PHYSICAL REVIEW D **94,** 084004 (2016)

Solving 3D relativistic hydrodynamical problems with weighted essentially nonoscillatory discontinuous Galerkin methods

Marcus Bugner,[1] Tim Dietrich,[1,2] Sebastiano Bernuzzi,[3,4] Andreas Weyhausen,[1] and Bernd Brügmann[1]

[1]*Theoretical Physics Institute, University of Jena, 07743 Jena, Germany*
[2]*Max-Planck-Institute for Gravitational Physics, Albert-Einstein-Institute, D-14476 Golm, Germany*
[3]*Theoretical Astrophysics, California Institute of Technology, 1200 East California Boulevard, Pasadena, California 91125, USA*
[4]*DiFeST, University of Parma, and INFN Parma I-43124 Parma, Italy*

(Received 21 December 2015; published 3 October 2016)

Discontinuous Galerkin (DG) methods coupled to weighted essentially nonoscillatory (WENO) algorithms allow high order convergence for smooth problems and for the simulation of discontinuities and shocks. In this work, we investigate WENO-DG algorithms in the context of numerical general relativity, in particular for general relativistic hydrodynamics. We implement the standard WENO method at different orders, a compact (simple) WENO scheme, as well as an alternative subcell evolution algorithm. To evaluate the performance of the different numerical schemes, we study nonrelativistic, special relativistic, and general relativistic test beds. We present the first three-dimensional simulations of general relativistic hydrodynamics, albeit for a fixed spacetime background, within the framework of WENO-DG methods. The most important test bed is a single Tolman-Oppenheimer-Volkoff star in three dimensions, showing that long term stable simulations of single isolated neutron stars can be obtained with WENO-DG methods.

DOI: 10.1103/PhysRevD.94.084004

I. INTRODUCTION

Over the last decade simulations in numerical general relativity have seen a tremendous improvement in accuracy and stability and have become an important tool for the study of high energy and strong gravitational field effects. To date numerical simulations are the only possibility to investigate complex astrophysical scenarios such as e.g. stellar collapse [1] and coalescing binary neutron stars [2]. Although numerical simulations are, in principle, not restricted by approximations beyond the numerical approximation, they are limited by the finite accuracy of the particular discretization method.

Among the different methods to solve partial differential equations like those of general relativity, the discontinuous Galerkin (DG) method has emerged in recent years as a particularly successful general purpose paradigm [3–5]. It can be argued that the DG method, more explicitly the DG finite element method or DG spectral element method, subsumes and combines several of the key advantages of traditional finite element and finite volume methods, e.g. [4]. In particular, the *discontinuous* Galerkin method works with element-local stencils, which is a great advantage for parallelization and the construction of complicated grids. Furthermore, DG methods offer easy access to hp adaptivity [6], where both the size of the computational elements (or cells) and the order of the polynomial approximation within each element can be adapted to the problem. For smooth solutions, DG methods approach the optimal order of exponential convergence of pseudospectral methods on multiple patches. In fact, certain DG methods are equivalent to pseudospectral methods with a specific penalty method for the patch boundaries [7]. For nonsmooth solutions, low order elements have been combined with various high resolution shock capturing (HRSC) schemes, for example in the form of weighted essentially nonoscillatory (WENO)-DG methods [8].

In this work we consider the application of DG methods to simulations in numerical general relativity coupled to general relativistic hydrodynamics (GRHD). Concretely, the goal is to compute the numerical evolution of spacetimes containing neutron stars. The governing differential equations are the time-dependent, nonlinear Einstein field equations for the spacetime geometry coupled to a relativistic fluid model. Most relativistic hydrodynamics simulations are based on the “Valencia formulation,” in which the matter field evolution is given in a conservative form [9].

Among the numerous numerical studies carried out in the field, most have been performed using finite difference (FD) and finite volume (FV) methods, with significant success. For the geometry (including black holes), high order finite differencing is the rule, often fourth to eighth order finite differences in space for structured adaptive mesh refinement (AMR), e.g. [10,11]. The matter part allows the formation of strong relativistic shocks, and a variety of finite volume (or finite difference) HRSC schemes have been developed [9,12]. For smooth solutions, pseudospectral methods have been very successful [13–16].

Recently, a convergence order of ∼3 was observed for high order matter formulations in [17–19].

DG methods for numerical relativity offer the usual list of attractive features. In particular, one goal would be to combine high order, smooth regions with lower order regions containing shocks. Compared to AMR with large, overlapping finite difference stencils, the DG spectral element method is more easily and more efficiently parallelizable, while still allowing high order approximations. However, there remain several open issues with regard to DG methods in numerical relativity, some of which are described and addressed in what follows.

The evolution equations of numerical relativity are a coupled system for the geometry (the metric variables) and the matter variables. While the matter equations are naturally given in a flux form [9], this is not the case for the geometry. Since a typical DG method starts with a flux-balance law, it is in principle straightforward to design a method for the matter part. On the other hand, for the geometric part one should either recast the equations in a hyperbolic flux form, or suggest less standard methods.

There have been essentially only four major efforts to employ DG methods for general relativity and/or GRHD. In [20], Zumbusch gives the first and so far only example for a complete DG method for the $(3+1)$-dimensional (short 3D) Einstein equations in vacuum. Discussed is a spacetime DG scheme mostly in the context of linearized equations and in a specific gauge, but the scheme also handles nonlinearities. So far there has not been an astrophysics application, say involving black holes or neutron stars. In [21], Brown *et al.* discuss a DG method for a first order reformulation of the Baumgarte-Shapiro-Shibata-Nakamura (BSSN) formulation of the vacuum equations, mostly with one-dimensional examples. In earlier work [22] by Field *et al.* the various nonlinearities and second derivatives of BSSN are successfully dealt with on a case by case basis. In [23], Radice and Rezzolla give a general discussion of the DG method for the matter equations in the standard flux form, without including the equation for the geometry. They present a working one-dimensional implementation for general relativistic matter in spherical symmetry. Recently in [24], Teukolsky summarized DG methods for curved spacetimes, including a discussion of nonconservative equations applicable to the Einstein equations.

In addition, there has been work on special relativistic hydrodynamics (SRHD). Zhao and Tang [25] were the first to apply the WENO-DG method of [8] to a variety of one-dimensional and two-dimensional test cases in SRHD, and the method turned out to be robust and reliable in capturing shocks.

The concrete target of the present work is to model a single stationary neutron star [a Tolman-Oppenheimer-Volkoff (TOV) star [26,27]] in 3D, although we perform a variety of tests in one dimension and two dimensions as well. The TOV star is computed in the Cowling approximation, which simplifies the problem by assuming that the geometry may be curved but does not depend on time, which in turn is compatible with the stationarity of the TOV star. The numerical evolution of the matter variables for a fixed metric is a standard approach that still allows to test key features of the hydrodynamics, including the treatment of the nondifferentiable density at the surface of the star. We leave the coupling to a dynamic geometry to future work.

In preparation for the simulations in full, 3D GRHD, we test the Runge-Kutta DG (RKDG) method coupled to a variety of WENO reconstructions for the equations of general relativistic hydrodynamics [9]. We reproduce the nonrelativistic standard results [8,28] as well as some of the special relativistic test cases of [25] for a third and a fifth order method, WENO3 and WENO5. We extend [25] by also considering WENO-Z [29] and the simple WENO limiters of [28]. Finally, we present the first application of RKDG WENO methods to a 3D TOV star in the Cowling approximation. The numerical experiments are implemented in the new `bamps` code [14] for spectral element methods. We import some methods from an existing full-featured finite difference AMR code for 3D GRHD, BAM [10,30–33].

When researching the available HRSC methods for DG, there is one issue related to shock resolution and efficiency that is well known but that does not always appear to receive the attention it deserves.

For FD or FV methods with HRSC, shocks are resolved within a few cell widths, which means within a few grid points. For DG methods, the standard approach is to employ WENO reconstruction based on cell averages [8]. In such WENO-DG methods, shocks are again resolved within a few cell widths, but each cell now contains p points (for polynomials of order $p-1$). The WENO3 stencil involves three cells and $3p$ points, and the WENO5 stencil involves five cells and $5p$ points. Effectively, the high resolution within each cell (the "subcell resolution") is lost if only the cell averages are used. For practical implementations a rough estimate is therefore that such WENO-DG methods could require about 27 or 125 times more resources for shock resolution in 3D than comparable FD or FV methods (these factors vary with the actual implementation).

For the evaluation of DG methods for GRHD it matters whether such methods are competitive to existing FD/FV methods in terms of efficiency. Hence we consider the following measures aimed at handling the comparatively low efficiency of cell-averaged WENO-DG methods. A common strategy is to limit the application of the WENO scheme to only those cells that need it, and there has been quite some work on so-called "troubled cell indicators" [34].

A recent development is the so-called "simple" WENO methods of [28], which effectively construct a compact stencil for high order WENO methods. For example, the fifth order WENO method is constructed from only three instead of five cells, using the high order information from the nearest neighbor cells to obtain fifth order. This leads to significant savings, but the method has not been widely tested yet. We include the compact/simple WENO method in our tests and report on some differences to the standard WENO method, in particular in 3D.

Another important development is a hybrid approach [35–38], which replaces troubled cells by an equidistant subgrid and applies FV shock capturing on these grids. This approach maintains the subcell resolution of FV methods, but increases the complexity of the implementation since two types of grids and special grid transfer operators are required. In our case the method is appealing because a full-featured FD implementation is already available [30,31]. If successful, the strategy would be to construct a high order DG method for regions where the solution is smooth, but to rely on established FD methods near shocks.

HRSC for DG comes at a cost since WENO-DG as well as the hybrid FD-DG method break the cell locality of the basic DG method. We consider both methods here to gain some insight into their relative merit.

In Sec. II, we introduce the DG method for 3D flux-balance laws, specify the equations of relativistic hydrodynamics, and discuss the WENO-DG and FD-DG methods. In Sec. III B, we summarize the numerical implementation. As basic tests we consider the advection equation and the Burgers equation in one dimension in Sec. IV, while one-dimensional and two-dimensional tests for SRHD are presented in Sec. V. The main results concern the evolution of a TOV star in Sec. VI. We conclude in Sec. VII. For completeness, we collect some relevant details of the basic one-dimensional DG method in Appendix A and compare convergence for different mass matrices in Appendix B.

Throughout the article dimensionless units are used, i.e. we set $c = G = M_\odot = 1$. We denote spacetime indices by $a, b, \ldots$ and indices over space dimensions by $i, j, \ldots$.

II. METHODS

A. Discontinuous Galerkin method

The hydrodynamical equations governing the time evolution of the matter fields can be cast as a nonlinear conservation law for a vector of variables, $\mathbf{u}(x,t)$, depending on time t and position $x \in \mathbb{R}^3$. The conservation law is given by

$$\partial_t \mathbf{u} + \partial_i \mathbf{f}^i(\mathbf{u}) = \mathbf{S}, \tag{1}$$

with the sources $\mathbf{S}$ and the fluxes $\mathbf{f}^i$. We summarize some of the relevant aspects of the DG method for conservation laws of scalar function on $\mathbb{R}$ in Appendix A, while simply stating the key equations for vector-valued functions on $\mathbb{R}^n$ here; see [4,25].

We consider a partition of $\mathbb{R}^n$ into cells I_j, $x \in I_j$, and define the finite dimensional approximation space

$$V^N \coloneqq \{v : v(x)|_{I_j} \in \mathbb{P}^N(I_j)\} \tag{2}$$

with $\mathbb{P}^N(I_j)$ denoting the finite dimensional space of polynomials on I_j of degree at most N. As in most standard applications, we set the polynomial order N as a constant over the whole partition. To deduce a DG scheme from Eq. (1), we want to find a function $\mathbf{u_n}(x)$ for which the weak form

$$\int_{I_j} v \partial_t \mathbf{u_n}\, \mathrm{d}V + \int_{\partial I_j} \mathbf{f}^i(\mathbf{u_n}) v n_i\, \mathrm{d}S - \int_{I_j} \mathbf{f}^i(\mathbf{u_n}) \partial_i v\, \mathrm{d}V = \int_{I_j} \mathbf{S} v\, \mathrm{d}V \tag{3}$$

holds for all $v \in V^N$. For simplicity, we denote the approximate/numerical solution $\mathbf{u_n}(x)$ as $\mathbf{u}(x)$ in the following. An important advantage of the DG scheme is that v does not need to be continuous at the cell boundaries. Therefore, no unambiguous definition of the fluxes at cell boundaries entering Eq. (3) exists. To overcome this issue, we introduce the numerical fluxes $\mathbf{f}^{*i}(\mathbf{u}_-, \mathbf{u}_+)$, which depend on the inside/outside cell limited value of $\mathbf{u}$ at the boundary, $\mathbf{u}_-$ and $\mathbf{u}_+$, and reproduce the original flux if $\mathbf{u}$ is continuous. A simple example of a numerical flux with this property is the local Lax-Friedrich (LLF) flux

$$\mathbf{f}^{*i}(\mathbf{u}_-, \mathbf{u}_+) n_i = \frac{1}{2}[\mathbf{f}^i(\mathbf{u}_-) n_i + \mathbf{f}^i(\mathbf{u}_+) n_i - \lambda(\mathbf{u}_+ - \mathbf{u}_-)], \tag{4}$$

where λ denotes the maximum absolute eigenvalue of the Jacobian $\partial(\mathbf{f}^i n_i)/\partial\mathbf{u}$. We use a LLF algorithm throughout this article. Writing out the numerical solution $\mathbf{u}(x,t)$ as an element of V^N explicitly,

$$\mathbf{u}|_{I_j}(x,t) = \sum_{k=0}^{N} \hat{\mathbf{u}}_k(t) v^k(x) \tag{5}$$

and v^k a basis of $\mathbb{P}^N(I_j)$ allows us to recast (3) as an algebraic equation for the unknown time derivatives $\partial_t \hat{\mathbf{u}}_k$. To evolve these coefficients in time, we use an explicit fourth order Runge-Kutta method.

For the actual implementation of Eq. (3), we map each I_j to a reference box $[-1,1]^3$ and define $N+1$ Legendre-Gauss-Lobatto (LGL) points $\xi^p \in [-1,1]$ for each direction. Given these points, we choose the basis v^k of $\mathbb{P}^N([-1,1]^3)$ to be the product of the corresponding Lagrange interpolating polynomials each applied to one component of ξ,

$$v^k \equiv v^{pqr} = \ell^p \ell^q \ell^r \tag{6}$$

with

$$\ell^p(\xi) = \prod_{\substack{j=1\\ j\neq p}}^{N+1} \frac{\xi - \xi^j}{\xi^p - \xi^j}, \tag{7}$$

i.e. we use a nodal DG formulation. The chosen basis allows us to use $\ell^p(\xi^r) = \delta^{pr}$ and simplifies the computation of the coefficients $\hat{\mathbf{u}}_{pqr} = \mathbf{u}(\xi^p, \xi^q, \xi^r)$ (interpolation condition). In contrast to the modal DG formulation, the flux and source coefficients are then easily determined by pointwise evaluations $\hat{\mathbf{f}}^i_{pqr} = \mathbf{f}^i(\hat{\mathbf{u}}_{pqr})$, $\hat{\mathbf{S}}_{pqr} = \mathbf{S}(\hat{\mathbf{u}}_{pqr})$. Defining the mass matrix

$$M^{ab} = \int_{-1}^{1} \ell^a(\xi)\ell^b(\xi)d\xi \tag{8}$$

and the stiffness matrix

$$S^{ab} = \int_{-1}^{1} \partial_\xi \ell^a(\xi)\ell^b(\xi)d\xi, \tag{9}$$

we separate analytic expressions and numerical variables in Eq. (3) to gain the semidiscrete scheme,

$$\begin{aligned}\partial_t \hat{\mathbf{u}}_{pqr} = &+ \frac{2}{\Delta x}(M^{-1}_{pa} S^{ab} \hat{\mathbf{f}}^1_{bqr} - M^{-1}_{pN} \mathbf{f}^{*1} n_1(1, \xi^q, \xi^r) \\ &- M^{-1}_{p0} \mathbf{f}^{*1} n_1(-1, \xi^q, \xi^r)) \\ &+ \frac{2}{\Delta y}(M^{-1}_{qa} S^{ab} \hat{\mathbf{f}}^2_{pbr} - M^{-1}_{qN} \mathbf{f}^{*2} n_2(\xi^p, 1, \xi^r) \\ &- M^{-1}_{q0} \mathbf{f}^{*2} n_2(\xi^p, -1, \xi^r)) \\ &+ \frac{2}{\Delta z}(M^{-1}_{ra} S^{ab} \hat{\mathbf{f}}^3_{pqb} - M^{-1}_{rN} \mathbf{f}^{*3} n_3(\xi^p, \xi^q, 1) \\ &- M^{-1}_{r0} \mathbf{f}^{*3} n_3(\xi^p, \xi^q, -1)) + \hat{\mathbf{S}}_{pqr}.\end{aligned} \tag{10}$$

Due to the choice of collocation points, the mass and stiffness matrix can be determined using Legendre-Gauss-Lobatto integration with the corresponding weights ω^p,

$$M^{ab} \approx \delta^{ab} \cdot \omega^a, \tag{11}$$

$$S^{ab} = \partial_\xi \ell^a(\xi^b) \cdot \omega^b. \tag{12}$$

Notice that Eq. (11) is just an approximation, while (12) is exact, since the $N+1$-point Legendre-Gauss-Lobatto integration is exact for polynomials of order $2N-1$. This approximation simplifies the scheme and brings M in a diagonal form. Furthermore, it is equal to a modal filter, which decreases the highest mode by a factor $N/(2N+1)$ [39].

B. WENO reconstruction methods

As a next step, we explain how to avoid oscillations and unphysical behavior caused by the Gibbs phenomenon. For this purpose we locate discontinuities and oscillations with the troubled cell indicator described in Sec. II B 1 and apply a WENO limiter reconstruction [8,25]. We introduce three different WENO reconstruction methods, the standard WENO approach (Sec. II B 2), the simple WENO algorithm [28] based on compact stencils (Sec. II B 3), and a WENO algorithm based on a subcell evolution (Sec. II C).

1. Troubled cell indication

Given the coefficients of the numerical solution $\hat{\mathbf{u}}_p(t)$ at time t, we can calculate the average of the polynomial $\mathbf{u}(x,t)$ over the grid patches $I_j = [a_j, b_j]$,

$$\mathbf{u_j} \coloneqq \frac{1}{\Delta x}\int_{a_j}^{b_j} \hat{\mathbf{u}}_p v^p(x)dx = \frac{1}{2}\int_{-1}^{1} \hat{\mathbf{u}}_p \ell^p(\xi)d\xi, \tag{13}$$

where we use the transformed basis polynomial $v^p(x) = \ell^p(\xi(x))$. We further denote the boundary values of $\mathbf{u}$ as

$$\mathbf{u}^-_j \coloneqq \mathbf{u}(a_j), \qquad \mathbf{u}^+_j \coloneqq \mathbf{u}(b_j) \tag{14}$$

and define the four differences:

$$\tilde{\mathbf{u}}^-_\mathbf{j} \coloneqq \mathbf{u_j} - \mathbf{u}^-_j, \qquad \tilde{\mathbf{u}}^+_\mathbf{j} \coloneqq \mathbf{u}^+_j - \mathbf{u_j} \tag{15}$$

$$\Delta_- \mathbf{u} \coloneqq \mathbf{u_j} - \mathbf{u_{j-1}}, \qquad \Delta_+ \mathbf{u} \coloneqq \mathbf{u_{j+1}} - \mathbf{u_j}. \tag{16}$$

We also introduce the minmod function

$$\mathrm{minmod}(x_1, x_2, \ldots, x_n) = \begin{cases} s \cdot \min_{1\leq j \leq n|x_j|} & \text{if } \mathrm{sign}(x_1) = \ldots = \mathrm{sign}(x_n) =: s \\ 0 & \text{otherwise}\end{cases} \tag{17}$$

and the modified total variation bounded (TVB) minmod function

$$\mathrm{minmod_{TVB}}(x_1, x_2, \ldots, x_n) = \begin{cases} a_1 & \text{if } |a_1| \leq M(\max_j \Delta x_j)^2 \\ \mathrm{minmod}(x_1, x_2, \ldots, x_n) & \text{otherwise}\end{cases}. \tag{18}$$

Here the constant $M > 0$ acts as a threshold for the troubled cell indication. The lower the M, the more cells will be marked as troubled. The particular choice of M is problem dependent. Our troubled cell indicator marks a grid patch as troubled if

$$\mathrm{minmod}_{\mathrm{TVB}}((\tilde{u}_j^-)^k, \Delta_- u^k, \Delta_+ u^k) \neq (\tilde{u}_j^-)^k \quad \text{or}$$
$$\mathrm{minmod}_{\mathrm{TVB}}((\tilde{u}_j^+)^k, \Delta_- u^k, \Delta_+ u^k) \neq (\tilde{u}_j^+)^k \tag{19}$$

for at least one component k. This is exemplary for a situation in which a component of $\mathbf{u}$ is not monotonous (because the arguments of minmod differ in sign) or its gradient inside a patch is larger than that of the neighboring patches (shock inside the cell).

In the case of multiple dimensions, we perform the one-dimensional troubled cell indication in every direction. A cell is marked as troubled if at least one of these indications results in a troubled state. To apply the one-dimensional algorithm, the boundary values used in (14) have to be modified, since the cell boundaries are not longer single points, but lines or surfaces. Therefore, we redefine $\mathbf{u}_\mathbf{j}^\pm$ by the boundary averages, i.e. for a 3D cell $I = [a_j, b_j] \times [a_k, b_k] \times [a_l, b_l]$ in x direction,

$$\mathbf{u}_{\mathbf{jkl}}^- \coloneqq \frac{1}{\Delta y \Delta z} \int_{a_k}^{b_k} \int_{a_l}^{b_l} \hat{\mathbf{u}}_{pqr} v^p(a_j) v^q(y) v^r(z)\, \mathrm{d}y\mathrm{d}z,$$
$$\mathbf{u}_{\mathbf{jkl}}^+ \coloneqq \frac{1}{\Delta y \Delta z} \int_{a_k}^{b_k} \int_{a_l}^{b_l} \hat{\mathbf{u}}_{pqr} v^p(b_j) v^q(y) v^r(z)\, \mathrm{d}y\mathrm{d}z. \tag{20}$$

2. Standard WENO reconstruction

In a standard WENO method of order $2w+1$, we construct $w+1$ stencils S_i around I_j, each as an aggregation of $w+1$ grid patches: $S_i = (I_{j-w+i}, I_{j-w+i+1}, \ldots, I_{j+i})$, $0 \le i \le w$. In Fig. 1 this partitioning is shown for $w = 2$. For each stencil, we construct a wth order polynomial $\mathbf{p}_i$, which has the same average as the numerical solution $\mathbf{u}$ over each grid patch in the stencil. That means solving the system

$$\mathbf{u_k} = \frac{1}{\Delta x} \int_{I_k} \mathbf{p}_i(x) \mathrm{d}x, \qquad \text{for all } I_k \in S_i \tag{21}$$

for the $w+1$ coefficients of each component of $\mathbf{p}_i$. Similarly, we construct a $2w$th order polynomial $\mathbf{q}$ fulfilling

$$\mathbf{u_k} = \frac{1}{\Delta x} \int_{I_k} \mathbf{q}(x) \mathrm{d}x, \quad \text{for all } I_k \in S, \tag{22}$$

with $S \coloneqq \cup_i S_i$ being the large stencil over all $2w+1$ grid patches. The fundamental concept is to approximate the solution in $[-1, 1]$ as a linear combination of the $\mathbf{p}_i$, which should give the same result as the higher order approximation $\mathbf{q}$ in smooth regions. This condition defines the linear (or ideal) weights γ_i satisfying

$$\mathbf{q}(x) = \sum_{i=1}^{w+1} \gamma_i(x) \mathbf{p}_i(x). \tag{23}$$

We emphasize that the γ_i depend on the point x where the approximation should hold. It is remarkable that although both sides of Eq. (23) depend intrinsically on the $2w+1$ averages $\mathbf{u_k}$ and the system is overdetermined (only $w+1$ variables), we could always find an exact solution for (23) in our tests. In regions where the solution is not smooth, the weights should be chosen such that the smoothest polynomial of $\mathbf{p}_i$ is preferred. For this purpose, we use a smoothness indicator as suggested in [40],

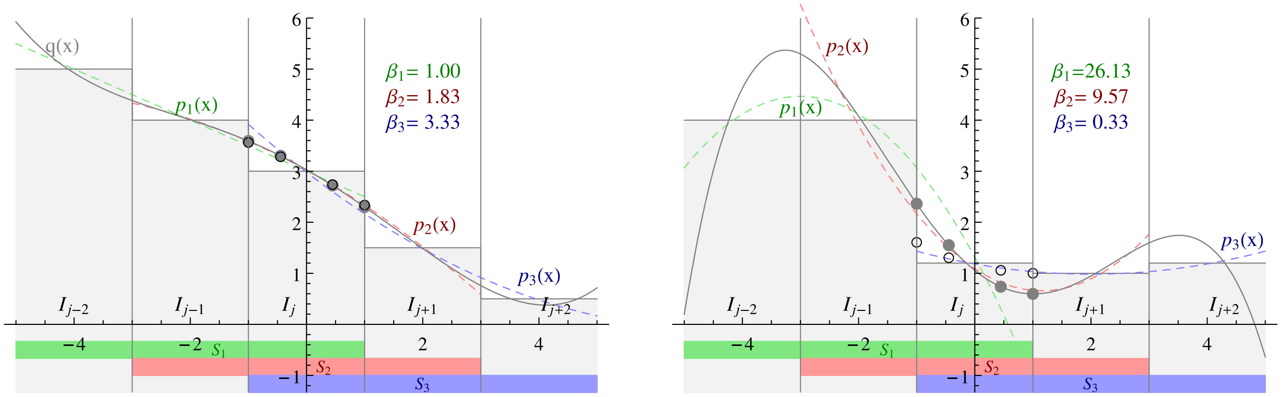

FIG. 1. The WENO-5 ($w = 2$) methodology applied in a smooth case (left figure) and a shock case (right figure). The values in the interval $x \in I_j = [-1, 1]$ are to be reconstructed from the five grid patch averages $u_{j-2}, u_{j-1}, u_j, u_{j+1}, u_{j+2}$. The three stencils S_1, S_2, S_3 are created as a clustering of three grid patches 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 II B, the smoothness indicators β_i are calculated for each stencil. A large β_i indicates nonsmoothness 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 fifth order polynomial q (filled gray circles).

$$\boldsymbol{\beta}_i = \sum_{l=1}^{w} \int_{I_j} \Delta x^{2l-1} \left(\frac{d^l}{dx^l} \mathbf{p}_i(x) \right)^2 \mathrm{d}x. \tag{24}$$

Because $\boldsymbol{\beta}_i$ is large for nonsmooth $\mathbf{p}_i$, the weights are chosen indirect proportional to $\boldsymbol{\beta}_i$. We use either the standard WENO choice

$$\tilde{\boldsymbol{\omega}}_i(x) = \frac{\gamma_i(x)}{(10^{-6} + \boldsymbol{\beta}_i)^2}, \tag{25}$$

or the improved WENO-Z version [29] for $w = 2$,

$$\tilde{\boldsymbol{\omega}}_i(x) = \gamma_i(x) \left(1 + \frac{|\boldsymbol{\beta}_3 - \boldsymbol{\beta}_1|}{\boldsymbol{\beta}_i + 10^{-40}} \right), \tag{26}$$

and normalize the results,

$$\boldsymbol{\omega}_i(x) = \frac{\tilde{\boldsymbol{\omega}}_i(x)}{\sum_{l=1}^{w+1} \tilde{\boldsymbol{\omega}}_l(x)}, \tag{27}$$

where $\boldsymbol{\omega}_i(x)$ are the final reconstruction weights. The reconstructed solution is then given by

$$\mathbf{u}^{\mathbf{WENO}}(x) = \sum_{i=1}^{w+1} \boldsymbol{\omega}_i(x) \mathbf{p}_i(x). \tag{28}$$

To generalize the presented reconstruction mechanism to two dimensions and 3D, we use the procedure described in [25]. For simplicity, we assume a rectilinear two-dimensional grid structure with $N+1$ grid points ξ^p per cell and direction. To reduce the full reconstruction of the cell I_{jk} to the one-dimensional case, we decouple the different directions. First we perform $2w+1$ one-dimensional WENO reconstructions in the x direction with input data

$$\{\mathbf{u}_{\mathbf{j-w},\tilde{\mathbf{k}}}, \mathbf{u}_{\mathbf{j-w+1},\tilde{\mathbf{k}}}, \ldots, \mathbf{u}_{\mathbf{j+w},\tilde{\mathbf{k}}}\}, \qquad k - w \leq \tilde{k} \leq k + w \tag{29}$$

to reconstruct the $N+1$ averages per cell,

$$\mathbf{u}^{\mathbf{p}}_{\mathbf{j},\tilde{\mathbf{k}}} \coloneqq \int_{I_{\tilde{j}\tilde{k}}} \mathbf{u}(\xi_p, y, t) dy, \qquad k - w \leq \tilde{k} \leq k + w, \quad 1 \leq p \leq m + 1. \tag{30}$$

Then, we can apply a second one-dimensional WENO reconstruction based on the one-dimensional averages in y direction with the input data

$$\{\mathbf{u}^{\mathbf{p}}_{\mathbf{j},\mathbf{k-w}}, \mathbf{u}^{\mathbf{p}}_{\mathbf{j},\mathbf{k-w+1}}, \ldots, \mathbf{u}^{\mathbf{p}}_{\mathbf{j},\mathbf{k+w}}\}, \qquad 1 \leq p \leq m + 1 \tag{31}$$

to get the two-dimensional reconstructed values inside the cell I_{jk},

$$\mathbf{u}^{\mathbf{WENO}}_{\mathbf{jk}}(\xi_p, \xi_q), \qquad 1 \leq p, q \leq m + 1. \tag{32}$$

3. *Simple WENO reconstruction*

To reconstruct the polynomial with a standard WENO method, the cell averages of many neighboring cells are needed. This leads to large computational costs and an undesirable smoothing of the solution. In [28], it is discussed that this standard procedure is not necessary in a DG method, since the neighboring cells yield more information than only a cell average value. This idea implies the simple WENO reconstruction, in which the standard WENO methodology is applied to the cell polynomial and the neighboring cell polynomials by redefining

$$\mathbf{p}_i(x) \coloneqq \mathbf{u}|_{I_{j+i}}(x) + \frac{1}{\Delta x} \int_{I_j} (\mathbf{u}|_{I_j}(x) - \mathbf{u}|_{I_{j+i}}(x)) dx, \quad i = -1, 0, 1. \tag{33}$$

The integral values cause a shift of the polynomials, so that they all have the same average value in cell I_j and the cell average is conserved during reconstruction. The corresponding expressions in (24) and (28) have to be substituted. Furthermore, we only use the next neighbors, which is setting $w = 1$ (three cell stencil) in all WENO formulas. Since with the new ansatz (33) every linear combination of the $\mathbf{p}_i(x)$ is a higher order approximation, there is no need to find special ideal weights as in the standard WENO method. Instead, we can freely choose the weights for all involved cells. In our tests, we choose $\gamma_{-1} = \gamma_1 = 1 \times 10^{-5}$, $\gamma_0 = 1 - 2\gamma_1 = 1 - 2 \times 10^{-5}$ for smooth setups and $\gamma_{-1} = \gamma_1 = 1 \times 10^{-3}$, $\gamma_0 = 1 - 2\gamma_1 = 1 - 2 \times 10^{-3}$ for problems with discontinuities.

C. Subcell evolution method

Finally, we consider a hybrid FD-DG method motivated by [35] where shock capturing is performed on a subgrid of equidistant grid points. The method of [35] is based on subgrids, an *a posteriori* troubled cell indicator, and a locally implicit time integrator. We decided to investigate the subgrid method separately without these other features, so we cannot compare directly to [35]. However, this allows us to isolate the effects due to the subcell algorithm and to establish its usefulness even in the absence of the other ingredients. There are open questions regarding the stability and accuracy of the subgrid method, in particular when used without the locally implicit time integration method (but see also [37,38]).

For our method we use the same troubled cell indicator as introduced in II B. If a cell I_j has been marked as troubled, we subdivide this cell in $2N+1$ equidistant subcells J_k containing a single point y_k each (where N is the polynomial order) and compute the value of the approximating polynomial on the individual subcells points,

$$\mathbf{v}_{\mathbf{k}} = \hat{\mathbf{u}}_p v^p(y_k), \qquad \text{for all } y_k \in I_j. \tag{34}$$

This map $\hat{\mathbf{u}}_p \mapsto \mathbf{v}_\mathbf{k}$ can be done with the subcell projection operator $\mathcal{P}$. The back projection $\mathcal{P}^{-1}$ is nontrivial, because the problem of finding a polynomial of order N to satisfy the given $2N+1$ equations (34) is overdetermined. Performing a least-squares fit of an Nth order polynomial for the $2N+1$ points turns out to be a good choice for a back projection. In our tests, we found the corresponding matrices for $\mathcal{P}$ and $\mathcal{P}^{-1}$ to be pseudoinverse. This is easy to verify, because whenever $\mathbf{v}_\mathbf{k}$ originate from an exact Nth order polynomial, a least-squares fit $\mathcal{P}^{-1}$ gives the exact polynomial coefficients, so $\mathcal{P}^{-1}\mathcal{P} = \mathbf{1}$ (but not necessarily $\mathcal{P}\mathcal{P}^{-1} = \mathbf{1}$). It is important to notice that contrary to [35], we use a projection matrix based on the point values in the subcells, not the averages $\mathbf{v}_\mathbf{k}$. This is necessary, since we want to employ a FD code on the subcells instead of a FV method, leading to a violation of conservation laws (e.g. of the rest mass), when a projection from top cell to subcells, or vice versa, is done. In our tests we found this defects decaying with order $N+1$, when we raise the grid resolution. We import all necessary routines of the BAM code [10,30,31]. In [30,41] this scheme is explained in detail. Further improvements allowing us to obtain high order convergence in smooth regions are presented in [41]. The general idea is to discretize Eq. (1) as

$$\partial_t \mathbf{u}_\mathbf{k} = \frac{2N+1}{\Delta x}\left(\mathbf{F}_{\mathbf{k}-\frac{1}{2}} - \mathbf{F}_{\mathbf{k}+\frac{1}{2}}\right) + \mathbf{S}_\mathbf{k} \qquad (35)$$

with Δx being the cell grid spacing, $\frac{\Delta x}{2N+1}$ the subcell grid spacing and $\mathbf{F}_{\mathbf{k}+\frac{1}{2}}$ the numerical flux at the boundary between subcells J_k and J_{k+1}.

The subcell interface values of the fluxes $\mathbf{f}_{\mathbf{k}\pm\frac{1}{2}}$ are computed with the LLF scheme. The necessary right and left states for the interface flux calculation are provided by a WENOZ [29,40] reconstruction from the given subcell values. Having evaluated the rhs of Eq. (35), we use an explicit fourth order Runge-Kutta method for the time step. After each Runge-Kutta substep the new subcell values are back projected to the DG grid by $\mathcal{P}^{-1}$. If the cell stays troubled in the next time step, the next evolution step is based on the subcell results without using the back-projected results.

III. IMPLEMENTATION

A. Relativistic hydrodynamics

Although we are working in cowling approximation, i.e. keeping the metric fixed, the matter fields are evolved dynamically on a curved spacetime background. We recast briefly the important equations and methods necessary to solve the general relativistic hydrodynamical equations; special relativity can be easily obtained by choosing flat spacetime.

1. 3 + 1 decomposition

Although we assume the spacetime to be fixed, we have to recast it in a suitable form for dynamical evolutions. This can be done with the help of a $3+1$ decomposition [42,43] (see [44–46] for textbook introductions) in which the four-dimensional spacetime metric is rewritten as

$$\mathrm{d}s^2 = -\alpha^2 \mathrm{d}t^2 + \gamma_{ij}(\mathrm{d}x^i + \beta^i \mathrm{d}t)(\mathrm{d}x^j + \beta^j \mathrm{d}t), \qquad (36)$$

where α is the lapse function, β^i the shift vector, and γ_{ij} the spatial metric. In the case of a flat spacetime $\alpha = 1, \beta^i = 0$, $\gamma_{ij} = \delta_{ij}$ employing Cartesian coordinates. Einstein's field equations split into two sets, the constraint equations and the evolution equations. For our single neutron stars tests, we recast the TOV equation [26,27] in $3+1$ form and solve it to obtain an ordinary differential equation. In addition to the $3+1$ split we perform a conformal transformation of the spatial metric,

$$\gamma_{ij} = \psi^4 \bar{\gamma}_{ij}, \qquad (37)$$

where ψ is the conformal factor and $\bar{\gamma}_{ij}$ the conformally related metric.

2. Hydrodynamic equations

According to Eq. (1) we denote the state vector collecting the conserved variables as $\mathbf{u}$, while $\mathbf{f}^i(\mathbf{u})$ are hydrodynamical fluxes, and $\mathbf{S}$ the source terms. The fluxes and the sources depend in general on the metric and matter fields. The conserved variables are $\mathbf{u} = \sqrt{\gamma}(D, S_k, \tau)$, and denote respectively the rest-mass density (D), the momentum density (S_k), and an internal energy (τ) measured by the Eulerian observer given by the particular spacetime foliation. $\gamma = \det \gamma_{ij}$ is the determinant of the spatial three-metric. The conserved variables $\mathbf{u}$ can be reconstructed from the primitive variables $\mathbf{w} = (\rho, v^i, \epsilon, p)$, i.e. rest-mass density, 3-velocity measured by the Eulerian observer, internal energy and pressure of the fluid, by the following equations:

$$D = W\rho, \qquad (38a)$$

$$S_k = W^2 \rho h v_k, \qquad (38b)$$

$$\tau = (W^2 \rho h - p) - D, \qquad (38c)$$

where W is the Lorentz factor, $W = 1/\sqrt{1 - v_i v^i}$ and h is the specific enthalpy $h = 1 + \epsilon + p/\rho$.

To close the system an equation of state (EOS) $p = P(\rho, \epsilon)$ is needed. In this work, we use a simple polytropic

$$P(\rho) = K\rho^\Gamma \qquad (39)$$

or an ideal gas EOS of the form

$$P(\rho, \epsilon) = (\Gamma - 1)\rho\epsilon, \tag{40}$$

where K is the polytropic constant and Γ is the adiabatic index. The particular implementation of the hydrodynamical equations follows [9,30].

However, due to the special choice of the background metric, the flux and source terms simplify dramatically by setting $\bar{\gamma}_{ij} = \delta_{ij}$ and $\beta^i = 0$ in all our examples.

3. Primitive recovery and atmosphere treatment

We evolve the conservative variables $\mathbf{u}$ by constructing the fluxes and source terms for every time slice. While $\mathbf{f}^i$ and $\mathbf{S}$ both contain the primitive variables $\mathbf{w}$ we have to recover those from the conservatives. The inverse relations of (38a)–(38c) are given by

$$\rho = \frac{D}{W}, \tag{41}$$

$$v^i = \frac{S^i}{\tau + D + p}, \tag{42}$$

$$\epsilon = \frac{\sqrt{(\tau + p + D)^2 - S^2} - Wp - D}{D}, \tag{43}$$

with $W = (\tau + p + D)/\sqrt{(\tau + p + D)^2 - S^2}$ and $S^2 = S_i S^i$. To make use of (41)–(43), we have to determine the pressure p.

The explicit primitive reconstruction goes as follows. First, we try to recover the primitive variables for the full equation of state including thermal components $p = P(\rho, \epsilon)$. For this reason a Newton-Raphson method is employed to compute the pressure p. If the method does not converge to the desired accuracy a cold equation of state $p = p(\rho)$ is used and we try to find with a Newton-Raphson method the density ρ.

As in most general relativistic hydrodynamic codes, we have to include an artificial atmosphere to solve the problem of fluid-vacuum interfaces. This allows long term stable and robust numerical simulations [47–49]. The atmosphere ρ_{atm} is computed according to

$$\rho_{\mathrm{atm}} = f_{\mathrm{atm}} \cdot \max[\rho(t = 0)]. \tag{44}$$

Whenever a point falls below the atmosphere threshold $\rho_{\mathrm{thr}} = f_{\mathrm{thr}} \cdot \rho_{\mathrm{atm}}$ during the evolution or the primitive reconstruction, it is set to the atmosphere value.

B. Numerical implementation

Throughout this article we employ the bamps code [14]. It is based on the method of lines with a pseudospectral decomposition in the spatial part and an explicit fourth order Runge-Kutta for the time stepping. It has been successfully used to study the gravitational wave collapse and it allows long term simulations of single black hole spacetimes with excision techniques. The program exhibits a hybrid p-thread/message passing interface (MPI) parallelization strategy and shows almost ideal scaling for up to several thousands of computing cores in vacuum simulations; see [14] for more details.

In this work we extend the bamps code by implementing (i) discontinuous Galerkin methods, (ii) a general relativistic hydrodynamics scheme for fixed background metrics, (iii) a simple high resolution shock capturing (HRSC) scheme as in [25], and (iv) a subcell-HRSC scheme [35]. This work is the first step towards a more general infrastructure for the simulation of compact binary systems where matter is present. Although bamps allows grid structures known as "cubed spheres" [50], we restrict ourselves to simple Cartesian boxes. However, a generalization could be achieved easily.

For the standard WENO implementation, we recast the crucial equations in matrix form, where all matrices can be precomputed from the geometry before evolution. During the actual simulation (i) the smoothness indicators are calculated from the cell averages as a quadratic form $\beta_i = Q_i^{kl} u_{i+k} u_{i+l}$; (ii) the weights are determined by (27); (iii) the value $p_i(\xi^q)$ of the approximating polynomial of stencil i at the collocation points ξ^q is evaluated from the cell averages by a matrix-vector multiplication $p_i(\xi^q) = C_i^{qr} u_{i+r}$ originating from (21); and (iv) the final reconstruction is calculated by (28). For simple WENO computations, the only difference is that in steps (i) and (iii) the matrices are larger, because the β_i and $p_i(\xi^q)$ not only depend on the averages of the neighbor cells, but the full polynomial given by $N + 1$ coefficients per cell.

In contrast to previous work, where no restriction algorithm (Sec. II B) was present, we need to communicate more then just the two-dimensional boundary layers of every cell. Therefore, we introduced a new grid distribution method to reduce the communication between different processors. The Cartesian grid consisting of $n = n_x n_y n_z$ boxes is distributed on p processes in such a way that communication between the processes is minimal. For this purpose, we perform a prime decomposition of $p = p_1 p_2 \ldots p_i$ and set the number of grids per direction $p_x = p_1, p_y = p_2, p_z = p_3$ initially. Let $p_{\min} = \min(p_x, p_y, p_z)$; we recalculate $p_{\min}$ as $p_{\min} \mapsto p_{\min} \cdot p_4$. For further p_j, we proceed in the same manner, so that each p_j is always multiplied with the smallest of p_x, p_y, p_z. Finally, we subdivide the full box grid in p_x parts in x direction, in p_y parts in y direction and in p_z parts in z direction. Each of these $p_x \cdot p_y \cdot p_z$ parts is mapped to one MPI process, which gives a simple box decomposition, which is almost cubical.

IV. SIMPLE TEST BEDS

A. Advection equation

As a first test for our new algorithms we consider the advection equation without a source ($S = 0$)

$$\partial_t u + \partial_x u = 0 \tag{45}$$

for a Gaussian peak on the interval $x \in [-1, 1]$,

$$u(x,0) = A\mathrm{e}^{(-x^2/\sigma^2)} + A\mathrm{e}^{(-(x-2)^2/\sigma^2)} + A\mathrm{e}^{(-(x+2)^2/\sigma^2)} \tag{46}$$

(we artificially add two peaks to gain smooth, periodic initial data) and a rectangular pulse (nonsmooth initial data)

$$u(x,0) = \begin{cases} 1 & \text{if } |x - x_0| < \sigma \\ 0 & \text{else} \end{cases}. \tag{47}$$

The convergence rate in the first test case ($A = 1$, $\sigma = 0.4$) is influenced by several effects; see Table I.

For our choice of polynomials with order N, we find convergence rates up to order $N + 1$, as expected. However, in an error regime beyond 10^{-10}, we observe a further drop in the convergence rates, because of the growing influence of truncation errors. Applying the standard WENO reconstruction procedure leads to slightly different results. Convergence for small numbers of n_x 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_x. The cell indicator only marks the cells around the maximum of the Gaussian peak as troubled, so the effective area, where the WENO reconstruction takes place, decreases. Since the reconstruction has a strong smoothing effect, the numerical results significantly differ from the analytic solution for small n_x and tend to the pure DG solution for large n_x. Comparing the two WENO implementations, we observe that the simple WENO algorithm shows slower convergence, but while the standard WENO is $\sim 1 - 2$ orders of magnitude less accurate than the pure DG evolution, the simple WENO performs much better, showing roughly the same L_1-errors as the pure DG evolution. Although we have not examined this in detail, it is plausible that the simple WENO reconstruction is closer to pure DG than standard WENO since simple WENO uses the full DG solution of the immediate neighbors.

For the second case Eq. (47), which we just 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 + standard WENO method. However, the differences are at most a factor of 2. The simple WENO algorithm has comparable errors as the DG + standard WENO method. Independent of the scheme we observe first order convergence, which is consistent with the expectation for a nonsmooth problem containing discontinuities.

In these and some of the following tests the DG method gives accurate results even when not combined with a reconstruction method, and it is clearly also the most efficient method. However, more demanding situations involving shocks typically require methods to handle the shocks explicitly.

TABLE I. Numerical errors and convergence orders for the advection equation problem (46) at $t = 10$ for different numbers of grid patches n_x and orders of DG polynomials N (CFL = 0.25, $M = 1$).

		DG		DG + WENO-7		DG + simple WENO	
n_x	N	L_1 error	Order	L_1 error	Order	L_1 error	Order
10	1	1.53×10^{-1}		2.92×10^{-1}		1.51×10^{-1}	
20		3.63×10^{-2}	2.08	1.40×10^{-1}	1.05	4.79×10^{-2}	1.65
40		5.34×10^{-3}	2.76	3.77×10^{-2}	1.90	6.64×10^{-3}	2.85
80		7.29×10^{-4}	2.87	7.15×10^{-3}	2.39	7.29×10^{-4}	3.18
160		1.24×10^{-4}	2.54	1.28×10^{-3}	2.48	1.24×10^{-4}	2.54
320		2.84×10^{-5}	2.12	2.36×10^{-4}	2.43	2.84×10^{-5}	2.12
10	3	2.04×10^{-4}		4.84×10^{-2}		2.12×10^{-4}	
20		1.02×10^{-5}	4.32	1.51×10^{-3}	4.99	1.02×10^{-5}	4.37
40		6.27×10^{-7}	4.03	4.99×10^{-5}	4.92	6.36×10^{-7}	4.01
80		3.90×10^{-8}	4.00	9.71×10^{-7}	5.68	3.98×10^{-8}	3.99
160		2.44×10^{-9}	4.00	1.60×10^{-8}	5.92	2.53×10^{-9}	3.97
320		1.52×10^{-10}	4.00	3.03×10^{-10}	5.72	1.62×10^{-10}	3.96
10	5	7.99×10^{-7}		9.95×10^{-2}		5.75×10^{-6}	
20		1.88×10^{-8}	5.40	1.11×10^{-2}	3.15	1.27×10^{-7}	5.49
40		8.92×10^{-10}	4.39	4.09×10^{-4}	4.76	5.26×10^{-9}	4.59
80		5.42×10^{-11}	4.04	8.67×10^{-6}	5.56	1.28×10^{-10}	5.35
160		3.40×10^{-12}	3.99	1.40×10^{-7}	5.95	1.52×10^{-11}	3.07
320		6.48×10^{-13}	2.39	7.66×10^{-10}	7.51	1.94×10^{-10}	

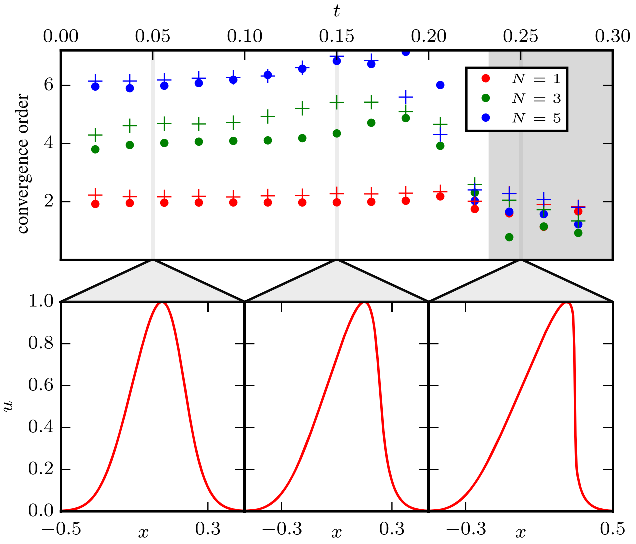

FIG. 2. Convergence rate during the evolution of a Gaussian wave packet for the Burgers equation (48): As expected, the convergence rate is around $N+1$ during the evolution of a smooth wave. At $t_{\mathrm{shock}} \approx 0.233$, a shock forms and the rates significantly drop down to first order convergence. When a standard WENO-7 reconstruction is used (crosses) the convergence rates are slightly higher than for the pure DG scheme (dots). The convergence rate is calculated from the errors of a $n_x = 160$ and a $n_x = 320$ run.

B. Burgers equation

The Burgers equation without source ($S = 0$),

$$\partial_t u + u \partial_x u = 0, \tag{48}$$

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

$$t_{\mathrm{shock}} = -\left(\min \frac{\partial u_0}{\partial x}\right)^{-1} \tag{49}$$

shocks appear during the evolution. We use this as a test bed for our code and evolve the initial Gaussian peak (46) with $A = 1$ and $\sigma = 0.2$. For this initial conditions, a shock forms at $t_{\mathrm{shock}} \approx 0.23316$. Similarly to our results for the advection equation we observe that the convergence rate is decreasing after t_{shock}; see Fig. 2. The lower panels show snapshots of the field u at times $t = 0.05$, 0.15, 0.25. Without WENO reconstruction (circles) we observe the expected convergence order of $N+1$ up to t_{shock}. Shortly before the shock formation at t_{shock} convergence starts to drop for all N (gray shaded region). Employing a standard WENO algorithm convergence is slightly above the expected $N+1$th order. As discussed for the advection equation, this is related to the amount of troubled cells, which are reconstructed. For higher resolution a smaller percentage of cells is reconstructed and consequently a faster convergence is observed. After the shock formation the convergence order drops also for DG + WENO to approximately first order convergence.

In addition, we prepared the initial conditions

$$u(x, 0) = 0.5 + \sin(x\pi) \tag{50}$$

and check convergence at $t = 0.5/\pi$ to compare with the results of [8]; Table II summarizes the results. Because of

TABLE II. Numerical errors and convergence orders for the Burgers equation problem (50) at $t = \frac{0.5}{\pi}$ for different numbers of grid patches n_x and orders of DG polynomials N.

		DG		DG + WENO-7		DG + simple WENO	
n_x	N	L_1 error	Order	L_1 error	Order	L_1 error	Order
10	1	5.34×10^{-2}		6.09×10^{-2}		8.23×10^{-2}	
20		1.45×10^{-2}	1.87	1.80×10^{-2}	1.75	1.61×10^{-2}	2.34
40		4.29×10^{-3}	1.76	4.66×10^{-3}	1.94	4.80×10^{-3}	1.75
80		1.24×10^{-3}	1.78	1.29×10^{-3}	1.84	1.24×10^{-3}	1.94
160		3.60×10^{-4}	1.78	3.69×10^{-4}	1.80	3.60×10^{-4}	1.78
320		1.02×10^{-4}	1.82	1.03×10^{-4}	1.83	1.02×10^{-4}	1.82
10	3	1.80×10^{-3}		3.94×10^{-3}		1.80×10^{-3}	
20		9.80×10^{-5}	4.20	1.50×10^{-4}	4.71	9.80×10^{-5}	4.20
40		6.36×10^{-6}	3.94	6.72×10^{-6}	4.48	6.36×10^{-6}	3.94
80		4.21×10^{-7}	3.91	4.22×10^{-7}	3.99	4.21×10^{-7}	3.91
160		2.71×10^{-8}	3.95	2.71×10^{-8}	3.95	2.71×10^{-8}	3.95
320		1.75×10^{-9}	3.95	1.75×10^{-9}	3.95	1.75×10^{-9}	3.95
10	5	3.58×10^{-5}		5.86×10^{-3}		3.58×10^{-5}	
20		7.61×10^{-7}	5.55	1.49×10^{-4}	5.29	7.60×10^{-7}	5.55
40		1.61×10^{-8}	5.56	1.33×10^{-6}	6.81	1.62×10^{-8}	5.54
80		2.97×10^{-10}	5.75	1.16×10^{-8}	6.83	2.98×10^{-10}	5.77
160		5.47×10^{-12}	5.76	1.63×10^{-10}	6.14	5.47×10^{-12}	5.76
320		1.15×10^{-13}	5.56	9.49×10^{-13}	7.43	1.15×10^{-13}	5.57

the smoothness of the solution, we observe for $N = 1$ polynomials second order convergence independent of the reconstruction method applied in the troubled cells. While the total L_1 error for DG + WENO-5 is approximately a factor of 2–3 larger than the pure DG evolution, we see that the DG + simple WENO algorithm performs as well as pure DG. For $N = 3$ polynomials, we expect fourth order convergence, which we can verify with the pure DG and the DG + simple WENO setup. The DG + WENO-5 algorithm shows a higher convergence rate for low resolutions, which is again caused by the fact that a larger number of cells decreases the interval where a reconstruction is performed.

V. SPECIAL RELATIVISTIC HYDRODYNAMICS

In the following section, we solve the GRHD conservation law (1) [9,30] without source terms and with $\alpha = \psi^4 = 1$ to consider special relativistic test cases, i.e. flat spacetimes. Regarding the EOS, we set $\Gamma = \frac{5}{3}$ and $K = 1$.

A. One-dimensional problems

As a first test, we consider a smooth sine wave propagating with constant speed. The initial conditions are

$$\begin{aligned} \rho(x,t) &= 1 + 0.2\sin(2\pi(x - v_x t)), \\ v_x(x,t) &= 0.2, \\ p(x,t) &= 1, \end{aligned} \tag{51}$$

inside the periodic one-dimensional domain $x \in [-1, 1]$ divided into n_x uniform grid patches. Viewing the L_1 errors and convergence rates (Table III), we find the convergence rate of the DG scheme to be $N + 1$, when we use polynomials $p \in \mathbb{P}^N([-1, 1])$. In Appendix B we compare numerical results for analytic mass matrix with mass lumping for $N = 1$.

Although we are dealing with a smooth problem a few cells around the maximum of the density ρ are marked as troubled. When we employ the standard WENO-5 or WENO-Z reconstruction method, we observe at least one order of magnitude larger absolute errors as in the pure DG case for the employed resolutions. Contrary to this, the convergence order is artificially higher than for the pure DG method. For the simple WENO method, we obtain absolute errors compatible or identical with the scheme without reconstruction and obtain a convergence order of $N + 1$ for an Nth order polynomial. In the case of the subcell evolution, i.e. when we project the grid patch data on a finer subcell treating this with the finite differencing method, we observe similar convergence rates. The subcell evolution itself is performed with a fifth order accurate scheme [51], which we verify with simulations using only subcells.

As a second test focusing on the ability of our scheme to deal with discontinuities, we consider the shock tube problem with initial conditions

$$(\rho, v_x, p)(x, 0) = \begin{cases} (10, 0, 13.33) & \text{if } x < 0.5 \\ (1, 0, 10^{-7}) & \text{if } x \geq 0.5 \end{cases} \tag{52}$$

TABLE III. Numerical errors and convergence orders for problem (51) at $t = 2$ for different numbers of grid patches n_x, orders of DG polynomials N and several shock resolution methods.

		DG		DG + WENO-5		DG + WENO-Z		DG + simple WENO		DG + subcells		Subcells only	
n_x	N	L_1 error	Order	L_1 error	Order	L_1 error	Order	L_1 error	Order	L_1 error	Order	L_1 error	Order
10	1	1.22×10^{-3}		8.46×10^{-2}		8.62×10^{-2}		1.80×10^{-2}		2.85×10^{-3}		3.22×10^{-6}	
20		2.73×10^{-4}	2.15	2.80×10^{-2}	1.59	2.69×10^{-2}	1.67	1.86×10^{-3}	3.27	1.95×10^{-3}	0.54	1.00×10^{-7}	5.00
40		6.72×10^{-5}	2.02	4.83×10^{-3}	2.53	4.77×10^{-3}	2.49	7.03×10^{-5}	4.72	4.07×10^{-4}	2.26	3.14×10^{-9}	4.99
80		1.67×10^{-5}	2.00	6.61×10^{-4}	2.86	6.43×10^{-4}	2.89	1.67×10^{-5}	2.06	8.86×10^{-5}	2.20	9.84×10^{-11}	4.99
160		4.18×10^{-6}	2.00	9.64×10^{-5}	2.77	8.73×10^{-5}	2.88	4.18×10^{-6}	2.00	2.02×10^{-5}	2.13	3.08×10^{-12}	4.99
320		1.04×10^{-6}	2.00	1.40×10^{-5}	2.77	1.44×10^{-5}	2.59	1.04×10^{-6}	2.00	4.31×10^{-6}	2.22	1.14×10^{-13}	4.75
10	3	4.27×10^{-6}		3.69×10^{-3}		9.67×10^{-4}		4.33×10^{-6}		1.58×10^{-5}		1.08×10^{-7}	
20		3.29×10^{-7}	3.70	4.52×10^{-5}	6.35	1.83×10^{-5}	5.72	3.21×10^{-7}	3.75	9.33×10^{-7}	4.08	3.39×10^{-9}	4.99
40		1.79×10^{-8}	4.20	7.37×10^{-7}	5.93	2.10×10^{-7}	6.44	1.76×10^{-8}	4.18	4.49×10^{-8}	4.37	1.06×10^{-10}	4.99
80		9.39×10^{-10}	4.25	1.09×10^{-8}	6.07	3.39×10^{-9}	5.95	9.50×10^{-10}	4.21	3.56×10^{-9}	3.65	3.31×10^{-12}	5.00
160		6.01×10^{-11}	3.96	1.58×10^{-10}	6.10	1.31×10^{-10}	4.69	6.06×10^{-11}	3.96	2.08×10^{-10}	4.09	1.07×10^{-13}	4.94
320		3.80×10^{-12}	3.97	6.96×10^{-12}	4.51	6.93×10^{-12}	4.24	3.84×10^{-12}	3.97	1.26×10^{-11}	4.04	2.06×10^{-14}	2.37
10	5	2.63×10^{-9}		8.53×10^{-3}		1.48×10^{-3}		3.79×10^{-8}		5.09×10^{-8}		1.78×10^{-8}	
20		3.86×10^{-11}	6.08	2.67×10^{-4}	4.99	2.13×10^{-5}	6.11	5.55×10^{-10}	6.09	1.36×10^{-9}	5.22	5.57×10^{-10}	4.99
40		6.13×10^{-13}	5.97	4.80×10^{-6}	5.79	1.89×10^{-7}	6.81	6.97×10^{-12}	6.31	1.04×10^{-11}	7.02	1.74×10^{-11}	4.99
80		4.64×10^{-14}	3.72	6.47×10^{-8}	6.21	1.54×10^{-9}	6.93	1.95×10^{-13}	5.15	3.93×10^{-13}	4.73	5.52×10^{-13}	4.97
160		8.63×10^{-14}		2.98×10^{-10}	7.76	1.30×10^{-11}	6.89	9.70×10^{-14}	1.00	1.20×10^{-13}	1.70	4.41×10^{-14}	3.64
320		1.80×10^{-13}		8.81×10^{-13}	8.40	6.47×10^{-13}	4.32	7.69×10^{-13}		2.48×10^{-13}		4.94×10^{-14}	

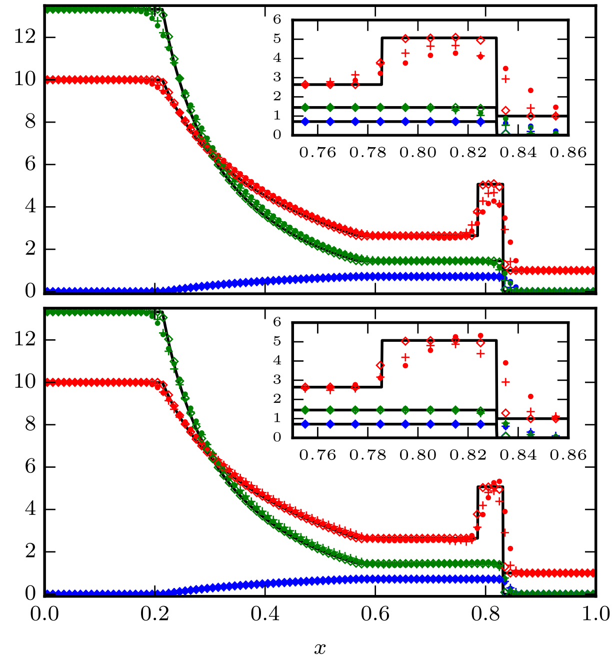

FIG. 3. Evolution of the special relativistic shock tube initial data (52) (density, red; velocity, blue; pressure, green): Numerical results using the standard WENO-3 (top, dots), standard WENO-5 (top, crosses), WENO-Z (bottom, dots), simple WENO (bottom, crosses) and the subcell evolution method (diamonds), compared to the analytical result (black line) at $t = 0.4$. For the troubled cell indication, we set $M = 5$.

on the domain $x \in [0, 1]$. The analytical solution for this problem in the context of SRHD is given by [52]. During our tests, we observe the troubled cell indicator to work reliably, since the grid patches which evolve the shock and the rarefaction wave are marked as troubled. All methods, the standard DG-WENO methods, the simple WENO approach as well as the subcell projection method, are able to provide a stable evolution of the shock tube problem, shown in Fig. 3.

B. Two-dimensional problems

Generalizing our results to more complex two-dimensional wave setups, we perform two tests as presented [25]: A shocklike test with the initial conditions

$$(\rho, v_x, v_y, p)(x,0) = \begin{cases} (0.03515, 0, 0, 0.163) & \text{if } x > 0, y > 0 \\ (0.1, 0.7, 0, 1) & \text{if } x < 0, y > 0 \\ (0.5, 0, 0, 1) & \text{if } x < 0, y < 0 \\ (0.1, 0, 0.7, 1) & \text{if } x > 0, y < 0 \end{cases}, \quad (53)$$

and a vortexlike test with the initial conditions

$$(\rho, v_x, v_y, p)(x,0) = \begin{cases} (0.5, 0.5, -0.5, 5.0) & \text{if } x > 0, y > 0 \\ (1, 0.5, 0.5, 5.0) & \text{if } x < 0, y > 0 \\ (3.0, -0.5, 0.5, 5.0) & \text{if } x < 0, y < 0 \\ (1.5, -0.5, -0.5, 5.0) & \text{if } x > 0, y < 0 \end{cases}, \quad (54)$$

with $(x, y) \in [-1, 1] \times [-1, 1]$. During the evolution of both cases, all initial discontinuities are captured by the troubled cell indicator. We get the results as shown in Fig. 4. We tested in detail the standard WENO and the DG + subcell scheme; the figures show that the WENO-5 and DG + subcell evolution give qualitatively the same results. In the case of the shock tube, Eq. (53), left panels, fewer cells are marked troubled for the DG + subcell scheme. Furthermore, the DG + subcell method resolves steep gradients better than the standard WENO reconstruction. This becomes most dominant in a domain around $x = y = -0.2$. However, due to the larger computational expenses the DG + subcell scheme is a factor of ~2.4 times slower than the standard WENO method.

The right panels of Fig. 4 represent the vortex test, cf. (54). As for the shock tube, both methods are able to resolve the structure properly. Again the DG + subcell method gives more accurate results, i.e. acting less dissipative keeping shock regions resolved, but also needs more computational resources and is ~3.2 times slower than the standard WENO implementation.

For a systematic comparison of the efficiency of the two methods we could increase the resolution of the WENO method until it matches the accuracy of the subcell scheme and then compare run times. We postpone a detailed study of efficiency since this is more relevant for real applications, where an important part of the computational load is how many cells actually require the more expensive shock treatment.

VI. GENERAL RELATIVISTIC HYDRODYNAMICS

As the final test of our new implementation, we consider relativistic material in a curved spacetime background and present results for a TOV star in Cowling approximation in one dimension, two dimensions, and 3D. Notice, however, that the one-dimensional and two-dimensional description is not identical to the 3D star. Being more specific, surfaces of constant densities correspond for the one-dimensional test to planes, for the two-dimensional test to cylindrical shells, and for the 3D test to spherical shells; cf. discussion below.

A. Initial configuration

Initial configurations for a single spherical symmetric neutron star are obtained by solving the TOV equation [26,27]. The four-metric for a TOV star is given by

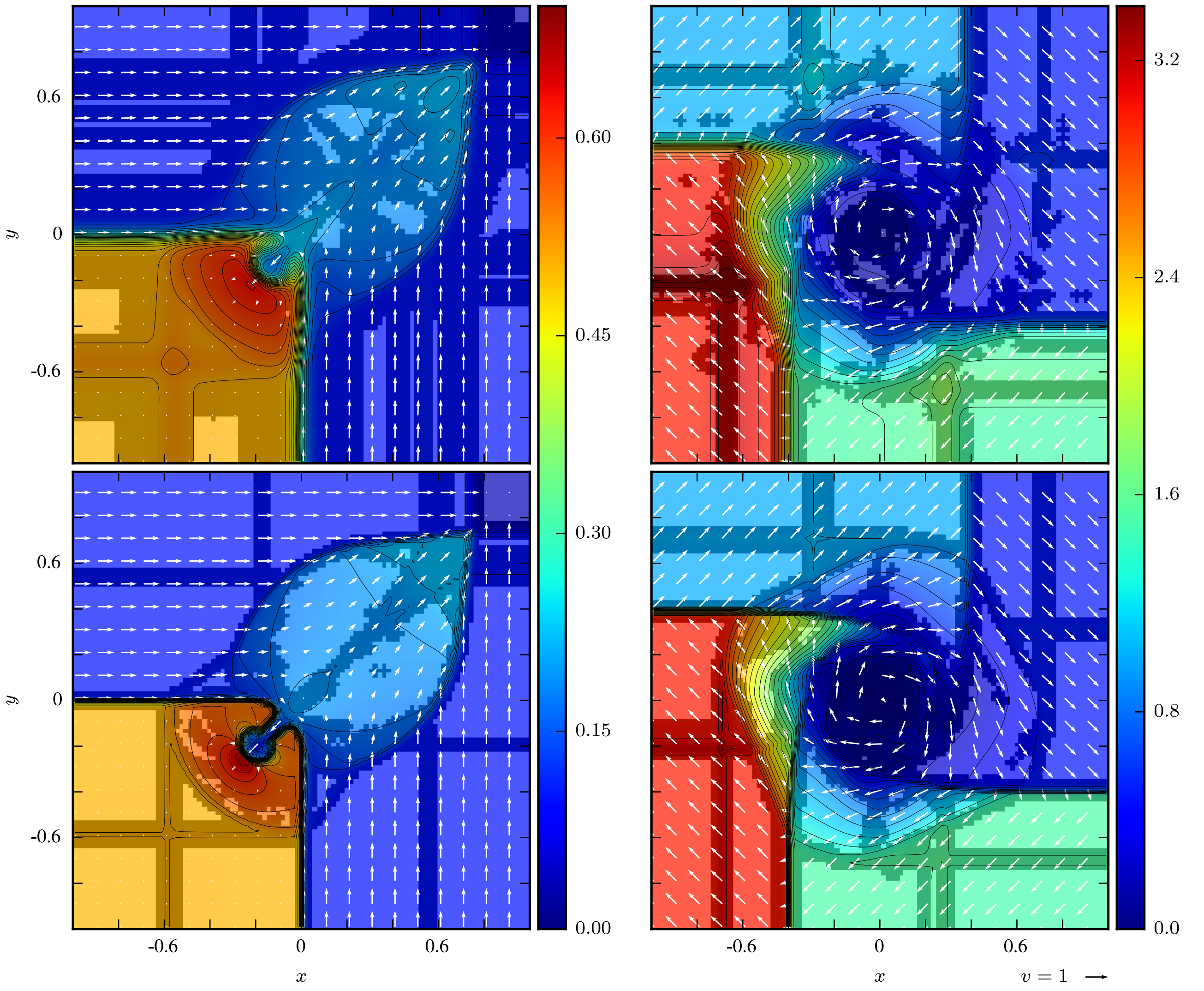

FIG. 4. Special relativistic hydrodynamics simulations in two dimensions for the shock tube problem (53) (left) and the vortex problem (54) (right) at $t = 0.8$, each evolved with the standard WENO-5 reconstruction (top) and the subcell evolution method (bottom) using $n = 100 \times 100$ grid patches, polynomials of order $N = 3$, $M = 5$, CFL $= 0.25$: Density plot with contours, corresponding velocity field (arrows) and troubled cells (shaded regions).

$$ds^2 = -e^{2\phi}dt^2 + \left(1 - \frac{2m}{R}\right)^{-1} dR^2 + R^2 d\Omega^2. \tag{55}$$

To obtain $m(R), \phi(R)$, and the pressure $p(R)$, the TOV equations,

$$\frac{d\rho}{dR} = (\rho(1+\epsilon) + p)\frac{m + 4\pi r^3 p}{R(R-2m)} \cdot \frac{1}{\frac{dp}{d\rho}}, \tag{56}$$

$$\frac{dm}{dR} = 4\pi R^2 \rho(1+\epsilon), \tag{57}$$

$$\frac{d\phi}{dR} = \frac{m + 4\pi R^3 p}{R(R-2m)}, \tag{58}$$

are solved with an explicit fourth order Runge-Kutta algorithm. As starting values $\rho(R=0) = \rho_{\text{central}}$, $m(R=0) = 0$, and $\phi(R=0) = 0$ are specified and the system is closed by the polytropic EOS Eq. (39) ($\Gamma = 2, K = 100$). Afterwards a coordinate transformation is performed to obtain the metric in isotropic coordinates, which we use for the evolution, because α and ψ^4 can easily be obtained from this form. This solution describes the spacetime of a static, spherically symmetric star. However, due to the discontinuity at the star surface and truncation errors, the evolution is nontrivial.

B. One-dimensional-TOV tests

Before we investigate the performance of our newly implemented algorithms in full 3D simulations, we

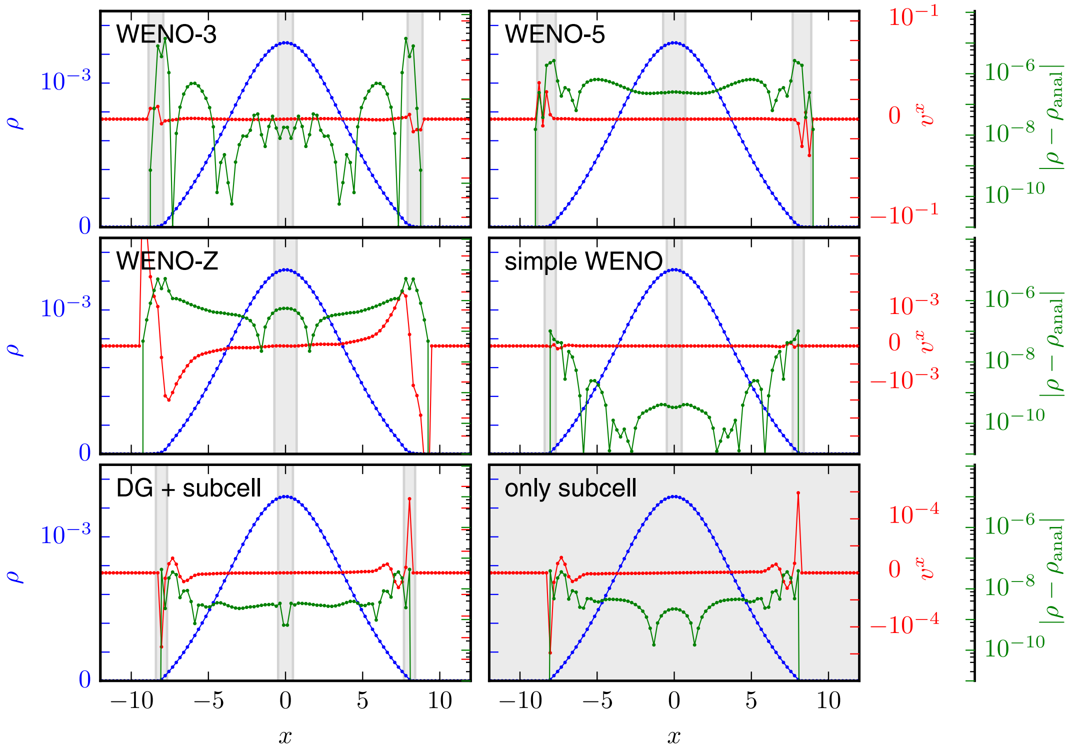

FIG. 5. Density (blue), density error (green) and velocity (red) cell averages at $t = 1000$ (≈40 light-crossing times) for a one-dimensional TOV star using $n = 100$ cells, polynomials of order $N = 3$, CFL $= 0.25$, $f_{\rm atm} = 1 \times 10^{-8}$ and $f_{\rm thr} = 100$ evolved with DG and several shock resolution methods. The cells marked as troubled at $t = 1000$ are colored in gray. The result of the pure subcell run, which is equivalent to a finite difference simulation, is shown for comparison.

consider configurations similar to the TOV star in just one or two dimensions. A natural option to consider is the Cartoon method [14,53,54], where the 3D equations are implemented on one-dimensional or two-dimensional domains by using boundary conditions derived from spherical or axial symmetry. However, for the present work we found that there is a simple alternative. We obtain a quasistationary test case in the Cowling approximation by considering the standard 3D TOV data on a one-dimensional line or two-dimensional plane and setting all derivatives not contained in the domain (say, the y and/or z direction) to 0. The numerical results indicate that the 3D TOV data are a stationary configuration under these conditions.

Setting certain derivatives at the boundary to 0 is similar to implementing a translation symmetry in that direction, while the boundary conditions to obtain the same evolution as for the TOV star in 3D are given by e.g. the Cartoon method. Referring to [14], for axisymmetry about the z axis we have $\partial_y u(x,0,z) = 0$ for a scalar, while for a vector $\partial_y v^y(x,0,z) = v^x/x$. Hence for scalars the derivative is 0, while for vectors it is not 0 in general. However, in the TOV example the relevant vector variables are in fact 0 as well. In particular, the velocity is 0, so again the result of the partial derivative at the boundary is 0. Notice also that no second derivatives are present in Eq. (1) and that due to the restriction to the Cowling approximation first and second derivatives present in the metric field equations do not affect the simulation. In conclusion, we expect a numerical evolution for the truncated one-dimensional and two-dimensional models that is similar but not identical to that of the full 3D-TOV example, while the analytic equilibrium configuration is the same.

We consider the one-dimensional-TOV model here and the two-dimensional-TOV model in the next section. Because of the smaller computational costs, we discuss in detail one-dimensional-TOV results for all reconstruction algorithms; in particular, we study WENO-3, WENO-5, WENO-Z, simple WENO reconstruction, as well as a DG + subcell and a pure subcell method for comparison. We have set in all our tests $f_{\rm atm} = 10^{-8}$ and $f_{\rm thr} = 10^2$. Figure 5 shows the density ρ (blue), the velocity v^x (red), and the difference $|\rho - \rho_{\rm anal}|$ (green), where $\rho_{\rm anal}$ refers to initial condition constructed according to Sec. VI A. All reconstruction algorithms lead to stable evolutions. In general we observe three regions of troubled cells, the left star surface, the maximum of the density, and the right star surface. During the evolution some troubled cells are activated or deactivated, which explains why for WENO-Z reconstruction at the presented time $t = 1000$ the surfaces are not marked as troubled.

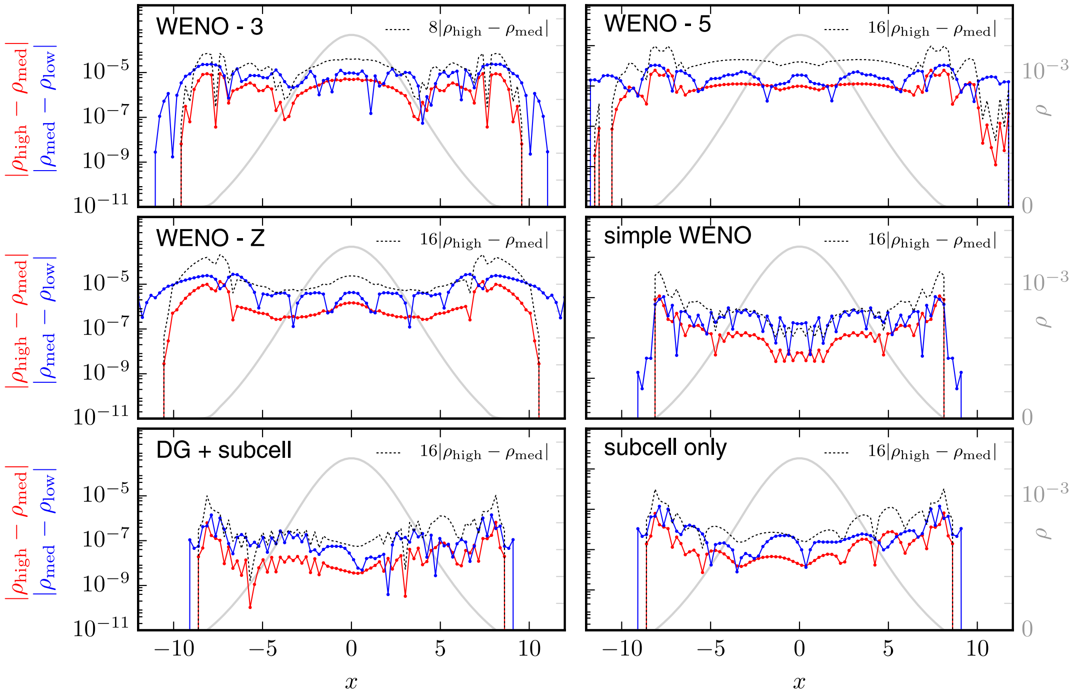

FIG. 6. Convergence test for a one-dimensional TOV star at $t = 100$ for three resolutions $n_{\rm high} = 100, n_{\rm mid} = 50, n_{\rm low} = 25$, polynomials of order $N = 3$, CFL $= 0.25$, $f_{\rm atm} = 1 \times 10^{-8}$ and $f_{\rm thr} = 100$ evolved with DG and several shock resolution methods.

We observe that WENO-3, WENO-5, and WENO-Z perform worst, i.e. large velocities are present at the stars' surface and $|\rho - \rho_{\rm anal}|$ is larger as for the other reconstruction mechanisms (notice the different y scales for v^x and $|\rho - \rho_{\rm anal}|$). The best results are obtained with the simple WENO and DG + subcell methods. The total L_1 errors of ρ for the given setup are 6.0×10^{-7} for simple WENO and 4.6×10^{-7} for DG + subcell method. The pure subcell evolution performs as well as the DG + subcell method.

The advantage of the simple WENO and subcell methods can be understood by considering the effectively higher resolution compared to the other schemes. In the standard WENO case, only the cell averages are used for the componentwise reconstruction; therefore the effective resolution drops depending on the employed polynomial order. In contrast, the simple WENO approach uses the full information of the polynomial inside the cell and additionally uses only three cells for the reconstruction; thus no significant performance loss is obtained and the simple WENO reconstruction is a factor 1.57 slower than the standard WENO-3 approach (a factor 1.40 slower than the standard WENO-5 approach). Finally, in the DG + subcell method points are added in problematic regions. Because of the additional computational effort due to the projection between top and subcells and the larger number of points in the troubled cells, the algorithm is a factor of ∼1.67 slower than the standard WENO method. Although not noticeable for one-dimensional setups, we encounter for higher dimensional setups a significantly larger amount of memory, i.e. a ∼2.7 times higher memory load for two-dimensional runs (∼4.8 times higher for 3D) when subcells are activated compared to standard WENO-3 simulations. Nevertheless the DG + subcell approach seems to be a valid choice for further development, because it (i) allows us to reuse well-tested FD schemes in troubled regions, (ii) gives the most accurate results due to an effectively higher resolution in troubled regions, and (iii) allows a speed up compared to the usually employed FD codes, because of a more effective DG method in large parts of the numerical domain.

In Fig. 6 we present a pointwise convergence test for all methods. We compare evolutions with 25,50,100 cells and use polynomials of order 3. The difference between the low and medium resolution is shown in blue, while the difference between the medium and high resolution is shown in red. We rescale the difference of the medium and high resolution according to the expected convergence order, i.e. third order for WENO-3 and fourth order for the other schemes. We observe that in all cases we obtain roughly the expected convergence order. Furthermore in the logarithmic plots it is clearly visible that for some setups the outer regions of the star are smeared out. In the case of standard WENO algorithms larger stencils (WENO-5 and WENO-Z) lead to a numerical solution where the outer star layers are not fixed and no sharp surface is visible; this improves for the WENO-3 reconstruction. Contrary to this, the simple WENO and DG + subcell method keep the surface of the star fixed. In all runs higher resolution improves the results and less material is leaving the star.

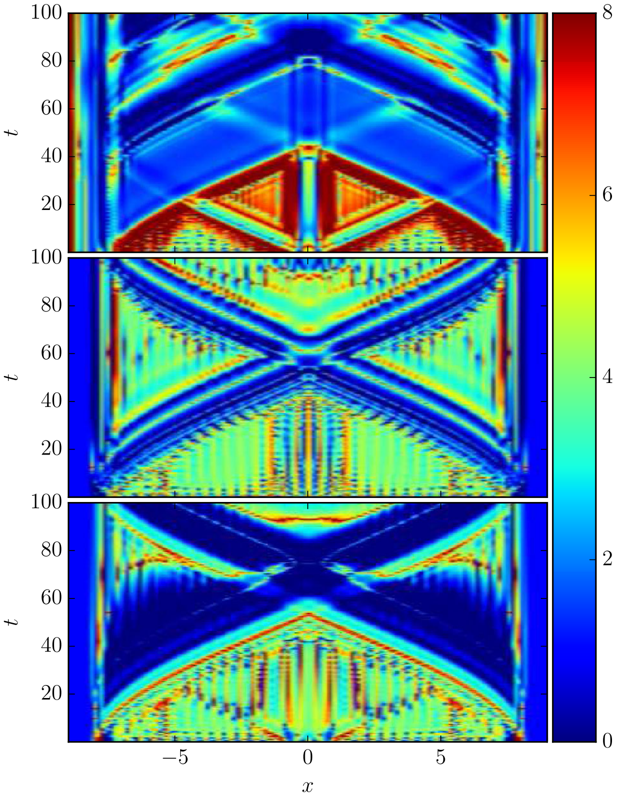

FIG. 7. Convergence order for a one-dimensional TOV star during evolution $t \in [0, 100]$ for two resolutions $n_{\text{high}} = 100$, $n_{\text{low}} = 50$, polynomials of order $N = 3$, CFL $= 0.25$, $f_{\text{atm}} = 1 \times 10^{-8}$ and $f_{\text{thr}} = 100$ evolved with DG and three shock resolution methods: standard WENO-3 (top), simple WENO (middle) and DG + subcell (bottom).

The simplicity of the one-dimensional-TOV star allows us to consider setups with higher resolution than achievable in the corresponding two-dimensional and 3D tests and a more detailed analysis becomes possible. A recurring question is how regions with low order convergence (because of low differentiability of the solution) affect regions where the solution is smooth. Specifically, do the regions of low order remain localized, or if not, how quickly does the loss of convergence spread through the entire domain? See for example [55], where the wave equation with discontinuous initial data is studied, for which analytic results are available in [56] predicting the growth of the nonconvergent area with, e.g., the square root of time, $\sim\sqrt{t}$.

Figure 7 shows the convergence order during the first stages of the evolution up to $t = 100$ for 50 and 100 cell setup. Presented are the WENO-3 (top panel), simple WENO (middle panel), and the subcell (bottom panel) evolution. For all panels, we observe that inside the star, where also cells are marked as troubled, the WENO-3 method shows ~third order convergence and the simple WENO method a convergence order above 4. Furthermore, while for WENO-3 the error seems to corrupt the convergence in the entire star it seems to be localized for simple WENO for the entire simulation. For the subcell evolution we observe that the convergence order at the stars' center lies between second and third order, which is consistent with the employed flux methods implemented in the FD subcells [41]. Artificially setting the center cells as nontroubled cures this problem and leads locally to higher order convergence. However, it has no influence on the global convergence order. More problematic, a large error is traveling inwards from the outer surface for all simulations, which leads to a lower convergence order for all setups.

For runs until $t = 1000$, there is further degradation of convergence. This is a delicate issue which deserves further investigation. We are not aware of comparable studies in the literature, although long term convergence is an important issue. Notice that this effect is not related to the movement of troubled cells. The region of troubled cells stays relatively fixed at the stars' surface. Nevertheless the convergence order drops across the entire star. Regarding this fact, it is debatable whether one can obtain high order convergence in more general setups, e.g. dynamical spacetimes and moving objects.

C. Two-dimensional TOV star

Considering the results of the previous section, the simple WENO and DG + subcell schemes seem to be preferable. However, we found that the simple WENO method performs worse in higher dimensional problems as in the one-dimensional case. Compared to the standard WENO reconstruction, where a smoother polynomial from several cell averages is constructed, the simple WENO methodology allows steeper gradients and has weaker smoothing influence. For runs of higher dimensional problems we observe this smoothing to be crucial for the stability of the evolution. Furthermore, the simple WENO computation underlies a significant slowdown in $d > 1$ dimensions, because the evaluation of the smoothness indicators is a quadratic form of all $(N + 1)^d$ coefficients. This is the reason why the standard WENO-3 scheme, which turns out to allow stable evolutions, is used instead.

We investigated the convergence of the two schemes, standard WENO-3 and DG + subcell, for the two-dimensional TOV star[1] regarding the density L_1 error. As shown in Fig. 8, we observe a convergence order of ~2 for the subcell scheme, which indicates that the evolution error originating in the subcells spreads over the grid and leads to a lower order of convergence. In comparison, the standard WENO-3 scheme converges in third order for

[1] As for the one-dimensional test, we employ Cartesian coordinates and due to the restriction to a fixed spacetime background also our two-dimensional-TOV example is in hydrodynamical equilibrium.

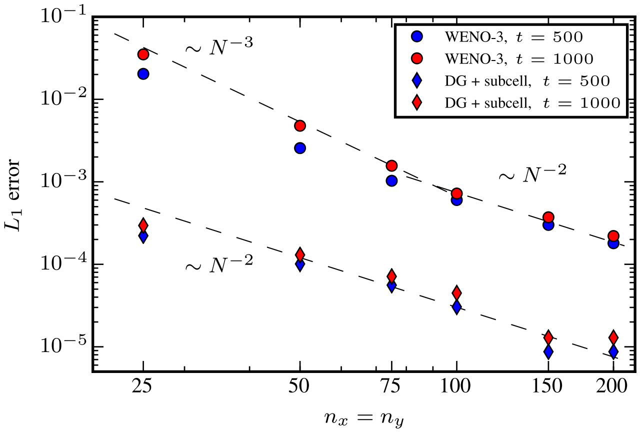

FIG. 8. Density L_1 error for a two-dimensional TOV star at $t = 500$, $t = 1000$ for six resolutions $n = 25$, 50, 75, 100, 150, 200, polynomials of order $N = 3$, CFL $= 0.25$ and $f_{\text{thr}} = 100$ evolved with DG + standard WENO-3 and the DG + subcell evolution method. For WENO-3 we set $f_{\text{atm}} = 1 \times 10^{-8}$; for the DG + subcell we set $f_{\text{atm}} = 1 \times 10^{-9}$. The dashed black lines correspond to second/third order convergence.

coarse grids. The subcell method causes a much higher memory load and longer calculation times (see Sec. VI B) because of the higher number of grid points in each direction. For this reason, we decided to use the standard WENO-3 method for the 3D simulation of a TOV star.

D. 3D TOV star

Considering a 3D TOV star, we are able to provide a stable simulation with a DG + standard WENO-3 method. Although we show our results up to $t = 1000$, there is no evidence of any instabilities for longer runs. We are considering two numerical setups at resolutions $25 \times 25 \times 25$ and $50 \times 50 \times 50$ and the initial configuration as a reference solution. After a short transition, the numerical simulations reach an almost steady structure as shown in Fig. 9. The density profiles along the x and y axis are shown as red and blue lines. The difference between the densities for $z = 0$ is presented as the gray shaded region. On the bottom panel, we present the computed convergence order. In large areas of the star second to fourth order convergence is present and even higher convergence in its center and outside areas near the surface. The latter can be explained by the failure of the coarse grid setup to keep the density on atmosphere level outside the star, whereas the fine grid setup does. The narrow band of low convergence (colored blue) inside the star can be explained as follows: The finer resolved solution stays closer to the density maximum in the star center and 0 at the star's surface; the opposite holds for the coarse resolution. Thus, the differences tend to 0, see the solid black line, and the convergence drops locally.

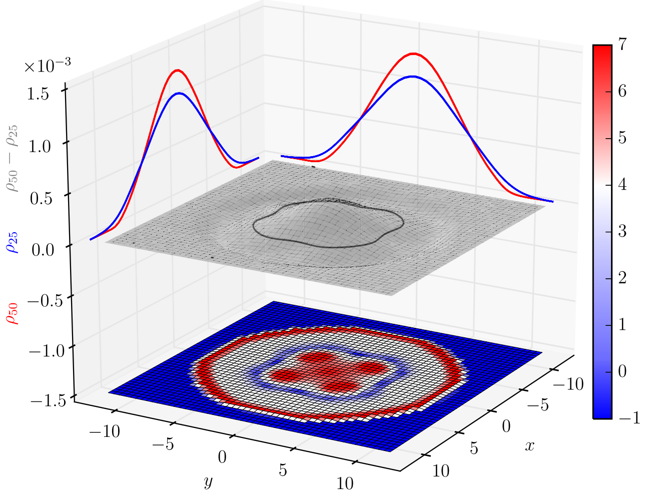

FIG. 9. Pointwise convergence order in the $z = 0$ plane for the density of a 3D TOV star at $t = 500$ using two resolutions $n_{\text{high}} = 50$, $n_{\text{low}} = 25$, polynomials of order $N = 3$, CFL $= 0.25$, $f_{\text{atm}} = 1 \times 10^{-8}$ and $f_{\text{thr}} = 100$ evolved with DG and standard WENO-3. The two density solutions for $n = 50$ (red) and $n = 25$ (blue) are shown on the axes $x = z = 0$ and $y = z = 0$. Their difference in the z-plane is shown in gray; the corresponding zero crossing is indicated by the gray contour line.

As a global measurement of the convergence order, we present the L_1 norm for the 3D TOV star in Fig. 10 for four different resolutions. Similar to the two-dimensional test case, we observe an almost third order convergence (black dashed line) for the standard WENO-3 algorithm with third order polynomials. Using a fitting function of the form $A \cdot n_x^{-b}$ for the L_1 error, the obtained convergence order is $b = 2.75$ for $t = 500$ and $b = 2.88$ for $t = 1000$ and thus close to the theoretical expected value.

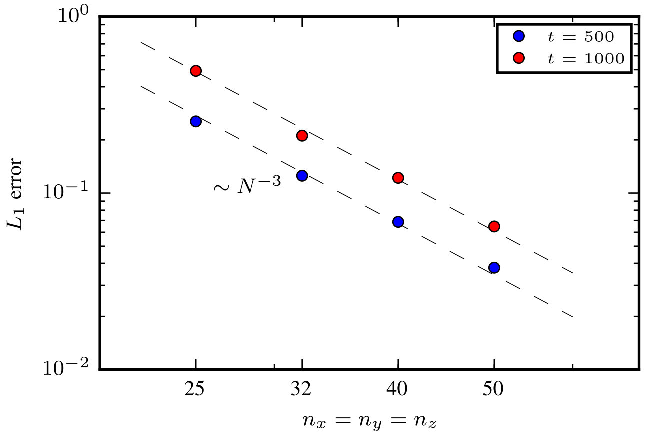

FIG. 10. Density L_1 error for a 3D TOV star at $t = 500$, $t = 1000$ for four resolutions $n = 20$, 32, 40, 50, polynomials of order $N = 3$, CFL $= 0.25$, $f_{\text{atm}} = 1 \times 10^{-8}$ and $f_{\text{thr}} = 100$ evolved with DG and standard WENO-3. The dashed black lines correspond to third order convergence.

VII. CONCLUSION

In this work, we presented new algorithms implemented in the existing bamps code: a DG, a WENO-DG, and a mixed FD + DG-algorithm combined with standard WENO [8,40] and a simple (compact) WENO scheme [28]. We tested all algorithms and reconstruction methods with a number of tests starting with the advection and Burgers equation, the main results being examples for special and general relativistic hydrodynamics. In almost all cases, we were able to obtain the expected convergence order for smooth solutions and also found a proper shock treatment in the case of jumps and discontinuities.

Our main result was the simulation of a single TOV star, which we modeled in the Cowling approximation, i.e. for static geometric variables. In fact, while it has not been attempted yet to apply the existing DG methods for the vacuum Einstein equations [20,21] to 3D GRHD, we have demonstrated recently that the pseudospectral multipatch method of bamps [14] works well for demanding 3D vacuum spacetimes (highly nonlinear gravitational waves that collapse to a black hole). The pseudospectral penalty method developed in [14] can be viewed as a special case of a DG method for the full Einstein equations in a nonflux form. Furthermore, the present work on GRHD and the wave-collapse simulations are compatible in the type of variables and equations they use, and can run with the same spectral element grid and polynomial basis functions (Chebyshev or Legendre-Gauss-Lobatto grids). Therefore, we do not expect any immediate obstacle to combine the existing geometry code with the new GRHD methods.

Let us note, however, that additional development may be required to handle more complicated physical models. As a step beyond the simple TOV star we have started to investigate a rotating star in equilibrium, but we encountered problems that require further study. Although stable simulations are possible, we leave a careful investigation to future work.

With regard to numerical methods, a simplifying restriction in our implementation was the usage of a simple troubled cell indicator. We intend to study the influence of different and more sophisticated troubled cell indicators. While our simple setup allowed an easy implementation and stable evolutions, it also marked maxima as troubled, which should be avoided in the future application of the code.

Keeping the number of employed cells fixed, we found that for one-dimensional problems the subcell and simple WENO algorithms were the most accurate ones. This can be easily understood, since the standard WENO method is based only on cell averages for the reconstruction. In contrast, the simple WENO method uses the knowledge of the entire polynomial, and the subcell methods resolved troubled cells with effectively $2N + 1$ times more points. In our examples, the simple WENO and subcell methods have some drawback for higher dimensions. Both methods come with a significant overhead, and it is planned to investigate more efficient implementations in the future. More importantly, the direct application of the simple WENO reconstruction in two dimensions led to unexpected instabilities, for example in the computation of the primitive from the conservative variables, which one should be able to avoid. This issue certainly deserves further study since the simple WENO method is very promising based on the one-dimensional results.

Due to the large computational cost of the subcell method, we only employed the standard WENO reconstruction in 3D and investigated the subcell method in two dimensions. In our two-dimensional examples, the subcell method turned out to be (as expected) approximately second order. For the standard WENO-DG method we found third order convergence for low and second order convergence for high resolutions. The observed third order convergence is consistent with our results in the full 3D simulation. However, for a high number of cells we noticed that a higher than second order convergence in the matter variables seems to be hard to obtain, since (i) the computation of the L_1 norm of the error emphasizes inaccurate, problematic regions, and (ii) errors propagate from the surface of the star through the neutron star and corrupt the order of convergence. Although this fact can be seen as a setback, DG methods allow a better parallelization and refinement strategy than fixed FD codes and are one of the most promising methods to take into account for future GRHD-code developments.

Our work can be seen as a first step towards a complete 3D-DG implementation for GRHD, since it employs DG-methods for GRHD problems beyond the limitation of spherical symmetry as in previous work. It is planned to further develop the numerical techniques by considering adaptive mesh refinement, and to extend the physics to full general relativity (beyond the Cowling approximation), which together will allow numerical simulations of astrophysical systems consisting of single and binary neutron stars.

ACKNOWLEDGMENTS

It is a pleasure to thank E Harms, D. Hilditch, N. Moldenhauer, M. Pilz, and H. Rüter for helpful discussions. We also thank S. Field and the authors of [57] for sharing numerical data that allowed the direct code comparison in B. This work was supported in part by DFG Grant No. SFB/Transregio 7 “Gravitational Wave Astronomy” and the Graduierten-Akademie Jena. The authors acknowledge the usage of computer resources at the GCS Supercomputer SuperMUC, JUROPA at Jülich Supercomputing Centre, and at the Institute of Theoretical Physics of the University of Jena.

APPENDIX A: DG METHOD FOR SCALAR CONSERVATION LAWS IN ONE DIMENSION

Following [4,5], we summarize some aspects of the DG method that already arise in the nonlinear, scalar, one-dimensional case. We add some details relevant to the present work concerning implementation issues and the equivalence of the once and twice integrated form of the equations. One of our goals is the combination of the DG method for relativistic matter with the pseudospectral penalty method of [14] for the geometry, which is not using a flux conservative form, but is close to the strong formulation of the DG method given below in (A7). Therefore we examine the question of how the discretized equations for the weak and strong form are related.

1. Derivation of the discretized equations

Consider

$$\partial_t u + \partial_x f(u) = 0 \tag{A1}$$

for a function $u(t,x)$ and a flux function $f(u(t,x))$ on the interval $I = [-1, 1]$. Given a space of test functions on I, we obtain the weak form of the conservation law by integration. For a test function $v(x)$,

$$(v, \partial_t u) + (v, \partial_x f) = 0. \tag{A2}$$

Later we assume that the scalar product of two functions is $(f_1, f_2) = \int_{-1}^{1} f_1(x) f_2(x) dx$, i.e. we assume the trivial measure which is the natural weight for the polynomial basis of Legendre polynomials.

Part of the DG method is a special treatment of the flux at the boundaries of I, where we replace the flux f by a nonunique choice of a numerical flux f^*. Integrating (A2) by parts in space we arrive at two versions of the conservation law,

$$(v, \partial_t u) - (f, \partial_x v) = -[v f^*], \tag{A3}$$

$$(v, \partial_t u) + (v, \partial_x f) = [v(f - f^*)], \tag{A4}$$

where $[g] = g(1) - g(-1)$ for any function $g(x)$. Equation (A3) is obtained by integrating by parts and replacing f by f^* at the boundary. Equation (A4) is obtained from (A3) by integrating by parts once more but leaving the resulting boundary term unchanged. We refer to (A3) and (A4) as the weak and strong form, respectively, or as the once and twice integrated flux equation to avoid confusion with the original strong form, (A1).

The nodal DG spectral element method is based on a choice of $N+1$ distinct nodes $x_i \in I$. Such nodes define the unique Nth-order Lagrange polynomials $\ell_i(x)$, for which $\ell_i(x_j) = \delta_{ij}$. We choose the x_i to be the collocation points for Legendre-Gauss (LG) or LGL integration. This allows the approximation of $u(x)$ by a nodal expansion that has the interpolation property $u(x_i) = u_i$, $I_m u(x) = \sum_{i=0}^{N} u_i \ell_i(x)$.

When the meaning is clear from context, we write u instead of $I_m u$. A key feature of the nodal expansion is that it works equally well for linear and nonlinear functions, in particular,

$$u(x) = \sum_{i=0}^{N} u_i \ell_i(x), \qquad f(u(x)) = \sum_{i=0}^{N} f_i \ell_i(x), \tag{A5}$$

where $f_i = f(u_i) = f(u(x_i))$.

The nodal approximation with Nth-order polynomials leads to discretized versions of the conservation laws (A3) and (A4). Choose test functions $v(x) = \ell_i(x)$, and insert (A5) to obtain

$$M \partial_t u - S^T f = -[\ell f^*], \tag{A6}$$

$$M \partial_t u + S f = [\ell (f - f^*)], \tag{A7}$$

where we have introduced the mass matrix M and stiffness matrix S,

$$M_{ij} = (\ell_i, \ell_j), \qquad S_{ij} = (\ell_i, \partial_x \ell_j). \tag{A8}$$

We use matrix notation and a summation convention, e.g. $Sf \equiv S_{ij} f_j \equiv \sum_{j=0}^{N} S_{ij} f_j$.

An important point is that in general the mass matrix is not diagonal, that is, the characteristic Lagrange polynomials are not necessarily orthogonal. Specifically, for LGL the mass matrix is not diagonal, while for LG it is diagonal. However, for both LGL and LG the matrix is symmetric and invertible. The stiffness matrix is directly related to the derivative matrix,

$$D_{ij} = \partial_x \ell_j(x_i), \tag{A9}$$

which approximates the pseudospectral derivative at the nodes by $(\partial_x u)(x_i) = D_{ij} u_j$. We have [4]

$$S = MD, \quad M^{-1} S = D, \quad M^{-1} S^T = M^{-1} D^T M. \tag{A10}$$

Given M, D, and a prescription for f^*, we solve the explicit time-integration problem based on (A6) or (A7),

$$\partial_t u - (M^{-1} D^T M) f = -M^{-1} [\ell f^*], \tag{A11}$$

$$\partial_t u + D f = M^{-1} [\ell (f - f^*)], \tag{A12}$$

for the discretized, time-dependent function values $u_i(t)$.

The method generalizes immediately to a partition of any interval $[a, b] \in \mathbb{R}$ into several elements I_j with an appropriate mapping of the coordinates and with a coupling of neighboring elements through f^*.

2. Implementation issues

Let us comment on some implementation issues, specifically for the LGL method. The nodes x_i and the LGL integration weights w_i are obtained from the Legendre polynomials, for which simple but stable and accurate algorithms are available, e.g. [5]. The nodes x_i are the $N-1$ roots of $\partial_x P_N(x)$ combined with the end points of the interval, -1 and 1, for a total of $N+1$ nodes. The integration weights are $w_i = 2/(N(N+1)P_N(x_i)^2)$. Various other quantities are determined without further reference to the Legendre polynomials by general formulas for Lagrange interpolation. The weights for barycentric interpolation are $c_i = \prod_{j=0,j\neq i}^{N} 1/(x_i - x_j)$. The derivative matrix is

$$D_{ij} = \partial_x \ell_j(x_i) = \frac{c_j}{c_i}\frac{1}{x_i - x_j}, \qquad i \neq j, \tag{A13}$$

$$D_{ii} = -\sum_{j=0}^{N} D_{ij}. \tag{A14}$$

The equation for the diagonal term ensures that the numerical derivative of a constant like $u_i = 1$ is 0 [58]. Since the end points are included among the nodes, $x_0 = -1$ and $x_N = 1$,

$$[\ell_i g]_{-1}^{1} = \ell_i(1)g(1) - \ell_i(-1)g(-1) = \delta_{iN}g_N - \delta_{i0}g_0. \tag{A15}$$

There are several ways to compute the mass matrix $M_{ij} = (\ell_i, \ell_j)$. One option is to perform the integration numerically according to the Gauss formula associated with the nodes, which approximates the integral of a function $g(x)$ using the integration weights w_i,

$$\int_{-1}^{1} g(x)dx \simeq \sum_{i=0}^{N} w_i g(x_i). \tag{A16}$$

This integration is exact if $g(x)$ is a polynomial of degree up to $2N+1$ for LG and up to $2N-1$ for LGL. Since the integrand $\ell_i\ell_j$ for the mass matrix is of degree $2N$, for LG the numerical integral is exact,

$$M_{ij} = (\ell_i, \ell_j) = (\ell_i, \ell_j)_N = w_i\delta_{ij}, \tag{A17}$$

where $(f,g)_N = \sum_i w_i f_i g_i$ denotes the numerical scalar product. However, for LGL we only obtain the approximation

$$M_{ij} = (\ell_i, \ell_j) \simeq (\ell_i, \ell_j)_N = w_i\delta_{ij}. \tag{A18}$$

It turns out that this approximation, also called mass lumping, is equivalent to a certain filter that strongly affects the highest mode in the Legendre basis and that can reduce the effective order of the approximation [39]. In the context of spectral element methods of comparatively high order, say $N = 10$, approximating M for LGL by the diagonal matrix as in (A18) is considered standard in [5]. However, for orders around $N = 2$, 3, 4, it is often preferable to evaluate $M_{ij} = (\ell_i, \ell_j)$ for LGL without approximation [4]. For example [4,39], $M^{-1} = VV^T$, where V is the generalized Vandermonde matrix for the normalized Legendre polynomials. This relation follows from the expansion of the Legendre polynomials in the Lagrange basis. Computing the difference to the diagonal approximation we find for LGL

$$M_{ij}^{-1} = \frac{1}{w_i}\delta_{ij} + \frac{N+1}{2}P_N(x_i)P_N(x_j). \tag{A19}$$

See also [59], where this relation is derived using the Sherman-Morrison formula. Alternatively, note that M can also be computed directly as the analytic integral (ℓ_i, ℓ_j), either by term by term integration after expanding the product $\ell_i(x)\ell_j(x)$, or by exact Gauss integration on a secondary grid with $N+2$ points. (In experiments, $N+3$ gives somewhat more accurate results.) However, we still have to find the inverse of M numerically. For large N, (A19) may be preferred.

3. Equivalence of once and twice integrated forms

For the continuum problem, we perform the integration by parts,

$$(v, \partial_x f) = [vf] - (f, \partial_x v). \tag{A20}$$

Under specific but quite general conditions the discretized equations satisfy the corresponding summation by parts property exactly. In this case the once and twice integrated DG methods are numerically identical. There may be round-off errors, but there are no systematic errors that only converge away with increasing N. This is fully explained in [39,60,61].

Given the present setup, it is straightforward to show algebraic equivalence of (A6) and (A7). The difference between those two equations is

$$Sf = [\ell f] - S^T f, \tag{A21}$$

$$S = [\ell\ell] - S^T, \tag{A22}$$

for all f_i, and independently of the choice of f^* or the computation of M.

In the transition to (A22) we use that f is approximated by an Nth order polynomial, (A5). By definition of S_{ij},

$$\begin{aligned} S_{ij} &= (\ell_i, \partial_x \ell_j) \\ &= [\ell_i \ell_j] - (\ell_j, \partial_x \ell_i) \\ &= [\ell_i \ell_j] - S_{ji}, \end{aligned} \tag{A23}$$

so the summation by parts property (A22) does indeed hold. Summation by parts is exact for LG and LGL even if S_{ij} is defined by numerical integration because

$$S_{ij} = (\ell_i, \partial_x \ell_j) = (\ell_i, \partial_x \ell_j)_N \tag{A24}$$

since $\ell_i \partial_x \ell_j$ is a polynomial of degree $2N - 1$.

It is instructive to make the summation by parts formula for the LGL method more explicit. From (A24) and (A16),

$$S_{ij} = \sum_k w_k \ell_i(x_k) \partial_x \ell_j(x_k) = w_i D_{ij}. \tag{A25}$$

(Incidentally, this means that $S_{ij} = M_{ik} D_{kj} = w_i D_{ij}$ for both LG and LGL.) Hence (A22) becomes

$$w_i D_{ij} = [\ell_i \ell_j] - w_j D_{ji}. \tag{A26}$$

We now restrict ourselves to the LGL case. *A priori* it is not clear how a simple rescaling and a transpose of the derivative matrix leads to the term $[\ell_i \ell_j] = (\delta_{iN} - \delta_{i0})\delta_{ij}$, which is a diagonal matrix with nonvanishing entries only in two of the corners. For $i = j$,

$$D_{ii} = \partial_x l_i(x_i) = \frac{1}{2w_i}(\delta_{iN} - \delta_{i0}), \tag{A27}$$

$$2w_i D_{ii} = \delta_{iN} - \delta_{i0} = [\ell_i \ell_i], \tag{A28}$$

so (A26) is satisfied on the diagonal. In particular, we see how the boundary terms come about. For $i \neq j$, (A26) becomes

$$D_{ij} = -\frac{w_j}{w_i}(D^T)_{ij}, \tag{A29}$$

from which we obtain with (A13) that

$$\frac{c_j}{c_i} = \frac{w_j}{w_i}\frac{c_i}{c_j}. \tag{A30}$$

In other words, the summation by parts rule implies for LGL points a relation between the integration weights w_i and the barycentric interpolation weights c_i,

$$(c_i^{\mathrm{LGL}})^2 = C w_i^{\mathrm{LGL}}, \tag{A31}$$

for some constant C. Surprisingly, the explicit relation between w_i^{LGL} and c_i^{LGL} was only found recently; see [62] on such relations for Jacobi polynomials. For our case,

$$c_i^{\mathrm{LGL}} = C_N (-1)^i \sqrt{w_i^{\mathrm{LGL}}}, \tag{A32}$$

where C_N is an explicitly known constant that depends on the number of points. In summary, for LGL (or analogously for LG), we can start from the general result on summation by parts and arrive at a partial proof of (A32), or we can start from relations like (A32) and prove summation by parts without directly using partial integration in the continuum.

APPENDIX B: ERROR AND CONVERGENCE FOR THE SMOOTH SINE WAVE POPAGATION, EQ. (51)

While this paper was under review, a detailed description of SpECTRE, a task-based DG code for relativistic astrophysics, appeared in [57]. We collaborated with the authors to compare numerical results for a specific example, which is an important code validation. In Table IV, we show additional data on the smooth flow

TABLE IV. Numerical errors and convergence orders for problem (51) at $t = 2$ for different numbers of grid patches n_x, DG polynomials of order $N = 1$ and $N = 3$, and no additional shock resolution method. The columns correspond to the following setups: (a) analytic mass matrix, L_1 error from primitive variables; (b) lumped mass matrix, L_1 error from primitive variables; and (c) lumped mass matrix, L_1 error from conservative variables.

		(a)		(b)		(c)	
n_x	$N+1$	L_1 error	Order	L_1 error	Order	L_1 error	Order
10	2	1.22×10^{-3}		2.04×10^{-2}		2.55×10^{-2}	
20		2.73×10^{-4}	2.15	5.28×10^{-3}	1.95	6.60×10^{-3}	1.95
40		6.72×10^{-5}	2.02	1.33×10^{-3}	1.99	1.66×10^{-3}	1.99
80		1.67×10^{-5}	2.00	3.33×10^{-4}	2.00	4.16×10^{-4}	2.00
160		4.18×10^{-6}	2.00	8.32×10^{-5}	2.00	1.04×10^{-4}	2.00
320		1.04×10^{-6}	2.00	2.08×10^{-5}	2.00	2.60×10^{-5}	2.00
10	4	4.27×10^{-6}		9.92×10^{-6}		1.24×10^{-5}	
20		3.29×10^{-7}	3.70	6.30×10^{-7}	3.98	7.88×10^{-7}	3.98
40		1.79×10^{-8}	4.20	3.66×10^{-8}	4.11	4.57×10^{-8}	4.11
80		9.39×10^{-10}	4.25	2.25×10^{-9}	4.02	2.81×10^{-9}	4.02
160		6.01×10^{-11}	3.96	1.40×10^{-10}	4.00	1.75×10^{-10}	4.00
320		3.80×10^{-12}	3.97	8.77×10^{-12}	4.00	1.10×10^{-11}	4.00

sine wave problem (51) in order to compare numerical results with Kidder *et al.* [57], and to comment on the influence of the mass matrix implementation. The setup (c) in Table IV is algorithmically equivalent to the one presented in Sec. VI.1.1 of [57], but different from the method described in the present paper. Comparing the L_1 errors for $N+1=2$ and $N+1=4$, called $N_C=N+1$ in [57], we find excellent agreement between the two codes. Comparing actual numbers the difference is not larger than $\approx 5\times 10^{-12}$. Comparing the setups (a), which we use here, and (b) in Table IV, we find that using the analytic mass matrix leads to smaller errors than those from the mass lumped (diagonal) matrix scheme. However, the difference is comparatively small for higher order schemes, as expected from the discussion after (A18) and [4,5,39].

[1] C. L. Fryer and K. C. New, Living Rev. Relativ. **14**, 1 (2011).

[2] J. A. Faber and F. A. Rasio, Living Rev. Relativ. **15**, 8 (2012).

[3] C. Canuto, M. Y. Hussani, A. Quarteroni, and T. A. Zang, *Spectral Methods* (Springer-Verlag, Berlin Heidelberg, 2006).

[4] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods* (Springer, New York, 2008).

[5] D. A. Kopriva, *Implementing Spectral Methods for Partial Differential Equations* (Springer, New York, 2009).

[6] G. Karniadakis and S. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics* (Oxford University Press, Oxford, 2005).

[7] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems* (Cambridge University Press, Cambridge, 2007).

[8] J. Qiu and C.-W. Shu, SIAM J. Sci. Comput. **26**, 907 (2005).

[9] J. A. Font, Living Rev. Relativ. **11**, 7 (2008).

[10] B. Brügmann, J. A. González, M. Hannam, S. Husa, U. Sperhake, and W. Tichy, Phys. Rev. D **77**, 024027 (2008).

[11] E. Schnetter, P. Diener, N. Dorband, and M. Tiglio, Classical Quantum Gravity **23**, S553 (2006).

[12] L. Rezzolla and O. Zanotti, *Relativistic Hydrodynamics* (Oxford University Press, Oxford, 2013).

[13] SpEC—Spectral Einstein Code, http://www.black-holes.org/SpEC.html.

[14] D. Hilditch, A. Weyhausen, and B. Brgmann, Phys. Rev. D **93**, 063006 (2016).

[15] W. Tichy, Phys. Rev. D **80**, 104034 (2009).

[16] P. Grandclément and J. Novak, Living Rev. Relativ. **12**, 1 (2009).

[17] D. Radice, L. Rezzolla, and F. Galeazzi, Mon. Not. R. Astron. Soc. **437**, L46 (2014).

[18] D. Radice, L. Rezzolla, and F. Galeazzi, Classical Quantum Gravity **31**, 075012 (2014).

[19] D. Radice, L. Rezzolla, and F. Galeazzi, ASP Conf. Ser. **498**, 121 (2015).

[20] G. Zumbusch, Classical Quantum Gravity **26**, 175011 (2009).

[21] J. D. Brown, P. Diener, S. E. Field, J. S. Hesthaven, F. Herrmann, A. H. Mroué, O. Sarbach, E. Schnetter, M. Tiglio, and M. Wagman, Phys. Rev. D **85**, 084004 (2012).

[22] S. E. Field, J. S. Hesthaven, S. R. Lau, and A. H. Mroue, Phys. Rev. D **82**, 104051 (2010).

[23] D. Radice and L. Rezzolla, Phys. Rev. D **84**, 024010 (2011).

[24] S. A. Teukolsky, J. Comput. Phys. **312**, 333 (2016).

[25] J. Zhao and H. Tang, J. Comput. Phys. **242**, 138 (2013).

[26] R. C. Tolman, Phys. Rev. **55**, 364 (1939).

[27] J. R. Oppenheimer and G. Volkoff, Phys. Rev. **55**, 374 (1939).

[28] X. Zhong and C.-W. Shu, J. Comput. Phys. **232**, 397 (2013).

[29] R. Borges, M. Carmona, B. Costa, and W. S. Don, J. Comput. Phys. **227**, 3191 (2008).

[30] M. Thierfelder, S. Bernuzzi, and B. Brügmann, Phys. Rev. D **84**, 044012 (2011).

[31] T. Dietrich, S. Bernuzzi, M. Ujevic, and B. Brügmann, Phys. Rev. D **91**, 124041 (2015).

[32] S. Bernuzzi, A. Nagar, M. Thierfelder, and B. Brügmann, Phys. Rev. D **86**, 044030 (2012).

[33] T. Dietrich, N. Moldenhauer, N. K. Johnson-McDaniel, S. Bernuzzi, C. M. Markakis, B. Brügmann, and W. Tichy, Phys. Rev. D **92**, 124007 (2015).

[34] J. Qiu and C.-W. Shu, SIAM J. Sci. Comput. **27**, 995 (2005).

[35] M. Dumbser, O. Zanotti, R. Loubere, and S. Diot, J. Comput. Phys. **278**, 47 (2014).

[36] O. Zanotti, F. Fambri, and M. Dumbser, Mon. Not. R. Astron. Soc. **452**, 3010 (2015).

[37] M. Sonntag and C.-D. Munz, in *Finite Volumes for Complex Applications VII—Elliptic, Parabolic and Hyperbolic Problems* (Springer, New York, 2014), pp. 945–953.

[38] A. Huerta, E. Casoni, and J. Peraire, Int. J. Numer. Methods Fluids **69**, 1614 (2012).

[39] G. Gassner and D. A. Kopriva, SIAM J. Sci. Comput. **33**, 2560 (2011).

[40] G.-S. Jiang and C.-W. Shu, J. Comput. Phys. **126**, 202 (1996).

[41] S. Bernuzzi *et al.* (to be published).

[42] R. Arnowitt, S. Deser, and C. W. Misner, in *Gravitation An Introduction to Current Research*, edited by L. Witten (John Wiley, New York, 1962), pp. 227–265.

[43] J. W. York, Jr., in *Sources of Gravitational Radiation*, edited by L. Smarr (Cambridge University Press, Cambridge, 1979), pp. 83–126.

[44] M. Alcubierre, *Introduction to 3+1 Numerical Relativity* (Oxford University Press, Oxford, 2008).

[45] T. W. Baumgarte and S. L. Shapiro, *Numerical Relativity: Solving Einstein's Equations on the Computer* (Cambridge University Press, Cambridge, 2010).

[46] E. Gourgoulhon, *3+1 Formalism in General Relativity* (Springer, Berlin, 2012).

[47] J. A. Font, M. Miller, W. M. Suen, and M. Tobias, Phys. Rev. D **61**, 044011 (2000).

[48] L. Baiotti, I. Hawke, P. J. Montero, F. Löffler, L. Rezzolla, N. Stergioulas, J. A. Font, and E. Seidel, Phys. Rev. D **71**, 024035 (2005).

[49] H. Dimmelmeier, J. A. Font, and E. Müller, Astron. Astrophys. **388**, 917 (2002).

[50] C. Ronchi, R. Iacono, and P. Paolucci, J. Comput. Phys. **124**, 93 (1996).

[51] A. Mignone, P. Tzeferacos, and G. Bodo, J. Comput. Phys. **229**, 5896 (2010).

[52] J. M. Martí and E. Müller, J. Fluid Mech. **258**, 317 (1994).

[53] F. Pretorius, Classical Quantum Gravity **22**, 425 (2005).

[54] M. Alcubierre, S. R. Brandt, B. Brügmann, D. Holz, E. Seidel, R. Takahashi, and J. Thornburg, Int. J. Mod. Phys. **D10**, 273 (2001).

[55] A. Klöckner, T. Warburton, and J. S. Hesthaven, Math. Mod. Nat. Phenom. **6**, 57 (2011).

[56] B. Cockburn and J. Guzman, SIAM J. Numer. Anal. **46**, 1364 (2008).

[57] L. E. Kidder *et al.*, arXiv:1609.00098.

[58] R. Baltensperger and M. R. Trummer, SIAM J. Sci. Comput. **24**, 1465 (2003).

[59] S. A. Teukolsky, J. Comput. Phys. **283**, 408 (2015).

[60] M. H. Carpenter and D. Gottlieb, J. Comput. Phys. **129**, 74 (1996).

[61] G. Gassner, SIAM J. Sci. Comput. **35**, A1233 (2013).

[62] H. Wang, D. Huybrechs, and S. Vandewalle, Math. Comput. **83**, 2893 (2014).