Flux-gradient and source term balancing for certain high resolution shock-capturing schemes

Vicent Caselles∗, Rosa Donat[†] and Gloria Haro∗

∗Departament de Tecnologia. Universitat Pompeu Fabra, Barcelona, Spain

† Dept. Matemàtica Aplicada, Facultat de Matemátiques, Universitat de València, Spain

November 4, 2004

Abstract

We present an extension of Marquina's flux formula, as introduced in [8], for the shallow water system. We show that the use of two different Jacobians at cell interfaces prevents the scheme from satisfying the exact C-property [2] while the approximate C-property is satisfied for higher order versions of the scheme. The use of a single Jacobian in Marquina's flux splitting formula leads to a numerical scheme satisfying the exact C-property, hence we propose a combined technique that uses Marquina's two sided decomposition when the two adjacent states are not close and a single decomposition otherwise. Finally, we propose a special treatment at wet/dry fronts and situations of dry bed generation.

Key Words: shallow water equations, high resolution shock capturing schemes, conservation property, well balanced schemes, flooding, drying.

AMS (MOS) subject classification: 65M06, 76M20, 35L65.

1 Introduction

Many practical problems involving shallow water flow in oceanography and atmospheric sciences are conveniently modeled by considering the one-dimensional or two dimensional Saint-Venant, or shallow water, system. This is a hyperbolic system of conservation laws that approximately describes various geophysical flows, such as rivers, coastal areas and even oceans when completed with a Coriolis term. For practical applications, the inclusion of a non-flat bottom topography is required, which leads to the occurrence of source terms of geometrical nature. The system can be written as follows

$$U_t + \mathrm{div}(F(U)) = S(\mathbf{x}, U). \tag{1}$$

Very often, numerical simulations for systems of this form are accomplished by using a fractional splitting technique, in which one alternates between solving the homogeneous system of conservation laws

$$U_t + \mathrm{div}(F(U)) = 0$$

and the system of ordinary differential equations,

$$U_t = S(\mathbf{x}, U).$$

This approach, however, performs very poorly in those situations where U_t is small relative to the other two terms, in particular when steady or quasi-steady solutions are being sought. For such solutions, highly accurate numerical simulations can only be obtained from numerical methods that 'respect' the balance that occurs between the flux gradient and the source term when U_t is small, and it is known [19] that this balance is not likely to be respected when using a fractional step approach.

The shallow water system is used to model real-life applications in which the flow regime is steady or quasi steady, and much effort has been devoted to design numerical techniques that are capable to preserve steady states at the discrete level as well as to accurately compute the evolution of small dynamical perturbations of these. The inclusion of the source term in a direct discretization of system (1) becomes then a non-trivial issue. Roe showed very early [21] that pointwise evaluation of the source term is not a suitable strategy, and the necessity of an *upwind* discretization of the source term is now widely recognized [21, 2, 19, 25].

For shallow water flow, the idea of 'source-term upwinding' lead Bermúdez and Vázquez-Cendón [2] to formulate the so-called C-property (for *Conservation* property), which prevents the propagation of parasitic waves in steady and quasi-steady flows. Independently, Greenberg and LeRoux [13] coined the term "well-balanced" for schemes that preserve steady states at the discrete level. These ideas have been explored and developed for shallow water flows in the recent literature [11, 1, 25] and a well established strategy is to combine a conservative scheme for homogeneous conservation laws with an 'appropriate' *upwind* discretization of the source term. The properties of the homogeneous 'solver' are important in the overall performance of the scheme and many of the main homogeneous solvers have been extended to the shallow water system. To name but a few, Bermúdez and Vázquez-Cendón [2] used Roe's scheme as the homogeneous solver. Vukovic and Sopta extended their approach [25] using the ENO and WENO schemes described in [15, 14, 16]. Relaxation schemes have also been considered [1, 6], etc.

In [7] Donat and Marquina develop a new numerical scheme that avoids the use of averaged quantities in computing the numerical flux function at cell interfaces. Marquina's numerical flux formula is based on the use of two jacobian matrices at each interface, from which unmixed, sided, characteristic information is extracted. It is seen in [7, 8] that this numerical flux can be used to design robust High Resolution Shock Capturing (HRSC) schemes with good properties with respect to certain numerical pathologies. In this paper we seek to obtain an extension of Marquina's flux formula for non-homogeneous conservation laws by incorporating the idea of flux gradient and source term balancing.

Since the source-term/flux-gradient balancing in [2, 25] is linked to the use of one Jacobian matrix at the interface, we follow a different strategy, described by Gascón and Corberán in [10]. There, the authors propose to write the source term in divergence form so that it can be incorporated into the flux vector of the homogeneous system to be later discretized in an upwind manner. To design the numerical technique to be applied to system (1) we proceed as in [22, 7]: we first examine the scalar case and then construct the numerical flux function for the system case by implementing the scalar numerical flux in each (local) characteristic field.

In studying the C-property for our extension of Marquina's scheme, we realize that the *exact* C-property is linked to the use of a unique Jacobian at each cell interface. It is interesting to notice that these 1-Jacobian schemes (1J in the rest of the paper) can also be considered as a flux-gradient/source-term balanced version of the Shu-Osher ENO schemes in [22], when applied to the shallow water system with topography. The extension turns out to be different from that in [25], since our 1J schemes satisfy the exact C property *independently* of the average interface state, while in [25], the exact C property is only obtained when the average interface state is the arithmetic mean of the left and right interface states.

The use of two Jacobian matrices prevents the scheme from satisfying the *exact* C-property, but the approximate C-property is satisfied as long as the order of accuracy is at least two. Since the use of two Jacobians at cell interfaces is still advantageous in situations where the left and right states are very different, we propose a *combined* 1J-2J scheme which seems to behave well in all cases. In addition, the 2J philosophy serves to design a simple correction that is able to handle dry states (existing or being formed) in a rather straightforward way.

The paper is organized as follows, section 2 contains the governing equations of the shallow water model. The numerical scheme is explained in section 3. It also contains the proof of the C-property as well as the special treatment of wet/dry fronts and dry bed generation. We give some examples of experiments in 1D in section 4. Finally, conclusions can be found in section 5.

2 The shallow water system

The two-dimensional shallow water equations represent mass and momentum conservation in two dimensional domains. They are derived by depth averaging the Navier-Stokes equations, neglecting diffusion of momentum by viscous and turbulent effects and not including wind effects or Coriolis force terms. Ignoring also friction losses, the source term is only due to the geometry of the bottom topography and the resulting system of equations becomes (see [24] and references therein)

$$U_t + F(U)_x + E(U)_y = S \tag{2}$$

$$\begin{pmatrix} h \\ q_1 \\ q_2 \end{pmatrix}_t + \begin{pmatrix} q_1 \\ \frac{q_1^2}{h} + \frac{1}{2}gh^2 \\ \frac{q_1 q_2}{h} \end{pmatrix}_x + \begin{pmatrix} q_2 \\ \frac{q_1 q_2}{h} \\ \frac{q_2^2}{h} + \frac{1}{2}gh^2 \end{pmatrix}_y = \begin{pmatrix} 0 \\ -ghz_x \\ -ghz_y \end{pmatrix}$$

where h is the water depth, q_1 and q_2 are the two components of the discharge, and z is the bottom topography. This is a two-dimensional hyperbolic system of conservation laws with source terms, whose eigenstructure is well known (see e.g. [24]).

The one-dimensional shallow water system

$$\begin{array}{lll} h_t & + & q_x = 0 \\ q_t & + & \left(\frac{q^2}{h} + \frac{1}{2}gh^2\right)_x = -ghz_x \end{array} \tag{3}$$

has also practical interest in itself. A flat bottom topography leads to an homogeneous system of conservation laws with two genuinely nonlinear fields. The mathematical theory for this homogeneous system is well known and can be found e.g. in [18, 24]. One important feature, which will be particularly relevant to our discussion is the fact that for a Riemann problem with left and right states (h_L, q_L), (h_R, q_R) the *depth positivity condition*

$$u_R - u_L \leq 2\sqrt{gh_L} + 2\sqrt{gh_R} \tag{4}$$

ensures a solution with $h > 0$. If this condition is not satisfied, the solution of the Riemann problem involves a forward rarefaction and a backward rarefaction with a dry-bed (equivalent to a vacuum state in gas-dynamics) in between [24].

3 The numerical scheme

As described in [8, 7], Marquina's scheme can be interpreted as a *characteristic based scheme* that avoids the use of any artificially constructed averaged quantities at cell interfaces.

Characteristic-based schemes exploit the eigenstructure of the Jacobian matrix of the flux vector $J(u) = F'(u)$ to transform a homogeneous nonlinear system into a system of (hopefully) nearly independent scalar equations. These can be independently discretized in an upwind manner and then the system is transformed back into the original variables using the same eigen-decomposition. This approach becomes one of the simplest ways to extend to systems numerical schemes designed for scalar equations.

As observed in [8], the Jacobian matrix of the convective flux vector is quite important to any characteristic based scheme, as it defines the local linearization of the nonlinear problem. It determines the transformation to the local characteristic fields, and thus the local upwind directions, as well as what quantities are to be upwind differenced.

In a typical characteristic-based scheme, only the values of the solution at cell centers are known, while the numerical flux functions are computed at cell boundaries. Some form of reconstruction (usually polynomial) is then required to compute the interface values required by the flux computation. While it makes little difference what sort of (consistent) approximation is used in smooth regions, the situation can change drastically between nodes in an unresolved steep gradient, or when the left and right states have significantly different properties. The results in [8, 7] demonstrate that the use of averaged Jacobians, interpolated from left and right nodal states, can lead to numerically pathological behavior in the approximate solution. Instead, Marquina's flux formula uses the unambiguous values for the left and right states to compute *two* characteristic decompositions at each cell interface. The upwind information is extracted from considering non-averaged information, which seems to help in avoiding (or diminishing) the aforementioned pathologies. For full details on the scheme for homogeneous systems (with applications), we refer the reader to [8, 7, 20].

The key step in the extension to (2) is the discretization of the source term, which can be split as follows,

$$S = S_1 + S_2 = \begin{pmatrix} 0 \\ -ghz_x \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -ghz_y \end{pmatrix}$$

We follow [10], and define the functions $B(x, y, t)$ and $C(x, y, t)$ as

$$B(x,y,t) = -\int_{\bar{x}}^{x} S_1(s,y,t)ds = \begin{pmatrix} 0 \\ \beta(x,y,t) \\ 0 \end{pmatrix}, \quad C(x,y,t) = -\int_{\bar{y}}^{y} S_2(x,s,t)ds = \begin{pmatrix} 0 \\ \kappa(x,y,t) \\ 0 \end{pmatrix}$$

with $\beta(x, y, t) = \int_{\bar{x}}^{x} gh(s, y, t)z_x(s, y)ds$ and $\kappa(x, y, t) = \int_{\bar{y}}^{y} gh(x, s, t)z_y(x, s)ds$. Then, we write system (2) as follows,

$$U_t + (F + B)_x + (E + C)_y = 0 \qquad (5)$$

The numerical technique we propose follows the general design principles in [22]. Hence, we consider a *method of lines* approach in which the time integration is performed via a TVD-Runge-Kutta method (see [22]). The spatial terms are treated in a dimension by dimension fashion, thus, it is sufficient to give a detailed description of the technique for the one-dimensional system (3). As in [8], we consider first the case of a scalar equation in one dimension.

3.1 Scalar equation

Let us consider the scalar, one dimensional version of system (3),

$$w_t + f(w)_x = s(x, w). \tag{6}$$

Following [10], we define $b(x,t) = \int_{\bar{x}}^{x} -s(y, w(y,t))dy$, and rewrite (6) as

$$w_t + g_x = 0 \tag{7}$$

with the *combined flux* function $g := f + b$.

We follow a *method of lines* approach and discretize the space variable in (7) in a 'conservative' fashion,

$$w_t + \frac{G_{i+1/2} - G_{i-1/2}}{\Delta x} = 0 \tag{8}$$

where $G_{i+1/2}$ is the numerical *combined-flux* function.

Dropping the time variable for the sake of simplicity, we note as in [22], that if $\phi(\xi)$ satisfies

$$g(w(x), x) = \frac{1}{\Delta x}\int_{x-\Delta x/2}^{x+\Delta x/2} \phi(\xi)d\xi$$

then

$$g(w(x), x)_x = \frac{1}{\Delta x}\left(\phi(x + \Delta x/2) - \phi(x - \Delta x/2)\right)$$

Thus, we seek to construct the numerical flux function $G_{i+1/2}$ as an approximation to $\phi(x_{i+1/2})$ of the appropriate order of accuracy. Following the construction *via primitive function* [22] we can accomplish this task simply by knowing $g_i = g(w_i, x_i) = f_i + b_i$.

It is well known that 'upwinding' is an essential part of any numerical scheme for hyperbolic conservation laws. In our construction of $G_{i+1/2}$, the upwind direction is determined by the sign of $f'(w)$ in $[x_i, x_{i+1}]$. This choice is simpler from that in [10], where the quantities that determine the upwind directions are numerically computed from g, but it is consistent with other implementations [21, 2, 19, 25] and justified by the examination of the particular case $f(w) = aw$, $s(x,w) = k(x)w$ [21]

$$w(x,t)_t + aw(x,t)_x = k(x)w(x,t) \tag{9}$$

whose the solution is

$$w(x,t) = w(x - at, 0) + \int_{\bar{x}}^{t} k(x - a(t-s))w(x - a(t-s), s)ds \tag{10}$$

and where each term on the RHS exhibits an upwind domain of dependence, which is determined by the wind direction $f'(w) = a$.

With these observations, we propose to use the ENO-RF construction (see [8, 22] or Algorithm ENO-3 below for specific details) directly on the *combined flux* data $g_i = f_i + b_i$, that is

$$G_{i+1/2} = \begin{cases} \hat{g}^{Roe}_{i+1/2} & \text{if } f' \text{ does not change sign in } [w_i, w_{i+1}] \\ \hat{g}^{LLF}_{i+1/2} & \text{else} \end{cases} \tag{11}$$

Here $\hat{g}^{Roe}$ refers to the ENO-Roe numerical flux construction and $\hat{g}^{LLF}$ the 'Local Lax-Friedrichs' construction in [22]. For the sake of completeness and ease of reference, we give the explicit formulas for the third order ENO reconstruction (see also [8]). For the ENO-Roe numerical flux we have

$$\hat{g}^{Roe}_{i+1/2} = g_k + c(2(i-k)+1)\Delta x + c_*(3(i-k_*)^2 - 1)\Delta x^2 \tag{12}$$

where the indexes k and k_*, and the quantities c and c_* are computed by the ENO-3 Algorithm.

Algorithm ENO-3

```
Step 1. If $f' > 0$ then $k = i$
                    else $k = i+1$
Step 2.    If $|g[k,k+1]| \leq |g[k-1,k]|$   then $k_* = k$, $c = \frac{1}{2}g[k,k+1]$
                                              else $k_* = k-1$, $c = \frac{1}{2}g[k-1,k]$
Step 3.       If $|g[k_*,k_*+1,k_*+2]| \leq |g[k_*-1,k_*,k_*+1]|$   then $c_* = \frac{1}{3}g[k_*,k_*+1,k_*+2]$
                                                                     else $c_* = \frac{1}{3}g[k_*-1,k_*,k_*+1]$
```

For the LLF-flux we need to define

$$g^+_l = .5(g_l + \alpha_{i+1/2} w_l), \qquad l = i-2,\dots,i+2 \tag{13}$$

$$g^-_l = .5(g_l - \alpha_{i+1/2} w_l), \qquad l = i-1,\dots,i+3 \tag{14}$$

where

$$\alpha_{i+1/2} = \max\{|f'(w)|, w \in [w_i, w_{i+1}]\}$$

then

$$\hat{g}^{LLF}_{i+1/2} = \hat{g}^+_{i+1/2} + \hat{g}^-_{i+1/2} \tag{15}$$

where

$$\hat{g}^+_{i+1/2} = g^+_i + c^+\Delta x + c^+_*(3(i-k^+_*)^2 - 1)\Delta x^2 \tag{16}$$

$$\hat{g}^-_{i+1/2} = g^-_{i+1} - c^-\Delta x + c^-_*(3(i-k^-_*)^2 - 1)\Delta x^2 \tag{17}$$

the quantities c^+, k^+_* and c^+_* are computed by setting $k = i$ in Step 1 of Algorithm ENO-3 and proceeding with g^+ instead of g as specified in Steps 2 and 3. For $\hat{g}^-_{i+1/2}$ we set $k = i+1$ in Step 1 and proceed analogously with g^-.

Taking into account formulas (12) to (17) we see that the numerical flux function $G_{i+1/2}$ in (11) can be written as follows:

$$G_{i+1/2} = \mathcal{G}_{i+1/2} + \Delta x C_{i+1/2} + \Delta x^2 D_{i+1/2} = \mathcal{G}_{i+1/2} + HOT_{i+1/2}, \tag{18}$$

where the term $\mathcal{G}_{i+1/2}$ collects the first order contribution while the terms containing $C_{i+1/2}$ and $D_{i+1/2}$ are second-order and third-order correction terms. Using equations (12) to (17) we can write

$$\mathcal{G}_{i+1/2} = \begin{cases} f_i + b_i & \text{if } f' > 0 \text{ in } [w_i, w_{i+1}] \\ f_{i+1} + b_{i+1} & \text{if } f' < 0 \text{ in } [w_i, w_{i+1}] \\ \frac{1}{2}(f_i + f_{i+1}) + \frac{1}{2}(b_i + b_{i+1}) + \frac{1}{2}\alpha_{i+1/2}(w_i - w_{i+1}) & \text{else} \end{cases} \tag{19}$$

Notice that given an index l,

$$g[l,l+1]=\frac{g_{l+1}-g_l}{\Delta x}=f[l,l+1]+\frac{1}{\Delta x}\int_{x_l}^{x_{l+1}}-s(w(x),x)dx$$

so that all the terms in $HOT_{i+1/2}$ involve only quantities of the form

$$b_{l,l+1}=\int_{x_l}^{x_{l+1}}-s(w(x),x)dx \tag{20}$$

From a numerical point of view, these quantities admit a better numerical treatment than the integral expressions of the form $b_l=\int_{x_{-1/2}}^{x_l}-s(w(x),x)dx$ appearing in (19). In order to design a computationally convenient method, we shall deduce an equivalent expression for the flux difference $\mathcal{G}_{i+1/2}-\mathcal{G}_{i-1/2}$ in terms of the quantities $b_{l,l+1}$. Observe that

$$\mathcal{G}_{i+1/2}-\frac{b_i}{2}-\frac{b_{i+1}}{2}=\begin{cases} f_i-\frac{1}{2}b_{i,i+1} & \text{if } f'>0 \text{ in } [w_i,w_{i+1}] \\ f_{i+1}+\frac{1}{2}b_{i,i+1} & \text{if } f'<0 \text{ in } [w_i,w_{i+1}] \\ \frac{1}{2}(f_i+\alpha_{i+1/2}w_i)+\frac{1}{2}(f_{i+1}-\alpha_{i+1/2}w_{i+1}) & \text{else} \end{cases} \tag{21}$$

where the RHS of expression (21) involves only integrals between two consecutive cells, i.e. $b_{l,l+1}$ terms.

Taking into account (21) and that $b_{i,j}=b_j-b_i$, we can re-write the first order flux difference as follows,

$$\begin{aligned}\mathcal{G}_{i+1/2}-\mathcal{G}_{i-1/2} &= \left(\mathcal{G}_{i+1/2}-\tfrac{1}{2}b_i-\tfrac{1}{2}b_{i+1}\right)-\left(\mathcal{G}_{i-1/2}-\tfrac{1}{2}b_i-\tfrac{1}{2}b_{i-1}\right)+\tfrac{1}{2}b_{i+1}-\tfrac{1}{2}b_{i-1}\\ &= \left(\mathcal{G}_{i+1/2}-\tfrac{1}{2}b_i-\tfrac{1}{2}b_{i+1}+\tfrac{1}{2}b_{i,i+1}\right)-\left(\mathcal{G}_{i-1/2}-\tfrac{1}{2}b_i-\tfrac{1}{2}b_{i-1}-\tfrac{1}{2}b_{i-1,i}\right)\\ &= \mathcal{G}^+_{i+1/2}-\mathcal{G}^-_{i-1/2}\end{aligned}$$

Notice that the *modified fluxes* $\mathcal{G}^{\pm}$ only involve integral expressions of the form $b_{l,l+1}$. Indeed, carrying out the algebra we easily arrive at

$$\mathcal{G}^+_{i+1/2}=\begin{cases} f_i & \text{if } f'>0 \text{ in } [w_i,w_{i+1}] \\ f_{i+1}+b_{i,i+1} & \text{if } f'<0 \text{ in } [w_i,w_{i+1}] \\ \frac{1}{2}(f_i+\alpha_{i+1/2}w_i)+\frac{1}{2}(f_{i+1}-\alpha_{i+1/2}w_{i+1})+\frac{1}{2}b_{i,i+1} & \text{else} \end{cases} \tag{22}$$

$$\mathcal{G}^-_{i+1/2}=\begin{cases} f_i-b_{i,i+1} & \text{if } f'>0 \text{ in } [w_i,w_{i+1}] \\ f_{i+1} & \text{if } f'<0 \text{ in } [w_l,w_r] \\ \frac{1}{2}(f_i+\alpha_{i+1/2}w_i)+\frac{1}{2}(f_{i+1}-\alpha_{i+1/2}w_{i+1})-\frac{1}{2}b_{i,i+1} & \text{else} \end{cases} \tag{23}$$

From these expressions, we see that $\mathcal{G}^+_{i+1/2}$ collects a contribution from the source term *only if* there is wind coming from the right at the interface. On the other hand, $\mathcal{G}^-_{i+1/2}$ collects a source term contribution *only if* there is wind coming from the left at the interface. Notice that

$$\mathcal{G}^+_{i+1/2}-\mathcal{G}^-_{i+1/2}=b_{i,i+1}$$

hence the *signed* numerical fluxes, $\mathcal{G}^{\pm}_{i+1/2}$, provide an effective *upwind* splitting of the source term contribution at the $i+1/2$ interface

It is illustrative to write down the flux difference for the particular case (9). For $a>0$ (analogously for $a<0$) we have

$$\mathcal{G}_{i+1/2}-\mathcal{G}_{i-1/2}=\mathcal{G}^+_{i+1/2}-\mathcal{G}^-_{i-1/2}=f_i-f_{i-1}+\int_{x_{i-1}}^{x_i}s(z,w(z))dz$$

this expression clearly displays the *upwind* contribution of the source term.

When the flux $f(w)$ is nonlinear, the expressions above provide easily computable numerical flux functions that incorporate an upwind source-term contribution. We refer to Figure 1 for a better understanding of which cell contains source term contributions, according to the sign of f'.

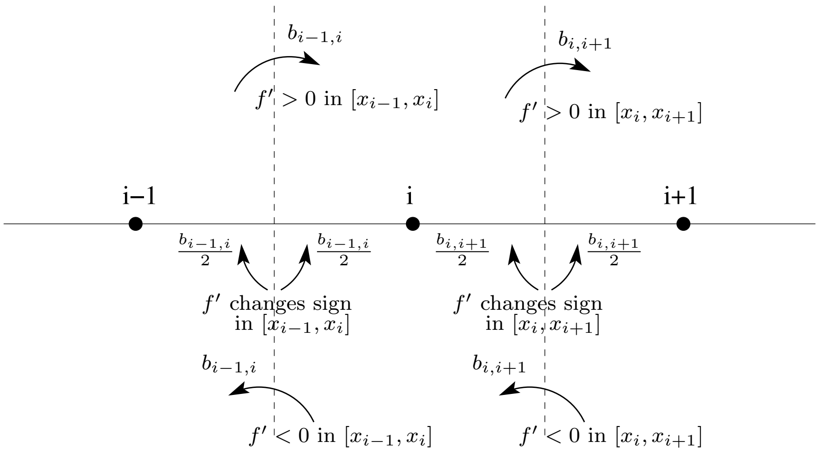

$b_{i-1,i}$
$f' > 0$ in $[x_{i-1}, x_i]$
$b_{i,i+1}$
$f' > 0$ in $[x_i, x_{i+1}]$
i−1
i
i+1
$\frac{b_{i-1,i}}{2}$
$\frac{b_{i-1,i}}{2}$
$\frac{b_{i,i+1}}{2}$
$\frac{b_{i,i+1}}{2}$
f' changes sign in $[x_{i-1}, x_i]$
f' changes sign in $[x_i, x_{i+1}]$
$b_{i-1,i}$
$b_{i,i+1}$
$f' < 0$ in $[x_{i-1}, x_i]$
$f' < 0$ in $[x_i, x_{i+1}]$

Figure 1: Contribution of source terms (at first order) involving the cell i and its both contiguous cells according to the sign of the characteristic velocities at the vicinity of each cell wall.

To obtain a high resolution scheme, let us define

$$G^+_{i+1/2} = \mathcal{G}^+_{i+1/2} + HOT_{i+1/2}, \qquad G^-_{i+1/2} = \mathcal{G}^-_{i+1/2} + HOT_{i+1/2}$$

where $HOT_{i+1/2} = G_{i+1/2} - \mathcal{G}_{i+1/2}$ are the higher order terms obtained from the polynomial expressions (12),(16) and (17). Thus

$$G_{i+1/2} - G_{i-1/2} = G^+_{i+1/2} - G^-_{i-1/2} \tag{24}$$

Using (24), the semi-discrete formulation (8) can be equivalently written as

$$w_t + \frac{G^+_{i+1/2} - G^-_{i-1/2}}{\Delta x} = 0 \tag{25}$$

where the computation of the combined fluxes $G^{\pm}_{i+1/2}$ only involve integrals of the source term between consecutive cell centers. Equation (24) will be the starting point for our extension to the system case.

A fully discrete expression is obtained by numerically approximating the integral terms $b_{i,i+1}$. it should be noted that the numerical integration technique employed here has an influence on the global order of accuracy. In the next section we shall describe our choice for the shallow water case.

3.2 Extension to nonlinear systems

To carry out the extension to the 1D system (3), we write it in the form $U_t + G_x = 0$, with $G(U,x) = F(U) + B(U,x)$ and $B(U,x) = (0, \int_{\bar{x}}^{x} ghz_x ds)^T$. We propose to use a semi-discrete formulation of the type

$$U_t + \frac{G^+_{i+1/2} - G^-_{i-1/2}}{\Delta x} = 0, \tag{26}$$

where the computation of $G^{\pm}_{i+1/2}$ only involves integral terms over consecutive cell centers and follows the basic design strategy in Marquina's flux formula: two states are computed at each side of a cell-interface, U^L and U^R, and the numerical flux functions are obtained by applying the scalar algorithm to "sided" local characteristic fluxes.

The states U^L and U^R at each side of a given interface are obtained by ENO interpolation of the physical variables as specified in [8]. Unless specifically stated, the order of the interpolation used to compute these states is the same as the order of the scheme.

Given $U^L = U^L_{i+1/2}$ and $U^R = U^R_{i+1/2}$, the left and right states at the $i+1/2$ cell-interface, the flux functions $G^{\pm}_{i+1/2}$ shall be defined as

$$G^{\pm}_{i+1/2} = \sum_{p=1}^{2} (\tilde{G}^{\pm}_{i+1/2})^{p,L} R^p(U^L) + (\tilde{G}^{\pm}_{i+1/2})^{p,R} R^p(U^R) \tag{27}$$

where $L^p(U^L)$, $R^p(U^L)$ ($L^p(U^R)$, $R^p(U^R)$), $p = 1, 2$, are the left and right eigenvectors of the Jacobian matrix $J(U) = F'(U)$, associated to the eigenvalues $\lambda^p(U^L)$ ($\lambda^p(U^R)$), and the local *modified* characteristic fluxes $(\tilde{G}^{p,\pm})^{L,R}$ are computed *as in the scalar case.* We give next a precise description of the computation of these numerical fluxes. In what follows $B_{j,j+1} = B_{j+1} - B_j = (0, \int_{x_j}^{x_{j+1}} ghz_x ds)^T$. In the fully discrete scheme, the integral in the second component is substituted by the discrete expression derived in the next section. We recall that for a scheme of order r ($r = 1, 2$ or 3 in this paper), the index j below runs from $j = i-r, \ldots, i+r$.

2J-Numerical flux function: For $p = 1, 2$

- If $\lambda^p(U^L) > 0$ and $\lambda^p(U^R) > 0$ then wind from the left: use ENO-Roe construction
  - Compute the first order contributions , $\mathcal{G}^{\pm}_{i+1/2}$, from (22) and (23), setting
    $f' > 0$; $f_j = L^p(U^L) \cdot F_j$, $b_{j,j+1} = L^p(U^L) \cdot B_{j,j+1}$.
  - Compute the $HOT^L_{i+1/2}$ terms using Algorithm ENO-3 with
    $k = i$, $g_j = L^p(U^L) \cdot (F_j + B_j)$.

  Then define
  $(\tilde{G}^{p,\pm}_{i+1/2})^L = \mathcal{G}^{\pm}_{i+1/2} + HOT^L_{i+1/2}, \qquad (\tilde{G}^{p,\pm}_{i+1/2})^R = 0.$

- If $\lambda^p(U^L) < 0$ and $\lambda^p(U^R) < 0$ then wind from the right: Use ENO-Roe construction.
  - Compute the first order contributions, $\mathcal{G}^{\pm}_{i+1/2}$, from (22) and (23), setting
    $f' < 0$; $f_j = L^p(U^R) \cdot F_j$, $b_{j,j+1} = L^p(U^R) \cdot B_{j,j+1}$.
  - Compute the $HOT^R_{i+1/2}$ terms using Algorithm ENO-3 with
    $k = i+1$, $g_j = L^p(U^R) \cdot (F_j + B_j)$

  Then define
  $(\tilde{G}^{p,\pm}_{i+1/2})^L = 0, \qquad (\tilde{G}^{p,\pm}_{i+1/2})^R = \mathcal{G}^{\pm}_{i+1/2} + HOT^R_{i+1/2};$

- If $\lambda^p(U^L)\lambda^p(U^R) \leq 0$ then mixed wind: Use the LLF construction.
  - Define $\alpha_{i+1/2} = \max(|\lambda^p(U^L)|, |\lambda^p(U^R)|)$

– Define

$$
\begin{aligned}
(\tilde{G}^{p,+}_{i+1/2})^L =&\quad L^p(U^L)\cdot \tfrac{1}{2}\left(F_i+\alpha_{i+1/2}U_i\right)+HOT^L_{i+1/2}\\
(\tilde{G}^{p,-}_{i+1/2})^L =&\quad L^p(U^L)\cdot \tfrac{1}{2}\left(F_i+\alpha_{i+1/2}U_i-B_{i,i+1}\right)+HOT^L_{i+1/2}
\end{aligned}
$$

Compute the $HOT^L_{i+1/2}$ terms as in (16) with $g^+_j=\frac{1}{2}L^p(U^L)(F_j+B_j+\alpha_{i+1/2}U_j)$.

$$
\begin{aligned}
(\tilde{G}^{p,+}_{i+1/2})^R =&\quad L^p(U^R)\cdot \tfrac{1}{2}\left(F_{i+1}-\alpha_{i+1/2}U_{i+1}+B_{i,i+1}\right)+HOT^R_{i+1/2}\\
(\tilde{G}^{p,-}_{i+1/2})^R =&\quad L^p(U^R)\cdot \tfrac{1}{2}\left(F_{i+1}-\alpha_{i+1/2}U_{i+1}\right)+HOT^R_{i+1/2}
\end{aligned}
$$

Compute the $HOT^L_{i+1/2}$ terms as in (17) with $g^-_j=\frac{1}{2}L^p(U^R)(F_j+B_j-\alpha_{i+1/2}U_j)$.

The extension just described complies with the basic design principles of Marquina's flux formula: the superscript L refers to characteristic information carried by a left-wind, while R refers to right-wind driven information. On the other hand, the superscripts ± in the local characteristic fluxes refer to the main (first-order) source term contribution in each characteristic field. As in the scalar case, the +-fluxes in the pth field collect source term contributions only if there is wind coming from the right at the interface. [1]

We also remark that when $U^L=U^R$ (for example when $U^L=U^R=U^A$ an average interface state), then the 2J-numerical flux function just described becomes a 1J numerical flux and (27) reduces to

$$
G^{\pm}_{i+1/2}=\sum_p(\tilde{G}^{p,\pm}_{i+1/2})^A R^p(U^A_{i+1/2}) \tag{28}
$$

This 1J-numerical flux becomes an extension of the ENO numerical fluxes in [22] to the shallow water system. As we shall see shortly, this extension turns out to be different from that in [25].

3.2.1 Numerical approximation of the source term contributions

For the shallow water system, the source term contribution involves the numerical approximation of integrals such as $\int_{x_i}^{x_{i+1}} ghz_x dx$. Assuming that the integrand is smooth and applying the trapezoidal rule

$$
\beta_{i,i+1}=\int_{x_i}^{x_{i+1}} ghz_x dx = g[(hz_x)_i+(hz_x)_{i+1}]\frac{\Delta x}{2}-(hz_x)_{xx}(\xi)\frac{\Delta x^3}{12} \tag{29}
$$

If the topography and the flow are smooth, we can write

$$
\begin{aligned}
(hz_x)_i = h_i(z_x)_i = h_i\left(\frac{z_{i+1}-z_i}{\Delta x}-\frac{\Delta x}{2}(z_{xx})_i+O(\Delta x^2)\right)\\
(hz_x)_{i+1} = h_{i+1}(z_x)_{i+1} = h_{i+1}\left(\frac{z_{i+1}-z_i}{\Delta x}+\frac{\Delta x}{2}(z_{xx})_{i+1}+O(\Delta x^2)\right)
\end{aligned}
$$

Replacing these terms in (29) we obtain

$$
\begin{aligned}
\beta_{i,i+1} &= \frac{g}{2}(z_{i+1}-z_i)(h_{i+1}+h_i)+\frac{\Delta x^2}{4}g(-h_i(z_{xx})_i+h_{i+1}(z_{xx})_{i+1})+O(\Delta x^3)\\
&= \frac{g}{2}(z_{i+1}-z_i)(h_{i+1}+h_i)+O(\Delta x^3)=\bar{\beta}_{i,i+1}+O(\Delta x^3)
\end{aligned}
$$

[1] We have also tested $(\tilde{G}^{p,\pm}_{i+1/2})^L = L^p(U^L)\cdot\frac{1}{2}\left(F_i+\alpha_{i+1/2}U_i\pm\frac{1}{2}B_{i,i+1}\right)+HOT^L_{i+1/2}$ and $(\tilde{G}^{p,\pm}_{i+1/2})^R = L^p(U^R)\cdot\frac{1}{2}\left(F_{i+1}-\alpha_{i+1/2}U_{i+1}\pm\frac{1}{2}B_{i,i+1}\right)+HOT^R_{i+1/2}$ without finding differences in the experimental results.

Thus, $\bar{\beta}_{i,i+1} = \frac{g}{2}(z_{i+1} - z_i)(h_{i+1} + h_i)$ provides a third order approximation to the integral of the source term between two consecutive cells, for smooth flows and smooth topography. This is compatible with the third order accurate flux computations carried out by the ENO-3 algorithm described above and is the choice used in our numerical schemes.

3.2.2 Study of the C-property

Numerical schemes specifically designed for the simulation of shallow water flows must be able to compute accurately steady states and small dynamical perturbations of these. Schemes that have this property are named *well balanced* by Le Roux and collaborators [13]. In [2], Bermúdez and Vázquez Cendón identify a key property that the scheme must satisfy in order to prevent the formation of numerically driven parasitic waves. This is the so called C property.

A scheme is said to satisfy the exact C-property if it is exact when applied to the stationary case $q \equiv 0$ and $h + z \equiv constant$ and if it is not exact but accurate to order $O(\Delta x^2)$ it is said to satisfy the approximate C-property.

To determine the behaviour of our 2J scheme with respect to the C-property, we study the flux difference $G^+_{i+1/2} - G^-_{i-1/2}$ for the quiescent stationary solution $q = 0, h + z = C$. Notice that in this case $\nabla(z + h) = 0$, thus

$$F(U) = \begin{pmatrix} 0 \\ \frac{1}{2}gh^2 \end{pmatrix} \quad B(U,x) = \begin{pmatrix} 0 \\ \int_{\bar{x}}^{x} ghz_x ds \end{pmatrix} = \begin{pmatrix} 0 \\ -\int_{\bar{x}}^{x} ghh_x ds \end{pmatrix}$$

hence

$$B_j = \begin{pmatrix} 0 \\ -\frac{g}{2}h_j^2 + constant \end{pmatrix} \quad G_j = F_j + B_j = \begin{pmatrix} 0 \\ constant \end{pmatrix}$$

and

$$B_{i,i+1} = B_{i+1} - B_i = \begin{pmatrix} 0 \\ -\frac{g}{2}(h_i^2 - h_{i+1}^2) \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{g}{2}(z_{i+1} - z_i)(h_{i+1} + h_i)) \end{pmatrix},$$

since $h_i + z_i = h_{i+1} + z_{i+1}$. This implies that that $\beta_{i,i+1} = \bar{\beta}_{i,i+1}$, i.e. the numerical approximation to the integral values are exact.

For quiescent steady state flows, the eigenvalues of the one dimensional system are: $\lambda_1(U) = -c$ and $\lambda_2(U) = c = \sqrt{gh}$, and the right and left eigenvectors $R^m(U) = (1, \lambda_m(U))^T$, $L^m(U) = \frac{1}{2}(1, \lambda_m^{-1}(U))$.

When $h > 0$, $\lambda_1 < 0$ and $\lambda_2 > 0$ always, and the 2J-algorithm easily leads to:

$$G^+_{i+1/2} - G^-_{i-1/2} = (\tilde{G}^{1,+}_{i+1/2})^R R^{1,R}_{i+1/2} + (\tilde{G}^{2,+}_{i+1/2})^L R^{2,L}_{i+1/2} - (\tilde{G}^{1,-}_{i-1/2})^R R^{1,R}_{i-1/2} - (\tilde{G}^{2,-}_{i-1/2})^L R^{2,L}_{i-1/2}$$

Since G_j is constant $\forall j$, all divided differences of the characteristic fluxes: $g_j^{p,L} = L^{p,L}G_j$ and $g_j^{p,R} = L^{p,R}G_j$ for $p = 1, 2$ are zero, hence all HOT terms are zero. In addition, the eigenvalues do not change sign, i.e. the flux functions are always computed with the ENO-Roe Algorithm and one can easily check that

$$(\tilde{G}^{1,+}_{i+1/2})^R = L^{1,R}_{i+1/2}(F_{i+1} + B_{i,i+1}) = -\frac{1}{4}g\frac{h_i^2}{c^R_{i+1/2}} \qquad (\tilde{G}^{2,+}_{i+1/2})^L = L^{2,L}_{i+1/2}F_i = \frac{1}{4}g\frac{h_i^2}{c^L_{i+1/2}}$$

$$(\tilde{G}^{2,-}_{i-1/2})^L = L^{2,L}_{i-1/2}(F_{i-1} - B_{i-1,i}) = \frac{1}{4}g\frac{h_i^2}{c^L_{i-1/2}} \qquad (\tilde{G}^{1,-}_{i-1/2})^R = L^{1,R}_{i-1/2}F_i = -\frac{1}{4}g\frac{h_i^2}{c^R_{i-1/2}}$$

Then, carrying out the algebra we get

$$\left(\frac{G^+_{i+1/2}-G^-_{i-1/2}}{\Delta x}\right)=\frac{gh_i^2}{4\Delta x}\begin{pmatrix}-\frac{1}{c^R_{i+1/2}}+\frac{1}{c^L_{i+1/2}}+\frac{1}{c^R_{i-1/2}}-\frac{1}{c^L_{i-1/2}}\\ 0\end{pmatrix} \tag{30}$$

We can observe that all terms in the second component cancel out. For the first component, assuming that h is a smooth function, the ENO interpolation procedure of order r would lead to the following estimate

$$h^{L,R}_{i+1/2}=h(x_{i+1/2})+O(\Delta x^r) \tag{31}$$

Hence, for $h>0$ and sufficiently smooth, we can conclude that

$$(h^{L,R}_{i+1/2})^{-1/2}=(h(x_{i+1/2}))^{-1/2}+O(\Delta x^r) \quad\Rightarrow\quad (c^{L,R}_{i+1/2})^{-1}=(c(x_{i+1/2}))^{-1}+O(\Delta x^r)$$

so that the approximate C-property is satisfied provided that $r\geq 3$.

Notice that for the first order scheme we have

$$\left(\frac{G^+_{i+1/2}-G^-_{i-1/2}}{\Delta x}\right)=\frac{gh_i^2}{4\Delta x}\begin{pmatrix}-\frac{1}{c_{i+1}}+\frac{1}{c_i}+\frac{1}{c_i}-\frac{1}{c_{i-1}}\\ 0\end{pmatrix}=\frac{gh_i^2}{4\Delta x}\begin{pmatrix}-\left(\frac{1}{c}\right)_{xx}\Delta x^2\\ 0\end{pmatrix}=\begin{pmatrix}O(\Delta x)\\ 0\end{pmatrix}$$

Thus, it seems reasonable to expect that second order accuracy would also ensure the approximate C-property, and this seems to be so computationally (see section 4), but we cannot ensure it analytically.

Notice that if $U^L=U^R$, i.e. we use only one Jacobian at each cell interface, the first term in (30) also cancels out, and that this cancellation occurs *independently of the interface state* and of the order of the scheme. Therefore, our 1J (one-Jacobian) scheme verifies the exact C-property. As mentioned in the introduction, the 1J scheme can be considered as a flux gradient-source term balanced extension of the ENO schemes in [22], different from that in [25].

Remark 1 *Our analysis shows that the cancellations necessary to obtain the C property must be accomplished at each cell boundary. The two essential ingredients at the $i+1/2$ interface are: a) only one Jacobian is computed at the interface and b) $\bar{\beta}_{i,i+1}=\beta_{i,i+1}$.*

3.2.3 Combined 1J-2J scheme

In light of the remarks above, It would seem that the use of two Jacobian evaluations at cell walls is not to be recommended for numerical simulations of steady or quasi-steady shallow water flows, unless the scheme is at least third order accurate. However, we have found that using two jacobians can still be advantageous in certain situations that are of interest in shallow water flow and that involve two very different states at both sides of a numerical interface (see experiments: Transcritical flow with shock in section 4.1.2 and the ones in sections 4.3.3 and 4.3.4)

We then propose a scheme that combines the use of one or two Jacobians, when either choice is most appropriate: When the interpolated states at each cell-wall, $U^L_{i+1/2}$ and $U^R_{i+1/2}$, are 'close', then, a single Jacobian decomposition is used at the $i+1/2$ interface. If there is a significant difference between the left and right states at the interface we construct the combined flux function using two Jacobian decompositions. However, if there is a sonic point nearby, in practice when $\lambda^p(U_i)\lambda^p(U_{i+1})\leq 0$, then even if the states are close, we still use the two sided decomposition. In this way, we expect to get both the benefits of *essentially* satisfying the exact C-property, by using mostly the 1J-numerical flux, and of the 2J scheme, at a few particular locations where it is known that Marquina's flux splitting technique has a better performance.

The combined 1J-2J numerical flux function we propose can be explicitly described as follows,

Algorithm 1J-2J : At the $i+1/2$ interface,

- Compute $U^L = U^L_{i+1/2}$ and $U^R = U^R_{i+1/2}$ with left and right-biased ENO interpolation, as in [8]
- Compute the eigenvalues $\lambda^p(U^L)$, $\lambda^p(U^R)$ for $p = 1, 2$.
- If $||U^L - U^R|| < \Delta x^s$ [2] and $\lambda^p(U^L)\lambda^p(U^R) > 0$ for $p = 1$ and 2 then (contiguous states are very close and no sonic point nearby: 1J-numerical flux)

  Define the average state at the interface, e.g. $U^A_{i+1/2} = \frac{1}{2}(U^L + U^R)$ [3]

$$G^{\pm}_{i+1/2} = \sum_p (\tilde{G}^{p,\pm}_{i+1/2})^A R^p(U^A_{i+1/2})$$

- Else, (contiguous states not close, or sonic nearby: 2J-numerical flux)

$$G^{\pm}_{i+1/2} = \sum_p (\tilde{G}^{p,\pm}_{i+1/2})^L R^p(U^L) + (\tilde{G}^{p,\pm}_{i+1/2})^R R^p(U^R)$$

3.3 Treatment of dry zones

One of the major problems that arises when simulating the movement of a fluid is the presence/occurrence of dry zones. Dry areas in shallow-water flow simulations can be handled in various ways, but Toro [24] warns that converting dry areas into *slightly wet* areas is both inappropriate and numerically 'dangerous' (see [24] for details).

Front tracking seems to be an adequate procedure but, since it is difficult to implement in several dimensions, other alternatives have also been explored in the literature (see [5, 4],[23, 26], [9] and references therein).

In this paper we follow the approach in [3]. To deal with existing dry areas, we give a threshold of minimum water depth below which h and q are set to zero and the cell is considered a dry cell. Wetting fronts are then included in the ordinary cell procedure in a through calculation that assumes zero water depth for the dry cells. The key point in this approach is the the use of a numerical flux function that can cope with zero-depth cells while maintaining, at the same time, the C-property.

Since the advancement of the wet/dry front should be determined by the wet state at the front, we adopt the following special treatment at cell-walls separating wet and dry states.

- If $h_i \neq 0$ and $h_{i+1} = 0$: (*wet/dry* front) then
  - Compute $U^L_{i+1/2}$ with ENO interpolation. Set $U^R_{i+1/2} = U^L_{i+1/2}$.
- If $h_i = 0$ and $h_{i+1} \neq 0$: (*dry/wet* front) then
  - Compute $U^R_{i+1/2}$ with ENO interpolation. Set $U^L_{i+1/2} = U^R_{i+1/2}$.

The interface flux always uses one Jacobian, evaluated at the wet state at the interface, hence the first condition to fulfill the exact C property is respected (see Remark 1).

[2] s is a power less than the order of the scheme r. In practice, we use $s = 1/2$ for $r = 1$ and $s = 1$ for $r = 2, 3$.

[3] in our numerical simulations we have found no difference between the arithmetic mean or the Roe mean.

3.3.1 Source term at wet/dry fronts

Notice that the discrete expression $\bar{\beta}_{i,i+1}$ in section 3.2.1 is not the exact value of the integral source term contribution $\beta_{i,i+1}$ at the boundary between a wet and a dry cell. In order to fulfill the second condition in Remark 1 we re-derive the discrete value of the integral contribution taking into account the existence of a dry cell.

We consider first a *wet/dry* interface (i.e. $h_i \neq 0$, $h_{i+1} = 0$). Let x_c be the the point where h becomes null. Then,

$$\beta_{i,i+1} = \int_{x_i}^{x_{i+1}} ghz_x dx = \int_{x_i}^{x_c} ghz_x dx = \frac{g}{2}(h_c + h_i)(z_c - z_i) + O(\Delta x^3) \tag{32}$$

where $h_c = h(x_c) = 0$ and $z_c = z(x_c)$. Let us assume that z and h are sufficiently smooth in $[x_i, x_c]$, then:

$$h_c = h_i + h_x(i)(x_c - x_i) + O(\Delta x^2) \tag{33}$$

$$z_c = z_i + z_x(i)(x_c - x_i) + O(\Delta x^2) \tag{34}$$

Assuming that $h_x(i) = O(1)$ we can write

$$z_c = z_i - z_x(i)\frac{h_i}{h_x(i)} + O(\Delta x^2) \tag{35}$$

Replacing z_c from (35) in (32), we obtain

$$\beta_{i,i+1} = -\frac{g}{2}\frac{z_x(i)}{h_x(i)}h_i^2 + O(\Delta x^2) \tag{36}$$

To obtain a discrete approximation, we substitute $z_x(i)$ and $h_x(i)$ by discrete estimations from 'wet' variables at cell centers, i.e.

$$\bar{\beta}_{i,i+1} = -\frac{g}{2}\frac{z_i - z_{i-1}}{h_i - h_{i-1}}h_i^2 \tag{37}$$

This approximation provides a discrete value for the source term contribution that satisfies the second condition in remark 1: If $h_i + z_i = h_{i-1} + z_{i-1}$ at the *wet* side, then $\bar{\beta}_{i,i+1} = \beta_{i,i+1} = -gh_i^2/2$. Hence the numerical flux function complies with the necessary conditions to obtain the cancellations necessary for the C-property.

For a *dry/wet* boundary, an analogous computation leads easily to

$$\beta_{i,i+1} = \frac{g}{2}\frac{z_x(i+1)}{h_x(i+1)}h_{i+1}^2 + O(\Delta x^2) \tag{38}$$

As before, approximating the derivatives from *wet* values leads to the following discrete source term contribution (that satisfies also the second condition in remark 1)

$$\bar{\beta}_{i,i+1} = \frac{g}{2}\frac{z_{i+2} - z_{i+1}}{h_{i+2} - h_{i+1}}h_{i+1}^2 \tag{39}$$

Note that in the redefinition of the integral (32) x_c is a point between x_i and x_{i+1} so the possible values of h_x in (33) are restricted to

$$h_x(i) \leq -\frac{h_i}{\Delta x} \text{ in the wet/dry front} \tag{40}$$

$$h_x(i+1) \geq \frac{h_{i+1}}{\Delta x} \text{ in the dry/wet front} \tag{41}$$

If inequality (40) (or (41)) does not hold, we assume that $x_c = x_i$ ($x_c = x_{i+1}$) and we use $\bar{\beta}_{i,i+1} = \frac{g}{2}(h_{i+1}+h_i)(z_{i+1}-z_i)$ as usually. Note also that inequalities (40) and (41) avoid automatically the particular cases $h_x(i) = 0$ and $h_x(i+1) = 0$.

3.3.2 Occurrence of Dry states from Wet flow

In section 2 we recall that dry areas appear in the solution of Riemann problems for shallow water flows with flat topography when the *depth positivity condition* $u^R - u^L \leq 2c^L + 2c^R$ is not satisfied. As noticed by Toro [24] the occurrence of dry areas is particularly hard to simulate with conventional *wet* Riemann solvers.

Throughout our numerical experimentation we have found that the type of viscosity ($\alpha_{i+1/2}$) considered in the LLF portion of Marquina's flux splitting formula is most important when dealing with the numerical formation of dry areas. Moreover, the way in which the interface states are computed seems to be also very important: U^L and U^R should be computed avoiding any mixed information. In numerical simulations involving dry states formed from wet states by rarefaction separation we have found a numerical treatment which seems to give consistent and reliable numerical results in all cases. The numerical treatment is as follows,

If $u_i - u_{i+1} > 2c_i + 2c_{i+1}$ holds,

- Compute U^L by an ENO interpolation *considering that the states at the right of the* $i + 1/2$ *interface are dry cells.*
- Compute U^R by an ENO interpolation *considering that the states to the left of the interface are dry cells.*
- For every p-th field the LLF construction is used substituting the numerical viscosity $\alpha_{i+1/2} = \max(|\lambda^p(U^L)|, |\lambda^p(U^R)|)$ by the following definition

$$\alpha_{i+1/2} := \frac{\hat{\lambda}^p_{i+1/2}(\lambda^p(U^R) + \lambda^p(U^L)) - 2\lambda^p(U^R)\lambda^p(U^L)}{\lambda^p(U^R) - \lambda^p(U^L)} \tag{42}$$

  where $\hat{\lambda}^p_{i+1/2}$ is the p-th eigenvalue of the Jacobian matrix evaluated at the Roe's average of contiguous states, i.e.

$$\hat{h}_{i+1/2} = \sqrt{h_i h_{i+1}}, \quad \hat{c}_{i+1/2} = \sqrt{g\frac{h_i + h_{i+1}}{2}} \quad \text{and} \quad \hat{u}_{i+1/2} = \frac{\sqrt{h_i}u_i + \sqrt{h_{i+1}}u_{i+1}}{\sqrt{h_i} + \sqrt{h_{i+1}}}$$

The choice for $\alpha_{i+1/2}$ is inspired in the entropy fix for Roe's scheme proposed by Harten and Hyman. For the scalar conservation law, the entropy-fixed flux can be written as ([17])

$$\begin{aligned} \tilde{F}(w^L, w^R) &= f^L + \lambda^L \frac{\lambda^R - \hat{\lambda}}{\lambda^R - \lambda^L}(w^R - w^L) \\ &= f^R - \lambda^R \frac{\hat{\lambda} - \lambda^L}{\lambda^R - \lambda^L}(w^R - w^L) \end{aligned} \tag{43}$$

where $\hat{\lambda}$ is a characteristic speed at the interface. Adding the two expressions for the RHS we get

$$\tilde{F}(w^L, w^R) = \frac{1}{2}(f^L + \alpha w^L) + \frac{1}{2}(f^R - \alpha w^R) \; ; \quad \alpha = \frac{\hat{\lambda}(\lambda^R + \lambda^L) - 2\lambda^R\lambda^L}{\lambda^R - \lambda^L}$$

For systems, $\hat{\lambda}^p$ is the pth eigenvalue of the Roe-matrix constructed from w^L and w^R. In figure 11(a) we show an example of the type of instabilities that occur when these corrections are not implemented.

4 Numerical Experiments

The following series of numerical experiments illustrate various features of the scheme. The test cases are standard in the literature. In all numerical simulations we use a CFL coefficient of 0.8 and a threshold of water depth equal to 10^{-4}.

4.1 Steady flow: Relevance of the C-property

4.1.1 Maintaining Steady flow

To give a numerical validation of our study on the C-property, we follow [25] and consider a smooth topography in a $20m$ channel given by

$$z(x) = 0.2e^{-(\frac{2}{5}(x-10)^2)}$$

and the two initial states shown in Figure 2. The left plot corresponds to a quiescent state ($u = 0$) and the right plot to a non-quiescent state with constant discharge 4.42 m^2/s. In table 1, we show measurements of the L^1-error and the numerical order at $T = 50s$ when using the 2J-scheme. We can see that the order of accuracy obtained in the quiescent simulation agrees with the theoretical analysis in section 3.2.2: the global error behaves like $O(\Delta x)$ for $r = 1$, and it seems to be of order larger than 2 for the third order scheme. It also shows that the second order method satisfies the approximate C-property although we could not ensure it theoretically. We obtain similar results for the non-quiescent case, as shown in table 1, even though no analytical study was presented for this case.

If we repeat the calculations using the 1J-2J scheme, then the error is within machine precision, i.e. the scheme satisfies the exact C-property, in the quiescent case. The table for the non-quiescent case deserves some explanation: We can observe that all the errors in this case are the same for any r. This $O(\Delta x^2)$ error actually comes from the trapezoidal rule used to approximate the integral source term contribution. We have verified experimentally that if we increase the order of the numerical integration, using for example Simpson's rule, the approximation order of the non-quiescent state is improved. We thus conjecture that the 1J-scheme preserves *all* steady states up to the order of the numerical integration procedure.

Usually, the bottom topography is not smooth. With the objective of evaluating the discretization of the source term in the presence of complex and possibly non-smooth geometry, the following experiment was proposed in a workshop on dam-break wave simulation [12]. The topography is tabulated in [12] and shown in figure 3. The initial condition is the water at rest at a level of 12m. The boundary conditions are a water level of 12m and no discharge. Numerical results obtained after a simulation of 200s are displayed in Figure 3. We can observe that the second order numerical scheme has not produced any visually detectable artificial movement of water. In table 3 we present the L^∞ errors in water depth and velocity after $T = 10.8s$ (600 nodes) with three different variants of the scheme.

4.1.2 Steady flow over a hump

This series of tests were proposed in the workshop [12] with the aim to evaluate the ability of numerical schemes to arrive and maintain the steady state over a non-flat topography. In these simulations a bottom topography of 25m length is defined as:

$$z(x) = \begin{cases} 0.2 - 0.005(x-10)^2 & \text{if} \quad 8\,m < x < 12\,m \\ 0 & \text{otherwise} \end{cases} \tag{44}$$

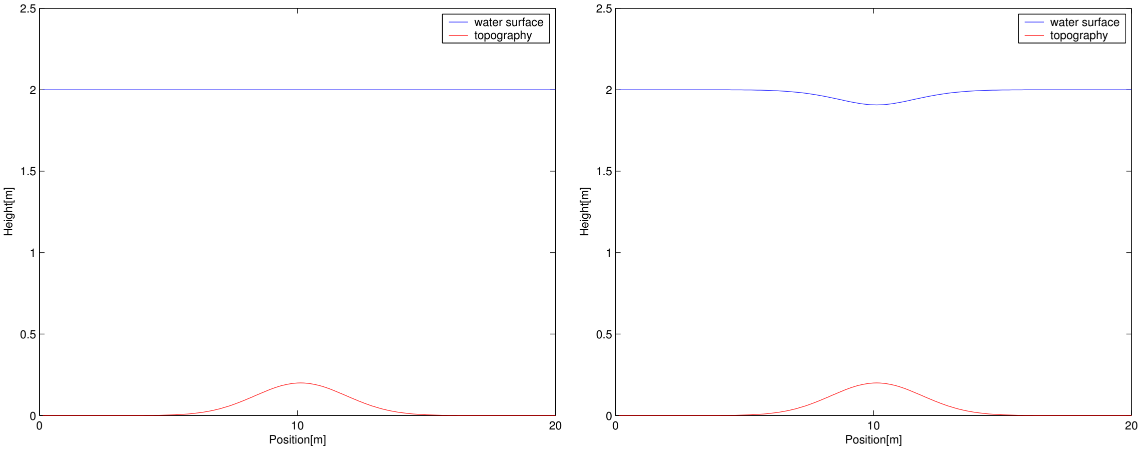


Figure 2: Exact water surface and topography at steady state (left: quiescent state, right: non-quiescent state).

QUIESCENT STATE

cells	n=1		n=2		n=3	
	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$
20		$1.9 \cdot 10^{-3}$		$5.47 \cdot 10^{-4}$		$2.62 \cdot 10^{-4}$
40	1.57	$6.48 \cdot 10^{-4}$	2.47	$9.84 \cdot 10^{-5}$	1.85	$7.24 \cdot 10^{-5}$
80	1.35	$2.53 \cdot 10^{-4}$	2.33	$1.95 \cdot 10^{-5}$	1.52	$2.52 \cdot 10^{-5}$
160	1.10	$1.17 \cdot 10^{-4}$	1.89	$5.26 \cdot 10^{-6}$	3.14	$2.84 \cdot 10^{-6}$
320	0.99	$5.95 \cdot 10^{-5}$	1.37	$2.02 \cdot 10^{-6}$	3.96	$1.81 \cdot 10^{-7}$

NON-QUIESCENT STATE

cells	n=1		n=2		n=3	
	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$
20		$1.7 \cdot 10^{-3}$		$7.26 \cdot 10^{-4}$		$2.92 \cdot 10^{-4}$
40	0.97	$8.46 \cdot 10^{-4}$	2.04	$1.76 \cdot 10^{-4}$	1.24	$1.23 \cdot 10^{-4}$
80	1.00	$4.22 \cdot 10^{-4}$	2.13	$4.00 \cdot 10^{-5}$	1.30	$5.01 \cdot 10^{-5}$
160	0.99	$2.11 \cdot 10^{-4}$	2.07	$9.5 \cdot 10^{-6}$	1.88	$1.36 \cdot 10^{-5}$
320	0.99	$1.06 \cdot 10^{-4}$	2.04	$2.30 \cdot 10^{-6}$	2.42	$2.53 \cdot 10^{-6}$

Table 1: Steady state with smooth topography. Error order measured at T=50s (2J-scheme).

QUIESCENT STATE

cells	n=1		n=2		n=3	
	$L^1 error$	$L^\infty error$	$L^1 error$	$L^\infty error$	$L^1 error$	$L^\infty error$
20	$2.22 \cdot 10^{-16}$	$6.66 \cdot 10^{-16}$	$3.44 \cdot 10^{-16}$	$4.44 \cdot 10^{-16}$	$6.10 \cdot 10^{-16}$	$8.88 \cdot 10^{-16}$
40	$1.28 \cdot 10^{-16}$	$4.44 \cdot 10^{-16}$	$7.22 \cdot 10^{-17}$	$2.22 \cdot 10^{-16}$	$3.77 \cdot 10^{-16}$	$8.88 \cdot 10^{-16}$
80	$1.23 \cdot 10^{-15}$	$1.78 \cdot 10^{-15}$	$1.64 \cdot 10^{-16}$	$4.44 \cdot 10^{-16}$	$1.03 \cdot 10^{-15}$	$1.55 \cdot 10^{-15}$
160	$1.57 \cdot 10^{-16}$	$6.66 \cdot 10^{-16}$	$6.41 \cdot 10^{-16}$	$1.33 \cdot 10^{-15}$	$1.44 \cdot 10^{-15}$	$2.00 \cdot 10^{-15}$
320	$5.76 \cdot 10^{-16}$	$1.11 \cdot 10^{-15}$	$2.89 \cdot 10^{-16}$	$8.88 \cdot 10^{-16}$	$1.66 \cdot 10^{-15}$	$2.22 \cdot 10^{-15}$

NON-QUIESCENT STATE

cells	n=1		n=2		n=3	
	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$	$L^1 order$	$L^1 error$
20		$1.15 \cdot 10^{-5}$		$1.15 \cdot 10^{-5}$		$1.15 \cdot 10^{-5}$
40	1.93	$3.00 \cdot 10^{-6}$	1.93	$3.00 \cdot 10^{-6}$	1.93	$3.00 \cdot 10^{-6}$
80	1.98	$7.59 \cdot 10^{-7}$	1.98	$7.59 \cdot 10^{-7}$	1.98	$7.59 \cdot 10^{-7}$
160	2.00	$1.90 \cdot 10^{-7}$	2.00	$1.90 \cdot 10^{-7}$	2.00	$1.90 \cdot 10^{-7}$
320	2.00	$4.76 \cdot 10^{-8}$	2.00	$4.76 \cdot 10^{-8}$	2.00	$4.76 \cdot 10^{-8}$

Table 2: Steady state with smooth topography. Error order measured at T=50s (1J-2J scheme).

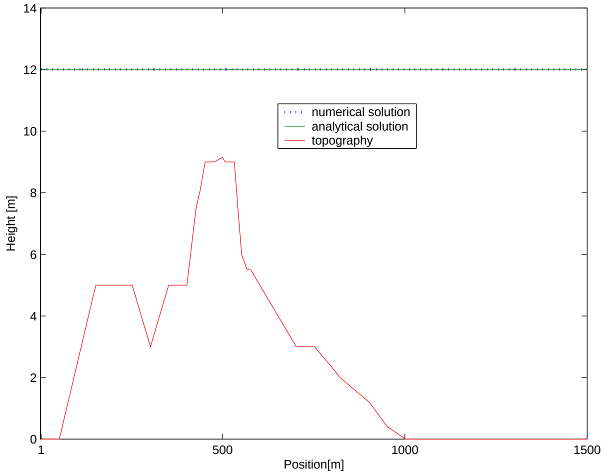


Figure 3: Water at rest over a complex geometry (T=200s, 300 nodes, 2nd order 2J-scheme)

Num. scheme	order	L^∞ error in h	L^∞ error in u
2J-scheme	r=1	$2.875 \cdot 10^{-1}$	2.0542
	r=2	$1.922 \cdot 10^{-1}$	$3.103 \cdot 10^{-1}$
	r=3	$4.870 \cdot 10^{-2}$	$4.030 \cdot 10^{-2}$
1J-scheme (Roe's average)	r=1	$3.553 \cdot 10^{-15}$	$3.780 \cdot 10^{-15}$
	r=2	$1.777 \cdot 10^{-15}$	$2.114 \cdot 10^{-15}$
	r=3	$3.553 \cdot 10^{-15}$	$2.534 \cdot 10^{-15}$
1J-2J scheme	r=1	$3.553 \cdot 10^{-15}$	$3.780 \cdot 10^{-15}$
	r=2	$1.777 \cdot 10^{-15}$	$2.556 \cdot 10^{-15}$
	r=3	$3.553 \cdot 10^{-15}$	$3.315 \cdot 10^{-15}$

Table 3: Quiescent state proposed in a workshop on dam-break wave simulation [12]. L^∞ error measured at T=10.8s.

In all cases, the initial conditions are $h + z = constant$ and $q = 0$. Depending on the initial and boundary conditions, the resulting flow could be at rest, subcritical or transcritical with or without shock. For these experiments we use a grid with 100 points.

Subcritical flow The initial conditions are $h + z = 2m$, $q = 0$. For the boundary conditions:

$$\begin{aligned} \text{downstream:} \quad h = & \quad 2\,m \\ \text{upstream:} \quad q = & \quad 4.42\,m^2/s \end{aligned}$$

In figure 4 we display the results obtained with the second order 2J scheme. No significant differences can be seen when using $r = 1$ or $r = 3$ or when using the 1J-2J scheme. We note that the oscillations observed in the discharge are of the order of $O(\Delta x^r)$.

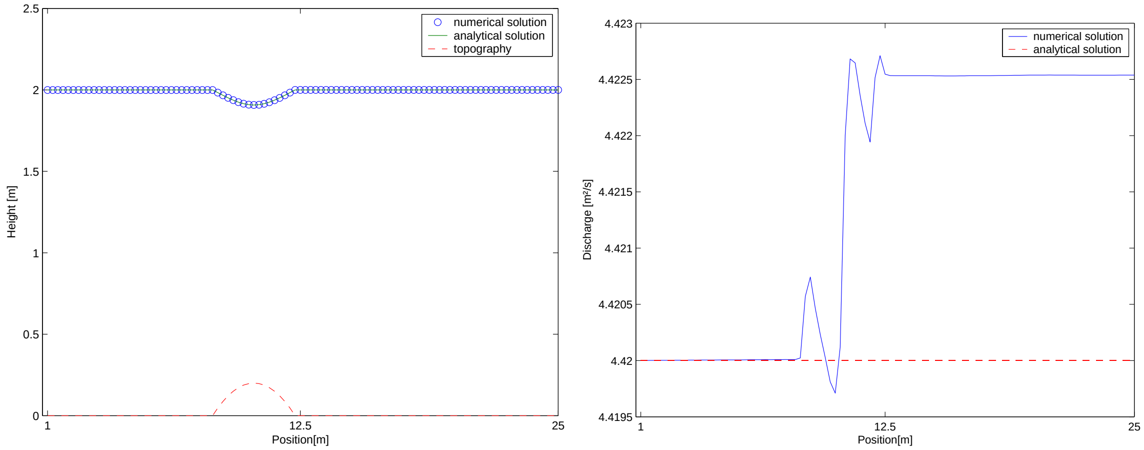


Figure 4: Subcritical flow over a hump (T=200s, 100 nodes, 2J-scheme $r = 2$). Water depth (left) and discharge (right)

Transcritical flow without shock Initial conditions: $h + z = .66m$, $q = 0$. Boundary conditions:

$$\begin{aligned} \text{downstream:} \quad h &= \quad 0.66\, m \text{ only when } F_r < 1 \\ \text{upstream:} \quad q &= \quad 1.53\, m^2/s \end{aligned}$$

where $F_r = u/\sqrt{gh}$ is the *Froude number.*

The results obtained after 200s of are shown in Figure 5. In the second order simulation shown in Figure 5 we appreciate a slight 'dog-leg'-like effect at the maximum of the topography, which is no longer visible in the 1J-2J simulation. The glitch is improved when using the 2J third order scheme which fits correctly the analytical solution (as the 1J-2J scheme).

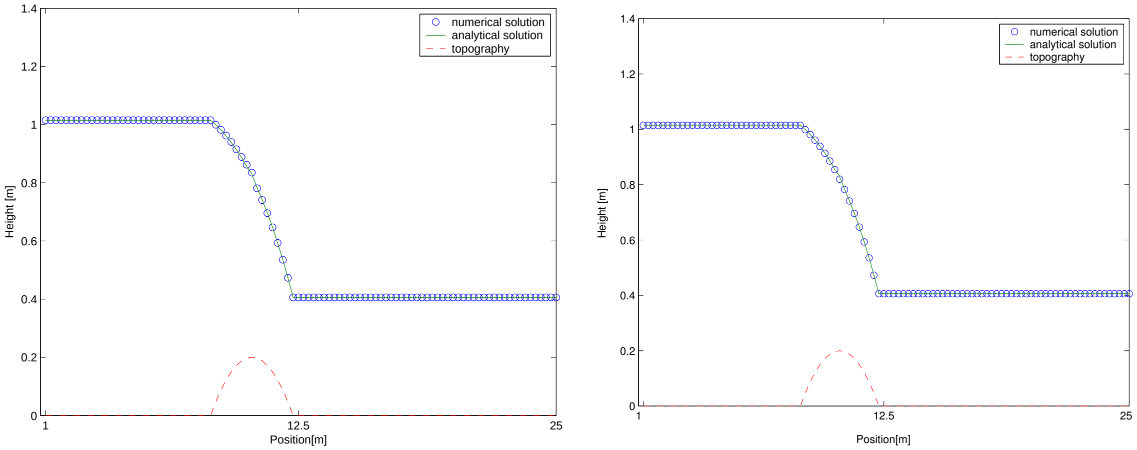


Figure 5: Transcritical flow without shock over a hump (T=200s, 100 nodes, 2nd order). Water depth: 2J-scheme (left) and 1J-2J scheme (right)

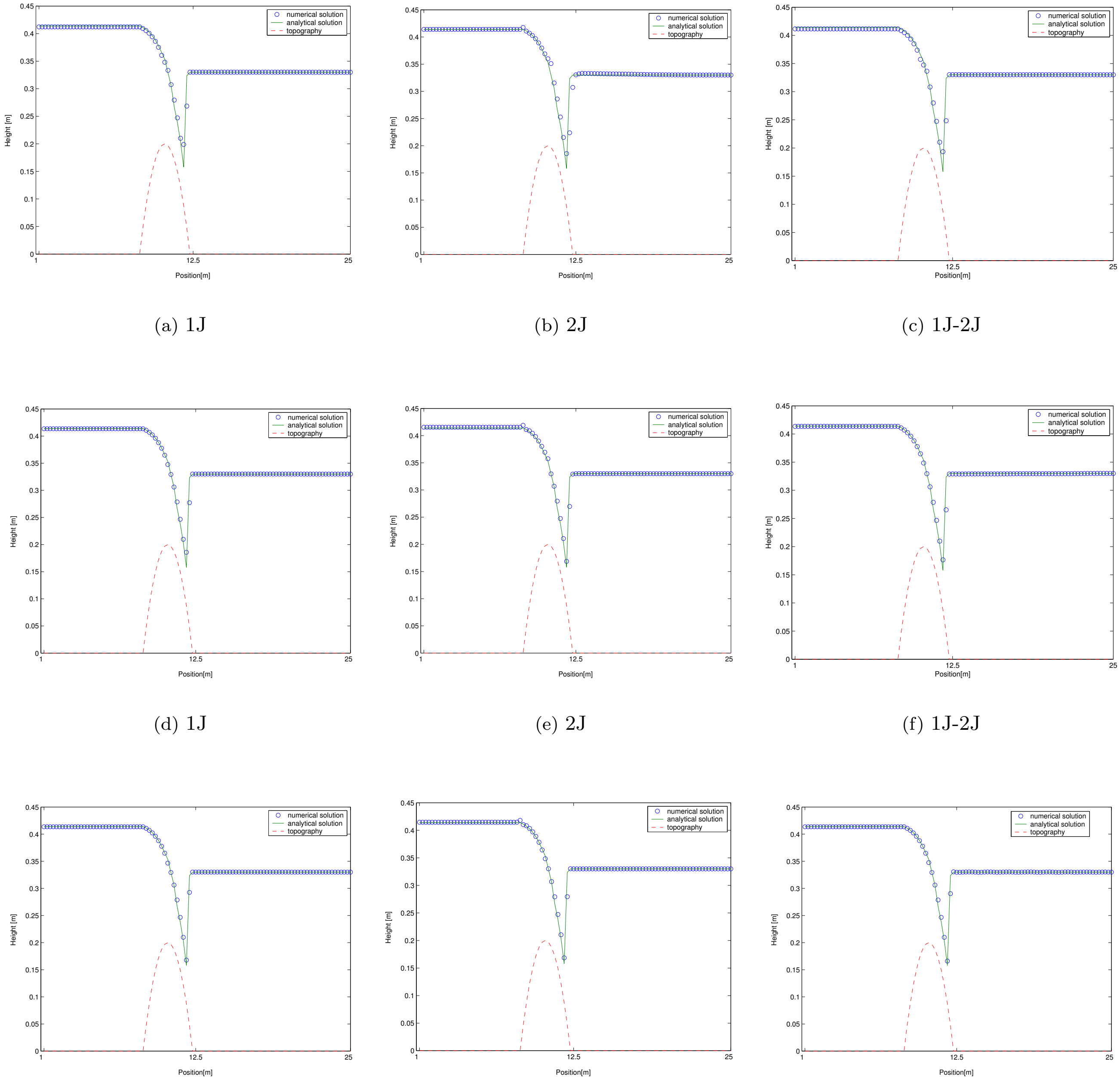
Transcritical flow with shock Initial conditions: $h + z = .33m$, $q = 0$. Boundary conditions:

$$\begin{aligned} \text{downstream:} \quad h &= \quad 0.33\, m \\ \text{upstream:} \quad q &= \quad 0.18\, m^2/s \end{aligned}$$

This test is particularly well suited to display the behavior of the schemes we propose. In Figure 6 we compare the results of the 1J-2J scheme with the 2J scheme for $r = 1, 2, 3$. We readily note that the 2J scheme gives a sharper resolution at the shock, however the glitches observed in the previous transcritical simulation distort in a noticeable manner the profile of the water surface in the first order simulation. The glitches improve with the order of accuracy, and the 1J-2J scheme produces the best resolution for each r.

4.2 Quasi stationary flow

In [19], Le Veque proposes a test involving a quasi-stationary flow in order to evaluate the capability of the scheme to accurately compute small perturbations of the water surface over a variable topography.



(a) 1J

(b) 2J

(c) 1J-2J

(d) 1J

(e) 2J

(f) 1J-2J

(g) 1J

(h) 2J

(i) 1J-2J

Figure 6: Transcritical flow with shock over a hump (T=200s, 100 nodes). Comparison of 1J and 2J scheme with the combined 1J-2J scheme. Top: 1st order. Middle: 2nd order. Bottom: 3rd order.

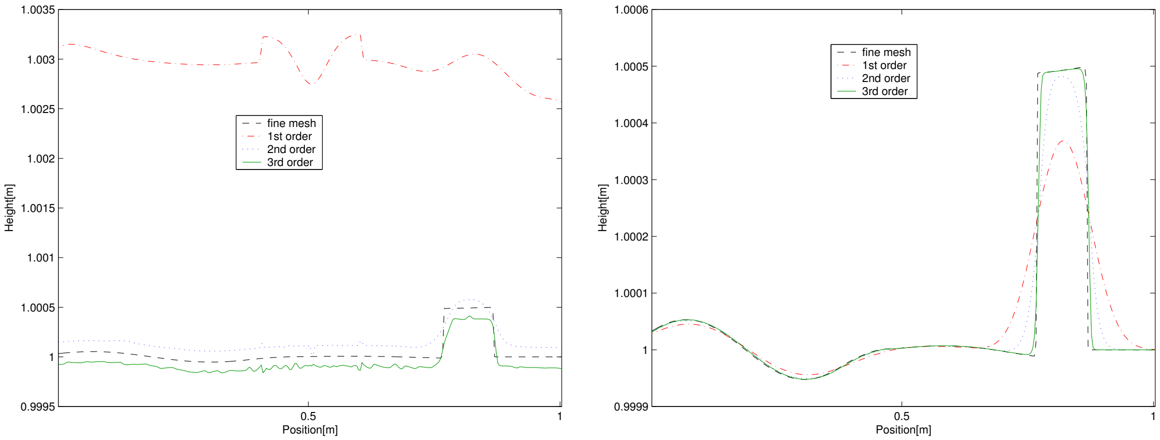
Le Veque uses this test to show the disadvantages of schemes that do not preserve steady states. The bottom topography is given as

$$z(x) = \begin{cases} 0.25(\cos(\pi(x-0.5)/0.1)+1) & \text{if} \quad |x-0.5| < 0.1 \\ 0 & \text{otherwise} \end{cases}$$

on $0 < x < 1$ with $g = 1$. The initial conditions are $q = 0$ and

$$h(x) := 1 - z(x) + \epsilon \text{ for } \quad 0.1 < x < 0.2,$$

which represents a small hump perturbation of the quiescent steady state $(h, u) = (1 - z, 0)$. The initial disturbance splits into two waves propagating with left and right characteristics speeds $\pm c$. A magnified view of the water surface after $0.7s$ with $\epsilon = 10^{-3}$, with 300 nodes is shown in Figure 7. The negative effect of not satisfying the exact or approximate C- property in the first order 2J scheme is revealed in spurious oscillations (of the order of the perturbation itself) over the topography (Fig. 7(a)). Mesh refinement (not shown) improves the results and the numerical solution converges to the true solution. The numerical results for the second and third order 2J scheme, which satisfy the *approximate* C-property, display a much smaller level of numerically generated noise. On the other hand, the results of the 1J-2J scheme are accurate and without spurious oscillations (Figures 7(b)).



(a) 2J scheme (300 nodes)

(b) 1J-2J scheme (300 nodes)

Figure 7: Quasi stationary case, water surface (T=0.7s with $\epsilon = 10^{-3}$)

4.3 Wet/dry fronts and dry bed generation

4.3.1 Drain on a non-flat bottom

This simulation was proposed in [9]. The bottom topography is the same as in (44) and the initial condition is $h + z = 0.5$, $q = 0$. At the right boundary, an outlet condition on a dry bed is simulated. The left boundary is a mirror state. The flow reaches a steady state with $h + z = .2$ and $q = 0$ to the left of the bump and $h = 0$ to its right.

In Figure 8 we show the evolution of the water surface as in [9]. Here we used the second order version of the 1J-2J scheme with 300 nodes. As noticed in [9], the non-preservation of discrete steady states leads to the wrong water height in this experiment. In our case, it is important to use the modification of the source term contribution specified in section 3.3.1. If $\bar{\beta}_{i,i+1} = g/2(z_{i+1} - z_i)(h_{i+1} + h_i)$ is used at wet/dry fronts, then the C property does not hold. In this case, the water level at the steady state in the numerical simulation (not shown) is below the exact value.

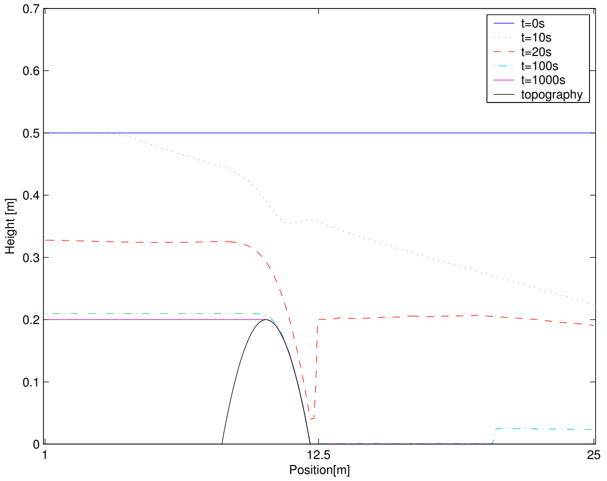


Figure 8: Drain on a non-flat bottom: water surface evolution (300 nodes, 2nd order 1J-2J scheme)

4.3.2 Oscillating lake

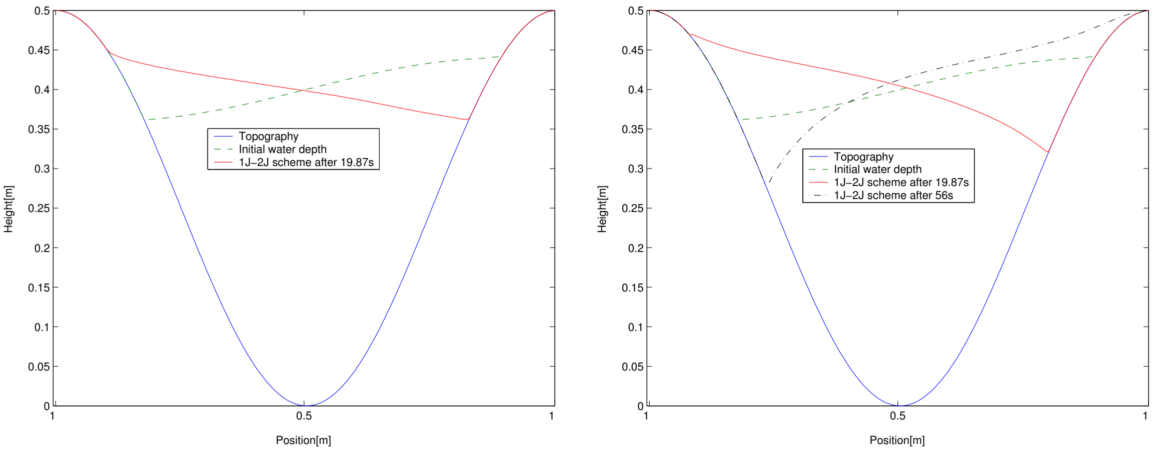
This test was proposed in [1]. The aim here is to asses the behaviour of the scheme in situations of wet/dry fronts over non-flat topographies. The topography simulates a lake bed with non-flat bottom and non-vertical shores,

$$z(x) = 0.5(1 - 0.5(cos(\pi(x - 0.5)/0.5) + 1))$$

The water surface of a lake at rest is perturbed initially with a small sinusoidal wave,

$$h(0, x) = \max(0, 0.4 - z(x) + 0.04(sin((x - 0.5)/0.25) + 0.04 \max(0, -0.4 + z(x))))$$

The flow oscillates in such a way that an interface between a wet cell and a dry cell has to be computed at each time step. The result after 19.87s with the 2nd order 1J-2J scheme can be seen in figure 9(a). The authors in [1] already pointed out the importance of using high order extensions of the scheme. In our case, the 1st order scheme does not damp the oscillations, instead we observe that the numerical solution leaves the domain in the course of time (see figure 9(b)). The 2nd and 3rd order versions of the scheme do maintain the periodic regime for all time.



(a) 2nd order

(b) 1st order

Figure 9: Oscillating lake (200 nodes, 2nd order 1J-2J scheme)

4.3.3 Dry bed generation by rarefaction separation

This is an experiment over flat topography ($z = 0$) proposed by Toro in [24]. The initial conditions

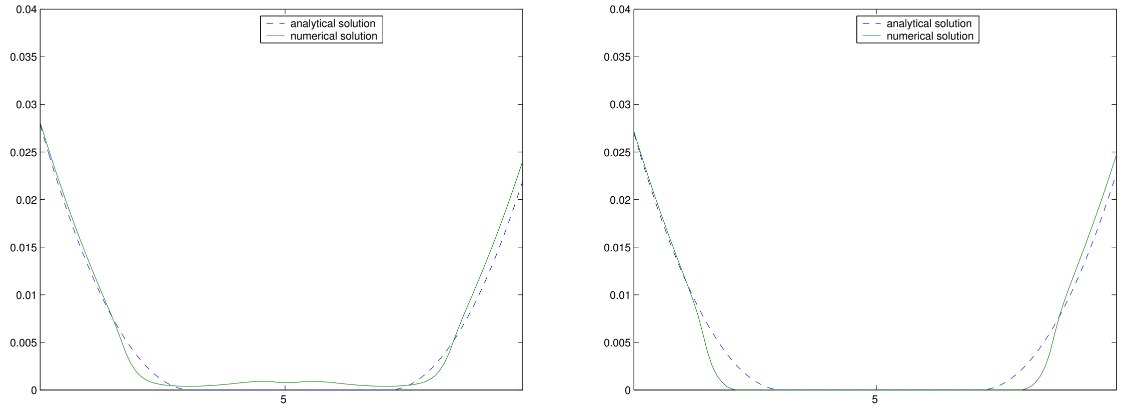
$$U(x,0) = \begin{cases} U_L = (h_L, q_L) = (0.1\,m, -0.3\,m^2/s) & \text{if} \quad x \leq 5\,m \\ U_R = (h_R, q_R) = (0.1\,m, 0.3\,m^2/s) & \text{if} \quad x > 5\,m \end{cases}$$

do not satisfy (4) at $x = 0$, hence a dry bed is formed instantaneously in the middle of a left going and right going rarefaction waves.

The generation of a dry state in between makes this test numerically difficult. In our context, it provides an example of a situation where the two-sided local characteristic decomposition does play a significant role. Our numerical experimentation shows that the 1J scheme cannot create a dry zone in the middle of the two rarefaction waves (with either of the two entropy fixes proposed, the regular LLF or H&H specified in (42))

As specified in section 3.3.2, two different modifications in the scheme are necessary in these situations: a) U^L and U^R should be computed avoiding any mixed information (considering an artificial dry state at the other side), b) compute the viscosity coefficient, $\alpha_{i+1/2}$, as in (42), i.e. using H&H entropy fix.

If we do not make any of these modifications in the 2J scheme (or the 1J-2J scheme), the dry zone is created but it is much larger than the analytical solution (simulation not shown). If we comply only with a) above and still use the viscosity coefficient in the regular LLF splitting (LLF-entropy fix), the dry bed is not created (see figure 10(a)). We can see in figure 10(b) that both modifications result in the generation of a dry zone (notice that it is still a little larger than the one in the analytical solution but not as large as the one obtained without any of the modifications). We also notice that when using the H&H entropy-fix (42) it is crucial to comply with a) above, otherwise instabilities such those in the next experiment (figure 11(a)) could be created.



(a) LLF entropy-fix only

(b) H&H entropy-fix and

Figure 10: Two rarefaction waves with dry bed generation (200 nodes, 2nd order, magnified view at center).

4.3.4 Dry bed generation with bottom topography

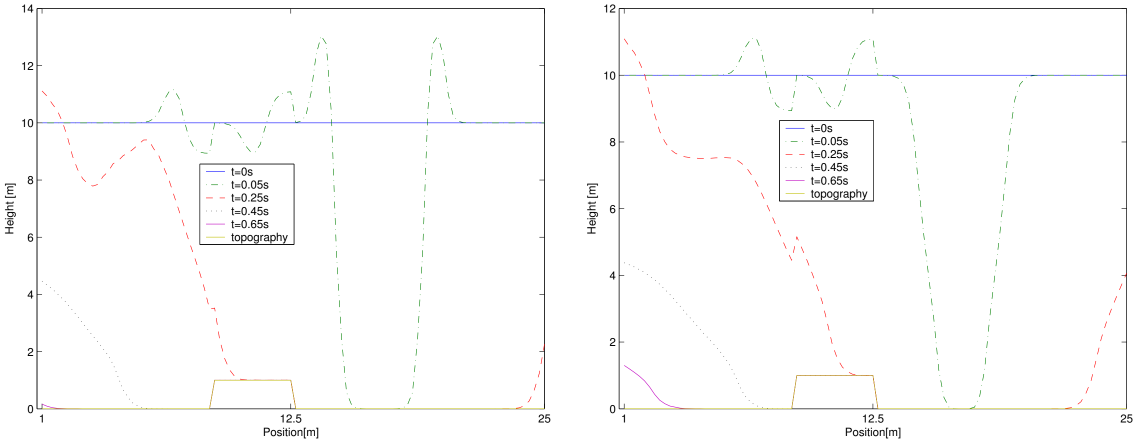
In [9], the basic test by Toro is modified by including a non-trivial topography, which is defined as

$$z(x) = \begin{cases} 1\,m & \text{if} \quad 25/3\,m < x < 12.5\,m \\ 0 & \text{otherwise} \end{cases}$$

with a length of 25 m. The initial conditions are $h + z = 10\,m$ and the discharge is -350 m^2/s if $x < 50/3$ m and 350 m^2/s otherwise.

As in the previous test, the initial data do not satisfy condition (4) so a dry bed, separating two rarefaction waves, is formed instantaneously over the flat portion to the right of the bump . The left-going wave interacts with the non-flat topography and as a consequence some waves are formed at the water surface.

The numerical results in figure 11 can be directly compared with those in [9]. Both results in figure 11 are computed switching to H&H entropy-fix when condition (4) is not satisfied. In Figure 11(a) we do not comply with condition a) i.e. allow mixing when computing U^L and U^R. In this case, instabilities are created when the left-going wave interacts with the bump in the topography. On the other hand, when complying with a), the result is more accurate (see figure 11(b)). In both simulations we have used the 1J-2J scheme (in this test, no noticeable differences with the 2J scheme can be observed). The same type of instabilities also occur when using the original scheme (without any of the modifications in section 3.3.2). If we only avoid mixing information in computing U^L and U^R at cell walls not satisfying (4), but always use the LLF entropy-fix the instabilities disappear but the dry bed still has not been created at $t = 0.05$s.



(a) Modified Harten-Hyman entropy-fix (with original ENO interpolation of states at interfaces)

(b) Modified Harten-Hyman entropy-fix (with modif. ENO interpolation of states at interfaces)

Figure 11: Dry bed generation by a double rarefaction wave over a step (100 nodes, 2nd order 2J scheme)

5 Conclusions

We have proposed an extension of Marquina's flux formula [7, 8] to the shallow water system with source terms coming from a non flat topography. The source term is included in a direct discretization of the system following a technique introduced in [10] by Gascón and Corberán.

We call our extension the 2J scheme, since it complies with the basic design principle in Marquina's flux formula: Two Jacobian evaluations are used at each interface in order to determine the characteristic information and the way in which this information is used in the numerical scheme. Since our 2J combined numerical flux does not verify the exact C-property, while a closely related 1J combined numerical flux does, we propose to use a 1J-2J combined numerical flux which *essentially* satisfies the C-property.

The results in this paper show a variety of situations where the *combined* 1J-2J scheme performs well. These situations involve, in addition to standard tests on shallow water flow over a hump in steady and quasi-steady situations, the generation of dry beds and flows over adverse slopes. The results indicate that the 1J-2J numerical flux provides a sufficiently simple numerical scheme that can be easily upgraded to obtain second and third order accuracy together with high resolution.

The scheme can be extended to 2D simulations in a dimension by dimension fashion and the addition of source terms coming from friction losses poses no special difficulties, since it is agreed that these terms do not need any special upwinding procedure. More realistic simulations in 2D is the subject of ongoing work.

Acknowledgements. The first author acknowledges partial support by the Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya. The first and third authors acknowledge partial support by PNPGC project, reference BFM2003-02125. The second author acknowledges partial support from the HYKE european network HPRN-CT-2002-00282. The third author acknowledges partial support by grant number FP2000-5801.

6 Some remarks on the conservativity of the scheme

In section 3.1 we have seen that in the scalar case we have

$$G_{i+1/2} - G_{i-1/2} = G^{+}_{i+1/2} - G^{-}_{i-1/2} \tag{45}$$

so the scheme continues being conservative if we use $G^{\pm}_{i+1/2}$ instead of $G_{i+1/2}$. To make the extension to nonlinear systems we simply apply, in each characteristic field, the same rules that we have found in the scalar case in order to compute the fluxes $G^{+}_{i+1/2}$ and $G^{-}_{i+1/2}$ that collect the source term contribution when the wind comes from the right and from the left respectively. We recall that in the system case the fluxes $G^{\pm}_{i+1/2}$ are computed from the contribution of the fluxes $(\tilde{G}^{\pm})^{p,R}$ and $(\tilde{G}^{\pm})^{p,L}$ in each characteristic field (see (27)). Thus, the construction of the algorithm makes possible that in the same interface $i+1/2$ the source term $B_{i,i+1}$ is projected using the left-biased eigenvectors in one characteristic field and using the right-biased eigenvectors in other field. As a consequence, (45) is not always verified and the error $\epsilon = (G_{i+1/2} - G_{i-1/2}) - (G^{+}_{i+1/2} - G^{-}_{i-1/2})$ that we make is

$$\epsilon = \sum_{p} \left(L^{p,S(p,x_{i+1/2})}_{i+1/2} \otimes R^{p,S(p,x_{i+1/2})}_{i+1/2} - L^{p,S(p,x_{i-1/2})}_{i-1/2} \otimes R^{p,S(p,x_{i-1/2})}_{i-1/2} \right) B_i \tag{46}$$

where $\otimes$ is the tensor product between vectors and $S(p, x_{i+1/2})$ is a discrete function with two possible values $\{L, R\}$. The function $S(p, x_{i+1/2})$ indicates which sided decomposition we use at interface $x_{i+1/2}$ and at p-th characteristic field. Indeed, if the conserved variables are sufficiently smooth, after (46) the error is $O(\Delta x^r)$ which is of the same order that the error order of the scheme. Thus, in smooth solutions, the extension to systems is 'conservative' (except for an error $O(\Delta x^r)$). On the other hand, if we use only one Jacobian, the error (46) is zero, so the scheme is always conservative. Moreover, the error (46) is only due to the first order terms of the source term contribution.

Let us describe an extension of the first order terms to systems which is conservative. The technique that we follow is basically the same, except that we write $B_i = \sum_{j=0}^{i-1} B_{j,j+1}$ and we divide $B_{i,i+1}$ into two pieces each one projected with its corresponding local-sided eigenvectors, ie,

$$(\tilde{G}^{p}_{i+1/2})^{L} R^{p}(U^{L}) = \begin{cases} L^{p}(U^{L}) F_i R^{p}(U^{L}) + \{i+\frac{1}{2}\}_p & \text{if } \lambda^p(U^L), \lambda^p(U^R) > 0 \\ 0 & \text{if } \lambda^p(U^L), \lambda^p(U^R) < 0 \\ \frac{1}{2} L^{p}(U^{L})(F_i + \alpha_{i+1/2} U_i) R^{p}(U^{L}) + \frac{1}{2}\{i+\frac{1}{2}\}_p & \text{else} \end{cases}$$

$$(\tilde{G}^{p}_{i+1/2})^{R} R^{p}(U^{R}) = \begin{cases} 0 & \text{if } \lambda^p(U^L), \lambda^p(U^R) > 0 \\ L^{p}(U^{R}) F_{i+1} R^{p}(U^{R}) + \{i+\frac{1}{2}\}_p + \{i+\frac{1}{2}, i+1\}_p & \text{if } \lambda^p(U^L), \lambda^p(U^R) < 0 \\ \frac{1}{2} L^{p}(U^{R})(F_{i+1} - \alpha_{i+1/2} U_{i+1}) R^{p}(U^{R}) + \frac{1}{2}\{i+\frac{1}{2}\}_p + \frac{1}{2}\{i+\frac{1}{2}, i+1\}_p & \text{else} \end{cases}$$

$$\text{where} \quad \begin{array}{ll} \{i+\frac{1}{2}\}_p & := \sum_{j=0}^{i} \left(L^{p}(U^{R}_{j-1/2}) B_{j-1/2,j} R^{p}(U^{R}_{j-1/2}) + L^{p}(U^{L}_{j+1/2}) B_{j,j+1/2} R^{p}(U^{L}_{j+1/2}) \right) \\ \{i+\frac{1}{2}, i+1\}_p & := L^{p}(U^{R}) B_{i+1/2,i+1} R^{p}(U^{R}) \end{array}$$

In practice, we take $B_{i,i+1/2} = B_{i+1/2,i+1} = \frac{1}{2} B_{i,i+1}$ in the experiments. We have then,

$$G_{i+1/2} = \sum_{p} (\tilde{G}^{p}_{i+1/2})^{L} R^{p}(U^{L}) + (\tilde{G}^{p}_{i+1/2})^{R} R^{p}(U^{R}) = \sum_{p} G^{p}_{i+1/2}$$

We define the fluxes that collect the source term contribution in the p-th characteristic field following the wind direction as,

$$G^{p,+}_{i+1/2} = G^{p}_{i+1/2} - \{i\}_p$$

$$G^{p,-}_{i+1/2} = G^{p}_{i+1/2} - \{i+1\}_p$$

where $\{i+1\}_p := \{i+\frac{1}{2}\}_p + \{i+\frac{1}{2}, i+1\}_p$. Thus, $G^{p,-}_{i-1/2} = G^{p}_{i-1/2} - \{i\}_p$ and we have the equality,

$$G^{+}_{i+1/2} - G^{-}_{i-1/2} = G_{i+1/2} - G_{i-1/2} \qquad \text{where } G^{\pm}_{i+1/2} = \sum_p G^{p,\pm}_{i+1/2}$$

We can see that this extension to systems is conservative even if we use two Jacobians. The final expression of fluxes $G^{p,+}_{i+1/2}$ and $G^{p,-}_{i+1/2}$ is

$$G^{p,+}_{i+1/2} = \begin{cases} L^p(U^L)F_iR^p(U^L) & \text{if } \lambda^p(U^L), \lambda^p(U^R) > 0 \\ L^p(U^R)F_{i+1}R^p(U^R) + L^p(U^L)B_{i,i+1/2}R^p(U^L) + L^p(U^R)B_{i+1/2,i+1}R^p(U^R) & \text{if } \lambda^p(U^L), \lambda^p(U^R) < 0 \\ \begin{array}{l} \frac{1}{2}L^p(U^L)(F_i + \alpha_{i+1/2}U_i)R^p(U^L) + \frac{1}{2}L^p(U^R)(F_{i+1} - \alpha_{i+1/2}U_{i+1})R^p(U^R) \\ +\frac{1}{2}L^p(U^L)B_{i,i+1/2}R^p(U^L) + \frac{1}{2}L^p(U^R)B_{i+1/2,i+1}R^p(U^R) \end{array} & \text{else} \end{cases}$$

$$G^{p,-}_{i+1/2} = \begin{cases} L^p(U^L)F_iR^p(U^L) - L^p(U^L)B_{i,i+1/2}R^p(U^L) - L^p(U^R)B_{i+1/2,i+1}R^p(U^R) & \text{if } \lambda^p(U^L), \lambda^p(U^R) > 0 \\ L^p(U^R)F_{i+1}R^p(U^R) & \text{if } \lambda^p(U^L), \lambda^p(U^R) < 0 \\ \begin{array}{l} \frac{1}{2}L^p(U^L)(F_i + \alpha_{i+1/2}U_i)R^p(U^L) + \frac{1}{2}L^p(U^R)(F_{i+1} - \alpha_{i+1/2}U_{i+1})R^p(U^R) \\ -\frac{1}{2}L^p(U^L)B_{i,i+1/2}R^p(U^L) - \frac{1}{2}L^p(U^R)B_{i+1/2,i+1}R^p(U^R) \end{array} & \text{else} \end{cases}$$

We note that this new scheme has the same construction that the one in section 3.2 with the difference that now we use $L^p(U^L)B_{i,i+1/2}R^p(U^L) + L^p(U^R)B_{i+1/2,i+1}R^p(U^R$ instead of $L^{p,S(p,x_{i+1/2})}_{i+1/2}B_{i,i+1}R^{p,S(p,x_{i+1/2})}_{i+1/2}$. As we have already said, these expressions ($G^{p,+}_{i+1/2}$ and $G^{p,-}_{i+1/2}$) correspond to the first order terms of the flux $G_{i+1/2}$ (thus making an abuse of notation). The HOT can be obtained as before or in the case of the source term, directly computing the high order extension of the new expression of the source term discretization.

Observe that interpreting $U^L = U^R$ the above notation encompasses the case of one Jacobian at each interface.

With this discretization of the source term and using (26) we have:

$$\sum_{j=0}^{N}(U_j)_t + \sum_{j=0}^{N}\frac{G^{+}_{j+1/2} - G^{-}_{j-1/2}}{\Delta x} = 0$$

Since

$$\sum_{j=0}^{N}\frac{G^{+}_{j+1/2} - G^{-}_{j-1/2}}{\Delta x} = \sum_{j=0}^{N-1}\frac{G^{+}_{j+1/2} - G^{-}_{j+1/2}}{\Delta x}$$

(where we assume that the boundary conditions are such that $G^{-}_{-1/2} = G^{+}_{N+1/2} = 0$) and

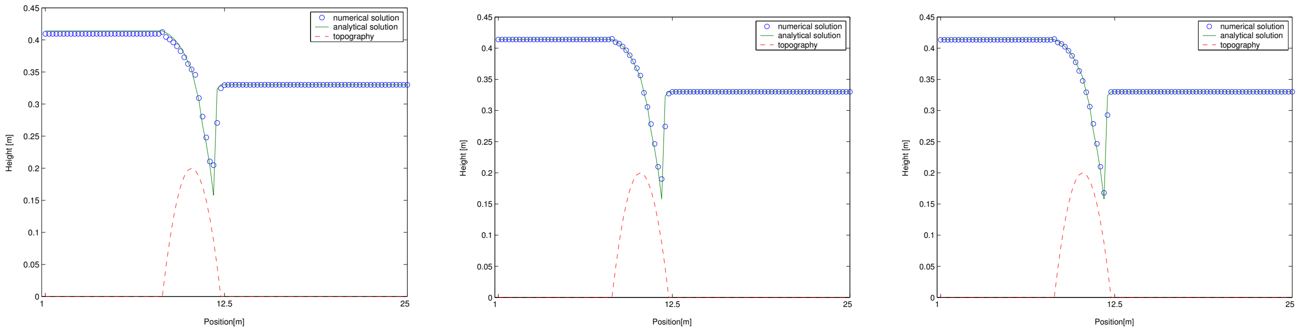
$$G^{+}_{j+1/2} - G^{-}_{j+1/2} = B_{j,j+1}$$

we obtain that

$$\Delta x \left(\sum_{j=0}^{N} U_j\right)_t + \sum_{j=0}^{N-1} B_{j,j+1} = 0$$

(we would add $G^{+}_{N+1/2} - G^{-}_{-1/2}$ in the right hand side of the above equation to include general boundary conditions).

We show in figure 12 the results obtained with this new discretization of the source term in the steady case corresponding to a transcritical flow with shock over a hump. We can compare the results with those in figure 6 which correspond to the scheme explained in section 3.2. We can observe that these results are quite similar (do not improve nor worsen them) to those of figure 6 so we choose the discretization based on integrals over cells which is more easily computable.



(a) 1st order

(b) 2nd order

(c) 3rd order

Figure 12: Transcritical flow with shock over a hump (T=200s, 100 nodes). New first order source term contribution based on integrals over semi-cells.

References

[1] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic projection for shallow water flows. *SIAM J. Sci. Comput.*, —:—, 2003.

[2] A. Bermúdez and Vázquez M. E. Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids*, 23(8):1049–1071, 1994.

[3] P. Brufau, Vázquez M. E., and P. García-Navarro. A numerical model for the flooding and drying of irregular domains. *International Journal for Numerical Methods in Fluids*, 39:247–275, 2002.

[4] P. Brufau and P. García-Navarro. Unsteady free surface flow simulation over complex topography with a multidimensional upwind technique. *Journal of Computational Physics*, 186:503–526, 2003.

[5] J. Burguete, P. García-Navarro, and R. Aliod. Numerical simulation of runoff extreme rainfall events in a mountain water catchment. *Natural Hazards and Earth System Sciences*, 2:109–117, 2002.

[6] A. I. Delis and T. Katsaounis. Relaxation schemes for the shallow water equations. *International Journal for Numerical Methods in Fluids*, 41:695–719, 2003.

[7] R. Donat and A. Marquina. Capturing Shock Reflections: An Improved Flux Formula. *Journal of Computational Physics*, 125:42–58, 1996.

[8] R. P. Fedkiw, B. Merriman, R. Donat, and S. Osher. The Penultimate Scheme for Systems of Conservation Laws: Finite Difference ENO with Marquina's Flux Splitting. *Progress in Numerical Solutions of Partial Differential Equations*, Arcachon, France, edited by M. Hafez, July 1998.

[9] T. Gallouët, J. M. Hérard, and N. Seguin. Some approximate Godunov schemes to compute shallow-water equations with topography. *Computers and Fluids*, 32:479–513, 2003.

[10] Ll. Gascón and J. M. Corberán. Construction of Second-Order TVD schemes for nonhomogeneous hyperbolic conservation laws. *Journal of Computational Physics*, 172:261–297, 2001.

[11] L. Gosse. A well-balanced flux-vector splitting scheme designed for hyperbolic systems of conservation laws with source terms. *Int. Journal of Computers and Mathematics*, 39:135–159, 2000.

[12] N. Goutal and F. Maurel. In *Proceedings of the 2nd Workshop on Dam-Break Wave Simulation*. EDF-DER Report HE-43/97/016/B, 1997.

[13] J. M. Greenberg and A. Y. LeRoux. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM J. Numerical Analysis*, 33:1–16, 1996.

[14] A. Harten, B. Engquist, S. Osher, and S.R. Chakracarthy. Uniformly high-order accurate non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

[15] A. Harten and S. Osher. Uniformly high-order accurate non-oscillatory schemes, I. *SIAM J. Numerical Analysis*, 24(2):279–309, 1987.

[16] G. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202, 1996.

[17] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser Verlag, 1992.

[18] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[19] R.J. LeVeque. Balancing source terms and flux gradients in high resolution Godunov methods. *Journal of Computational Physics*, 146:346, 1998.

[20] A. Marquina and P. Mulet. A flux-split algorithm applied to conservative models for multicomponent compressible flows. *Journal of Computational Physics*, 185(1):120–138, 2003.

[21] P. L. Roe. Upwind differencing schemes for hyperbolic conservation laws with source terms. In *Proceedings of Nonlinear Hyperbolic Problems, edited by C. Carasso, P. Raviart, and D. Serre, Lecture Notes in Mathematics, Springer-Verlag*, volume 1270, pages 41–51, 1986.

[22] C. W. Shu and S. Osher. Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes II (two). *Journal of Computational Physics*, 83:32–78, 1989.

[23] P. A. Sleigh, P. H. Gaskell, M. Berzins, and N. G. Wright. An unstructured finite-volume algorithm for predicting flow in rivers and estuaries. *Computers and Fluids*, 27(4):479–508, 1998.

[24] E. F. Toro. *Shock-Capturing Methods for Free-Surface Shallow Flows*. John Wiley & Sons, Ltd, 2001.

[25] S. Vukovic and L. Sopta. ENO and WENO Schemes with the Exact Conservation Property for One-Dimensional Shallow Water Equations. *Journal of Computational Physics*, 179:593–621, 2002.

[26] D. H. Zhao, H. W. Shen, G. Q. Tabios, J. S. Lai, and W. Y. Tan. Finite-volume two-dimensional unsteady-flow model for river basins. *Journal of Hydraulic Engineering, ASCE7*, 120:863–883, 1994.