High order well-balanced finite volume methods for multi-dimensional systems of hyperbolic balance laws

Jonas P. Berberich\textsuperscript{a}, Praveen Chandrashekar\textsuperscript{b}, Christian Klingenberg\textsuperscript{a,\ast}

\textsuperscript{a}Dept. of Mathematics, Univ. of Würzburg, Emil-Fischer-Str. 40, 97074 Würzburg, Germany
\textsuperscript{b}TIFR Center for Applicable Mathematics, Bengaluru, Karnataka 560065, India

Abstract

We introduce a general framework for the construction of well-balanced finite volume methods for hyperbolic balance laws. The phrase \textit{well-balancing} is used in a wider sense, since the method can be applied to exactly follow any solution of any system of hyperbolic balance laws in multiple spatial dimensions. The solution has to be known a priori, either as an analytical expression or as discrete data. The proposed framework modifies the standard finite volume approach such that the well-balancing property is obtained. The potentially high order of the method is maintained under the modification. We show numerical tests for the compressible Euler equations with and without gravity source term and with different equations of state, and for the equations of compressible ideal magnetohydrodynamics. Different grid geometries and reconstruction methods are used. We demonstrate high order convergence numerically.

\textit{Keywords:} finite-volume methods, well-balancing, compressible Euler equations with gravity, ideal magnetohydrodynamics, hyperbolic balance laws

1. Introduction

Several problems in engineering and science are modeled by conservation properties. A common approach is to describe their evolution using hyperbolic conservation laws. However, only in few cases can these equations be solved analytically. Numerical methods to approximate solutions are used to forecast the behavior of the system. One successful approach amongst these is the finite volume method, based on Godunov’s method \cite{Godunov}. Finite volume methods introduce discretization errors. Typically, they are only exact on the constant state, some polynomials (in the case of higher order methods), and maybe other numerical stationary states, which coincide with the stationary states of the equations in very few cases.

As soon as external forces enter the modeled system, a source term has to be added to the hyperbolic conservation laws turning these into hyperbolic balance laws. The constant state is not a solution of these anymore. Instead, other static or stationary solutions take its place. However, standard methods are not
able to maintain these solutions accurately but introduce discretization errors. This gives rise to the need for the development of so-called well-balanced methods, i.e. methods which are designed to be exact on special stationary solutions of the system.

In the well-known shallow water equations with non-flat bottom topography, the most widely considered static state, which is the lake-at-rest solution, can be formulated in a closed form. This favors the construction of well-balanced methods for this system. There is a rich literature about well-balanced methods for shallow water equations ([2] [3] [4] [5] and references therein) and related systems like the Ripa model ([6] [7] and references therein). This includes higher order methods for static [8] and non-static stationary states [9]. For the Euler equations with gravitational potential, on the other hand, static solutions have to be found by solving a differential equation for density and pressure. The second equation relating these quantities is the equation of state (EoS). This makes the construction of well-balanced methods much more delicate and typically restricts the resulting method to special cases. Many methods have been developed for some classes of hydrostatic states assuming an ideal gas EoS. Examples are given in [10] [11] [12] [13] [14] [15] [16]. There are also higher order methods, see e.g. [17] [18].

While the hydrostatic equation for compressible Euler equations with gravity is basically one-dimensional, the spatial structure of this relation is much richer for compressible ideal magnetohydrodynamics (MHD) equations with gravity since it includes off-diagonal terms. In [19] a well-balanced method for MHD is derived to compute waves on the stationary background. This method is designed to balance isothermal hydrostatic states of the Euler equations together with a magnetic field, which satisfies certain stationarity conditions and is known a priori. Part of this method, namely considering deviations to a background magnetic field, goes back to to Tanaka [20] and is also used by Powell et al. [21]. To do so, the background magnetic field is assumed to be static and free of as well rotation as divergence.

There are different approaches to obtain the well-balanced property. Some methods are based on a relaxation approach, in which the hydrostatic equation is included in the relaxation system [22] [23] [24]. Another widespread idea is the hydrostatic reconstruction, i.e. reconstruction of variables which are constant if the system is in the considered stationary state. An early example of this method is [2] for the shallow water system. For Euler equations this approach has been used in [25] [26] [10] [12] [13] [17] [16].

The methods for Euler equations mentioned before are restricted to a certain EoS and certain classes of hydrostatic solutions. For astrophysical applications, for example, this restriction is a severe limitation. Equations of state describing physics in the stellar interior are much more complex than the EoS of an ideal gas.
More general methods have been developed in [27, 28, 29, 30]. The well-balanced methods introduced in these publications can be applied for any EoS. They are exact on certain hydrostatic solutions, in all other cases they are exact on a second order approximation of the considered hydrostatic solution. In [26] a second order well-balanced method for Euler equations with gravity is introduced. This method can be applied for any hydrostatic solution of Euler equations with any EoS if the hydrostatic solution is known. The method is then exact up to machine precision. The method is extended to higher order in [17]. Notably, there is also an extension to stationary states with non-zero velocity in the same article.

The method we present in this paper is designed in the manner of the method in [26]. It uses the idea of hydrostatic reconstruction and a modification of the source term discretization to obtain the well-balanced property. The main point in which it differs from all of the methods mentioned above is that our method is not restricted to a certain system of hyperbolic balance laws. Instead, we present a general framework which modifies finite volume methods for any hyperbolic conservation or balance laws such that they obtain the well-balancing property. Also, the method can be used to balance any solution if it is known before and can be given in an analytical expression or as discrete data. Due to this property we use the phrase well-balancing in a broader sense than it is typically used in literature. This reference solution, which is chosen to be well-balanced, can even depend on time, as we will show in this paper.

Our method is also general in the possibility of combination with other modules of a finite volume scheme: It can be applied on any grid system, with any numerical flux function, reconstruction routine, source term discretization, and ODE solver for time-discretization. It allows for higher order in the sense that, if all these components are high order accurate, the resulting method is also high order accurate.

There are several applications in which the hydrostatic solution is given. Consider the application of stellar astrophysics; there, the EoS is often given in the form of a table since there is no analytical expression. Consequently, hydrostatic solutions which depend on the EoS can only be found numerically and given in the form of discrete data. While methods which incorporate analytical expressions are not able to exactly maintain these hydrostatic solutions, it is very well possible with the methods in [26, 17] and the method we present in this paper. Especially, if we consider the better approximation of stellar structure which is given by a stationary state including rotation, our method can be applied to maintain this stationary solution. Another example from astrophysical application are rotating Keplerian disks. These are two-dimensional disks of matter which follows Newton’s laws of motion in the gravitational field of a massive attractor. One way to describe this disk is a stationary solution of Euler equations with gravity including non-zero velocities. Since this is not a hydrostatic solution, conventional well-balanced methods can not preserve this solution. A special method designed for this application is introduced in [31].
this paper we will show that our method is also able to preserve this solution on different grids. Besides
the applicability to any system of hyperbolic balance laws, the balancing of moving and time-depending
solutions is one of the key features of our method.

The rest of the paper is structured as follows: In Section 2 we introduce the standard finite volume
framework for systems of hyperbolic conservation laws in three spatial dimensions on arbitrary grids. In
Section 3 we introduce our general well-balanced modification for this framework. The well-balanced
property we claim for our method is then shown in Section 4. The validity of the well-balanced property
also depends on a consistent choice of boundary conditions. Therefore, we add a discussion about well-
balanced boundary conditions in Section 6. In Section 7 we add some remarks regarding our method con-
cerning structure, accuracy, and the range of application. To emphasize how simple it is to add our method
to an existing finite volume code, we comment on the implementation of the method in Section 8. Finally,
in Section 9 we show a variety of numerical tests. These range from applications on Euler equations to
ideal magnetohydrodynamics (MHD) equations. They include classical well-balanced tests on the balance
laws and also test on the homogeneous hyperbolic conservation laws. Different equations of state are used
for the Euler equations. We include a test in which the well-balanced solution is not analytically known
but has been obtained numerically. Also, we present tests in which the well-balanced solution depends on
time. We verify higher order accuracy for solutions close to and far away from the well-balanced solution
numerically. To show the efficiency of the method, we present CPU time comparisons of simulations with
and without the well-balanced modification in Section 10.

2. A standard finite volume method

In this section we present the standard higher order finite volume framework for three-dimensional
hyperbolic balance laws. There is a rich literature on these methods, e.g. [32, 33].

Consider the 3-d system of hyperbolic balance laws
\[ \partial_t q(x, t) + \nabla \cdot F(q(x, t)) = s(q(x, t), x, t) \]  
(1)

with \( F = (f_1, f_2, f_3) \), where \( f_l \) is the flux in \( l \)-direction. Using any mesh we divide the domain into \( N \) control
volumes. For the \( i \)-th control volume \( \Omega_i \) \( (i \in \{1, \ldots, N\}) \) we define the cell-average
\[ Q_i(t) := \frac{1}{V_i} \int_{\Omega_i} q(x, t) dx, \]  
(2)
where \( V_i = |\Omega_i| \) is the control cell volume. Integrating Eq. (1) over \( \Omega_i \) and applying the divergence theorem yields the evolution equation for \( Q_i \)

\[
\frac{d}{dt} Q_i(t) + \frac{1}{V_i} \int_{\partial \Omega_i} \mathcal{F}(q(x, t)) \cdot n(x) d\sigma = \frac{1}{V_i} \int_{\Omega_i} s(q(x, t), x) dx.
\]  

(3)

An equivalent formulation which is useful for the following discretization is

\[
\frac{d}{dt} Q_i(t) = -\frac{1}{V_i} \sum_{k \in NC_i} \int_{\partial \Omega_{ik}} \mathcal{F}(q(x, t)) \cdot n(x) d\sigma + \frac{1}{V_i} \int_{\Omega_i} s(q(x, t), x, t) dx,
\]  

(4)

where \( NC_i \) is the set of indexes of all control volumes sharing an interface with \( \Omega_i \). For the discretization of the interface fluxes we use a numerical flux function \( F(\cdot, \cdot, n) \) consistent with \( n \cdot \mathcal{F} \). The consistency conditions are Lipschitz continuity in the first two arguments and the relation \( F(q, q, n) = n \cdot \mathcal{F}(q) \) for all normalized vectors \( n \). We apply this discretization to Eq. (4) and obtain

\[
\frac{d}{dt} Q_i(t) = -\frac{1}{V_i} \sum_{k \in NC_i} \int_{\partial \Omega_{ik}} F \left( Q_{i,rec}^q(x, t), Q_{k,rec}^q(x, t), n(x) \right) d\sigma + \frac{1}{V_i} \int_{\Omega_i} s(Q_{i,rec}^q(x, t), x, t) dx,
\]  

(5)

where the reconstructed functions \( Q_{i,rec}^q, Q_{k,rec}^q \) are obtained using a consistent conservative reconstruction routine on the cell average values \( Q \). Examples for popular consistent conservative reconstruction routines can be found in [34, 35, 36, 33] and simple methods in the appendix. In the next step we use numerical quadrature rules for the interface flux integral and a discretization of the source term integral. The semi-discrete method is then

\[
\frac{d}{dt} Q_i(t) = -\frac{1}{V_i} \sum_{k \in NC_i} \left( \sum_{j=1}^{M} \omega_j F \left( Q_{i,rec}^j(x_{ikj}, t), Q_{k,rec}^j(x_{ikj}, t), n(x_{ikj}) \right) \right) + \frac{1}{V_i} I_{\Omega_i} \left[ s(Q_{i,rec}^q, x, t) \right].
\]  

(6)

\( M \) is the number of quadrature points at the interfaces, \( x_{ikj} \) are the \( M \) quadrature points at the \( ik \) interface and \( \omega_j \) are the corresponding weights. The symbol \( I_{\Omega_i} \) denotes a consistent discretization of the integral over the argument in the domain \( \Omega \). The quadrature rules and source term discretizations we use in our tests are given in the appendix.

The semi-discrete scheme Eq. (6) is \( k \)-th order accurate if the applied reconstruction routine, interface flux quadrature and source term discretization are all at least \( k \)-th order accurate. It can then be evolved in time using a \( k \)-th order accurate ODE solver to obtain a \( k \)-th order accurate fully discrete scheme.

3. The well-balanced modification of the standard finite volume method

In this section we will introduce a well-balanced modification for the three-dimensional finite volume method presented in Section[2]. Reducing it to one or two spatial dimensions is straightforward.
Let \( \tilde{q} \) be a given continuous solution of Eq. (1). Plugging this reference solution \( \tilde{q} \) into Eq. (3) we get
\[
\frac{d}{dt} \mathbf{Q}(t) + \frac{1}{V_i} \int_{\partial \Omega_i} \mathbf{F}(\tilde{q}(x, t)) \cdot \mathbf{n} \, d\sigma = \frac{1}{V_i} \int_{\Omega_i} \mathbf{s}(\tilde{q}(x, t), x) \, dx,
\]
where \( \mathbf{Q} \) is the average of the reference solution \( \tilde{q} \) in the \( i \)-th control volume. In the next step we subtract Eq. (7) from Eq. (3) to obtain
\[
\frac{d}{dt} \mathbf{Q}(t) - \frac{d}{dt} \mathbf{\tilde{Q}}(t) + \frac{1}{V_i} \int_{\partial \Omega_i} (\mathbf{F}(\mathbf{q}(x, t)) - \mathbf{F}(\tilde{q}(x, t))) \cdot \mathbf{n} \, d\sigma = \frac{1}{V_i} \int_{\Omega_i} \mathbf{s}(\mathbf{q}(x, t), x, t) - \mathbf{s}(\tilde{q}(x, t), x, t) \, dx.
\]
Now, let us rewrite Eq. (8) in terms of the deviation from the reference solution
\[
\Delta \mathbf{q} := \mathbf{q} - \tilde{\mathbf{q}}, \quad \Delta \mathbf{Q} := \mathbf{Q} - \tilde{\mathbf{Q}}.
\]
This yields
\[
\frac{d}{dt} (\Delta \mathbf{Q}(t)) = -\frac{1}{V_i} \sum_{k \in NC_i} \int_{\partial \Omega_k} (\mathbf{F}(\Delta \mathbf{q} + \tilde{\mathbf{q}})(x, t)) - \mathbf{F}(\tilde{\mathbf{q}}(x, t))) \cdot \mathbf{n} \, d\sigma \\
+ \frac{1}{V_i} \int_{\Omega_i} \mathbf{s}(\Delta \mathbf{q} + \tilde{\mathbf{q}})(x, t, x, t) - \mathbf{s}(\tilde{\mathbf{q}}(x, t), x, t) \, dx,
\]
where \( NC_i \) is the set of indexes of all control volumes sharing an interface with \( \Omega_i \). At this point, we start to discretize. For that we define a numerical flux difference approximation
\[
\Delta \tilde{\mathbf{F}}(\Delta \mathbf{Q}^L, \Delta \mathbf{Q}^R, \tilde{\mathbf{q}}, \mathbf{n}) := \mathbf{F}(\Delta \mathbf{Q}^L + \tilde{\mathbf{q}}, \Delta \mathbf{Q}^R + \tilde{\mathbf{q}}, \mathbf{n}) - \mathbf{n} \cdot \mathbf{F}(\tilde{\mathbf{q}}) \approx \mathbf{n} \cdot (\mathbf{F}(\Delta \mathbf{q} + \tilde{\mathbf{q}}) - \mathbf{F}(\tilde{\mathbf{q}))},
\]
where \( \mathbf{F}(\cdot, \cdot, \mathbf{n}) \) is a numerical flux function consistent with \( \mathbf{n} \cdot \mathbf{F} \). We apply this discretization to Eq. (10) and obtain
\[
\frac{d}{dt} (\Delta \mathbf{Q}(t)) = -\frac{1}{V_i} \sum_{k \in NC_i} \int_{\partial \Omega_k} \Delta \tilde{\mathbf{F}}(\Delta \mathbf{Q}_i^{\text{rec}}(x, t), \Delta \mathbf{Q}_k^{\text{rec}}(x, t), \tilde{\mathbf{q}}(x, t), \mathbf{n}(x)) \, d\sigma \\
+ \frac{1}{V_i} \int_{\Omega_i} \mathbf{s}(\Delta \mathbf{Q}^{\text{rec}} + \tilde{\mathbf{q}})(x, t, x, t) - \mathbf{s}(\tilde{\mathbf{q}}(x, t), x, t) \, dx,
\]
where the reconstructed functions \( \Delta \mathbf{Q}_i^{\text{rec}}, \Delta \mathbf{Q}_k^{\text{rec}} \) are obtained using a consistent conservative reconstruction routine on the cell average values \( \Delta \mathbf{Q} \). In the next step we use numerical quadrature rules for the interface flux integral and a discretization of the source term integral. The semi-discrete method is then
\[
\frac{d}{dt} (\Delta \mathbf{Q}(t)) = -\frac{1}{V_i} \sum_{k \in NC_i} \sum_{j=1}^{M} \omega_j \Delta \tilde{\mathbf{F}}(\Delta \mathbf{Q}_i^{\text{rec}}(x_{ik}, t), \Delta \mathbf{Q}_k^{\text{rec}}(x_{ik}, t), \tilde{\mathbf{q}}(x_{ik}, t), \mathbf{n}(x_{ik})) \\
+ \frac{1}{V_i} I_{\text{ext}} \mathbf{s}(\Delta \mathbf{Q}^{\text{rec}} + \tilde{\mathbf{q}}(x, t), x, t) - \frac{1}{V_i} I_{\text{ext}} \mathbf{s}(\tilde{\mathbf{q}}(x, t), x, t).
\]
where the notations used are as in Eq. (6).
As in the standard method, this semi-discrete scheme Eq. (13) is $k$-th order accurate if the applied reconstruction routine, interface flux quadrature and source term discretization are all at least $k$-th order accurate. It can then be evolved in time using a $k$-th order accurate ODE solver to obtain a $k$-th order accurate fully discrete scheme.

4. Proof of the well-balanced property

In this section we show the well-balanced property of our method.

**Theorem 4.1.** The modified finite volume method introduced in Section 3 satisfies the following property:

If

$$\Delta Q_i = 0 \quad \forall i \in \{1, \ldots, N\} \quad (14)$$

at initial time, then this holds for all $t > 0$.

**Proof:** Let $\Delta Q_i = 0$ for all $i \in I$. The consistency of the applied reconstruction leads to $\Delta Q_i^{rec} \equiv 0$ at all flux quadrature points. The flux consistency then yields

$$\Delta F(\Delta Q^L, \Delta Q^R, \bar{q}, n) = \Delta F(0, 0, \bar{q}, n) = n \cdot \mathcal{F}(\bar{q}) - n \cdot \mathcal{F}((\bar{q}) = 0. \quad (15)$$

Now, consider the contribution from the source term: With $\Delta Q_i = 0$ the source term discretization in Eq. (13) reduces to

$$I_{x \in \Omega} \left[ s((\Delta Q_i^{rec} + \bar{q})(x, t), x, t) \right] - I_{x \in \Omega} \left[ s(\bar{q}(x, t), x, t) \right] = I_{x \in \Omega} \left[ s(\bar{q}(x, t), x, t) \right] - I_{x \in \Omega} \left[ s(\bar{q}(x, t), x, t) \right] = 0. \quad (16)$$

We have shown that the right hand side in Eq. (13) vanishes and thus the initial data $\Delta Q_i = 0$ are conserved for all time.

In Theorem 4.1 the formulation of the well-balanced property is quite simple and maybe not intuitive. If we formulate the result in terms of the actual solution, it might read like this:

**Corollary 4.2.** If the initial condition $Q_i(t = 0), i = 1, \ldots, N$, equals the cell averages of the reference solution $\bar{Q}_i(t = 0), i = 1, \ldots, N$, the computed solution equals the reference solution for all time.

5. The treatment of discrete reference solutions

It is not necessary that the reference solution in the well-balanced modification introduced in Section 3 has to be known analytically. It can also be given in the form of discrete data. For consistency with the system of balance laws Eq. (1) it is important that the discrete data which are used for the reference solution
converge to a solution of Eq. (1) when the computational grid is refined. To ensure the high order of our method, this convergence should also be of high order. When only discrete data are given we need a method to compute values at the interface quadrature points and cell-averaged values for the grid on which we use our well-balanced method. In this section we will describe how this can be done. Depending on the form in which the discrete reference data are given, the smoothness of the data, and the required order of accuracy, different methods for this reconstruction have to be used. Here, we will give two examples.

5.1. Example 1: pointwise 1-d data on a fine grid

One application of our method is well-balancing hydrostatic solutions of Euler equations. Especially in physical applications with complex EoS hydrostatic solutions have to be obtained by numerical methods. Even for multi-dimensional simulations, the underlying hydrostatic solution can be one-dimensional in its nature (see cases (a) and (b) below). Now assume such a numerically approximated hydrostatic solution in one spatial dimension is given in the form of point values \( q_{hs}^i, i = 1, 2, \ldots, L \) (in conservative variables) on a fine equidistant grid. Assume it is supposed to be used in a two-dimensional third order accurate modified finite volume method as introduced in Section 3 on a Cartesian grid. For that, in a first step, we use a cubic spline interpolation to construct a continuous function \( q_{hs}^{1-d}(x) \) (e.g. [37]). This function is then extended to two spatial dimensions. How this is done depends on the symmetry of the 2-d problem which allowed the reduction to a 1-d hydrostatic solution. We consider two different cases:

(a) Suppose we have an essentially 1-d hydrostatic solution where gravitational force is at an angle \( \alpha \in [0, 2\pi] \) to x-axis. Then we extend the one-dimensional hydrostatic solution \( q_{hs}^{1-d} \) to a two-dimensional solution via \( q^{hs}(x) := q_{hs}^{1-d}(x \cos(\alpha) + y \sin(\alpha)) \).

(b) Suppose we have a radial hydrostatic solution with a gravity vector pointing towards the center \( x_{\text{center}} \). In that case we extend the hydrostatic solution back to 2-d by setting \( q^{hs}(x) := q_{hs}^{1-d}(||x - x_{\text{center}}||) \).

The values for the reference solution at interface quadrature points can then be evaluated pointwise as \( \tilde{q}(x, t) = q^{hs}(x) \). The cell average values of the reference solution are computed using the third order accurate 2-d quadrature rule Eq. (A.11). Method (a) is used in the test in Section 9.1.3.

5.2. Example 2: reference solution from a highly resolved finite volume simulation

Consider a reference solution given as numerical solution of a finite volume simulation on a two-dimensional structured static grid with curvilinear coordinates as described in Appendix A.1. In this case the data are given as cell averages \( \hat{Q}_{ij}^{n} \), instead of point values. Assume this reference solution is supposed to be used in a second order accurate two-dimensional modified finite volume method as described in
The grid to be used is a coarser version of the grid on which the reference solution has been computed, such that all interfaces on the coarse grid coincide with interfaces of the fine grid. For each time step \( t^n \) of the stored reference data we map the fine grid on the coarse grid with

\[
\tilde{Q}_{ij}(t^n) = \frac{1}{\tilde{V}_{ij}} \sum_{k,l \in \tilde{Q}_{ij}} \tilde{V}_{kl} \tilde{Q}_{kl}^{n},
\]

where all quantities with \( \tilde{\cdot} \) correspond to the coarse grid and all quantities with \( \hat{\cdot} \) correspond to the fine grid. The values of \( \tilde{Q}_{ij} \) at intermediate times are obtained via interpolation in time. Note, that the interpolation has to be conservative in the sense that

\[
\sum_{(i,j) \in S} \tilde{V}_{ij} \tilde{Q}_{ij}(t^n) = \sum_{(i,j) \in S} \tilde{V}_{ij} \tilde{Q}_{ij}(t^n+1),
\]

for any \( t \in [t^n, t^{n+1}] \) if \( \sum_{(i,j) \in S} \tilde{V}_{ij} \tilde{Q}_{ij}(t^n+1) = \sum_{(i,j) \in S} \tilde{V}_{ij} \tilde{Q}_{ij}(t^n) \). This has to hold for any subset \( S \) of the set of grid indexes. We choose linear interpolation in time, i.e.

\[
\tilde{Q}_{ij}(t) = \frac{t^{n+1} - t}{\rho_{n+1} - \rho_n} \tilde{Q}_{ij}(t^n) + \frac{t - t^n}{\rho_{n+1} - \rho_n} \tilde{Q}_{ij}(t^{n+1}), \quad \text{for} \quad t \in (t^n, t^{n+1}),
\]

which satisfies this property. Note that this conservation property poses a severe restriction for higher order in time interpolation. For a second order method only one interface quadrature point at the center of the interface is needed. For that we can -in the computational coordinates- just interpolate linearly from the cell-averages interpreted as second order approximation to the cell-centered values. Thus we have

\[
\tilde{q}(x_{i+\frac{1}{2}}, j, t) = \frac{1}{2} (\tilde{Q}_{ij}(t) + \tilde{Q}_{i+1,j}(t)) \quad \text{and} \quad \tilde{q}(x_{i, j+\frac{1}{2}}, t) = \frac{1}{2} (\tilde{Q}_{ij}(t) + \tilde{Q}_{i,j+1}(t)).
\]

### 6. Boundary conditions

In the previous sections (including the proof of the well-balanced property) we omitted to include boundary conditions in the discussion. Yet, the validity of the well-balanced property also depends on the correct choice of boundary conditions. In this section we will describe boundary conditions which have all of the following properties:

- They are compatible with the well-balancing property.
- They support the potentially high order of the scheme.
- They have relevance for actual application.

Some of the proposed numerical boundary conditions require knowledge of the reference solution outside the domain. If this is not given, one can simply extrapolate the reference solution to the ghost cells. This will not affect the well-balanced property nor order of accuracy, if the extrapolation is done with a sufficiently high order.
**Extrapolation boundary conditions:** One way to treat boundaries is the extrapolation of data in the domain to ghost cells. This can be done with high order to support the high order of the applied scheme. In our method we extrapolate the deviations $\Delta Q$. In the case that $\Delta Q_i = 0$ holds for all control volumes in the domain, this will also be true for the extrapolated states. Hence, the well-balanced property also holds at the boundary. Extrapolation boundary conditions for one and two spatial dimensions with different orders of accuracy can be found in Appendix A.2.

**Wall boundary conditions:** To mimic the effect of a solid wall in a finite volume simulation, the values in the cells next to the boundary are copied to ghost cells outside the boundary. The normal velocity component switches the sign. Assuming that the reference solution is compliant to the boundary conditions, we can simply apply wall boundary conditions to the deviation average states $\Delta Q$ in our well-balanced method.

**Periodic boundary conditions:** In the case of our method the deviations to the reference states are communicated to the opposite side of the domain. Periodic boundary conditions are described in Appendix A.2.

### 7. Remarks on the well-balanced method

**Remark 7.1.** If a stationary solution is chosen as reference solution (which is the case for classical well-balancing applications), the time derivative of the reference solution vanishes by definition. This leads to $\frac{d}{dt}Q_i = \frac{d}{dt}(\Delta Q_i)$. The described method can then also be used to directly evolve the $Q_i$ in time instead of $\Delta Q_i$. For that the scheme can be adapted by just substituting all $\Delta Q$ terms with the corresponding $Q - \tilde{Q}$ terms. The reconstruction has then to be applied on the states $Q - \tilde{Q}$.

**Remark 7.2.** Note that

$$s((\bar{q} + \Delta q)(t, x), x, t) - s((\bar{q}(t, x), x, t)) = s(\Delta q(t, x), x, t)$$  \hspace{1cm} (19)$$

if the source term $s$ in Eq. (1) is linear in the first argument. Due to the linearity of the corresponding source term discretizations, this relation then also holds for the discretized source terms. Naming examples, this is the case for the gravitational source term in Euler or ideal MHD equations and the bottom topography source term in the shallow water equations.

**Remark 7.3.** One argument for high order accuracy in the finite volume framework is the polynomial accuracy. We can say that a method is $k$-th order, if reconstruction, interface flux quadrature, source term discretization, and boundary condition are exact on $k$-th order polynomials. Our method has this property.
for the deviation \( \Delta Q \) to the reference solution. Since the reference solution \( \tilde{q} \) need not to be a polynomial, this does not translate to \( Q = \tilde{Q} + \Delta Q \). However, because of
\[
\|Q - Q^{\text{exact}}\| = \|\Delta Q + \tilde{Q} - \Delta Q^{\text{exact}} - \tilde{Q}\| = \|\Delta Q - \Delta Q^{\text{exact}}\|
\]
(20)
for any discrete norm the method is also \( k \)-th order accurate in \( Q \) (assuming that as well \( \tilde{q} \) as \( \Delta q^{\text{exact}} \) are sufficiently smooth). The notations \( Q^{\text{exact}} \) and \( \Delta Q^{\text{exact}} \) denote the cell-average of the exact analytical solutions opposed to the cell-averages of the numerically approximated solutions \( Q \) and \( \Delta Q \).

The accuracy will also be demonstrated numerically in Section 9. In the following two Remarks 7.4 and 7.5 we discuss the scope of application of our method.

**Remark 7.4.** Our well-balanced method can even be beneficially applied if there is no source term. Applications could include stationary solutions based on vorticity in multi-dimensional simulations. Corresponding numerical tests are presented in this paper.

**Remark 7.5.** We want to emphasize that the reference solution in our well-balanced method can very well be time-dependent. This does not change the consistency, accuracy, or the well-balanced property formulated in Theorem 4.1. Thus, we use the phrase “well-balancing” in a wider sense than it is typically used.

8. Notes on the implementation

We have seen that our well-balanced method can be applied for a wide range of problems. In this section we will discuss another property useful in application: The method is also easy to implement. This holds especially if there is an existing finite volume code which shall be modified to obtain a well-balancing capability. In a typical finite volume code the following changes have to be applied:

1. Implement a function returning \( \tilde{q} \) or an array carrying the cell averages and values at flux quadrature points if \( \tilde{q} \) is time-independent. In the second case \( \tilde{q} \) can just be set alongside the initial condition.

2. After the program received the initial data for the \( Q_i \), it should transform to the deviations \( \Delta Q_i = Q_i - \tilde{Q}_i \).

3. In the routine evaluating the numerical flux, compute \( \Delta \tilde{F} (\Delta Q^L, \Delta Q^R, \tilde{q}, n) \) instead of \( n \cdot \tilde{F} (\Delta q + \tilde{q}) \). Basically, this just means subtracting the exact flux after evaluation of the numerical flux.

4. This step is only necessary if the source term is not linear in \( q \) (see Remark 7.2): Evaluate the source term at the states \( \tilde{Q}_i + \Delta Q_i \) and \( \tilde{Q}_i \). The difference of these source terms is added to the residual.
5. Do not forget to correct the output routine such that the states \( \mathbf{Q}_i = \mathbf{\tilde{Q}}_i + \Delta \mathbf{Q}_i \) are written to the output.

Let us remind of Remark 7.1 and point out that an alternative implementation could also evolve \( \mathbf{q} \) instead of \( \Delta \mathbf{q} \) in time if \( \mathbf{\tilde{q}} \) is time-independent. In some codes this can be the easier way.

9. Numerical tests of the scheme

We test the method we proposed using a simple python finite volume code. It is built in a modular way, such that different methods can be applied.

**Grids:** Two-dimensional discretization is realized using a structured grid. In some tests we use curvilinear grids. The implementation of these grids restricts the total method to second order accuracy. Note that this is not a general statement. This restriction is due to the special implementation of the grid in our code. Higher order can only be achieved with a Cartesian grid in our code.

**Numerical flux function:** As numerical flux function we use the Lax–Friedrichs flux [32], since it is simple and can be applied for any hyperbolic system. In some tests we use the Roe’s approximate Riemann solver for Euler equations [38] to obtain more accurate results.

**First order method:** To formally obtain a first order method we use constant reconstruction to obtain the interface values. The numerical fluxes are computed at the center of the interfaces. The source term is evaluated at the cell-center. For the gravity source term used in our tests, we need the gradient \( \nabla \phi \) of the given gravitational potential. This gradient is computed exactly at the cell-center.

**Second order method:** To formally obtain a second order method we use a conservative linear reconstruction to obtain the interface values. This is the only difference to the first order method.

**Third order method:** To formally obtain a third order method we use a conservative parabolic reconstruction to obtain the interface values. In the two-dimensional case this is realized using the \( P_{\text{opt}} \) polynomial from [36]. In the two-dimensional case, the numerical fluxes are evaluated at the Gauss–Legendre quadrature points. The source term is evaluated by exactly integrating a parabolic interpolation polynomial. We interpolate from cell-centered states which are obtained via conservative parabolic reconstruction from the cell-average states.

**Seventh order method:** To formally obtain a one-dimensional seventh order method we use a conservative \( P_6 \) polynomial reconstruction to obtain the interface and cell-centered values. On the cell-centered values multiplied with the point-wise gravity acceleration we interpolate the source term using a \( P_6 \) polynomial and integrate exactly.

**Boundary conditions:** If the setup has periodic character we use periodic boundary conditions. Otherwise we extrapolate the states to ghost cells with a sufficiently high spatial order. If we use the third order
method, for example, we extrapolate using parabolas.

Time-stepping: We evolve the semi-discrete scheme in time using an explicit third order four stage Runge–Kutta method from [39]. For the seventh order method we use the explicit tenth order 17 stage Runge–Kutta method from [40].

For brevity we only give a short description of the methods in this place. The interested reader can find details of the methods in Appendix A. Note, that we use simple and straight forward methods for the tests in this section. However, our well-balanced method is not restricted to these. Instead, one can also use more sophisticated methods, like e.g. WENO-type methods for reconstruction or numerical flux functions designed for special problems (e.g. a low Mach number compliant method for Euler equations like in [41]).

9.1. Euler equations with gravity

The 2-d compressible Euler equations which model the balance laws of mass, momentum, and energy under the influence of gravity are given by

$$\frac{\partial}{\partial t} \mathbf{q} + \frac{\partial}{\partial x} \mathbf{f} + \frac{\partial}{\partial y} \mathbf{g} = \mathbf{s},$$

(21)

where the conserved variables, fluxes and source terms are

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ (E + p)u \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ (E + p)v \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ -\rho \partial_x \phi \\ -\rho \partial_y \phi \\ 0 \end{bmatrix},$$

(22)

with $\rho, p > 0$. Moreover, $E = \rho \varepsilon + \frac{1}{2} \rho |v|^2 + \rho \phi$ is the total energy per unit volume with the velocity $v = (u, v)^T$ and specific internal energy $\varepsilon$. The scalar function $\phi$ is a given gravitational potential. An additional relation between density, pressure, and specific internal energy is given in the form of an equation of state (EoS).

In our tests we will use the ideal gas EoS

$$p = (\gamma - 1) \rho \varepsilon$$

(23)

with $\gamma = 1.4$, although our well-balanced method can be applied for Euler equations with any EoS.

The 2-d Euler equations can be reduced to 1-d Euler equations by setting $\mathbf{g} = 0$ and removing the $\rho v$ equation. It can be reduced to homogeneous Euler equations by setting $\mathbf{s} = 0$.

9.1.1. 1-d isothermal hydrostatic solution

We consider an isothermal hydrostatic solution of the 1-d compressible Euler equations with gravitational source term and the ideal gas equation of state given by

$$\phi(x) = \sin(2\pi x), \quad \tilde{\rho}(x) = \tilde{\rho}(x) = \exp(-\phi(x)), \quad \tilde{u} \equiv 0.$$  

(24)
Table 1: $L^1$-errors for an isothermal hydrostatic solution of the Euler equations after time $t = 2$ computed using different methods. The setup is described in Section 9.1.1.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\rho$ error</th>
<th>$\rho u$ error</th>
<th>$E$ error</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard, first order</td>
<td>1.19156249e-01</td>
<td>2.17623832e-02</td>
<td>1.63898613e-01</td>
</tr>
<tr>
<td>well-balanced, first order</td>
<td>0.00000000e+00</td>
<td>0.00000000e+00</td>
<td>0.00000000e+00</td>
</tr>
<tr>
<td>standard, second order</td>
<td>5.38535901e-04</td>
<td>3.05814377e-05</td>
<td>7.59110453e-04</td>
</tr>
<tr>
<td>well-balanced, second order</td>
<td>0.00000000e+00</td>
<td>0.00000000e+00</td>
<td>0.00000000e+00</td>
</tr>
<tr>
<td>standard, third order</td>
<td>3.57840305e-04</td>
<td>2.24036794e-05</td>
<td>4.93235762e-04</td>
</tr>
<tr>
<td>well-balanced, third order</td>
<td>0.00000000e+00</td>
<td>0.00000000e+00</td>
<td>0.00000000e+00</td>
</tr>
<tr>
<td>standard, seventh order</td>
<td>1.49009288e-08</td>
<td>1.22670251e-10</td>
<td>1.49174239e-08</td>
</tr>
<tr>
<td>well-balanced, seventh order</td>
<td>0.00000000e+00</td>
<td>0.00000000e+00</td>
<td>0.00000000e+00</td>
</tr>
</tbody>
</table>

We set these data on a 1-d grid with 128 grid cells on the domain $[0, 1]$. These initial data are evolved up to the final time $t = 2$ using the first, second, and third order method with the standard method and the well-balanced method each. In the well-balanced method we set the initial data Eq. (24) as time-independent reference solution. The $L^1$-errors at final time compared to the initial grid can be seen in Table 1. We see that there is no error when the well-balanced method is applied.

Remark 9.1. To the reader, especially the reader with experience in well-balanced methods, it might seem unusual and unconvincing that the error is actually zero. Most implementations of well-balanced methods still produce a machine error, even if the discretization error is eliminated in the well-balanced method. One reason for that is that typically the source term is balanced against the fluxes. Since the fluxes and source terms are computed in an inherently different way, the difference of these is not exactly zero but some value close to machine precision. In our method, we balance fluxes against fluxes and source term against source term. Thus, the differences can cancel out exactly. If this is the case depends on the specific implementation.

9.1.2. 1-d isothermal hydrostatic solution with perturbation

We add a perturbation to the pressure such that our initial conditions are

$$
\begin{align*}
\rho(x) &= \tilde{\rho}(x), \\
u(x) &= \tilde{u}(x), \\
p(x) &= \tilde{p}(x) + \eta \exp\left(-100 \left(x - \frac{1}{2}\right)^2\right) 
\end{align*}
$$

in the domain $[0, 1]$. We choose $\eta = 10^{-3}$ to test the convergence of our method. We evolve this initial setup up to time $t = 0.2$ using our well-balanced method (first to third order and seventh order). The
results and convergence rates are shown in Table 2. As a reference solution we use a numerical solution computed with the seventh order standard scheme on a grid with 4096 cells. All convergence rates match our expectations. The convergence rate for the seventh order scheme drops in the last step, since the error approaches machine precision. In Figs. 1 and 2, density deviations at time $t = 0.2$ for the test with $\eta = 10^{-5}$ are shown. In Fig. 1 we see, that the discretization error on the hydrostatic background dominates the total error when the second order standard method is used with a low resolution. For higher resolutions, the perturbation can be resolved correctly. In Fig. 2 on the other hand, it gets evident that the second order well-balanced method is capable of correctly resolving the perturbation on a coarse grid. This is due to the fact that there are no discretization errors on the hydrostatic background as we have already seen in Section 9.1.1.

Figure 1: Perturbation on a hydrostatic atmosphere. The test setup is described in Section 9.1.2. The density deviation from the hydrostatic background is shown at time $t = 0.5$ for the perturbation $\eta = 1 \times 10^{-5}$. The second order (if not stated explicitly) standard method is used with different resolutions.

9.1.3. 2-d numerically approximated hydrostatic solution

In stellar astrophysical applications, the hydrostatic state of the star can often be given in a discrete form. In this test we will show that our well-balanced method can be used if the reference solution is given in the form of discrete data in a table.

We assume the following given data: Let the gravitational potential be $\phi(x) = \phi(x, y) = x + y$ and the temperature $T(x) = 1 - 0.1\phi(x)$. We assume an ideal gas with radiation pressure, which satisfies the EoS

$$p = \rho T + T^4,$$

(26)
Table 2: $L^1$-errors and convergence rates for a small pressure perturbation on an isothermal hydrostatic solution of the Euler equations after time $t = 0.2$. Different standard and well-balanced methods are used. The setup is described in Section 9.1.2.

### First order well-balanced method

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\rho$ error</th>
<th>$\rho$ rate</th>
<th>$\rho u$ error</th>
<th>$\rho u$ rate</th>
<th>$E$ error</th>
<th>$E$ rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>2.61e-03</td>
<td>–</td>
<td>2.05e-03</td>
<td>–</td>
<td>5.73e-03</td>
<td>–</td>
</tr>
<tr>
<td>512</td>
<td>1.37e-03</td>
<td>0.9</td>
<td>1.10e-03</td>
<td>0.9</td>
<td>3.08e-03</td>
<td>0.9</td>
</tr>
<tr>
<td>1024</td>
<td>7.05e-04</td>
<td>1.0</td>
<td>5.72e-04</td>
<td>0.9</td>
<td>1.60e-03</td>
<td>0.9</td>
</tr>
<tr>
<td>2048</td>
<td>3.57e-04</td>
<td>1.0</td>
<td>2.92e-04</td>
<td>1.0</td>
<td>8.15e-04</td>
<td>1.0</td>
</tr>
<tr>
<td>4096</td>
<td>1.80e-04</td>
<td>1.0</td>
<td>1.48e-04</td>
<td>1.0</td>
<td>4.11e-04</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### Second order well-balanced method

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\rho$ error</th>
<th>$\rho$ rate</th>
<th>$\rho u$ error</th>
<th>$\rho u$ rate</th>
<th>$E$ error</th>
<th>$E$ rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>2.01e-05</td>
<td>–</td>
<td>2.28e-05</td>
<td>–</td>
<td>5.98e-05</td>
<td>–</td>
</tr>
<tr>
<td>512</td>
<td>4.82e-06</td>
<td>2.1</td>
<td>5.63e-06</td>
<td>2.0</td>
<td>1.49e-05</td>
<td>2.0</td>
</tr>
<tr>
<td>1024</td>
<td>1.18e-06</td>
<td>2.0</td>
<td>1.40e-06</td>
<td>2.0</td>
<td>3.73e-06</td>
<td>2.0</td>
</tr>
<tr>
<td>2048</td>
<td>2.93e-07</td>
<td>2.0</td>
<td>3.51e-07</td>
<td>2.0</td>
<td>9.36e-07</td>
<td>2.0</td>
</tr>
<tr>
<td>4096</td>
<td>7.31e-08</td>
<td>2.0</td>
<td>8.77e-08</td>
<td>2.0</td>
<td>2.34e-07</td>
<td>2.0</td>
</tr>
</tbody>
</table>

### Third order well-balanced method

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\rho$ error</th>
<th>$\rho$ rate</th>
<th>$\rho u$ error</th>
<th>$\rho u$ rate</th>
<th>$E$ error</th>
<th>$E$ rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>3.76e-06</td>
<td>–</td>
<td>3.75e-06</td>
<td>–</td>
<td>1.11e-05</td>
<td>–</td>
</tr>
<tr>
<td>512</td>
<td>4.71e-07</td>
<td>3.0</td>
<td>4.71e-07</td>
<td>3.0</td>
<td>1.39e-06</td>
<td>3.0</td>
</tr>
<tr>
<td>1024</td>
<td>5.90e-08</td>
<td>3.0</td>
<td>5.90e-08</td>
<td>3.0</td>
<td>1.75e-07</td>
<td>3.0</td>
</tr>
<tr>
<td>2048</td>
<td>7.37e-09</td>
<td>3.0</td>
<td>7.38e-09</td>
<td>3.0</td>
<td>2.18e-08</td>
<td>3.0</td>
</tr>
<tr>
<td>4096</td>
<td>9.21e-10</td>
<td>3.0</td>
<td>9.23e-10</td>
<td>3.0</td>
<td>2.73e-09</td>
<td>3.0</td>
</tr>
</tbody>
</table>

### Seventh order well-balanced method

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\rho$ error</th>
<th>$\rho$ rate</th>
<th>$\rho u$ error</th>
<th>$\rho u$ rate</th>
<th>$E$ error</th>
<th>$E$ rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>1.60e-08</td>
<td>–</td>
<td>1.48e-08</td>
<td>–</td>
<td>4.22e-08</td>
<td>–</td>
</tr>
<tr>
<td>256</td>
<td>1.34e-10</td>
<td>6.9</td>
<td>1.24e-10</td>
<td>6.9</td>
<td>3.58e-10</td>
<td>6.9</td>
</tr>
<tr>
<td>512</td>
<td>1.07e-12</td>
<td>7.0</td>
<td>9.92e-13</td>
<td>7.0</td>
<td>2.86e-12</td>
<td>7.0</td>
</tr>
<tr>
<td>1024</td>
<td>9.80e-15</td>
<td>6.8</td>
<td>8.56e-15</td>
<td>6.9</td>
<td>2.48e-14</td>
<td>6.9</td>
</tr>
<tr>
<td>2048</td>
<td>4.49e-15</td>
<td>1.1</td>
<td>2.16e-15</td>
<td>2.0</td>
<td>6.88e-15</td>
<td>1.8</td>
</tr>
</tbody>
</table>
Figure 2: Perturbation on a hydrostatic atmosphere. The test setup is described in Section 9.1.2. The density deviation from the hydrostatic background is shown at time \( t = 0.5 \) for the perturbation \( \eta = 1e -5 \). The second order well-balanced method is used with different resolutions (if not stated explicitly).

Table 3: \( L^1 \)-errors for the numerically integrated hydrostatic solution of the Euler equations with radiation pressure after time \( t = 2 \) computed using different methods. The setup is described in Section 9.1.3.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \rho ) error</th>
<th>( \rho u ) error</th>
<th>( \rho v ) error</th>
<th>( E ) error</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard, first order</td>
<td>4.78e-03</td>
<td>1.56e-03</td>
<td>1.56e-03</td>
<td>5.10e-03</td>
</tr>
<tr>
<td>well-balanced, first order</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
</tr>
<tr>
<td>standard, second order</td>
<td>2.63e-06</td>
<td>6.19e-07</td>
<td>6.19e-07</td>
<td>6.34e-06</td>
</tr>
<tr>
<td>well-balanced, second order</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
</tr>
<tr>
<td>standard, third order</td>
<td>2.98e-06</td>
<td>6.66e-08</td>
<td>6.66e-08</td>
<td>6.84e-06</td>
</tr>
<tr>
<td>well-balanced, third order</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
</tr>
</tbody>
</table>

where the temperature \( T \) is defined implicitly via

\[
\varepsilon = \frac{T}{\gamma - 1} + \frac{3}{\rho} T^4. \tag{27}
\]

Using Chebfun \[43\] in the numerical software MATLAB we solve the 1-d hydrostatic equation and EoS for density and pressure corresponding to the given temperature profile. The result is shown in Fig. 3. The data are stored as point values on a fine grid (10,000 data points). The data are set on the 2-d grid using the procedure (a) from Section 5.1. We use a \( 64 \times 64 \) grid to evolve the hydrostatic initial condition to the final time \( t = 2 \). For the conversion between pressure and internal energy we use Newton’s method to solve for the temperature. The \( L^1 \)-errors at final time are shown in Table 3. In all tests using the well-balanced modification, there is no error at the final time.
9.1.4. Double Gresho vortex

In this test we use a vortex for homogeneous 2-d Euler equations first introduced in [44]. The pressure and angular velocity of this vortex in dependence of the distance to the center are given by

\[
\hat{u}(r) = \begin{cases} 
5r, & \text{if } 0 \leq r < 0.2, \\
2 - 5r, & \text{if } 0.2 \leq r < 0.4 \\
0, & \text{if } 0.4 \leq r,
\end{cases}
\]

\[
\hat{p}(r) = \begin{cases} 
5 + \frac{25}{2}r^2, & \text{if } 0 \leq r < 0.2, \\
9 - 4\ln(0.2) + \frac{25}{2}r^2 - 20r + 4\ln(r), & \text{if } 0.2 \leq r < 0.4 \\
3 + 4\ln(2), & \text{if } 0.4 \leq r.
\end{cases}
\]  

(28)

The radial velocity is zero and the density is \( \rho \equiv 1 \). In our test we set up the domain \([0,1] \times [0,2]\) with two Gresho vortices centered at \((0.5,0.5)\) and \((0.5,1.5)\) respectively. The vortexes are advected with the velocity \( \mathbf{v}_0 = (u_0, v_0)^T = (0.2, 0.4)^T \) and the boundaries are periodic. At time \( t = 5 \) the exact solution of this initial data equals the initial setup. We apply our first order well-balanced method on a \(64 \times 128\) grid to evolve the initial condition up to final time \( t = 5 \). We use Roe’s numerical flux functions and a linear reconstruction. Only the vortex initially (and finally) centered at \((0.5,0.5)\) is included in the reference solution. The result is illustrated in Fig. 4.
Figure 4: Illustration for the double Gresho vortex test from Section 9.1.4. The absolute velocity after subtraction of the constant advection velocity is shown for the initial (left panel) and final (right panel) time. The vortex which is included in the reference solution is preserved while the other one is diffused and deformed.
Table 4: \( L^1 \)-errors for the 2-d Euler equations wave in a gravitational field after time \( t = 0.1 \) computed using different methods. The setup is described in Section 9.1.5.

<table>
<thead>
<tr>
<th>method</th>
<th>grid</th>
<th>( \rho ) error</th>
<th>( \rho u ) error</th>
<th>( \rho v ) error</th>
<th>( E ) error</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard, first order</td>
<td>Cartesian</td>
<td>1.47e-02</td>
<td>1.90e-02</td>
<td>1.90e-02</td>
<td>1.39e-01</td>
</tr>
<tr>
<td>well-balanced, first order</td>
<td>Cartesian</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
</tr>
<tr>
<td>standard, second order</td>
<td>Cartesian</td>
<td>7.12e-04</td>
<td>7.82e-04</td>
<td>7.82e-04</td>
<td>2.50e-03</td>
</tr>
<tr>
<td>well-balanced, second order</td>
<td>Cartesian</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
</tr>
<tr>
<td>standard, third order</td>
<td>Cartesian</td>
<td>8.68e-05</td>
<td>9.98e-05</td>
<td>9.98e-05</td>
<td>4.50e-03</td>
</tr>
<tr>
<td>well-balanced, third order</td>
<td>Cartesian</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
</tr>
<tr>
<td>standard, second order</td>
<td>polar</td>
<td>2.45e-04</td>
<td>2.33e-04</td>
<td>2.33e-04</td>
<td>3.43e-04</td>
</tr>
<tr>
<td>well-balanced, second order</td>
<td>polar</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
</tr>
</tbody>
</table>

9.1.5. 2-d Euler wave in gravitational field

To demonstrate that we can follow time-dependent solutions exactly with our method we use a problem from \cite{18} and \cite{45} which involves a known exact solution of the 2-d Euler equations with gravity given by

\[
\rho(t, x, y) = 1 + \frac{1}{5} \sin(\pi(x + y - t(u_0 + v_0))), \quad u(t, x, y) = u_0, \quad v(t, x, y) = v_0, \\
\hat{\rho}(t, x, y) = p_0 + t(u_0 + v_0) - x - y + \frac{1}{5\pi} \cos(\pi(x + y - t(u_0 + v_0))).
\]

The gravitational potential is \( \phi(x) = x + y \), the EoS is the ideal gas EoS. In accordance to \cite{18} and \cite{45} we choose \( u_0 = v_0 = 1, \ p_0 = 4.5 \) on the domain \([0, 1]^2\). We use the standard method and the well-balanced method, first, second, and third order each, to evolve the initial data with \( t = 0 \) to a final time \( t = 0.1 \). The grid resolution is 64 \times 64 on a Cartesian grid. We also include tests with the second order methods on a polar grid with the same grid resolution. The \( L^1 \)-errors at final time are presented in Table 4. If the well-balanced method is used, there is no error.

9.1.6. Perturbation on the 2-d Euler wave in gravitational field

In this test we want to verify the order of accuracy for perturbations to time-dependent reference solutions if the well-balanced method is used. For this we use the initial setup from Eq. (29) and add a pressure perturbation:

\[
\rho(t = 0, x, y) = \hat{\rho}(t = 0, x, y), \ u(t = 0, x, y) = \hat{u}(t = 0, x, y), \ v(t = 0, x, y) = \hat{v}(t = 0, x, y), \\
p(t = 0, x, y) = \hat{p}(t = 0, x, y) + \eta \exp\left(-100\left(\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2\right)\right).
\]
Table 5: $L^1$-errors and convergence rates for a pressure perturbation ($\eta = 0.1$) on the wave in a gravitational field solution of the 2-d Euler equations after time $t = 0.1$. The third order standard and well-balanced method are used. The setup is described in Section 9.1.6.

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>$\rho$ error</th>
<th>$\rho$ rate</th>
<th>$\rho u$ error</th>
<th>$\rho u$ rate</th>
<th>$\rho v$ error</th>
<th>$\rho v$ rate</th>
<th>$E$ error</th>
<th>$E$ rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Third order standard method</td>
<td>64</td>
<td>2.17e-04</td>
<td>–</td>
<td>2.86e-04</td>
<td>–</td>
<td>2.86e-04</td>
<td>–</td>
<td>1.71e-03</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>3.68e-05</td>
<td>2.6</td>
<td>4.87e-05</td>
<td>2.6</td>
<td>4.87e-05</td>
<td>2.6</td>
<td>3.16e-04</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>5.02e-06</td>
<td>2.9</td>
<td>6.55e-06</td>
<td>2.9</td>
<td>6.55e-06</td>
<td>2.9</td>
<td>4.29e-05</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>5.66e-07</td>
<td>3.1</td>
<td>7.38e-07</td>
<td>3.1</td>
<td>7.38e-07</td>
<td>3.1</td>
<td>4.85e-06</td>
<td>3.1</td>
</tr>
</tbody>
</table>

Table 6: $L^1$-errors and convergence rates for a small pressure perturbation ($\eta = 10^{-5}$) on the wave in a gravitational field solution of the 2-d Euler equations after time $t = 0.1$. The third order well-balanced method are used. The setup is described in Section 9.1.6.

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>$\rho$ error</th>
<th>$\rho$ rate</th>
<th>$\rho u$ error</th>
<th>$\rho u$ rate</th>
<th>$\rho v$ error</th>
<th>$\rho v$ rate</th>
<th>$E$ error</th>
<th>$E$ rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Third order well-balanced method</td>
<td>64</td>
<td>1.22e-04</td>
<td>–</td>
<td>1.99e-04</td>
<td>–</td>
<td>1.99e-04</td>
<td>–</td>
<td>1.64e-03</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>2.48e-05</td>
<td>2.3</td>
<td>3.77e-05</td>
<td>2.4</td>
<td>3.77e-05</td>
<td>2.4</td>
<td>3.07e-04</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>3.55e-06</td>
<td>2.8</td>
<td>5.22e-06</td>
<td>2.9</td>
<td>5.22e-06</td>
<td>2.9</td>
<td>4.18e-05</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>4.22e-07</td>
<td>3.1</td>
<td>6.10e-07</td>
<td>3.1</td>
<td>6.10e-07</td>
<td>3.1</td>
<td>4.74e-06</td>
<td>3.1</td>
</tr>
</tbody>
</table>
We evolve these initial data to time $t = 0.1$ using the third order standard and well-balanced method with $\eta = 0.1$. The $L^1$ errors and corresponding convergence rates are presented in Table 5. As reference solution for determining the error we use a numerically approximated solution computed using the third order standard method on a $1024^2$ grid. In this test we use exact boundary conditions for the standard method, which means that we evaluate the states in the ghost cells at any time from Eq. (29). We see third order convergence for both methods. However, there seems to be no significant benefit from using the well-balanced method in this test. The choice of $\eta = 0.1$ leads to a large discretization error in the perturbation which seems to dominate the total error. This is necessary since we use a solution computed from the standard method as a reference to compute the errors. For smaller perturbations the standard method fails to produce a sufficiently accurate solution. To show the improved accuracy of the well-balanced modification we add a convergence test with a small perturbation of $\eta = 10^{-5}$. To produce a sufficiently accurate reference solution we have to use the third order well-balanced method on a $1024^2$ grid. The errors and convergence rates can be seen in Table 6. Comparing the error of the well-balanced method on the $64^2$ grid in Table 6 to the error of the third order standard method on the unperturbed solution in Table 4, we see that the well-balanced method is significantly more accurate.

9.1.7. 2-d Keplerian disk

Consider a stationary solution given by \[ \tilde{\rho} = 1, \quad \tilde{u}(x,y) = -\sin(\alpha(x,y)) \frac{Gm_s}{r(x,y)}, \quad \tilde{v}(x,y) = \cos(\alpha(x,y)) \frac{Gm_s}{r(x,y)}, \quad \tilde{p} = 1 \] (31)

with the gravitational potential $\phi(r) = -\frac{Gm_s}{r}$ and $r = \sqrt{x^2 + y^2}$, $\alpha = \arctan(\frac{y}{x})$, $G = m_s = 1$. We use the initial conditions

\[
\rho(x,y) = \begin{cases} 
2 & \text{if } (x - 1.2)^2 + (y - 1.0)^2 < 0.15^2 \\
\tilde{\rho} & \text{else}
\end{cases}
\] (32)

and $u = \tilde{u}, v = \tilde{v}, p = \tilde{p}$ on the domain $[-2,2] \times [0.5,2]$. We chose the domain such that we omit the singularity in the velocity at $(x, y) = (0,0)$. In Fig. 5 results of numerical tests are illustrated. The second order standard and well-balanced methods are applied on a polar grid with $30 \times 350$ cells and a Cartesian grid with $128 \times 128$ cells. In the Cartesian grid we take out the center with $r < 1$ using Dirichlet boundary conditions. We also use Dirichlet boundary conditions at all outer boundaries. Since there is a discontinuity in the initial setup we apply a minmod slope limiter. The density at time $t = 2.5$ for each simulation is shown in Fig. 5. The exact solution of this problem can be seen in \[31\]. Since this is an purely advective problem and there is no radial component to the velocity, the quantity $\| \rho - 1 \|_2$, which describes the
average distance of the density perturbation to the center, is conserved for all time in the exact solution. For our simulations we measure the quantity $d = \|\rho(t = 2.5) - 1\|_2/\|\rho(t = 0) - 1\|_2$ as a measure of the quality of the numerical solutions. For the exact solution we have $d = 1$ for all time. The values of $d$ are shown in the captions in Fig. 5.

In the tests with the standard method (Figs. 5a and 5c) we see discretization errors in the Keplerian disk solution Eq. (31). This introduces radial velocities, the advection of the spot of increased density has
a component towards the center. In the tests using our well-balanced methods (Figs. 5b and 5d), the result is free of discretization errors in the Keplerian disk solution Eq. (31). The advection is more accurate, the only errors are diffusion errors. The polar grid is more suitable for this test problem, since it is adapted to the radial geometry. The test using our well-balanced method on the Cartesian grid (Fig. 5b) is more diffusive than the one on the polar grid (Fig. 5d), yet we see that the well-balanced modification improves the result significantly.

9.2. Homogeneous compressible ideal magnetohydrodynamics

The 2-d compressible ideal magnetohydrodynamics equations which model the conservation of mass, momentum, magnetic field, and energy are given by

\[
\frac{\partial}{\partial t} \mathbf{q} + \frac{\partial}{\partial x} \mathbf{f} + \frac{\partial}{\partial y} \mathbf{g} = 0. \tag{33}
\]

The conserved variables and fluxes are

\[
\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ B_x \\ B_y \\ E \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p + \frac{1}{2}(B_y^2 - B_x^2) \\ \rho uv - B_x B_y \\ 0 \\ B_x u - v B_y \\ u(E + p + \frac{1}{2}B_y^2 - \frac{1}{2}B_x^2 - v B_x B_y) \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} \rho v \\ \rho uv - B_x B_y \\ \rho v^2 + p + \frac{1}{2}(B_x^2 - B_y^2) \\ B_x v - u B_y \\ 0 \\ v(E + p + \frac{1}{2}B_x^2 - \frac{1}{2}B_y^2) - u B_x B_y \end{bmatrix}, \tag{34}
\]

where \(B_x, B_y\) are the \(x\)- and \(y\)-component of the magnetic field. The total energy is \(E = \rho e + \frac{1}{2} \rho |\mathbf{v}|^2 + \frac{1}{2}(B_x^2 + B_y^2)\). All other quantities are defined as for the Euler equations. We use the same EoS as for the Euler equations.

One can also define 2-d compressible ideal MHD equations such that they include the \(\rho w\) and \(B_z\) components. This is in principle reasonable due to the genuine three-dimensional interactions between velocity and magnetic field. In our tests we set \(\rho w\) and \(B_z\) to zero and there is no difference if we omit the corresponding equations.

9.2.1. Stationary Vortex - Long Time

We consider the following exact solution of the homogeneous 2-d ideal MHD equations:

\[
\begin{align*}
\hat{x} &= x - tu_0, \\
\hat{y} &= y - tv_0, \\
u &= u_0 - k_p e^{\frac{r^2}{\sigma^2}} \hat{y}, \\
v &= v_0 + k_p e^{\frac{r^2}{\sigma^2}} \hat{x}, \\
p &= 1 + \left(\frac{m_p^2}{2} (1 - r^2) - \frac{k_p^2}{2}\right) e^{r^2}. \tag{35}
\end{align*}
\]
This setup describes a stationary vortex which is advected through the domain with the velocity \((u_0, v_0)\).

The domain is \([-5, 5] \times [-5, 5]\). One vortex turnover-time is \(t_{\text{turnover}} = \frac{2\pi}{\sqrt{ep}} \approx 3.81\). In a first test we set \(m_p = k_p = 0.1, u_0 = v_0 = 0\) and run the test up to \(t = 100t_{\text{turnover}}\) on a 32\(\times\)32 grid. We use the well-balanced method and the reference solution equals the initial data. The numerical error at final time compared to the initial setup is exactly zero in all conservative variables.

9.2.2. Stationary vortex - order of accuracy

In a second test with the vortex from Section 9.2.1 we want to see if the well-balanced method converges as expected, even if the reference solution which is set deviates from the actual solution over time. For that we set \(m_p = k_p = 0.1, u_0 = v_0 = 0\) in the initial condition. As reference solution we use the same vortex but with \(u_0 = v_0 = 1\). In Table 7 the \(L^1\) errors and rates at final time \(t = 0.2\) are presented for the formally first, second, and third order accurate well-balanced method. We omitted the errors for \(\rho v\) and \(B_y\). Due to the symmetry of the setup these errors equal the errors in \(\rho u\) and \(B_x\) respectively. We see that even if the reference solution moves away from the actual solution over time the method is consistent with the expected order of accuracy.

10. Computational cost of the modification

Well-balanced methods are constructed to improve the accuracy with which solutions of balance laws are approximated. In the previous section we have shown that usage of our well-balanced modification can improve the accuracy of a simulation significantly. On the other hand, an increase of computational effort can countervail the gain in accuracy, if it is too high. In this section we will compare the computation times of simulations using our well-balanced modification to simulations using the corresponding standard method and show that the increase in CPU time is moderate.

10.1. The procedure

To compare the methods, we will run tests with different setups and grid resolutions using a standard method and the corresponding well-balanced method. We use the simple python code described in Section 9 on a single CPU. Each test is repeated 20 times and the wall clock times are measured. We compute the average wall-clock time and standard deviations of the single runs for every test. We compute and present the ratio of average wall clock time for the well-balanced compared to the standard method. The errors are computed by adding the relative standard deviations of the well-balanced and standard method.
Table 7: $L^1$-errors and convergence rates for a 2-d MHD stationary vortex after time $t = 0.2$. Different methods are used. In the well-balanced method, a reference solution is chosen, which deviates from the actual solution over time. The setup is described in Section 9.2.2.

<table>
<thead>
<tr>
<th></th>
<th>$N$</th>
<th>$\rho$ error</th>
<th>$\rho$ rate</th>
<th>$\rho u$ error</th>
<th>$\rho u$ rate</th>
<th>$B_z$ error</th>
<th>$B_z$ rate</th>
<th>$E$ error</th>
<th>$E$ rate</th>
</tr>
</thead>
</table>
| **First order well-balanced method**
| 32            | 2.10e-03 | –            | 1.62e-02     | –              | 1.26e-02     | –           | 8.81e-02  | –        |
| 64            | 8.31e-04 | 1.3          | 8.28e-03     | 1.0           | 6.90e-03     | 0.9         | 4.70e-02  | 0.9      |
| 128           | 3.52e-04 | 1.2          | 4.15e-03     | 1.0           | 3.57e-03     | 1.0         | 2.42e-02  | 1.0      |
| 256           | 1.59e-04 | 1.1          | 2.08e-03     | 1.0           | 1.81e-03     | 1.0         | 1.23e-02  | 1.0      |
| 512           | 7.53e-05 | 1.1          | 1.04e-03     | 1.0           | 9.12e-04     | 1.0         | 6.17e-03  | 1.0      |
| **Second order well-balanced method**
| 32            | 5.08e-04 | –            | 1.93e-03     | –              | 1.49e-03     | –           | 9.60e-03  | –        |
| 64            | 1.48e-04 | 1.8          | 5.01e-04     | 1.9           | 2.96e-04     | 2.3         | 1.62e-03  | 2.6      |
| 128           | 3.92e-05 | 1.9          | 1.38e-04     | 1.9           | 7.62e-05     | 2.0         | 2.93e-04  | 2.5      |
| 256           | 9.95e-06 | 2.0          | 3.61e-05     | 1.9           | 1.99e-05     | 1.9         | 5.90e-05  | 2.3      |
| 512           | 2.49e-06 | 2.0          | 9.25e-06     | 2.0           | 5.11e-06     | 2.0         | 1.31e-05  | 2.2      |
| **Third order well-balanced method**
| 32            | 2.31e-04 | –            | 1.17e-03     | –              | 8.92e-04     | –           | 5.26e-03  | –        |
| 64            | 2.52e-05 | 3.2          | 1.50e-04     | 3.0           | 1.24e-04     | 2.8         | 7.33e-04  | 2.8      |
| 128           | 3.28e-06 | 2.9          | 1.87e-05     | 3.0           | 1.59e-05     | 3.0         | 9.56e-05  | 2.9      |
| 256           | 4.54e-07 | 2.8          | 2.32e-06     | 3.0           | 2.00e-06     | 3.0         | 1.22e-05  | 3.0      |
| 512           | 5.71e-08 | 3.0          | 2.89e-07     | 3.0           | 2.51e-07     | 3.0         | 1.54e-06  | 3.0      |
| **Third order standard method**
| 32            | 3.34e-04 | –            | 3.74e-03     | –              | 3.76e-03     | –           | 1.25e-03  | –        |
| 64            | 2.29e-05 | 3.9          | 5.06e-04     | 2.9           | 5.08e-04     | 2.9         | 9.99e-05  | 3.6      |
| 128           | 1.52e-06 | 3.9          | 6.46e-05     | 3.0           | 6.47e-05     | 3.0         | 8.64e-06  | 3.5      |
| 256           | 1.03e-07 | 3.9          | 8.12e-06     | 3.0           | 8.12e-06     | 3.0         | 8.70e-07  | 3.3      |
| 512           | 7.57e-09 | 3.8          | 1.02e-06     | 3.0           | 1.02e-06     | 3.0         | 1.01e-07  | 3.1      |
10.2. The tests

To compare the runtimes we use test setups described in Section \ref{sec:tests}. In a first test, we use the perturbed one-dimensional isothermal solution described in Section \ref{sec:isothermal_solution} with a final time $t = 0.02$. The first, second, third, and seventh order 1-d methods are applied. The ratio of wall clock times for the tests with and without the well-balanced modification can be seen in Fig. 6. Roe’s approximate Riemann solver has been used in the tests. The reference solution $\tilde{q}$ used in the well-balanced modification is constant in time in this test case. It is hence computed once and stored in an array. From the figure we see that we can expect an increase in CPU time of about 10%.

As a second test setup we choose the Keplerian disk from Section \ref{sec:keplerian_disk}. We evolve it to time $t = 0.125$ with a first and second order method on a polar grid. As in the previous test, the reference solution is time-independent and thus only computed once. We see an increase in CPU time of about 10 – 20% when using the well-balanced method.

To also test the increase of CPU time consumption for a simulation in which the reference solution is time dependent, we use the Euler wave in a gravitational field with perturbation from Section \ref{sec:euler_wave}. In this test, the reference solution is computed from a function every time it is used (which happens in every intermediate step). The result of these tests can be seen in Fig. 8. We see an increase in CPU time of about 20 – 30% if the well-balanced method is used.

Figure 6: Ratio of the average wall-clock times for the well-balanced and the standard method. The one-dimensional test case with static reference solution from Section \ref{sec:isothermal_solution} is used. Values and errors are determined as described in Section \ref{sec:performance}.
Figure 7: Ratio of the average wall-clock times for the well-balanced and the standard method. The two-dimensional test case with stationary reference solution from Section 9.1.7 is used on a polar grid. Values and errors are determined as described in Section 10.1.

Figure 8: Ratio of the average wall-clock times for the well-balanced and the standard method. The two-dimensional test case with time-dependent reference solution from Section 9.1.6 is used. Values and errors are determined as described in Section 10.1.
11. Summary and conclusions

We introduced a new general framework for the construction of well-balanced finite volume methods for hyperbolic balance laws. A standard finite volume method is modified such that it evolves the deviation from a reference solution instead of the actual solution. This makes the scheme exact on the reference solution. The finite volume method can include any consistent reconstruction, numerical flux function, interface quadrature, source term discretization, and ODE solver for time discretization. Thus it can have arbitrarily high order. The method can be defined on any computational grid geometry. One can view our method as a high order extension of [26] and [17] to all known solutions of all hyperbolic balance laws.

In numerical tests with Euler and MHD equations on different grids we could verify that the method can successfully be applied to exactly maintain static and stationary solutions or even follow time-dependent solutions. For that, the solution has to be known either analytically or in the form of discrete data. The latter case is especially interesting for complex applications like stellar astrophysics, where static states of the Euler equations with gravity can be obtained numerically but only in few cases analytically. Also, for the case of differentially rotating stars, our method can be applied for well-balancing since it can include non-zero velocities in stationary states.

High order accuracy has been confirmed in numerical experiments. For practical applications we emphasize that our well-balanced method can be easily implemented in existing finite volume codes with minimal effort. Practical advice is given. Also, in a series of numerical tests we have shown that the increase in computational time is moderate. There are applications in which the well-balanced solution is not known beforehand. In this case, one of the existing well-balanced methods for the considered balance law has to be applied, if there is one. In all cases, in which the well-balanced solution is known, our simple framework can be applied to obtain the well-balanced property.

Acknowledgments

The research of Jonas Berberich is supported by the Klaus Tschira Foundation. Praveen Chandrashekar would like to acknowledge the support received from Airbus Foundation Chair on Mathematics of Complex Systems at TIFR-CAM, Bangalore.

References


Appendix A. Details of the applied finite volume schemes

In all numerical tests we use structured grids. Hence, in the description of the details we restrict to structured grids. Some parts of the scheme, such as the reconstruction methods, are applied to \( Q \) in the standard method, to \( \Delta Q \) in the well-balanced method. We will denote the states with \( U \). Depending on the method we have \( U = Q \) or \( U = \Delta Q \).

Appendix A.1. Curvilinear coordinates

We define a 2-d curvilinear coordinate system. The coordinates in physical space are \( x = (x, y) \), the coordinates in computational space are \( \xi = (\xi, \eta) \). The \((i, j)\)-th cell is denoted \( \Omega_{i,j} \) in the physical space and by \( \hat{\Omega}_{i,j} \) in the computational space. We can rewrite Eq. (21) in the computational coordinates as

\[
J^{-1} \frac{\partial}{\partial \xi} q + \frac{\partial}{\partial \xi} \hat{f} + \frac{\partial}{\partial \eta} \hat{g} = J^{-1} s, \tag{A.1}
\]

where

\[
J^{-1} := \begin{bmatrix}
\frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\
\frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta}
\end{bmatrix}, \quad \hat{f} := \begin{bmatrix}
\frac{\partial x}{\partial \xi} f + \frac{\partial x}{\partial \eta} g \\
\frac{\partial y}{\partial \xi} f + \frac{\partial y}{\partial \eta} g
\end{bmatrix}, \quad \hat{g} := \begin{bmatrix}
\frac{\partial x}{\partial \xi} f + \frac{\partial x}{\partial \eta} g \\
\frac{\partial y}{\partial \xi} f + \frac{\partial y}{\partial \eta} g
\end{bmatrix}. \tag{A.2}
\]

To solve Eq. (21) on the curvilinear physical grid, we can now solve Eq. (A.1) on a Cartesian grid. We construct the grid from the nodes and approximate the derivatives of the coordinate transform using central differences on the nodal coordinates. This implementation of curvilinear grids restricts the scheme to second order accuracy. We can achieve higher order accuracy only on Cartesian grids. More details on the finite volume method on a curvilinear mesh can e.g. be found in [26].

Polar grid: The polar grid can be defined by the function

\[
x(\xi) := \begin{bmatrix}
\xi \sin(\eta) \\
\xi \cos(\eta)
\end{bmatrix} \tag{A.3}
\]

for \( \xi > 0, \eta \in [0, 2\pi) \). Note, that this functions can not be inverted at \( \xi = 0 \), i.e. \( x = 0 \). Hence, the origin in physical coordinates has to be omitted, when this grid is used.
Appendix A.2. Implementation of boundary conditions

Boundary conditions are implemented on the structured grid by using ghost cells to artificially extend the computational domain. In this section we assume the physical domain is in the cells \( \Omega_{ij}, (i, j) \in \{0, \ldots, N\} \times \{0, \ldots, M\} \). The necessary amount of ghost cells depends on the stencil of the method, especially on the quadrature and reconstruction (plus one row of ghost cells for the flux computation at the boundary). The values in the ghost cells are set after each intermediate Runge–Kutta step. In the following, the used boundary conditions are presented for 2-d grids. If there is no description for the 1-d method, the method is reduced to 1-d in the trivial way.

**Periodic boundary conditions:** To obtain periodic boundary conditions, the states in the ghost cells are set to the values

\[
Q_{ij} = Q_{kl} \quad \text{with} \quad k = i \mod N + 1, \quad l = j \mod M + 1.
\]  

(A.4)

**Constant extrapolation:** The constant extrapolation boundary conditions are suitable to be used with a first order accurate method. They are obtained by setting

\[
Q_{ij} = Q_{kl} \quad \text{with} \quad k = \min(N, \max(0, i)), \quad l = \min(M, \max(0, j))
\]  

(A.5)

in the ghost cells.

**Linear extrapolation:** In combination with a second order accurate method on a Cartesian grid, the following linear extrapolation to the ghost cells can be used as boundary conditions: We set

\[
Q_{i-1,j} = (1 + k)Q_{0,j} - kQ_{1,j}, \quad Q_{i-1,k} = (1 + k)Q_{0,M} - kQ_{1,1},
\]

\[
Q_{N+k,j} = (1 + k)Q_{N,j} - kQ_{N-1,j}, \quad Q_{i,M+k} = (1 + k)Q_{i,M} - kQ_{i,M-1}
\]  

(A.6)

for \((i, j) \in \{0, \ldots, N\} \times \{0, \ldots, M\}, k = 1, 2\). The diagonal ghost cells \((Q_{N+1,M+1}, \text{etc.})\) are not needed for the second order scheme used in our tests.

**Parabolic extrapolation:** Parabolic extrapolation to the ghost cells is suitable to be used in combination with a third order accurate method. On a 1-d equidistant grid we set the values in the ghost cells to

\[
Q_{N+1} = 3Q_N - 3Q_{N-1} + Q_{N-2}, \quad Q_{N+2} = 6Q_N - 8Q_{N-1} + 2Q_{N-2},
\]

\[
Q_{-1} = 3Q_0 - 3Q_1 + Q_2, \quad Q_{-2} = 6Q_0 - 8Q_1 + 2Q_2.
\]  

(A.7)
On a two-dimensional grid we have to use a genuine 2-dimensional parabolic extrapolation to obtain third order accuracy. The basis for that is a 2-d parabolic reconstruction parabola. We use the \( P_{\text{opt}} \) parabola from [36] which is given by

\[
q(x, y) \approx \frac{\Delta x \Delta y}{24} \left( \frac{12(x - x_i)^2(-2Q_{i,j} + Q_{i+1,j} + Q_{i-1,j})}{\Delta x^2} + \frac{6(x - x_i)(y - y_j)(Q_{i+1,j+1} - Q_{i-1,j+1} - Q_{i+1,j-1} + Q_{i-1,j-1})}{\Delta x \Delta y} + \frac{12(y - y_j)^2(-2Q_{i,j} + Q_{i,j+1} + Q_{i,j-1})}{\Delta y^2} + \frac{12(Q_{i,j+1} - Q_{i,j-1})(y - y_j)}{\Delta y} + 28Q_{i,j} - Q_{i+1,j} - Q_{i,j+1} - Q_{i,j-1} - Q_{i-1,j} \right) \quad (A.8)
\]

for \((x, y) \in \Omega_{ij}\). The values in the ghost cells are then computed by integrating the closest possible reconstruction parabola (computed from only values inside the domain) over the ghost cell. This yields

\[
Q_{N+k,j} = \frac{1}{2} \left( k^2(-2Q_{N-1,j} + Q_{N,j} + Q_{N-2,j}) + k(-4Q_{N-1,j} + 3Q_{N,j} + Q_{N-2,j}) + 2Q_{N,j} \right) \quad (A.9)
\]

and correspondingly for \(\Omega_{-k,j}, \Omega_{i,M+k}, \text{and} \Omega_{i,-k}\) with \(k = 1, 2, \ldots\) and \((i, j) \in \{1, N-1\} \times \{1, M-1\}\). At the upper right edge we obtain

\[
Q_{N+k,M+l} = \frac{1}{4} k^2 (-4Q_{N-1,M-1} + 2Q_{N,M-1} + 2Q_{N-2,M-1}) + \frac{1}{4} k ((l+1)(Q_{N,M} - Q_{N-2,M} - Q_{N,M-2} + Q_{N-2,M-2}) - 8Q_{N-1,M-1} + 6Q_{N,M-1} + 2Q_{N-2,M-1}) + \frac{1}{4} (-4(l+1)^2Q_{N-1,M-1} + 2(l+2)(l+1)Q_{N-1,M} + (l+1)Q_{N,M} - (l+1)Q_{N-2,M} + 2l(l+1)Q_{N-1,M-2} - (l+1)Q_{N,M-2} + (l+1)Q_{N-2,M-2} + 4Q_{N,M-1}) \quad (A.10)
\]

for a ghost cell \(\Omega_{N+k,M+l}(k, l \in \{0, 1, 2, \ldots\} \setminus \{0\}^2)\) and correspondingly for the other edges.

Appendix A.3. Volume quadrature

To approximate 2-d volume integrals with third order accuracy on a Cartesian grid we use the Gauß–Legendre quadrature rule

\[
Q_{ij} \approx \frac{1}{162} \left[ 16q(x_{ij}) + 20 \left( q(x_{ij}^N) + q(x_{ij}^E) + q(x_{ij}^S) + q(x_{ij}^W) \right) + 25 \left( q(x_{ij}^{NE}) + q(x_{ij}^{SE}) + q(x_{ij}^{SW}) + q(x_{ij}^{NW}) \right) \right] \quad (A.11)
\]
with the quadrature points
\[
x^N_{ij} = (x_i, y_j + h\Delta y), \quad x^S_{ij} = (x_i, y_j - h\Delta y), \quad x^{NE}_{ij} = (x_i + h\Delta x, y_j + h\Delta y), \quad x^{SE}_{ij} = (x_i + h\Delta x, y_j - h\Delta y),
\]
\[
x^E_{ij} = (x_i + h\Delta x, y_j), \quad x^W_{ij} = (x_i - h\Delta x, y_j), \quad x^{NW}_{ij} = (x_i - h\Delta x, y_j + h\Delta y), \quad x^{SW}_{ij} = (x_i - h\Delta x, y_j - h\Delta y)
\]
with \( h = \sqrt{\frac{3}{30}} \).

Appendix A.4. Reconstruction

Different reconstruction methods are applied depending on the formal order and the number of dimensions of the scheme.

Constant reconstruction: The reconstructed interface values for constant reconstruction are given by
\[
U^L_{i+\frac{1}{2}} = U_i, \quad U^R_{i+\frac{1}{2}} = U_{i+1}
\]
on a one-dimensional grid and by
\[
U^L_{i+\frac{1}{2},j} = U_{i,j}, \quad U^R_{i+\frac{1}{2},j} = U_{i+1,j}, \quad U^L_{i,j+\frac{1}{2}} = U_{i,j}, \quad U^R_{i,j+\frac{1}{2}} = U_{i,j+1}
\]
for two-dimensional grid.

Linear reconstruction: The reconstructed interface values for linear reconstruction are given by
\[
U^L_{i+\frac{1}{2}} = U_i + \frac{1}{4}(U_{i+1} - U_{i-1}), \quad U^R_{i+\frac{1}{2}} = U_{i+1} - \frac{1}{4}(U_{i+2} - U_i)
\]
on a one-dimensional grid and by
\[
U^L_{i+\frac{1}{2},j} = U_{i,j} + \frac{1}{4}(U_{i+1,j} - U_{i-1,j}), \quad U^R_{i+\frac{1}{2},j} = U_{i+1,j} - \frac{1}{4}(U_{i+2,j} - U_{i,j}),
\]
\[
U^L_{i,j+\frac{1}{2}} = U_{i,j} + \frac{1}{4}(U_{i,j+1} - U_{i,j-1}), \quad U^R_{i,j+\frac{1}{2}} = U_{i,j+1} - \frac{1}{4}(U_{i,j+2} - U_{i,j})
\]
for two-dimensional grid.

Parabolic reconstruction: The reconstructed interface values for parabolic reconstruction are given by
\[
U^L_{i+\frac{1}{2}} = \frac{1}{6}(2U_{i+1} + 5U_i - U_{i-1}), \quad U^R_{i+\frac{1}{2}} = \frac{1}{6}(2U_i + 5U_{i+1} - U_{i+2})
\]
on a one-dimensional grid. For the two-dimensional parabolic reconstruction we use the \( P_{\text{opt}} \) polynomial from [36]. For this reconstruction we need a polynomial in every cell since the flux is evaluated at several interface quadrature points.
**Seventh order reconstruction**: The interface values obtained using a one-dimensional seventh order accurate conservative reconstruction are

\[
U_{L}^{i+\frac{1}{2}} := \frac{1}{420} (-3U_{i-3} + 25U_{i-2} - 101U_{i-1} + 319U_{i} + 214U_{i+1} - 38U_{i+2} + 4U_{i+3}),
\]

\[
U_{R}^{i+\frac{1}{2}} := \frac{1}{420} (4U_{i-2} - 38U_{i-1} + 214U_{i} + 319U_{i+1} - 101U_{i+2} + 25U_{i+3} - 3U_{i+4}).
\]

\[(A.18)\]

**Appendix A.5. Interface quadrature**

For the two-dimensional third order method it is necessary to apply a quadrature rule to compute the interface flux. For that we use the Gauß–Legendre quadrature rule. For that, in Eq. (13), we use \(M = 3\) and the weights

\[
\omega_1 = \omega_3 = \frac{5}{18}, \quad \omega_2 = \frac{4}{9}.
\]

\[(A.19)\]

The quadrature points are

\[
x_{i+\frac{1}{2},k,1} = \sqrt{\frac{3}{5}} x_{i+\frac{1}{2},k-\frac{1}{2}} + \left(1 - \sqrt{\frac{3}{5}}\right) x_{i+\frac{1}{2},k}, \quad x_{i+\frac{1}{2},k,2} = x_{i+\frac{1}{2},k}, \quad x_{i+\frac{1}{2},k,3} = \sqrt{\frac{3}{5}} x_{i+\frac{1}{2},k+\frac{1}{2}} + \left(1 - \sqrt{\frac{3}{5}}\right) x_{i+\frac{1}{2},k}
\]

\[(A.20)\]

at the interface between the \(\Omega_{ik}\) and the \(\Omega_{i+1,k}\) control volumes and

\[
x_{i,k+\frac{1}{2},1} = \sqrt{\frac{3}{5}} x_{i-\frac{1}{2},k+\frac{1}{2}} + \left(1 - \sqrt{\frac{3}{5}}\right) x_{i,k}, \quad x_{i,k+\frac{1}{2},2} = x_{i,k+\frac{1}{2}}, \quad x_{i,k+\frac{1}{2},3} = \sqrt{\frac{3}{5}} x_{i,k+\frac{1}{2}} + \left(1 - \sqrt{\frac{3}{5}}\right) x_{i,k}.
\]

\[(A.21)\]

**Appendix A.6. Source term quadrature**

In some tests we use a gravity source term for Euler equations. The source term component in the momentum equation has to be approximated to sufficiently high order.

**Second order source term discretization**: For the first and second order method, we use the second order accurate source term discretization

\[- \frac{1}{\Delta x} \int_{\Omega_i} \rho(x)g(x)dx \approx -\bar{\rho}(\Omega_i) g(x_i)\]

\[(A.22)\]
in the one-dimensional case and

\[- \frac{1}{|\Omega_{ij}|} \int_{\Omega_{ij}} \rho(x)g(x)dx \approx -\bar{\rho}(\Omega_{ij}) g(x_{ij}) \]  
(A.23)

in the two-dimensional case in the momentum equation. The cell-averaged density is denoted \( \bar{\rho} \) and the gravitational acceleration \( \mathbf{g} = \nabla \phi \) is given exactly at the cell-center.

Higher order source terms for the Cartesian grid are obtained by exactly integrating a sufficiently high order interpolation on the cell-centered values. Cell-centered values of \( \rho \) are obtained using sufficiently high order conservative reconstruction.

**Third order source term discretization:** For the third order scheme the described procedure yields the approximation

\[- \frac{1}{\Delta x} \int_{\Omega} \rho(x)g(x)dx \approx -\frac{1}{24} \left[ \rho(x_{i-1}) g(x_{i-1}) + 22 \rho(x_i) g(x_i) + \rho(x_{i+1}) g(x_{i+1}) \right] \]  
(A.24)

with

\[ \rho(x_i) = -\frac{1}{24} \left( \bar{\rho}(\Omega_{i-1}) - 26 \bar{\rho}(\Omega_i) + \bar{\rho}(\Omega_{i+1}) \right), \]  
(A.25)

\[ \rho(x_{i+1}) = -\frac{1}{24} \left( \bar{\rho}(\Omega_i) - 26 \bar{\rho}(\Omega_{i+1}) + \bar{\rho}(\Omega_{i+2}) \right), \]  
(A.26)

\[ \rho(x_{i-1}) = -\frac{1}{24} \left( -26 \bar{\rho}(\Omega_{i-1}) + \bar{\rho}(\Omega_{i-2}) + \bar{\rho}(\Omega_i) \right) \]  
(A.27)

in the one-dimensional case and

\[- \frac{1}{|\Omega_{ij}|} \int_{\Omega_{ij}} \rho(x)g(x)dx \approx -\frac{1}{24} \left[ 20 \rho(x_{i,j}) g(x_{i,j}) + \rho(x_{i+1,j}) g(x_{i+1,j}) \right. \right.

\[ \left. \left. + \rho(x_{i,j+1}) g(x_{i,j+1}) + \rho(x_{i,j-1}) g(x_{i,j-1}) + \rho(x_{i-1,j}) g(x_{i-1,j}) \right] \]  
(A.28)

with

\[ \rho(x_{i,j}) = -\frac{1}{24} \left( \bar{\rho}(\Omega_{i-1,j}) + \bar{\rho}(\Omega_{i+1,j}) + \bar{\rho}(\Omega_{i,j-1}) + \bar{\rho}(\Omega_{i,j+1}) - 28 \bar{\rho}(\Omega_{i,j}) \right), \]  
(A.29)

\[ \rho(x_{i,j+1}) = -\frac{1}{24} \left( \bar{\rho}(\Omega_{i,j+1}) + \bar{\rho}(\Omega_{i,j-1}) + \bar{\rho}(\Omega_{i,j+1}) - 28 \bar{\rho}(\Omega_{i,j+1}) + \bar{\rho}(\Omega_{i,j+2}) + \bar{\rho}(\Omega_{i,j}) \right), \]  
(A.30)

\[ \rho(x_{i+1,j}) = -\frac{1}{24} \left( \bar{\rho}(\Omega_{i+1,j}) + \bar{\rho}(\Omega_{i+1,j}) + \bar{\rho}(\Omega_{i,j+1}) + \bar{\rho}(\Omega_{i,j}) + \bar{\rho}(\Omega_{i,j}) \right), \]  
(A.31)

\[ \rho(x_{i,j-1}) = -\frac{1}{24} \left( \bar{\rho}(\Omega_{i,j-1}) + \bar{\rho}(\Omega_{i,j-1}) - 28 \bar{\rho}(\Omega_{i,j-1}) + \bar{\rho}(\Omega_{i,j-2}) + \bar{\rho}(\Omega_{i,j}) \right), \]  
(A.32)

\[ \rho(x_{i-1,j}) = -\frac{1}{24} \left( \bar{\rho}(\Omega_{i-1,j}) + \bar{\rho}(\Omega_{i-1,j}) - 28 \bar{\rho}(\Omega_{i-1,j}) + \bar{\rho}(\Omega_{i-2,j}) + \bar{\rho}(\Omega_{i,j}) \right) \]  
(A.33)
in the two-dimensional case

**Seventh order source term discretization:** For the one-dimensional seventh order scheme the described procedure yields the approximation

\[
-\frac{1}{\Delta x} \int_{\Omega} \rho(x)g(x)dx \approx -\frac{1}{967680} \left[ -57249g(x_{i-1})\rho(x_{i-1}) - 57249g(x_{i+1})\rho(x_{i+1}) \\
+ 5058g(x_{i-2})\rho(x_{i-2}) + 5058g(x_{i+2})\rho(x_{i+2}) \\
- 367g(x_{i-3})\rho(x_{i-3}) - 367g(x_{i+3})\rho(x_{i+3}) - 862564g(x_i)\rho(x_i) \right] \tag{A.34}
\]

with

\[
\rho(x_{i-3}) = \frac{1}{107520} (-75\bar{p}_1 + 954\bar{p}_{i-1} - 7621\bar{p}_{i-2} + 121004\bar{p}_{i-3} - 7621\bar{p}_{i-4} + 954\bar{p}_{i-5} - 75\bar{p}_{i-6}), \tag{A.35}
\]

\[
\rho(x_{i-2}) = \frac{1}{107520} (954\bar{p}_1 - 7621\bar{p}_{i-1} - 75\bar{p}_{i-2} + 121004\bar{p}_{i-3} - 7621\bar{p}_{i-4} + 954\bar{p}_{i-5} - 75\bar{p}_{i-6}), \tag{A.36}
\]

\[
\rho(x_{i-1}) = \frac{1}{107520} (-7621\bar{p}_1 + 121004\bar{p}_{i-1} + 954\bar{p}_{i+1} - 7621\bar{p}_{i-2} - 75\bar{p}_{i-3} + 954\bar{p}_{i-4} - 75\bar{p}_{i-5}), \tag{A.37}
\]

\[
\rho(x_i) = \frac{1}{107520} (121004\bar{p}_1 - 7621\bar{p}_{i-1} - 7621\bar{p}_{i+1} + 954\bar{p}_{i-2} + 954\bar{p}_{i+2} - 75\bar{p}_{i-3} - 75\bar{p}_{i+3}), \tag{A.38}
\]

\[
\rho(x_{i+1}) = \frac{1}{107520} (-7621\bar{p}_1 + 954\bar{p}_{i-1} + 121004\bar{p}_{i+1} - 75\bar{p}_{i-2} - 7621\bar{p}_{i+2} + 954\bar{p}_{i+3} - 75\bar{p}_{i+4}), \tag{A.39}
\]

\[
\rho(x_{i+2}) = \frac{1}{107520} (954\bar{p}_1 - 75\bar{p}_{i-1} - 7621\bar{p}_{i+1} + 121004\bar{p}_{i+2} - 7621\bar{p}_{i+3} + 954\bar{p}_{i+4} - 75\bar{p}_{i+5}), \tag{A.40}
\]

\[
\rho(x_{i+3}) = \frac{1}{107520} (-75\bar{p}_1 + 954\bar{p}_{i+1} - 7621\bar{p}_{i+2} + 121004\bar{p}_{i+3} - 7621\bar{p}_{i+4} + 954\bar{p}_{i+5} - 75\bar{p}_{i+6}). \tag{A.41}
\]