SIAM J. SCI. COMPUT.
Vol. 38, No. 5, pp. B773–B807

WELL-BALANCED UNSTAGGERED CENTRAL SCHEMES FOR THE EULER EQUATIONS WITH GRAVITATION*

R. TOUMA†, U. KOLEY‡, AND C. KLINGENBERG§

Abstract. We consider the Euler equations with gravitational source term and propose a new well-balanced unstaggered central finite volume scheme, which can preserve the hydrostatic balance state exactly. The proposed scheme evolves a nonoscillatory numerical solution on a single grid, avoids the time consuming process of solving Riemann problems arising at the cell interfaces, and is second-order accurate both in time and space. Furthermore, the numerical scheme follows a well-balanced discretization that first discretizes the gravitational source term according to the discretization of the flux terms, and then mimics the surface gradient method and discretizes the density and energy according to the discretization of steady state density and energy functions, respectively. Finally, several numerical experiments demonstrating the performance of the well-balanced schemes in both one and two spatial dimensions are presented. The results indicate that the new scheme is accurate, simple, and robust.

Key words. Euler equations, well-balanced, unstaggered central schemes, finite volume methods, gravitational field

AMS subject classification. 65M08

DOI. 10.1137/140992667

1. Introduction.

1.1. The model. Many interesting physical phenomena are modeled by the Euler equations with gravitational source terms. These equations express the conservation of mass, momentum, and energy, which take the form in two dimensions

$$\begin{cases} \rho_t + (\rho u)_x + (\rho v)_y = 0, \\ (\rho u)_t + \left(\rho u^2 + p\right)_x + (\rho uv)_y = -\rho\phi_x, \\ (\rho v)_t + (\rho uv)_x + \left(\rho v^2 + p\right)_y = -\rho\phi_y, \\ E_t + ((E+p)u)_x + ((E+p)v)_y = -\rho u\phi_x - \rho v\phi_y. \end{cases} \tag{1.1}$$

Here, ρ denotes the fluid density, (u, v) is the velocity field, p represents the pressure, and $E = \frac{1}{2}\rho(u^2 + v^2) + p/(\gamma - 1)$ is the nongravitational energy which includes the kinetic and internal energy of the fluid. Furthermore, γ is the ratio of specific heats and $\phi = \phi(x, y)$ is the time independent gravitational potential. When the variation of the unknowns in the y-direction is negligible, one may find the one-dimensional version of (1.1) by setting v and all the derivatives in the y-direction to zero, thus

*Submitted to the journal's Computational Methods in Science and Engineering section October 23, 2014; accepted for publication (in revised form) June 21, 2016; published electronically September 15, 2016.

http://www.siam.org/journals/sisc/38-5/99266.html

†Computer Science and Mathematics, Lebanese American University, P.O. Box 13-5053, Chouran, Beirut, Lebanon (rony.touma@lau.edu.lb).

‡Tata Institute of Fundamental Research Centre, Centre For Applicable Mathematics, Post Bag No. 6503, GKVK Post Office, Sharada Nagar, Chikkabommasandra, Bangalore 560065, India (ujjwal@math.tifrbng.res.in).

§Institut für Mathematik, Julius-Maximilians-Universität Würzburg, Campus Hubland Nord, Emil-Fischer-Strasse 30, 97074, Würzburg, Germany (klingenberg@mathematik.uni-wuerzburg.de).

obtaining the system

$$\begin{cases} \rho_t + (\rho u)_x = 0, \\ (\rho u)_t + \left(\rho u^2 + p\right)_x = -\rho\phi_x, \\ E_t + \left((E+p)u\right)_x = -\rho u\phi_x. \end{cases} \tag{1.2}$$

Equation (1.2) has been used to study the atmospheric phenomena that are essential in numerical weather prediction [2], and in climate modeling as well as in a wide variety of contexts in astrophysics such as modeling solar climate or simulating supernova explosions [11, 7].

The Euler equation with gravitation (1.1) amounts to a system of *balance laws*,

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = -\mathbf{S}(\mathbf{U}), \tag{1.3}$$

where

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{pmatrix}, \text{ and } \mathbf{S} = \begin{pmatrix} 0 \\ \rho\phi_x \\ \rho\phi_y \\ \rho u\phi_x + \rho v\phi_y \end{pmatrix}.$$

Here $\mathbf{U}$ denotes the vector of unknowns, $\mathbf{F}$ and $\mathbf{G}$ are the flux vectors, and $\mathbf{S}$ is the source vector. The special case of $\mathbf{S} = 0$ reads

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = 0, \tag{1.4}$$

which is termed a system of conservation law.

It is well known that solutions of conservation law (1.4) and, likewise, solutions of the balance law (1.3), can develop shock discontinuities in a finite time, independent of whether the initial data are smooth or not. Hence the solutions of balance laws (1.3) are considered in the weak sense and are well-defined as long as source $\mathbf{S}$ remains uniformly bounded. Furthermore, these weak solutions may not be unique. Additional admissibility criteria or *entropy conditions* need to be imposed in order to select the physically relevant solution.

1.2. Steady states. An important issue which arises in connection with balance laws such as the Euler equation with gravitation (1.2) is the simulation of their steady states. A steady state for (1.2) is a solution that is constant in time, in which the source term is exactly balanced by the flux gradient. The importance of near steady state flows occurs in astrophysics, in particular, in the simulation of core-collapse supernova explosions, where the nascent neutron star slowly settling to an equilibrium albeit the explosion, taking place in a highly dynamic environment just above the nascent neutron star, does not set in for another few hundreds of ms [7]. Here, the interest is in accurate long term simulations of near stationary states.

For the static gravitational potential $\phi(x)$, we are interested in preserving the following stationary solution for (1.2): The velocity is zero, i.e., $u = 0$ and the pressure exactly balances the gravitational force

$$p_x = -\rho\phi_x. \tag{1.5}$$

The above steady state models the so-called *mechanical equilibrium* and is incomplete to some extent as the density and pressure stratifications are not uniquely specified. Another thermodynamic is needed (e.g., entropy or temperature) to uniquely

determine the equilibrium. Two important classes of stable hydrostatic equilibria are given by constant entropy, i.e., isentropic, and constant temperature, i.e., isothermal, respectively. As a concrete example, we will concentrate on the isothermal case [16].

In what follows, we consider a special steady state solution to (1.2),

$$\rho = \rho(x), \qquad u = 0, \qquad p_x = -\rho\phi_x \tag{1.6}$$

with a *constant temperature* and zero velocity. For an ideal gas, we have the relation

$$p = \rho RT, \tag{1.7}$$

where R is the gas constant and T is the temperature. Substituting (1.7) in the steady state equation $p_x = -\rho\phi_x$ yields

$$\rho(x) = \rho_0 \exp\left(-\frac{\phi}{RT}\right), \tag{1.8}$$

which essentially leads to the special steady state

$$\rho(x) = \rho_0 \exp\left(-\frac{\phi}{RT}\right), \qquad u = 0, \qquad p = \rho RT = RT\rho_0 \exp\left(-\frac{\phi}{RT}\right). \tag{1.9}$$

The simplest and most commonly encountered case in literature is the linear gravitational potential field, i.e., $d\phi/dx = g$, with the corresponding hydrostatic balance

$$\rho(x) = \rho_0 \exp\left(-\frac{g\rho_0 x}{p_0}\right), \qquad u = 0, \qquad p = \rho RT = p_0 \exp\left(-\frac{g\rho_0 x}{p_0}\right). \tag{1.10}$$

The importance of steady states such as the above equilibrium (1.10) lies in the fact that in many situations of interest, the dynamics is realized as a perturbation of the steady states. As examples, consider the simulation of small perturbations on a gravitationally stratified atmosphere such as those arising in numerical weather prediction [2] and the simulation of waves in steller atmospheres [11, 6]. We mention that more general steady states (e.g., the polytropic case [20]) can't be handled directly by the method developed in this article, and will be the topic of an upcoming paper.

1.3. Well-balanced schemes. A challenge in the numerical analysis of balance laws is to maintain these steady states, and to compute their perturbations accurately. Indeed, if a scheme cannot balance the effects of convective fluxes and source term, it may introduce spurious oscillations near equilibria, unless the mesh size is extremely refined. Many astrophysical problems involve the hydrodynamical evolution in a gravitational field, therefore, it is essential to correctly capture the effect of gravitational force in the simulations, especially if a long-time integration is involved, for example, in modeling star and galaxy formulations. Standard numerical schemes with naive discretizations of the source term might not preserve the steady state. This implies that the scheme does not keep a discrete form of (1.10) stationary in time. The error can be at least of the order of truncation error for each time step and can lead to large deviations from the steady state for long-time scales. Furthermore, computing small perturbations of (1.10) is not possible due to lack of balancing. A numerical scheme which preserves a discrete version of a steady-state-like (1.10) is termed *well-balanced* with respect to the steady state. Well-balanced schemes are essential for computing perturbations of steady states.

Well balanced schemes for systems of balance laws (1.3) are still undergoing extensive development. In fact, the pioneering paper of LeVeque [12] was one of the first to propose a well-balanced scheme for the shallow water equations with bottom topography. Then a variety of well-balanced schemes have been designed to approximate the ocean at rest steady state that arises in the shallow water equations with non-trivial bottom topography. A very limited list of references includes [11, 1, 3, 9, 10, 15] and other references therein. Furthermore, LeVeque and Bale [13] proposed the quasi-steady wave propagation methods for an ideal gas object to a static gravitational field. A Riemann problem is introduced in the center of each grid cell such that the flux difference exactly cancels the source term. A different strategy for the construction of well-balanced discretizations with respect to dominant hydrostatics has been proposed by Botta et al. [2] for the nearly hydrostatic flows belonging to a certain class of solutions. A well-balanced scheme that preserves a discrete version of some hydrostatic steady states of (1.2) has been presented in [6, 11, 13, 16].

The key principle underlying the design of most of the aforementioned well-balanced schemes consisted of replacing the piecewise constant cell averages, used as inputs to finite volume schemes, with values constructed from a *local discrete hydrostatic equilibrium*. This results in a first-order scheme. The design of a second-order scheme requires using a well-balanced piecewise linear reconstruction with respect to the local discrete hydrostatic equilibrium. Recently, in [8], well-balanced high-order finite volume schemes were designed which preserve discrete equilibria, corresponding to a large class of physically stable hydrostatic steady states.

Finally, we mention the paper by Desveaux et al. [5]. They developed an approximate Riemann solver using the formalism of Harten, Lax, and van Leer, which takes into account the source term. The well-balanced solver is based on a finite volume method, where the source term is somehow incorporated into the Riemann solver. The resulting numerical scheme was proven to be robust, to preserve exactly the hydrostatic atmosphere, and to preserve an approximation of *all the other steady state solutions*.

1.4. Aim of this paper. We aim to develop a new unstaggered central scheme which is an unstaggered adaptation of the Nessyahu and Tadmor [14] scheme. Briefly, this new method is based on a careful projection of the numerical solution obtained on the staggered cells, back onto the original cells. In this paper we construct, analyze, and implement a new unstaggered, well-balanced, nonoscillatory, and second-order accurate central scheme for the one- and two-dimensional systems of Euler equations with gravitation. Two main features characterize the proposed well-balanced schemes: first a special discretization of the source term according to the discretization of the flux divergence and, second, a proper projection of the numerical solution obtained on the dual cells, back onto the original cells. The latter step is performed according to the surface gradient method discussed in [17, 4].

The main advantages of the proposed schemes are

(a) it is a second-order accurate approximation of the one- and two-dimensional systems (1.2) and (1.1). It is a second-order accurate approximation of the one- and two-dimensional systems (1.2) and (1.1);
(b) it enjoys the simplicity of the Riemann-free-solver approach;
(c) it is well-balanced; exactly maintains the steady state requirement at the discrete level.

The proposed well-balanced scheme is successfully applied and classical Euler equations with gravitation problems are solved, both in one and two space dimensions.

The steady state requirement is exactly satisfied at the discrete level and the obtained numerical results are in good agreement with corresponding ones appearing in the recent literature, thus confirming the potential of the proposed method.

The rest of the paper organized as follows: The well-balanced, unstaggered, finite volume one-dimensional scheme is presented in section 2, and the two-dimensional scheme is presented in section 3. Numerical results are presented in section 4 and finally a brief summary of this paper is presented in section 5.

2. Schemes for one-dimensional Euler equations with gravitation. In this section we develop a second-order accurate central unstaggered well-balanced finite volume method for the Euler equation with gravity systems that preserves the steady state requirement of type (1.9) in the case of a linear gravitational field (1.10):

$$\rho = \exp(-gx), \qquad u = 0, \qquad p = \exp(-gx). \tag{2.1}$$

Remark 2.1. Note that, in general, the steady state requirement (1.10) can be recast as follows: For any $\alpha, \beta > 0$,

$$\rho = \alpha \exp(-\beta g x), \qquad u = 0, \qquad p = \frac{\alpha}{\beta} \exp(-\beta g x).$$

In this presentation we have chosen to work with $\alpha = \beta = 1$ to avoid unnecessary redundancy.

We first rewrite the source term following the approach proposed in [16] and reformulate the Euler equation with gravity system (1.2) as follows:

$$\begin{cases} \mathbf{u}_t + f(\mathbf{u})_x = S(\mathbf{u}),\ x \in \Omega,\ t > 0, \\ \mathbf{u}(x,0) = \mathbf{u}_0(x), \end{cases} \tag{2.2}$$

where $\Omega \subset \mathbb{R}$ is a bounded spatial domain, and

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, f(\mathbf{u}) = \begin{pmatrix} \rho u \\ \left(\rho u^2 + p\right) \\ (E+p)u \end{pmatrix}, \text{ and } S(\mathbf{u}) = \begin{pmatrix} 0 \\ \rho \exp(gx) \left(\exp(-gx)\right)_x \\ \rho u \exp(gx) \left(\exp(-gx)\right)_x \end{pmatrix}.$$

Here we have replaced the gravitational source term $-\rho g$ by $\rho \exp(gx) \left(\exp(-gx)\right)_x$, and also the term $-\rho u g$ is treated in a similar fashion. The main advantage of such a change is to let the source term and the corresponding flux term enjoy a similar form in the case of a steady state solution.

2.1. The grid and notation. We start by introducing some notation needed to define the fully discrete finite volume schemes. Throughout this paper we reserve Δx and Δt to denote small positive numbers that represent the spatial and temporal discretization parameters, respectively, of the numerical schemes. Given $\Delta x, \Delta t > 0$, let $D_\pm, D_0$ denote the discrete forward, backward and central differences, respectively, in spatial direction, i.e.,

$$D_\pm g(x) = \pm \frac{1}{\Delta x} \left(g(x \pm \Delta x) - g(x)\right),$$
$$D_0 g(x) = \frac{1}{2} \left(D_+ g(x) + D_- g(x)\right),$$

for any function $g : \mathbb{R} \to \mathbb{R}$ admitting point values. For $i \in \mathbb{Z}$, we set $x_i = i\Delta x$ and for $n = 1, \ldots, N$, where $N\Delta t = T$ for some fixed time horizon $T > 0$, we set $t^n = n\Delta t$.

For any function $g = g(x)$ admitting point values we write $g_i = g(x_i)$, and similarly for any function $h = h(x,t)$ admitting point values we write $h_i^n = h(x_i, t^n)$. Moreover, we introduce the spatial and temporal grid cells

$$C_i = [x_{i+1/2}, x_{i+1/2}], \quad D_{i+1/2} = [x_i, x_{i+1}], \text{ and } R^n_{i+1/2} = D_{i+1/2} \times [t^n, t^{n+1}],$$

where $x_{i\pm 1/2} = x_i \pm \Delta x/2$ with $\Delta x = x_{i+1/2} - x_{i-1/2} = x_{i+1} - x_i$.

Furthermore we introduce the average and, respectively, the jump of any grid function ρ across the interfaces $x_{i+\frac{1}{2}}$ and x_i:

$$\overline{\rho}_{i+\frac{1}{2}} := \frac{\rho_i + \rho_{i+1}}{2}, \quad \overline{\rho}_i := \frac{\rho_{i+\frac{1}{2}} + \rho_{i-\frac{1}{2}}}{2}, \quad [\![\rho]\!]_{i+\frac{1}{2}} := \rho_{i+1} - \rho_i, \quad [\![\rho]\!]_i := \rho_{i+\frac{1}{2}} - \rho_{i-\frac{1}{2}}.$$

Finally, we shall also use following notations for the steady state density function (cf. (2.15)):

$$[\![\rho]\!]^s_i := \rho^s_{i+\frac{1}{2}} - \rho^s_{i-\frac{1}{2}}, \quad \overline{\rho}^s_i := \frac{\rho^s_{i+\frac{1}{2}} + \rho^s_{i-\frac{1}{2}}}{2}.$$

2.2. One-dimensional scheme. To this end, we construct an unstaggered central scheme which preserves the steady state requirement (2.1). This method we develop computes the numerical solution on a single grid but uses "ghost" staggered cells to avoid the resolution of the Riemann problems at the cell interfaces when updating the numerical solution. Piecewise linear reconstructions of the numerical solution defined at the center of the ghost cells send back the updated solution to the original grid.

In what follows, without any loss of generality, we assume that the numerical solution $\mathbf{u}_i^n$ to system (2.2) is known at time t^n at the centers x_i of the control cells C_i. To construct the numerical solution $\mathbf{u}_i^{n+1}$ at time $t^{n+1} = t^n + \Delta t$ on the control cells C_i, we shall follow a standard finite volumes procedure; we first define the piecewise linear interpolants $\mathcal{L}_i(x, t^n)$ that approximate the exact solution $\mathbf{u}(x, t^n)$ on the cells C_i as follows:

$$\mathcal{L}_i(x, t^n) = \mathbf{u}_i^n + (x - x_i)\frac{(\mathbf{u}_i^n)'}{\Delta x} \quad \forall x \in C_i, \tag{2.3}$$

where $(\mathbf{u}_i^n)'/\Delta x$ is a limited numerical gradient that approximates the spatial partial derivative $\frac{\partial}{\partial x}\mathbf{u}(x,t^n)|_{x=x_i}$. Throughout this paper, we have used van Leer's monotonized centered limiter (MC-θ), where the slope of the reconstruction is chosen as

$$(\mathbf{u}_i^n)' = \text{minmod}\left[\theta[\![\mathbf{u}]\!]^n_{i-\frac{1}{2}}, \frac{\mathbf{u}^n_{i+1} - \mathbf{u}^n_{i-1}}{2}, \theta[\![\mathbf{u}]\!]^n_{i+\frac{1}{2}}\right],$$

where θ is chosen such that $1 \le \theta \le 2$ and the minmod function is defined by

$$\text{minmod}(a,b,c) = \begin{cases} \text{sign}(a)\ \min\{|a|,|b|,|c|\} & \text{if } \text{sign}(a) = \text{sign}(b) = \text{sign}(c), \\ 0 & \text{otherwise.} \end{cases}$$

It is worth mentioning that to overcome the disadvantage of excessive numerical viscosity, present in the case of first-order piecewise constant interpolants, we use high resolution MUSCL-type interpolants (2.3). We now integrate the balance law (2.2) on

the rectangle $R^n_{i+1/2}$ and apply Green's theorem to the integral on the left to obtain

$$\int\int_{R^n_{i+\frac{1}{2}}} [\mathbf{u}_t + f(\mathbf{u})_x] dR = \int\int_{R^n_{i+\frac{1}{2}}} S(\mathbf{u}) dR,$$

$$\oint_{\partial R^n_{i+\frac{1}{2}}} (f(\mathbf{u})dt - \mathbf{u}dx) = \int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} S(\mathbf{u}) dx dt.$$

Performing elementary calculations, we obtain

$$-\int_{x_i}^{x_{i+1}} \mathbf{u}(x,t^n)dx + \int_{t^n}^{t^{n+1}} f\left(\mathbf{u}(x_{i+1},t)\right) dt + \int_{x_i}^{x_{i+1}} \mathbf{u}(x,t^{n+1})dx - \int_{t^n}^{t^{n+1}} f\left(\mathbf{u}(x_i,t)\right) dt = \int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} S(\mathbf{u}) dx dt. \tag{2.4}$$

Assuming that the solution $\mathbf{u}(x,t^{n+1}) \approx \mathcal{L}(x,t^{n+1})$ is a piecewise linear function defined at the center of the cells $D_{i+1/2}$, the mean-value theorem leads to

$$\int_{x_i}^{x_{i+1}} \mathbf{u}(x,t^{n+1})dx = \Delta x\, \mathcal{L}(x_{i+\frac{1}{2}},t^{n+1}) = \Delta x\ \mathbf{u}^{n+1}_{i+\frac{1}{2}}.$$

On the other hand the solution $\mathbf{u}(x,t^n) \approx \mathcal{L}(x,t^n)$ is a piecewise linear function defined at the center of the cells C_i; the mean-value theorem leads to

$$\int_{x_i}^{x_{i+1}} \mathbf{u}(x,t^n)dx = \int_{x_i}^{x_{i+1/2}} \mathbf{u}(x,t^n)dx + \int_{x_{i+1/2}}^{x_{i+1}} \mathbf{u}(x,t^n)dx = \frac{\Delta x}{2}\mathcal{L}_i(x_{i+\frac{1}{4}},t^n) + \frac{\Delta x}{2}\mathcal{L}_{i+1}(x_{i+\frac{3}{4}},t^n) := \Delta x\ \mathbf{u}^n_{i+\frac{1}{2}}. \tag{2.5}$$

Therefore, the forward projected solution $\mathbf{u}^n_{i+\frac{1}{2}}$ at time t^n is calculated as follows:

$$\mathbf{u}^n_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathcal{L}_i\left(x_i + \frac{\Delta x}{4}, t^n\right) + \mathcal{L}_{i+1}\left(x_{i+1} - \frac{\Delta x}{4}, t^n\right)\right) = \overline{\mathbf{u}}^n_{i+\frac{1}{2}} - \frac{1}{8}[\![(\mathbf{u}^n)']\!]_{i+\frac{1}{2}}. \tag{2.6}$$

Substituting in (2.4), we obtain

$$\mathbf{u}^{n+1}_{i+\frac{1}{2}} = \mathbf{u}^n_{i+\frac{1}{2}} - \frac{1}{\Delta x}\left[\int_{t^n}^{t^{n+1}} f\left(\mathbf{u}(x_{i+1},t)\right) dt - \int_{t^n}^{t^{n+1}} f\left(\mathbf{u}(x_i,t)\right) dt\right] + \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} S(\mathbf{u}) dx dt. \tag{2.7}$$

We see from (2.7) that the numerical solution $\mathbf{u}^{n+1}_{i+\frac{1}{2}}$, computed at time t^{n+1}, is obtained at the center of the control cells $D_{i+1/2}$.

On the other hand the flux integrals in (2.7) are approximated with second-order accuracy using the midpoint quadrature rule, leading to

$$\mathbf{u}^{n+1}_{i+\frac{1}{2}} = \mathbf{u}^n_{i+1/2} - \Delta t D_+ f(\mathbf{u}_i^{n+\frac{1}{2}}) + \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} S\left(\mathbf{u}(x,t)\right) dx dt, \tag{2.8}$$

where the predicted solution values at time $t^{n+1/2}$ are obtained using a first-order Taylor expansion in time as well as the balance law as

(2.9)
$$\begin{aligned}\mathbf{u}(x_i, t^{n+\frac{1}{2}}) &\approx \mathbf{u}(x_i, t^n) + \frac{\Delta t}{2}\mathbf{u}_t(x_i, t^n) \approx \mathbf{u}_i^n + \frac{\Delta t}{2}\Big(- f(\mathbf{u})_x|_{(x_i,t^n)} + S(\mathbf{u})|_{(x_i,t^n)}\Big)\\ &\approx \mathbf{u}_i^n + \frac{\Delta t}{2}\Big(- \frac{f_i'}{\Delta x} + S_i^n\Big) := \mathbf{u}_i^{n+\frac{1}{2}},\end{aligned}$$

where $f_i'/\Delta x$ is an approximate flux derivative and $S_i^n \approx S(\mathbf{u}_i^n)$ is a second-order approximation of the source term at time t^n on the cell C_i and is defined using a sensor function as follows:

$$S_i^n = S_{i,L}^n + S_{i,R}^n + S_{i,C}^n \tag{2.10}$$

with

$$S_{i,L}^n = \frac{s_i^2\,(1 - s_i)\,(2 - s_i)}{6}\begin{pmatrix} 0 \\ \theta\rho_i^n \exp(gx_i) D_-(\exp(-gx_i)) \\ \theta\rho_i^n u_i^n \exp(gx_i) D_-(\exp(-gx_i)) \end{pmatrix},$$

$$S_{i,R}^n = \frac{s_i^2\,(1 + s_i)\,(2 - s_i)}{2}\begin{pmatrix} 0 \\ \theta\rho_i^n \exp(gx_i) D_+(\exp(-gx_i)) \\ \theta\rho_i^n u_i^n \exp(gx_i) D_+(\exp(-gx_i)) \end{pmatrix},$$

$$S_{i,C}^n = \frac{s_i\,(s_i + 1)\,(s_i - 1)}{6}\begin{pmatrix} 0 \\ \rho_i^n \exp(gx_i) D_0(\exp(-gx_i)) \\ \rho_i^n u_i^n \exp(gx_i) D_0(\exp(-gx_i)) \end{pmatrix}.$$

Note that the sensor function s_i, appearing in the discretization of the source term, forces the discretization of the term "$(\exp(-gx))_x$" to follow the same discretization of the term p_x which appears in the flux function, and is defined by

$$s_i = \begin{cases} -1 & \text{if } p_i' = \theta\, D_- p_i, \\ 1 & \text{if } p_i' = \theta\, D_+ p_i, \\ 0 & \text{if } p_i' = 0, \\ 2 & \text{if } p_i' = D_0 p_i, \end{cases} \tag{2.11}$$

where $1 \le \theta \le 2$ is the parameter of the MC-θ limiter. For further information on sensor functions, the interested reader is referred to [4], [21], and [22].

As for the integral of the source term in (2.8), it is also approximated with second-order accuracy using centered differences and the midpoint quadrature rule as follows:

$$\int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} S\left(\mathbf{u}(x,t)\right) dx\; dt \approx \Delta t\, \Delta x\, S(\mathbf{u}_i^{n+\frac{1}{2}}, \mathbf{u}_{i+1}^{n+\frac{1}{2}})$$

with

$$S(\mathbf{u}_i^{n+\frac{1}{2}}, \mathbf{u}_{i+1}^{n+\frac{1}{2}}) = \begin{pmatrix} 0 \\ \overline{\rho^{n+1/2}\, \exp(gx)}\big|_{i+\frac{1}{2}}\, D_+ \exp(-gx_i) \\ \overline{\rho^{n+1/2}\, \exp(gx)}\big|_{i+\frac{1}{2}}\, \overline{u}_{i+\frac{1}{2}}^n\, D_+ \exp(-gx_i) \end{pmatrix}. \tag{2.12}$$

Finally, we proceed with a backward projection step of the obtained numerical solution $\mathbf{u}_{i+1/2}^{n+1}$ back onto the cells C_i and generate the numerical solution $\mathbf{u}_i^{n+1}$ as follows:

$$\mathbf{u}_i^{n+1} = \overline{\mathbf{u}}_i^{n+1} - \frac{1}{8}[\![(\mathbf{u}^{n+1})']\!]_i, \tag{2.13}$$

where $(\mathbf{u}_{i+1/2}^{n+1})'$ denotes a limited numerical gradient that approximates the spatial partial derivative $\frac{\partial}{\partial x}\mathbf{u}(x, t^{n+1})|_{x=x_{i+\frac{1}{2}}}$. This finishes the description of the central finite volume scheme.

To show that the proposed finite volume scheme (2.13) preserves the steady state (2.1), we first show that $\mathbf{u}_{i+1/2}^{n+1} = \mathbf{u}_{i+1/2}^{n}$. In that context, we have the following theorem.

THEOREM 2.1. *Let the numerical solution* $\mathbf{u}_i^n$ *of the one-dimensional Euler equation with gravity system be updated using the finite volume method* (2.8) *and* (2.13). *Moreover, assume that at time* t^n *the steady state requirement* (2.1) *is satisfied by* $\mathbf{u}_i^n$, *i.e.,*

$$u_i^n = 0 \quad \textit{and} \quad \rho_i^n = \exp(-gx_i) = p_i^n. \tag{2.14}$$

Then,

(a) *the predicted solution given by* (2.9) *satisfies* $\mathbf{u}_i^{n+1/2} = \mathbf{u}_i^n$;

(b) *the forward projected solution given by* (2.8) *and* (2.12) *satisfies* $\mathbf{u}_{i+1/2}^{n+1} = \mathbf{u}_{i+1/2}^{n}$.

Proof. To prove (a), we first recall that the prediction step is obtained from (2.9). Observe that if, for example $(p_x)_i^n$ is discretized by the MC-θ parameter using the backwards difference, i.e., $p_i' = \theta D_- p_i$, then the sensor function s_i takes on the value -1, and the discretized source term becomes $S_i^n = S_{i,L}^n$. Furthermore, since $\mathbf{u}_i^n$ satisfies the steady state requirement, then $u_i^n = 0$ and therefore the first and third components in the flux divergence as well as in the source term in (2.9) are zero. Thus the first and third components of $\mathbf{u}_i^{n+1/2}$ are the same as those of $\mathbf{u}_i^n$. The second component of $\mathbf{u}_i^{n+1/2}$ is computed as follows:

$$\begin{aligned}(\rho u)_i^{n+1/2} &= (\rho u)_i^n + \frac{\Delta t}{2}\left(-p_i' + \rho_i^n \exp(gx_i)\theta D_- \exp(-gx_i)\right)\\ &= (\rho u)_i^n + \frac{\Delta t}{2}\left(-\theta D_- p_i^n + \rho_i^n \exp(gx_i)\theta D_- \exp(-gx_i)\right),\end{aligned}$$

and since $\mathbf{u}_i^n$ satisfies the steady state requirement (2.14), then

$$(\rho u)_i^{n+1/2} = (\rho u)_i^n + \frac{\Delta t}{2\Delta x}\theta\left(-D_- p_i^n + D_- \exp(-gx_i)\right) = (\rho u)_i^n,$$

and therefore we conclude that $\mathbf{u}_i^{n+1/2} = \mathbf{u}_i^n$ holds.

Next we move on to the proof of (b), i.e., we show that $\mathbf{u}_{i+1/2}^{n+1} = \mathbf{u}_{i+1/2}^{n}$ provided the solution $\mathbf{u}_i^n$ satisfies the steady state requirement (2.14). From (2.8), we know that

$$\mathbf{u}_{i+\frac{1}{2}}^{n+1} = \mathbf{u}_{i+1/2}^{n} - \frac{\Delta t}{\Delta x}\left[f(\mathbf{u}_{i+1}^{n+\frac{1}{2}}) - f(\mathbf{u}_i^{n+\frac{1}{2}})\right] + \Delta t S(\mathbf{u}_i^{n+\frac{1}{2}}, \mathbf{u}_{i+1}^{n+\frac{1}{2}}).$$

But since $\mathbf{u}_i^{n+1/2} = \mathbf{u}_i^n$, then the first and third components of the flux as well as

the first and third components of the source term are all identically zero. This means that the corresponding first and third components of both $\mathbf{u}_{i+\frac{1}{2}}^{n+1}$ and $\mathbf{u}_{i+\frac{1}{2}}^{n}$ are equal. As for the second component of $\mathbf{u}_{i+\frac{1}{2}}^{n+1}$, it is updated as follows (keeping in mind that $p_i^{n+1/2} = \rho_i^{n+1/2} = p_i^n = \rho_i^n = \exp(-gx_i)$):

$$\begin{aligned}(\rho u)_{i+1/2}^{n+1} &= (\rho u)_{i+1/2}^{n} - \Delta t D_+ p_i^{n+1/2} + \Delta t \, \overline{\rho^{n+1/2} \exp(gx)}\big|_{i+\frac{1}{2}} D_+ \exp(-gx_i) \\ &= (\rho u)_{i+1/2}^{n}.\end{aligned}$$

Thus we conclude that whenever the numerical solution $\mathbf{u}_i^n$ satisfies the steady state requirement at time t^n, then the updated numerical solution $\mathbf{u}_{i+1/2}^{n+1}$ on the staggered dual cells $D_{i+1/2}$ remains unchanged, i.e., $\mathbf{u}_{i+1/2}^{n+1} = \mathbf{u}_{i+1/2}^{n}$ for all i. ∎

Observe that from Theorem 2.1, it is easy to conclude that if the steady state requirement was satisfied at the discrete level at time t^n at the center of the cells C_i, it will remain as such at time t^{n+1} but only at the center of the staggered cells $D_{i+1/2}$. However, the back projection step prescribed in (2.9) fails, in general, to fulfill the steady state requirement at time t^{n+1} at the center of the cells C_i and therefore an additional treatment is required. In this presentation, we extend the *surface gradient method* [17] initially developed for the shallow water equations in [4, 21], and later adapted for the Ripa system in [22], to the case of systems of Euler equations with gravity.

Since in the steady state case the velocity component is zero, the forward and backward projection steps based on the surface gradient method will be performed to both the first component ρ and and third component E of the numerical solution $\mathbf{u}_i^n$. We first assume that the numerical solution $\mathbf{u}_i^n$ obtained at time t^n satisfies the steady state requirement (2.14) and we define the steady state density function $\rho_{i+1/2}^s$ at the interfaces of the cells C_i (or at the centers of the cells $D_{i+1/2}$) needed for the forward projection of ρ_i^n and E_i^n. Similarly, we define the steady state energy function $E_{i+1/2}^s$ needed for the forward projection of third component E_i^n. We first describe the forward projection step of the first component ρ_i^n and then we argue similarly for the third component E_i^n. In what follows, knowing $\rho_{i+1/2}^s$ at the points $x_{i+1/2}$ we linearize the steady state density function ρ^s on the cells C_i as

$$\rho^s(x) = \rho_i^s + \frac{1}{\Delta x}(x - x_i)[\![\rho]\!]_i^s \quad \forall x \in C_i. \tag{2.15}$$

Note that at the cell centers the relation $\rho_i^s = \overline{\rho}_i^s$ remains valid. Since both $\rho^n : C_i \to \mathbb{R}$ and $\rho^s : C_i \to \mathbb{R}$ are linear on the control cells C_i, we follow the surface gradient method and define the function $H(x) = \rho^n(x) - \rho^s(x)$. The linearization $H(x) = H_i + H_i'(x - x_i)$ on the cells C_i is obtained by using a limiting procedure of the numerical derivatives of $H_i = \rho_i^n - \rho_i^s$. We then calculate the numerical gradient $(\rho_i^n)'$ indirectly using both ρ_i^s and H_i as follows:

$$(\rho_i^n)' = H_i' + \frac{1}{\Delta x}[\![\rho]\!]_i^s. \tag{2.16}$$

Similarly, for the projection step of $\rho_{i+1/2}^{n+1}$ back onto the original cells C_i at time t^{n+1}, we consider the surface gradient method and linearize $\rho_{i+1/2}^{n+1}$ in terms of $\widetilde{\rho}_{i+1/2}^s$ and $\widetilde{H}_{i+1/2} = \rho_{i+1/2}^{n+1} - \widetilde{\rho}_{i+1/2}^s$, and calculate the numerical gradient $(\rho_{i+1/2}^{n+1})'$ (required in (2.13)) as follows:

$$(\rho_{i+1/2}^{n+1})' = \widetilde{H}_{i+1/2}' + \frac{1}{\Delta x}[\![\rho]\!]_{i+\frac{1}{2}}^s, \tag{2.17}$$

where $\widetilde{\rho}^s_{i+1/2}$ is the corrected steady state function value on the staggered cells $D_{i+1/2}$ defined as follows:

$$\widetilde{\rho}^s_{i+1/2} = \rho^s_{i+1/2} - \frac{1}{2}\left(\rho^s_{i+1/2} - \overline{\rho}^s_{i+\frac{1}{2}}\right) = \frac{1}{2}\left(\rho^s_{i+1/2} + \overline{\rho}^s_{i+\frac{1}{2}}\right). \tag{2.18}$$

The reason for this correction in the steady state density function is due to the fact the steady state is linear only on the cells C_i but not on the cells $D_{i+1/2}$.

The forward and backward projection steps of the energy component follow a similar procedure. For the forward projection step we linearize the energy E^n_i using the function $\mathcal{H}_i = E^n_i - E^s_i$ and calculate its numerical gradient as follows:

$$(E^n_i)' = \mathcal{H}'_i + \frac{1}{\Delta x}[\![E]\!]^s_i, \tag{2.19}$$

while for the back projection step the linearization is performed using the corrected values of the steady state energy $\widetilde{E}_{i+1/2}$,

$$(E^{n+1}_{i+1/2})' = \widetilde{\mathcal{H}}'_{i+1/2} + \frac{1}{\Delta x}[\![\rho]\!]^s_{i+\frac{1}{2}}, \tag{2.20}$$

where $\widetilde{\mathcal{H}}_{i+1/2} = E^{n+1}_{i+1/2} - \widetilde{E}^s_{i+1/2}$ and $\widetilde{E}^s_{i+1/2}$ is the corrected steady state energy on the dual cells

$$\widetilde{E}^s_{i+1/2} = E^s_{i+1/2} - \frac{1}{2}\left(E^s_{i+1/2} - \overline{E}^s_{i+\frac{1}{2}}\right) = \frac{1}{2}\left(E^s_{i+1/2} + \overline{E}^s_{i+\frac{1}{2}}\right). \tag{2.21}$$

We are now in a position to state and prove the theorem that confirms preservation of a discrete version of steady state (1.10) by the approximate solution $\mathbf{u}^n_i$.

THEOREM 2.2. *Let the approximate solution $\mathbf{u}^n_i$ of the Euler equation with gravity system* (2.2) *updated using the finite volume method* (2.8) *and under the hypotheses of Theorem* 2.1 *along with the surface-gradient-based forward projection step* (2.6), (2.16), (2.19) *and backward projection step* (2.13), (2.17), *and* (2.20). *Then the scheme* (2.8) *has the following properties:*

(a) *Accuracy: It is a second-order accurate approximation of the Euler equation with gravity system* (2.2).
(b) *Well-balanced: It preserves the steady state* (1.10), *i.e., if $\mathbf{u}^n_i$ satisfies* (1.10) *then the updated solution $\mathbf{u}^{n+1}_i$ also satisfies* (1.10).

Proof. First, we mention that second-order accuracy of the scheme is guaranteed if the numerical gradient vector satisfies

$$\frac{1}{\Delta x}\mathbf{u}'_i = \frac{\partial}{\partial x}\mathbf{u}(x,t)|_{x=x_i} + \mathcal{O}(\Delta x).$$

In fact, a straightforward truncation error analysis shows that the local truncation error is $\mathcal{O}(\Delta x^2)$ which confirms (a). To prove (b), we first calculate the forward projected solution $\rho^n_{i+1/2}$ using (2.6),

$$\rho^n_{i+1/2} = \overline{\rho}^n_{i+\frac{1}{2}} - \frac{\Delta x}{8}[\![(\rho^n)']\!]_{i+\frac{1}{2}}, \tag{2.22}$$

where $(\rho^n_i)'$ is a numerical gradient obtained using (2.16). Taking into account that $\rho^n_i = \rho^s_i$, then H'_i vanishes and (2.22) becomes

$$\rho^n_{i+1/2} = \overline{\rho}^n_{i+\frac{1}{2}} - \frac{1}{8}\left([\![\rho]\!]^s_{i+1} - [\![\rho]\!]^s_i\right), \tag{2.23}$$

but since $\rho^s_{i+1/2}$ is assumed to be linear inside the control cells C_i, then $[\![\rho]\!]^s_i = 2(\rho^s_{i+1/2} - \rho^s_i) = 2(\rho^s_i - \rho^s_{i-1/2})$, thus (2.23) becomes

$$\rho^n_{i+1/2} = \overline{\rho}^n_{i+\frac{1}{2}} + \frac{1}{2}\left(\rho^s_{i+1/2} - \overline{\rho}^s_{i+\frac{1}{2}}\right). \tag{2.24}$$

Next, the back projection of $\rho^{n+1}_{i+1/2}$ is performed using (2.13) as follows:

$$\rho^{n+1}_i = \overline{\rho}^{n+1}_i - \frac{\Delta x}{8}[\![(\rho^{n+1})']\!]_i, \tag{2.25}$$

where $(\rho^{n+1}_{i+1/2})'$ is calculated using the corrected steady state density function as described in (2.17). Note that (2.18) and (2.24) yield

(2.26)

$$\begin{aligned}\widetilde{H}_{i+1/2} &= \rho^{n+1}_{i+1/2} - \widetilde{\rho}^s_{i+1/2} = \rho^n_{i+1/2} - \widetilde{\rho}^s_{i+1/2} = \overline{\rho}^n_{i+\frac{1}{2}} + \frac{1}{2}\left(\rho^s_{i+1/2} - \overline{\rho}^s_{i+\frac{1}{2}}\right) - \widetilde{\rho}^s_{i+1/2} \\ &= \overline{\rho}^n_{i+\frac{1}{2}} - \overline{\rho}^s_{i+\frac{1}{2}} = \frac{1}{2}\left(\rho^n_i - \rho^s_i\right) + \frac{1}{2}\left(\rho^n_{i+1} - \rho^s_{i+1}\right) = 0.\end{aligned}$$

Equations (2.17), (2.25), (2.24), and (2.26) give

$$\begin{aligned}\rho^{n+1}_i &= \overline{\rho}^{n+1}_i - \frac{\Delta x}{8}[\![(\rho^{n+1})']\!]_i = \overline{\rho}^n_i - \frac{1}{8}\left([\![\rho]\!]^s_{i+\frac{1}{2}} - [\![\rho]\!]^s_{i-\frac{1}{2}}\right) \\ &= \frac{1}{2}\left(\overline{\rho}^n_{i+\frac{1}{2}} + \frac{1}{2}\left(\rho^s_{i+1/2} - \overline{\rho}^s_{i+\frac{1}{2}}\right) + \overline{\rho}^n_{i-\frac{1}{2}} + \frac{1}{2}\left(\rho^s_{i-1/2} - \overline{\rho}^s_{i-\frac{1}{2}}\right)\right) \\ &\quad - \frac{1}{8}\left([\![\rho]\!]^s_{i+\frac{1}{2}} - [\![\rho]\!]^s_{i-\frac{1}{2}}\right) = \frac{1}{2}(\rho^n_i + \rho^s_i) = \rho^n_i,\end{aligned}$$

because the steady state (2.14) is maintained at time t^n and thus $\rho^n_{i-1} - \rho^s_{i-1} = \rho^n_i - \rho^s_i = \rho^n_{i+1} - \rho^s_{i+1} = 0$.

To keep the presentation fairly short we have only provided details for the component ρ. However, we note that the same proofs apply mutatis mutandis also for the component E. Hence, we conclude that $E^{n+1}_i = E^n_i$. ∎

3. Schemes for two-dimensional Euler equations with gravitation. In this section, we develop a second-order accurate central unstaggered well-balanced finite volume method for the Euler equation with gravitation system in two space dimensions (1.1). Note that the hydrostatic balance we would like to preserve at the discrete level is the constant temperature and zero velocity steady state solution,

$$\rho = \rho_0 \exp\left(-\frac{\phi}{RT}\right), \qquad u = v = 0, \qquad p = \rho RT = RT\rho_0 \exp\left(-\frac{\phi}{RT}\right), \tag{3.1}$$

and the steady state solution corresponding to the linear gravitational potential field,

$$\frac{d\phi}{dx} = g_1 \quad \text{and} \quad \frac{d\phi}{dy} = g_2$$

takes the form

$$\rho = \exp(-(g_1 x + g_2 y)), \qquad u = v = 0, \qquad p = \exp(-(g_1 x + g_2 y)). \tag{3.2}$$

As before, the gas dynamics equations (1.1) can be reformulated in the form of a balance law as follows

$$\mathbf{u}_t + \partial_x f(\mathbf{u}) + \partial_y g(\mathbf{u}) = S(\mathbf{u}), \quad (x,y) \in \Omega,\, t > 0, \tag{3.3}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded spatial domain, and

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad f(\mathbf{u}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{pmatrix}, \quad g(\mathbf{u}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{pmatrix}, \text{ and}$$

$$S(\mathbf{u}) = \begin{pmatrix} 0 \\ \rho \exp(g_1 x + g_2 y) \left(\exp(-g_1 x - g_2 y)\right)_x \\ \rho \exp(g_1 x + g_2 y) \left(\exp(-g_1 x - g_2 y)\right)_y \\ \exp(g_1 x + g_2 y) \left(\rho u \left(\exp(-g_1 x - g_2 y)\right)_x + \rho v \left(\exp(-g_1 x - g_2 y)\right)_y\right) \end{pmatrix},$$

along with the initial condition $\mathbf{u}(x,y,t=0) = \mathbf{u}_0(x,y)$ for all $(x,y) \in \Omega$.

3.1. The grid and notation. We start by introducing some notation needed to define the fully discrete finite volume schemes. We reserve Δx, Δy, and Δt to denote small positive numbers that represent the spatial (in x and y directions respectively) and temporal discretization parameters, respectively, of the numerical schemes. Given $\Delta x, \Delta y, \Delta t > 0$, let $D^x_\pm, D^y_\pm, D^x_0, D^y_0$ denote the discrete forward, backward, and central differences, respectively, in spatial directions, i.e.,

$$D^x_\pm g(x,y) = \pm \frac{1}{\Delta x}\left(g(x \pm \Delta x, y) - g(x,y)\right), \quad D^x_0 g(x,y) = \frac{1}{2}\left(D^x_+ g(x,y) + D^x_- g(x,y)\right),$$
$$D^y_\pm g(x,y) = \pm \frac{1}{\Delta y}\left(g(x, y \pm \Delta y) - g(x,y)\right), \quad D^y_0 g(x,y) = \frac{1}{2}\left(D^y_+ g(x,y) + D^y_- g(x,y)\right)$$

for any function $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ admitting point values. For $i, j \in \mathbb{Z}$, we set $x_i = i\Delta x, y_j = j\Delta y$ and for $n = 1, \ldots, N$, where $N\Delta t = T$ for some fixed time horizon $T > 0$ we set $t^n = n\Delta t$.

For any function $g = g(x,y)$ admitting point values we write $g_{i,j} = g(x_i, y_j)$, and, similarly for any function $h = h(x,y,t)$ admitting point values we write $h^n_{i,j} = h(x_i, y_j, t^n)$. Moreover, let us introduce the spatial and temporal grid cells

$$C_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}], \quad D_{i+1/2,j+1/2} = [x_i, x_{i+1}] \times [y_j, y_{j+1}],$$
$$\text{and } R^n_{i+1/2,j+1/2} = D_{i+1/2,j+1/2} \times [t^n, t^{n+1}],$$

where $x_{i\pm1/2} = x_i \pm \Delta x/2$, $y_{j\pm1/2} = y_j \pm \Delta y/2$ with $\Delta x = x_{i+1/2} - x_{i-1/2} = x_{i+1} - x_i$ and $\Delta y = y_{j+1/2} - y_{j-1/2} = y_{j+1} - y_j$.

Furthermore we introduce the jump and, respectively, the average of any grid function ρ across the interfaces $x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, x_i$, and y_j:

$$\overline{\rho}_{i,j+\frac{1}{2}} := \frac{\rho_{i,j} + \rho_{i,j+1}}{2}, \quad \overline{\rho}_{i+\frac{1}{2},j} := \frac{\rho_{i,j} + \rho_{i+1,j}}{2}, \quad \overline{\rho}_{i,(j)} := \frac{\rho_{i,j+\frac{1}{2}} + \rho_{i,j-\frac{1}{2}}}{2},$$

$$\overline{\rho}_{(i),j} := \frac{\rho_{i+\frac{1}{2},j} + \rho_{i-\frac{1}{2},j}}{2}, \quad [\![\rho]\!]_{i,j+\frac{1}{2}} := \rho_{i,j+1} - \rho_{i,j}, \quad [\![\rho]\!]_{i+\frac{1}{2},j} := \rho_{i+1,j} - \rho_{i,j},$$

$$[\![\rho]\!]_{i,(j)} := \rho_{i,j+\frac{1}{2}} - \rho_{i,j-\frac{1}{2}}, \quad [\![\rho]\!]_{(i),j} := \rho_{i+\frac{1}{2},j} - \rho_{i-\frac{1}{2},j}.$$

3.2. Two-dimensional scheme. Following the same strategy, as explained for the one-dimensional scheme, we construct an unstaggered central scheme which preserves the steady state requirement (3.2). To do that, we assume that the numerical solution $\mathbf{u}_i^n$ to system (3.3) is known at time t^n at the centers (x_i, y_j) of the control cells $C_{i,j}$. To construct the numerical solution $\mathbf{u}_i^{n+1}$ at time $t^{n+1} = t^n + \Delta t$ on the control cells $C_{i,j}$, we shall follow a standard finite volumes procedure; we first define the piecewise linear interpolants $\mathcal{L}_{i,j}(x, y)$ that approximate the exact solution of system (3.3) on the cells $C_{i,j}$ as

$$\mathcal{L}_{i,j}(x, y, t^n) := \mathbf{u}_{i,j}^n + (x - x_i)\frac{(\mathbf{u}_{i,j}^{n,x})'}{\Delta x} + (y - y_j)\frac{(\mathbf{u}_{i,j}^{n,y})'}{\Delta y} \quad \forall (x, y) \in C_{i,j}, \tag{3.4}$$

where $(\mathbf{u}_{i,j}^{n,x})'/\Delta x$ and $(\mathbf{u}_{i,j}^{n,x})'/\Delta x$ are limited numerical gradients that approximate the spatial partial derivative $\frac{\partial}{\partial x}\mathbf{u}(x, y_j, t^n)|_{x=x_i}$ and $\frac{\partial}{\partial y}\mathbf{u}(x_i, y, t^n)|_{y=y_j}$, respectively, using the van Leer's MC-θ limiter. Next, we integrate the balance law (3.3) on the rectangular box $R^n_{i+1/2,j+1/2}$ to get

$$\int_{R^n_{i+1/2,j+1/2}} (\partial_t \mathbf{u} + \partial_x f(\mathbf{u}) + \partial_y g(\mathbf{u}))\, dV = \int_{R^n_{i+1/2,j+1/2}} S(\mathbf{u})\, dV. \tag{3.5}$$

Invoking Green's theorem and taking into account that $\mathbf{u}(x, y) \approx \mathcal{L}_{i,j}(x, y)$ on the cells $C_{i,j}$, we obtain

$$\begin{aligned} \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}^{n} &- \frac{1}{\Delta x \Delta y}\int_{t^n}^{t^{n+1}} \int_{\partial D_{i+\frac{1}{2},j+\frac{1}{2}}} f(\mathbf{u}) \cdot n_x \, dA \, dt \\ &- \frac{1}{\Delta x \Delta y}\int_{t^n}^{t^{n+1}} \int_{\partial D_{i+\frac{1}{2},j+\frac{1}{2}}} g(\mathbf{u}) \cdot n_y \, dA \, dt \\ &+ \frac{1}{\Delta x \Delta y}\int\int\int_{R^n_{i+\frac{1}{2},j+\frac{1}{2}}} S(\mathbf{u})\, dV, \end{aligned} \tag{3.6}$$

where (n_x, n_y) is the outer unit normal vector to $\partial D_{i+1/2,j+1/2}$. The integral of the source term is approximated to second-order accuracy using the midpoint quadrature rule

$$\int\int\int_{R^n_{i+\frac{1}{2},j+\frac{1}{2}}} S(\mathbf{u}) dV \approx \Delta t\, \Delta x\, \Delta y\; S\left(\mathbf{u}_{i,j}^{n+\frac{1}{2}}, \mathbf{u}_{i+1,j}^{n+\frac{1}{2}}, \mathbf{u}_{i,j+1}^{n+\frac{1}{2}}, \mathbf{u}_{i+1,j+1}^{n+\frac{1}{2}}\right).$$

Expanding the spatial flux integrals in (3.6) and applying the midpoint quadrature rules to the time integrals we obtain

$$\begin{aligned} \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}^{n} &- \frac{\Delta t}{2}\left[D_+^x f(\mathbf{u}_{i,j}^{n+\frac{1}{2}}) + D_+^x f(\mathbf{u}_{i,j+1}^{n+\frac{1}{2}})\right] \\ &- \frac{\Delta t}{2}\left[D_+^y g(\mathbf{u}_{i,j}^{n+\frac{1}{2}}) + D_+^y g(\mathbf{u}_{i+1,j}^{n+\frac{1}{2}})\right] \\ &+ \Delta t \cdot S\left(\mathbf{u}_{i,j}^{n+\frac{1}{2}}, \mathbf{u}_{i+1,j}^{n+\frac{1}{2}}, \mathbf{u}_{i,j+1}^{n+\frac{1}{2}}, \mathbf{u}_{i+1,j+1}^{n+\frac{1}{2}}\right), \end{aligned} \tag{3.7}$$

where $\mathbf{u}^n_{i+1/2,j+1/2}$ is the projected solution at time t^n on the staggered dual cells

$D_{i+1/2,j+1/2}$, and is evaluated using a Taylor expansion in space as follows:

$$\begin{aligned}\mathbf{u}^n_{i+1/2,j+1/2} = \frac{1}{2}\left(\overline{\mathbf{u}}^n_{i+\frac{1}{2},j} + \overline{\mathbf{u}}^n_{i+\frac{1}{2},j+1}\right) \\ - \frac{\Delta x}{16}\left(\llbracket \mathbf{u}^{n,x}\rrbracket_{i+\frac{1}{2},j} + \llbracket \mathbf{u}^{n,x}\rrbracket_{i+\frac{1}{2},j+1}\right) \\ - \frac{\Delta y}{16}\left(\llbracket \mathbf{u}^{n,y}\rrbracket_{i,j+\frac{1}{2}} + \llbracket \mathbf{u}^{n,y}\rrbracket_{i+1,j+\frac{1}{2}}\right).\end{aligned} \tag{3.8}$$

The solution at time t^{n+1} on the cells C_{ij} of the original grid is then obtained using a back projection step as follows:

$$\begin{aligned}\mathbf{u}^{n+1}_{i,j} = \frac{1}{2}(\overline{\mathbf{u}}^{n+1}_{i,j-\frac{1}{2}} + \overline{\mathbf{u}}^{n+1}_{i,j+\frac{1}{2}}) \\ - \frac{\Delta x}{16}\left(\llbracket \mathbf{u}^{n+1,x}\rrbracket_{(i),j-\frac{1}{2}} + \llbracket \mathbf{u}^{n+1,x}\rrbracket_{(i),j+\frac{1}{2}}\right) \\ - \frac{\Delta y}{16}\left(\llbracket \mathbf{u}^{n+1,y}\rrbracket_{i-\frac{1}{2},(j)} + \llbracket \mathbf{u}^{n+1,y}\rrbracket_{i+\frac{1}{2},(j)}\right),\end{aligned} \tag{3.9}$$

where $(\mathbf{u}^{n+1,x}_{i+1/2,j+1/2})'$ and $(\mathbf{u}^{n+1,y}_{i+1/2,j+1/2})'$ denote a limited numerical gradient that approximates the spatial partial derivative $\frac{\partial}{\partial x}\mathbf{u}(x, y_{j+\frac{1}{2}}, t^{n+1})|_{x=x_{i+\frac{1}{2}}}$ and $\frac{\partial}{\partial y}\mathbf{u}(x_{i+\frac{1}{2}}, y, t^{n+1})|_{y=y_{j+\frac{1}{2}}}$, respectively.

On the other hand, the predicted solution values $\mathbf{u}^{n+1/2}_{i,j}$ at the intermediate time step $t^{n+1/2}$ in (3.7) are estimated using a first-order Taylor expansion in time and the balance law (3.3):

$$\mathbf{u}^{n+\frac{1}{2}}_{i,j} = \mathbf{u}^n_{i,j} + \frac{\Delta t}{2}\left(-\frac{f'_{i,j}}{\Delta x} - \frac{g'_{i,j}}{\Delta y} + S^n_{i,j}\right), \tag{3.10}$$

where $f'_{i,j}/\Delta x$ and $g'_{i,j}/\Delta y$ are approximate flux derivatives and $S^n_{i,j} \approx S(\mathbf{u}^n_{i,j})$ is a second-order approximation of the source term at time t^n on the cell $C_{i,j}$ and is defined using a sensor function as follows:

$$S^n_{i,j} \approx \begin{pmatrix} S_1 = 0 \\ S_2 \\ S_3 \\ S_4 \end{pmatrix}, \tag{3.11}$$

where

$$S_2 = \begin{cases} \rho^n_{i,j}\exp(g_1x_i + g_2y_j)\Theta D^x_-\left(\exp(-g_1x_i - g_2y_j)\right) & \text{if } s_2 = -1, \\ 0 & \text{if } s_2 = 0, \\ \rho^n_{i,j}\exp(g_1x_i + g_2y_j)\Theta D^x_+\left(\exp(-g_1x_i - g_2y_j)\right) & \text{if } s_2 = 1, \\ \rho^n_{i,j}\exp(g_1x_i + g_2y_j)D^x_0\left(\exp(-g_1x_i - g_2y_j)\right) & \text{if } s_2 = 2; \end{cases} \tag{3.12}$$

$$S_3 = \begin{cases} \rho^n_{i,j}\exp(g_1x_i + g_2y_j)\Theta D^y_-\left(\exp(-g_1x_i - g_2y_j)\right) & \text{if } s_3 = -1, \\ 0 & \text{if } s_3 = 0, \\ \rho^n_{i,j}\exp(g_1x_i + g_2y_j)\Theta D^y_+\left(\exp(-g_1x_i - g_2y_j)\right) & \text{if } s_3 = 1, \\ \rho^n_{i,j}\exp(g_1x_i + g_2y_j)D^y_0\left(\exp(-g_1x_i - g_2y_j)\right) & \text{if } s_3 = 2; \end{cases}$$

$$S_4 = u^n_{i,j}S_2 + v^n_{i,j}S_3.$$

Note that the above parameters s_2 and s_3 are two sensor parameters that force the discretization of $(\exp(-g_1x - g_2y))_x$ and $(\exp(-g_1x - g_2y))_y$ according to the discretizations of p_x and p_y, respectively. They are defined as

$$s_2 = \begin{cases} -1 & \text{if } (p^x_{i,j})' = \Theta D^x_- p_{i,j}, \\ 0 & \text{if } (p^x_{i,j})' = 0, \\ 1 & \text{if } (p^x_{i,j})' = \Theta D^x_+ p_{i,j}, \\ 2 & \text{if } (p^x_{i,j})' = D^x_0\, p_{i,j}\,. \end{cases} \quad \text{and} \quad s_3 = \begin{cases} -1 & \text{if } (p^y_{i,j})' = \Theta D^y_- p_{i,j}, \\ 0 & \text{if } (p^y_{i,j})' = 0, \\ 1 & \text{if } (p^y_{i,j})' = \Theta D^y_+ p_{i,j}, \\ 2 & \text{if } (p^y_{i,j})' = D^y_0\, p_{i,j}. \end{cases} \tag{3.13}$$

The parameter $1 \le \Theta \le 2$ appearing in the formulas for S_2 and S_3 is the MC-Θ limiter parameter.

Finally, we discretize the integral of the source term in using the midpoint quadrature rule in order to ensure second-order accuracy both in space and time:

$$S(\mathbf{u}^{n+\frac12}_{i,j}, \mathbf{u}^{n+\frac12}_{i+1,j}, \mathbf{u}^{n+\frac12}_{i,j+1}, \mathbf{u}^{n+\frac12}_{i+1,j+1}) = \begin{pmatrix} \mathcal{S}_1 = 0 \\ \mathcal{S}_2 \\ \mathcal{S}_3 \\ \mathcal{S}_4 \end{pmatrix} \tag{3.14}$$

with

$$\mathcal{S}_1 = 0,$$

$$\mathcal{S}_2 = \frac12 \Big(\overline{\rho^{n+1/2} \exp(g_1x + g_2y)}\big|_{i+\frac12,j} D^x_+ \exp(-g_1x_i - g_2y_j) + \overline{\rho^{n+1/2} \exp(g_1x + g_2y)}\big|_{i+\frac12,j+1} D^x_+ \exp(-g_1x_i - g_2y_{j+1}) \Big),$$

$$\mathcal{S}_3 = \frac12 \Big(\overline{\rho^{n+1/2} \exp(g_1x + g_2y)}\big|_{i,j+\frac12} D^y_+ \exp(-g_1x_i - g_2y_j) + \overline{\rho^{n+1/2} \exp(g_1x + g_2y)}\big|_{i+1,j+\frac12} D^y_+ \exp(-g_1x_{i+1} - g_2y_j) \Big),$$

$$\begin{aligned} \mathcal{S}_4 = \frac12 \Big(& \overline{\rho^{n+1/2} \exp(g_1x + g_2y)}\big|_{i+\frac12,j} \overline{u}^{n+1/2}_{i+\frac12,j} D^x_+ \exp(-g_1x_i - g_2y_j) \\ & + \overline{\rho^{n+1/2} \exp(g_1x + g_2y)}\big|_{i+\frac12,j+1} \overline{u}^{n+1/2}_{i+\frac12,j+1} D^x_+ \exp(-g_1x_i - g_2y_{j+1}) \\ & + \overline{\rho^{n+1/2} \exp(g_1x + g_2y)}\big|_{i,j+\frac12} \overline{v}^{n+1/2}_{i,j+\frac12} D^y_+ \exp(-g_1x_i - g_2y_j) \\ & + \overline{\rho^{n+1/2} \exp(g_1x + g_2y)}\big|_{i+1,j+\frac12} \overline{v}^{n+1/2}_{i+1,j+\frac12} D^y_+ \exp(-g_1x_{i+1} - g_2y_j) \Big). \end{aligned}$$

Note that special discretization of the source term will help us to show that the updated solution $\mathbf{u}^{n+1}_{i+1/2,j+1/2}$ at time t^{n+1} is equal to the forward projected solution $\mathbf{u}^n_{i+1/2,j+1/2}$ at time t^n. In fact, we have the following theorem.

THEOREM 3.1. *Let the numerical solution $\mathbf{u}^n_{i,j}$ of the two-dimensional Euler with gravity system be updated using the finite volume method* (3.7) *and* (3.9). *Moreover,*

assume that at time t^n the steady state (3.2) *is satisfied by* $\mathbf{u}_{i,j}^n$ *at the discrete level, i.e.,*

$$\rho_{i,j}^n = p_{i,j}^n = \exp(-g_1 x_i - g_2 y_j) \quad \textit{and} \quad u_{i,j}^n = v_{i,j}^n = 0. \tag{3.15}$$

Then,

(a) *the predicted solutions given by* (3.10) *satisfy* $\mathbf{u}_{i,j}^{n+1/2} = \mathbf{u}_{i,j}^n$*;*

(b) *the solutions given by* (3.7) *and* (3.8) *satisfy* $\mathbf{u}_{i+1/2,j+1/2}^{n+1} = \mathbf{u}_{i+1/2,j+1/2}^n$.

Proof. To prove (a), we proceed componentwise. In what follows, we first show that $\rho_{i,j}^{n+1/2} = \rho_{i,j}^n$. From (3.10) we have (keeping in mind that $u_{i,j}^n = v_{i,j}^n = 0$ in an equilibrium state)

$$\rho_{i,j}^{n+1/2} = \rho_{i,j}^n + \frac{\Delta t}{2}\left[-\left((\rho u)_{i,j}^{n,x}\right)' - \left((\rho v)_{i,j}^{n,y}\right)' + S_1\right] = \rho_{i,j}^n.$$

Next, we show that $(\rho u)_{i,j}^{n+1/2} = (\rho u)_{i,j}^n$. Again from (3.10) and taking into account that $u_{i,j}^n = v_{i,j}^n = 0$ in the case of an equilibrium state, we obtain

$$\begin{aligned}(\rho u)_{i,j}^{n+1/2} &= (\rho u)_{i,j}^n + \frac{\Delta t}{2}\left[-\left((\rho u^2 + p)_{i,j}^{n,x}\right)' - \left((\rho u v)_{i,j}^{n,y}\right)' + S_2\right] \\ &= (\rho u)_{i,j}^n + \frac{\Delta t}{2}\left[-\left(p_{i,j}^x\right)' + S_2\right].\end{aligned} \tag{3.16}$$

If $\left(p_{i,j}^x\right)'$ is discretized using the backwards difference (i.e., $s_2 = -1$ in (3.13)), then (3.12) leads to

$$S_2 = \rho_{i,j}^n \exp(g_1 x_i + g_2 y_j)\Theta D_-^x \exp(-g_1 x_i - g_2 y_j).$$

Then using (3.15), (3.16) leads to $(\rho u)_{i,j}^{n+1/2} = 0 = (\rho u)_{i,j}^n$. In a similar fashion, we show that $(\rho u)_{i,j}^{n+1/2} = 0 = (\rho u)_{i,j}^n$ if s_2 takes other values. We argue similarly to show that $(\rho v)_{i,j}^{n+1/2} = 0 = (\rho v)_{i,j}^n$.

Finally, we have to show that $E_{i,j}^{n+1/2} = E_{i,j}^n$. In fact, from (3.10) and (3.15) we obtain

$$E_{i,j}^{n+1/2} = E_{i,j}^n + \frac{\Delta t}{2}\left[-\left(u(E+p)\right)_{i,j}^{n,x}\right)' - \left(v(E+p)\right)_{i,j}^{n,y}\right)' + S_4\Big] = E_{i,j}^n.$$

This concludes the proof of (a).

To prove (b), we also proceed componentwise. Observe that, from (3.7), (3.14), and (3.15) we have

$$\begin{aligned}\rho_{i+1/2,j+1/2}^{n+1} = \rho_{i+1/2,j+1/2}^n &- \frac{\Delta t}{2}\left[D_+^x(\rho u)_{i,j}^{n+1/2} + D_+^x(\rho u)_{i,j+1}^{n+1/2}\right] \\ &- \frac{\Delta t}{2}\left[D_+^y(\rho v)_{i,j}^{n+1/2} + D_+^y(\rho v)_{i+1,j}^{n+1/2}\right] + \Delta t\mathcal{S}_1 = \rho_{i+1/2,j+1/2}^n.\end{aligned}$$

Likewise we have

(3.17)
$$(\rho u)^{n+1}_{i+1/2,j+1/2} = (\rho u)^{n}_{i+1/2,j+1/2} - \frac{\Delta t}{2}\left[D^x_+(\rho u^2+p)^{n+1/2}_{i,j} + D^x_+(\rho u^2+p)^{n+1/2}_{i,j+1}\right] - \frac{\Delta t}{2}\left[D^y_+(\rho uv)^{n+1/2}_{i,j} + D^y_+(\rho uv)^{n+1/2}_{i+1,j}\right] + \Delta t\mathcal{S}_2.$$

Keeping in mind that the predicted solution is invariant in time ($\mathbf{u}^{n+1/2}_{i,j} = \mathbf{u}^n_{i,j}$) and equilibrium condition (3.15), the term $\mathcal{S}_2$ can be rewritten as

$$\mathcal{S}_2 = \frac{1}{2}\left(D^x_+ p^{n+1/2}_{i,j} + D^x_+ p^{n+1/2}_{i,j+1}\right).$$

Therefore (3.17) becomes

$$\begin{aligned}(\rho u)^{n+1}_{i+1/2,j+1/2} &= (\rho u)^{n}_{i+1/2,j+1/2} - \frac{\Delta t}{2}\left(D^x_+ p^{n+1/2}_{i,j} + D^x_+ p^{n+1/2}_{i,j+1}\right) + \Delta t\mathcal{S}_2\\ &= (\rho u)^{n}_{i+1/2,j+1/2}.\end{aligned}$$

Similarly, from (3.7), we have

(3.18)
$$(\rho v)^{n+1}_{i+1/2,j+1/2} = (\rho v)^{n}_{i+1/2,j+1/2} - \frac{\Delta t}{2}\left[D^y_+(\rho v^2+p)^{n+1/2}_{i,j} + D^y_+(\rho u^2+p)^{n+1/2}_{i+1,j}\right] - \frac{\Delta t}{2}\left[D^x_+(\rho uv)^{n+1/2}_{i,j} + D^y_+(\rho uv)^{n+1/2}_{i,j+1}\right] + \Delta t\mathcal{S}_3.$$

Again, keeping in mind that $\mathbf{u}^{n+1/2}_{i,j} = \mathbf{u}^n_{i,j}$ and equilibrium condition (3.15), the term $\mathcal{S}_3$ can be rewritten as

$$\mathcal{S}_3 = \frac{1}{2}\left(D^y_+ p^{n+1/2}_{i,j} + D^y_+ p^{n+1/2}_{i+1,j}\right)$$

Therefore (3.18) becomes

$$\begin{aligned}(\rho v)^{n+1}_{i+1/2,j+1/2} &= (\rho v)^{n}_{i+1/2,j+1/2} - \frac{\Delta t}{2}\left[D^y_+ p^{n+1/2}_{i,j} + D^y_+ p^{n+1/2}_{i+1,j}\right] + \Delta t\mathcal{S}_3\\ &= (\rho v)^{n}_{i+1/2,j+1/2}.\end{aligned}$$

Finally, from (3.7), we have

(3.19)
$$E^{n+1}_{i+1/2,j+1/2} = E^{n}_{i+1/2,j+1/2} - \frac{\Delta t}{2}\left[D^x_+((E+p)u)^{n+1/2}_{i,j} + D^x_+((E+p)u)^{n+1/2}_{i,j+1}\right] - \frac{\Delta t}{2}\left[D^y_+((E+p)v)^{n+1/2}_{i,j} + D^y_+((E+p)v)^{n+1/2}_{i+1,j}\right] + \Delta t\mathcal{S}_4.$$

Again, (3.15) implies that $\mathcal{S}_4 = 0$, and (3.19) becomes

$$E^{n+1}_{i+1/2,j+1/2} = E^{n}_{i+1/2,j+1/2}.$$

Thus, we conclude the updated numerical solution $\mathbf{u}^{n+1}_{i+1/2,j+1/2}$ is equal to the forward projected solution at time t^n, i.e., $\mathbf{u}^{n+1}_{i+1/2,j+1/2} = \mathbf{u}^n_{i+1/2,j+1/2}$. ∎

Next, we extend the surface gradient method to two-dimensional central schemes for the Euler equation with gravity systems in order to perform the forward and backward projection steps. As in the one-dimensional case, we introduce the new function $H^n(x,y) = \rho^n(x,y) - \rho^s(x,y)$, where $\rho^s(x,y) = \exp(-g_1 x - g_2 y)$ is the steady state density. Then we calculate $(\rho^{n,x}_{i,j})'$ and $(\rho^{n,y}_{i,j})'$ indirectly using the newly introduced function $H^n_{i,j}$ as $(\rho^{n,x}_{i,j})' = (H^{n,x}_{i,j})' + (\rho^{s,x}_{i,j})'$ and $(\rho^{n,y}_{i,j})' = (H^{n,y}_{i,j})' + (\rho^{s,y}_{i,j})'$ using a limiting procedure of numerical gradients. Recall that this is only necessary for the forward projection step of $\rho^n_{i,j}$, i.e., for the calculation of $\rho^n_{i+1/2,j+1/2}$ through equation (3.8). We assume that ρ^s is initially given at the cell interfaces, i.e., the $\rho^s_{i+1/2,j+1/2}$ are known and then we define the cell centered values to be

$$\rho^s_{i,j} = \frac{1}{2}\left(\overline{\rho}^s_{(i),j-1/2} + \overline{\rho}^s_{(i),j+1/2}\right) \tag{3.20}$$

and then we define the function $H(x,y)$ as

$$H^n(x,y) = H^n_{i,j} + (H^{n,x}_{i,j})'(x - x_i) + (H^{n,y}_{i,j})'(y - y_j),$$

where $\nabla H^n_{i,j} = ((H^{n,x}_{i,j})', (H^{n,y}_{i,j})')$ is a limited numerical gradient of $H^n_{i,j}$ Next, we calculate the numerical spatial partial derivatives of $\rho^n_{i,j}$ as follows:

$$(\rho^{n,x}_{i,j})' = (H^{n,x}_{i,j})' + \frac{1}{2\Delta x}\left([\![\rho]\!]^s_{(i),j+1/2} + [\![\rho]\!]^s_{(i),j-1/2}\right) \tag{3.21}$$

and

$$(\rho^{n,y}_{i,j})' = (H^{n,y}_{i,j})' + \frac{1}{2\Delta y}\left([\![\rho]\!]^s_{i+1/2,(j)} + [\![\rho]\!]^s_{i-1/2,(j)}\right). \tag{3.22}$$

Note that in the case of an equilibrium solution $\mathbf{u}^n_{i,j}$ only the values of $H^n_{i,j}$ vanish but not necessarily the values of $H^n_{i+1/2,j+1/2}$ unless we correct the values of $\rho^s_{i+1/2,j+1/2}$ in the back projection step while calculating the values of $\rho^{n+1}_{i,j}$ using (3.9). The correction we apply in this work is as follows:

$$\widetilde{\rho}^s_{i+1/2,j+1/2} = \rho^s_{i+1/2,j+1/2} - \frac{1}{2}\left[\rho^s_{i+1/2,j+1/2} - \frac{\overline{\rho}^s_{(i),j-1/2} + \overline{\rho}^s_{(i),j+1/2}}{2}\right] \tag{3.23}$$

and then we define the function $\widetilde{H}^{n+1}_{i+1/2,j+1/2} = \rho^{n+1}_{i+1/2,j+1/2} - \widetilde{\rho}^s_{i+1/2,j+1/2}$. The numerical spatial derivatives of $\rho^{n+1}_{i+1/2,j+1/2}$ can now be indirectly computed using the numerical gradient of $\widetilde{H}^{n+1}_{i+1/2,j+1/2}$ as follows:

$$(\rho^{n+1,x}_{i+1/2,j+1/2})' = (\widetilde{H}^{n+1,x}_{i+1/2,j+1/2})' + \frac{[\![\rho]\!]^s_{(i),j+1/2} + [\![\rho]\!]^s_{(i+1),j+1/2}}{2\Delta x} \tag{3.24}$$

and similarly

$$(\rho^{n+1,y}_{i+1/2,j+1/2})' = (\widetilde{H}^{n+1,y}_{i+1/2,j+1/2})' + \frac{[\![\rho]\!]^s_{i+1/2,(j)} + [\![\rho]\!]^s_{i+1/2,(j+1)}}{2\Delta y}. \tag{3.25}$$

Note that the forward and backward projection steps of the energy component are handled in a similar way.

In the forward projection step we linearize the energy $E^n_{i,j}$ at time t^n using the function $\mathcal{H}_{i,j} = E^n_{i,j} - E^s_{i,j}$ and then calculate the spatial numerical derivatives as follows:

$$(E^{n,x}_{i,j})' = (\mathcal{H}^{n,x}_{i,j})' + \frac{1}{2\Delta x}\left(\llbracket E\rrbracket^s_{(i),j+1/2} + \llbracket E\rrbracket^s_{(i),j+1/2}\right)$$

and

$$(E^{n,y}_{i,j})' = (\mathcal{H}^{n,y}_{i,j})' + \frac{1}{2\Delta y}\left(\llbracket E\rrbracket^s_{i+1/2,(j)} + \llbracket E\rrbracket^s_{i-1/2,(j)}\right).$$

Here we assume that the $E^s_{i+1/2,j+1/2}$ is the energy of the equilibrium solution and is initially given at the cell's corners and satisfying the equation $E^s_{i+1/2,j+1/2} = p^s_{i+1/2,j+1/2}/(\gamma-1)$. We define the cell centered equilibrium state energy to be

$$E^s_{i,j} = \frac{1}{2}\left(\overline{E}^s_{(i),j-1/2} + \overline{E}^s_{(i),j+1/2}\right).$$

For the back projection step the linearization is performed using the corrected values of the equilibrium energy defined as follows:

$$\widetilde{E}^s_{i+1/2,j+1/2} = E^s_{i+1/2,j+1/2} - \frac{1}{2}\left[E^s_{i+1/2,j+1/2} - \frac{1}{2}\left(\overline{E}^s_{i+1/2,j} + \overline{E}^s_{i+1/2,j+1}\right)\right].$$

Then we introduce the function $\widetilde{\mathcal{H}}^{n+1}_{i+1/2,j+1/2} = E^{n+1}_{i+1/2,j+1/2} - \widetilde{E}^s_{i+1/2,j+1/2}$ and calculate the numerical spatial partial derivatives of $E^{n+1}_{i+1/2,j+1/2}$ indirectly using numerical derivatives of the discrete functions $\widetilde{\mathcal{H}}^{n+1}_{i+1/2,j+1/2}$ and $E^s_{i,j}$ as follows:

$$(E^{n+1,x}_{i+1/2,j+1/2})' = (\widetilde{\mathcal{H}}^{n+1,x}_{i+1/2,j+1/2})' + \frac{1}{2\Delta x}\left(\llbracket E\rrbracket^s_{(i),j+1/2} + \llbracket E\rrbracket^s_{(i+1),j+1/2}\right)$$

and similarly

$$(E^{n+1,y}_{i+1/2,j+1/2})' = (\widetilde{\mathcal{H}}^{n+1,y}_{i+1/2,j+1/2})' + \frac{1}{2\Delta y}\left(\llbracket E\rrbracket^s_{i+1/2,(j)} + \llbracket E\rrbracket^s_{i+1/2,(j+1)}\right).$$

To complete the derivation of the two-dimensional well-balanced central scheme for the Euler equations with gravity system we still need to show that if the numerical solution $\mathbf{u}^n_{i,j}$ corresponds to an equilibrium state solution, then the updated solution $\mathbf{u}^{n+1}_{i,j}$ remains as such, i.e., $\mathbf{u}^n_{i,j} = \mathbf{u}^{n+1}_{i,j}$. Regarding this, we have the following theorem.

THEOREM 3.2. *Let the approximate solution $\mathbf{u}^n_{i,j}$ of the Euler equation with gravity system* (3.3) *be updated using the finite volume method* (3.7) *and* (3.9) *and under the hypotheses of Theorem* 3.1 *along with the surface-gradient-based forward projection step and backward projection step. Then the scheme* (3.9) *has the following properties:*

(a) *Accuracy: It is a second-order accurate approximation of the Euler equation with gravity system* (3.3).

(b) *Well-balanced: It preserves the steady state* (3.2)*, i.e., if $\mathbf{u}^n_i$ satisfies* (3.2) *then the updated solution $\mathbf{u}^{n+1}_i$ also satisfies* (3.2).

Proof. A straightforward truncation error analysis shows that the local truncation error is $\mathcal{O}(\Delta x^2)$ which confirms (a). To prove (b), we shall proceed componentwise and show that $\rho^n_{i,j} = \rho^{n+1}_{i,j}$ and $E^n_{i,j} = E^{n+1}_{i,j}$. Note that the equations $\rho u^n_{i,j} = \rho u^{n+1}_{i,j}$

and $\rho v^n_{i,j} = \rho v^{n+1}_{i,j}$ follow immediately because in the equilibrium state we have $u^n_{i,j} = v^n_{i,j} = 0$.

In what follows, we first calculate the forward projected solution. Recall that for both the first and fourth components of $\mathbf{u}^n_{i,j}$, the forward projection step (3.8) is performed using the surface gradient method. The forward projection step of $\rho^n_{i,j}$ at time t^n is performed using (3.8), (3.21), and (3.22) as follows:

(3.26)

$$\begin{aligned}
\rho^n_{i+\frac12,j+\frac12} = {} & \frac14\left(\rho^n_{i,j}+\rho^n_{i+1,j}+\rho^n_{i,j+1}+\rho^n_{i+1,j+1}\right)\\
& + \frac{\Delta x}{16}\Big((H^{n,x}_{i,j})' + (H^{n,x}_{i,j+1})'\\
& \qquad + \frac{1}{2\Delta x}\left(\llbracket\rho\rrbracket^s_{(i),j-\frac12} + \llbracket\rho\rrbracket^s_{(i),j+\frac12} + \llbracket\rho\rrbracket^s_{(i),j+\frac12} + \llbracket\rho\rrbracket^s_{(i),j+\frac32}\right)\Big)\\
& - \frac{\Delta x}{16}\Big((H^{n,x}_{i+1,j})' + (H^{n,x}_{i+1,j+1})'\\
& \qquad + \frac{1}{2\Delta x}\left(\llbracket\rho\rrbracket^s_{(i+1),j-\frac12} + \llbracket\rho\rrbracket^s_{(i+1),j+\frac12} + \llbracket\rho\rrbracket^s_{(i+1),j+\frac12} + \llbracket\rho\rrbracket^s_{(i+1),j+\frac32}\right)\Big)\\
& + \frac{\Delta y}{16}\Big((H^{n,y}_{i,j})' + (H^{n,y}_{i+1,j})'\\
& \qquad + \frac{1}{2\Delta y}\left(\llbracket\rho\rrbracket^s_{i-\frac12,(j)} + \llbracket\rho\rrbracket^s_{i+\frac12,(j)} + \llbracket\rho\rrbracket^s_{i+\frac12,(j)} + \llbracket\rho\rrbracket^s_{i+\frac32,(j)}\right)\Big)\\
& - \frac{\Delta y}{16}\Big((H^{n,y}_{i,j+1})' + (H^{n,y}_{i+1,j+1})'\\
& \qquad + \frac{1}{2\Delta y}\left(\llbracket\rho\rrbracket^s_{i-\frac12,(j+1)} + \llbracket\rho\rrbracket^s_{i+\frac12,(j+1)} + \llbracket\rho\rrbracket^s_{i+\frac12,(j+1)} + \llbracket\rho\rrbracket^s_{i+\frac32,(j+1)}\right)\Big).
\end{aligned}$$

Since at time t^n the numerical solution $\mathbf{u}^n_{i,j}$ corresponds to an equilibrium solution then the relation $H^n_{i,j} = \rho^n_{i,j} - \rho^s_{i,j} = 0$ is maintained for all i, j, and therefore all spatial partial numerical derivatives of $H^n_{i,j}$ in (3.26) are zero. Furthermore, (3.20) leads to

$$\begin{aligned}
\llbracket\rho\rrbracket^s_{(i),j-\frac12} + \llbracket\rho\rrbracket^s_{(i),j+\frac12} &= 2\left(\left(\rho^s_{i+\frac12,j-\frac12} + \rho^s_{i+\frac12,j+\frac12}\right) - 2\rho^s_{i,j}\right),\\
\llbracket\rho\rrbracket^s_{(i),j+\frac12} + \llbracket\rho\rrbracket^s_{(i),j+\frac32} &= 2\left(\left(\rho^s_{i+\frac12,j+\frac12} + \rho^s_{i+\frac12,j+\frac32}\right) - 2\rho^s_{i,j+1}\right),\\
\llbracket\rho\rrbracket^s_{(i+1),j-\frac12} + \llbracket\rho\rrbracket^s_{(i+1),j+\frac12} &= 2\left(2\rho^s_{i+1,j} - \left(\rho^s_{i+\frac12,j-\frac12} + \rho^s_{i+\frac12,j+\frac12}\right)\right),\\
\llbracket\rho\rrbracket^s_{(i+1),j+\frac12} + \llbracket\rho\rrbracket^s_{(i+1),j+\frac32} &= 2\left(2\rho^s_{i+1,j+1} - \left(\rho^s_{i+\frac12,j+\frac12} + \rho^s_{i+\frac12,j+\frac32}\right)\right),
\end{aligned} \tag{3.27}$$

and, similarly, using (3.20) we get the following relations:

$$\begin{aligned}
\llbracket\rho\rrbracket^s_{i-\frac12,(j)} + \llbracket\rho\rrbracket^s_{i+\frac12,(j)} &= 2\left(\left(\rho^s_{i-\frac12,j+\frac12} + \rho^s_{i+\frac12,j+\frac12}\right) - 2\rho^s_{i,j}\right),\\
\llbracket\rho\rrbracket^s_{i+\frac12,(j)} + \llbracket\rho\rrbracket^s_{i+\frac32,(j)} &= 2\left(\left(\rho^s_{i+\frac12,j+\frac12} + \rho^s_{i+\frac32,j+\frac12}\right) - 2\rho^s_{i+1,j}\right),\\
\llbracket\rho\rrbracket^s_{i-\frac12,(j+1)} + \llbracket\rho\rrbracket^s_{i+\frac12,(j+1)} &= 2\left(2\rho^s_{i,j+1} - \left(\rho^s_{i-\frac12,j+\frac12} + \rho^s_{i+\frac12,j+\frac12}\right)\right),\\
\llbracket\rho\rrbracket^s_{i+\frac12,(j+1)} + \llbracket\rho\rrbracket^s_{i+\frac32,(j+1)} &= 2\left(2\rho^s_{i+1,j+1} - \left(\rho^s_{i+\frac12,j+\frac12} + \rho^s_{i+\frac32,j+\frac12}\right)\right).
\end{aligned} \tag{3.28}$$

Substituting the relations (3.27) and (3.28) into (3.26) while taking into account

$H^n_{i,j} = 0$, we obtain

(3.29)

$$\begin{aligned}
&\rho^n_{i+\frac{1}{2},j+\frac{1}{2}} \\
&\quad= \frac{1}{4}\left(\rho^n_{i,j} + \rho^n_{i+1,j} + \rho^n_{i,j+1} + \rho^n_{i+1,j+1}\right) \\
&\qquad+ \frac{1}{16}\left((\rho^s_{i+\frac{1}{2},j-\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{1}{2}}) - 2\rho^s_{i,j}\right) + \frac{1}{16}\left((\rho^s_{i+\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{3}{2}}) - 2\rho^s_{i,j+1}\right) \\
&\qquad- \frac{1}{16}\left(2\rho^s_{i+1,j} - (\rho^s_{i+\frac{1}{2},j-\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{1}{2}})\right) \\
&\qquad- \frac{1}{16}\left(2\rho^s_{i+1,j+1} - (\rho^s_{i+\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{3}{2}})\right) \\
&\qquad+ \frac{1}{16}\left((\rho^s_{i-\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{1}{2}}) - 2\rho^s_{i,j}\right) + \frac{1}{16}\left((\rho^s_{i+\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{3}{2},j+\frac{1}{2}}) - 2\rho^s_{i+1,j}\right) \\
&\qquad- \frac{1}{16}\left(2\rho^s_{i,j+1} - (\rho^s_{i-\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{1}{2}})\right) \\
&\qquad- \frac{1}{16}\left(2\rho^s_{i+1,j+1} - (\rho^s_{i+\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{3}{2},j+\frac{1}{2}})\right)
\end{aligned}$$

which simplifies leading to

$$\begin{aligned}
\rho^n_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}\left(\rho^n_{i,j} + \rho^n_{i,j+1} + \rho^n_{i+1,j} + \rho^n_{i+1,j+1}\right) - \frac{1}{4}\left(\rho^s_{i,j} + \rho^s_{i,j+1} + \rho^s_{i+1,j} + \rho^s_{i+1,j+1}\right) \\
+ \frac{1}{8}\left(4\rho^s_{i+\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{3}{2}} + \rho^s_{i+\frac{1}{2},j-\frac{1}{2}} + \rho^s_{i-\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{3}{2},j+\frac{1}{2}}\right).
\end{aligned}$$

Similarly, for the back projection step (3.9), we follow the surface gradient method. Keeping in mind that $\widetilde{H}^{n+1}_{i+1/2,j+1/2} = \rho^{n+1}_{i+1/2,j+1/2} - \widetilde{\rho}^s_{i+1/2,j+1/2}$ and using (3.24) and (3.25) we get

(3.30)

$$\begin{aligned}
\rho^{n+1}_{i,j} &= \frac{1}{4}\left(\rho^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}} + \rho^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}} + \rho^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}} + \rho^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}\right) \\
&\quad+ \frac{\Delta x}{16}\Big((\widetilde{H}^{n+1,x}_{i-1/2,j-1/2})' + (\widetilde{H}^{n+1,x}_{i-1/2,j+1/2})' \\
&\qquad\quad+ \frac{1}{2\Delta x}\left([\![\rho]\!]^s_{(i-1),j-\frac{1}{2}} + [\![\rho]\!]^s_{(i),j-\frac{1}{2}} + [\![\rho]\!]^s_{(i-1),j+\frac{1}{2}} + [\![\rho]\!]^s_{(i),j+\frac{1}{2}}\right)\Big) \\
&\quad- \frac{\Delta x}{16}\Big((\widetilde{H}^{n+1,x}_{i+1/2,j-1/2})' + (\widetilde{H}^{n+1,x}_{i+1/2,j+1/2})' \\
&\qquad\quad+ \frac{1}{2\Delta x}\left([\![\rho]\!]^s_{(i),j-\frac{1}{2}} + [\![\rho]\!]^s_{(i+1),j-\frac{1}{2}} + [\![\rho]\!]^s_{(i),j+\frac{1}{2}} + [\![\rho]\!]^s_{(i+1),j+\frac{1}{2}}\right)\Big) \\
&\quad+ \frac{\Delta y}{16}\Big((\widetilde{H}^{n+1,y}_{i-1/2,j-1/2})' + (\widetilde{H}^{n+1,y}_{i+1/2,j-1/2})' \\
&\qquad\quad+ \frac{1}{2\Delta y}\left([\![\rho]\!]^s_{i-\frac{1}{2},(j-1)} + [\![\rho]\!]^s_{i-\frac{1}{2},(j)} + [\![\rho]\!]^s_{i+\frac{1}{2},(j-1)} + [\![\rho]\!]^s_{i+\frac{1}{2},(j)}\right)\Big) \\
&\quad- \frac{\Delta y}{16}\Big((\widetilde{H}^{n+1,y}_{i-1/2,j+1/2})' + (\widetilde{H}^{n+1,y}_{i+1/2,j+1/2})' \\
&\qquad\quad+ \frac{1}{2\Delta y}\left([\![\rho]\!]^s_{i-\frac{1}{2},(j)} + [\![\rho]\!]^s_{i-\frac{1}{2},(j+1)} + [\![\rho]\!]^s_{i+\frac{1}{2},(j)} + [\![\rho]\!]^s_{i+\frac{1}{2},(j+1)}\right)\Big).
\end{aligned}$$

Note that when $H_{i,j} = \rho^n_{i,j} - \rho^s_{i,j} = 0$, then $\widetilde{H}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = \rho^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} - \widetilde{\rho}^s_{i+\frac{1}{2},j+\frac{1}{2}}$ remains zero and therefore $(\widetilde{H}^{n+1,x}_{i+\frac{1}{2},j+\frac{1}{2}})' = (\widetilde{H}^{n+1,y}_{i+\frac{1}{2},j+\frac{1}{2}})' = 0$; (3.30) becomes

$$\begin{aligned}
\rho^{n+1}_{i,j} = {} & \frac{1}{2}\left(\overline{\widetilde{\rho}}^s_{(i),j-\frac{1}{2}} + \overline{\widetilde{\rho}}^s_{(i),j+\frac{1}{2}}\right) \\
& + \frac{1}{32}\left([\![\rho]\!]^s_{(i),j-\frac{1}{2}} + [\![\rho]\!]^s_{(i-1),j-\frac{1}{2}} + [\![\rho]\!]^s_{(i),j+\frac{1}{2}} + [\![\rho]\!]^s_{(i-1),j+\frac{1}{2}}\right) \\
& - \frac{1}{32}\left([\![\rho]\!]^s_{(i),j-\frac{1}{2}} + [\![\rho]\!]^s_{(i+1),j-\frac{1}{2}} + [\![\rho]\!]^s_{(i),j+\frac{1}{2}} + [\![\rho]\!]^s_{(i+1),j+\frac{1}{2}}\right) \\
& + \frac{1}{32}\left([\![\rho]\!]^s_{i-\frac{1}{2},(j-1)} + [\![\rho]\!]^s_{i-\frac{1}{2},(j)} + [\![\rho]\!]^s_{i+\frac{1}{2},(j-1)} + [\![\rho]\!]^s_{i+\frac{1}{2},(j)}\right) \\
& - \frac{1}{32}\left([\![\rho]\!]^s_{i-\frac{1}{2},(j)} + [\![\rho]\!]^s_{i-\frac{1}{2},(j+1)} + [\![\rho]\!]^s_{i+\frac{1}{2},(j)} + [\![\rho]\!]^s_{i+\frac{1}{2},(j+1)}\right)
\end{aligned}$$

which reduces to

$$\begin{aligned}
\rho^{n+1}_{i,j} = {} & \frac{1}{2}\left(\overline{\widetilde{\rho}}^s_{(i),j-\frac{1}{2}} + \overline{\widetilde{\rho}}^s_{(i),j+\frac{1}{2}}\right) \\
& - \frac{1}{16}\left(\overline{\rho}^s_{(i-1),j-\frac{1}{2}} + \overline{\rho}^s_{(i-1),j+\frac{1}{2}} + \overline{\rho}^s_{(i+1),j-\frac{1}{2}} + \overline{\rho}^s_{(i+1),j+\frac{1}{2}}\right) \\
& + \frac{1}{16}\left(\overline{\rho}^s_{(i),j-\frac{1}{2}} + \overline{\rho}^s_{(i),j+\frac{1}{2}} + \overline{\rho}^s_{(i),j-\frac{1}{2}} + \overline{\rho}^s_{(i),j+\frac{1}{2}}\right) \\
& - \frac{1}{16}\left(\overline{\rho}^s_{i-\frac{1}{2},(j-1)} + \overline{\rho}^s_{i+\frac{1}{2},(j-1)} + \overline{\rho}^s_{i-\frac{1}{2},(j)} + \overline{\rho}^s_{i+\frac{1}{2},(j)}\right) \\
& + \frac{1}{16}\left(\overline{\rho}^s_{i-\frac{1}{2},(j)} + \overline{\rho}^s_{i+\frac{1}{2},(j)} + \overline{\rho}^s_{i-\frac{1}{2},(j)} + \overline{\rho}^s_{i+\frac{1}{2},(j)}\right).
\end{aligned} \tag{3.31}$$

But from (3.20) we know that

$$\begin{aligned}
&\rho^s_{i,j} = \frac{1}{2}\left(\overline{\rho}^s_{(i),j-\frac{1}{2}} + \overline{\rho}^s_{(i),j+\frac{1}{2}}\right), \qquad \rho^s_{i-1,j} = \frac{1}{2}\left(\overline{\rho}^s_{(i-1),j-\frac{1}{2}} + \overline{\rho}^s_{(i-1),j+\frac{1}{2}}\right), \\
&\rho^s_{i+1,j} = \frac{1}{2}\left(\overline{\rho}^s_{(i+1),j-\frac{1}{2}} + \overline{\rho}^s_{(i+1),j+\frac{1}{2}}\right), \quad \rho^s_{i,j-1} = \frac{1}{2}\left(\overline{\rho}^s_{i-\frac{1}{2},(j-1)} + \overline{\rho}^s_{i+\frac{1}{2},(j-1)}\right), \\
&\rho^s_{i,j+1} = \frac{1}{2}\left(\overline{\rho}^s_{i-\frac{1}{2},(j+1)} + \overline{\rho}^s_{i+\frac{1}{2},(j+1)}\right).
\end{aligned} \tag{3.32}$$

Thus (3.31) becomes

$$\rho^{n+1}_{i,j} = \frac{1}{2}\left(\overline{\widetilde{\rho}}^s_{(i),j-\frac{1}{2}} + \overline{\widetilde{\rho}}^s_{(i),j+\frac{1}{2}}\right) - \frac{1}{4}\Big(\rho^s_{i-1,j} + \rho^s_{i+1,j} + \rho^s_{i,j-1} + \rho^s_{i,j+1} - 4\rho^s_{i,j}\Big). \tag{3.33}$$

On the other using (3.23) we write the following equations:

$$\begin{aligned}
\widetilde{\rho}^s_{i+1/2,j+1/2} &= \frac{1}{8}\left(4\rho^s_{i+\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{3}{2}} + \rho^s_{i+\frac{1}{2},j-\frac{1}{2}} + \rho^s_{i-\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{3}{2},j+\frac{1}{2}}\right), \\
\widetilde{\rho}^s_{i-1/2,j-1/2} &= \frac{1}{8}\left(4\rho^s_{i-\frac{1}{2},j-\frac{1}{2}} + \rho^s_{i-\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i-\frac{1}{2},j-\frac{3}{2}} + \rho^s_{i-\frac{3}{2},j-\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j-\frac{1}{2}}\right), \\
\widetilde{\rho}^s_{i-1/2,j+1/2} &= \frac{1}{8}\left(4\rho^s_{i-\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i-\frac{1}{2},j+\frac{3}{2}} + \rho^s_{i-\frac{1}{2},j-\frac{1}{2}} + \rho^s_{i-\frac{3}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{1}{2}}\right), \\
\widetilde{\rho}^s_{i+1/2,j-1/2} &= \frac{1}{8}\left(4\rho^s_{i+\frac{1}{2},j-\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j+\frac{1}{2}} + \rho^s_{i+\frac{1}{2},j-\frac{3}{2}} + \rho^s_{i-\frac{1}{2},j-\frac{1}{2}} + \rho^s_{i+\frac{3}{2},j-\frac{1}{2}}\right).
\end{aligned}$$

Adding these together and using the relations in (3.32), we obtain

$$\left(\overline{\widetilde{\rho}}^s_{(i),j-\frac{1}{2}} + \overline{\widetilde{\rho}}^s_{(i),j+\frac{1}{2}}\right) = \frac{1}{16}\Big(4\rho^s_{i,j-1} + 4\rho^s_{i-1,j} + 4\rho^s_{i,j+1} + 4\rho^s_{i+1,j} + 16\rho^s_{i,j}\Big).$$

Therefore (3.33) becomes

$$\begin{aligned}\rho_{i,j}^{n+1} &= \frac{1}{32}\Big[4\rho_{i,j-1}^{s} + 4\rho_{i-1,j}^{s} + 4\rho_{i,j+1}^{s} + 4\rho_{i+1,j}^{s} + 16\rho_{i,j}^{s}\Big] \\ &\quad - \frac{1}{8}\Big(\rho_{i,j-1}^{s} + \rho_{i-1,j}^{s} + \rho_{i,j+1}^{s} + \rho_{i+1,j}^{s} - 4\rho_{i,j}^{s}\Big) \\ &= \rho_{i,j}^{s} = \rho_{i,j}^{n}. \end{aligned} \tag{3.34}$$

This means that if the numerical solution $\rho_{i,j}^{n}$ at time t^n is an equilibrium state solution, then the updated solution $\rho_{i,j}^{n+1}$ is such that $\rho_{i,j}^{n+1} = \rho_{i,j}^{n}$.

To keep the presentation fairly short we have only provided details for the component ρ. However, the proofs for the component E are similar and we omit the proofs for brevity in the exposition. Hence, combining all the results, we conclude that $\mathbf{u}_{i,j}^{n+1} = \mathbf{u}_{i,j}^{n}$. ∎

4. Numerical experiments. The fully discrete finite volume schemes developed in sections 2 and 3 have been tested on suitable numerical experiments in order to demonstrate their effectiveness.

4.1. One-dimensional experiments. We first validate the one-dimensional well-balanced central scheme for the Euler equation with gravity equations and solve some classical problems from the recent literature. In what follows, we compare our results with the numerical results of a fifth-order well-balanced finite difference WENO (weighted essentially nonoscillatory) method given in [16]. Note that, as is the case in central finite volume methods, and for stability purposes of the numerical scheme, the time step Δt is dynamically calculated using the eigenvalues $\lambda_k, k = 1, \ldots, p$ of the Jacobian matrix $\partial f(\mathbf{u})/\partial \mathbf{u}$ as follows:

$$\Delta t = CFL \frac{\Delta x}{\max(|\lambda_k|)},$$

where $0 \leq CFL \leq 0.5$ is the Current–Friedrichs–Lewy stability parameter.

4.1.1. Shock tube under gravitational field. We consider, for our first numerical experiment, Sod's standard shock tube problem with gravitational field (1.2) as considered in [16]. The computational domain is the interval $[0, 1]$, and the initial conditions feature two constant states $U_l = [1, 0, 1]$ if $x \leq 0.5$ and $U_r = [0.125, 0, 0.1]$ if $x \geq 0.5$, where $U = [\rho, u, p]$. The gravitational field ϕ is such that $\phi_x = g = 1$ and the gas adiabatic constant γ is equal to 1.4. The well-balanced scheme is applied with reflective boundary conditions, and the numerical solution is computed at time $t = 0.2$ on 100, 200, and 400 grid points. The obtained results are reported in Figure 1 where we plot the pressure, density, momentum, and velocity. It is worth mentioning that the obtained numerical results are in good agreement with those presented in [16] in the sense that we also see that due to the gravitational force, the density distribution is pulling towards the left direction, and negative velocity appears in some regions. In order to validate the proposed numerical scheme we solved this same shock tube problem but with zero gravity; the resulting problem is the classical Sod shock tube problem. The obtained numerical solution at the final time time $t_f = 0.164$ using the proposed scheme (dotted curve) are compared to the exact solution (solid curve) of the Riemann problem and are reported in Figure 2.

4.1.2. Isothermal equilibrium solution. In this experiment, we validate the well-balancing property of the proposed scheme. This test case was initially proposed by LeVeque and Bale in [13] and later considered in [18] and [19] in order to demonstrate the capability of the numerical scheme to capture small perturbations

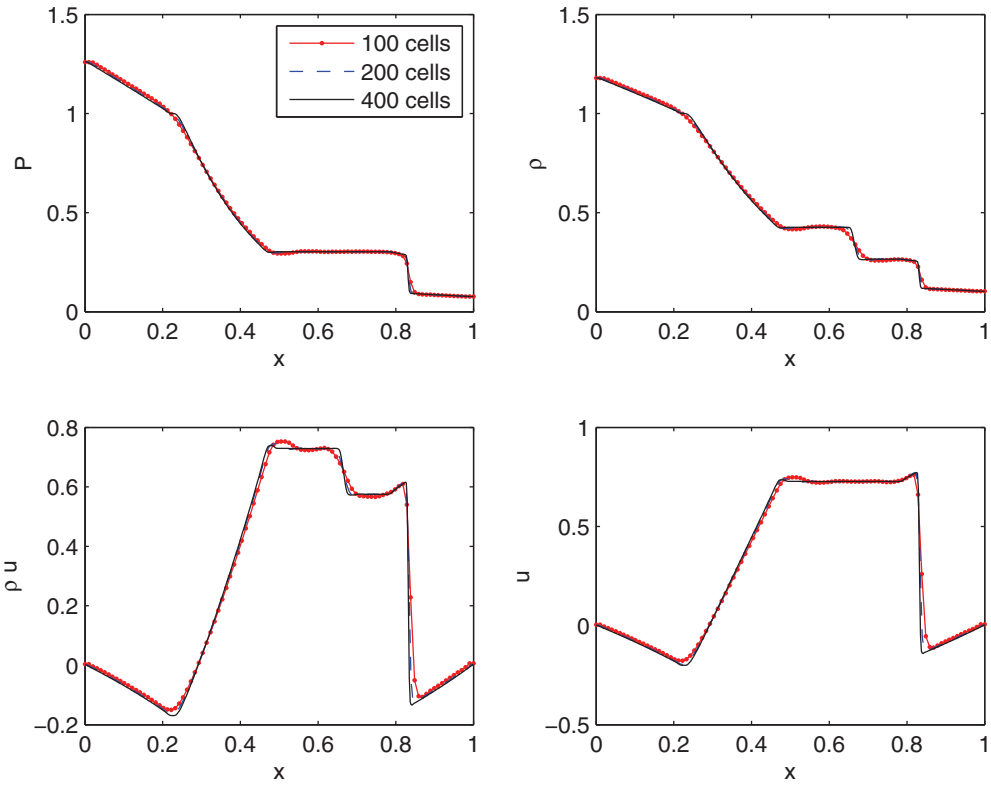

FIG. 1. *Solution of the one-dimensional shock tube problem with gravity at time* $t = 0.2$.

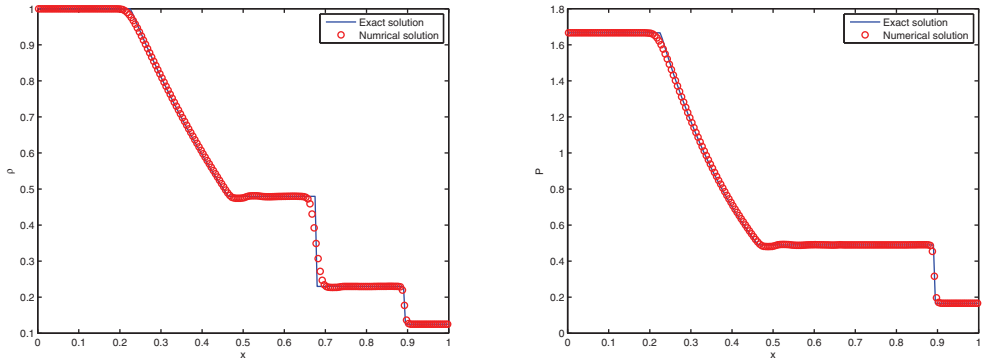

FIG. 2. *Solution of the one-dimensional Sod shock tube problem without gravity at time* $t = 0.164$.

of a steady state. The computational domain is the interval $[0, 1]$, and a linear gravitational field is considered with $\phi_x = g = 1$. Furthermore, we assume an ideal gas with constant $\gamma = 1.4$; the corresponding isothermal equilibrium solution is, therefore, $\rho(x, t) = \rho_0(x) = \exp(-x)$, $p(x, t) = p_0(x) = \exp(-x)$, and $u_0(x) = 0$ for all x, $t \geq 0$. The initial conditions are taken to be exactly the same as the steady state solution; we compute the numerical solution on 200 grid points until the final time $t = 0.25$. Figure 3 shows the profile of the density ρ (top left), momentum (top right), energy

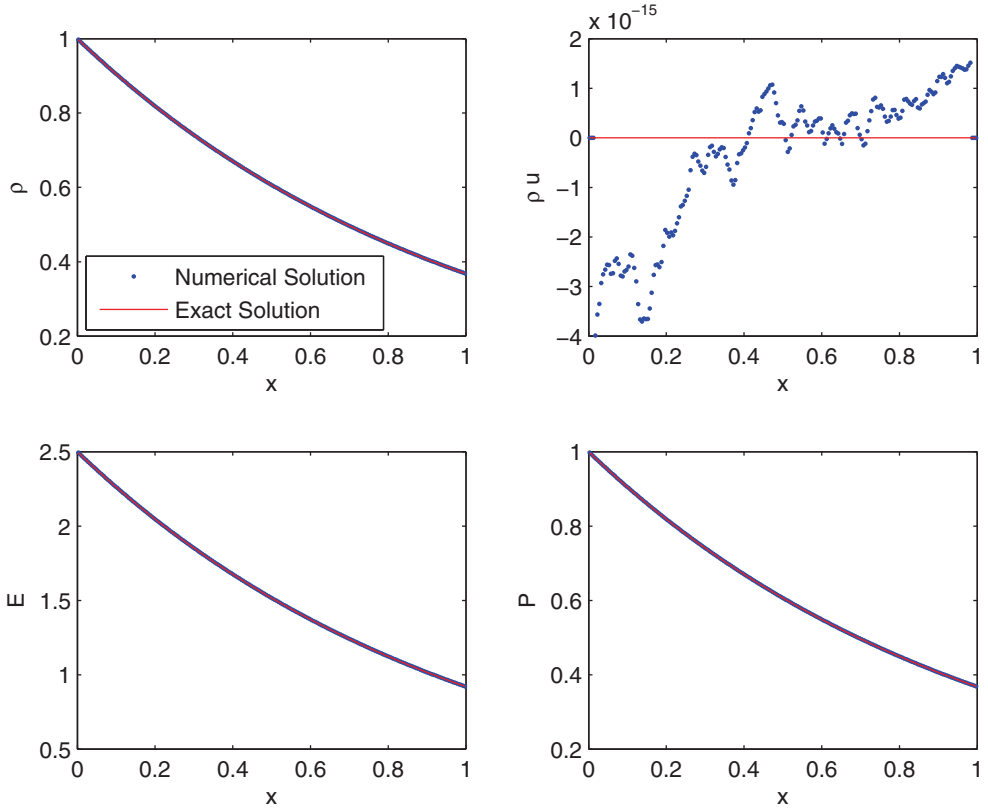

FIG. 3. *Solution of the one-dimensional equilibrium state problem at time* $t = 0.2$ *on* 200 *grid points.*

TABLE 1
L^1 *errors.*

N	L_1error ρ	Order	L_1error ρu	Order	L_1error ρE	Order
100	3.48E-15		1.90E-15		7.22E-13	
200	9.39E-16	1.89	5.25E-16	1.86	1.97E-13	1.87
400	2.33E-16	2.01	1.34E-16	1.97	5.01E-14	1.98

(bottom left), and pressure (bottom right) obtained using the proposed well-balanced scheme (dotted curve); the reference solution (solid curve) is the graph of the exact solution (steady state solution). The numerical solution remains a good match with the reference solution, thus confirming the well-balancing property of the proposed scheme. From the graph of the momentum, we see that the solution curve remains stationary with a velocity that does not exceed 2×10^{-15} in absolute value. The L^1 error and the order of convergence of the mass density and pressure on an increasing mesh were computed; the obtained results reported in Table 1 validate the order of convergence of the numerical scheme. Moreover, we see that L^1 errors are comparable with the L^1 errors presented in [16, Table 5.1].

4.1.3. Perturbation of an equilibrium solution. Our next experiment features a perturbation of the equilibrium state to demonstrate the effectiveness of our scheme. The perturbation is imposed on the initial pressure and the initial conditions

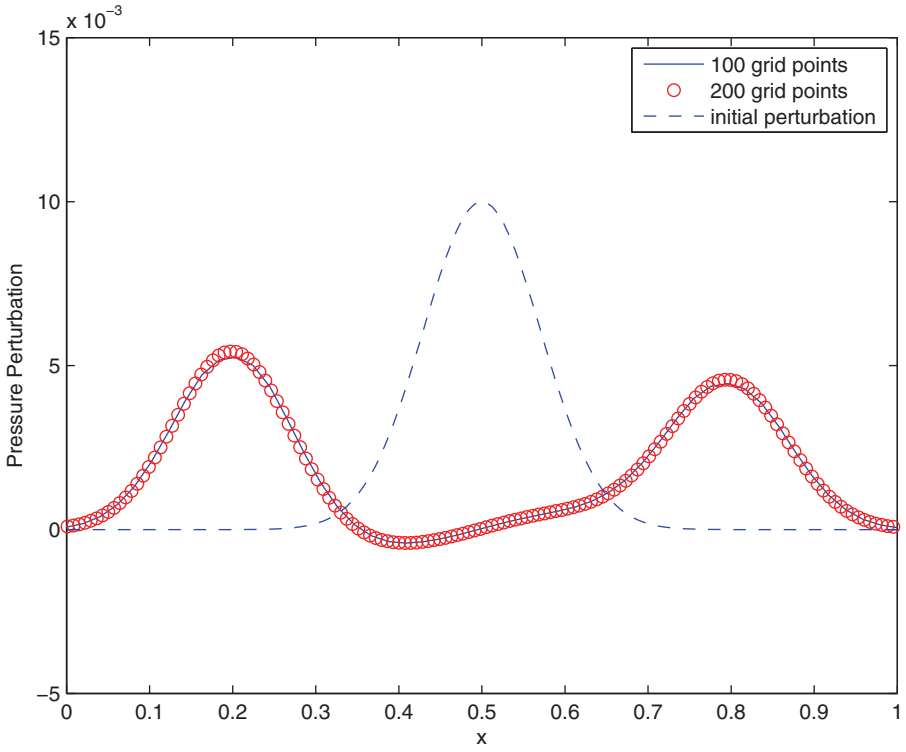

FIG. 4. *Perturbation of an equilibrium state; profile of the pressure disturbance at time* $t = 0.2$ *on* 100 *and* 200 *grid points.*

are set as follows:

$$\rho(x, t = 0) = \rho_0(x) = \exp(-x),$$
$$p(x, t = 0) = p_0(x) = \exp(-x) + \eta \exp(-100(x - 0.5)^2),$$

and the initial velocity is $u(x, t = 0) = u_0(x) = 0$. As in [16] we set $\eta = 0.01$ and compute the numerical solution until time $t = 0.25$ on 200, 400 grid points. The disturbance due to the perturbation splits into two waves propagating towards the endpoints of the computational domain where simple transmissive boundary conditions are set. The obtained results are shown in Figure 4, where we plot the pressure perturbation. Moreover, these results confirm that our method is good enough to capture small as well as large perturbations with a coarse mesh of 200 mesh points. We have also compared the numerical results obtained using our proposed scheme to those obtained using a well-balanced approximate Riemann solver [5] especially designed for the Euler equations with gravitation on 20,000 points. The comparison is reported in Figure 5 where we show the graph of the pressure perturbation on 100 grid points (dotted-dashed curve), 200 grid points (dotted curve), and the reference solution obtained on 20,000 grid points (solid curve). A good agreement between the results is observed in Figure 5. Figure 6 shows the profile of the pressure disturbance of the same problem obtained using $\eta = 0.0001$. We note also that the Riemann-solver-free nature of our scheme allows us to avoid the resolution of the Riemann problems arising at the cell interfaces, as compared to [5].

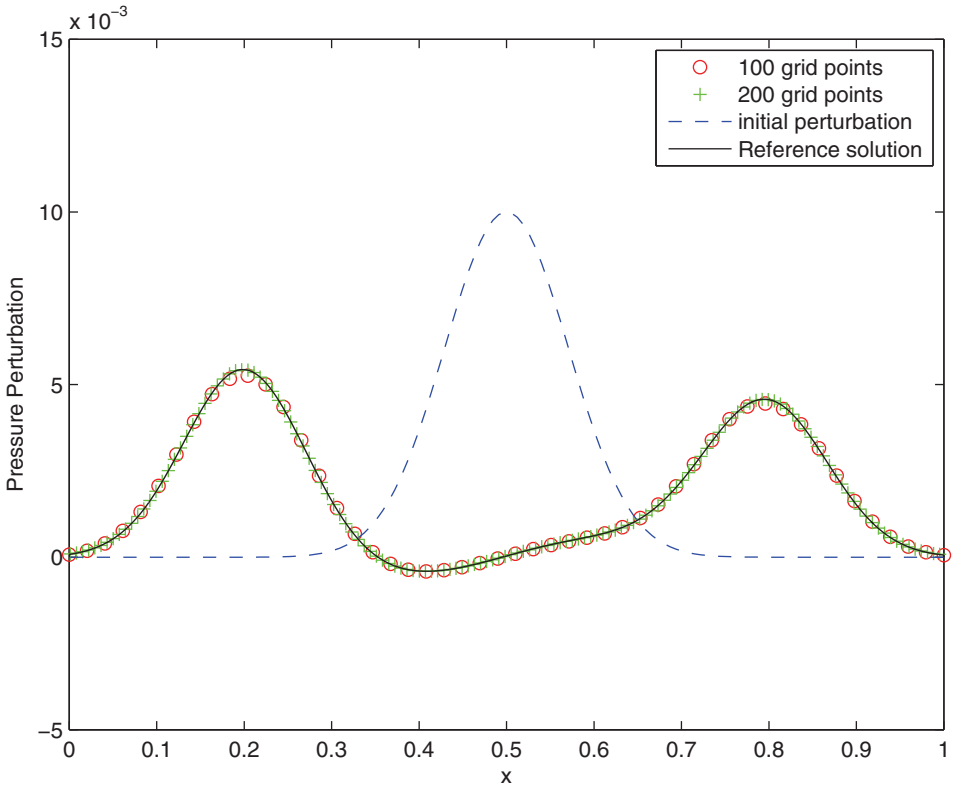

FIG. 5. *Perturbation of an equilibrium state; profile of the pressure disturbance at time* $t = 0.2$ *on* 100 *and* 200 *grid points. The reference solution was obtained on a fine grid using the method described in* [5].

4.2. Two-dimensional experiments. We now validate the two-dimensional well-balanced central scheme we developed for the Euler equations with gravity and solve classical problems from the recent literature.

4.2.1. Two-dimensional shock tube problem. This experiment features a two-dimensional extension of the shock tube problem considered in [16]. The computational domain is the square $[0,1]^2$, and the initial conditions feature two constant states $U_l = [1,0,0,1]$ if $x \leq 0.5$ and $U_r = [0.125,0,0,0.1]$ if $x \geq 0.5$, where $U = [\rho, u, v, p]$. The numerical solution of system (3.3) is computed at the final time $t_f = 0.2$ using reflective boundary conditions, and the obtained numerical results are reported in Figure 7 where we show the profiles of the mass density (left) and the pressure (right) along the x-axis (solid curve); the solution of the corresponding one-dimensional problem, also shown in the graphs, and both one- and two-dimensional schemes show a good agreement.

4.2.2. Isothermal equilibrium solution and order of convergence. This test case is used to validate the well-balanced property of the proposed two-dimensional scheme, i.e., its capability of maintaining equilibrium states at the discrete level. As in [16], we consider for our computational domain the unit square $[0,1]^2$, we set the gas constant $\gamma = 1.4$, and we set $g_1 = g_2 = 1$. The isothermal equilibrium state under

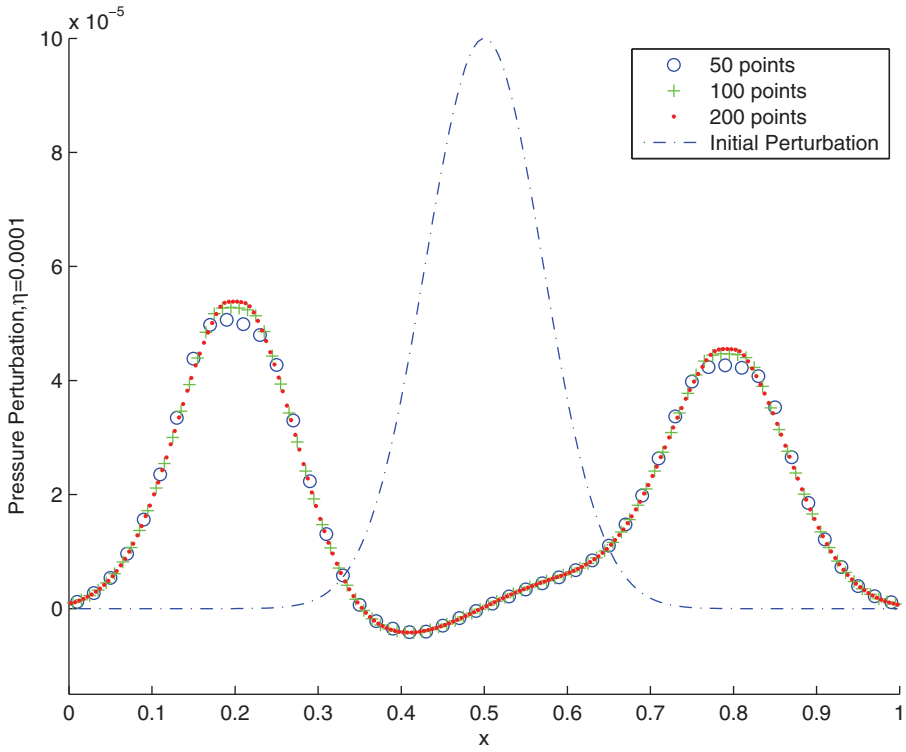

FIG. 6. *Perturbation of an equilibrium state; profile of the pressure disturbance at time* $t = 0.25$ *on* 50, 100, *and* 200 *grid points.*

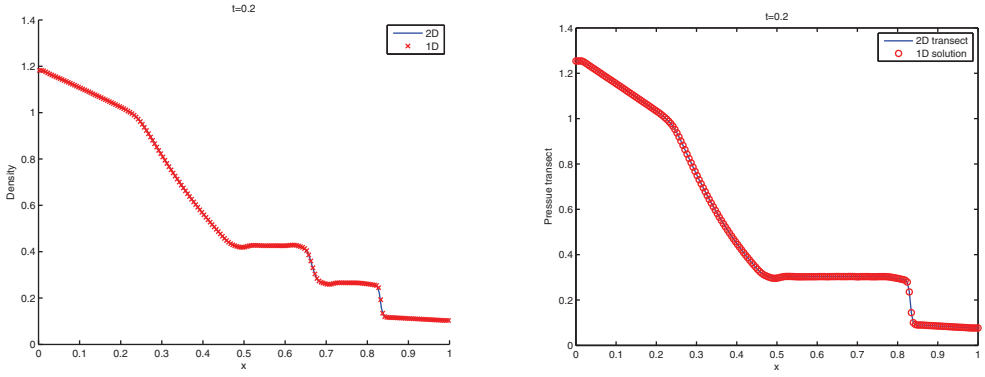

FIG. 7. *Two-dimensional shock tube problem: comparison between a cross section along the x-axis of the two-dimensional solution versus the solution of the one-dimensional problem, both computed on* 200 *grid points.*

consideration then takes the form

$$\rho(x,y) = \rho_0 \exp\left(-\frac{\rho_0}{p_0}(g_1 x + g_2 y)\right), u(x,y) = v(x,y) = 0,$$
$$p(x,y) = p_0 \exp\left(-\frac{\rho_0}{p_0}(g_1 x + g_2 y)\right)$$

with the parameters $\rho_0 = 1.21$ and $p_0 = 1$. We set the initial conditions to be exactly

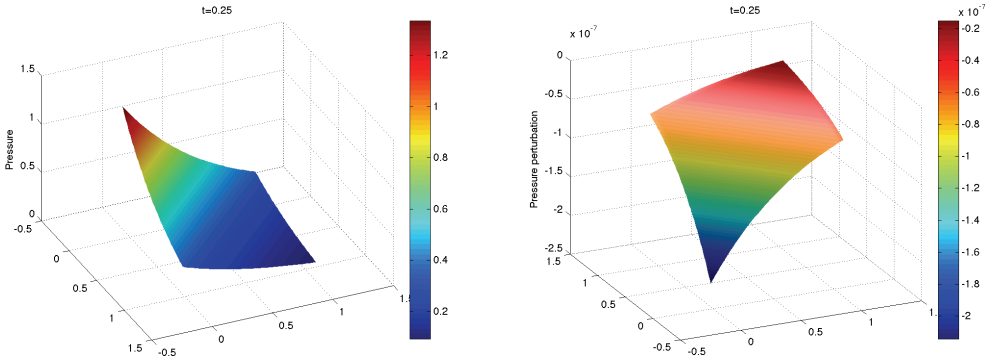

FIG. 8. *Profiles of the pressure at the final time (left), and the pressure's deviation from its steady state (right) on* 60×60 *grid points.*

TABLE 2
L^1 *errors.*

N	L^1error ρ	Order	L^1error ρu	Order	L^1error ρv	Order	L^1error ρE	Order
10^2	1.757 e-03		6.916e-08		6.916e-08		3.631e-03	
20^2	4.436e-04	1.986	1.693 e-08	2.030	1.693e-08	2.030	9.166e-04	1.986
40^2	1.112e-04	1.997	4.209e-09	2.008	4.209e-09	2.008	2.297-04	1.997

the equilibrium state solution of the problem and we compute the numerical solution on 60×60 grid points. The obtained numerical results at time $t_f = 0.25$ are reported in Figure 8 showing the profile of the pressure at the final time (left) and the pressure's deviation from its equilibrium state (right). Figure 8 (right) shows that the obtained numerical solution satisfies the steady state requirement at the discrete level and the pressure remains within 10^{-7} from its equilibrium state value, thus confirming the capability of the proposed scheme to handle the problem of equilibrium state solution for the Euler equation with gravity system. The L^1 error and the order of convergence of the mass density and pressure on an increasing mesh were computed; the results obtained are reported in Table 2. They validate the order of convergence of the numerical scheme.

4.2.3. Perturbation of an isothermal equilibrium solution. In this test case, we introduce a perturbation to the equilibrium problem. We follow the same configuration as in the previous test case, as well as in [16] and we consider the pressure perturbation

$$p(x, y, t = 0) = p_0 \exp\left(-\frac{\rho_0}{p_0}(g_1 x + g_2 y)\right) + \eta \exp\left(-\frac{100\rho_0}{p_0}((g_1 x - 0.3)^2 + (g_2 y - 0.3)^2)\right)$$

centered at the point $(0.3, 0.3)$ with η the nonzero parameter and set as 0.001 in this experiment. We compute the numerical solution until time $t_f = 0.15$ on 50×50 grid points and we consider simple transmissive boundary conditions. The obtained numerical results are reported in Figure 9 where we show the profile of density perturbation (left) and its contour lines (right) and in Figure 10 where we show the profile of the pressure perturbation (left) and its contour lines (right). The obtained numerical results are in good agreement with those presented in [16], thus confirming

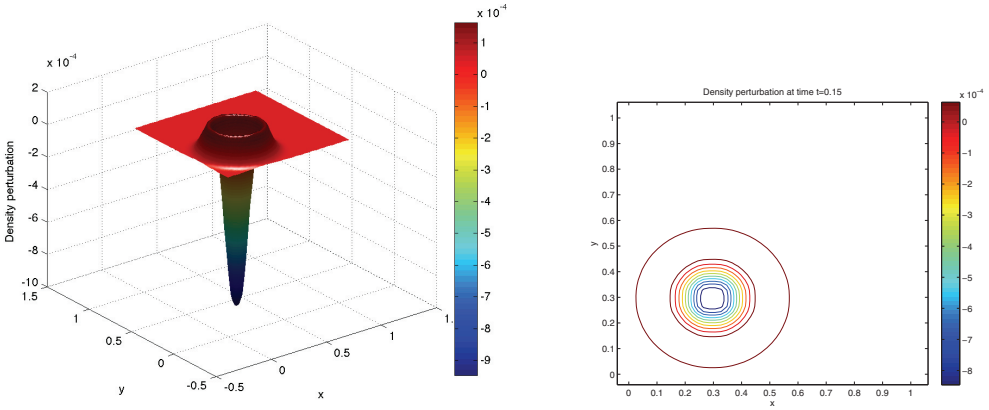

FIG. 9. *Profile of the perturbation of an equilibrium state; density disturbance at time* $t = 0.25$ *on* 50×50 *grid points.*

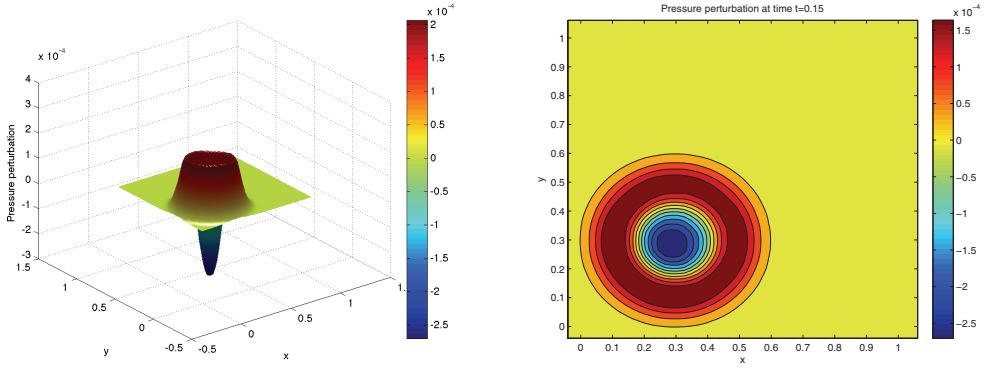

FIG. 10. *Profile of the perturbation of an equilibrium state; pressure disturbance at time* $t = 0.15$ *on* 50×50 *grid points.*

the potential of the proposed scheme to handle small perturbations of equilibrium solutions.

4.2.4. Unidirectional perturbation of an equilibrium solution. In this test case, we consider a two-dimensional extension of the one-dimensional perturbation problem of an equilibrium state considered previously. We consider for our computational domain the unit square which we discretize using 50 grid points. A unidirectional gravitational field is considered with constants $g_1 = 1$ and $g_2 = 0$. The corresponding isothermal equilibrium solutions for the density and the pressure are $\rho(x, y, t) = \rho_0(x, y, t = 0) = \exp(-g_1 x)$ and $p(x, y, t) = p_0(x, y, t = 0) = \exp(-g_1 x)$, respectively. The initial conditions are set to be the equilibrium solution for the density, $u = v = 0$, and a small perturbation of the equilibrium pressure defined by

$$p_0(x, y, t = 0) = \exp(-g_1 x) + \eta \exp(-100(x - 0.5)^2)$$

with $\eta = 0.001$. The numerical solution is computed at the final time $t_f = 0.25$, and the obtained results are reported in Figure 11 where we show the density perturbation (left) and the pressure perturbation (right). A comparison between a cross section along the line $y = 0.5$ of the pressure perturbation (dotted curve) and the solution of

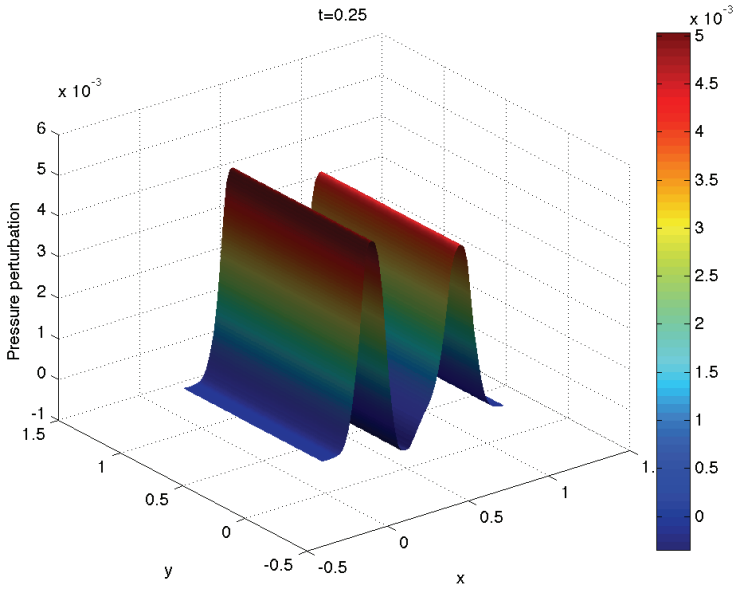

FIG. 11. *Profile of the pressure perturbation of the equilibrium state problem obtained at time* $t = 0.25$ *on* 50×50 *grid points.*

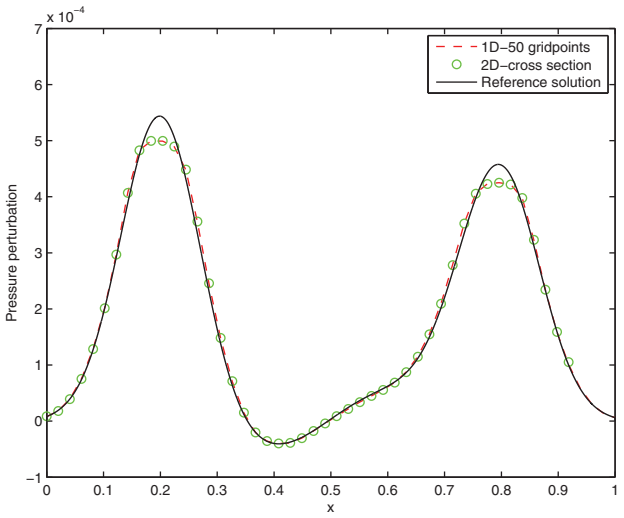

FIG. 12. *Cross section of the pressure perturbation along the line* $y = 0.5$ *obtained using the two-dimensional numerical scheme (*$\circ$ *markers); the solid curve is the solution of the corresponding one-dimensional problem with* $\eta = 0.001$.

the corresponding one-dimensional problem (dashed curve) is given in Figure 12; both curves are in good agreement with the reference solution (solid line) obtained using the solver developed in [5] on 20,000 grid points, thus confirming the potential of the proposed schemes to handle the two-dimensional Euler equation with gravity problems.

4.2.5. Circular Riemann problem. For the final experiment, we consider a circular Riemann problem subject to a gravitational field. The computational

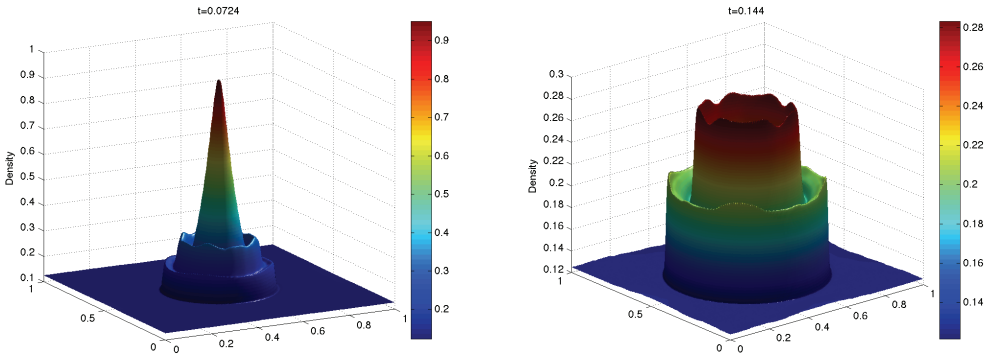

FIG. 13. *Circular Riemann problem: profile of the density at time* $t = 0.0724$ *(left) and* $t = 0.144$ *(right) on* 50×50 *grid points. Two circular shock waves are propagating outward and an rarefaction wave is moving towards the center of the computational domain to form a downward shock wave.*

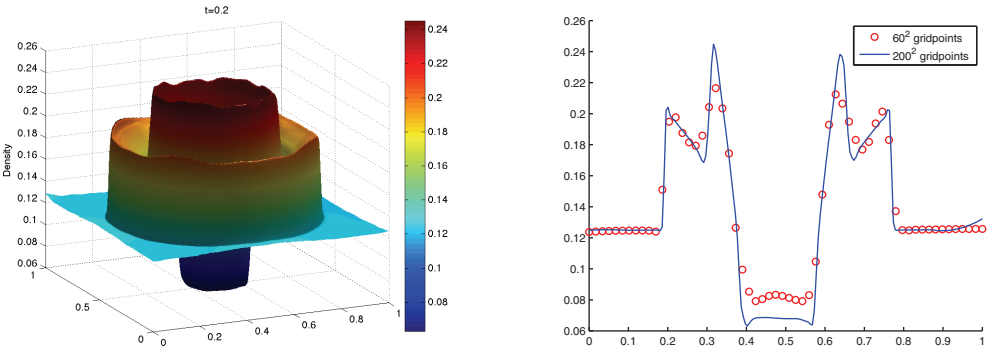

FIG. 14. *Circular Riemann problem: profile of the density at the final time* $t_f = 0.2$ *(left) and two cross sections of the density along the line* $y = x$ *obtained on* 60^2 *and* 200^2 *grid points.*

domain is the unit square, and the initial conditions feature two constant states $U_{in} = [\rho, \rho u, \rho v, E] = [1, 0, 0, 1]$ and $U_{out} = [0.125, 0, 0, 0.1]$ separated by the circle centered at the point $(0.5, 0.5)$ with a radius $r = 0.1$. The gravitational constants are $g_1 = g_2 = 1$. The numerical solution is computed at the final time $t_f = 0.2$ using the proposed well-balanced scheme and the obtained results on 50^2 grid points are reported in Figures 13 and 14 (left) where we show the profiles of the mass density at different times. The solution at time $t = 0.0742$ (Figure 13 (left)) shows two circular shocks propagating outward and a rarefaction wave is propagating towards the center of the computational domain. The shock waves are further developed at time $t = 0.144$ (Figure 13 (right)) and the rarefaction is about to become a downward shock wave. Figure 14 (left) shows the profile of the density at the final time $t_f = 0.2$. Figure 14 (right) shows two cross sections of the mass density along the line $y = x$ obtained on 60^2 and 200^2 grid points. Both curves are in good agreement thus confirming the potential of the proposed scheme to handle the Euler equation with gravity problems.

5. Conclusion. In this work we developed a well-balanced unstaggered central finite volume method for the numerical solution of systems of Euler equations with gravity in one- and two-space dimensions. The proposed method is shown to satisfy exactly the isothermal equilibrium at the discrete level and is characterized by its

simplicity. In fact the proposed method avoids the resolution of the Riemann problems arising at the cell interfaces thanks to staggered dual cells intermediately used while updating the numerical solution. Careful projections of the updated solutions back onto the original cells retrieves the solution values at the cell centers. To ensure well-balancing, sensor functions are carefully used to discretize the source term of the Euler equation with gravity system according to the discretization of the divergence of the flux function; furthermore a special adaptation of the surface gradient method is employed for the forward and backward projections of the linearly defined numerical solution. The proposed scheme is then validated and successfully applied to solve classical problems arising in the recent literature; the obtained numerical results are in very good agreement with their corresponding ones appearing in the literature thus confirming the potential of the proposed schemes to handle isothermal systems of the Euler equation with gravity equations for gas dynamics.

REFERENCES

[1] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame, *A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows*, SIAM J. Sci. Comput., 25 (2004), pp. 2050–2065.

[2] N. Botta, S. Langenberg, R. Klein, and S. Lützenkirchen, *Well balanced finite volume methods for nearly hydrostatic flows*, J. Comput. Phys., 196 (2004), pp. 539–565.

[3] M. Castro, J. M. Gallardo, and C. Parés, *High order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. Applications to shallow-water systems*, Math. Comp., 75 (2006), pp. 1103–1134.

[4] N. Črnjarić-Žic, S. Vuković, and L. Sopta, *Balanced central NT schemes for the shallow water equations*, in Proceedings of the Conference on Applied Mathematics and Scientific Computing, Springer, Dordrecht, the Netherlands, 2005, pp. 171–185.

[5] V. Desveaux, M. Zenk, C. Berthon, and C. Klingenberg, *A well-balanced scheme to capture non-explicit steady states in the Euler equations with gravity*, Internat. J. Numer. Methods Fluids, 81 (2016), pp. 104–127.

[6] F. Fuchs, A. McMurry, S. Mishra, N. H. Risebro, and K. Waagan, *High order well-balanced finite volume schemes for simulating wave propagation in stratified magnetic atmospheres*, J. Comput. Phys., 229 (2010), pp. 4033–4058.

[7] H. T. Janka, K. Langanke, A. Marek, G. Martínez-Pinedo, and B. Müller, *Theory of core-collapse supernovae*, Phys. Rep., 442 (2007), pp. 38–74.

[8] R. Käppeli and S. Mishra, *Well-balanced schemes for the Euler equations with gravitation*, J. Comput. Physics., 259 (2014), pp. 199–219.

[9] D. Kröner and M. D. Thanh, *Numerical solutions to compressible flows in a nozzle with variable cross-section*, SIAM J. Numer. Anal., 43 (2005), pp. 796–824.

[10] P. G. Lefloch and M. D. Thanh, *The Riemann problem for fluid flows in a nozzle with discontinuous cross-section*, Commun. Math. Sci, 1 (2003), pp. 763–797.

[11] R. J. LeVeque, *A well-balanced path-integral f-wave method for hyperbolic problems with source terms*, J. Sci. Comput., 48 (2011), pp. 209–226.

[12] R. J. LeVeque, *Balancing source terms and flux gradients in high-resolution Godunov methods: The quasi-steady wave-propagation algorithm*, J. Comput. Phys., 146 (1998), pp. 346–365.

[13] R. J. LeVeque and D. S. Bale, *Wave propagation methods for conservation laws with source terms*, in Proceedings of the 7th International Conferences on Hyperbolic Problems, 1998, Birkhäuser, Basel, 1999, pp. 609–618.

[14] H. Nessyahu and E. Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, J. Comput. Phys., 87 (1990), pp. 408–463.

[15] S. Noelle, Y. Xing, and C.-W. Shu, *High-order well-balanced finite volume WENO schemes for shallow water equation with moving water*, J. Comput. Phys., 226 (2007), pp. 29–58.

[16] Y. Xing and C.-W. Shu, *High order well-balanced WENO scheme for the gas dynamics equations under gravitational fields*, J. Sci. Comput., 54 (2013), pp. 645–662.

[17] J. G. Zhou, D. M. Causon, C. G. Mingham, and D. M. Ingram, *The surface gradient method for the treatment of source terms in the shallow water equations*, J. Comput. Phys., 168 (2001), pp. 1–25.

[18] J. Luo, K. Xu, and N. Liu, *A well-balanced symplecticity-preserving gas-kinetic scheme for hydrodynamic equations under gravitational field*, SIAM J. Sci. Comput., 33 (2011), pp. 2356–2381.

[19] C. T. Tian, K. Xu, K. L. Chan, and L. C. Deng, *A three-dimensional multidimensional gas-kinetic scheme for the Navier-Stokes equations under gravitational fields*, J. Comput. Phys., 226 (2007), pp. 2003–2027.

[20] G. Li and Y. Xing, *High order finite volume WENO schemes for the Euler equations under gravitational fields*, J. Comput. Phys., 316 (2016), pp. 145–163.

[21] R. Touma and S. Khankan, *Well-balanced unstaggered central schemes for one and two-dimensional shallow water equation systems*, Appl. Math. Comput., 218 (2012), pp. 5948–5960.

[22] R. Touma and C. Klingenberg, *Well-balanced central finite volume methods for the Ripa system*, Appl. Numer. Math., 2015, pp. 42–68.