



**HAL**  
open science

## A composite finite volume scheme for the Euler equations with source term on unstructured meshes

Mohamed Boujoudar, Emmanuel Franck, Philippe Hoch, Clément Lasuen,  
Yoann Le Henaff, Paul Paragot

► **To cite this version:**

Mohamed Boujoudar, Emmanuel Franck, Philippe Hoch, Clément Lasuen, Yoann Le Henaff, et al..  
A composite finite volume scheme for the Euler equations with source term on unstructured meshes.  
ESAIM: Proceedings and Surveys, In press, pp.1-22. hal-04543142

**HAL Id: hal-04543142**

**<https://hal.science/hal-04543142>**

Submitted on 11 Apr 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

## CEMRACS PROJECT: A COMPOSITE FINITE VOLUME SCHEME FOR THE EULER EQUATIONS WITH SOURCE TERM ON UNSTRUCTURED MESHES\*

M. BOUJODAR<sup>1</sup>, E. FRANCK<sup>2</sup>, P. HOCH<sup>3</sup>, C. LASUEN<sup>4</sup>, Y. LE HÉNAFF<sup>5</sup> AND P. PARAGOT<sup>6</sup>

**Abstract.** In this work we focus on an adaptation of the method described in [1] in order to deal with source term in the 2D Euler equations. This method extends classical 1D solvers (such as VFFC, Roe, Rusanov) to the two-dimensional case on unstructured meshes. The resulting schemes are said to be *composite* as they can be written as a convex combination of a purely node-based scheme and a purely edge-based scheme. We combine this extension with the ideas developed by Alouges, Ghidaglia and Tajchman in an unpublished work [2] – focused mainly on the 1D case – and we propose two attempts at discretizing the source term of the Euler equations in order to better preserve stationary solutions. We compare these discretizations with the “usual” centered discretization on several numerical examples.

### CONTENTS

1. Introduction	1
2. 1D Scheme	3
2.1. Flux based scheme	3
2.2. Well Balanced (Enhanced consistency)	4
3. 2D Schemes	5
3.1. Edge-based schemes	7
3.2. Generalization to the nodes and composite scheme	8
3.3. Boundary conditions	9
3.4. Properties of the composite scheme	9
4. Discretization of the source term	10
4.1. Well Balanced (Enhanced consistency) in 2D	10
4.2. Anti-derivative using a Poisson equation	11
4.3. Sub-volumes upwinding	11
5. Numerical results	12
5.1. 1D gravity	13

\* We would like to warmly thank the organizing committee of the CEMRACS 2022.

<sup>1</sup> Mohammed VI Polytechnic University, Morocco

<sup>2</sup> Université de Strasbourg, CNRS, Inria, IRMA, F-67000 Strasbourg, France

<sup>3</sup> CEA, DAM, DIF, 91297, Arpajon Cedex, France

<sup>4</sup> CEA, DAM, DIF, 91297, Arpajon Cedex, France

<sup>5</sup> Université de Rennes, IRMAR UMR 6625, Centre INRIA de l'Université de Rennes (MINGuS), France

<sup>6</sup> Laboratoire J.A. Dieudonné, Université Côte d'Azur, France

5.2. 2D gravity	13
5.3. Friction	13
5.4. 2D Riemann problem	14
6. Conclusion	15
A. Construction of the diffusion scheme (30)	18
References	21

## 1. INTRODUCTION

This work is concerned with the discretisation of hyperbolic systems with source term using finite volume methods on an unstructured mesh. Here we will concentrate on the Euler equations with source term which model the evolution of a gas:

$$\partial_t U + \operatorname{div} F(U) = S(\mathbf{x}, U). \quad (1)$$

The unknown is  $U = (\rho, \rho \mathbf{u}, \rho E)^T \in \mathbb{R}^4$ . The density of the fluid is  $\rho \in \mathbb{R}_+$ , its velocity is  $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$  and its total energy is  $E \in \mathbb{R}_+$ . The space variable is denoted by  $\mathbf{x} \in \mathbb{R}^2$ . The *physical flux*  $F$  reads:

$$F(U) = (F_1(U) \ F_2(U)) = \begin{pmatrix} \rho u_1 & \rho u_2 \\ \rho u_1^2 + P & \rho u_1 u_2 \\ \rho u_1 u_2 & \rho u_2^2 + P \\ (\rho E + P) u_1 & (\rho E + P) u_2 \end{pmatrix}, \quad P = (\gamma - 1)\rho \left( E - \frac{1}{2} \|\mathbf{u}\|^2 \right). \quad (2)$$

Fluxes are an important component of physical models, and as such their discretization in numerical models is essential. Finite volume schemes are based on the idea of “numerical fluxes”, and when arbitrary polygonal meshes are used, one can decide to approximate the numerical fluxes either at the nodes or edges of the computational cells. The first approach is called *pure nodal finite volume method* because the numerical fluxes are approximated at the nodes of the mesh, and the second approach is called *pure edge finite volume method* because they are approximated at the edges of the mesh.

In Equation (2) appear the quantities  $P$  and  $\gamma$ :  $P$  is the pressure, and it is assumed to follow the ideal gas law. The constant  $\gamma$  is taken to be the ideal gas constant, i.e.  $\gamma = 1.4$ .

Finally, in order to complete our physical model (1)–(2), we need to specify the source term  $S = S(\mathbf{x}, U)$ , which is supposed to be a smooth function of its two variables. We focus on the following cases:

- independent from  $U$ :  $S = S(\mathbf{x})$ ,
- gravity type:  $S = (0, \rho \nabla \zeta, \rho \nabla \zeta \cdot \mathbf{u})^T$ , where  $\zeta = \zeta(\mathbf{x})$  is a smooth function,
- friction type:  $S = (0, -\lambda \rho \mathbf{u}, -\alpha \lambda \rho \|\mathbf{u}\|^2)^T$ , where  $\alpha$  and  $\lambda$  are constants.

The purpose of this study is to generalize on arbitrary two-dimensional polygonal meshes the ideas developed in [2]. In that work, the authors detail a discretization of the source term that allows to exactly preserve the stationary solutions to (1), in the one-dimensional case. We will focus on the two-dimensional framework, and discuss the one-dimensional counterpart in Section 2 as an introduction to their method.

Several studies showed that node-based methods are efficient in Lagrangian update hydrodynamics [3, 4] and allow to obtain a local mesh update that is consistent with geometric conservation laws. It is not the case for the edge-based approach [5]. However, the node-based methods sometimes display parasitic behaviors that may be treated using ad-hoc processes [1, 3, 6]. On the other hand, pure edge-based finite volume schemes perform well and benefit most from the one-dimensional numerical flux design theory. For first-order explicit finite volume schemes, the propagation is performed via cell adjacency (*i.e.* edges), and this may generate an important (numerical) travel time delay on certain specified meshes (see for instance [1, Sect. 1]).

This motivates the use of *composite* schemes, which make a continuous bridge between edge-based and node-based finite volume schemes. This type of schemes was introduced in [1] to solve the homogeneous 2D Euler

equations. The numerical results show that these schemes perform well in terms of stability. For instance, in [1, 7, 8] the polygonal pure nodal GLACE scheme from [3] and the composite GLACE conical degenerate extension from [7, 8] were compared on the Sedov test (for the Euler Gas Dynamics) solved with Lagrange update formulation. The authors of these papers showed that the composite GLACE conical degenerate extension provides robust numerical results. However, the pure nodal scheme exhibits numerical instability due to lack of numerical viscosity in the diagonal direction, and furthermore both schemes are first order in time and space.

The purpose of the present work is to develop a composite flux scheme that extends the ideas of Alouges, Ghidaglia and Tajchman [2] in the 2D case.

The objective of the method presented in [2] is not different from the motivation of well-balancedness as introduced in the pioneering works by Gosse, Leroux, and Greenberg [9, 10]: the idea is to discretize Equation (1) in such a way that stationary solutions to the continuous problem – or at least a subclass of them – are also stationary solutions to the discretized problem. Well-balanced methods have been studied for almost three decades now (e.g. [11–15]), and their efficiency is not to be proved anymore. They have been used in many applications, among others: [16–22] for shallow waters equations, or [23–25] for the Burgers’ or Euler’s equations.

The difference between well-balancedness and *enhanced consistency* (a term coined in [2] and mentioned/used in [26]) lies in the way the source term is discretized. In the former approach, the source term is generally treated in a particular way, different from the discretization of the other quantities in (1). In the latter approach, the source term is written as the divergence of some (approximate) function, and the two divergence operators on the left and right of the equation can then be discretized in the same way. We are using the ideas of [2] throughout this work, and to emphasize the difference of discretization between their method and other well-balanced method, we shall use the term *enhanced consistency* for the ideas presented in this work (even if this is a subclass of well balanced method).

However, upon a closer look at [2], we believe the 2D case needed more investigation, and our work is an attempt in this direction, also using the multidimensional awareness of composite flux schemes.

The paper is organised as follows: in Section 2 we give a brief overview of the VFFC scheme in the 1D framework, and we explain in details the main idea of [2]. In Section 3 we define a *composite* scheme along the lines of [1], which is a generalization of the VFFC scheme in 2D. Section 4 is devoted to the treatment of the source term and several methods are detailed. Eventually, some numerical tests are provided in order to compare the different methods.

## 2. 1D SCHEME

In this Section, we briefly recall the 1D VFFC scheme from [27], which discretizes the 1D version of (1). This scheme is an Eulerian flux-based scheme. We also present the notion of *enhanced consistency* as introduced in [2], which consists in discretizing the source term so as to preserve exactly the stationary solutions of (1).

### 2.1. Flux based scheme

We consider a space discretization  $\{x_{j+1/2}\}_j$  of a given interval  $[0, L]$ , and define the cells  $\Omega_j = [x_{j-1/2}, x_{j+1/2}]$ . Their length is denoted by  $\Delta x_j = x_{j+1/2} - x_{j-1/2}$  and we also need to define their centroid  $\{x_j\}_j$ :  $x_j = (x_{j+1/2} + x_{j-1/2})/2$ .

A classical finite volume scheme in time writes :

$$\Delta x_j \frac{\mathcal{U}_j^{n+1} - \mathcal{U}_j^n}{\Delta t^n} + \mathcal{G}^{j+1/2,n} - \mathcal{G}^{j-1/2,n} = \mathcal{S}_j^n, \quad \forall j,$$

where the superscript  $n$  denotes current time  $t^n$ , assuming a time discretization  $\{t^k\}_{k \geq 0}$ . The calligraphic letters  $\mathcal{U}, \mathcal{F}$  and  $\mathcal{S}$  correspond to the discretized approximations of  $U, F$  and  $S$ . In order to simplify notations,

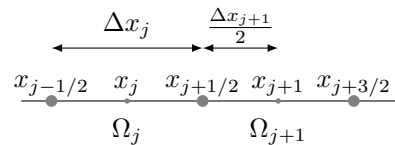


FIGURE 1. 1D discretization

we will omit the superscript  $n$  and only leave the superscript  $n + 1$ :

$$\Delta x_j \frac{\mathcal{U}_j^{n+1} - \mathcal{U}_j}{\Delta t} + \mathcal{G}^{j+1/2} - \mathcal{G}^{j-1/2} = \mathcal{S}_j, \quad \forall j, \quad (3)$$

The numerical source term is denoted by  $\mathcal{S}_j$  and can be computed using several formulas (see for instance Section 2.2).

The quantity  $\mathcal{G}_{j+1/2}$  is the numerical flux between the cells  $j$  and  $j + 1$ . To obtain a *flux-based* scheme, we define it as:

$$\mathcal{G}^{j+1/2} = \frac{\mathcal{F}_j + \mathcal{F}_{j+1}}{2} + \Lambda_{j+\frac{1}{2}} \frac{\mathcal{F}_j - \mathcal{F}_{j+1}}{2}, \quad \Lambda_{j+\frac{1}{2}} = \text{sign}(J(\mathcal{U}_{j+1/2})),$$

where the sign of a matrix is defined below, see Definition 2.1. Here,  $J(\mathcal{U}_{j+1/2})$  denotes the Jacobian matrix of the mapping  $U \mapsto F(U)$  evaluated at  $\mathcal{U}_{j+1/2}$ .

The quantity  $\mathcal{U}_{j+1/2}$  is an approximation of the solution at the interface between the cells  $\Omega_j$  and  $\Omega_{j+1}$ . One of the simplest approximations is the VFFC average (see [27]), and in this case  $\mathcal{U}_{j+1/2}$  is given by  $\mathcal{U}_{j+1/2} = (\mathcal{U}_j + \mathcal{U}_{j+1})/2$ . We emphasize the fact that  $\mathcal{U}_j$  and  $\mathcal{U}_{j+1}$  are only the unknowns of the finite volume scheme, and no additional knowledge about the solution is required to compute  $\mathcal{G}^{j+1/2}$ .

The matrix  $\Lambda_{j+\frac{1}{2}}$  is called an “upwind matrix”. We also emphasize that the superscript  $j+1/2$  in  $\mathcal{G}^{j+1/2}$  has to be understood now as a “global index” in the mesh, and that  $\mathcal{G}^{j+1/2}$  is the numerical flux evaluated at the degree of freedom<sup>1</sup>  $j + 1/2$ . More details about the notion of “global” or “local” indices are given in Section 3.

A natural way of discretizing the source term is the following:

$$\mathcal{S}_j = S(x_j, \mathcal{U}_j) \Delta x_j. \quad (4)$$

However, as explained in [2], this choice of  $\mathcal{S}_j$  does not allow to preserve exactly the stationary solutions. It is also neither exact nor even more precise on the source than on the rest. Indeed, the “approximate Well-Balanced” discretization [28] are just more accurate, which in practice, is enough. We present in Section 2.2 the general procedure described designed in [2] aimed at preserving numerically the stationary solutions.

**Definition 2.1** (Sign of a matrix). Let  $J \in \mathbb{R}^{m \times m}$  be a diagonalizable matrix of size  $m \times m$ . Let  $\{\lambda_i\}_{i=1}^m$  denote its eigenvalues,  $R$  the matrix composed of the eigenvectors associated to the eigenvalues, and  $L$  the matrix associated to the left eigenvectors of  $J$  (*i.e.* the eigenvectors of  $J^T$ ). We know that:

$$J = R \text{diag}(\{\lambda_i\}) L, \quad RL = LR = I_m,$$

and we define:

$$\text{sign}(J) = R \text{diag}(\{\text{sign}(\lambda_i)\}) L. \quad (5)$$

In Definition 2.1, unless stated explicitly, the term “eigenvector” refers to the right eigenvectors, that is the vectors  $v$  such that  $Jv = \lambda v$  for some eigenvalue  $\lambda$ .

## 2.2. Well Balanced (Enhanced consistency)

One of the main disadvantages of the discretization (4) is that, even if the scheme were to be initialized with an exact stationary solution, the approximate solution eventually drifts away from the stationary solution. However, it is desirable to be able to accurately capture exact stationary solutions at the discrete level. This defines our first criterion:

<sup>1</sup>“Degree of freedom” will later be abbreviated *dof*.

**Criterion 2.2.** *If, at some iteration  $n$ , we have:*

$$\forall j, U^n(x_j) = U^*(x_j) \quad \text{with} \quad \frac{d}{dx} F(U^*) = S(x, U^*(x)),$$

then

$$\forall j, U^{n+1}(x_j) = U^n(x_j).$$

This criterion means that we want the scheme to be able to “capture” (or at least get an accurate approximation of) continuous stationary solutions. We define (enhanced consistency)  $\Phi = F(U^*)$ . As explained in [2], a way of satisfying Criterion 2.2 is to define the source term as:

$$\begin{aligned} \mathcal{S}_j &= \frac{1}{2} [(I + \Lambda_{j+1/2})\Phi(x_j) + (I - \Lambda_{j+1/2})\Phi(x_{j+1}) - (I + \Lambda_{j-1/2})\Phi(x_{j-1}) - (I - \Lambda_{j-1/2})\Phi(x_j)] \\ &= \frac{1}{2} [(I - \Lambda_{j+1/2})(\Phi(x_{j+1}) - \Phi(x_j)) + (I + \Lambda_{j-1/2})(\Phi(x_j) - \Phi(x_{j-1}))]. \end{aligned} \quad (6)$$

Owing to the well-known formula:

$$\Phi(x_{j+1}) - \Phi(x_j) = \int_{x_j}^{x_{j+1}} S(y, U^*(t, y)) dy, \quad (7)$$

we are able to rewrite (6) as

$$\mathcal{S}_j = \frac{1}{2} \left( \int_{x_{j-1}}^{x_{j+1}} S(y, U^*(t, y)) dy - \Lambda_{j+1/2} \int_{x_j}^{x_{j+1}} S(y, U^*(t, y)) dy + \Lambda_{j-1/2} \int_{x_{j-1}}^{x_j} S(y, U^*(t, y)) dy \right).$$

Since  $U^*$  is not known *a priori*, we define the source term as

$$\mathcal{S}_j = \frac{1}{2} \left( \int_{x_{j-1}}^{x_{j+1}} S(y, U(t, y)) dy - \Lambda_{j+1/2} \int_{x_j}^{x_{j+1}} S(y, U(t, y)) dy + \Lambda_{j-1/2} \int_{x_{j-1}}^{x_j} S(y, U(t, y)) dy \right), \quad (8)$$

and Criterion 2.2 is still satisfied. As noticed in [2], when  $S$  is piecewise constant ( $S = S_j = S(x_j, U(x_j))$  on cell  $j$ ), Equation (8) can be reformulated in a simpler way:

$$\mathcal{S}_j = \frac{1}{2} (I + \Lambda_{j-1/2}) \left( \frac{\Delta x_{j-1}}{2} S_{j-1} + \frac{\Delta x_j}{2} S_j \right) + \frac{1}{2} (I - \Lambda_{j+1/2}) \left( \frac{\Delta x_j}{2} S_j + \frac{\Delta x_{j+1}}{2} S_{j+1} \right). \quad (9)$$

This simple formula for the discretized source term  $\mathcal{S}$  seems, at first sight, more reasonable than a centered discretization of the source term. The work [2] provides numerical experiments to show that the discretization (9) is indeed better than the centered discretization. The two-dimensional setting cannot be treated in the same way, since it not possible to write (7), nor to split (8) as easily as in 1d.

We now turn to a description of the two-dimensional framework, then propose some discretizations of the source term, and finally asses their efficiency on numerical examples.

### 3. 2D SCHEMES

Let  $\Omega = [0, 1]^2$ , and define  $\mathcal{T}$  a polygonal mesh over  $\Omega$ . We denote by  $\Omega_j$  a polygonal cell of the mesh  $\mathcal{T}$ , and let  $\mathbf{x}_{r-1}$ ,  $\mathbf{x}_r$  and  $\mathbf{x}_{r+1}$  three consecutive vertices of the polygon  $\Omega_j$ . Define

- the middle of the edge  $[\mathbf{x}_r, \mathbf{x}_{r+1}]$ :  $\mathbf{x}_{r+\frac{1}{2}} = \frac{\mathbf{x}_r + \mathbf{x}_{r+1}}{2}$ ,

- the outward normal to the edge directed from  $\mathbf{x}_r$  to  $\mathbf{x}_{r+1}$ ,  $[\mathbf{x}_r, \mathbf{x}_{r+1}]$ :  $\mathbf{C}_j^{r+\frac{1}{2}} = -(\mathbf{x}_{r+1} - \mathbf{x}_r)^\perp$ ,
- the normal to the node  $r$ :  $\mathbf{C}_j^r = \frac{1}{2} (\mathbf{C}_j^{r+\frac{1}{2}} + \mathbf{C}_j^{r-1/2})$ ,

where, for any vector  $\mathbf{v} = (v_1, v_2) \in \mathbb{R}^2$ , we use the usual notation  $\mathbf{v}^\perp = (-v_2, v_1)$  which corresponds to a rotation of angle  $+\frac{\pi}{2}$ . In what follows we present the *composite schemes*, as a combination of “node-based schemes” – which involve  $\mathbf{C}_j^r$  – and “edge-based schemes” – which involve  $\mathbf{C}_j^{r+\frac{1}{2}}$ . This is not new, and very much inspired from [1].

**Remark 3.1.** We acknowledge that it normally does not make sense to talk about a “normal vector to a node”. However, the definition we use (which has been borrowed from [1]) is always well-defined for polygonal meshes. For all of our purposes, it is well-suited and we shall use this definition throughout this work.

For the sake of clarity, we start by explaining below the different sums that can be performed over the mesh, for an arbitrary function  $g = g(x)$ . By convention, for any given cell we will use integers  $r$  to denote vertices of the cell, also called nodes. They are supposed to be labelled consecutively, so that it makes sense to talk about the edge  $[\mathbf{x}_r, \mathbf{x}_{r+1}]$ . Moreover, we implicitly suppose  $r \in \{1, \dots, V\}$  for a polygonal cell with  $V$  vertices (with the loop convention that  $\mathbf{x}_{V+1} = \mathbf{x}_1$ ). We also denote by half-integers  $r + \frac{1}{2}$  the middle of the edge  $[\mathbf{x}_r, \mathbf{x}_{r+1}]$ . Our conventions are illustrated in Figure 2.

For a given cell  $\Omega_j$  we denote by  $dof \in \{r : r \in \Omega_j\} \cup \{r + \frac{1}{2} : r + \frac{1}{2} \in \Omega_j\}$  any degree of freedom of

$\Omega_j$  (it can be a node or the middle of an edge), and the notation  $g_j^{dof}$  denotes the evaluation of the function  $g$  at the node or edge  $dof$  in cell  $\Omega_j$ . Sometimes  $dof$  will also denote a general degree of freedom in the mesh, *i.e.* it can be any node or middle of an edge in the mesh. The meaning of  $dof$  will be clear from the context (think of  $r$  and  $r + \frac{1}{2}$  as “local indices” within a given cell, and of  $dof$  as either a “local” or “global” index, depending on the context).

- $\sum_{r \in \Omega_j} g_j^r$ : sum over all the vertices of the cell  $j$ ,
- $\sum_{r+\frac{1}{2} \in \Omega_j} g_j^{r+\frac{1}{2}}$ : sum over all the mid-edge points of the cell  $j$ ,
- $N_{dof} = \sum_{i: dof \in \Omega_i} 1$ : number of cells that contains the given degree of freedom  $dof$ ,
- $\sum_{i: dof \in \Omega_i} g_i^{dof}$ : sum, for a given degree of freedom in the mesh, over all the cells that contains this degree of freedom,
- $\sum_{j \in \mathcal{T}} g_j$ : sum over all cells in the mesh,
- $\sum_{r \in nodes(\mathcal{T})} g^r$ : sum over all nodes in the mesh,
- $\sum_{r+\frac{1}{2} \in edges(\mathcal{T})} g^{r+\frac{1}{2}}$ : sum over all mid-edge points in the mesh,
- $h$ : the maximum edge length among all edges in the mesh.

The normal vectors  $\mathbf{C}_j^r$  satisfy the following equalities:

$$\sum_{r \in \Omega_j} \mathbf{C}_j^r = \sum_{r+\frac{1}{2} \in \Omega_j} \mathbf{C}_j^{r+\frac{1}{2}} = \mathbf{0}, \quad \forall j \in \mathcal{T}, \quad (10)$$

$$\sum_{i: dof \in \Omega_i} \mathbf{C}_i^{dof} = \mathbf{0}, \quad \forall dof \in \{r : r \in \Omega_j\} \cup \{r + 1/2 : r + 1/2 \in \Omega_j\}. \quad (11)$$

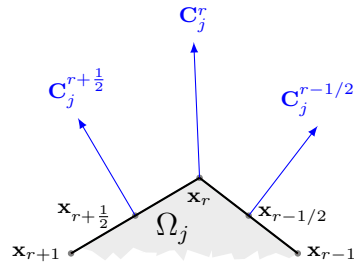
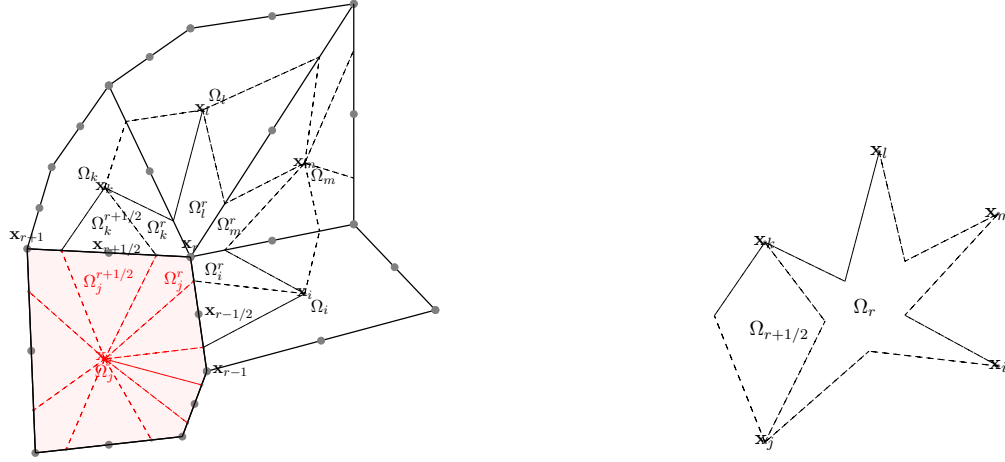


FIGURE 2. Normal vectors at nodes and edges: composite set (not to scale).



(A) Generic cell  $\Omega_j$  with its sub-cells  $\Omega_j^{dof}$  defined around nodes  $r$  and edges  $r + \frac{1}{2}$ . (B) Associated composite dual cell around node  $\mathbf{x}_r$ :  $\Omega_r = \cup_{i:r \in i} \Omega_i^r$ , and around edge  $\mathbf{x}_{r+\frac{1}{2}}$ :  $\Omega_{r+\frac{1}{2}} = \Omega_j^{r+\frac{1}{2}} \cup \Omega_k^{r+\frac{1}{2}}$ .

FIGURE 3. Illustration of subcells (left) and composite dual cells around nodes and edge (right).

Moreover, we have the following result, whose proof can be found in [29]:

**Proposition 3.2.** *Let  $g \in \mathcal{C}^2(\mathbb{R}^2; \mathbb{R})$ . Then, for all  $\theta \in [0, 1]$ :*

$$\frac{1}{|\Omega_j|} \int_{\partial\Omega_j} g \mathbf{n} = \frac{1}{|\Omega_j|} \left[ (1 - \theta) \sum_{r \in \Omega_j} g(\mathbf{x}_r) \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2} \in \Omega_j} g(\mathbf{x}_{r+\frac{1}{2}}) \mathbf{C}_j^{r+\frac{1}{2}} \right] + \mathcal{O}(h), \quad (12)$$

and the remainder term in (12) vanishes if  $g$  is an affine function.

We define the sub-volumes (see Figure 3 for an illustration)

$$|\Omega_j^r| = \frac{1 - \theta}{2} (\mathbf{x}_r - \mathbf{x}_j) \cdot \mathbf{C}_j^r, \quad |\Omega_j^{r+\frac{1}{2}}| = \frac{\theta}{2} (\mathbf{x}_{r+\frac{1}{2}} - \mathbf{x}_j) \cdot \mathbf{C}_j^{r+\frac{1}{2}}, \quad (13)$$

so that the volume of the cell  $\Omega_j$  reads

$$|\Omega_j| = \sum_{r \in \Omega_j} |\Omega_j^r| + \sum_{r+\frac{1}{2} \in \Omega_j} |\Omega_j^{r+\frac{1}{2}}| = \sum_{dof \in \Omega_j} |\Omega_j^{dof}|. \quad (14)$$

We end this section by giving a definition that will be useful in the rest of the paper, and which generalizes Definition 2.1 to the two-dimensional case.

**Definition 3.3.** We define the Jacobian matrix of the physical flux function  $\mathcal{F}$ , given by (2), along the unit direction vector  $\mathbf{n} = (n_1, n_2)$  as

$$J(\mathcal{U}, \mathbf{n}) = n_1 \nabla F_1(U) + n_2 \nabla F_2(U).$$

This matrix is diagonalizable because the system (1) is hyperbolic. Moreover, we have:

$$J(U, \mathbf{n}) = -J(U, -\mathbf{n}), \quad (15)$$

hence, by (5),

$$\text{sign}(J(U, \mathbf{n})) = -\text{sign}(J(U, -\mathbf{n})). \quad (16)$$



### 3.1. Edge-based schemes

We now present a 2D edge-based extension of the 1D VFFC scheme described in Section 2. First, integrate Equation (1) on a given cell  $\Omega_j$ , then use Proposition 3.2 with  $\theta = 1$ . We obtain that the flux is approximated by

$$\int_{\Omega_j} \operatorname{div} F(U) = \int_{\partial\Omega_j} F(U) \cdot \mathbf{n} \approx \sum_{r+\frac{1}{2} \in \Omega_j} \mathcal{G}^{r+\frac{1}{2}} \cdot \mathbf{C}_j^{r+\frac{1}{2}}.$$

Consider an edge labelled  $r + \frac{1}{2}$  that does not lie at the boundary of the computational domain (we will deal later with the boundary conditions), and let  $j, k$  the labels of its two support cells. That is,  $r + \frac{1}{2} \in \Omega_j \cap \Omega_k$ . We define:

$$\mathcal{G}^{r+\frac{1}{2}} = \frac{\mathcal{F}_j + \mathcal{F}_k}{2} + \Lambda_j^{r+\frac{1}{2}} \frac{\mathcal{F}_j - \mathcal{F}_k}{2}, \quad \Lambda_j^{r+\frac{1}{2}} = \operatorname{sign} J \left( \mathcal{U}_{r+\frac{1}{2}}, \mathbf{n}_j^{r+\frac{1}{2}} \right), \quad (17)$$

where the sign of a matrix is defined in Definition 3.3, and  $\mathbf{n}_j^{r+\frac{1}{2}} = \mathbf{C}_j^{r+\frac{1}{2}} / \|\mathbf{C}_j^{r+\frac{1}{2}}\|$ . Moreover, by Definition 3.3,

$$\Lambda_j^{r+\frac{1}{2}} = -\Lambda_k^{r+\frac{1}{2}}. \quad (18)$$

Using (18), we note that equation (17) writes:

$$\mathcal{G}^{r+\frac{1}{2}} = \frac{1}{N_{r+\frac{1}{2}}} \sum_{i:r+\frac{1}{2} \in \Omega_i} \left( I + \Lambda_i^{r+\frac{1}{2}} \right) \mathcal{F}_i, \quad (19)$$

which will be useful in Section 3.2. Moreover, equation (18) also yields:

$$\sum_{i:r+\frac{1}{2} \in \Omega_i} \Lambda_i^{r+\frac{1}{2}} = 0. \quad (20)$$

Finally, the average state  $\mathcal{U}_{r+\frac{1}{2}}$  is given by the VFFC average (see [27]):

$$\mathcal{U}_{r+\frac{1}{2}} = \frac{|\Omega_j| \mathcal{U}_j + |\Omega_k| \mathcal{U}_k}{|\Omega_j| + |\Omega_k|}. \quad (21)$$

**Remark 3.4.** To emphasize the fact that  $\mathcal{U}_{r+\frac{1}{2}}$  depends on the cells  $j$  and  $k$ , we may also write it  $\mathcal{U}_{(j,k)}$ .

### 3.2. Generalization to the nodes and composite scheme

We present here the *composite* scheme as defined in [1]. Using Proposition 3.2, we have

$$\int_{\partial\Omega_j} F(U) \cdot \mathbf{n} \approx (1 - \theta) \sum_{r \in \Omega_j} \mathcal{G}^r \cdot \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2} \in \Omega_j} \mathcal{G}^{r+\frac{1}{2}} \cdot \mathbf{C}_j^{r+\frac{1}{2}},$$

and the *composite* scheme reads:

$$|\Omega_j| \frac{\mathcal{U}_j^{n+1} - \mathcal{U}_j}{\Delta t} + (1 - \theta) \sum_{r \in \Omega_j} \mathcal{G}^r \cdot \mathbf{C}_j^r + \theta \sum_{r+1/2 \in \Omega_j} \mathcal{G}^{r+\frac{1}{2}} \cdot \mathbf{C}_j^{r+1/2} = \mathcal{S}_j. \quad (22)$$

The edge flux  $\mathcal{G}^{r+\frac{1}{2}}$  is defined in Equation (19), and we define the node flux  $\mathcal{G}^r$  in a similar manner:

$$\mathcal{G}^r = \frac{1}{N_r} \sum_{i:r \in \Omega_i} \left( I + \Lambda_i^r \right) \mathcal{F}_i. \quad (23)$$

A natural choice for defining  $\Lambda_i^r$  would be

$$\Lambda_i^r = \tilde{\Lambda}_i^r, \quad \tilde{\Lambda}_i^r := \frac{1}{N_r - 1} \sum_{k:r \in \Omega_k, k \neq i} \text{sign } J(\mathcal{U}_{(i,k)}, \mathbf{n}_i^r).$$

where  $\mathcal{U}_{(i,k)}$  is the notation for the VFFC average between cells  $i$  and  $k$  given in Remark 3.4, and  $\mathbf{n}_i^r = \mathbf{C}_i^r / \|\mathbf{C}_i^r\|$ . However, as opposed to equation (20), we do not have *a priori*:

$$\sum_{i:r \in \Omega_i} \Lambda_i^r = 0. \quad (24)$$

This property is essential to ensure the consistency of the scheme: indeed, this is a key argument in the proof of Proposition 3.6.

A way of satisfying equation (24) is to define  $\Lambda_i^r$  as:

$$\Lambda_i^r = \tilde{\Lambda}_i^r - \frac{1}{N_r} \sum_{k:r \in \Omega_k} \tilde{\Lambda}_k^r. \quad (25)$$

One can recover the edge-based scheme (resp. the node-based scheme) by setting  $\theta = 1$  (resp.  $\theta = 0$ ) in the scheme (22).

### 3.3. Boundary conditions

In Section 5, we will work with two types of boundary conditions. The first type is the Dirichlet condition, applied as follows: let *dof* be a degree of freedom where the solution is imposed to be  $\mathcal{U}_{\text{boundary}}$ . The numerical flux is given by:  $\mathcal{G}^{dof} = \mathcal{F}(\mathcal{U}_{\text{boundary}})$ . We make the assumption that the exact solution is known on the boundary of the computational domain. Using ghost cells, we fill them by the evaluation of exact solution at their centroid.

The second type of boundary conditions is *outflow* and we apply it to a non-stationary two dimensional Riemann problem (see Figure 13), in this case we fill the ghost cells by extrapolation of the values of their adjacent cell inside the domain.

### 3.4. Properties of the composite scheme

**Proposition 3.5** (Global conservativity). *The scheme (22) is globally conservative, i.e.*

$$\sum_{j \in \mathcal{T}} |\Omega_j| \mathcal{U}_j^{n+1} = \sum_{j \in \mathcal{T}} |\Omega_j| \mathcal{U}_j + \Delta t \sum_{j \in \mathcal{T}} \mathcal{S}_j,$$

up to some boundary terms.

*Proof.* We sum (22) over  $j \in \mathcal{T}$ :

$$\sum_{j \in \mathcal{T}} \left( |\Omega_j| \frac{\mathcal{U}_j^{n+1} - \mathcal{U}_j^n}{\Delta t} - \mathcal{S}_j \right) = -(1 - \theta) \sum_{j \in \mathcal{T}} \sum_{r \in \Omega_j} \mathcal{G}^r \cdot \mathbf{C}_j^r - \theta \sum_{j \in \mathcal{T}} \sum_{r + \frac{1}{2} \in \Omega_j} \mathcal{G}^{r + \frac{1}{2}} \cdot \mathbf{C}_j^{r + \frac{1}{2}}.$$

We can then exchange the sum over  $\{j \in \mathcal{T}\}$  and the ones over  $\{r \in \Omega_j\}$  and  $\{r + \frac{1}{2} \in \Omega_j\}$ , and use the relation (11) to obtain:

$$\sum_{j \in \mathcal{T}} \sum_{r \in \Omega_j} \mathcal{G}^r \cdot \mathbf{C}_j^r = \sum_{r \in \text{nodes}(\mathcal{T})} \sum_{i:r \in \Omega_i} \mathcal{G}^r \cdot \mathbf{C}_i^r = \sum_{r \in \text{nodes}(\mathcal{T})} \mathcal{G}^r \cdot \underbrace{\left( \sum_{i:r \in \Omega_i} \mathbf{C}_i^r \right)}_{=0} = 0,$$

and:

$$\sum_{j \in \mathcal{T}} \sum_{r+\frac{1}{2} \in \Omega_j} \mathcal{G}^{r+\frac{1}{2}} \cdot \mathbf{C}_j^{r+\frac{1}{2}} = \sum_{r \in \text{edges}(\mathcal{T})} \sum_{i:r+\frac{1}{2} \in \Omega_i} \mathcal{G}^{r+\frac{1}{2}} \cdot \mathbf{C}_i^{r+\frac{1}{2}} = \sum_{r+\frac{1}{2} \in \text{edges}(\mathcal{T})} \mathcal{G}^{r+\frac{1}{2}} \cdot \underbrace{\left( \sum_{i:r+\frac{1}{2} \in \Omega_i} \mathbf{C}_i^{r+\frac{1}{2}} \right)}_{=0} = 0.$$

□

**Proposition 3.6** (Consistency of the fluxes). *The fluxes (17) and (23) are consistent.*

*Proof.* Let  $r + \frac{1}{2}$  be the index of an inner edge and  $j, k$  the indices of its support cells. If  $\mathcal{U}_j = \mathcal{U}_k = \bar{\mathcal{U}}$ , then using (17), we have  $\mathcal{G} = \mathcal{F}(\bar{\mathcal{U}})$ . Similarly, let  $r$  be the index of an inner node, meaning it does not lie on the boundary of the computational domain. Assuming that  $\mathcal{U}_i = \bar{\mathcal{U}}$  for any cell  $i$  such that  $r \in \Omega_i$ , we use equations (23) and (24), and obtain:

$$\mathcal{G}^r = \mathcal{F}(\bar{\mathcal{U}}) + \frac{1}{N_r} \underbrace{\left( \sum_{i:r \in \Omega_i} \Lambda_i^r \right)}_{=0} \mathcal{F}(\bar{\mathcal{U}}) = \mathcal{F}(\bar{\mathcal{U}}).$$

□

#### 4. DISCRETIZATION OF THE SOURCE TERM

In this Section, we study different ways of computing the numerical source term  $\mathcal{S}_j$  which satisfy a 2D version of Criterion 2.2 (ideally) and are different from the classical discretization:  $\mathcal{S}_j = |\Omega_j| S(\mathbf{x}_j, \mathcal{U}_j)$ .

##### 4.1. Well Balanced (Enhanced consistency) in 2D

The reasoning is identical to Section 2.2. We want to compute  $\mathcal{S}_j$  so as to satisfy the following property, which is a 2D generalization of Criterion 2.2:

**Criterion 4.1.** *If, at some iteration  $n$ , we have  $U^n(\mathbf{x}_j) = U^*(\mathbf{x}_j)$  for all  $j$ , and  $U^*$  is such that  $\text{div } F(U^*) = S(\mathbf{x}, U^*(\mathbf{x}))$ , then*

$$\forall j, U^{n+1}(\mathbf{x}_j) = U^n(\mathbf{x}_j).$$

We define (enhanced consistency)  $\Phi^* = F(U^*)$ , and consider the edge (resp. node) flux as defined in (19) (resp. (23)). In order to satisfy Criterion 4.1, the source term would need to be discretized as:

$$\mathcal{S}_j = (1 - \theta) \sum_{r \in \Omega_j} \frac{1}{N_r} \left[ \sum_{i:r \in \Omega_i} (I_4 + \Lambda_i^r) \Phi_i^* \right] \cdot \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2} \in \Omega_j} \frac{1}{N_{r+\frac{1}{2}}} \left[ \sum_{i:r+\frac{1}{2} \in \Omega_i} (I_4 + \Lambda_i^{r+\frac{1}{2}}) \Phi_i^* \right] \cdot \mathbf{C}_j^{r+\frac{1}{2}}.$$

However, we do not know *a priori*  $\Phi^*$  nor  $U^*$ . Therefore we define  $\mathcal{S}_j$  as:

$$\mathcal{S}_j = (1 - \theta) \sum_{r \in \Omega_j} \frac{1}{N_r} \left[ \sum_{i:r \in \Omega_i} (I_4 + \Lambda_i^r) \Phi_i \right] \cdot \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2} \in \Omega_j} \frac{1}{N_{r+\frac{1}{2}}} \left[ \sum_{i:r+\frac{1}{2} \in \Omega_i} (I_4 + \Lambda_i^{r+\frac{1}{2}}) \Phi_i \right] \cdot \mathbf{C}_j^{r+\frac{1}{2}}, \quad (26)$$

where  $\Phi$  is a solution to  $\text{div } \Phi = S(\mathbf{x}, U^n)$ . The upwind matrices are still given by equations (17) and (25). The scheme (22) can then be written as:

$$|\Omega_j| \frac{\mathcal{U}_j^{n+1} - \mathcal{U}_j^n}{\Delta t} + (1 - \theta) \sum_{r \in \Omega_j} \tilde{\mathcal{G}}^r \cdot \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2} \in \Omega_j} \tilde{\mathcal{G}}^{r+\frac{1}{2}} \cdot \mathbf{C}_j^{r+\frac{1}{2}} = 0, \quad (27)$$

with:

$$\tilde{\mathcal{G}}^{dof} = \frac{1}{N_{dof}} \sum_{i:dof \in \Omega_i} \left( I_4 + \Lambda_i^{dof} \right) (\mathcal{F}_i - \Phi_i). \quad (28)$$

However, in contrast with the 1D case, it is not possible in the general case to find an expression of  $\mathcal{S}_j$  as a function of  $S$  and therefore to end up with a formula that is similar to (8). In Section 4.2 we present a way of computing  $\Phi$  by solving a Poisson equation, and in Section 4.3, we try to generalize formula (9) to the 2D framework, without guarantying that Property 4.1 is still satisfied.

## 4.2. Anti-derivative using a Poisson equation

First of all, let us clarify what we are looking for in this section. By “anti-derivative”, it is meant taking a function in  $W \in R^p$ , we look at a function  $\Phi \in R^{d \times p}$  such that  $W = \text{div } \Phi$ . In a one-dimensional setting, the divergence operator is simply its derivative, hence the anti-derivative is the same as the primitive function (in this case unique up to an additional constant).

This idea comes from [2] (see also in an other context [30]). In order to solve  $\text{div } \Phi = S$ , we solve the Poisson problem  $\Delta \psi = S$ , and let  $\Phi = \nabla \psi$ . We impose homogeneous Dirichlet boundary conditions for  $\psi$ . The construction of the scheme is detailed in Appendix A. With this method, the unknowns  $(\Phi_i)_i$  that appear in (28) are given by:

$$|\Omega_j| \Phi_j = \sum_{r \in \Omega_j} |\Omega_j^r| \Phi_r + \sum_{r+\frac{1}{2} \in \Omega_j} |\Omega_j^{r+\frac{1}{2}}| \Phi_{r+\frac{1}{2}}, \quad \Phi_{r+\frac{1}{2}} = \frac{\Phi_r + \Phi_{r+1}}{2}, \quad (29)$$

where the sub-volumes are defined in (13) and the anti-derivative at the nodes satisfies

$$\sum_{r \in \Omega_j} \Phi_r \cdot \mathbf{C}_j^r = |\Omega_j| S_j, \quad (30)$$

with

$$\left( \sum_{i|r \in \Omega_i} \mathbf{C}_i^r \otimes (\mathbf{x}_i - \mathbf{x}_r) \right) \Phi_r^T = \sum_{i|r \in \Omega_i} \psi_i \otimes \mathbf{C}_i^r. \quad (31)$$

Equations (30)-(31) define a linear system whose size is the number of cells and whose unknown is  $\psi$ . Once this system is solved, we compute the anti-derivative at the nodes with equation (31) and the values at the centers of the cells with equation (29).

### Dirichlet boundary conditions

Let  $dof$  be a degree of freedom where the solution is imposed to be  $\mathcal{U}_{\text{boundary}}$ . The numerical flux (28) is given by:  $\tilde{\mathcal{G}}^{dof} = \mathcal{F}(\mathcal{U}_{\text{boundary}}) - \Phi_{dof}$ , where  $\Phi_{dof}$  is defined in (30) (31).

## 4.3. Sub-volumes upwinding

Pioneering works have been done on the construction of upwind method for source see e.g. [31, 32]. They give a way to define natural upwind discretization of source when the flux is approximated by some edge ARS: flux difference or flux splitting technique. We propose here to discretize the source term by upwinding the sub-volumes around  $r$  or  $r + 1/2$ :

$$\mathcal{S}_j = \sum_{dof \in \Omega_j} \frac{1}{N_{dof}} \sum_{i:dof \in \Omega_i} \left( I_4 + \Lambda_i^{dof} \right) |\Omega_i^{dof}| S_i \quad \text{or} \quad (32)$$

$$= \sum_{r \in \Omega_j} \frac{1}{N_r} \sum_{i:r \in \Omega_i} \left( I_4 + \Lambda_i^r \right) |\Omega_i^r| S_i + \sum_{r+1/2 \in \Omega_j} \frac{1}{2} \sum_{i:r+1/2 \in \Omega_i} \left( I_4 + \Lambda_i^{r+1/2} \right) |\Omega_i^{r+1/2}| S_i, \quad (33)$$

where  $|\Omega_i^r|$  (resp.  $|\Omega_i^{r+\frac{1}{2}}|$ ) is the sub-volume in cell  $i$  linked to node  $r$  (resp. mid-edge  $r + \frac{1}{2}$ ). See Figure 3 in the case  $0 < \theta < 1$ .

For example: for  $\theta = 1$ , by letting  $k$  the adjacent cell of  $j$  with respect to local edge  $(r + 1/2)$ , the formula (32) gives:

$$\sum_{r+1/2 \in \Omega_j} \frac{1}{2} \left( (I_4 + \Lambda_k^{r+1/2}) |\Omega_k^{r+1/2}| S_k + (I_4 + \Lambda_j^{r+1/2}) |\Omega_j^{r+1/2}| S_j \right). \quad (34)$$

Knowing that  $\Lambda_j^{r+1/2} = -\Lambda_k^{r+1/2}$ , this formula gives a fully upwinding discretization at  $r + 1/2$  in cell  $j$ . Indeed, if one consider that all eigenvalues are all strictly positives (resp. all strictly negatives), we recover a local source flux at  $r + 1/2$  equal to  $|\Omega_j^{r+1/2}| S_j$  (resp.  $|\Omega_k^{r+1/2}| S_k$ ).

For the edge-based flux discretization at  $r + 1/2$  ( $\theta = 1$ ), a more natural extension (with respect to the geometric interpretation of the 1D case) with integral form (9) writes:

$$\sum_{r+1/2 \in \Omega_j} \frac{1}{2} (I_4 - \Lambda_j^{r+1/2}) \left( |\Omega_k^{r+1/2}| S_k + |\Omega_j^{r+1/2}| S_j \right), \quad (35)$$

Unfortunately, this formula (35) has not yet been tested and like the 1D case, it does not satisfy the full upwinding property.

In the next section, we will compare discretizations 4.2 and 4.3 with the centered scheme for which  $\mathcal{S}_j = S_j |\Omega_j|$ .

## 5. NUMERICAL RESULTS

We present here some numerical tests to compare the two discretizations of the source term presented in the Section 4 against the usual centered discretization. The computational domain is  $\Omega = [0, 1]^2$ . For some stationary test cases, analytical solutions are known, and thus we also use these solutions as initial data. We also present a nonstationary example.

The timestep is given by  $\Delta t = C_{CFL} \Delta x$  with the coefficient  $C_{CFL}$  depending on the test case. We perform convergence analysis on Cartesian and random meshes. A random mesh is defined as a uniform Cartesian mesh where the inner nodes are randomly perturbed (see Figure 4). The final time is  $T = 0.1$ .

For each numerical case, we display the results obtained with  $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$ . The value  $\theta = \frac{1}{2}$  (resp.  $\theta = 1$ ) corresponds to a scheme exactly halfway between node and edge scheme (resp. to a purely edge scheme). The value  $\theta = \frac{\pi}{4} \approx 0.78$  correspond to the value obtained when considering a conical mesh with curved edges and letting the curvature tend to zero (see [1]).

For all test cases, we display  $h$  on the  $x$ -axis, with  $h = \frac{1}{N}$  and  $N \in \mathbb{N}^*$  the number of discretization points of the interval  $[0, 1]$ . The same number of discretization points  $N$  is used in both dimensions.

We do not display the results for  $\theta = 0$ , since it has been observed (see [1] for instance) that non-physical phenomena can occur, even in the simplest frameworks. As an illustration of these non-physical phenomena, we show in Figure 5 a solution obtained for the homogenous Euler equations, with Sedov-type test case (initial fluid at rest, and density and pressure localised in only one cell). We observe that  $\theta = 0$  yields a propagation only along the corners, as opposed to the results obtained with  $\theta = 0.5$  and  $\theta = 1$  where the solution propagates as expected.

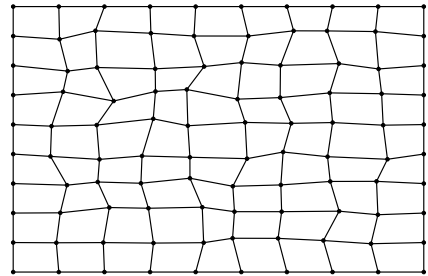


FIGURE 4. Example of a random mesh.

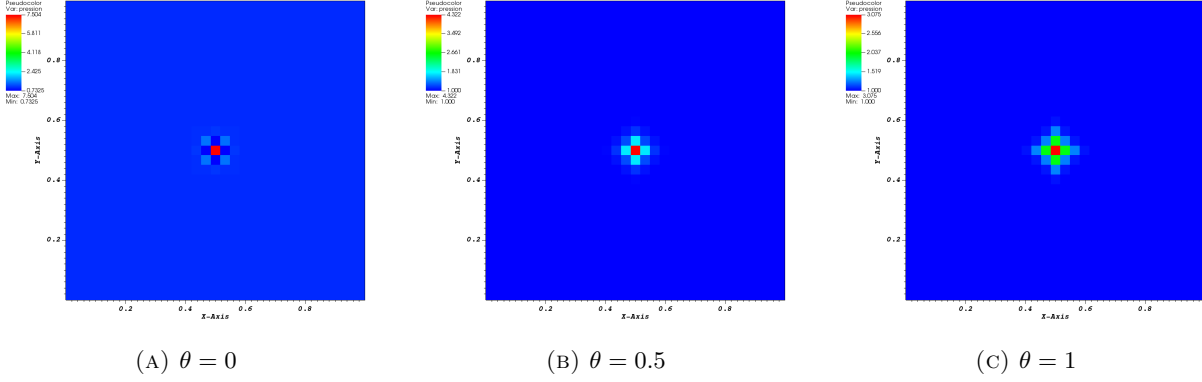


FIGURE 5. Illustration of the non-physical propagation results obtained on Cartesian mesh, with  $\theta = 0$  (left), compared to the physical propagation with  $\theta > 0$  (center:  $\theta = 0.5$ , right:  $\theta = 1$ ).

Generally, the case  $\theta = 0$  is fixed by some *ad-hoc* processes, for example in the context of Euler equations (in Lagrangian update context), the pure nodal GLACE scheme exhibit non physical behavior corrected by adding subzonal entropy [33].

One goal of the current work is to show that the benefits of the purely node schemes can be recovered when using a composite scheme while attenuating the issues observed. Namely, taking  $\theta > 0$  gives a way of curing pathologies of pure nodal schemes by a generic method.

### 5.1. 1D gravity

We set  $\mathbf{u} = \mathbf{0}$ . We add a source term of the form  $S = (0, \rho \nabla \zeta, \rho \nabla \zeta \cdot \mathbf{u})^T$  with  $\zeta(\mathbf{x}) = \mathbf{g} \cdot \mathbf{x}$  and  $\mathbf{g} = (-g, 0)$ . In the 1D framework, Equation (1) can be written as:

$$\frac{dP}{dx} = -\rho g, \quad P = (\gamma - 1)\rho E. \quad (36)$$

We choose  $P$  of the form:  $P(x) = 2g(x_0 - x) > 0$ ,  $x_0 = 1.5$ . The density is  $\rho = 2$ . We set  $C_{CFL} = 0.25$ . The results obtained on a uniform mesh with different discretizations of the source term are given in Figure 6. The results obtained on a random mesh with different discretizations of the source term are given in Figure 7. We can see that all the methods are first order convergent for  $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$ .

### 5.2. 2D gravity

We set  $\mathbf{u} = \mathbf{0}$ . We add a source term of the form  $S = (0, \rho \nabla \zeta, \rho \nabla \zeta \cdot \mathbf{u})^T$  with  $\zeta(\mathbf{x}) = -g\|\mathbf{x} - \mathbf{x}_0\|^2$ . Equation (1) can be written as:

$$\nabla P = \rho \nabla \zeta, \quad P = (\gamma - 1)\rho E.$$

A solution writes:  $P = \rho = e^\zeta$  and  $E = 1/(\gamma - 1)$ . We set  $C_{CFL} = 0.25$ . The results obtained on a uniform mesh with different discretizations of the source term are given in Figure 9. The results obtained on a random mesh with different discretizations of the source term are given in Figure 10. The schemes are all first order convergent for  $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$ .

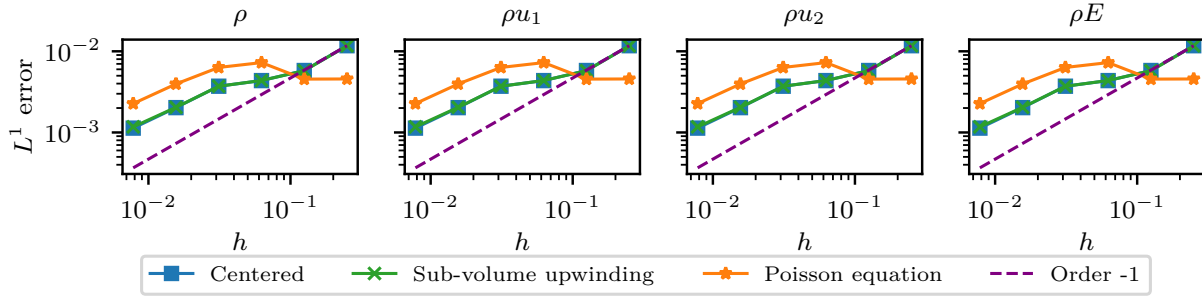
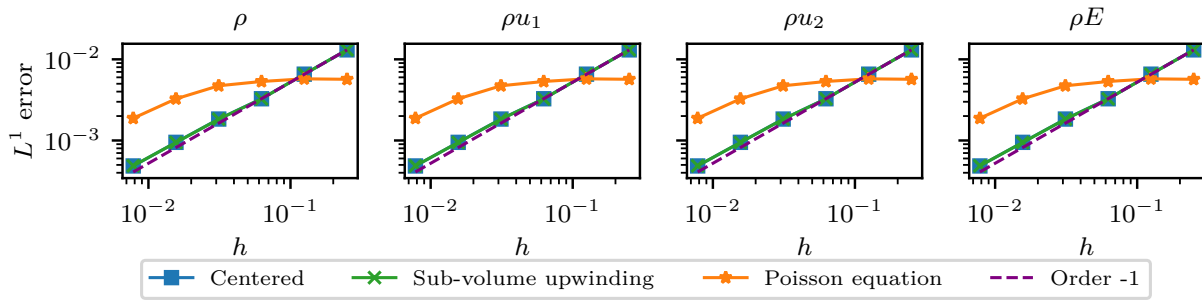
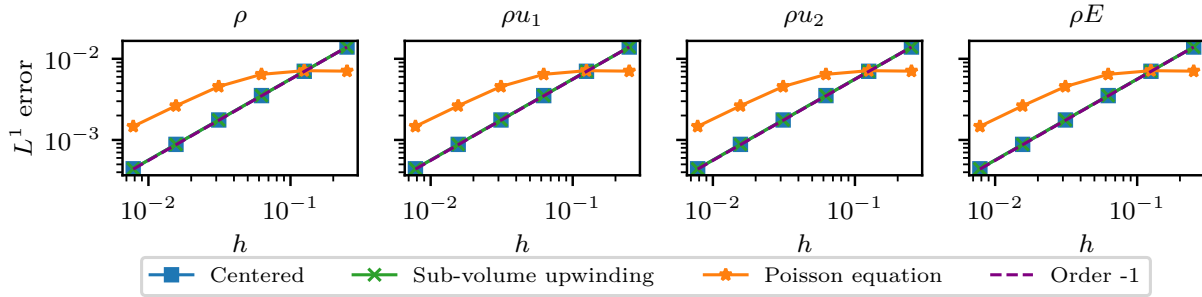
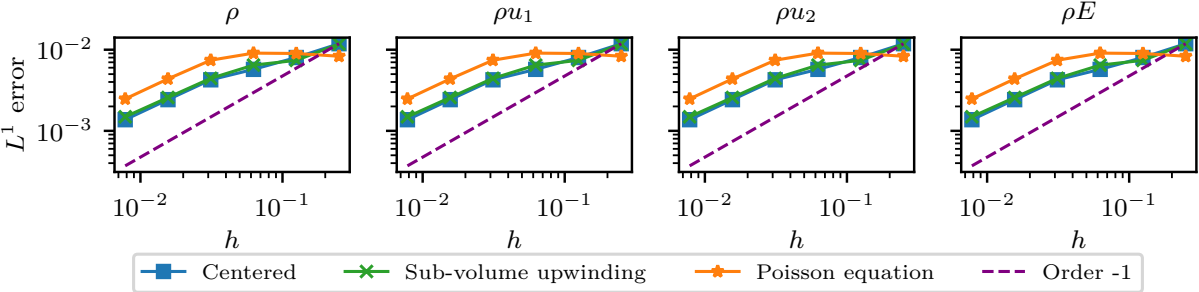
(A)  $\theta = \frac{1}{2}$ .(B)  $\theta = \frac{\pi}{4}$ .(C)  $\theta = 1$ .

FIGURE 6.  $L^1$  error on  $\rho$  (leftmost column),  $\rho u_1$  (second column),  $\rho u_2$  (third column) and  $\rho E$  (rightmost column), for the test case 5.1 – 1D gravity on a uniform mesh with  $\theta = \frac{1}{2}$  (top row),  $\theta = \frac{\pi}{4}$  (middle row), and  $\theta = 1$  (bottom row).

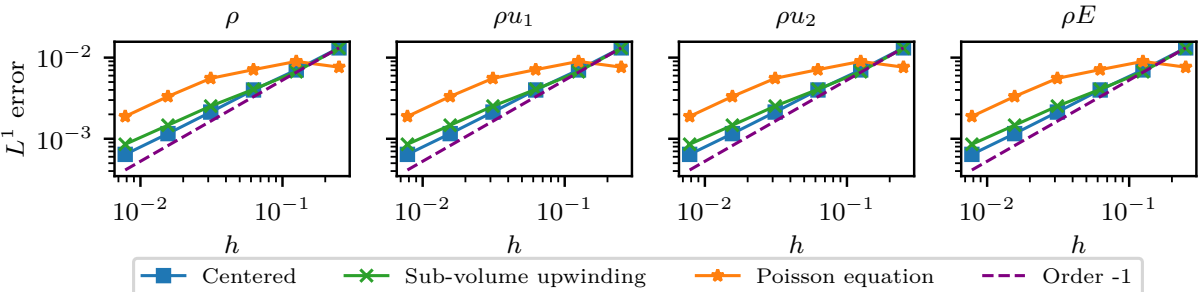
### 5.3. Friction

The source term is given by  $S = (0, -\lambda\rho\mathbf{u}, -\alpha\lambda\rho\|\mathbf{u}\|^2)^T$  with  $\alpha = \gamma/(\gamma - 1)$  and  $\lambda = 1$ . A solution to the system (1) writes:

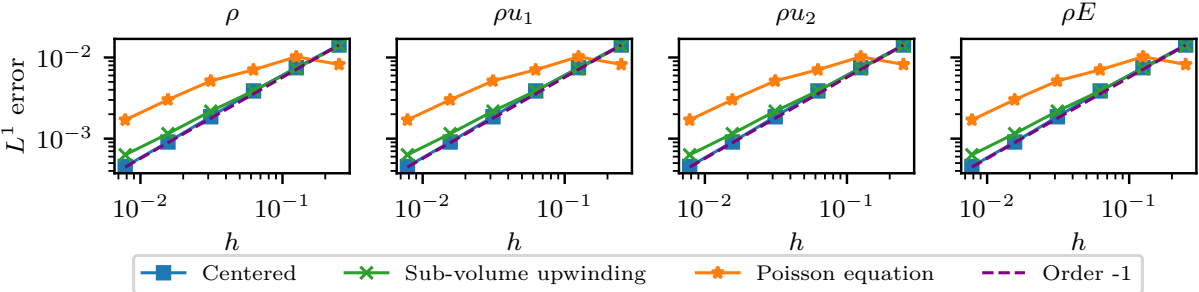
$$\rho = 1, \quad \mathbf{u} = \frac{1}{\sqrt{2}}(1, 1), \quad E(\mathbf{x}) = -\frac{\lambda}{\gamma - 1}\mathbf{u} \cdot \mathbf{x} + 1 + \frac{2\lambda}{\gamma - 1}.$$



(A)  $\theta = \frac{1}{2}$ .



(B)  $\theta = \frac{\pi}{4}$ .



(C)  $\theta = 1$ .

FIGURE 7.  $L^1$  error on  $\rho$  (leftmost column),  $\rho u_1$  (second column),  $\rho u_2$  (third column) and  $\rho E$  (rightmost column), for the test case 5.1 – 1D gravity on a random mesh with  $\theta = \frac{1}{2}$  (top row),  $\theta = \frac{\pi}{4}$  (middle row), and  $\theta = 1$  (bottom row).

One can easily check that  $E > \|\mathbf{u}\|^2/2$ . We set  $C_{CFL} = 0.25$ . The results obtained on a uniform mesh with different discretizations of the source term are given in Figure 11. The results obtained on a random mesh with different discretizations of the source term are given in Figure 12. The methods all are first order convergent for  $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$ .



#### 5.4. 2D Riemann problem

In this subsection we display the results obtained at time  $T = 0.25$  on a two-dimensional Riemann problem. Here, we consider the Lax-Liu test case number 12 of [34], where the initial data on each quadrant is piecewise constant (see Figure 8). Here, outflow boundary conditions are imposed.

We also add a one-dimensional gravity term  $\mathbf{g} = (0, -g)$ , with  $g = 5$  (see Figures 13b, 13c).

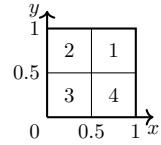


FIGURE 8. 2D Riemann problem Lax-Liu (case 12)

### 6. CONCLUSION

In this work we presented two methods to compute the numerical source term of the 2D Euler equations using ideas from [1, 2]. The fluxes are discretized using a composite scheme that generalizes the VFFC scheme (see [27]) to unstructured 2D meshes. We compared the centered discretization of the source term (discretized at the center of computational cells) with a local and a global approach. The local approach uses the sub-volumes of computational cells, and performs slightly better for a computational complexity of the same order of magnitude. The global approach solves a Poisson equation on the whole domain, however it is much more expensive and does not perform significantly better. The results that we obtain are not as good as we expected, but on the basis of composite (node/edge) numerical fluxes, we have now other ways to explore, for example extending the work of [32].

Nevertheless, on the one-dimensional gravity case, the sub-volume and centered discretizations yield very similar results. This benefit is observable on the two-dimensional gravity test, and in this case the sub-volume approach gives the same rate of convergence but with a lower error constant. The sub-volume approach is then preferable for source terms of type gravity.

#### A. CONSTRUCTION OF THE DIFFUSION SCHEME (30)

We follow [29, 35, 36] and propose here a composite scheme to solve the following Poisson problem:

$$\Delta v = f, \quad (37)$$

where  $v$  and  $f$  are scalar valued functions and with homogeneous Dirichlet boundary conditions. Equation (37) is integrated over a cell  $\Omega_j$ :

$$\int_{\partial\Omega_j} \nabla v \cdot \mathbf{n} = \int_{\Omega_j} f. \quad (38)$$

Using Proposition 3.2, we can approximate:

$$\int_{\partial\Omega_j} \nabla v \cdot \mathbf{n} \approx (1 - \theta) \sum_{r \in \Omega_j} (\nabla v)(\mathbf{x}_r) \cdot \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2} \in \Omega_j} (\nabla v)(\mathbf{x}_{r+\frac{1}{2}}) \cdot \mathbf{C}_j^{r+\frac{1}{2}}. \quad (39)$$

The gradient at the nodes is computed as follows. Using a Taylor expansion, one can write:

$$v_j = v_r + (\nabla v)(\mathbf{x}_r) \cdot (\mathbf{x}_j - \mathbf{x}_r) + O(h^2). \quad (40)$$

Multiplying (40) by  $\mathbf{C}_j^r$  and summing over all the cells around  $r$ :

$$\sum_{i|r \in \Omega_i} v_i \mathbf{C}_i^r = v_r \sum_{i|r \in \Omega_i} \mathbf{C}_i^r + \left( \sum_{i|r \in \Omega_i} \mathbf{C}_i^r \otimes (\mathbf{x}_i - \mathbf{x}_r) \right) (\nabla v)(\mathbf{x}_r) + O(h^3).$$

The sum  $v_r \sum_{i|r \in \Omega_i} \mathbf{C}_i^r$  always vanishes. Indeed, if the node  $r$  is inside the domain, we have  $\sum_{i|r \in \Omega_i} \mathbf{C}_i^r = \mathbf{0}$ . Otherwise, the node  $r$  is on the boundary and  $v_r$  is given by the Dirichlet boundary condition, which is 0. Therefore a good approximation of the gradient at node  $r$ , denoted by  $\mathbf{w}_r$ , reads as:

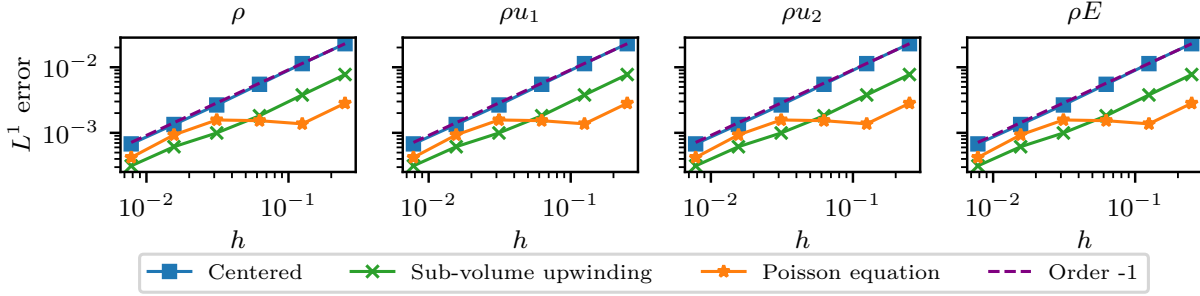
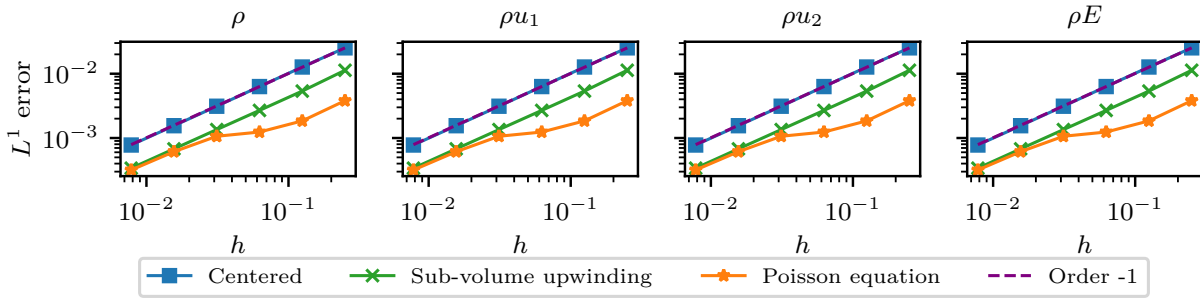
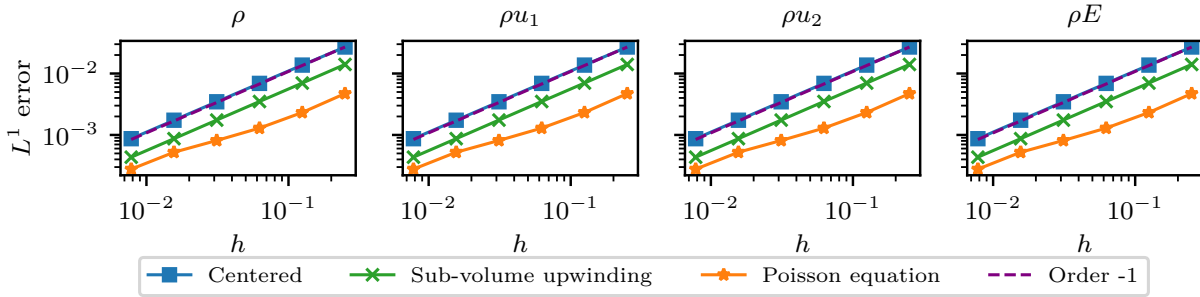

 (A)  $\theta = \frac{1}{2}$ .

 (B)  $\theta = \frac{\pi}{4}$ .

 (C)  $\theta = 1$ .

FIGURE 9.  $L^1$  error on  $\rho$  (leftmost column),  $\rho u_1$  (second column),  $\rho u_2$  (third column) and  $\rho E$  (rightmost column), for the test case 5.2 – 2D gravity on a uniform mesh with  $\theta = \frac{1}{2}$  (top row),  $\theta = \frac{\pi}{4}$  (middle row), and  $\theta = 1$  (bottom row).

$$\left( \sum_{i|r \in \Omega_i} \mathbf{C}_i^r \otimes (\mathbf{x}_i - \mathbf{x}_r) \right) \mathbf{w}_r = \sum_{i|r \in \Omega_i} v_i \mathbf{C}_i^r, \quad (41)$$

and the gradient at the edge  $r + 1/2$  is defined by:

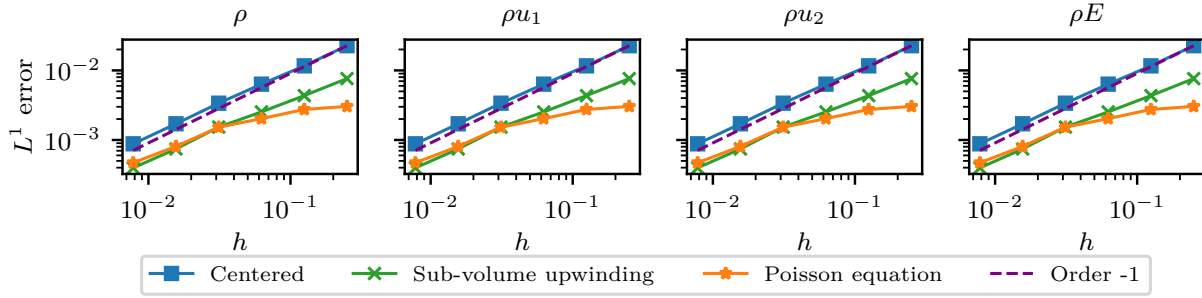
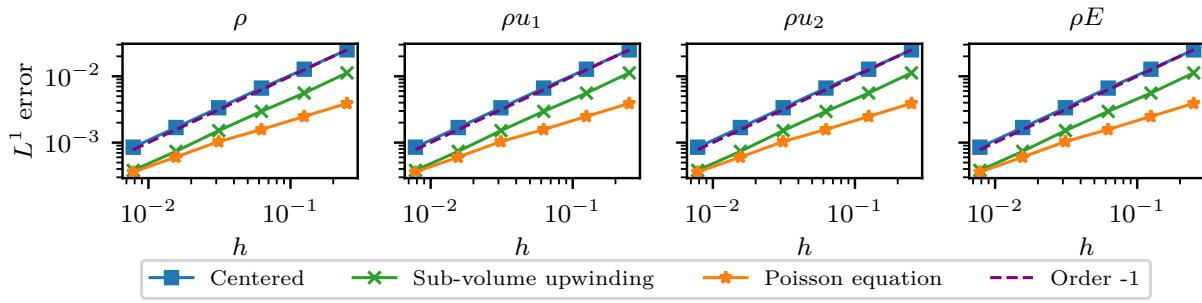
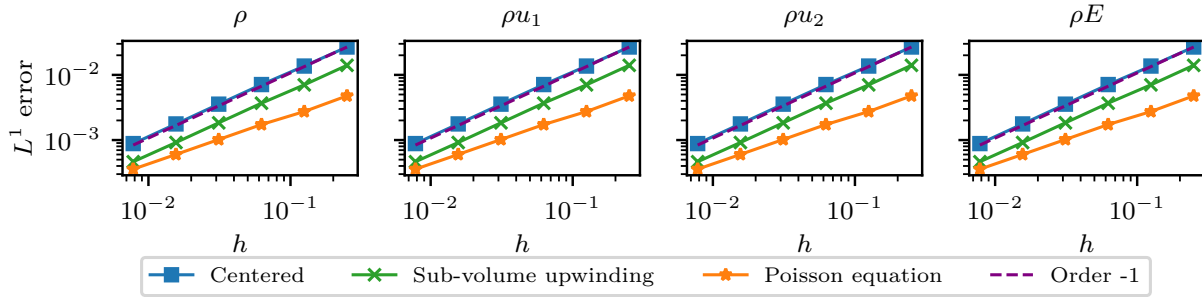
(A)  $\theta = \frac{1}{2}$ .(B)  $\theta = \frac{\pi}{4}$ .(C)  $\theta = 1$ .

FIGURE 10.  $L^1$  error on  $\rho$  (leftmost column),  $\rho u_1$  (second column),  $\rho u_2$  (third column) and  $\rho E$  (rightmost column), for the test case 5.2 – 2D gravity on a random mesh with  $\theta = \frac{1}{2}$  (top row),  $\theta = \frac{\pi}{4}$  (middle row), and  $\theta = 1$  (bottom row).

$$\mathbf{w}_{r+1/2} = \frac{\mathbf{w}_r + \mathbf{w}_{r+1}}{2}. \quad (42)$$

Therefore  $\mathbf{w}_{dof}$  is well defined and is consistent with  $\nabla v(\mathbf{x}_{dof})$ . In addition, using (42), the diffusion flux (39) reads:

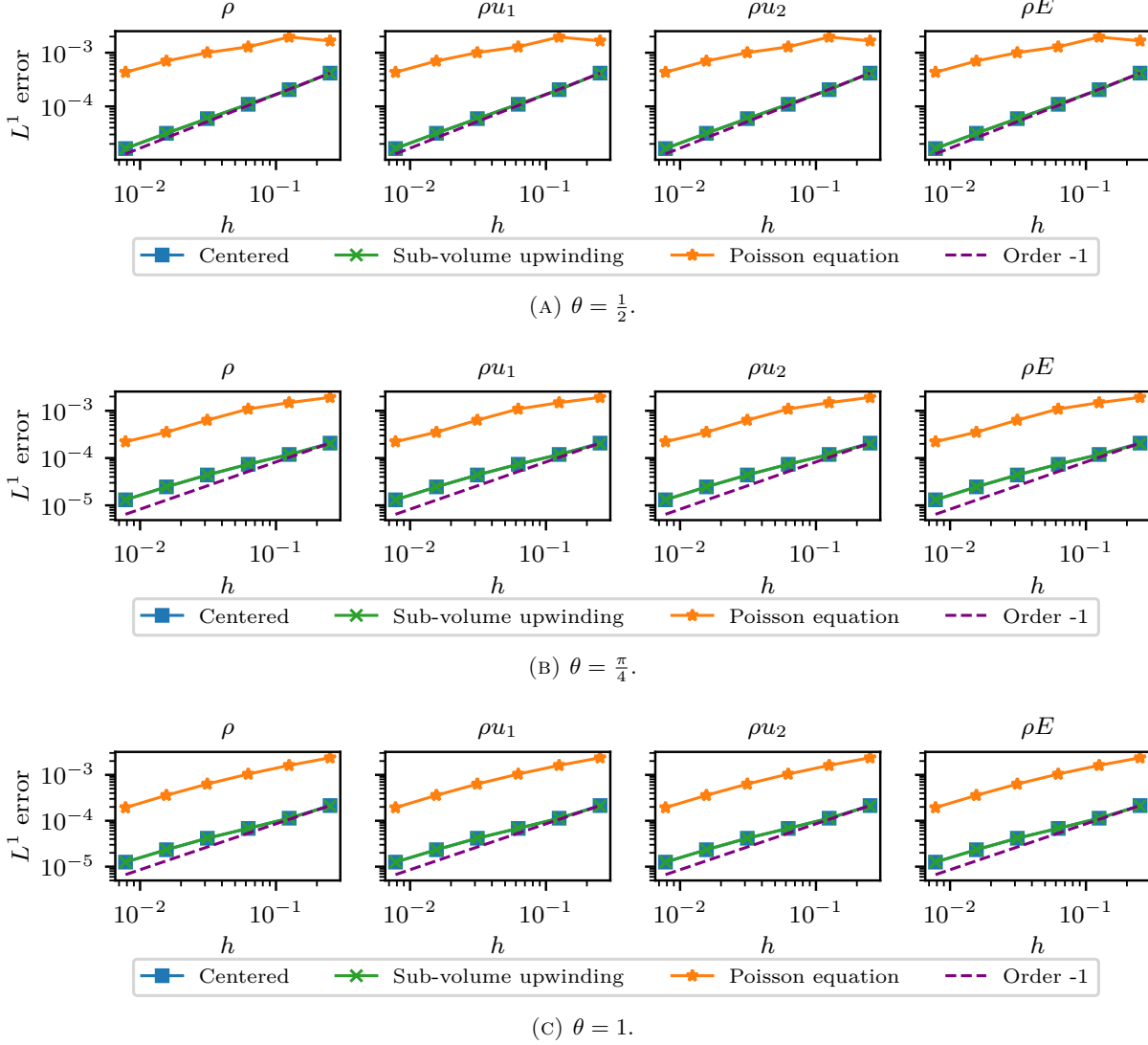


FIGURE 11.  $L^1$  error on  $\rho$  (leftmost column),  $\rho u_1$  (second column),  $\rho u_2$  (third column) and  $\rho E$  (rightmost column), for the test case 5.3 – Friction on a uniform mesh with  $\theta = \frac{1}{2}$  (top row),  $\theta = \frac{\pi}{4}$  (middle row), and  $\theta = 1$  (bottom row).

$$(1 - \theta) \sum_{r \in \Omega_j} \mathbf{w}_r \cdot \mathbf{C}_j^r + \theta \sum_{r + \frac{1}{2} \in \Omega_j} \mathbf{w}_{r + \frac{1}{2}} \cdot \mathbf{C}_j^{r + \frac{1}{2}} = \sum_{r \in \Omega_j} \mathbf{w}_r \cdot \mathbf{C}_j^r.$$

The matrix in (41) can be shown to be invertible if the mesh is not too deformed (see [35] and [37]). In practice, it is invertible as soon as the number of cells that contain the node  $r$  is different from 1. For the nodes located at the corner of the mesh (only one support cell, denoted as  $j$ ), we approximate:  $\mathbf{w}_r = v_j \mathbf{C}_j^r / \|\mathbf{C}_j^r\|^2$ .

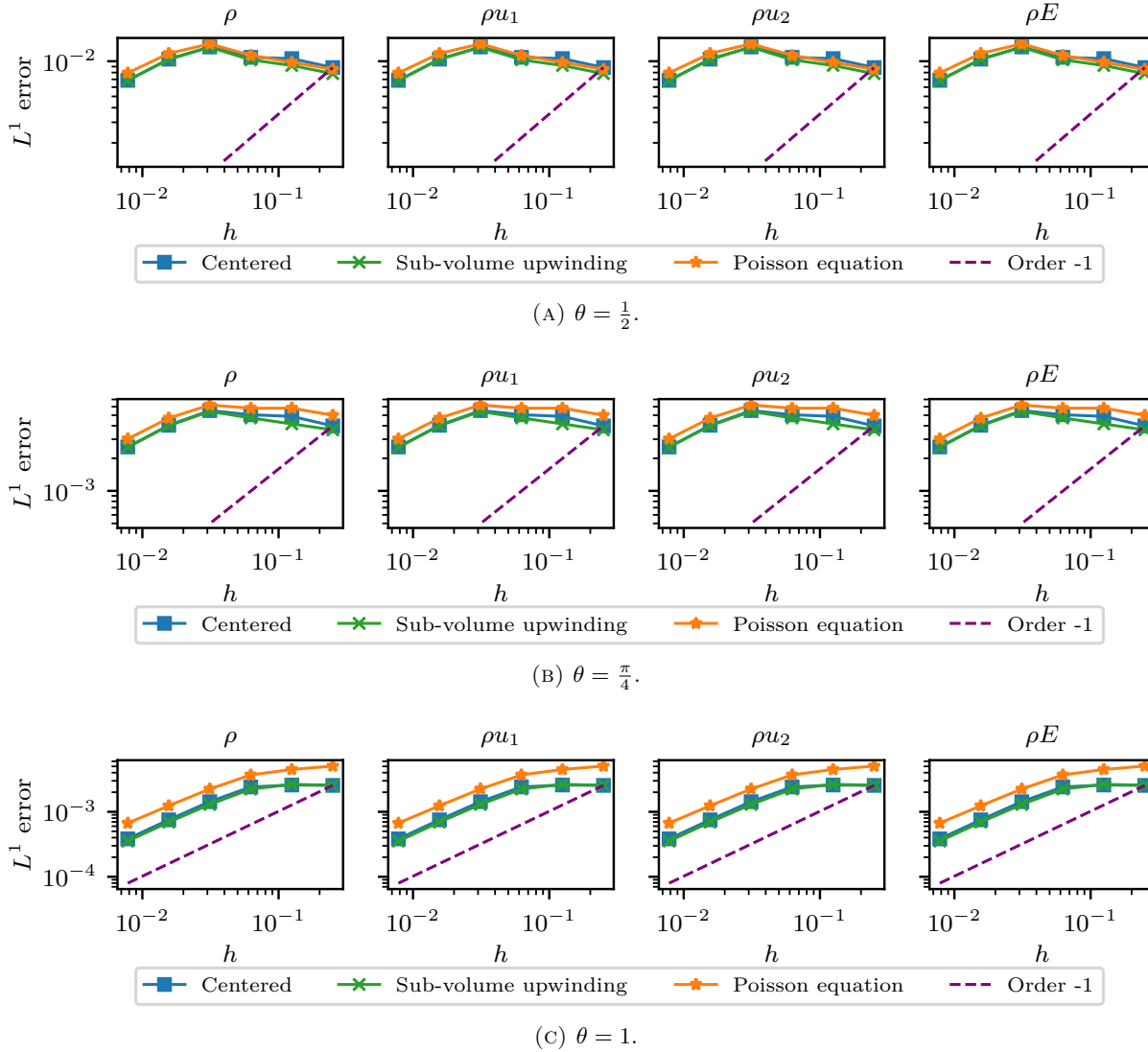


FIGURE 12.  $L^1$  error on  $\rho$  (leftmost column),  $\rho u_1$  (second column),  $\rho u_2$  (third column) and  $\rho E$  (rightmost column), for the test case 5.3 – Friction on a random mesh with  $\theta = \frac{1}{2}$  (top row),  $\theta = \frac{\pi}{4}$  (middle row), and  $\theta = 1$  (bottom row).

## REFERENCES

- [1] P. Hoch, “Nodal extension of approximate riemann solvers and nonlinear high order reconstruction for finite volume method on unstructured polygonal and conical meshes: the homogeneous case,” *HAL*, 2022.
- [2] F. Alouges, J.-M. Ghidaglia, and M. Tajchman, “On the interaction of upwinding and forcing for nonlinear hyperbolic systems of conservation laws,” 1999. Available at [https://www.researchgate.net/profile/Francois-Alouges/publication/2354257\\_On\\_the\\_Interaction\\_of\\_Upwinding\\_and\\_Forcing\\_for\\_Nonlinear\\_Hyperbolic\\_Systems\\_of\\_Conservation\\_Laws/links/541f083d0cf241a65a1aa84f/On-the-Interaction-of-Upwinding-and-Forcing-for-Nonlinear-Hyperbolic-Systems-of-Conservation-Laws.pdf](https://www.researchgate.net/profile/Francois-Alouges/publication/2354257_On_the_Interaction_of_Upwinding_and_Forcing_for_Nonlinear_Hyperbolic_Systems_of_Conservation_Laws/links/541f083d0cf241a65a1aa84f/On-the-Interaction-of-Upwinding-and-Forcing-for-Nonlinear-Hyperbolic-Systems-of-Conservation-Laws.pdf).
- [3] B. Després and C. Mazeran, “Lagrangian gas dynamics in two dimensions and lagrangian systems,” *Archive for Rational Mechanics and Analysis*, vol. 178, no. 3, pp. 327–372, 2005.

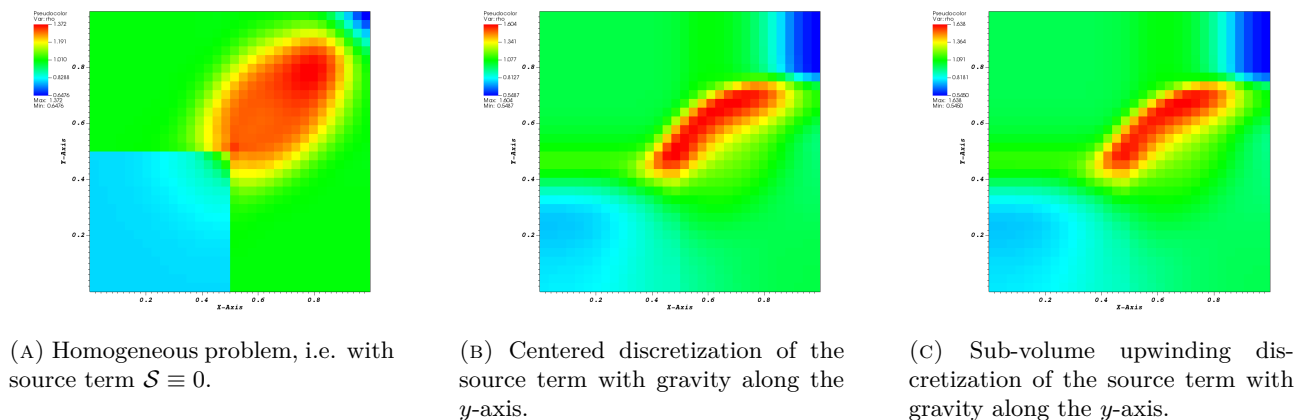


FIGURE 13. Approximate solution to a 2D Riemann problem, at final time  $T = 0.25$ . Solution obtained with no source term (left), with the centered discretization of the source term (center), and with the upwind discretization of the source term (right). For the three results shown,  $N_x = N_y = 32$ ,  $\theta = \pi/4$ ,  $C_{CFL} = .25$ .

- [4] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovadia, “A cell-centered lagrangian scheme for two-dimensional compressible flow problems,” *SIAM Journal on Scientific Computing*, vol. 29, no. 4, pp. 1781–1824, 2007.
- [5] D. J. Benson, “Computational methods in lagrangian and eulerian hydrocodes,” *Computer methods in Applied mechanics and Engineering*, vol. 99, no. 2-3, pp. 235–394, 1992.
- [6] E. Franck, “Design and numerical analysis of asymptotic preserving schemes on unstructured meshes. application to the linear transport and friedrichs systems,” *HAL*, vol. 2012, 2012.
- [7] A. Bernard-Champmartin, E. Deriaz, P. Hoch, G. Samba, and M. Schaefer, “Extension of centered hydrodynamical schemes to unstructured deforming conical meshes: the case of circles,” in *ESAIM: Proc.*, vol. 38, pp. 135–162, EDP Sciences, 2012.
- [8] B. Boutin, E. Deriaz, P. Hoch, and P. Navaro, “Extension of ALE methodology to unstructured conical meshes,” in *ESAIM: Proc.*, vol. 32, pp. 31–55, EDP Sciences, 2011.
- [9] L. Gosse and A.-Y. Leroux, “Un schema-equilibre adapte aux lois de conservation scalaires non-homogenes,” 1996.
- [10] J. M. Greenberg and A. Y. Leroux, “A well-balanced scheme for the numerical processing of source terms in hyperbolic equations,” *SIAM Journal on Numerical Analysis*, vol. 33, no. 1, p. 1–16, 1996.
- [11] F. Bouchut, *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources*. Frontiers in Mathematics, Basel ; Boston: Birkäuser, 2004.
- [12] M. Castro, J. M. Gallardo, J. A. López-García, and C. Parés, “Well-balanced high order extensions of godunov’s method for semilinear balance laws,” *SIAM Journal on Numerical Analysis*, vol. 46, no. 2, p. 1012–1039, 2008.
- [13] M. Castro, T. M. de Luna, and C. Parés, *Chapter 6 - well-balanced schemes and path-conservative numerical methods*, vol. 18 of *Handbook of numerical analysis*, p. 131–175. Elsevier, 2017. Citation Key: CASTRO2017131.
- [14] P. Chandrashekar and M. Zenk, “Well-balanced nodal discontinuous galerkin method for euler equations with gravity,” *Journal of Scientific Computing*, vol. 71, no. 3, p. 1062–1093, 2017.
- [15] E. Franck and L. Mendoza, “Finite volume scheme with local high order discretization of hydrostatic equilibrium for euler equations with external forces,” *Journal of Scientific Computing*, vol. 69, pp. 314–354, 2016.
- [16] R. J. LeVeque, “Balancing source terms and flux gradients in high-resolution godunov methods: The quasi-steady wave-propagation algorithm,” *Journal of Computational Physics*, vol. 146, no. 1, p. 346–365, 1998.
- [17] J. Shi, “A steady-state capturing method for hyperbolic systems with geometrical source terms,” *Mathematical Modelling and Numerical Analysis*, vol. 35, no. 4, p. 631–645, 2001.
- [18] P. Brufau, M. E. Vázquez-Cendón, and P. García-Navarro, “A numerical model for the flooding and drying of irregular domains: 1d and 2d modelling of shallow water flow,” *International Journal for Num. Meth. in Fluids*, vol. 39, no. 3, p. 247–275, 2002.
- [19] 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 Journal on Scientific Computing*, vol. 25, no. 6, p. 2050–2065, 2004.
- [20] M. J. Castro Díaz, T. Chacón Rebollo, E. D. Fernández-Nieto, and C. Parés, “On well-balanced finite volume methods for nonconservative nonhomogeneous hyperbolic systems,” *SIAM Journal on Scientific Computing*, vol. 29, no. 3, p. 1093–1126, 2007.

- [21] A. Canestrelli, A. Siviglia, M. Dumbser, and E. F. Toro, “Well-balanced high-order centred schemes for non-conservative hyperbolic systems. applications to shallow water equations with fixed and mobile bed,” *Advances in Water Resources*, vol. 32, no. 6, p. 834–844, 2009.
- [22] E. G. Fernández, M. J. C. Díaz, M. Dumbser, and T. M. De Luna, “An arbitrary high order well-balanced adler-dg numerical scheme for the multilayer shallow-water model with variable density,” *Journal of Scientific Computing*, vol. 90, no. 1, p. 52, 2022.
- [23] 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: Well-balanced scheme for the euler equations with gravity,” *International Journal for Num. Meth. in Fluids*, vol. 81, no. 2, p. 104–127, 2016.
- [24] M. J. Castro and C. Parés, “Well-balanced high-order finite volume methods for systems of balance laws,” *Journal of Scientific Computing*, vol. 82, no. 2, p. 48, 2020.
- [25] J. P. Berberich, P. Chandrashekar, and C. Klingenberg, “High order well-balanced finite volume methods for multi-dimensional systems of hyperbolic balance laws,” *Computers & Fluids*, vol. 219, p. 104858, 2021.
- [26] D. Chauveheid, “A new algorithm for surface tension forces in the framework of the FVCF–ENIP method,” *European Journal of Mechanics - B/Fluids*, vol. 50, pp. 175–186, 2015.
- [27] J.-M. Ghidaglia, A. Kumbaro, and G. L. Coq, “Une méthode volumes finis à flux caractéristiques pour la résolution numérique des systèmes hyperboliques de lois de conservation,” *C.R. Acad. Sci. Paris*, vol. 322, pp. 981–988, 1996.
- [28] M. H. Abbasi, S. Naderi Lordejani, C. Berg, L. Iapichino, W. H. A. Schilders, and N. van de Wouw, “An approximate well-balanced upgrade of godunov-type schemes for the isothermal euler equations and the drift flux model with laminar friction and gravitation,” *International Journal for Num. Meth. in Fluids*, vol. 93, no. 4, pp. 1110–1142, 2021.
- [29] X. Blanc, P. Hoch, and C. Lasuen, “Positive composite finite volume schemes for the diffusion equation on unstructured meshes.” working paper or preprint, 2022.
- [30] P. Hoch and O. Pironneau, “A vector Hamilton–Jacobi formulation for the numerical simulation of Euler flows,” *Comptes Rendus. Mathématique*, vol. 342, no. 2, pp. 151–156, 2006.
- [31] P. Roe, “Upwind differencing schemes for hyperbolic conservation laws with source terms,” in *Nonlinear Hyperbolic Problems, Proc. Adv. Res. Workshop, St. Etienne, 1986*, vol. 1270, pp. 41–45, 1987.
- [32] A. Bermudez and M. Vazquez, “Upwind methods for hyperbolic conservation laws with source terms,” *Comput. & Fluids*, pp. 1049–1071, 1994.
- [33] B. Després and E. Labourasse, “Stabilization of cell-centered compressible lagrangian methods using subzonal entropy,” *J. Comput. Phys.*, vol. 231, pp. 6559–6595, 2012.
- [34] P. Lax and X.-D. Liu, “Solution of two-dimensional riemann problems of gas dynamics by positive schemes,” *SIAM Journal on Scientific Computing*, vol. 19, pp. 319–340, 1998.
- [35] C. Buet, B. Després, and E. Franck, “Design of asymptotic preserving finite volume schemes for the hyperbolic heat equation on unstructured meshes,” *Numerische Mathematik*, vol. 122, no. 2, pp. 227–278, 2012.
- [36] X. Blanc, V. Delmas, and P. Hoch, “Asymptotic preserving schemes on conical unstructured 2d meshes,” *International Journal for Num. Meth. in Fluids*, vol. 93, no. 8, pp. 2763–2802, 2021.
- [37] X. Blanc, P. Hoch, and C. Lasuen, “Proof of uniform convergence for a cell-centered AP discretization of the hyperbolic heat equation on conical meshes.” working paper or preprint, 2022.