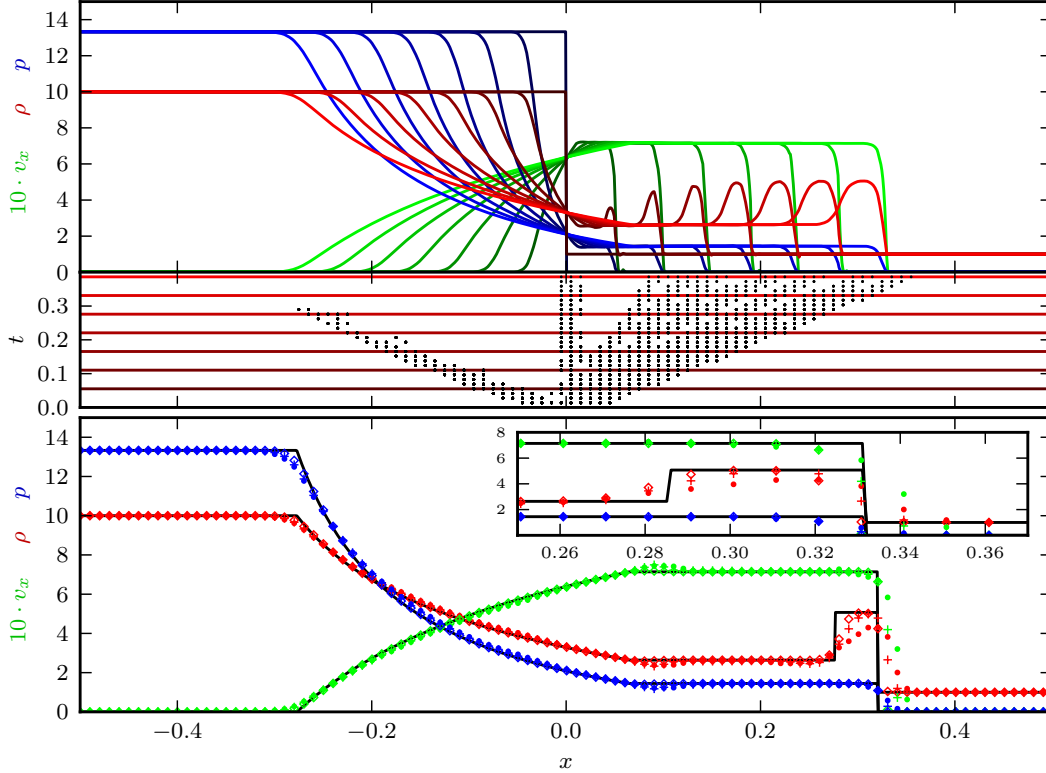
**Figure 4.4:** Testing mass conservation and influence of metric identities on the DG scheme. Top: The relative baryonic mass change for polynomial order  $N = 3$ , different number of grid patches  $N_{\text{sub}}$  and different methods to calculate the Jacobian. When the metric identities are fulfilled numerically (solid line), mass conservation violation is rapidly decreasing to machine precision as resolution is increased. Bottom: The dynamics of the smooth flow density  $\rho$  for initial data (4.10) and  $N_{\text{sub}} = 3$ . The subpatch boundaries are marked by the thick black lines.

we see that the variable  $D$  fulfills a strict conservation law, i.e. the *baryonic mass*

$$M_b := \int_{\Omega} \sqrt{\gamma} \rho W d^3x \stackrel{\text{SRHD}}{=} \int_{\Omega} \rho W d^3x \quad (4.11)$$

does not change over time, as long as no matter leaves the domain  $\Omega$ . By evolving initial data (4.10), we test the numerical conservation of baryonic mass and show the results in Fig. 4.4. Obviously, conservation is improved by increasing the resolution, i.e. increasing the number of subpatches  $N_1 = N_3 = N_{\text{sub}}$ . Apart from that, we see a clear improvement of mass conservation when a Jacobian that exactly fulfills the metric identities is used (solid lines). Compared to the analytical Jacobian (dotted lines), mass conservation is orders of magnitude better for high resolution. Even though the pulse is traveling through strongly distorted grid patches, mass conservation is perfectly fulfilled up to machine precision already for  $N_{\text{sub}} = 12$ .

From these test examples, we can conclude that the DG and the SV scheme do clearly show the expected convergence behavior, as long as the numerical solution is smooth. In the case of Burgers equation, convergence rates drop as a shock develops. In the following subsection, we want to give a closer investigation of the shock capturing techniques.



**Figure 4.5:** The one-dimensional special relativistic Riemann problem. For all runs we chose  $N = 3$ ,  $N_1 = 100$ , an ideal gas EOS with  $\Gamma = 5/3$  and  $M = 50$  for the troubled cell indication. Top: Numerical evolution of initial data (4.12) with the SV method (density: red, velocity: green, pressure: blue), time proceeds from dark to light colored lines. Middle: The troubled cells in a spacetime diagram for the DG+WENO-5 method, indicated by black points. Horizontal lines are color coded in accordance with the top panel, indicating the times at which the numerical solution for  $\rho$  is shown. Bottom: Numerical results using the DG+WENO-3 (dots), DG+WENO-5 (crosses) and the SV method (diamonds), compared to the analytical solution (black line) at  $t = 0.4$ .

### SRHD shock tests

Again for the GRHD conservation law eqs. (2.35-37) in flat spacetime, we consider the one-dimensional Riemann problem

$$(\rho, v_x, p)(x, t = 0) = \begin{cases} (10, 0.0, 13.33) & x < 0 \\ (1, 0.0, 10^{-7}) & x \geq 0, \end{cases} \quad (4.12)$$

on the domain  $x \in [-1, 1]$ . The analytical solution for this problem in the context of SRHD is given by [Martí and Müller, 1994]. This is considered a standard test for shock capturing schemes and we can directly compare our results with [Zhao and Tang, 2013; Radice and Rezzolla, 2011]. Qualitatively, the evolution shows a shock wave traveling through a low density medium (see top panel of Fig. 4.5). The analytical solution consists of a left traveling rarefaction wave, a contact discontinuity and the right traveling shock wave. During our tests, we observe the troubled cell

indicator to work reliably (see middle panel of Fig. 4.5). The subpatches containing shocks, discontinuities or kinks in the solution are marked as troubled. After a timestep, only 10-25% of all cells are troubled cells for  $M = 50$ . In all other cells the pure DG scheme is used for the evolution. This ratio directly depends on the parameter  $M$ , that characterizes the modified minmod function (3.63). As expected, choosing a lower  $M$  leads to a larger number of troubled cells. In the shock tube tests, we use the SV method globally (instead of hybrid DG+SV), because DG fails for all threshold values  $\rho_{SV}$ , as the shock is not restricted to low density regions. Looking at the bottom panel of Fig. 4.5, we see that the computed solution agrees nicely with the analytical solution for the three different shock capturing methods (WENO-3, WENO-5 and SV). We usually apply the reconstruction methods on the conserved quantities. However, in the 1D and 2D SRHD shock tests, the results are slightly improved by reconstructing the characteristic variables. Fewer cells are then marked as troubled and shocks are sharper in the numerical simulation. This is also apparent from Fig. 4.6, where conservative reconstruction is directly compared with characteristic reconstruction for the two-dimensional cases. The necessary left and right eigenvectors are given in [Banyuls et al., 1997; Rezzolla and Zanotti, 2013]. All wave features of the analytical solution are well resolved. The numerical solution shows no oscillatory artifacts at discontinuities. This comes at the price of clear smoothing effects close to the shocks and kinks. From these tests we see that the smoothing effect is smaller for the DG+WENO-5 than for the DG+WENO-3 evolution. This is somehow expected from the higher order of the reconstruction. However, the SV method appears to resolve the shocks slightly better than either WENO method.

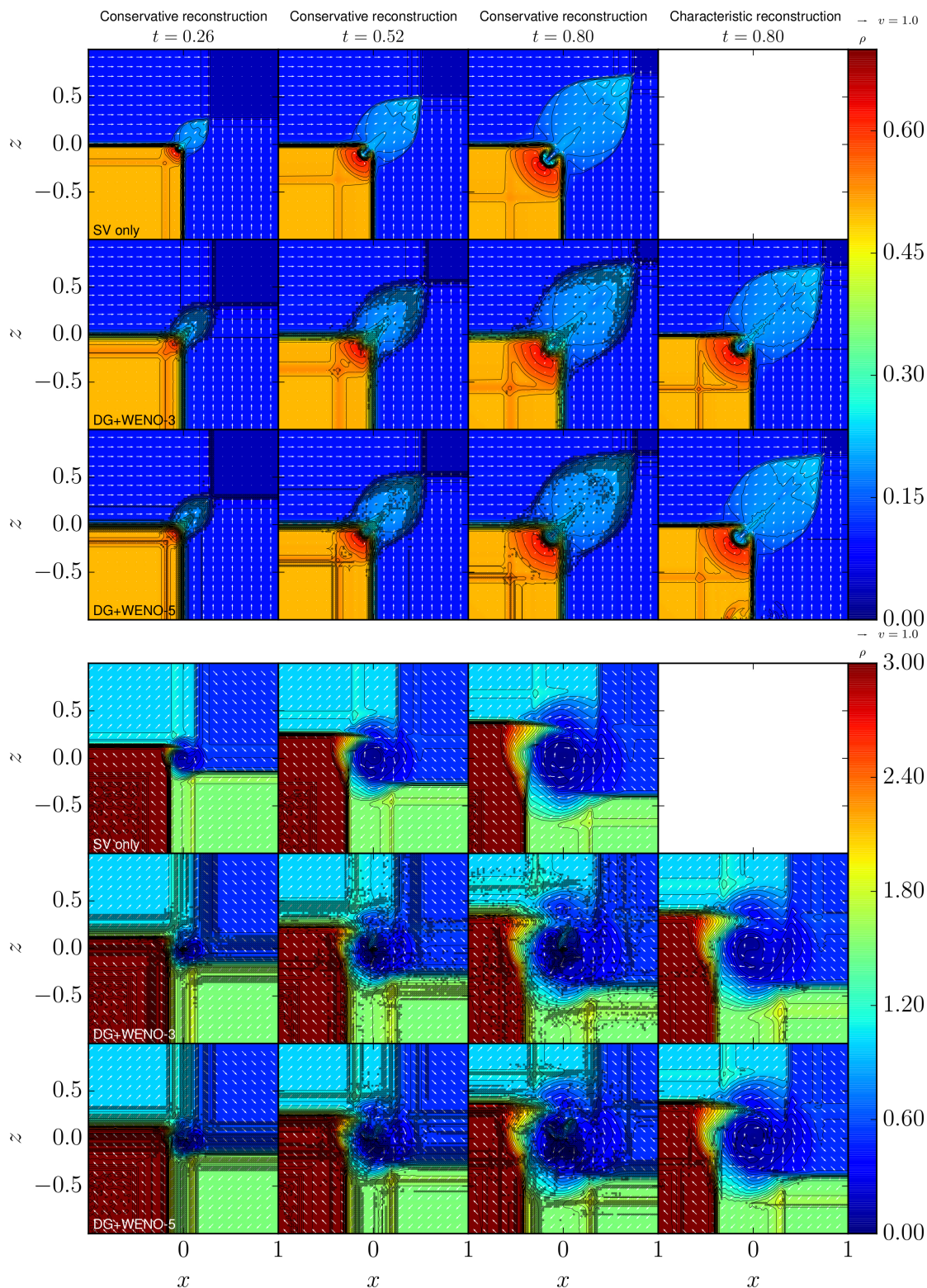
To conclude the tests in flat spacetime, we consider two-dimensional special relativistic shock problems: One as a 2D generalization of the formerly presented 1D Riemann problem

$$(\rho, v_x, v_z, p)(x, z, t = 0) = \begin{cases} (0.03515, 0, 0, 0.163) & x \geq 0, z \geq 0 \\ (0.1, 0.7, 0, 1) & x < 0, z \geq 0 \\ (0.5, 0, 0, 1) & x < 0, z < 0 \\ (0.1, 0, 0.7, 1) & x \geq 0, z < 0 \end{cases}, \quad (4.13)$$

and a vortex sheet problem

$$(\rho, v_x, v_z, p)(x, z, t = 0) = \begin{cases} (0.5, 0.5, -0.5, 5.0) & x \geq 0, z \geq 0 \\ (1, 0.5, 0.5, 5.0) & x < 0, z \geq 0 \\ (3.0, -0.5, 0.5, 5.0) & x < 0, z < 0 \\ (1.5, -0.5, -0.5, 5.0) & x \geq 0, z < 0 \end{cases}, \quad (4.14)$$





**Figure 4.6:** Two-dimensional special relativistic shock problems. For all runs we chose  $N = 3$ ,  $N_1 = N_3 = 100$ , an ideal gas EOS with  $\Gamma = 5/3$  and  $M = 50$  for the troubled cell indication. Top: The two-dimensional shock problem (4.13). Bottom: The vortex problem (4.14). For the shock capturing, the SV (top row), WENO-3 (middle row) and WENO-5 (bottom row) methods are used. Time evolution is shown from the left to the right for three different times. For the DG+WENO methods, the final state at  $t = 0.8$  is shown with reconstruction applied on the (i) conservative variables; (ii) characteristic variables. Density (color coding and contour lines), velocity field (arrows) and troubled cells (shaded circles) are depicted. Adapted from [Bug1].



on the domain  $(x, z) \in [-1, 1] \times [-1, 1]$ . We found these two test cases in [Zhao and Tang, 2013]. We again tested the DG+WENO-3, DG+WENO-5 and the SV scheme. Fig. 4.6 shows that all these shock capturing methods give qualitatively the same results. During the evolution of both setups all initial discontinuities are captured by the troubled cell indicator (shaded circles). As in the one-dimensional tests, we apply the SV scheme globally. In the case of the shock problem (top part of Fig. 4.6), the SV method resolves steep gradients better than the traditional WENO reconstruction. This becomes most noticeable in the “mushroom cloud” area around  $x = z = -0.2$ . However due to the larger computational expenses of the SV method and its application on the whole grid, the SV scheme is a factor of  $\approx 2.4$  times slower than the DG+WENO methods. For the vortex test (bottom), all methods are again able to resolve the structure properly. The SV method gives more accurate results, i.e. shocks remain sharp and spatially local, but is again  $\approx 3.2$  times slower than the DG+WENO implementations. When we compare columns (a) and (b) in Fig. 4.6, we see that characteristic reconstruction gives much cleaner results, particularly in the vortex tests. The conservative reconstruction almost fails around  $x = z = 0$  and many more cells are marked troubled. We can directly compare our results from the characteristic reconstruction runs with those in [Zhao and Tang, 2013]. The numerical solution and the distribution of troubled cells looks qualitatively very similar.

## 4.2 Spherical accretion

As a first application away from flat space, we chose the steady state solution for the accretion of matter onto a black hole [Michel, 1972]. This solution is still smooth and the spacetime is assumed to be static. In our tests, we will therefore only evolve the matter variables on a fixed curved spacetime background. In what follows, we denote this the *Cowling approximation* [Cowling, 1941]. Although this example constitutes only a slight increase of complexity compared to the previous tests, it allows us to validate the correct implementation of non-trivial metric quantities.

### Initial data

In [Michel, 1972], the equations of motion for a spherical symmetric flow of matter towards a compact object have been solved for a static spacetime. Since this solution is also static, it serves us as both initial data for the simulation and the analytic solution for error estimates. We want to give the main steps of its derivation. We start again from the conservation laws of mass and energy,

$$\nabla_{\mu} \rho u^{\mu} = \partial_r (\sqrt{-g} \rho u^r) = 0, \quad \nabla_{\mu} T_0^{\mu} = \partial_r (\sqrt{-g} T_0^r) - \Gamma_{0\mu}^{\nu} T_{\nu}^{\mu} = \partial_r (\sqrt{-g} T_0^r), \quad (4.15)$$

where (2.27) and the exclusive dependence on the radial coordinate  $r$  is used. The last equality is found by exploiting symmetry and time-independence of  $T^{\mu\nu}$ . With the perfect fluid energy-momentum tensor (2.21), the integrated equations

$$\sqrt{-g}\rho u^r = C_1 = \text{const} \quad (4.16)$$

$$\sqrt{-g}\rho h u^r u_0 = C_2 = \text{const} \quad (4.17)$$

and the combined equation

$$h^2 u_0^2 = C_3 = \text{const} \quad (4.18)$$

are obtained. As the background spacetime, the Schwarzschild solution in Kerr-Schild coordinates is considered:

$$ds^2 = - \left(1 - \frac{2M}{r}\right) dt^2 + \frac{4M}{r} dt dr + \left(1 + \frac{2M}{r}\right) dr^2 + r^2 d\Omega^2. \quad (4.19)$$

From that and the normalization of  $u^\mu$  one finds

$$u_0 = \sqrt{-g_{tt} + (u^r)^2}, \quad \sqrt{-g} = r^2 \sin \theta \quad (4.20)$$

for this metric. Furthermore, the 3+1 quantities are easily read off from (4.19):

$$\alpha = \sqrt{\frac{r}{r+2M}}, \quad \beta^r = \frac{2M}{r+2M}, \quad \beta_r = \frac{2M}{r}, \quad \gamma_{ij} = \text{diag} \left(1 + \frac{2M}{r}, r^2, r^2 \sin^2 \theta\right). \quad (4.21)$$

It turns out that the equations of motion, eqs. (4.16-18), are characterized by a critical point and only solutions that pass through this point are physically meaningful. This can be seen by plugging (4.20) into eqs. (4.16-18) and differentiating. The form

$$\frac{du^r}{u^r} \left( V - \frac{(u^r)^2}{1 - \frac{2M}{r} + (u^r)^2} \right) + \frac{dr}{r} \left( 2V - \frac{M}{r \left(1 - \frac{2M}{r} + (u^r)^2\right)} \right) = 0 \quad (4.22)$$

with  $V = \frac{dh}{d\rho} \frac{\rho}{h}$  is obtained, suggesting that at the critical point both bracketed expressions should be zero to guarantee monotone inflow. Given a critical point at radius  $r_c$ , this determines the radial inflow velocity at  $r_c$ ,

$$u_c^r = \frac{M}{2r_c}, \quad (4.23)$$

and some properties of the matter via

$$V_c = \frac{M}{2r \left(1 - \frac{2M}{r} + (u^r)^2\right)} = \frac{(u_c^r)^2}{1 - 3(u_c^r)^2}. \quad (4.24)$$

Choosing the polytropic EOS  $p = K\rho^\Gamma$  yields

$$h = 1 + \frac{K\Gamma\rho^{\Gamma-1}}{\Gamma-1}, \quad V = \frac{K\Gamma\rho^\Gamma(\Gamma-1)}{K\Gamma\rho^\Gamma + (\Gamma-1)\rho} \Rightarrow K = \frac{V(\Gamma-1)}{(\Gamma-1-V)\Gamma\rho^{\Gamma-1}}, \quad (4.25)$$

so that by specifying  $K$ , the critical density at  $r_c$  is predetermined. In our tests however, we specify the critical density  $\rho_c$  and compute  $K$  in accordance with (4.25). The complete preparation of initial data is done as explained in [Papadopoulos and Font, 1998] and goes as follows: i) Choose  $(r_c, \rho_c)$  and  $\Gamma$  for the polytropic gas; ii) Compute  $u_c^r$  from (4.23) and  $K$  from eqs. (4.24,25); iii) Compute the constants  $C_1$  and  $C_3$  from eqs. (4.16,18); iv) Apply a Newton-Raphson root finder to solve (4.18) for  $u_r$ :

$$f(u^r) = h^2 u_0^2 \stackrel{(4.16)}{=} \left[ 1 + \frac{K\Gamma \left( \rho_c \frac{r_c^2 u_c^r}{r^2 u^r} \right)^{\Gamma-1}}{\Gamma-1} \right]^2 \left[ 1 - \frac{2M}{r} + (u^r)^2 \right] - C_3; \quad (4.26)$$

v) Once  $u^r$  and  $C_1$  are known, all other hydrodynamical quantities can be determined by

$$\rho = \frac{C_1}{r^2 u^r}, \quad p = K\rho^\Gamma, \quad \varepsilon = \frac{p}{(\Gamma-1)\rho}, \quad (4.27)$$

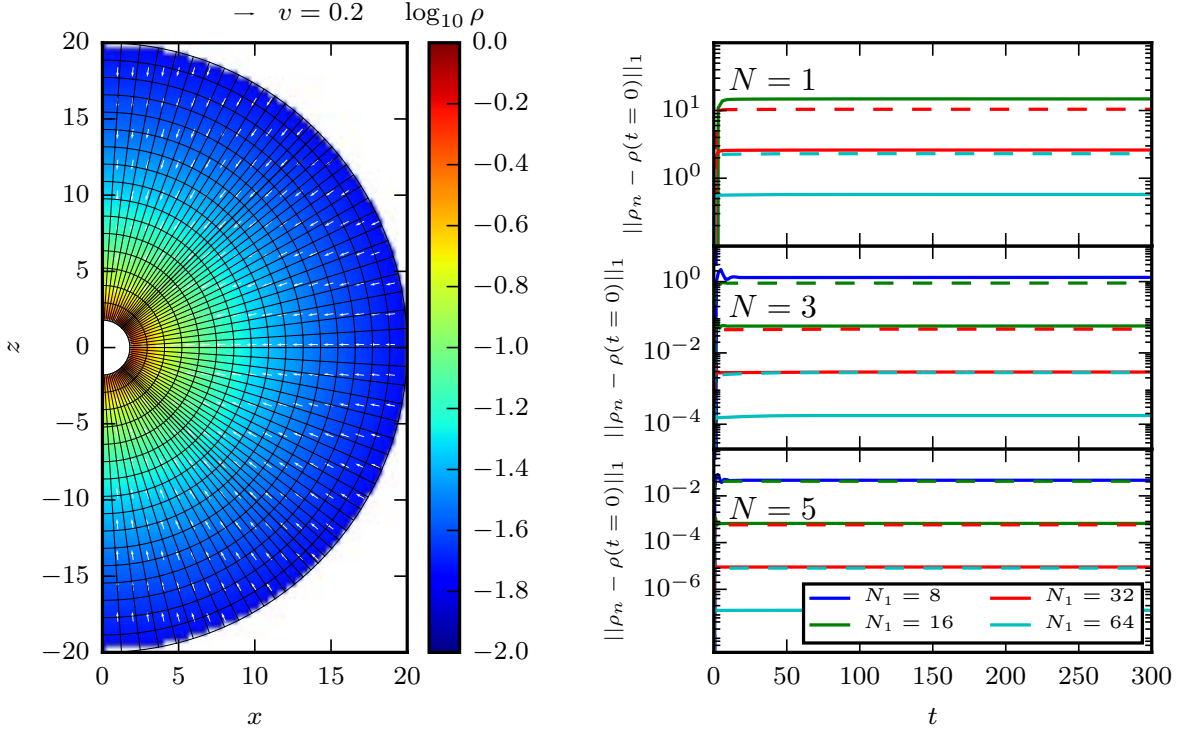
and from the 3+1 split of  $u^\mu$  (see Sec. 2.1, making use of the components of  $n^\mu$ )

$$v^r = \frac{u^r}{\alpha u^0} + \frac{\beta^r}{\alpha}. \quad (4.28)$$

### Test results

We consider two test cases: i) Static accretion, i.e. evolution of initial data (4.27,28). Since these are static, no fields should change during the evolution. The changes in the hydrodynamical fields are used as an error estimate. This test can be done with a pure DG method, since all fields stay smooth during the evolution; ii) Dynamical accretion, i.e. preparation of a low density fluid in the whole domain and injection of a high density fluid from the outer boundary, corresponding to (4.27,28). After a short transition time, the numerical solution tends to the static solution (4.27,28) and we can again compare both to obtain an error measure. For this test shock capturing methods have to be employed. It was also considered a test case in [Radice and Rezzolla, 2011].

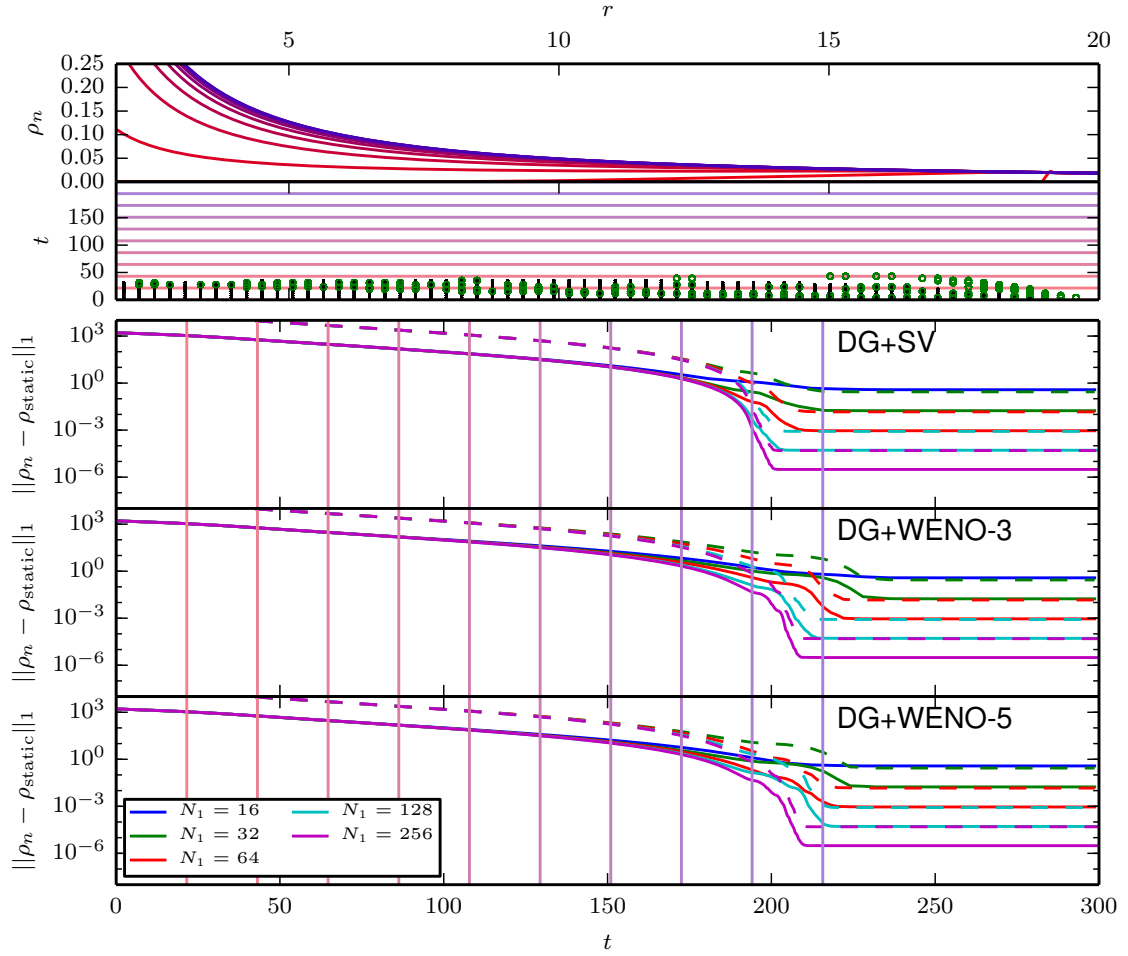
We use the first scenario, the static accretion, as a test for the pure DG method. Although the physical setup is spherically symmetric, we employ a two-dimensional grid setup and the axisymmetric cartoon method. Following this approach, we can additionally check the DG method on a curvilinear grid. To cut the singularity out of the domain, we establish only outer shell grids and omit the inner cube and transition



**Figure 4.7:** Static accretion test results for two-dimensional pure DG runs with the axisymmetric cartoon method. Left: Logarithmic Density (color coding) and velocity field (arrows) are shown for polynomial order  $N = 3$  and 16 radial and angular grid patches,  $N_1 = N_{2,3} = 16$ , at time  $t = 300$ . The plot is qualitatively identical to an initial data plot, because quantities change only marginally. Right: Numerical errors for different polynomial order  $N$  and number of grid patches  $N_1 = N_{2,3}$ . The solid line is the absolute difference of the numerical solution and the initial solution for  $\rho$ , integrated over the domain ( $L_1$  norm). The dashed lines are scaled according to  $(N + 1)$ -th order convergence.

shell patches of the bumps grid. There is no need for specific boundary conditions at the inner boundary  $r = 1.8$ , because all characteristic fields are outgoing there. On the outer boundary  $r = 20$ , we just enforce the analytical solution. The grid setup and the numerical solution at  $t = 300$  for a  $N = 3$ ,  $N_1 = N_{2,3} = 16$  run is shown in the left part of Fig. 4.7. Qualitatively, all hydrodynamical fields do not change during the evolution, as expected for a static solution. A more precise comparison with the analytic solution is shown in the right panel of Fig. 4.7. Taking the difference in the density as an error measure, we see clean convergence of order  $N + 1$ . This is another confirmation that the DG method has been implemented correctly on curvilinear grid patches, obviously giving expected convergence orders for smooth solutions. Furthermore, we verified the method for a non-trivial background spacetime.

In the second test scenario, the dynamical accretion, we combined DG with the different shock capturing methods. The numerical grid is again the reduced bumps grid, consisting only of outer shells. For this test, we exploit the full symmetry of the system and use the effectively one-dimensional spherical cartoon method. As initial data, we fill the whole domain with a low density gas  $\rho = 1.28 \cdot 10^{-6}$ , set the fluid



**Figure 4.8:** Dynamical accretion test with the spherically symmetric cartoon method: A constant high density fluid inflow is modeled at the outer boundary  $r = 20$ . Results from runs with polynomial order  $N = 3$  are shown. Top: Time evolution for the inflow case. Depicted is the density  $\rho$  from the DG+WENO-3 simulation with  $M = 0$ , time increases from red to blue. Middle: The troubled cells in a spacetime diagram for the DG+WENO-3 method (green circles) and the DG+SV method with  $\rho_{SV} = 9 \cdot 10^{-3}$  (black dots). Horizontal lines are color coded in accordance with the top panel, indicating the times at which the numerical solution for  $\rho$  is shown. Bottom: Numerical errors for different shock capturing methods and number of grid patches  $N_1$ . The solid line is the absolute difference of the numerical solution and the static solution for  $\rho$ , integrated over the domain ( $L_1$  norm). The dashed lines are scaled according to fourth order convergence.

velocity to zero and calculate pressure and internal energy according to a polytropic EOS. On the outer boundary, we impose the analytical solution. The dynamics of the infalling matter is depicted in the top panel of Fig. 4.8. All shock capturing methods are able to provide a stable evolution of the dynamical accretion. From the middle panel of Fig. 4.8, we see that the troubled cell indicator follows the inwards traveling shock successfully. After the transient, when the static solution is reached, no cells are marked troubled and the pure DG scheme is applied again. We compare the numerically obtained static solution with (4.27,28). As seen from the bottom panel of Fig. 4.8, the high convergence order of the DG scheme is recovered as the solution

reaches stationarity. This observation is independent of the used shock capturing method. The observation that there is no effect of the the lower order schemes at the end could again be due to the fact that all characteristic speeds are non-zero. Effectively, all lower order artifacts are just “transported” off the grid (comparable to the advection test, see Fig 4.2).

### 4.3 Stable neutron star

As a next step towards real astrophysical application of the code, we consider a single static neutron star. We make use of the simplest solution of the Einstein field equations containing matter, the *Tolman-Oppenheimer-Volkoff* (TOV) solution [Tolman, 1939; Oppenheimer and Volkoff, 1939]. The solution is entirely determined by the central density  $\rho_c$  of the star under a prescribed EOS. For each EOS, there exists a maximal  $\rho_c$ , up to which the configuration is in stable equilibrium. Above this threshold, stars are in unstable equilibrium and either collapse to a black hole or migrate to a stable configuration when perturbed. We consider these unstable TOV stars in Sec. 4.4. TOV solutions in general are spherically symmetric and time independent. Therefore, the Einstein equations turn into a set of ordinary differential equations (ODEs), dependent only on a radial coordinate  $r$ . We obtain the TOV solution numerically, by integrating this ODE system to very high accuracy with a Runge-Kutta method, see details below. Like in the spherical accretion tests, we again compare the evolved solution to this initial solution for an error estimate.

#### Numerical solution of the TOV equations

We want to give a short summary on how we prepare the TOV solution numerically. Some parts of this derivation can be found in [Carroll, 2003; Wald, 1984]. Starting from the Schwarzschild metric in Schwarzschild coordinates

$$ds^2 = -e^{2\phi} dt^2 + e^{2\psi} dr^2 + r^2 d\Omega^2, \quad (4.29)$$

and the perfect fluid energy-momentum tensor (2.21), the Einstein equations take the form:

$$\begin{aligned} 8\pi(\rho h - p)e^{2\phi}r^2 &= e^{2(\phi-\psi)} [2r\psi' - 1 + e^{2\psi}] \\ 8\pi pr^2 &= e^{-2\psi} [2r\phi' + 1 - e^{2\psi}] \\ 8\pi p &= e^{-2\psi} \left[ \phi'' + (\phi')^2 - \phi'\psi' - \frac{\psi' - \phi'}{r} \right]. \end{aligned} \quad (4.30)$$

Since both,  $G_{\mu\nu}$  and  $T_{\mu\nu}$  are diagonal for this problem, we expect maximally four independent equations. Furthermore, the equations originating from  $\theta\theta$ - and  $\varphi\varphi$ -component are identical up to factor  $\sin^2\theta$ , which reduces the system to the three

equations (4.30). The EOS, which relates  $\rho, p$  and  $h$ , closes the system. Therefore the three unknown functions in (4.30) are the metric potentials  $\phi(r)$  and  $\psi(r)$ , and one of the hydrodynamical quantities  $p(r)$ ,  $\rho(r)$  or  $h(r)$ . From the conservation of energy-momentum (2.1b), we gain an additional useful equation

$$p' + \rho h \phi' = 0. \quad (4.31)$$

The remaining system can be substantially simplified by replacing the potential  $\psi$  with the gravitational mass  $m(r)$

$$1 - \frac{2m}{r} := e^{-2\psi}. \quad (4.32)$$

Substituting this into the first two equations of (4.30) and employing (4.31) yields three of the final four ODEs, that are solved to prepare the TOV solution numerically:

$$m' = 4\pi(\rho h - p)r^2, \quad \phi' = \frac{4\pi p r^3 + m}{r(r-2m)}, \quad p' = -\rho h \frac{4\pi p r^3 + m}{r(r-2m)}. \quad (4.33)$$

In principle, solving these three equations would be sufficient to gain valid initial data for a spherically symmetric static star. However, the standard approach is to choose coordinates, such that the spatial part of the metric is explicitly conformally flat. This simplifies its setup in Cartesian coordinates significantly. We transform the radial coordinate and introduce the isotropic radius  $\bar{r} = \bar{r}(r)$ , so that

$$ds^2 = -e^{2\phi} dt^2 + e^{2\bar{\psi}} (d\bar{r}^2 + \bar{r}^2 d\Omega^2) = -e^{2\phi} dt^2 + e^{2\bar{\psi}} (d\bar{x}^2 + d\bar{y}^2 + d\bar{z}^2). \quad (4.34)$$

Comparing the two metrics (4.29) and (4.34), we find two new equations which characterize the coordinate transformation:

$$e^{2\bar{\psi}} \bar{r}^2 = r^2, \quad e^{2\bar{\psi}} d\bar{r}^2 = e^{2\psi} dr^2. \quad (4.35)$$

Combining these two relations and (4.32) yields the differential equation

$$\frac{d\bar{r}}{\bar{r}} = \left(1 - \frac{2m}{r}\right)^{-\frac{1}{2}} \frac{dr}{r} = \frac{1 - \left(1 - \frac{2m}{r}\right)^{\frac{1}{2}}}{\left(1 - \frac{2m}{r}\right)^{\frac{1}{2}}} \frac{dr}{r} + \frac{dr}{r}, \quad (4.36)$$

where we split the singular term after the first equality into a regular ( $m \sim r^3$ ) and a singular (but analytically integrable) part. This step is important for a successful numerical integration to find  $\bar{r}(r)$ . From (4.36) we gain through analytical integration

$$\ln(\bar{r}) - \ln(\bar{r}(r_0)) - \ln(r) + \ln(r_0) = \ln\left(\frac{\bar{r}}{r}\right) - \ln\left(\frac{\bar{r}(r_0)}{r_0}\right) = \int_{r_0}^r \frac{1 - \left(1 - \frac{2m}{r}\right)^{\frac{1}{2}}}{r \left(1 - \frac{2m}{r}\right)^{\frac{1}{2}}} dr \quad (4.37)$$

In the limit  $r_0 \rightarrow 0$ , the term  $\bar{r}(r_0)/r_0$  is positive and regular. The specific ratio for the limit  $r_0 \rightarrow 0$  is freely specifiable and can be absorbed in an integration constant  $C$ :

$$\bar{r} = rC \exp\left(\int_0^r \frac{1 - \left(1 - \frac{2m}{\tilde{r}}\right)^{\frac{1}{2}}}{\tilde{r} \left(1 - \frac{2m}{\tilde{r}}\right)^{\frac{1}{2}}} d\tilde{r}\right) \quad (4.38)$$

At the surface of the star,  $r = R$ , the hydrodynamical fields tend to  $\rho \rightarrow 0$ ,  $p \rightarrow 0$ ,  $h \rightarrow 1$  and  $m \rightarrow M$ , where  $M$  is the total gravitational mass of the star. Outside the star, these fields do not change and (4.36) is analytically integrated:

$$\bar{r} = \frac{1}{2} \left( \sqrt{r^2 - 2Mr} + r - M \right). \quad (4.39)$$

Here, the integration constant has been fixed, so that the two radii agree at infinity. With this last ingredient, we can summarize the established TOV star initial data procedure:

We set

$$m(0) = 0, \quad \phi(0) = 0, \quad p(0) = p(\rho_c), \quad \lambda(0) = 0 \quad (4.40)$$

and numerically integrate the following ODEs by an adaptive stepsize Runge-Kutta method of order four:

$$m' = 4\pi(\rho h - p)r^2, \quad \phi' = \frac{4\pi pr^3 + m}{r(r - 2m)}, \quad p' = -\rho h \frac{4\pi pr^3 + m}{r(r - 2m)}, \quad \lambda' = \frac{1 - \left(1 - \frac{2m}{r}\right)^{\frac{1}{2}}}{r \left(1 - \frac{2m}{r}\right)^{\frac{1}{2}}} dr \quad (4.41)$$

Once the pressure  $p$  approaches zero, we shrink the stepsize to get an accurate result for the star surface  $R$ . We store the values  $M = m(R)$ ,  $\Phi = \phi(R)$  and  $\Lambda = \lambda(R)$ . For  $r \geq R$ , we set the metric variables according to the Schwarzschild metric of mass  $M$ . Furthermore, an atmosphere is modeled as a gas with comparably low density  $\rho_{\text{atm}} \ll \rho_c$ :

$$m = M, \quad \phi = \frac{1}{2} \ln\left(1 - \frac{2M}{r}\right), \quad p = p(\rho_{\text{atm}}), \quad \bar{r} = \frac{1}{2} \left( \sqrt{r^2 - 2Mr} + r - M \right). \quad (4.42)$$

This density is chosen small, but non-zero, in order to prevent the primitive reconstruction from being singular. As a last step, we fix the constants of integration in



**Table 4.4:** Configurations for single, non-spinning neutron star code tests. Only when solving for initial data, a polytropic EOS with  $\Gamma = 2$ ,  $K = 100$  is considered. In the actual time evolutions, an ideal gas EOS with  $\Gamma = 2$  is employed.

Configuration	Gravitational Mass	Radius	Central density
STOV	$1.40M_{\odot}$	$9.59 \equiv 14.16 \text{ km}$	$1.28 \cdot 10^{-3} \equiv 7.90 \cdot 10^{14} \frac{\text{g}}{\text{cm}^3}$
UTOV	$1.45M_{\odot}$	$5.84 \equiv 8.62 \text{ km}$	$8.00 \cdot 10^{-3} \equiv 4.93 \cdot 10^{15} \frac{\text{g}}{\text{cm}^3}$

order to match the solutions for  $r < R$  and  $r \geq R$  by assigning

$$\phi \leftarrow \phi - \Phi + \frac{1}{2} \ln \left( 1 - \frac{2M}{R} \right), \quad \bar{r} \leftarrow \frac{r}{2R} \left( \sqrt{R^2 - 2MR} + R - M \right) \exp(\lambda - \Lambda) \quad (4.43)$$

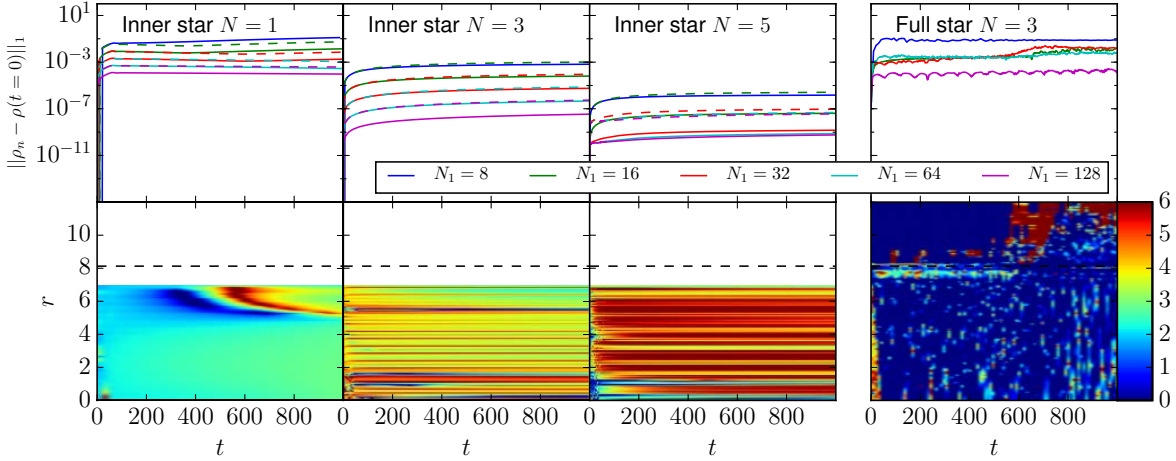
for  $r < R$ . The metric potential  $\bar{\psi}$  can be easily recovered from (4.35) for all  $r$ . The final 3+1 quantities are:

$$\alpha = \exp(\phi), \quad \beta^i = 0, \quad \gamma_{ij} = \frac{\bar{r}^2}{r^2} \eta_{ij}. \quad (4.44)$$

All these data fields are stored with respect to the radius  $r$ . Since we need the data at an arbitrary isotropic radius  $\bar{r}$ , an interpolation is necessary and the data arrays have to be of sufficiently fine resolution.

### Influence of the stellar surface

As mentioned before, an artificial atmosphere is usually imposed in numerical relativity simulations of neutron stars. During the test phase of our implementation, we found that this atmosphere has crucial influence on the whole scheme. To substantiate this statement, we performed an evolution of the very same TOV star configuration STOV (see Tab. 4.4), including and excluding its surface from the numerical domain. In the first case, we set  $\rho_{\text{atm}} = 1.28 \cdot 10^{-11}$  for  $r \geq R$ . In the second case, the TOV solution (4.41) is fixed at the outer grid boundary. For these tests, the spacetime is still considered static, i.e. we work in Cowling approximation. In both setups, a spherically symmetric cartoon DG method of different polynomial order  $N$  is used for the evolution. The results are shown in Fig. 4.9. As in the spherical accretion test, we used the difference between numerical and analytical density solution as an error indicator. For the evolutions of the interior of the star, depicted in the three left columns of Fig. 4.9, we see the expected convergence behavior for this error. We can again confirm, that the DG method enables convergence of order  $N + 1$ , also for a general relativistic setup with non-trivial background spacetime. For  $N = 5$ , saturation is quickly reached, as the error decreases to  $\sim 10^{-11}$  for  $N_1 = 32$ . However, in the evolution of the entire star, convergence is obviously corrupted. A close look at the corresponding pointwise convergence order plot in Fig. 4.9 (bottom right)

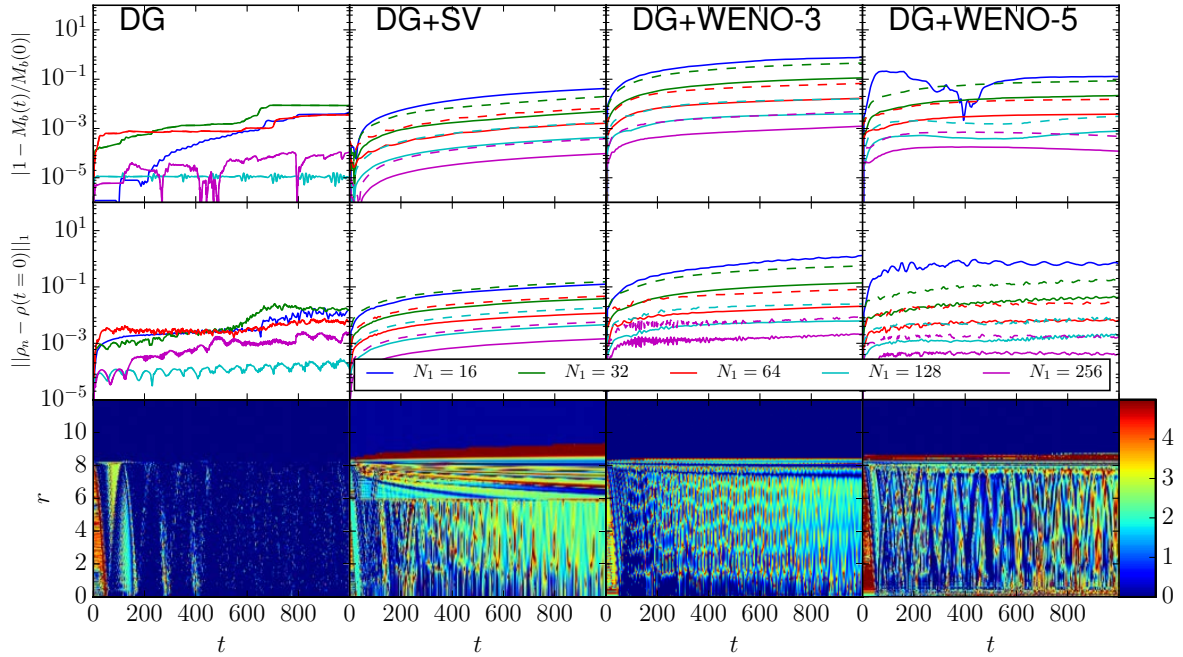


**Figure 4.9:** Code tests for the configuration STOV in Cowling approximation with the spherically symmetric cartoon method. In the three left columns, the interior of the star is simulated with analytic outer boundary conditions. The star surface is not inside the domain. Each column represents a polynomial order  $N$  of the DG scheme. The rightmost column shows a realistic setup of the whole star including its surface. It is surrounded by a low density atmosphere  $\rho_{\text{atm}} = 1.28 \cdot 10^{-11}$ . Top: Numerical errors for different number of grid patches  $N_1$ . The solid line is the absolute difference of the numerical solution and the initial solution for  $\rho$ , integrated over the domain ( $L_1$  norm). The dashed lines are scaled according to  $(N + 1)$ -th order convergence. Bottom: Pointwise convergence in a spacetime diagram. The numerical errors from the  $N_1 = 16, 32$  runs were used to determine the convergence order. The star surface is indicated by the dashed line.

reveals, that initial high order convergence in the stars interior is completely gone after  $t \sim 50$ . The same effect is even more visible for higher resolution runs, as shown in the bottom left plot of Fig. 4.10. After that point in time, no clear convergence order can be read off. The same impression is visible from the integrated errors (top row in Fig. 4.9), which are unstructured compared to the runs that do not take the star surface into account. From these tests, we draw the conclusion that a naive DG setup is not able to perform stable evolutions of entire neutron stars without further modification. As in other NR codes, the artificial atmosphere seems to be a major error source, leading to a fast loss of high order convergence. We think that two strategies could cure this issue: i) The application of shock capturing techniques, which is the way we follow in this thesis; ii) A numerical grid, that is carefully adapted to the star surface. This would include the solution of the initial free-boundary problem for an generic neutron star surface, which is - to the best of our knowledge - not yet available.

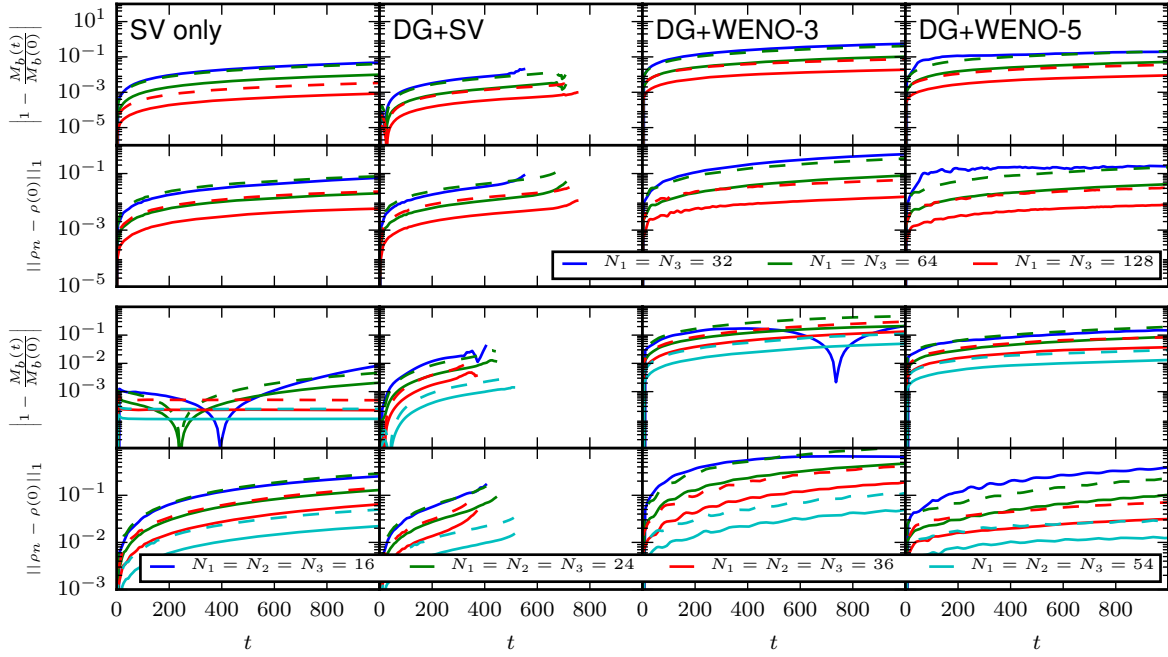
### TOV star tests in Cowling approximation

We again perform an evolution of the configuration STOV as in the previous subsection, but combine the DG scheme with the WENO-3, WENO-5 and SV shock capturing methods. The entire star with surface and artificial atmosphere is considered. These runs are similar to the main test discussed in [Bug1], although not



**Figure 4.10:** Code tests for the configuration STOV with the spherically symmetric cartoon method and polynomial order  $N = 3$ . The columns represent different shock capturing strategies that we employ additionally to the DG scheme. For the WENO schemes, a troubled cell parameter  $M = 1 \cdot 10^{-5}$  was used, while for the SV scheme we set  $\rho_{SV} = 1.28 \cdot 10^{-4}$ . Top: Mass conservation as indicated by the relative baryonic mass change. Middle: Numerical density error. Both are shown for different number of grid patches  $N_1$ . The dashed lines are scaled according to second order convergence. Bottom: Pointwise convergence in a spacetime diagram. The numerical errors in the density from the  $N_1 = 128, 256$  runs were used to determine the convergence order.

completely identical. In [Bug1], we set all  $y$ - and  $z$ -derivatives equal to zero, while in this work, real spherical symmetry via the cartoon method is enforced. The results for these runs are shown in Fig. 4.10. All tests employ a DG method of order  $N = 3$ . As a reference, we show again the pure DG case, i.e. the first column in Fig. 4.10 is identical to the last column in Fig. 4.9. As an additional error indicator, we monitor again the conservation of the baryonic mass (4.11). The pointwise convergence is estimated from high resolutions  $N_1 = 128, 256$ . As mentioned before, convergence is clearly lost through the interaction with the stellar surface if no shock capturing is used. That said, all of the shock capturing methods considered seem to cure this problem partially. At least second order convergence in both the density error and the mass conservation is achieved if DG is combined with one of these methods. Also in the pointwise convergence plots we can confirm, that convergence is not decaying over time, if shock capturing is active. Comparing the errors, SV and WENO-5 give lower errors than the WENO-3 reconstruction. However, these methods are also significantly slower (see Tab. 4.5 for a running time comparison of full GR TOV simulations). While the WENO-3 (WENO-5) method is a factor 1.14 – 1.22 (1.27 – 1.42) slower than pure DG, the hybrid DG+SV method comes with large overhead of a



**Figure 4.11:** Code tests for the configuration STOV with the axisymmetric cartoon method (top panel) and in full 3D (bottom panel) with polynomial order  $N = 3$ . The columns represent different shock capturing strategies that we employ additionally to the DG scheme. For the WENO schemes, a troubled cell parameter  $M = 1 \cdot 10^{-5}$  was used, while for the hybrid DG+SV scheme we set  $\rho_{\text{SV}} = 1.28 \cdot 10^{-4}$ . Top row: Mass conservation as indicated by the relative baryonic mass change. Bottom row: Numerical density error. Both are shown for different number of grid patches  $N_1$ . The dashed lines are scaled according to second order convergence.

factor 2.13–2.60. The main bottleneck is the conversion of the 50 spacetime variables, which are only available at the collocation points, to SV averages. These are required to calculate the source terms for the SV scheme. The running times increase by a factor of  $\sim 4$  when  $N_1$  is doubled, as expected for one-dimensional simulations.

So far all neutron star tests have been performed with the spherically symmetric cartoon method, i.e. in one spatial dimension. As a technical test we evolved the same configuration STOV in 2D, with the axisymmetric cartoon method, and in full 3D. The results of these runs are summarized in Fig. 4.11. Looking again at baryonic mass conservation and density error, we see that the WENO shock capturing methods allow stable long-term simulations in 2D and 3D. These methods again show convergence of second order. As in the previous one-dimensional runs, the errors are slightly higher for the WENO-3 method as compared to WENO-5. In the hybrid DG+SV methods, instabilities at the DG/SV interfaces occur, which ultimately lead to a failure of all 2D and 3D simulations. The 3D runs break after shorter simulation time than the 2D runs. For both second order convergence can be observed up to the crash. We discussed the reason for these instabilities at the end of Sec. 3.3.2. The naive implementation of the DG/SV interface in bamps does not respect the conservation

**Table 4.5:** Comparison of code running time in seconds for full GR TOV star evolutions up to  $t = 1000$ . Runs were done with the spherically symmetric cartoon method and polynomial order  $N = 3$ . A desktop machine with 32 GB of memory and 4 4.00 GHz Intel i7 cores was used. Comparison of different shock capturing strategies and number of grid patches  $N_1$ .

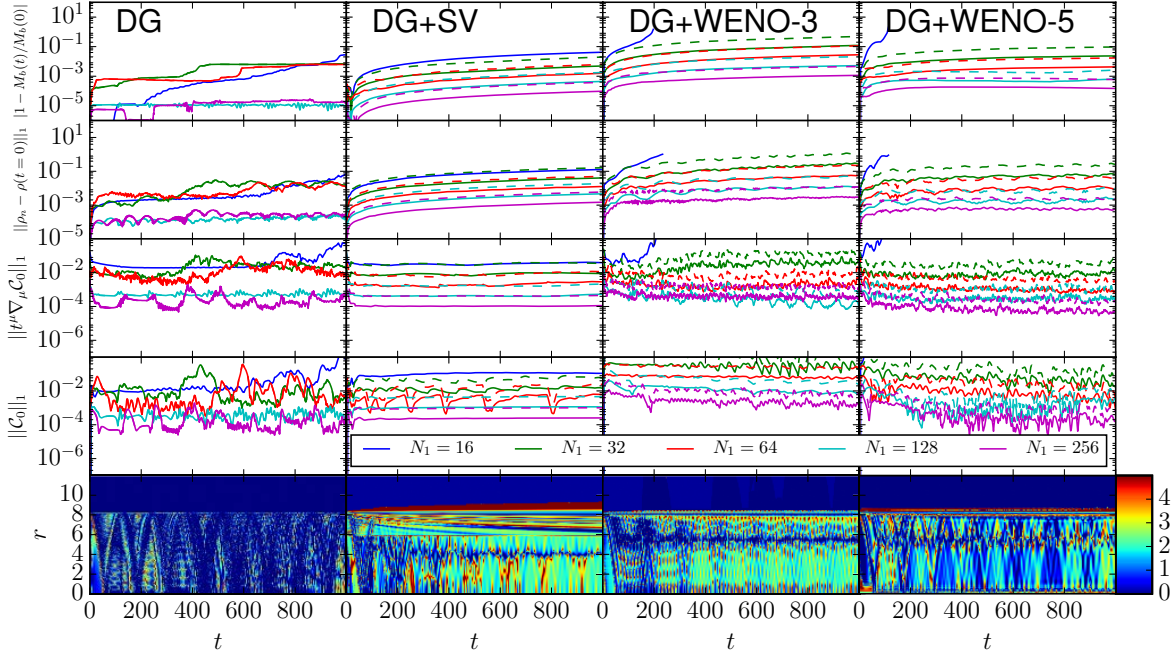
$N_1$	DG	DG + WENO-3	DG + WENO-5	DG + SV
16	24.57			51.90
32	83.43	94.94	105.50	189.38
64	300.50	367.74	428.73	752.17
128	1175.94	1366.16	1589.13	3063.96
256	4763.36	5488.51	6394.64	11835.02

and outflow conditions as defined in [Kopriva, 1996]. Applying the mortar method as in [Choi, 2015] would be a possible solution. Simulations are stable and convergent when no DG/SV interface is present (i.e. when SV is used everywhere, see left column in Fig. 4.11). Moreover, hybrid DG+SV simulations turned out to be stable in 1D. As the DG/SV interface is only one point in one-dimensional runs, our method and the mortar method are identical in this case. Both facts support our assumption about the source of the observed instabilities in two and three dimensions.

Performing spherically symmetric neutron star simulations in full 3D is obviously wasteful regarding code running times. Comparing the DG+WENO-3 runs, the spherically symmetric cartoon simulation is  $\sim 100$  times faster than the axisymmetric cartoon and  $\sim 20000$  times faster than the full 3D counterpart. Comparing methods, the DG+WENO-3 method is 1.41 (6.50) times faster than the DG+WENO-5 (SV only) method in 3D, and 1.63 (2.26) times faster than the DG+WENO-5 (SV only) method in 2D. Obviously, the time consuming conversion of DG polynomial values to SV averages leads to an unacceptable overhead in a fully three-dimensional hybrid DG+SV method.

### Fully general relativistic stable TOV star test

As a last stable, non-rotating star test, we want to discard the Cowling approximation and evolve the fully general relativistic coupled system of spacetime and matter. The initial configuration is again STOV. We employ the full bamps code and evolve the spacetime variables with the pseudospectral penalty method, while the matter variables are treated with DG and shock capturing methods. For the GHG system, we use the damping parameters  $\gamma_0 = 0.1, \gamma_2 = 1, \gamma_3 = \gamma_4 = 0.5$ . Besides the density deviation and the mass conservation, we can use the GHG constraints  $\mathcal{C}_\alpha$  (see Sec. 2.3) as an error measure. According to [Lindblom et al., 2006], the time derivative  $t^\mu \nabla_\mu \mathcal{C}_0$  is closely related with the *Hamiltonian constraint*  $\mathcal{H}$ , which is usually used as an error indicator in NR. Similarly,  $t^\mu \nabla_\mu \mathcal{C}_i$  is related to the *momentum constraint*  $\mathcal{M}_i$ . These four constraints naturally emerge from a 3+1 split of the Einstein field equations, which decompose into 4 constraint equations and 6



**Figure 4.12:** Code tests for the configuration STOV in full GR with the spherically symmetric cartoon method and polynomial order  $N = 3$ . The columns represent different shock capturing strategies that we employ additionally to the DG scheme. For the WENO schemes, a troubled cell parameter  $M = 1 \cdot 10^{-5}$  was used, while for the SV scheme we set  $\rho_{SV} = 1.28 \cdot 10^{-4}$ . First row: Mass conservation as indicated by the relative baryonic mass change. Second row: Numerical density error. Third row: The GHG constraint  $t^\mu \nabla_\mu \mathcal{C}_0$ , as an indirect measure for the Hamiltonian constraint. Fourth row: The GHG constraint  $\mathcal{C}_0$ . All error measures are shown for different number of grid patches  $N_1$ . The dashed lines are scaled according to second order convergence. Bottom: Pointwise convergence in a spacetime diagram. The numerical errors in the density from the  $N_1 = 128, 256$  runs were used to determine the convergence order.

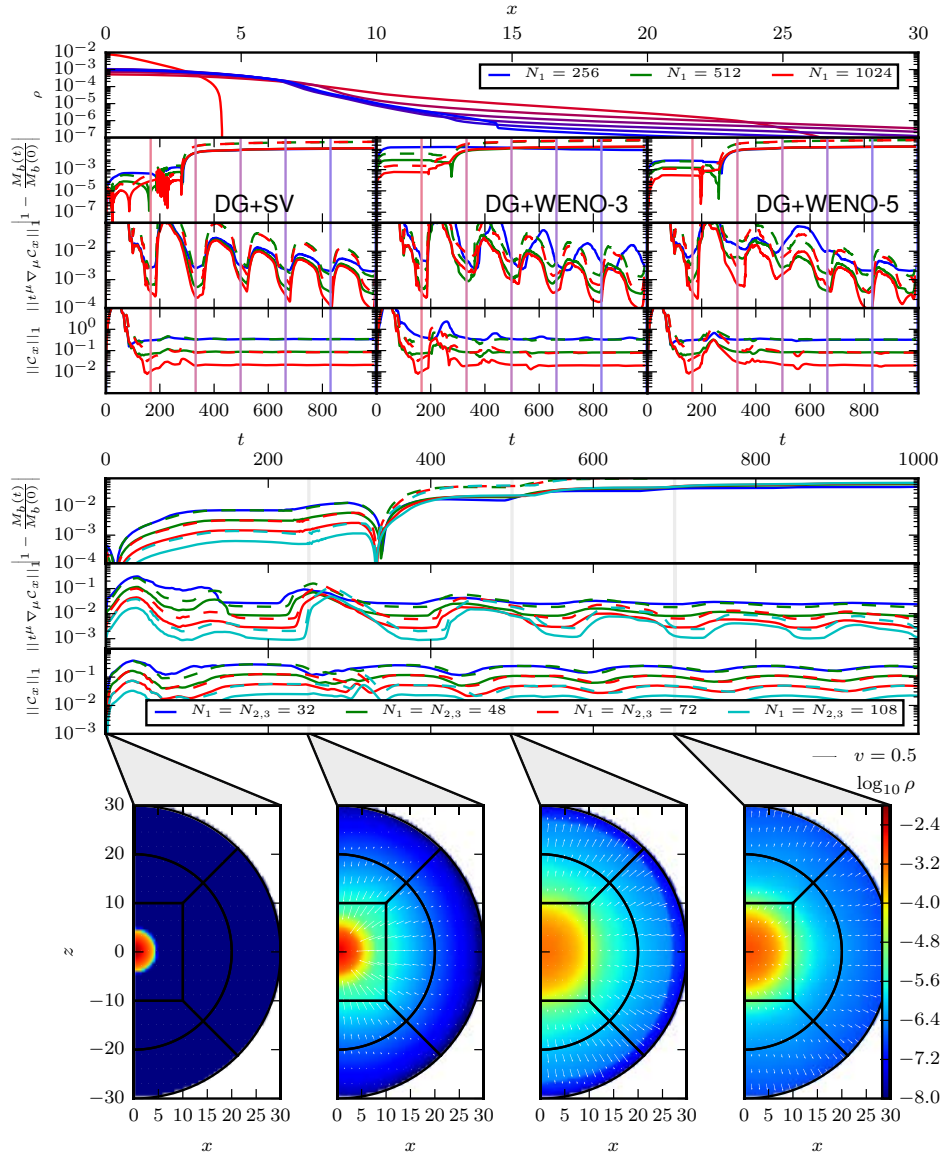
evolution equations for the extrinsic curvature  $K_{ij}$ . We show all these error measures in Fig. 4.12. Qualitatively, the data look similar to those of the Cowling approximation runs. We still observe second order convergence in all error indicators and for all shock capturing methods. The only exception is the highest resolution run for the DG+WENO-3 method, where convergence in  $t^\mu \nabla_\mu \mathcal{C}_0$  seems to fail. We have no plausible explanation for this behavior, especially since all other error measures do not show this lack of convergence. We can only assume, that it is caused by the finite atmosphere level or the initial data interpolation onto the bumps grid. Apart from this, the code evolves the full GR setup as expected. Comparing the pointwise convergence graphs with those of the Cowling approximation runs, the plots look almost identical. The evolution of the matter variables seems to be almost unaffected by engaging the spacetime evolution in bumps. The one-dimensional full GR runs are a factor  $\sim 2.3$  ( $\sim 2.0$ ) slower than their Cowling counterparts, when WENO-3 (WENO-5) is active. For the SV method, this factor is only  $\sim 1.2$ , as the hybrid DG+SV approach is more time consuming than WENO reconstruction. All code running times for these tests

are listed in Tab. 4.5.

## 4.4 Unstable neutron star

In this section, we again consider a TOV solution (4.41), but choose a high central density  $\rho_c$  so that the star is in unstable equilibrium. We denote this configuration as UTOV in Tab. 4.4, modeled as an ideal gas with  $\Gamma = 2$ ,  $\kappa = 100$  and an artificial atmosphere  $\rho_{\text{atm}} = 1.28 \cdot 10^{-12}$ . Although this solution is still static, small perturbations of the numerical solution lead to non-trivial dynamics of the system [Font et al., 2002]: Either the star collapses to a black hole, or it migrates to a stable star configuration of the same mass. This initial setup is considered a first fundamental test in many studies on numerical relativity codes [Cordero-Carrión et al., 2009; Bernuzzi and Hilditch, 2010; Radice and Rezzolla, 2011; Thierfelder et al., 2011]. Also in this work, it constitutes the first dynamical, fully general relativistic testbed. For the GHG system, we use the damping parameters  $\gamma_0 = 0.1$ ,  $\gamma_2 = 1$ ,  $\gamma_3 = \gamma_4 = 0.5$ . To trigger the dynamics, we reduce the density to 94% of its actual value and update the other hydrodynamical fields according to the EOS. Of course, this procedure is clearly violating the constraint equations. However, our results are qualitatively very close to the solutions shown in [Font et al., 2002; Cordero-Carrión et al., 2009], in which the constraint equations were satisfied. The density modification causes the star to migrate from a compact  $R = 5.84 = 8.62$  km,  $\rho_c = 7.52 \cdot 10^{-3} = 4.63 \cdot 10^{15} \frac{\text{g}}{\text{cm}^3}$  configuration to a less compact  $R \approx 8.5 = 12.55$  km,  $\rho_c \approx 8.80 \cdot 10^{-4} = 5.41 \cdot 10^{14} \frac{\text{g}}{\text{cm}^3}$  configuration within  $\Delta t = 1000 = 4.93$  ms. This migration process is dominated by shock waves in density and internal energy, which rapidly dissipate energy as the star oscillates and finally settles to the stable equilibrium. An impression of the time evolution is given in Fig. 4.13. The evolution with the spherically symmetric cartoon method is depicted in the top panel of the figure, while in the bottom part, data from the axisymmetric cartoon runs is shown. The whole evolution is spherically symmetric and we again perform the axisymmetric cartoon run only for technical testing. As the simultaneous evolution of the GHG system is required and boundary conditions should be applied on spherical surfaces, the evolution can not be done on a central box grid in 2D. We use the standard bumps cubed-ball grid, which is why we can only apply the SV method for shock capturing in the axisymmetric cartoon runs. Looking again at the mass conservation in both, one- and two-dimensional runs, we observe that mass is conserved only up to a certain time  $t \approx 300$ . This is due to the finite numerical grid and corresponds to the time, when the first shock wave reaches the outer grid boundary. The second snapshot in Fig. 4.13 shows the system shortly prior to that. Until  $t \approx 300$ , the error in mass conservation converges again with second order. Furthermore, second order convergence is observed for





**Figure 4.13:** Code tests for the configuration UTOV in full GR with the spherically symmetric cartoon method (top panel) and the axisymmetric cartoon method (bottom panel) with polynomial order  $N = 3$ . Top panel first row: Time evolution of the density for  $t \in [0, 1000]$  at 7 evenly distributed times. Top panel second to fourth row: The columns represent different shock capturing strategies that we employ additionally to the DG scheme. For the WENO schemes, a troubled cell parameter  $M = 1 \cdot 10^{-5}$  was used, while for the SV scheme we set  $\rho_{\text{SV}} = 1 \cdot 10^{-3}$ . Second row: Mass conservation as indicated by the relative baryonic mass change. Third row: The GHG constraint  $C_x$ . Fourth row: The GHG constraint  $t^\mu \nabla_\mu C_x$ , as an indirect measure for the momentum constraint  $\mathcal{M}_x$ . Bottom panel rows: The same error measures as in the top panel, listed for the axisymmetric cartoon runs with the DG+SV method. Bottom panel snapshots: The dynamics of the migration test. Logarithmic density (color coding) and velocity (arrows) are depicted for  $N_{\text{sub}} = 32$ . The patch boundaries are marked by the thick black lines. All error measures are shown for different number of grid patches. The dashed lines are scaled according to second order convergence.

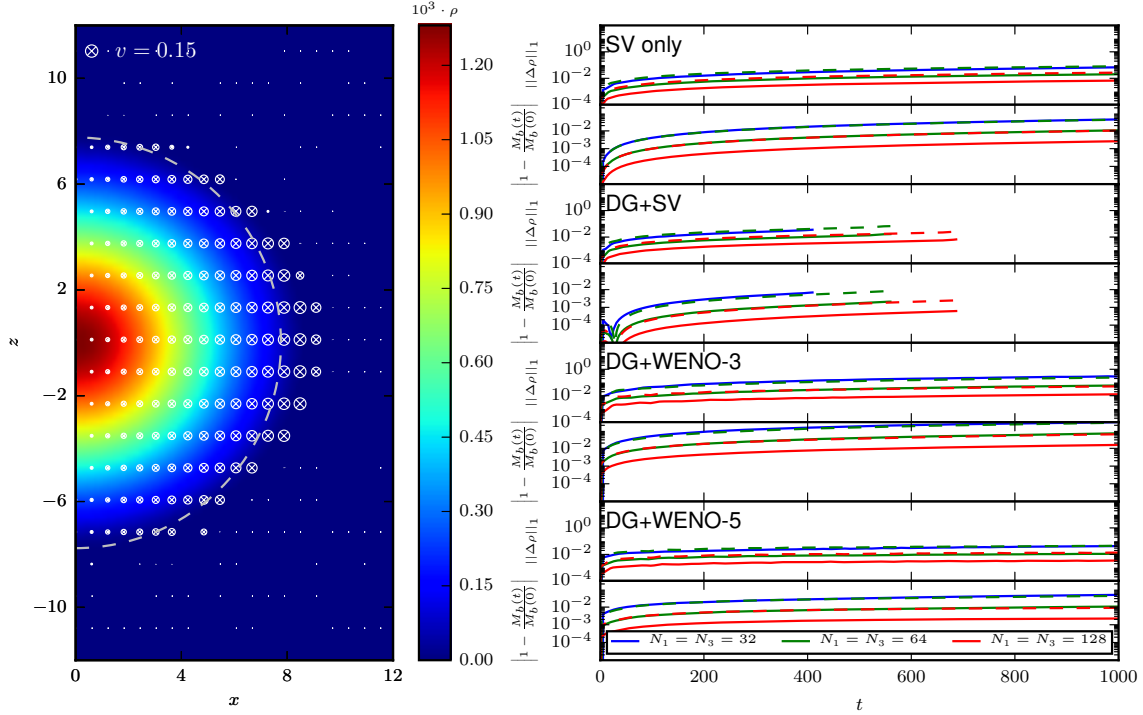


the GHG constraint  $\mathcal{C}_x$  in all runs. For the runs in 1D, we realized that very high resolution is necessary for a successful simulation with WENO methods. If we chose  $N_1$  too small, these methods just damp the sharp star profile too drastically. At these high resolutions, we see no clear convergence behavior in the  $t^\mu \nabla_\mu \mathcal{C}_x$  constraint. We expect that this is due to the comparatively big constraint violation in the modified initial data. For lower resolution, as in the two-dimensional runs, convergence is slightly clearer. In all simulations we observe that the convergence of this constraint changes periodically in time. This can be explained by the repetitive emission of shock waves in density and internal energy. These form a discontinuity inside the numerical domain, so that only first order convergence can be expected (e.g. third bottom snapshot in Fig. 4.13). As soon as the shock leaves the grid, second order convergence is regained (e.g. fourth bottom snapshot in Fig. 4.13). As a last remark, we want to comment on the stability of the hybrid DG+SV method in these runs. It appears to be stable for this setup, but as the star density is below  $\rho_{\text{SV}} = 1 \cdot 10^{-3}$  almost everywhere for  $t > 500$ , the SV method is employed in large parts of the grid and DG/SV interfaces are rare. Therefore, these runs are in no contradiction to the previously observed instabilities in the higher dimensional DG+SV method.

## 4.5 Rotating neutron star

In the last section of this chapter, we want to present a test beyond spherical symmetry: the rotating neutron star (RNS). With this setup, we return to stationary solutions of GRHD, although only axisymmetric. Furthermore, this testbed provides steep velocity gradients, as the surface of the star moves with  $\sim 0.15c$  for our configuration. Unfortunately, the computation of RNS initial data is not as straightforward as for the TOV star. So far, no analytical consistent solution for interior and exterior of a RNS is known. Several numerical methods for RNS initial data computation are discussed in the comprehensive work [Stergioulas, 2003]. The general approach is to either use a slow rotation approximation [Hartle, 1967; Hartle and Sharp, 1967] or to solve the full Einstein equations (2.1c) in a Newton-Raphson like approach [Butterworth and Ipser, 1976]. In the latter method, the spin of the neutron star is slightly increased with each iteration step. In each such step, the linearized Einstein equations are solved. In this work, we compute the RNS initial data with the code presented in [Stergioulas and Friedman, 1995], which is based on the integral form of these equations [Komatsu et al., 1989]. The code computes the RNS solution on a sufficiently high resolved  $r - \cos(\theta)$ -grid, so that it can be interpolated to the bumps grid as in the TOV case. As in all other neutron star tests, an ideal gas EOS with  $\Gamma = 2$ ,  $\kappa = 100$  and an artificial atmosphere  $\rho_{\text{atm}} = 1.28 \cdot 10^{-15}$  is employed.

As the RNS solution is stationary, we can again compare the numerical solu-



**Figure 4.14:** Code tests for a rotating neutron star in Cowling approximation with the axisymmetric cartoon method and polynomial order  $N = 3$ . Left: Density (color coding) and velocity field (perpendicular to plane) are shown for  $N_1 = N_3 = 128$  grid patches, at time  $t = 1000$ , i.e. after 2.4 revolutions. The plot is qualitatively identical to an initial data plot, because quantities change only marginally. Right: Error measures for different number of grid patches  $N_{1,3}$  and shock capturing strategies that we employ additionally to the DG scheme. For the WENO schemes, a troubled cell parameter  $M = 1 \cdot 10^{-5}$  is used, while for the hybrid DG+SV scheme we set  $\rho_{SV} = 1.28 \cdot 10^{-4}$ . Top rows: Numerical density error. Bottom rows: Mass conservation as indicated by the relative baryonic mass change. The dashed lines are scaled according to second order convergence.

tion with the initial data for an error estimate. This is depicted for the Cowling approximation simulations in Fig. 4.14. Furthermore, the density and velocity profile of a neutron star with a central density  $\rho_c = 1.28 \cdot 10^{-3}$  and revolution time of  $\tau = 416.49 = 2.05$  ms after 2.4 rotations is shown. These profiles change only marginally during evolution. Note that the star is tidally deformed due to its rotation. As a reference, the star surface of a TOV star with the same gravitational mass  $M = 1.47 M_\odot$  is marked by the gray dashed line. Focusing on the errors shown in the right part of Fig. 4.14, all observations are compatible with the two-dimensional stable TOV evolutions. The DG+WENO methods and the standalone SV method are able to perform stable and second order convergent simulations of the rotating neutron star. The hybrid DG+SV method tends to fail, because of the simplistic interface implementation that we already discussed. Up to the point of failure, the hybrid method is also second order convergent. As in the other tests, the errors from the DG+WENO-3 runs are slightly bigger than those from the DG+WENO-5 or SV runs.

# Chapter 5

## Neutron star head-on collision

As a proof of principle we perform a neutron star merger simulation with our implementation and extract the emitted gravitational waves. These final tests also constitute a new type of simulation which is now accessible to the `bamps` code: the collision of two compact objects. Besides the evolution strategies discussed in this work, an initial data solver for elliptic equations was developed and integrated in `bamps` by Hannes Rüter [Bug3]. The method will be explained as one of the main topics in his Ph.D. thesis. In this work, we only want to give a basic overview on the initial data solution. First, we motivate the elliptic equations, that have to be solved in order to obtain valid initial data. A 3+1 split of the Einstein field equations yields four constraint equations on the induced metric  $\gamma_{ij}$  and the extrinsic curvature  $K_{ij}$ . However, it is unclear which of their components should be freely specifiable and which are constrained. A successful formulation of the constraint equations is the *conformal thin sandwich* approach [York, 1999]. This is based on a conformal decomposition of  $\gamma_{ij} = \psi^4 \bar{\gamma}_{ij}$  and  $K_{ij} = \psi^{-2} \bar{A}_{ij} + \frac{1}{3} K \psi^4 \bar{\gamma}_{ij}$ , so that  $\bar{\gamma}_{ij}$ ,  $\partial_t \bar{\gamma}_{ij}$ ,  $K$ , and  $\alpha$  are freely choosable, while four equations are solved for the *conformal factor*  $\psi$  and  $\beta^i$ . As we chose an initially flat conformal metric  $\bar{\gamma}_{ij} = \delta_{ij}$  and furthermore  $K_{ij} = 0$ ,  $p^i = 0$ , the shift equation is trivially solved by  $\beta^i = 0$ . The last remaining equation for  $\psi$  is

$$0 = \partial_i \partial^i \psi + 2\pi E \psi^5 \quad (5.1)$$

with  $E = \rho h W^2 - p$ . With the solution of (5.1), the initial metric is easily obtained as  $\gamma_{ij} = \psi^4 \delta_{ij}$ . To solve the elliptic equation (5.1), it is reduced to a first order system

$$\begin{aligned} 0 &= \delta^{ij} \partial_i r_j + 2\pi E \psi^5, \\ 0 &= \partial_i \psi - r_i, \end{aligned} \quad (5.2)$$

and “evolved” in analogy to the Jacobi method [Press et al., 1986] as

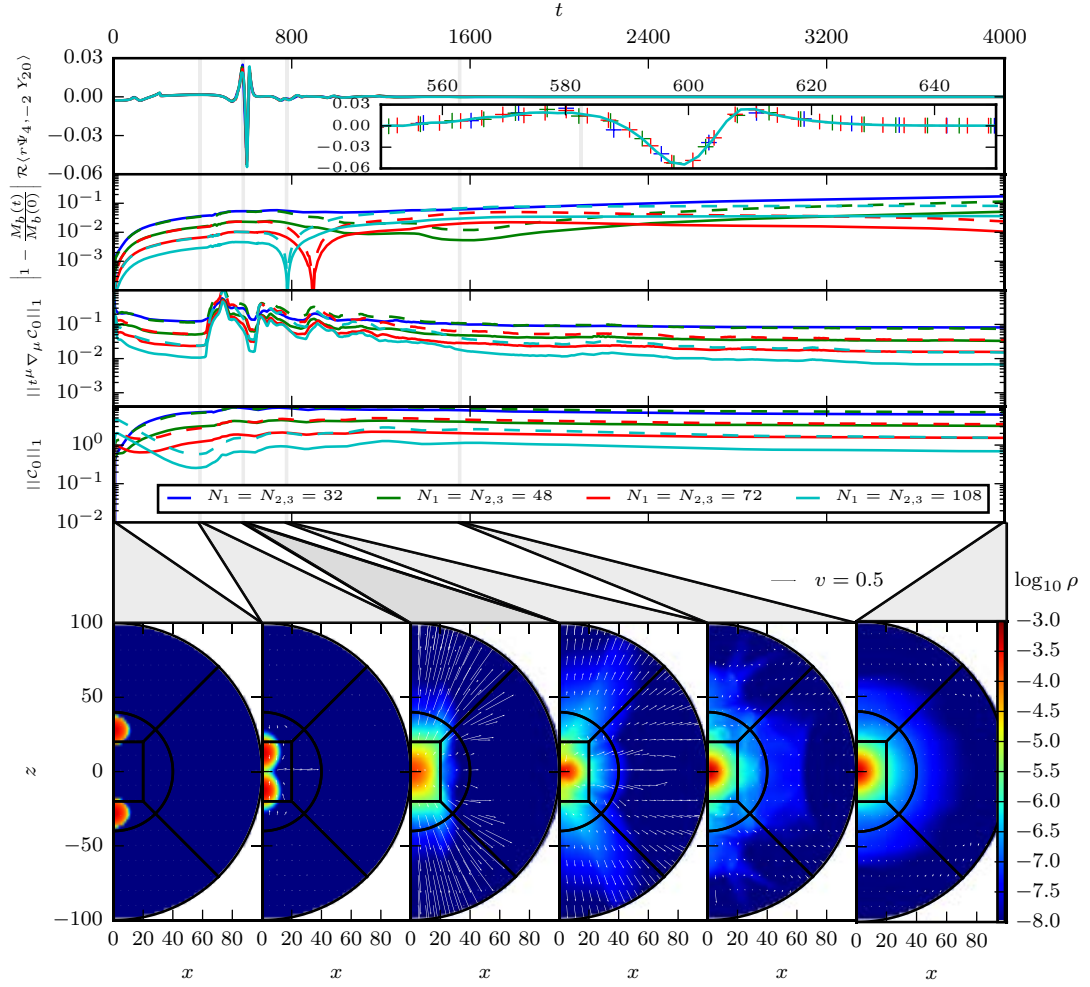
$$\begin{aligned}\partial_t \psi &= \delta^{ij} \partial_i r_j + 2\pi E \psi^5, \\ \partial_t r_i &= \partial_i \psi - r_i.\end{aligned}\tag{5.3}$$

This relaxation can be performed with the same pseudospectral method and on the same numerical grid which is usually used for evolutions in `bamps`. The grid patch boundary conditions are again implemented as penalties (see. Sec. 3.1, eq. (3.19)). The outer boundary conditions are more delicate, especially if Dirichlet boundary conditions have to be imposed, see [Bug3] for details. In our case Robin boundary conditions are applied, i.e. the conformal factor is assumed to take the form

$$\psi \simeq 1 + \frac{A}{r}\tag{5.4}$$

in the limit of large  $r$ . From this expression, the derivatives  $s^i \partial_i \psi$  and  $s^i \partial_i r_j$  can be determined at the boundary and imposed via the Bjørhus method. As an initial guess for  $\psi$  and as initial data for  $\rho$ ,  $p$ ,  $\varepsilon$  and  $\alpha$  we chose two superposed TOV stars. These stars are centered in the  $x$ - $y$ -plane and shifted on the  $z$ -axis by  $\frac{d}{2}$  and  $-\frac{d}{2}$ , respectively. In our example runs, we chose this initial distance of the star centers to be  $d = 80 = 118.15$  km. In isolation, the stars would have a radius  $r = 10.33 = 15.28$  km and equal gravitational masses  $M_1 = M_2 = 0.56 M_\odot$ . For the EOS, we chose a polytropic EOS with  $K = 80$ ,  $\Gamma = 2$  for the initial data and an ideal EOS for the evolution. The initial matter velocity field is set to zero and the artificial atmosphere value is chosen  $\rho_{\text{atm}} = 1.28 \cdot 10^{-15}$ .

We employ the pure SV method to evolve the matter variables. As observed in the test section, the hybrid DG+SV approach requires conservative DG/SV interfaces to guarantee stable evolutions. On the other hand, DG+WENO methods are only implemented for box grids, but spherical boundaries are needed to impose proper boundary conditions for the GHG system. As depicted in the snapshots at the bottom of Fig. 5.1, the neutron stars start falling towards each other. After  $t \approx 400 = 1.97$  ms, the outer stellar layers touch and the stars come into contact. At  $t \approx 485 = 2.39$  ms, the stellar cores merge and a maximum density (four times larger than the initial central density of the stars) is reached. At  $t \approx 600$ , 0.57 ms later, the first shock waves of low density matter with high internal energy pass the outer boundary of the numerical grid at  $r = 148$  km. Several consecutive shock waves transfer matter out of the numerical domain, so that mass conservation is not expected to hold from this point in time. Also at  $t \approx 600 = 2.96$  ms, the maximum of gravitational radiation would be measured at  $r = 148$  km. To extract a gravitational waveform from the data, we follow the formalism explained in section 3.3 of [Bishop and Rezzolla, 2016]. An

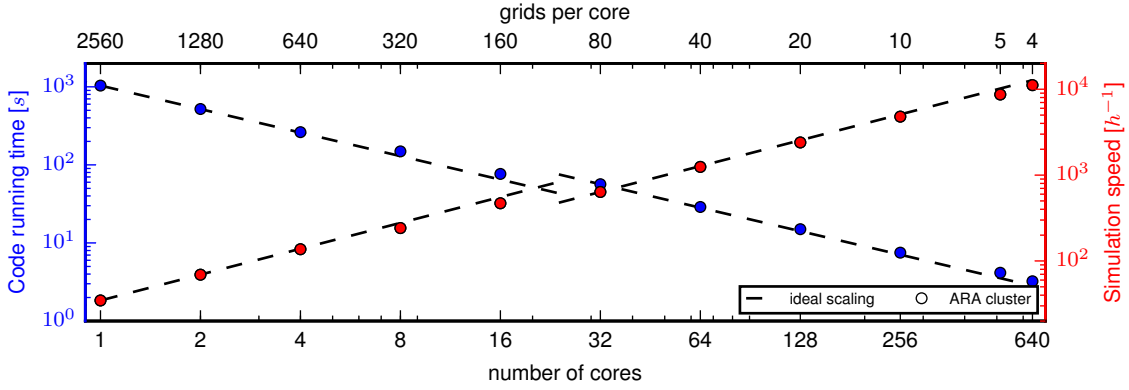


**Figure 5.1:** The head-on collision of two neutron stars with equal mass  $M_1 = M_2 = 0.56M_\odot$  in full GR, employing axisymmetric cartoon method and SV scheme for the matter treatment. First row:  $(2, 0)$  mode of the  $\Psi_4$  gravitational waveform, rescaled according to the finite extraction radius at  $r = 100 = 148$  km. Inset: Magnification of the same waveform around its maximum. Waveform data of the highest resolved run is shown as a line, while data from lower resolution runs are marked by crosses. Second row: Mass conservation as indicated by the relative baryonic mass change. Third row: The GHG constraint  $t^\mu \nabla_\mu \mathcal{C}_0$ , as an indirect measure for the Hamiltonian constraint  $\mathcal{H}$ . Fourth row: The GHG constraint  $\mathcal{C}_0$ . Bottom snapshots: The dynamics of the head-on collision. Logarithmic density (color coding) and velocity (arrows) are depicted for  $N_{\text{sub}} = 32$ . The patch boundaries are marked by the thick black lines. All error measures are shown for different number of grid patches. The dashed lines are scaled according to second order convergence.

orthonormal null tetrad is built as

$$l^\mu = \frac{1}{\sqrt{2}} (n^\mu + e_r^\mu), \quad k^\mu = \frac{1}{\sqrt{2}} (n^\mu - e_r^\mu), \quad \bar{m}^\mu = \frac{1}{\sqrt{2}} (e_\theta^\mu - ie_\varphi^\mu) \quad (5.5)$$

from the normal vector  $n$  and the spherical polar coordinate basis  $e_r, e_\theta, e_\varphi$ . Note that the latter vectors are normalized with respect to  $\gamma_{ij}$ , e.g. it is  $\gamma_{ij} e_r^i e_r^j = 1$ . Then, we define the Newman-Penrose scalar  $\psi_4$  [Newman and Penrose, 1962], which describes



**Figure 5.2:** Scaling test for the neutron star head-on collision, performed on the High-Performance-Computing cluster “ARA” of the Friedrich Schiller University Jena. The compute nodes are each equipped with two 12 core 2.2 GHz Intel Xeon processors and 128 GB RAM. For a short simulation of final time  $t = 10$ , code running time and corresponding simulation speed are depicted. Ideal scaling is indicated by black dashed lines.

gravitational radiation in the asymptotic limit, as:

$$\psi_4 = -C_{\alpha\beta\mu\nu} k^\alpha \bar{m}^\beta k^\mu \bar{m}^\nu, \quad (5.6)$$

with the Weyl tensor  $C_{\alpha\beta\mu\nu}$ . As  $\psi_4$  transforms like a spin-weight  $-2$  field under tetrad rotations of  $m, \bar{m}$ , we project  $\psi_4$  onto the spherical harmonics of spin-weight  $-2$ ,  ${}_{-2}Y_{lm}$  to analyze the individual modes  $(l, m)$ . For the head-on collision we expect (and confirm) that the  $(2, 0)$  mode is the dominant contribution to the gravitational waveform. The extracted waveform is qualitatively consistent with a similar black hole setup in [Sperhake et al., 2007]. After  $t \approx 1600 = 7.88$  ms the system is in a slightly oscillating single neutron star configuration. At the end of the simulation at  $t = 4000 = 19.71$  ms, a stable star has formed and all fields are changing only marginally. In Fig. 5.1, we also monitored the GHG constraints throughout the full merger process. These converge again with second order, except for a time period shortly after the contact of the stars, when the dynamics are dominated by strong shocks and discontinuities in all matter fields.

As a last benchmark, we tested the scaling behavior of our implementation on the High-Performance-Computing cluster “ARA” of the Friedrich Schiller University Jena. We evolved again the neutron star head-on collision, but only up to final time  $t = 10$  and measured the code running time for different number of processes (or cores). More specifically, we registered the time that the slowest process used for the evolution (no grid/project setup, initial data loading, etc.). The results are shown in Fig. 5.2. The scaling of the code is close to ideal in the whole tested range. The only exception is a scaling factor of 1.4 when the process number is doubled from 16 to 32. This is because each “ARA” node has 24 cores, so that for all larger process numbers, communication between nodes has to be taken into account.

# Chapter 6

## Numerical experiments with a dual foliation

As a first numerical test for the novel DF formalism [Hilditch, 2015] (see Sec. 2.4), we considered the scalar wave equation in flat spacetime:

$$\square\psi = -\partial_T^2\psi + \partial_{\underline{i}}\partial^{\underline{i}}\psi = 0. \quad (6.1)$$

Employing hyperboloidal coordinates and using the DF approach, we want to place the outer boundary of the computational domain at future null infinity. The main idea is to treat the wave equation as usual in the cube and transition region of the grid (see Fig. 3.2, Fig. 6.1). However, in the outer shells, hyperboloidal coordinates and the DF formalism are employed. The results of this test are presented in [Bug2]. In this chapter, we focus on the derivation of the numerical evolution system and on the actual implementation steps that were necessary in the `bamps` code. It should serve as an comprehensible example on the application of the DF formalism.

### Derivation of the DF equations of motion

We start from a first order reduction of the wave equation (6.1) in Cartesian coordinates  $X^\mu = (T, X^{\underline{i}})$ , with  $\pi := -\partial_T\psi$ ,  $\phi_{\underline{i}} := \partial_{\underline{i}}\psi$  and the corresponding reduction constraint  $\mathcal{C}_{\underline{i}} = \partial_{\underline{i}}\psi - \phi_{\underline{i}}$ :

$$\begin{aligned} \partial_T\psi + \pi &= 0 \\ \partial_T\pi + \partial^{\underline{i}}\phi_{\underline{i}} &= 0 \\ \partial_T\phi_{\underline{i}} + \partial_{\underline{i}}\pi - \gamma_2\mathcal{C}_{\underline{i}} &= 0. \end{aligned} \quad (6.2)$$

The  $\gamma_2$  term is a constraint damping term and has been included to relate this wave equation testbed with the future application - the GHG equations (2.44) - as closely as possible. For the next steps, the use of hyperboloidal coordinates and

the compactification, the radial direction is treated in a special way. This is why we have to transform (6.2) to adapted coordinates, which are exactly the local patch coordinates  $u^i$  introduced in Sec. 3.1. Just for this chapter and for the sake of consistency with [Bug2], we rename the index of these coordinates from  $i$  to  $\underline{i}'$  and reserve the index  $i$  for hyperboloidal coordinates. In local patch coordinates  $X^{\underline{\mu}'} = (T, R, \theta^{\underline{A}'})$ , we find

$$\begin{aligned} \partial_T \psi + \pi &= 0 \\ \partial_T \pi + \partial^{\underline{j}'} \phi_{\underline{j}'} + \phi_{\underline{i}'} \partial^{\underline{i}'} J^{\underline{j}'}_{\underline{j}'} &= 0 \\ \partial_T \phi_{\underline{i}'} + \partial_{\underline{i}'} \pi - \gamma_2 \mathcal{C}_{\underline{i}'} &= 0. \end{aligned} \quad (6.3)$$

or in matrix form

$$\partial_T \underbrace{\begin{pmatrix} \psi \\ \pi \\ \phi_{\underline{i}'} \end{pmatrix}}_{\mathbf{u}} = \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -\gamma^{\underline{j}'\underline{k}'} \\ \gamma_2 \delta_{\underline{i}'}^{\underline{j}'} & -\delta_{\underline{i}'}^{\underline{j}'} & 0 \end{pmatrix}}_{\mathbf{A}^{\underline{j}'}} \partial_{\underline{j}'} \underbrace{\begin{pmatrix} \psi \\ \pi \\ \phi_{\underline{k}'} \end{pmatrix}}_{\mathbf{s}} + \underbrace{\begin{pmatrix} -\pi \\ -S^\pi \\ -\gamma_2 \phi_{\underline{i}'} \end{pmatrix}}_{\mathbf{s}}, \quad (6.4)$$

where we introduced the abbreviation

$$S^\pi := \phi_{\underline{i}'} \partial^{\underline{j}'} J^{\underline{i}'}_{\underline{j}'} = \frac{2}{R^2} \left( \phi_R R + \left[ 1 + (\theta^{1'})^2 + (\theta^{2'})^2 \right] \left[ \theta^{1'} \phi_{\theta^{1'}} + \theta^{2'} \phi_{\theta^{2'}} \right] \right). \quad (6.5)$$

This form now complies with the desired form of the ‘‘uppercase’’ PDE (2.53) when we set  $A = 1$ ,  $B^{\underline{i}'} = 0$ . Therefore, (2.46) directly implies  $W = \alpha$ . The specific choice of coordinates for the outer shells given by eqs. (3.8,9) leads to a block structure of the flat metric:

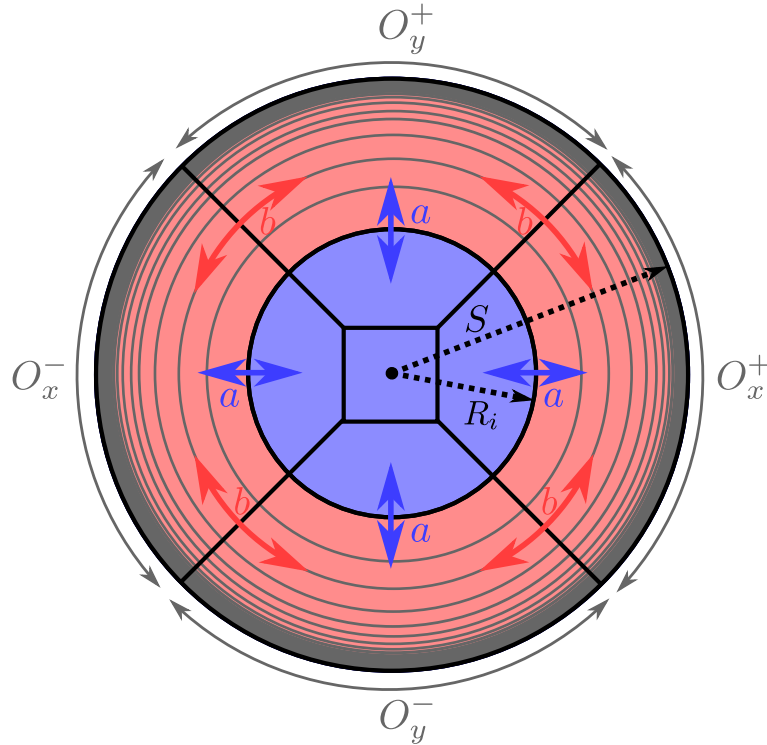
$$\gamma^{\underline{j}'\underline{k}'} = \begin{pmatrix} 1 & 0 \\ 0 & q^{\theta^{\underline{A}'}\theta^{\underline{B}'}} \end{pmatrix}, \quad q^{\theta^{\underline{A}'}\theta^{\underline{B}'}} := \frac{1 + (\theta^{1'})^2 + (\theta^{2'})^2}{R^2} \begin{pmatrix} 1 + (\theta^{1'})^2 & \theta^{1'} \theta^{2'} \\ \theta^{1'} \theta^{2'} & 1 + (\theta^{2'})^2 \end{pmatrix}. \quad (6.6)$$

Once we give a concrete coordinate transformation, we can easily apply the DF recipe as derived in Sec. 2.4. As suggested, we want to use hyperboloidal coordinates  $x^\mu = (t, r, \theta^A)$  as the lowercase coordinate system on the outer shells. The coordinate transformation is of the form:

$$T(t, r) = t + H(R(r)), \quad R(t, r) = R(r) = R_i + \frac{r - R_i}{\Omega(r)}, \quad \theta^{\underline{A}'} = \delta_A^{\underline{A}'} \theta^A, \quad (6.7)$$

where  $H$  is called the *height function* and  $\Omega$  is called the *compress function*.  $R_i$  is the radius, at which outer shells and transition shells share a boundary. The compress function allows for a compactification, i.e. to ‘‘squeeze’’ the infinite spatial domain





**Figure 6.1:** The DF approach as adapted to the bumps grid layout: In the inner region  $r = R < R_i$  (blue), the evolution equations are given in global Cartesian coordinates, treated as described in Sec. 3.1. In the outer shells (red), the novel DF scheme is employed. Hyperboloidal coordinates and a spatial compactification are used. The gray circles are circles of uniformly increasing Cartesian radius. The coordinate radius  $r = S$  is mapped to  $R = \infty$ . The blue (red) arrows denote the type a (b) inner-to-outer (shell-to-shell) patch communications, that require a non-trivial transformation step for the tensor variables (see text).

into a finite coordinate interval. We choose

$$\Omega(r) = 1 - \left( \frac{r - R_i}{S - R_i} \right)^n \quad (6.8)$$

with the compactification exponent  $n = 2$ . Eq. (6.8) fulfills  $R(R_i) = R_i$  and maps  $r = S$  to  $R = \infty$  (gray circles in Fig. 6.1). Interestingly, the explicit form of the height function itself is not needed for this application. However, its derivative enters the numerical scheme. Demanding unit outgoing radial coordinate lightspeed, one finds the condition  $H' = 1 - R'^{-1}$  for  $H'$ . The Jacobians for the transformation are given by

$$J^{\underline{\mu}'}_{\mu} = \begin{pmatrix} 1 & H'R' & 0 \\ 0 & R' & 0 \\ 0 & 0 & \mathbf{1}_2 \end{pmatrix}, \quad J^{\mu}_{\underline{\mu}'} = \begin{pmatrix} 1 & -H' & 0 \\ 0 & \frac{1}{R'} & 0 \\ 0 & 0 & \mathbf{1}_2 \end{pmatrix}. \quad (6.9)$$

Equating this to (2.52), we immediately read off the lowercase boost vector,

$$v_r = -\frac{H'R'}{W}, \quad v_{\theta^A} = 0, \quad (6.10)$$

and from that the lowercase shift,

$$\beta^r = -W^2 \frac{H'}{R'}, \quad \beta^{\theta^A} = 0. \quad (6.11)$$

Then using the  $T$ - $t$ -component of the Jacobian, we get  $1 = \frac{W}{A} (\alpha - \beta^i v_i)$  and together with the expressions for shift and boost vector this yields

$$W^2 = \frac{1}{1 - H'^2}. \quad (6.12)$$

In the same way, we find the uppercase boost vector,

$$V_R = \frac{\alpha}{W} H' = H', \quad V_{\theta^A} = 0, \quad (6.13)$$

and the vanishing combination

$$W \left( v^i - \frac{\beta^i}{\alpha} \right) = 0. \quad (6.14)$$

The spatial part of the Jacobian is simply  $\Phi^i_{j'} = \text{diag}(R'^{-1}, 1, 1)$ . Now everything is in place to write down the DF transformed PDEs (2.56). We calculate the matrix  $(1 + \mathbf{A}^{j'} V_{j'})$  and its inverse:

$$\left( 1 + \mathbf{A}^{j'} V_{j'} \right)^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -H' & 0 \\ \gamma_2 H' & -H' & 1 & 0 \\ 0 & 0 & 0 & \mathbf{1}_2 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{\gamma_2 H'^2}{1-H'^2} & \frac{1}{1-H'^2} & \frac{H'}{1-H'^2} & 0 \\ -\frac{\gamma_2 H'}{1-H'^2} & \frac{H'}{1-H'^2} & \frac{1}{1-H'^2} & 0 \\ 0 & 0 & 0 & \mathbf{1}_2 \end{pmatrix}. \quad (6.15)$$

Eq. (2.56) simplifies to

$$\begin{aligned} \partial_t \mathbf{u} &= \left( 1 + \mathbf{A}^{j'} V_{j'} \right)^{-1} \left( \mathbf{A}^R R'^{-1} \partial_r \mathbf{u} + \mathbf{A}^{\theta^A} \delta_{\theta^A}^{\theta^A} \partial_{\theta^A} \mathbf{u} + \mathbf{S} \right) \\ &= - \left( 1 + \mathbf{A}^{j'} V_{j'} \right)^{-1} \begin{pmatrix} \pi \\ R'^{-1} \partial_r \phi_R + q^{\theta^A \theta^B} \delta_{\theta^A}^A \partial_{\theta^A} \phi_{\theta^B} + S^\pi \\ R'^{-1} \partial_r \pi + \gamma_2 (\phi_R - R'^{-1} \partial_r \psi) \\ \delta_{\theta^A}^A \partial_{\theta^A} \pi + \gamma_2 (\phi_{\theta^A} - \delta_{\theta^A}^A \partial_{\theta^A} \psi) \end{pmatrix}. \end{aligned} \quad (6.16)$$

Performing the matrix multiplication gives the final DF evolution system for the outer shells as implemented in the `bamps` code. Note that the derivatives are taken with re-

spect to the lowercase (hyperboloidal) coordinates, while the tensor components (e.g.  $\phi_{\underline{y}}$ ) are still with respect to the uppercase coordinate basis. We already emphasized this feature in the introductory section 2.4. The full outer shell evolution system is:

$$\begin{aligned}
\partial_t \psi &= -\pi \\
\partial_t \pi &= -\frac{1}{1-H'^2} \left( R'^{-1} \partial_r \phi_R + q^{\theta A' \theta B'} \delta_{\theta A'}^{\theta A} \partial_{\theta A} \phi_{\theta B'} + S^\pi \right) \\
&\quad - \frac{H'}{1-H'^2} \left( R'^{-1} \partial_r \pi - \gamma_2 \left( R'^{-1} \partial_r \psi - \phi_R + H' \pi \right) \right) \\
\partial_t \phi_R &= -\frac{H'}{1-H'^2} \left( R'^{-1} \partial_r \phi_R + q^{\theta A' \theta B'} \delta_{\theta A'}^{\theta A} \partial_{\theta A} \phi_{\theta B'} + S^\pi \right) \\
&\quad - \frac{1}{1-H'^2} \left( R'^{-1} \partial_r \pi - \gamma_2 \left( R'^{-1} \partial_r \psi - \phi_R + H' \pi \right) \right) \\
\partial_t \phi_{\theta A'} &= -\delta_{\theta A'}^{\theta A} \partial_{\theta A} \pi + \gamma_2 \left( \delta_{\theta A'}^{\theta A} \partial_{\theta A} \psi - \phi_{\theta A'} \right). \tag{6.17}
\end{aligned}$$

Of course, the same system can be derived by straightforwardly using the full Jacobian  $J^\mu_{\underline{\mu}'}$ . Nevertheless, the DF approach is adapted to the 3+1 picture and gives a geometrical interpretation of this transformation. Moreover, it allows to the evolution of the Jacobian, i.e. to change the properties of the coordinate system dynamically during the evolution. In [Bug2] the dynamic lightspeed control for the DF GHG system is derived as a possible future application of this mechanism.

### Implementation

To evolve (6.17) forward in time, the spatial derivatives are again calculated using the pseudospectral method (see Sec. 3.1). However, now these derivatives (with respect to the local patch coordinates) can be directly used without a Jacobian multiplication. This is different to what is the usual approach in bamps and what is still done in the inner part of the domain, where a global Cartesian coordinate basis is employed. This seems like an advantage, but comes with an additional difficulty: Since in (6.17) tensor components are given in a patch type dependent local basis, these valences have to be transformed for the patch-to-patch communication. In particular since the outer shells are different mappings of the master patch, the ‘‘angular’’ coordinates have different meanings on patches with different orientation. We denote this shell-to-shell communication as *type b* in Fig. 6.1. The same issue occurs for the communication between the inner part of the domain and the outer shells. Here the Cartesian components of tensors have to be converted to the local patch components and vice versa. We denote this inner-to-outer communication as *type a* in Fig. 6.1. Of course, both these transformations are straight forward and rather simple for this example. We just want to list it as one additional feature that had to be added to the code infrastructure. For type a, the covector components

**Table 6.1:** Listing of matrices for the transformation step during patch-to-patch communication in the DF scheme. In the top part of the table, the matrices  $M$  for the shell-to-shell (type b) communication (6.19) are listed, depending on target and source patch orientation. The local “angular” coordinates  $\theta^1, \theta^2$  are those of the target patch. In the bottom row of the table, the permutation matrices for the inner-to-outer (type a) communication are given according to (3.11).

Target \ Source	$O_x^+$	$O_x^-$	$O_y^+$	$O_y^-$	$O_z^+$	$O_z^-$
$O_x^+$	-	-	$\begin{pmatrix} 0 & +1 \\ -1 & -\theta^1 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ +1 & \theta^1 \end{pmatrix}$	$\begin{pmatrix} -\theta^2 & -1 \\ +1 & 0 \end{pmatrix}$	$\begin{pmatrix} \theta^2 & -1 \\ +1 & 0 \end{pmatrix}$
$O_x^-$	-	-	$\begin{pmatrix} 0 & +1 \\ -1 & \theta^1 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ +1 & -\theta^1 \end{pmatrix}$	$\begin{pmatrix} -\theta^2 & +1 \\ -1 & 0 \end{pmatrix}$	$\begin{pmatrix} \theta^2 & +1 \\ -1 & 0 \end{pmatrix}$
$O_y^+$	$\begin{pmatrix} -\theta^2 & -1 \\ +1 & 0 \end{pmatrix}$	$\begin{pmatrix} \theta^2 & -1 \\ +1 & 0 \end{pmatrix}$	-	-	$\begin{pmatrix} 0 & +1 \\ -1 & -\theta^1 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ +1 & \theta^1 \end{pmatrix}$
$O_y^-$	$\begin{pmatrix} -\theta^2 & +1 \\ -1 & 0 \end{pmatrix}$	$\begin{pmatrix} \theta^2 & +1 \\ -1 & 0 \end{pmatrix}$	-	-	$\begin{pmatrix} 0 & +1 \\ -1 & \theta^1 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ +1 & -\theta^1 \end{pmatrix}$
$O_z^+$	$\begin{pmatrix} 0 & +1 \\ -1 & -\theta^1 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ +1 & \theta^1 \end{pmatrix}$	$\begin{pmatrix} -\theta^2 & -1 \\ +1 & 0 \end{pmatrix}$	$\begin{pmatrix} \theta^2 & -1 \\ +1 & 0 \end{pmatrix}$	-	-
$O_z^-$	$\begin{pmatrix} 0 & +1 \\ -1 & \theta^1 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ +1 & -\theta^1 \end{pmatrix}$	$\begin{pmatrix} -\theta^2 & +1 \\ -1 & 0 \end{pmatrix}$	$\begin{pmatrix} \theta^2 & +1 \\ -1 & 0 \end{pmatrix}$	-	-
$P_{\underline{i}\underline{j}}^i$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix}$

transform as  $\phi_{\underline{i}} = P_{\underline{i}}^j J_{\underline{j}}^k \phi_k$  and  $\phi_i = J_i^j P_j^k \phi_k$  subject to the master patch Jacobian for transformation eqs. (3.8,9)

$$\begin{aligned}
 J_{\underline{i}}^j &= (1 + (\theta^1)^2 + (\theta^2)^2)^{-3/2} \begin{pmatrix} (1 + (\theta^1)^2 + (\theta^2)^2)^2 & -r\theta^1 & -r\theta^2 \\ \theta^1 (1 + (\theta^1)^2 + (\theta^2)^2)^2 & r(1 + (\theta^2)^2) & -r\theta^1\theta^2 \\ \theta^2 (1 + (\theta^1)^2 + (\theta^2)^2)^2 & -r\theta^1\theta^2 & r(1 + (\theta^2)^2) \end{pmatrix}, \\
 J_i^j &= (1 + (\theta^1)^2 + (\theta^2)^2)^{-1/2} r^{-1} \begin{pmatrix} r & r\theta^1 & r\theta^2 \\ -(1 + (\theta^1)^2 + (\theta^2)^2)\theta^1 & 1 + (\theta^1)^2 + (\theta^2)^2 & 0 \\ -(1 + (\theta^1)^2 + (\theta^2)^2)\theta^2 & 0 & 1 + (\theta^1)^2 + (\theta^2)^2 \end{pmatrix}, \tag{6.18}
 \end{aligned}$$

and a permutation of coordinates (3.11). The permutation matrices depend on the shell orientation and are listed in Tab. 6.1. In type b, the shell-to-shell communication, the local patch components from the source shell patch have to be transformed to local patch components of the target shell patch with a different orientation. The first local patch coordinate  $r$  has the same meaning on all shell patches, independent from the orientation. Therefore, its transformation is trivial and the mapping is given by:

$$\phi_i^{\text{target}} = \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{M} \end{pmatrix} \phi_j^{\text{source}}. \tag{6.19}$$

However, the mapping  $M$  of the “angular” components is non-trivial and depends on both source and target shell orientation. We list these matrices in Tab. 6.1 as well.

Although the equations of motion (6.17) have regular coefficients on the right hand side, they are formed from divergent quantities as  $r \rightarrow S$ . Therefore care is needed in the implementation if we are to maintain accuracy and avoid “NaNs”. Fortunately the complete equations can be built from the following regular combinations as implemented in bamps:

$$\begin{aligned} \frac{1}{R} &= \frac{(r-S)(r-2R_i+S)}{r^2R_i+R_i^3-r(3R_i^2-2R_iS+S^2)} \xrightarrow{r \rightarrow S} 0, \\ \frac{1}{R'} &= \frac{(r-S)^2(r-2R_i+S)^2}{(R_i-S)^2(r^2-2rR_i+2R_i^2-2R_iS+S^2)} \xrightarrow{r \rightarrow S} 0, \\ \frac{R'}{R^2} &= \frac{(R_i-S)^2(r^2-2rR_i+2R_i^2-2R_iS+S^2)}{(r^2R_i+R_i^3-r(3R_i^2-2R_iS+S^2))^2} \xrightarrow{r \rightarrow S} \frac{2}{(S-R_i)^2}. \end{aligned} \quad (6.20)$$

Since the height function is related to  $R'$ , we also find

$$\frac{R'^{-1}}{1-H'^2} = -\frac{R'}{2R'-1} \xrightarrow{r \rightarrow S} -\frac{1}{2}, \quad \frac{H'R'^{-1}}{1-H'^2} = \frac{R'-1}{2R'-1} \xrightarrow{r \rightarrow S} \frac{1}{2}. \quad (6.21)$$

By choosing the constraint damping to falloff as  $\gamma_2 \sim R^{-1}$ , all expressions in (6.17) have a regular limit, except for the terms containing  $S^\pi$ . Looking at (6.5), the dominating  $\phi_R$  coefficient is proportional to  $R'/R$ , which diverges as  $r \rightarrow S$ . Luckily, we can express the field  $\phi_R$  in terms of ingoing and outgoing characteristic  $\phi_R = \frac{1}{2}(u^+ - u^-)$  and make use of the characteristic fields falloff  $u^+ \sim R^{-1}$ ,  $u^- \sim R^{-2}$ , which exactly compensates the divergent coefficient. We apply L'Hôpital's rule to find

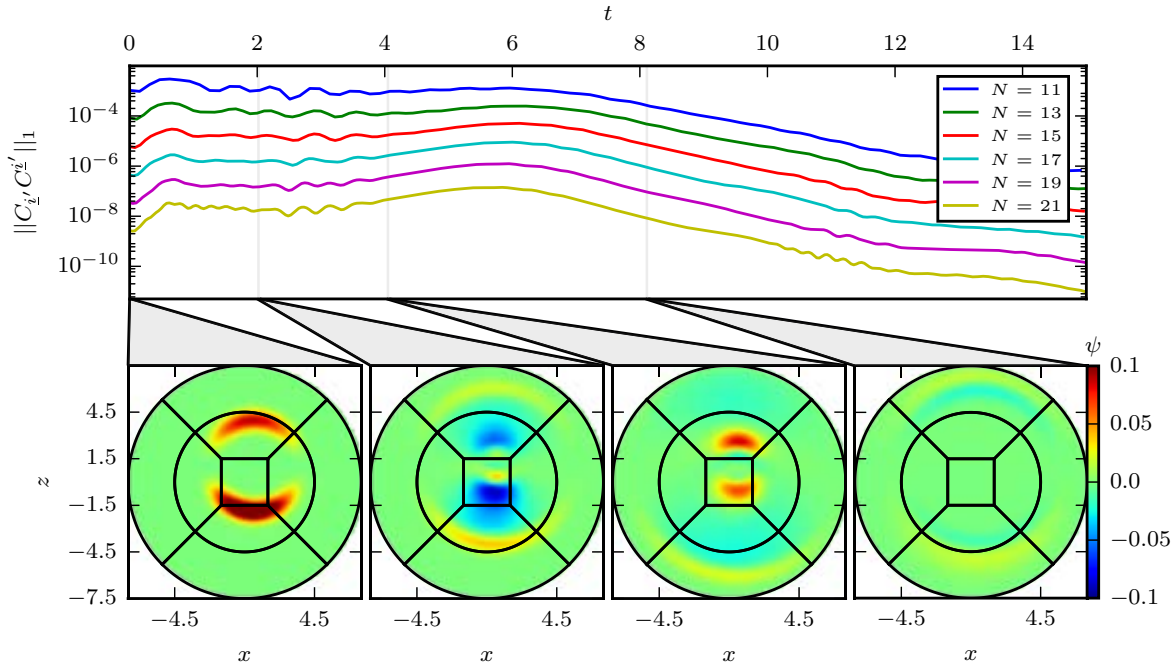
$$\begin{aligned} -\frac{2}{R(1-H'^2)}\phi_R &= \frac{R'^2}{R(1-2R')} (u^+ - u^-) \xrightarrow{r \rightarrow S} \frac{\partial_r(u^+)}{2} = \frac{1}{2}\partial_r(\pi + \phi_R) \\ -\frac{2H'}{R(1-H'^2)}\phi_R &= \frac{R'(R'-1)}{R(2R'-1)} (u^+ - u^-) \xrightarrow{r \rightarrow S} \frac{\partial_r(u^+)}{2} = \frac{1}{2}\partial_r(\pi + \phi_R). \end{aligned} \quad (6.22)$$

This is implemented in the obvious way by adjusting the sources at, and only at null-infinity  $r = S$ .

## Numerical test

With regard to the future application - the extraction of gravitational waves at future null-infinity - we test our DF implementation with a pure (2, 2) mode wave excitation. The initial data is given by

$$\psi(r, \theta, \varphi) = \sqrt{\frac{15}{32\pi}} \exp \left[ -\left( \frac{r-r_0}{\sigma} \right)^2 \right] \frac{\sin^2 \theta \cos 2\varphi}{\sqrt{1+R(r)^2}}, \quad (6.23)$$



**Figure 6.2:** Wave equation evolution with a dual foliation. Top: Convergence of the norm of the constraints  $C_i$  as resolution is increased in our wave equation experiments. Bottom: The basic dynamics of  $\psi$  in the plane  $y = 0$ , starting with initial data (6.23). We see that initially there is a pulse traveling outwards which is absorbed at null-infinity, the outer boundary of the plot. After traveling inwards the second pulse also propagates off of the grid. This evolution was performed in full 3D with  $19^3$  points per patch. The patches each consist of one grid. Their boundaries are marked by the thick black lines

where  $\theta$  and  $\phi$  are standard spherical coordinates and should not be confused with the shell coordinates  $\theta^1, \theta^2$ . Furthermore, the radii  $r$  and  $R(r)$  in (6.23) denote the distance to another origin, than the grid origin. For the present full 3D tests, we shifted this origin for the initial data to  $x = 1.0, y = 0.3, z = 0.5$ . We further chose  $r_0 = 3, \sigma = 0.6$ , the constraint damping parameter  $\gamma_2 = 1$  and the grid measures  $R_i = 4.5, S = 7.5$ . We want to emphasize that since the outer boundary  $r = S$  represents future null-infinity, no outer boundary conditions are required. The tests were performed on a desktop machine with 32 GB of memory and 8 4.00 GHz Intel i7 cores. We run the code in parallel with 4 MPI instances. With our standard setup, a 3D run with polynomial degree  $N = 20$  per dimension computes at roughly 27 time units per hour. The dynamics of a typical evolution with initial data (6.23) are presented in Fig. 6.2. The wave propagates out and leaves the domain through null-infinity almost without reflection. We find that the expected fall-off of the field at the outer boundary is violated as differs slightly from zero. This effect however converges exponentially with resolution. The outgoing wave does leave behind a small amount of noise which we see as  $C_i$  constraint violation. Also this violation converges exponentially with resolution as demonstrated in Fig. 6.2. Since our asymptotics requires a constraint damping parameter  $\gamma_2$  falloff as  $R$  increases, a concern may be that the constraint

damping scheme is ineffective in these evolutions. However, from our tests we still find effective constraint damping.

In extended numerical tests, we also investigated the influence of spectral filters, regularization, source terms and other than  $n = 2$  compactification exponents. In [Bug2], all these additional influencing factors of the scheme are discussed in full detail. However, a crucial remark on filtering should be given within this thesis: Using the filter in all directions, which is expected to be necessary for nonlinear problems, we find that the method is unstable. We suspected and confirmed, that this is caused by the fact that the radial filter does not respect the expected falloff of the fields. Therefore, a more carefully constructed radial filter should be used in future applications.





# Chapter 7

## Conclusion

### Summary

In this thesis, a new numerical method for the simulation of neutron stars in full general relativity was implemented and tested. It is based on a pseudospectral penalty method for evolving the spacetime variables, as presented in [Weyhausen, 2014; Hilditch et al., 2016], and a DG scheme in combination with HRSC techniques for the matter evolution. We thus extended the existing `bamps` code to make it applicable for GRHD simulations.

At the first stage of implementation we reproduced standard results, e.g. for advection and burgers equation and the equations of SRHD. In these tests we confirmed a clean convergence order  $N + 1$  for DG schemes with maximal polynomial order  $N$  and smooth solutions. During a collaboration with the authors of [Kidder et al., 2016], we found that the analytic mass matrix in a DG method leads to smaller numerical errors than those from a mass lumped (diagonal) matrix. For low order DG schemes, this can result in  $\sim 20$  times smaller errors. We further confirmed that numerically satisfying the metric identities is crucial for the mass conservation property of a DG scheme, as stated in [Teukolsky, 2015a].

We implemented two different strategies to capture shocks in matter simulations: WENO limiting in the context of DG methods [Qiu and Shu, 2005; Zhao and Tang, 2013] and the SV evolution [Radice and Rezzolla, 2011]. We found that both approaches allow the treatment of special relativistic shock problems. In the case of WENO limiting, the well-definedness of ideal reconstruction weights is non-trivial. We gave a proof that in a WENO- $(2w + 1)$  method these weights are well-defined almost everywhere in the DG element, except for at most  $w + 1$  points.

In our static spacetime single neutron star tests, we found that a stable, convergent simulation of these systems is not feasible with a pure DG scheme. The stellar surface and the artificial atmosphere in neutron star simulations are responsible for instabilities when a pure DG method is employed. Combining it instead with the WENO or SV

method allows for stable simulations with convergence order  $\sim 2$  [Bug1]. In our tests, the SV method lead to smaller (similar) errors as compared to the corresponding WENO-3 (WENO-5) runs. On the other hand, the SV method comes at significantly higher computational cost. Furthermore, we observed instabilities in the hybrid DG/SV approach due to the neglect of conservation at DG/SV interfaces [Kopriva, 1996].

Apart from the one-dimensional non-rotating neutron star runs in Cowling approximation, we extended our code and performed tests in higher dimensions (3D TOV star), for rotating stars (RNS) and for truly dynamical systems in full GR (migration test, head-on collision). Throughout these tests we confirmed second order convergence for baryonic mass conservation, Hamilton and momentum constraint and the GHG constraints. We exploit geometric symmetries in our simulations by employing the cartoon method. Therefore, code running times for the setups in this thesis are usually not longer than a few hours.

We showed that binary systems of neutron stars can be evolved with the `bamps` code. We used the pseudospectral penalty method for the spacetime evolution and the SV method to evolve the matter variables. For these runs, the scaling of our implementation on a High-Performance-Computing cluster is close to ideal scaling. Furthermore we extracted the corresponding gravitational wave signal for the coalescence.

In order to improve the gravitational wave extraction in future NR codes, we accomplished the first implementation of the DF formalism [Hilditch, 2015]. It enabled us to evolve the flat spacetime wave equation on a numerical grid, which extends to future null infinity. Testing our implementation, we were able to obtain stable and converging simulations of the wave equation on compactified hyperboloidal shells [Bug2]. Those can be seen as both, a proof of principle of the DF approach and a prototype application, aiming for the DF implementation of the GHG equations.

### Future prospects

Besides the advances presented in [Zumbusch, 2009; Radice and Rezzolla, 2011; Teukolsky, 2015a; Kidder et al., 2016], this thesis can be seen as another step towards a DG framework in NR. However, still more work has to be done to compete with established and optimized finite volume/difference codes. Obviously in our implementation a stabilization of the DG/SV method is desirable as the next short-term goal. This can be realized by implementing the mortar method as in [Choi, 2015]. Furthermore, the multidimensional reconstruction in the SV method should be improved as discussed in Sec. 3.3.2. As a last step, the implementation of a more elaborate troubled cell indicator for the SV method would then enable a hybrid DG/SV evolution of binary neutron stars.

As we found the DG-WENO methods to be fast and reliable in our shock capturing tests, a generalization to curvilinear grids is another possible mid-term goal. This is an inevitable step towards the BNS simulation with DG-WENO methods, as it requires a two-dimensional grid with spherical surface. This generalization is ambitious, as the effects of the multidimensional WENO reconstruction on the code efficiency are unclear.

Although shock capturing methods were necessary in our implementation and a pure DG method was not sufficient for convergent neutron star simulations, we suppose that this is not generally true. A numerical grid adapted to the neutron star surface might enable pure DG neutron star simulations in the future. However, this would involve the proper analytical treatment of the matter/vacuum phase transition at the star surface. A first step towards a better understanding of the characteristic behavior of the GRHD equations in the vacuum limit can be done by means of the DF formalism [Bug3].

The DF formalism itself is still in the early stages of development, but could become a main utility in future NR. An obvious application is the GHG evolution on compactified hyperboloidal shells and the extraction of gravitational waves at future null infinity. Of course, a full DF GHG implementation is only feasible with a lot of effort. One of the first steps in that direction would be the implementation of a radial filter that respects the expected falloff of the solution.



# Appendix

## A Recovery of the primitive variables

What we describe in this appendix is adopted from [Martí and Müller, 1996; Thierfelder et al., 2011]. The conserved variables  $(D, p_i, \tau)$  for the matter evolution can be expressed by the primitive variables  $(\rho, \varepsilon, p, v^i)$  through the following mapping:

$$W(v_i) = (1 - v^i v_i)^{-\frac{1}{2}} \quad (\text{A.1})$$

$$D = \rho W \quad (\text{A.2})$$

$$p_i = \rho h W^2 v_i \quad (\text{A.3})$$

$$\tau = (\rho h W^2 - p - \rho W) \quad (\text{A.4})$$

Because of the typical non-linearity introduced by the relativistic Lorentz factor  $W$ , we can not find an analytical inversion of this mapping. Instead, we recover the primitive variables using the following relations:

$$p_i p^i = (\rho h W^2)^2 v^2 = (\tau + D + p)^2 v^2 =: A(p)^2 v^2 \quad (\text{A.5})$$

$$W(p)^2 = \frac{1}{1 - v^2} = \frac{A^2}{A^2 - p_i p^i}, \quad \frac{dW}{dp} = -\frac{p_i p^i}{\sqrt{A^2 - p_i p^i}^3} \quad (\text{A.6})$$

$$\rho(p) = \frac{D}{W}, \quad \frac{d\rho}{dp} = \frac{D p_i p^i}{\sqrt{A^2 - p_i p^i} A^2} \quad (\text{A.7})$$

$$\varepsilon(p) = h - 1 - \frac{p}{\rho} = \frac{A}{\rho W^2} - 1 - \frac{p}{\rho} = \frac{1}{D} \left( \frac{A}{W} - D - p W \right) \quad (\text{A.8})$$

$$\frac{d\varepsilon}{dp} = \frac{1}{D} \frac{p p_i p^i}{\sqrt{A^2 - p_i p^i}^3}. \quad (\text{A.9})$$

Since we managed to express  $\rho$  and  $\varepsilon$  as functions of  $p$  and the conserved variables, we can apply a root finding procedure on

$$f(p) = p_{\text{EOS}}(\rho(p), \varepsilon(p)) - p \quad (\text{A.10})$$

to find  $p$ . We employ a Newton-Raphson algorithm

$$p^{(i+1)} = p^{(i)} - \frac{f(p)}{f'(p)} \quad (\text{A.11})$$

$$f'(p) = \frac{dp_{\text{EOS}}}{d\rho} \frac{d\rho}{dp} + \frac{dp_{\text{EOS}}}{d\varepsilon} \frac{d\varepsilon}{dp} - 1, \quad (\text{A.12})$$

and use as  $p^{(0)}$  either the pressure value of the previous timestep, or a value so that  $A^2 - p_i p^i$  is positive. From what we derived above, we can immediately find  $\rho$  and  $\varepsilon$  from  $p$  and furthermore

$$v^i = \frac{p^i}{A}. \quad (\text{A.13})$$

If  $\rho$  is smaller than some freely specifiable threshold  $100\rho_{\text{atm}}$  at some point in the primitive recovery, we set  $\rho = \rho_{\text{atm}}$  to avoid unphysical values in the density. Of course, the implementation of this artificial atmosphere is unphysical by itself and leads to additional numerical errors. Therefore,  $\rho_{\text{atm}}$  should be chosen as small as possible.

## B Derivation of the GHG representation of the Ricci tensor

The most complicated part in verifying (2.39) is the bookkeeping of all contractions of  $g_{\mu\nu}$  and its first derivatives. Thus, we introduce the following abbreviations:

$$A_{\mu\nu}^1 = g^{\alpha\beta} g^{\delta\epsilon} g_{\mu\alpha,\delta} g_{\nu\beta,\epsilon}, \quad A_{\mu\nu}^2 = g^{\alpha\beta} g^{\delta\epsilon} g_{\mu\alpha,\delta} g_{\nu\epsilon,\beta}, \quad (\text{B.14})$$

$$B_{\mu\nu} = g^{\alpha\beta} g^{\delta\epsilon} g_{\mu\alpha,\delta} g_{\beta\epsilon,\nu}, \quad (\text{B.15})$$

$$C_{\mu\nu}^1 = g^{\alpha\beta} g^{\delta\epsilon} g_{\mu\alpha,\nu} g_{\beta\delta,\epsilon}, \quad C_{\mu\nu}^2 = g^{\alpha\beta} g^{\delta\epsilon} g_{\mu\alpha,\nu} g_{\delta\epsilon,\beta}, \quad (\text{B.16})$$

$$D_{\mu\nu} = g^{\alpha\beta} g^{\delta\epsilon} g_{\alpha\delta,\mu} g_{\beta\epsilon,\nu}, \quad (\text{B.17})$$

$$E_{\mu\nu}^1 = g^{\alpha\beta} g^{\delta\epsilon} g_{\mu\nu,\alpha} g_{\beta\delta,\epsilon}, \quad E_{\mu\nu}^2 = g^{\alpha\beta} g^{\delta\epsilon} g_{\mu\nu,\alpha} g_{\delta\epsilon,\beta}. \quad (\text{B.18})$$

We also use

$$\partial_\alpha (g^{\mu\beta} g_{\beta\nu}) = 0 \quad \Rightarrow \quad g_{,\alpha}^{\mu\nu} = -g^{\mu\beta} g^{\nu\delta} g_{\beta\delta,\alpha}. \quad (\text{B.19})$$

For the left hand side of (2.39) (the Ricci tensor) we evaluate the terms:

$$\begin{aligned} \partial_\alpha \Gamma_{\mu\nu}^\alpha &= \frac{1}{2} g^{\alpha\beta} (g_{\nu\beta,\mu\alpha} + g_{\mu\beta,\nu\alpha} - g_{\mu\nu,\alpha\beta}) + \frac{1}{2} g_{,\alpha}^{\alpha\beta} (g_{\nu\beta,\mu} + g_{\mu\beta,\nu} - g_{\mu\nu,\beta}) \\ &\stackrel{\text{(B.19)}}{=} \frac{1}{2} g^{\alpha\beta} (2g_{\alpha(\mu,\nu)\beta} - g_{\mu\nu,\alpha\beta}) - \frac{1}{2} (2C_{(\mu\nu)}^1 - E_{\mu\nu}^1) \end{aligned} \quad (\text{B.20})$$

$$\partial_\mu \Gamma_{\alpha\nu}^\alpha = \frac{1}{2} g^{\alpha\beta} g_{\alpha\beta,\mu\nu} + \frac{1}{2} g_{,\mu}^{\alpha\beta} g_{\alpha\beta,\nu} \stackrel{\text{(B.19)}}{=} \frac{1}{2} g^{\alpha\beta} g_{\alpha\beta,\mu\nu} - \frac{1}{2} D_{\mu\nu} \quad (\text{B.21})$$

$$\Gamma_{\mu\nu}^\alpha \Gamma_{\beta\alpha}^\beta = \frac{1}{4} g^{\alpha\beta} g^{\delta\epsilon} (2g_{\beta(\mu,\nu)} - g_{\mu\nu,\beta}) g_{\delta\epsilon,\alpha} = \frac{1}{4} (2C_{(\mu\nu)}^2 - E_{\mu\nu}^2) \quad (\text{B.22})$$

$$\begin{aligned} \Gamma_{\mu\alpha}^\beta \Gamma_{\nu\beta}^\alpha &= \frac{1}{4} g^{\beta\delta} g^{\alpha\epsilon} (g_{\alpha\delta,\mu} + g_{\mu\delta,\alpha} - g_{\mu\alpha,\delta}) (g_{\beta\epsilon,\nu} + g_{\nu\epsilon,\beta} - g_{\nu\beta,\epsilon}) \\ &= \frac{1}{4} (D_{\mu\nu} - 2A_{\mu\nu}^1 + 2A_{\mu\nu}^2). \end{aligned} \quad (\text{B.23})$$

We express the terms of the right hand side of (2.39) as:

$$\begin{aligned} \nabla_{(\mu} \Gamma_{\nu)} &= \partial_{(\mu} g^{\alpha\beta} \Gamma_{\nu)\alpha\beta} - \Gamma_{(\mu\nu)}^\delta g^{\alpha\beta} \Gamma_{\delta\alpha\beta} \\ &= \frac{1}{2} g^{\alpha\beta} (2g_{\alpha(\mu,\nu)\beta} - g_{\alpha\beta,\mu\nu}) + \frac{1}{2} g_{(\mu}^{\alpha\beta} (2g_{|\nu)\alpha,\beta} - g_{\alpha\beta,|\nu}) - \Gamma_{\mu\nu}^\delta g^{\alpha\beta} \Gamma_{\delta\alpha\beta} \\ &\stackrel{\text{(B.19)}}{=} \frac{1}{2} g^{\alpha\beta} (2g_{\alpha(\mu,\nu)\beta} - g_{\alpha\beta,\mu\nu}) - \frac{1}{2} g^{\alpha\delta} g^{\beta\epsilon} (2g_{\delta\epsilon,(\mu} g_{\nu)\beta,\alpha} - g_{\delta\epsilon,(\mu} g_{\alpha\beta,|\nu)}) - \\ &\quad \frac{1}{4} g^{\alpha\beta} g^{\delta\epsilon} (g_{\mu\delta,\nu} + g_{\nu\delta,\mu} - g_{\mu\nu,\delta}) (g_{\alpha\epsilon,\beta} + g_{\beta\epsilon,\alpha} - g_{\alpha\beta,\epsilon}) \\ &= \frac{1}{2} g^{\alpha\beta} (2g_{\alpha(\mu,\nu)\beta} - g_{\alpha\beta,\mu\nu}) - \frac{1}{2} (2B_{(\mu\nu)} - D_{\mu\nu}) - \\ &\quad \frac{1}{4} (4C_{(\mu\nu)}^1 - 2C_{(\mu\nu)}^2 - 2E_{(\mu\nu)}^1 + E_{\mu\nu}^2) \end{aligned} \quad (\text{B.24})$$

$$\begin{aligned} g^{\alpha\beta} g^{\delta\epsilon} \Gamma_{\mu\alpha\delta} \Gamma_{\nu\beta\epsilon} &= \frac{1}{4} g^{\alpha\beta} g^{\delta\epsilon} (g_{\mu\alpha,\delta} + g_{\mu\delta,\alpha} - g_{\alpha\delta,\mu}) (g_{\nu\beta,\epsilon} + g_{\nu\epsilon,\beta} - g_{\beta\epsilon,\nu}) \\ &= \frac{1}{4} (2A_{\mu\nu}^1 + 2A_{\mu\nu}^2 - 4B_{(\mu\nu)} + D_{\mu\nu}). \end{aligned} \quad (\text{B.25})$$

Finally, we verify the relation for the Ricci tensor

$$\begin{aligned} R_{\mu\nu} &= \partial_\alpha \Gamma_{\mu\nu}^\alpha - \partial_\mu \Gamma_{\alpha\nu}^\alpha + \Gamma_{\mu\nu}^\alpha \Gamma_{\beta\alpha}^\beta - \Gamma_{\mu\alpha}^\beta \Gamma_{\nu\beta}^\alpha \\ &= \frac{1}{2} g^{\alpha\beta} (2g_{\alpha(\mu,\nu)\beta} - g_{\mu\nu,\alpha\beta} - g_{\alpha\beta,\mu\nu}) - \\ &\quad \frac{1}{4} (4C_{(\mu\nu)}^1 - 2E_{\mu\nu}^1 - D_{\mu\nu} - 2C_{(\mu\nu)}^2 + E_{\mu\nu}^2 - 2A_{\mu\nu}^1 + 2A_{\mu\nu}^2) \\ &= -\frac{1}{2} g^{\alpha\beta} g_{\mu\nu,\alpha\beta} + \left( \nabla_{(\mu} \Gamma_{\nu)} + \frac{1}{2} (2B_{(\mu\nu)} - D_{\mu\nu}) + \frac{1}{4} (4C_{(\mu\nu)}^1 - 2C_{(\mu\nu)}^2 - 2E_{(\mu\nu)}^1 + E_{\mu\nu}^2) \right) \\ &\quad - \frac{1}{4} (4C_{(\mu\nu)}^1 - 2E_{\mu\nu}^1 - D_{\mu\nu} - 2C_{(\mu\nu)}^2 + E_{\mu\nu}^2 - 2A_{\mu\nu}^1 + 2A_{\mu\nu}^2) \\ &= -\frac{1}{2} g^{\alpha\beta} g_{\mu\nu,\alpha\beta} + \nabla_{(\mu} \Gamma_{\nu)} - \frac{1}{4} (2A_{\mu\nu}^1 + 2A_{\mu\nu}^2 - 4B_{(\mu\nu)} + D_{\mu\nu}) + A_{\mu\nu}^1 \\ &= -\frac{1}{2} g^{\alpha\beta} g_{\mu\nu,\alpha\beta} + \nabla_{(\mu} \Gamma_{\nu)} + g^{\alpha\beta} g^{\delta\epsilon} (g_{\mu\alpha,\delta} g_{\nu\beta,\epsilon} - \Gamma_{\mu\alpha\delta} \Gamma_{\nu\beta\epsilon}). \end{aligned} \quad (\text{2.39})$$

## C Derivation of the explicit Gauss-Legendre-Lobatto mass matrix

We denote the Legendre polynomials as  $P_i$  and their analytical and numerical inner product with respect to Gauss-Legendre-Lobatto nodes and weights as

$$\int_{-1}^1 P_i P_j dx = \frac{2}{2i+1} \delta_{ij} := h_i \delta_{ij}, \quad \sum_{k=1}^{N+1} P_i(x_k) P_j(x_k) \omega_k =: \langle P_i, P_j \rangle =: \gamma_i \delta_{ij}. \quad (\text{C.26})$$

Because numerical integration is exact up to order  $2N-1$ ,  $h_i = \gamma_i$  for  $i < N$ , whereas  $\gamma_N = \frac{2}{N} \neq h_N$ . Next, we decompose the Lagrange polynomials  $\ell_i$  along the Legendre polynomial basis  $\ell_i = a_i^k P_k$  and get for the unknown coefficients:

$$P_j(x_i) \omega_i = \langle \ell_i, P_j \rangle = a_j^k \gamma_j \quad \Rightarrow \quad a_i^j = \frac{\omega_i}{\gamma_j} P_j(x_i) \quad (\text{C.27})$$

$$\Rightarrow \quad \ell_i(x) = \omega_i \sum_{k=0}^N \frac{P_k(x_i) P_k(x)}{\gamma_k}. \quad (\text{C.28})$$

We calculate the mass matrix  $M_{ij}$  as

$$M_{ij} = \int_{-1}^1 \ell_i \ell_j dx = \omega_i \omega_j \sum_{k,l=0}^N \frac{P_k(x_i) P_l(x_j) h_k \delta_{kl}}{\gamma_k \gamma_l} \quad (\text{C.29})$$

$$= \omega_i \omega_j \sum_{k=0}^N \frac{P_k(x_i) P_k(x_j)}{\gamma_k} + \omega_i \omega_j \left( \frac{h_N}{\gamma_N^2} - \frac{1}{\gamma_N} \right) P_N(x_i) P_N(x_j) \quad (\text{C.30})$$

$$= \omega_j \ell_i(x_j) + \omega_i \omega_j \underbrace{\left( \frac{h_N}{\gamma_N^2} - \frac{1}{\gamma_N} \right)}_{=: \alpha} P_N(x_i) P_N(x_j) \quad (\text{C.31})$$

$$= \omega_j \delta_{ij} + \alpha \omega_i P_N(x_i) \omega_j P_N(x_j) \quad (\text{C.32})$$

It is straightforward to verify, that for a matrix of the form  $M = A + u \otimes v^T$ , the corresponding inverse is given by

$$(M^{-1}) = A^{-1} - \frac{(A^{-1}u) \otimes (A^{-1}v)^T}{1 + v^T A u}. \quad (\text{C.33})$$

For the mass matrix,  $A$  is diagonal and therefore trivial to invert. For the final result, as used in the code we gain:

$$(M^{-1})^{ij} = \frac{\delta_{ij}}{\omega_j} - \alpha \frac{P_N(x_i) P_N(x_j)}{1 + \alpha \gamma_N} = \frac{\delta_{ij}}{\omega_j} + \frac{N+1}{2} P_N(x_i) P_N(x_j). \quad (\text{3.58})$$



# Bibliography

- [LIGO] LIGO - Laser Interferometer Gravitational Wave Observatory, <http://www.ligo.caltech.edu/>.
- [VIRGO] Virgo/EGO - European Gravitational Observatory, <http://www.ego-gw.it/>.
- [GEO] GEO 600, <http://www.geo600.org/>.
- [KAGRA] KAGRA - Kamioka Gravitational Wave Detector, <http://gwcenter.icrr.u-tokyo.ac.jp/en/>.
- [IND] INDIGO - Indian Initiative in Gravitational-wave Observations, <http://www.gw-indigo.org>.
- [ET] Einstein Telescope - <http://www.et-gw.eu/>.
- [LAL] LSC Algorithm Library - <https://wiki.ligo.org/DASWG/LALSuite>.
- [SpEC] SpEC - Spectral Einstein Code, <http://www.black-holes.org/SpEC.html>.
- [BLAS] BLAS: Basic Linear Algebra Subroutines: <http://www.netlib.org/blas/>.
- [LISA] LISA (laser interferometer space antenna), proposal for a gravitational wave detector in space. Preprint, Max-Planck-Institut für Quantenoptik, MPQ 177, May 1993.
- [Abbott et al., 2013] Abbott, B. P. et al. (2013). Prospects for Observing and Localizing Gravitational-Wave Transients with Advanced LIGO and Advanced Virgo. [Living Rev. Rel.19,1(2016)].
- [Abbott et al., 2016a] Abbott, B. P. et al. (2016a). GW151226: Observation of Gravitational Waves from a 22-Solar-Mass Binary Black Hole Coalescence. *Phys. Rev. Lett.*, 116(24):241103.
- [Abbott et al., 2016b] Abbott, B. P. et al. (2016b). Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.*, 116(6):061102.
- [Abbott et al., 2017] Abbott, B. P. et al. (2017). Gw170104: Observation of a 50-solar-mass binary black hole coalescence at redshift 0.2. *Phys. Rev. Lett.*, 118:221101.
- [Ajith et al., 2011] Ajith, P., Hannam, M., Husa, S., Chen, Y., Brüggmann, B., Dorband, N., Müller, D., Ohme, F., Pollney, D., Reisswig, C., Santamaría, L., and Seiler, J. (2011). Inspirational-merger-ringdown waveforms for black-hole binaries with non-precessing spins. *Phys. Rev. Lett.*, 106:241101.
- [Alcubierre, 2008] Alcubierre, M. (2008). *Introduction to 3+1 Numerical Relativity*. Oxford University Press, Oxford.
- [Alcubierre et al., 2001] Alcubierre, M., Brandt, S. R., Brüggmann, B., Holz, D., Seidel, E., Takahashi, R., and Thornburg, J. (2001). Symmetry without symmetry: Numerical simulation of axisymmetric systems using Cartesian grids. *Int. J. Mod. Phys. D*, 10(3):273–289.

- [Arnowitt et al., 1962] Arnowitt, R., Deser, S., and Misner, C. W. (1962). The dynamics of general relativity. In Witten, L., editor, *Gravitation: An Introduction to Current Research*, pages 227–265. Wiley, New York.
- [Baiotti et al., 2005] Baiotti, L., Hawke, I., Montero, P. J., Löffler, F., Rezzolla, L., Stergioulas, N., Font, J. A., and Seidel, E. (2005). Three-dimensional relativistic simulations of rotating neutron star collapse to a Kerr black hole. *Phys. Rev. D*, 71:024035.
- [Baiotti and Rezzolla, 2016] Baiotti, L. and Rezzolla, L. (2016). Binary neutron-star mergers: a review of Einstein’s richest laboratory.
- [Baltensperger and Trummer, 2003] Baltensperger, R. and Trummer, M. R. (2003). Spectral differencing with a twist. *J. Sci. Comp.*, 24(5):1465–1487.
- [Banyuls et al., 1997] Banyuls, F., Font, J. A., Ibáñez, J. M., Martí, J. M., and Miralles, J. A. (1997). Numerical 3+1 general-relativistic hydrodynamics: A local characteristic approach. *Astrophys. J.*, 476:221.
- [Barth and Jespersen, 1989] Barth, T. and Jespersen, D. (1989). The design and application of upwind schemes on unstructured meshes. In *27th Aerospace sciences meeting*, page 366.
- [Baumgarte and Shapiro, 1998] Baumgarte, T. W. and Shapiro, S. L. (1998). On the Numerical integration of Einstein’s field equations. *Phys. Rev. D*, 59:024007.
- [Baumgarte and Shapiro, 2010] Baumgarte, T. W. and Shapiro, S. L. (2010). *Numerical Relativity: Solving Einstein’s Equations on the Computer*. Cambridge University Press, Cambridge.
- [Bernuzzi and Hilditch, 2010] Bernuzzi, S. and Hilditch, D. (2010). Constraint violation in free evolution schemes: comparing BSSNOK with a conformal decomposition of Z4. *Phys. Rev. D*, 81:084003.
- [Bishop and Rezzolla, 2016] Bishop, N. T. and Rezzolla, L. (2016). Extraction of gravitational waves in numerical relativity. *Living Reviews in Relativity*, 19(1):2.
- [Bjørhus, 1995] Bjørhus, M. (1995). The ODE Formulation of Hyperbolic PDEs Discretized by the Spectral Collocation Method. *SIAM J. Sci. Comput.*, 16(3):542–557.
- [Blanchet, 2006] Blanchet, L. (2006). Gravitational Radiation from Post-Newtonian Sources and Inspiralling Compact Binaries. *Living Rev. Relativity*, 2006-4. [Article in Online Journal] cited on 2 Jun 2006, <http://www.livingreviews.org/Articles/lrr-2006-4/download/index.html>.
- [Bonnement, A. et al., 2011] Bonnement, A., Fajraoui, T., Guillard, H., Martin, M., Mouton, A., Nkonga, B., and Sangam, A. (2011). Finite volume method in curvilinear coordinates for hyperbolic conservation laws. *ESAIM: Proc.*, 32:163–176.
- [Borges et al., 2008] Borges, R., Carmona, M., Costa, B., and Don, W. S. (2008). An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *J. Comp. Phys.*, 227(6):3191–3211.
- [Boyd, 1989] Boyd, J. P. (1989). *Chebyshev & Fourier Spectral Methods*, volume 49 of *Lecture Notes in Engineering*. Springer-Verlag, Berlin.
- [Boyle and Mroué, 2009] Boyle, M. and Mroué, A. H. (2009). Extrapolating gravitational-wave data from numerical simulations. *Phys. Rev. D*, 80:124045.
- [Bruhat, 1952] Bruhat, Y. (1952). Theoreme d’existence pour certains systemes d’equations aux derivees partielles non lineaires. *Acta Mathematica*, 88:141–225.

- [Buonanno and Damour, 1999] Buonanno, A. and Damour, T. (1999). Effective one-body approach to general relativistic two-body dynamics. *Phys. Rev. D*, 59:084006.
- [Butterworth and Ipser, 1976] Butterworth, E. M. and Ipser, J. R. (1976). On the structure and stability of rapidly rotating fluid bodies in general relativity. I - The numerical method for computing structure and its application to uniformly rotating homogeneous bodies. *ApJ*, 204:200–223.
- [Canuto et al., 2006] Canuto, C., Hussani, M. Y., Quarteroni, A., and Zang, T. A. (2006). *Spectral Methods*. Springer-Verlag, Berlin Heidelberg.
- [Carroll, 2003] Carroll, S. M. (2003). *Spacetime and Geometry: An Introduction to General Relativity*. Benjamin Cummings.
- [Choi, 2015] Choi, J. J. (2015). Hybrid spectral difference/embedded finite volume method for conservation laws. *Journal of Computational Physics*, 295:285 – 306.
- [Cockburn et al., 1990] Cockburn, B., Hou, S., and Shu, C.-W. (1990). Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws iv: The multidimensional case. *Math. Comp.*, 54:545–581.
- [Cockburn and Shu, 1989] Cockburn, B. and Shu, C.-W. (1989). Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws ii: General framework. *Math. Comp.*, 52:411–435.
- [Cockburn and Shu, 1998] Cockburn, B. and Shu, C.-W. (1998). Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws v: Multidimensional systems. *J. Comp. Phys.*, 141:199–224.
- [Cooley and Tukey, 1965] Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19:297–301.
- [Cordero-Carrión et al., 2009] Cordero-Carrión, I., Cerdá-Durán, P., Dimmelmeier, H., Jaramillo, J. L., Novak, J., and Gourgoulhon, E. (2009). Improved constrained scheme for the einstein equations: An approach to the uniqueness issue. *Phys. Rev. D*, 79(2):024017.
- [Cowling, 1941] Cowling, T. G. (1941). The non-radial oscillations of polytropic stars. *Mon. Not. R. Astr. Soc.*, 101:367–+.
- [Damour, 2001] Damour, T. (2001). Coalescence of two spinning black holes: An effective one-body approach. *Phys. Rev. D*, 64:124013.
- [Dessart et al., 2009] Dessart, L., Ott, C., Burrows, A., Rosswog, S., and Livne, E. (2009). Neutrino signatures and the neutrino-driven wind in Binary Neutron Star Mergers. *Astrophys.J.*, 690:1681.
- [Duez et al., 2008] Duez, M. D., Foucart, F., Kidder, L. E., Pfeiffer, H. P., Scheel, M. A., and Teukolsky, S. A. (2008). Evolving black hole-neutron star binaries in general relativity using pseudospectral and finite difference methods. *Phys. Rev.*, D78:104015.
- [Dumbser et al., 2013] Dumbser, M., Zanotti, O., Hidalgo, A., and Balsara, D. (2013). ADER-WENO Finite Volume Schemes with Space-Time Adaptive Mesh Refinement. *J. Comput. Phys.*, 248:257–286.
- [Dumbser et al., 2014] Dumbser, M., Zanotti, O., Loubere, R., and Diot, S. (2014). A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *J. Comp. Phys.*, 278:47–75.
- [Eichler et al., 1989] Eichler, D., Livio, M., Piran, T., and Schramm, D. N. (1989). Nucleosynthesis, neutrino bursts and gamma-rays from coalescing neutron stars. *Nature*, 340:126.

- [Einstein, 1915a] Einstein, A. (1915a). Die feldgleichungen der gravitation. *Preuss. Akad. Wiss. Berlin, Sitzungsber.*, pages 844–847.
- [Einstein, 1915b] Einstein, A. (1915b). Zur allgemeinen relativitätstheorie. *Preuss. Akad. Wiss. Berlin, Sitzungsber.*, pages 778–786.
- [Field et al., 2009] Field, S. E., Hesthaven, J. S., and Lau, S. R. (2009). Discontinuous galerkin method for computing gravitational waveforms from extreme mass ratio binaries. *Class.Quant.Grav.*, 26:165010.
- [Font et al., 1994] Font, J., Ibáñez, J., Martí, J., and Marquina, A. (1994). Multidimensional relativistic hydrodynamics: Characteristic fields and modern high-resolution shock-capturing schemes. *Astron. Astrophys.*, 282:304.
- [Font et al., 2002] Font, J. A., Goodale, T., Iyer, S., Miller, M., Rezzolla, L., Seidel, E., Stergioulas, N., Suen, W.-M., and Tobias, M. (2002). Three-dimensional general relativistic hydrodynamics. II. Long-term dynamics of single relativistic stars. *Phys. Rev. D*, 65:084024.
- [Font et al., 2000] Font, J. A., Miller, M., Suen, W. M., and Tobias, M. (2000). Three-dimensional numerical general relativistic hydrodynamics: Formulations, methods, and code tests. *Phys. Rev. D*, 61:044011.
- [Fourier, 1822] Fourier, J.-B.-J. (1822). *Théorie Analytique de la Chaleur*. Firmin Didot.
- [Freiburghaus et al., 1999] Freiburghaus, C., Rosswog, S., and Thielemann, F.-K. (1999). R-process in neutron star mergers. *Astrophys. J. Lett.*, 525:L121.
- [Gassner and Kopriva, 2011] Gassner, G. and Kopriva, D. A. (2011). A comparison of the dispersion and dissipation errors of gauss and gauss–lobatto discontinuous galerkin spectral element methods. *SIAM Journal of Scientific Computing*, 33:2560–2579.
- [Gourgoulhon, 2012] Gourgoulhon, E. (2012). *3+1 Formalism in General Relativity*. Springer, Berlin.
- [Gundlach et al., 2005] Gundlach, C., Martin-Garcia, J. M., Calabrese, G., and Hinder, I. (2005). Constraint damping in the Z4 formulation and harmonic gauge. *Class. Quantum Grav.*, 22:3767–3774.
- [Harten et al., 1983] Harten, A., Lax, P. D., and van Leer, B. (1983). *SIAM Rev.*, 25:35.
- [Hartle, 1967] Hartle, J. B. (1967). Slowly rotating relativistic stars. 1. Equations of structure. *Astrophys. J.*, 150:1005–1029.
- [Hartle and Sharp, 1967] Hartle, J. B. and Sharp, D. H. (1967). Variational Principle for the Equilibrium of a Relativistic, Rotating Star. *ApJ*, 147:317.
- [Hesthaven et al., 2007] Hesthaven, J. S., Gottlieb, S., and Gottlieb, D. (2007). *Spectral Methods for Time-Dependent Problems*. Cambridge University Press, Cambridge.
- [Hesthaven and Warburton, 2008] Hesthaven, J. S. and Warburton, T. (2008). *Nodal Discontinuous Galerkin Methods*. Springer, New York.
- [Hilditch, 2015] Hilditch, D. (2015). Dual Foliation Formulations of General Relativity.
- [Hilditch and Ruiz, 2016] Hilditch, D. and Ruiz, M. (2016). The initial boundary value problem for free-evolution formulations of General Relativity.
- [Hilditch et al., 2017] Hilditch, D., Weyhausen, A., and Bruegmann, B. (2017). Evolutions of centered Brill waves with a pseudospectral method.

- [Hilditch et al., 2016] Hilditch, D., Weyhausen, A., and Brüggmann, B. (2016). Pseudospectral method for gravitational wave collapse. *Phys. Rev.*, D93(6):063006.
- [Hulse and Taylor, 1975] Hulse, R. A. and Taylor, J. H. (1975). Discovery of a Pulsar in a Binary System. *ApJ*, 195:L51–55.
- [Jiang and Shu, 1996] Jiang, G.-S. and Shu, C.-W. (1996). Efficient implementation of weighted ENO schemes. *J. Comp. Phys.*, 126(1):202–228.
- [Karniadakis and Sherwin, 2005] Karniadakis, G. and Sherwin, S. (2005). *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, Oxford.
- [Kidder et al., 2016] Kidder, L. E. et al. (2016). SpECTRE: A Task-based Discontinuous Galerkin Code for Relativistic Astrophysics.
- [Komatsu et al., 1989] Komatsu, H., Eriguchi, Y., and Hachisu, I. (1989). Rapidly rotating general relativistic stars. I – Numerical method and its application to uniformly rotating polytropes. *Mon. Not. R. Astron. Soc.*, 237:355–379.
- [Kopriva, 1996] Kopriva, D. A. (1996). A conservative staggered-grid chebyshev multidomain method for compressible flows. ii. a semi-structured method. *Journal of Computational Physics*, 128(2):475 – 488.
- [Kopriva, 2006] Kopriva, D. A. (2006). Metric identities and the discontinuous spectral element method on curvilinear meshes. *J. Sci. Comp.*, 26(3):301–327.
- [Kopriva, 2009] Kopriva, D. A. (2009). *Implementing Spectral Methods for Partial Differential Equations*. Springer.
- [Lehner et al., 2005] Lehner, L., Reula, O., and Tiglio, M. (2005). Multi-block simulations in general relativity: high order discretizations, numerical stability and applications. gr-qc/0507004.
- [Lichnerowicz, 1944] Lichnerowicz, A. (1944). L'intégration des équations de la gravitation relativiste et le problème des  $n$  corps. *J. Math. Pures et Appl.*, 23:37–63.
- [Lindblom et al., 2006] Lindblom, L., Scheel, M. A., Kidder, L. E., Owen, R., and Rinne, O. (2006). A new generalized harmonic evolution system. *Class. Quant. Grav.*, 23:S447–S462.
- [Lindblom and Szilágyi, 2009] Lindblom, L. and Szilágyi, B. (2009). An Improved Gauge Driver for the GH Einstein System. *Phys. Rev.*, D80:084019.
- [Liu and Osher, 1994] Liu, X. and Osher, S. (1994). Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115:200–212.
- [Liu et al., 2008] Liu, Y. T., Shapiro, S. L., Etienne, Z. B., and Taniguchi, K. (2008). General relativistic simulations of magnetized binary neutron star mergers. *Phys. Rev. D*, 78:024012.
- [Martí et al., 1991] Martí, J. M., Ibáñez, J. M., and Miralles, J. M. (1991). Numerical relativistic hydrodynamics: Local characteristic approach. *Phys. Rev. D*, 43:3794.
- [Martí and Müller, 1994] Martí, J. M. and Müller, E. (1994). The analytical solution of the riemann problem in relativistic hydrodynamics. *J. Fluid Mech.*, 258:317–333.
- [Martí and Müller, 1996] Martí, J. M. and Müller, E. (1996). Extension of the piecewise parabolic method to one-dimensional relativistic hydrodynamics. *J. Comput. Phys.*, 123:1–14.
- [Michel, 1972] Michel, F. C. (1972). Accretion of matter by condensed objects. *Astrophys. Spa. Sci.*, 15:153.

- [MPI] MPI. The message passing interface (mpi) standard. <http://www.ncs.anl.gov/mpi/>.
- [Narayan et al., 1992] Narayan, R., Paczynski, B., and Piran, T. (1992). Gamma-ray bursts as the death throes of massive binary stars. *Astrophys. J.*, 395:L83.
- [Newman and Penrose, 1962] Newman, E. T. and Penrose, R. (1962). An approach to gravitational radiation by a method of spin coefficients. *J. Math. Phys.*, 3(3):566–578. erratum in *J. Math. Phys.* 4, 998 (1963).
- [Oppenheimer and Volkoff, 1939] Oppenheimer, J. R. and Volkoff, G. M. (1939). On Massive Neutron Cores. *Phys. Rev.*, 55:374–381.
- [Papadopoulos and Font, 1998] Papadopoulos, P. and Font, J. A. (1998). Relativistic hydrodynamics around black holes and horizon adapted coordinate systems. *Phys. Rev. D*, 58:024005.
- [Pollney et al., 2009] Pollney, D., Reisswig, C., Dorband, N., Schnetter, E., and Diener, P. (2009). Asymptotic falloff of local waveform measurements in numerical relativity. *Phys. Rev. D*, 80:121502.
- [Pollney et al., 2011] Pollney, D., Reisswig, C., Schnetter, E., Dorband, N., and Diener, P. (2011). High accuracy binary black hole simulations with an extended wave zone. *Phys. Rev. D*, 83:044045.
- [Press et al., 1986] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1986). *Numerical Recipes*. Cambridge University Press, Cambridge, England.
- [Pretorius, 2005] Pretorius, F. (2005). Numerical relativity using a generalized harmonic decomposition. *Class. Quant. Grav.*, 22:425–451.
- [Pretorius, 2006] Pretorius, F. (2006). Simulation of binary black hole spacetimes with a harmonic evolution scheme. *Class. Quantum Grav.*, 23:S529–S552.
- [Qiu and Shu, 2005] Qiu, J. and Shu, C.-W. (2005). Runge–Kutta discontinuous Galerkin method using WENO limiters. *J. Sci. Comp.*, 26(3):907–929.
- [Radice and Rezzolla, 2011] Radice, D. and Rezzolla, L. (2011). Discontinuous Galerkin methods for general-relativistic hydrodynamics: formulation and application to spherically symmetric spacetimes. *Phys.Rev.*, D84:024010.
- [Radice and Rezzolla, 2012] Radice, D. and Rezzolla, L. (2012). THC: a new high-order finite-difference high-resolution shock-capturing code for special-relativistic hydrodynamics.
- [Read et al., 2009] Read, J. S., Markakis, C., Shibata, M., Uryū, K., Creighton, J. D. E., and Friedman, J. L. (2009). Measuring the neutron star equation of state with gravitational wave observations. *Phys. Rev. D*, 79:124033.
- [Reed and Hill, 1973] Reed, W. and Hill, T. (1973). Triangular mesh methods for neutron transport equation. *Tech. report LA-UR-73-479*.
- [Rezzolla and Zanotti, 2013] Rezzolla, L. and Zanotti, O. (2013). *Relativistic hydrodynamics*. Oxford University Press.
- [Rinne et al., 2007] Rinne, O., Lindblom, L., and Scheel, M. A. (2007). Testing outer boundary treatments for the Einstein equations. *Class. Quant. Grav.*, 24:4053–4078.
- [Roe, 1981] Roe, P. L. (1981). Approximate riemann solvers, parameter vectors and difference schemes. *J. Comput. Phy.*, 43:357.

- [Ronchi et al., 1996] Ronchi, C., Iacono, R., and Paolucci, P. (1996). The “cubed sphere”: A new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.*, 124(1):93 – 114.
- [Shi et al., 2002] Shi, J., Hu, C., and Shu, C.-W. (2002). A technique of treating negative weights in WENO schemes. *J. Comp. Phys.*, 175(1):108–127.
- [Shibata and Nakamura, 1995] Shibata, M. and Nakamura, T. (1995). Evolution of three-dimensional gravitational waves: Harmonic slicing case. *Phys. Rev. D*, 52:5428–5444.
- [Shibata and Uryū, 2000] Shibata, M. and Uryū, K. (2000). Simulation of merging binary neutron stars in full general relativity: Gamma=2 case. *Phys. Rev. D*, 61:064001. gr-qc/9911058.
- [Shu, 1997] Shu, C. W. (1997). Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. Lecture notes ICASE Report 97-65; NASA CR-97-206253, NASA Langley Research Center. <http://techreports.larc.nasa.gov/icase/1997/icase-1997-65.pdf>.
- [Shu and Osher, 1989] Shu, C. W. and Osher, S. J. (1989). Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes, II. *J. Comput. Phys.*, 83:32.
- [Sperhake et al., 2007] Sperhake, U., Brüggmann, B., González, J. A., Hannam, M. D., and Husa, S. (2007). Head-On collisions of different initial data. In Kleinert, H., Jantzen, R. T., and Ruffini, R., editors, *Proceedings of the Eleventh Marcel Grossmann Meeting on General Relativity*, page 1612, Singapore. World Scientific, Singapore.
- [Stergioulas, 2003] Stergioulas, N. (2003). Rotating stars in relativity. *Living Rev. Relativity*, 6:3.
- [Stergioulas and Friedman, 1995] Stergioulas, N. and Friedman, J. L. (1995). Comparing models of rapidly rotating relativistic stars constructed by two numerical methods. *Astrophys. J.*, 444:306.
- [Taracchini et al., 2014] Taracchini, A., Buonanno, A., Khanna, G., and Hughes, S. A. (2014). Small mass plunging into a Kerr black hole: Anatomy of the inspiral-merger-ringdown waveforms.
- [Taylor et al., 2010] Taylor, N. W., Kidder, L. E., and Teukolsky, S. A. (2010). Spectral methods for the wave equation in second-order form. *Phys.Rev.*, D82:024037.
- [Teukolsky, 2015a] Teukolsky, S. A. (2015a). Formulation of discontinuous galerkin methods for relativistic astrophysics. *J. Comput. Phys.*, 312:333–356.
- [Teukolsky, 2015b] Teukolsky, S. A. (2015b). Short note on the mass matrix for Gauss–Lobatto grid points. *J. Comp. Phys.*, 283:408 – 413.
- [Thierfelder et al., 2011] Thierfelder, M., Bernuzzi, S., and Brüggmann, B. (2011). Numerical relativity simulations of binary neutron stars. *Phys. Rev. D*, 84:044012.
- [Tichy, 2006] Tichy, W. (2006). Black hole evolution with the BSSN system by pseudo-spectral methods. *Phys. Rev. D*, 74:084005.
- [Tolman, 1939] Tolman, R. C. (1939). Static Solutions of Einstein’s Field Equations for Spheres of Fluid. *Phys. Rev.*, 55:364–373.
- [Wald, 1984] Wald, R. M. (1984). *General relativity*. The University of Chicago Press, Chicago.

- [Wang, 2002] Wang, Z. J. (2002). Spectral (finite) volume method for conservation laws on unstructured grids. basic formulation. *Journal of Computational Physics*, 178(1):210–251.
- [Waxman, 2004] Waxman, E. (2004). Extra-galactic sources of high-energy neutrinos. *New J.Phys.*, 6:140.
- [Weyhausen, 2014] Weyhausen, A. (2014). *Numerical Methods for Collapsing Gravitational Waves*. PhD thesis, Friedrich-Schiller-Universität Jena.
- [Wheeler, 1964] Wheeler, J. (1964). Geometrodynamics and the issue of the final state. In DeWitt, C. and DeWitt, B., editors, *Relativity, Groups, and Topology (Les Houches, France, 1964)*, pages 316–520. Gordon and Breach, New York.
- [Winicour, 1998] Winicour, J. (1998). Characteristic evolution and matching. *Living Rev. Relativity*, 1:5. [Online article].
- [Yamamoto et al., 2008] Yamamoto, T., Shibata, M., and Taniguchi, K. (2008). Simulating coalescing compact binaries by a new code SACRA. *Phys. Rev.*, D78:064054.
- [York, 1999] York, J. W. (1999). Conformal ‘thin-sandwich’ data for the initial-value problem of general relativity. *Phys. Rev. Lett.*, 82:1350–1353.
- [York, 1979] York, Jr., J. W. (1979). Kinematics and dynamics of general relativity. In Smarr, L., editor, *Sources of Gravitational Radiation*, pages 83–126, Cambridge. Cambridge University Press.
- [Zanotti and Dumbser, 2014] Zanotti, O. and Dumbser, M. (2014). A high order special relativistic hydrodynamic and magnetohydrodynamic code with space–time adaptive mesh refinement. *Comput.Phys.Commun.*, 188:110–127.
- [Zhao and Tang, 2013] Zhao, J. and Tang, H. (2013). Runge–Kutta discontinuous Galerkin methods with WENO limiter for the special relativistic hydrodynamics. *J. Comp. Phys.*, 242:138.
- [Zhong and Shu, 2013] Zhong, X. and Shu, C.-W. (2013). A simple weighted essentially nonoscillatory limiter for Runge–Kutta discontinuous Galerkin methods. *J. Comp. Phys.*, 232(1):397–415.
- [Zumbusch, 2009] Zumbusch, G. (2009). Finite Element, Discontinuous Galerkin, and Finite Difference evolution schemes in spacetime. *Class. Quant. Grav.*, 26:175011.



# Publications

## *Regular Articles*

- [Bug1] M. Bugner, T. Dietrich, S. Bernuzzi, A. Weyhausen, B. Brügmann (2015), *Solving 3D relativistic hydrodynamical problems with weighted essentially nonoscillatory discontinuous Galerkin methods*, Phys.Rev.D.94 (2016) 084004, arXiv:1508.07147 [gr-qc].

## *Preprints*

- [Bug2] D. Hilditch, E. Harms, M. Bugner, H. Rüter, B. Brügmann (2016), *The evolution of hyperboloidal data with the dual foliation formalism: Mathematical analysis and wave equation tests*, arXiv:1609.08949 [gr-qc].
- [Bug3] H. Rüter, D. Hilditch, M. Bugner, B. Brügmann (2017), *Hyperbolic Relaxation Method for Elliptic Equations*, arXiv:1708.07358 [gr-qc].

## *In Preparation*

- [Bug3] A. Schoepe, D. Hilditch, M. Bugner (2017), *Revisiting hyperbolicity of relativistic fluids*



# Acknowledgement

These final words of the thesis mark the end of four unforgettable Ph.D. years in Jena. I am grateful to so many people who accompanied and supported me and made this period a time, that I will always remember.

I want to thank my supervisor Prof. Bernd Brügmann, who gave me the opportunity to study the beautiful theory of general relativity and to be part of both the fantastic NR group and the TPI community in Jena.

In particular, I want to thank David Hilditch for his endless support during the last four years and for explaining me GR, its mathematical concepts and physical implications. As a postdoc, teacher and very good friend he is irreplaceable and this thesis would never have been possible without him.

I am tremendously thankful for meeting the best office mates and colleagues one can imagine. I thank Toni Dietrich, better known as the “Rodelkönig”, for countless shared running kilometers, for being my roommate during conferences and for watching “Machete” with me. I thank Martin Fruhnert for being host of so many beautiful “Fest der Volksmusik” public viewing events. I thank Enno Harms for letting me win at least some of our FIFA chess tournaments and for so many spectacular table tennis rallies. I thank Niclas Moldenhauer for valuable discussions on domino related pendulums and for not destroying all table tennis balls on the planet, yet. I thank Matthias Pilz for selling me his keyboard, for inviting me to the “Saalehorizontale-Staffellauf” and for together saying: “We know nothing about WENO”. I thank Hannes Rüter for his positive vibes and for making music together. Arigatō! I thank Andreas Weyhausen for being a fantastic supervisor during my Praktikum in Jena and for trustingly handing over the bamps code to me. That was a mistake. I thank Reetika Dudi and Vivek Chaurasia for introducing me to the Indian culture and food and for supporting me during the last and hardest stage of the Ph.D. Special thanks for proof reading and eliminating so many typos and, comma, related, mistakes, goes to David, Toni, Hannes and Niclas.

I also want to thank Andre Sternbeck for his help with IT related problems and Sabine Irmer and Jens Schmidt for an outstanding administration of the “ARA” cluster.

I am grateful to Professors Ammon, Fritzsche, Gies, Lotze and Meinel, for a friendly and focused collaboration in planning and evaluating exercises and the “Physics Day”.

I am deeply thankful to my family. I thank my parents and grandparents who raised, guided and supported me through all the years of school, studies and the Ph.D. You taught me curiosity and helped me find my way. I thank my brother Tom for teaching me the high art of table tennis and Skat. Finally, my endless gratitude and love goes to my girlfriend Constanze for accompanying me through all ups and downs in the past 3 years. When I was struggling with work, physics and myself, you were there, encouraged me and helped me to go on. I am grateful for all the shared moments in the past and even more for those, that are yet to come.

# Additional Material

## Talks and poster presentations

- 15.03. - 20.03.2015 - (Berlin) DPG Frühjahrstagung 2015:  
*Talk “Discontinuous Galerkin methods in general relativistic hydrodynamics”*
- 08.12. - 09.12.2015 - (Tübingen) Grant Application “Physics & Astrophysics of Neutron Stars”:  
*Poster “Neutron Star Binaries: Gravitational Dynamics”*
- 11.02. - 12.02.2016 - (Jülich) NIC Symposium 2016:  
*Poster “Gravitational Dynamics of Neutron Star Binaries in Numerical Relativity”*
- 29.02. - 04.03.2016 - (Hamburg) DPG Frühjahrstagung 2016:  
*Talk “Discontinuous Galerkin methods in dynamical neutron star simulations”*
- 28.09.2016 - (Jena) Michael Stifel Center General Assembly 2016:  
*Poster “Gravitational Dynamics of Neutron Star Binaries in Numerical Relativity”*
- 14.03. - 17.03.2017 - (Bremen) DPG Frühjahrstagung 2017:  
*Talk “Discontinuous Galerkin methods in general relativistic neutron star simulations”*

## Other conferences and workshops

- 24.02. - 27.02.2014 - (Dresden) Parallel Programming Workshop Spring 2014
- 17.03. - 21.03.2014 - (Berlin) DPG Frühjahrstagung 2014
- 29.08. - 02.09.2016 - (Paris) Gravitational-Wave Astronomy Meeting
- 08.09. - 10.09.2016 - (Oppurg) Workshop on Numerical and Mathematical Relativity

# Curriculum Vitae

## Personal details

Name Marcus  
Surname Bugner  
Date of Birth 13th of September 1989  
Place of Birth Köthen (Anhalt)

## Education

2008 High school diploma (1.2) Ludwigsgymnasium Köthen  
2013 Diploma in Physics (1.1) Dresden University of Technology  
since 2013 PhD student at the Friedrich-Schiller-University Jena

## Professional Experience

2009-2010 Teaching assistant at Dresden University of Technology  
08/2011 Internship Numerical Relativity Group with Prof. Brüggemann  
2012-2013 Research assistant at MPI for the Physics of Complex Systems in  
Dresden  
since 2013 Research assistant at the Friedrich-Schiller-University Jena

## Prizes

2008 DPG High school graduate prize  
2017 Public's Choice Award at the 3 Minute Thesis Competition Jena

---

Signature

## Teaching

Semester	Course	Lecturer
WS 2013/2014	Elektrodynamik	Prof. S. Fritzsche
SS 2014	Quantenmechanik I	Prof. B. Brüggemann
WS 2014/2015	Theoretische Mechanik (Lehramt)	Prof. K. H. Lotze
SS 2015	Computational Physics II	Dr. D. Hilditch
WS 2015/2016	Computational Physics III	Prof. B. Brüggemann
SS 2016	Quantenmechanik I	Prof. R. Meinel
WS 2016/2017	Quantenmechanik I	Prof. H. Gies

# Zusammenfassung

Diese Arbeit dokumentiert die Implementierung und Überprüfung einer neuen numerischen Methodik zur allgemeinrelativistischen Simulation von Neutronensternen.

Hierbei wurden die GHG-Gleichungen mit Hilfe einer pseudospektralen Penalty-Methode diskretisiert, um die Metrik-Variablen zu evolvieren. Zur Behandlung der hydrodynamischen Variablen wurde sowohl eine diskontinuierliche Galerkin-Methode, als auch ein Finite-Volumen-Verfahren auf krummlinigen Gittern implementiert. Zusätzlich wurden dem Code Techniken zur Schockerkennung und -behandlung hinzugefügt.

Diese Bestandteile wurden in umfangreichen Tests validiert. Hierbei wurden numerische Fehler in der Massenerhaltung, in der Einhaltung von physikalischen Zwangsbedingungen und im Vergleich mit analytischen Lösungen betrachtet. Die zugehörigen Konvergenzordnungen aller Methoden wurden untersucht. Dabei wurde eine Konvergenzordnung  $\sim 2$  bei aktiver Schockbehandlung und mögliche Konvergenzordnungen  $> 2$  für die reine diskontinuierliche Galerkin-Methode festgestellt. Weiterhin wurde bestätigt, dass die numerische Einhaltung der metrischen Identitäten essentiell für die Massenerhaltung in Fluidynamik-Simulationen ist.

Mit Hilfe der reinen diskontinuierlichen Galerkin-Methode konnten physikalische Systeme ohne Schocks, wie etwa die Michel-Bondi-Akkretion, erfolgreich simuliert werden. Bei Neutronensternsimulationen hingegen wurden numerische Instabilitäten an der Sternoberfläche und ein Verlust der Konvergenz beobachtet. Erst eine aktive Schockbehandlung erlaubte die stabile und numerisch konvergente Simulation von statischen, rotierenden oder oszillierenden Neutronensternen.

Dank der vorgestellten Modifikationen des `bamps` Codes ist dieser nun in der Lage, die frontale Kollision zweier Neutronensterne zu simulieren. Neben der Extraktion des zugehörigen Gravitationswellensignals wurde die Simulation genutzt, um das nahezu ideale Skalierungsverhalten der Rechenzeit von `bamps` aufzuzeigen.

Abschließend wurde der neuartige Duale-Blätterung-Formalismus angewandt, um die Wellengleichung auf einem numerischen Gitter zu lösen, welches die lichtartige Zukunftsunendlichkeit beinhaltet. Wir zeigen erste erfolgreiche Tests dieses Verfahrens, welches in Zukunft zur genaueren Extraktion von Gravitationswellen aus numerischen Simulationen genutzt werden könnte.

# Selbstständigkeitserklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel und Literatur angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Bei der Auswahl und Auswertung dieser Arbeit haben mir die nachstehend aufgeführten Personen in beratender Weise unentgeltlich geholfen:

- Dr. Sebastiano Bernuzzi
- Prof. Bernd Brüggemann
- Dr. Tim Dietrich
- Dr. Enno Harms
- Dr. David Hilditch
- Dr. Niclas Moldenhauer
- M.Sc. Hannes Rüter

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder andere Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Die geltende Promotionsordnung der Physikalisch-Astronomischen Fakultät ist mir bekannt.

Ich versichere ehrenwörtlich, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Jena, 27.06.2017

---

Unterschrift