Adaptive mesh refinement method. Part 1: Automatic thresholding based on a distribution function

Kévin Pons, Mehmet Ersoy

Adaptive mesh refinement method.
Part 1: Automatic thresholding based on a distribution function.

Kévin Pons and Mehmet Ersoy

Abstract The accurate numerical simulation of large scale flows, together with the detailed modeling of flooding or drying of small-scale regions, is a difficult and a challenging problem. Adaptive mesh method allows, in principle, to solve accurately those scales. However in practice, on one hand, the lack of *a priori* or efficient *a posteriori* error estimates, especially for multidimensional hyperbolic problems, make the analysis harder. On the other hand, once a mesh refinement criterion is chosen, the difficult problem is to determine the mesh refinement threshold parameter which is certainly the most important part of the adaptive process. The smaller this parameter is, the higher the number of cells refined is at the expense of the computational cost. In this work, we numerically investigate different refinement criteria and we present a general procedure to determine automatically a mesh refinement threshold for any given mesh refinement criterion. To this end the decreasing rearrangement (distribution) function of the mesh refinement criterion is introduced to catch relevant scales. The efficiency of the automatic thresholding method is illustrated through the one dimensional Saint-Venant system. Multidimensional and real life applications such as Tsunamis propagations are dealt in the second part.

1 Introduction

Depending on the complexity of the equations solved, the flow may develop many length scales such as shocks, steep gradients, large and small oscillations and the

Kévin Pons
Principia S.A.S., Zone Athélia 1, 215 voie Ariane, 13705 La Ciotat cedex, France, e-mail: Kevin.Pons@principia.fr,
Université de Toulon, IMATH EA 2134, 83957 La Garde, France, e-mail: Kevin.Pons@univ-tln.fr

Mehmet Ersoy
Université de Toulon, IMATH EA 2134, 83957 La Garde, France, e-mail: Mehmet.Ersoy@univ-tln.fr

relative importance of flow features becomes difficult to detect. The computation on uniform mesh being too expensive, adaptive mesh is certainly the most relevant approach to solve the problem. As a non-exhaustive list of references, one can refer to [5, 4, 39, 15, 22, 12] and also [18, 25, 6, 33, 36].

The first category of methods concerns the so-called Adaptive Mesh Refinement (AMR) technique. A mesh refinement criterion S, error indicator or a flow feature detector, is applied to all cells of levels $l \leqslant l_{\max}$ where $l_{\max}$ is the maximum number of refinement levels allowed. A cell C_k will be refined or coarsened if $S(x) > \beta$ or $S(x) < \beta$ where β is a given threshold parameter. This procedure which consists to split or to coarse a cell can be regarded as a particular adaptive moving mesh method. In the classical AMR approach, the maximum level of mesh refinement level $l_{\max}$ is fixed by the user and it mostly yields to dyadic, quadtree, or octree meshing according to the dimension of the problem. In the standard adaptive moving mesh strategy, the number of cells during the time evolution process is kept constant and the involved cells size are different.

In both cases, the adaptive method is driven by a ”suitable” indicator. The indicators based on the flow gradient are popular and often used due to its ”low-cost” and simplicity of numerical implementation. It can be based on the mass, the momentum or on a multi-criteria (a linear or a non linear combination of mass-momentum or other equation involved in the system). Unlike flow gradient based methods, the indicators based on error estimates are in general either not available or too expensive to compute in practice. The form of the error estimate can be *a priori* or *a posteriori*.

The *a priori* methods provide a theoretical bound of the discretization error of the form

$$\varepsilon_{\delta x} \leqslant C(\mathbf{w})\delta x^p$$

for some constant $C = C(\mathbf{w})$ depending on the unknown $\mathbf{w}$, where p is the order of discretization and δx is the mesh size. This approach does not require to know the numerical solution contrary to the *a posteriori* error methods. The computation of *a posteriori* error estimates is mainly based on the analysis of the numerical solution in terms of the truncation error using a Taylor expansion. The most popular method is the Richardson extrapolation. Such estimates can also be obtained properly by analyzing the numerical dissipation [19] or the flux consistency error as performed in this paper.

If the mesh refinement criterion is ”accurate” enough and if the numerical scheme under considerations is stable, then the yielding numerical errors remain homogeneous to the local error when the mesh is refined. To define how much the numerical solution have to be accurate we define a threshold parameter which can be regarded as the local accuracy ε (as much as the criterion is close to the local error). However, as said before finding such a mesh refinement criterion is in practice difficult. In most cases, unlike the error, the mesh refinement indicator can exhibit some oscillations at small scales yielding to oscillating numerical solutions if the threshold parameter is not small enough. On the contrary, if it is set too small, too many cells are unnecessarily refined. Therefore, for almost all adaptive methods a ”tunable” threshold parameter is unavoidable and should be calibrated to balance the accuracy of the solution and the computational cost. Indeed, if this tunable parameter is fixed ”correctly”, the error indicator will determine which elements require refinement and those in which refinement may not be necessary. As a straightforward consequence, the resolution is placed only in the regions in which it is needed and this procedure will be considerably less expensive than a simulation on uniform mesh.

The definition of the tunable threshold parameter is case-dependent and therefore cannot be apply blindly to any problems. Therefore, choosing a suitable threshold parameter is certainly the most difficult step in the adaptive process. To select automatically the threshold parameter, we propose a new method based on the decreasing rearrangement of the mesh refinement criterion.

The paper is organized as follows. Section 2 is devoted to the general presentation of the finite volume scheme and the AMR method. In Sect. 3, we present and numerically investigate several mesh refinement criteria through a Riemann test case for the one dimensional Saint-Venant system. The exact solution being known, we study the behavior of several mesh refinement criteria with respect to the number of mesh points that we confront to the local error. Then, we numerically compute the solution for several mesh refinement threshold and we dress the table of convergence. In particular, we highlight that even if the mesh refinement criterion is not of error type, it can drive the adaptive process efficiently if the threshold parameter is chosen appropriately. Finally in Sect. 4, after a review of thresholding methods, we propose an automatic thresholding method based on an analytical distribution function which takes into account all the features of any given mesh refinement criterion. In particular, we show that a simple mesh refinement criterion based on the flow gradient with automatic thresholding can be comparable to a more complex efficient local error estimate (here the local error itself).

2 Finite volume approximation and AMR technique

This section summarizes the main features of the finite volume approximation and the adaptive mesh refinement method for a general non-linear hyperbolic equation

$$\partial_t \mathbf{w} + \partial_x \mathbf{f}(\mathbf{w}) = 0 \tag{1}$$

where $\mathbf{w}(x,t) \in \mathbb{R}^d$, $\mathbf{f}(\mathbf{w}(x,t))$ stand respectively for the unknown vector and the multidimensional flux at a space-time point (x,t), $x \in \mathbb{R}^d$ is the coordinate, $t > 0$ is the time and d is the dimension of the problem.

2.1 First order finite volume approximation

We suppose that the computational domain $\Omega \subset \mathbb{R}^d$ is split into a set of control volumes, also referred as cells, $\Omega = \cup_k C_k$ of mesh size $\delta x_k = |C_k|$. We define the discrete times by

$$t_{n+1} = t_n + \delta t_n$$

where the time step δt_n satisfies a Courant, Friedrichs, Levy (CFL) condition.

On a given cell C_k of center x_k, noting

$$\boldsymbol{w}_k^n \simeq \frac{1}{\delta x_k} \int_{C_k} \boldsymbol{w}(x, t_n)\, dx$$

the approximation of the mean value of the unknown $\boldsymbol{w}(x,t)$ on C_k at time t_n, and integrating (1) over $C_k \times (t_n, t_{n+1})$, we obtain:

$$\int_{C_k} \mathbf{w}(x, t_{n+1})\, dx - \int_{C_k} \mathbf{w}(x, t_n)\, dx + \int_{t_n}^{t_{n+1}} \sum_a \int_{\partial C_{k/a}} \mathbf{f}(\mathbf{w}) \cdot \boldsymbol{n}_{k/a}\, ds\, dt = 0$$

where $\boldsymbol{n}_{k/a}$ denotes the unit normal vector to the boundary $\partial C_{k/a}$ between cells k and a.
Noting $F\left(\mathbf{w}_k^n, \mathbf{w}_a^n, \boldsymbol{n}_{k/a}\right)$ the approximation of the flux

$$\mathbf{f}\left(\mathbf{w}_k^n, \mathbf{w}_a^n, \boldsymbol{n}_{k/a}\right) \approx \frac{1}{\delta t_n} \int_{t_n}^{t_{n+1}} \int_{\partial C_{k/a}} \mathbf{f}(\mathbf{w}) \cdot \boldsymbol{n}_{k/a} ds\, dt\ ,$$

and dividing by δx_k, we obtain the first order finite volume approximation of Eqs. (1) (see for instance [16, 37, 13]):

$$\mathbf{w}_k^{n+1} = \mathbf{w}_k^n - \frac{\delta t_n}{\delta x_k} \sum_a \mathbf{f}\left(\mathbf{w}_k^n, \mathbf{w}_a^n, \boldsymbol{n}_{k/a}\right)$$

where $\mathbf{f}\left(\mathbf{w}_k^n, \mathbf{w}_a^n, \boldsymbol{n}_{k/a}\right)$ is defined via any three-point solver (see for instance [28, 32, 37]).

In this paper, we have considered a first order space-time discretization using the Godunov solver, i.e. the numerical flux $\mathbf{f}\left(\mathbf{w}_k^n, \mathbf{w}_a^n, \boldsymbol{n}_{k/a}\right)$ is computed with the exact solution of the 1D Riemann problem at the interface k/a with the states w_k^n and w_a^n (for further details see, for instance, [37]). The numerical scheme can be extended to high order space-time discretization combined with a local time stepping method to save computational time as done in [12] for instance.

2.2 Adaptive mesh refinement method

According to the finite volume approximation defined in Sect. 2.1, a local non negative mesh refinement criterion S_k^n is introduced (a detailed discussion on this topic is addressed in Sect. 3) and computed on each cell C_k at time t_n and compared, for instance, to the average

$$S_m = \frac{1}{|\Omega|} \sum_k S_k^n \delta x_k \tag{2}$$

where $|\Omega|$ is the domain size and δx_k is the mesh size. A mesh refinement threshold coefficient $\beta \geqslant 0$ is defined to determine the ratio of the mesh refinement criterion leading to mesh refinement or mesh coarsening.

For each cell C_k, we then process as follows

- if $\frac{S_k^n}{S_m} > \beta$, the mesh is refined, and
- if $\frac{S_k^n}{S_m} < \beta$ the mesh is coarsened.

To avoid excessive refinement, a maximum level $l_{\max}$ is defined. The adaptive process stops when the level l of the cell C_k reaches the maximum level $l_{\max}$. For the definition of the new states on coarsened or splitted cells, we refer to [12]. Moreover, for consistency reasons, the difference of level between the refined cells and the neighbor ones does not exceed 2 (see [12] for further details).

If the mesh refinement threshold β is set too small, the results will be *a priori* accurate at the expense of CPU time because too many cells will be refined. The ”ideal” threshold parameter should provide a balance between accuracy of the results and the computational time. In the most of applications, without *a priori* knowledge on the mean value S_m of the mesh refinement criterion S, it is quite difficult to define what is the ”ideal balancing” and therefore β *has to be calibrated for each numerical experiment.* A more detailed study of existing thresholding strategies is presented in Sect. 4.1.

3 Mesh refinement criteria

The first step in a mesh adaptivity is to construct/find a mesh refinement criterion S which provides information on the flow features such as steep gradients, discontinuities, discontinuities of high derivatives, . . . Once a mesh refinement criterion is chosen, the adaptive process described above can be applied. In what follows, we focus on several type of mesh refinement criteria.

Roughly speaking, a mesh refinement criterion is efficient if it coincides almost everywhere with the local discretization error,

$$\varepsilon_{\delta x}(x,t) := |\mathbf{w}(x,t) - \mathbf{w}_{\delta x}(x,t)|$$

where $|\cdot|$, δx, $\mathbf{w} \neq \mathbf{w}_{\delta x} \in \mathbb{R}^d$ stand respectively for a norm in $\mathbb{R}^d$, the mesh size, the exact solution and the numerical solution.

To describe the efficiency of a mesh refinement criterion S, we introduce the local efficiency index. To this end, without loss of generality, dividing S and $|\mathbf{w}(x,t) - \mathbf{w}_{\delta x}(x,t)|$ by 1 (to obtain non dimensional variables), we define the local efficiency index as follows

$$\text{Efficiency}(x,t) = \frac{|S(x,t)|}{|\mathbf{w}(x,t) - \mathbf{w}_{\delta x}(x,t)|} \,. \tag{3}$$

The analysis of the efficiency index depends, *a priori*, on the normalization constant (here 1). With this definition, the more the mesh refinement criterion S is close to the real error, the more the local efficiency index is close to the value 1.

However, in practice and especially for multidimensional equations, such mesh refinement criterion is unavailable. Nevertheless, relaxing the definition, an ”efficient” mesh refinement criterion can be still defined if at least

- it is able to detect the area where the local discretization error is non zero and,
- if it decreases (as a function of mesh points) like the local discretization error.

In that case, the limit of the local efficiency index as δx goes to 0 converge to a constant.

The mesh refinement criterion can be constructed through several techniques and roughly speaking, it can be classified as error type and ”geometrical” type (typically based on the flow gradient or high derivatives of the solution).

Error estimates can be determined by *a priori* or *a posteriori* methods. The *a priori* methods provides a theoretical bound of the discretization error of the form

$$\varepsilon_{\delta x} \leqslant C(\mathbf{w})\delta x^p$$

for some constant $C(\mathbf{w})$ where p is the order of discretization. These approaches do not require to know the numerical solution contrary to the *a posteriori* error methods.

The ”geometrical” methods are mainly based on the flow gradient, high order derivatives of the numerical solution or any combination of them (yielding to multicriteria methods).

3.1 The numerical test case and the mesh refinement criteria

In what follows, we propose to study several mesh refinement criteria through the resolution of a Riemann problem for the one dimensional Saint-Venant system.

Riemann problem for the Saint-Venant system.

The one dimensional Saint-Venant system (or non linear shallow water equations) is

$$\begin{aligned} \partial_t h + \partial_x(hu) = 0 \\ \partial_t(hu) + \partial_x(hu^2) + \frac{g}{2}\partial_x h^2 = 0 \end{aligned} \tag{4}$$

where the unknowns $h(x,t)$ and $u(x,t) \in \mathbb{R}$ model respectively the height of the water and the depth-averaged velocity of the water at a space-time point (x,t), $x \in \mathbb{R}$ is the downstream coordinate, $t > 0$ is the time, and g is the gravitational constant $g \approx 9.81$ m/s^2. One can show that, System (4) admits a mathematical entropy (energy):

$$E(h,u) = \frac{hu^2}{2} + \frac{gh^2}{2}$$

which satisfies the entropy (energy) relation for smooth solutions

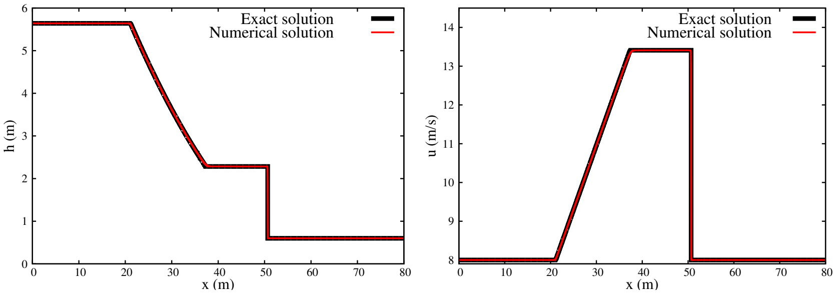
$$\partial_t E(h,u) + \partial_x \left(\left(E(h,u) + \frac{gh^2}{2} \right) u \right) = 0 \tag{5}$$

and used in the sequel as a mesh refinement criterion.

In the sequel, we consider the following Riemann data

$$\forall x \in [0,80],\ \mathbf{w}(x,0) = (h(x,0),u(x,0)) \begin{cases} (5.64, 8) & \text{if } x < 20\,, \\ (0.6, 8) & \text{if } x > 20\,. \end{cases} \tag{6}$$

The exact solution of System (4) with the initial condition (6) can therefore be explicitly computed, see for instance [37, 38, 14]. The exact solution is composed of a left rarefaction and a right shock wave (see Fig. 1). The left rarefaction represents, that we will call later on, the ”smooth” flows.



(a) water height

(b) water speed

Fig. 1 Exact (thick black line) and numerical (thin red line) solution at time $t = 2$ s computed on a uniform grid with $N = 10000$ mesh points

Mesh refinement criteria.

In this test case, the preponderant relative error concerns mainly the fluid velocity u. Therefore, one can define as the ”**error estimate**”, the exact local discretization error,

$$S_k^n := |u_k^n - u_{\text{ex}}(x_k, t_n)|$$

where u_{ex} stands for the exact water velocity. This mesh refinement criterion will be referred by **Criterion 0** in what follows.

Let us now present the other mesh refinement criteria:

Criterion 1: Following [12, 3], we introduce the numerical density of entropy production as the discretization of the Eq. (5)

$$S(x,t) = \partial_t E(h,u) + \partial_x \left(\left(E(h,u) + \frac{gh^2}{2} \right) u \right) \, .$$

The quantity $S(x,t)$ is the so-called entropy production and it is well-known that S becomes negative if the solution develops discontinuities and is zero if the solution is smooth. At the discrete level and in practice, the numerical density of entropy production $S_k^n = S(x_k,t_n)$ is non positive even for smooth solution except for constant profile. Therefore, it provides *a posteriori* information as a **local error indicator**. We will see more precisely that it is not an *a priori* error but only an error indicator known as a **shock criterion type** (see for instance [40, 27] and [12, 17, 3]). In the sequel we consider $S_k^n = -S(x_k,t_n)$ to compare with the positive criterion 0.

Criterion 2: Instead of introducing *a posteriori* error estimates based on the truncation error as classically done (see for instance [8]), we derive a mesh refinement criterion based on the local flux consistency error (see Appendix, Eq. (18), for further details). It reads

$$\begin{aligned} S_k^n =& 2\delta x_k \max\left(1, \left|u(x_{k+1/2},t_n)\right|\right) \times \\ & \max_{i=1,2} \left| \left(\lambda_i(x_{k+1/2},t_n) \left(1 - \lambda_i(x_{k+1/2},t_n) \frac{\delta t_n}{2\delta x} \right) \partial_x I_i(x_{k+1/2},t_n)) \right) \right| \end{aligned}$$

where $\boldsymbol{I} = (I_1 = u + 2\sqrt{gh}, I_2 = u - 2\sqrt{gh})$, $\lambda_1 = u - \sqrt{gh} \leqslant \lambda_2 = u + \sqrt{gh}$ and $x_{k+1/2}$ stands respectively for the Riemann invariants, the eigenvalues of the convection matrix $A(\mathbf{w}) = \frac{\partial \mathbf{f}(\mathbf{w})}{\partial \mathbf{w}}$ and the cell interface. Let us note that thanks to the term $\partial_x I_i = \partial_x u + (-1)^{i+1} \sqrt{g} \frac{\partial_x h}{\sqrt{h}}$ steep gradients or discontinuities of the solution are intrinsically taken into account. This criterion is an ***a posteriori* error estimates**.

Criterion 3 ("geometrical type"): The criteria of geometrical type are generally based on the gradient or high derivatives of the unknown and are defined as $\frac{\partial^p h}{\partial x^p}$, $\frac{\partial^p u}{\partial x^p}$ or any combination (multi-criteria). Let us also emphasize that, following [24, 26, 29], for shallow water problems on non flat topography Z, it is more relevant to use the gradient of $\eta = h + Z$ than h since this quantity is non-zero only in the wave propagation region while the gradient of h can be very large even in regions where the water is at rest due to variations of the topography. Therefore, for its simplicity, we consider the one based on the gradient of η,

$$S_k^n = \left| \partial_x \eta(x_k,t_n) \right| \, .$$

Criterion 4 ("geometrical-error type"): According to the first order of discretization of Eqs. (4), one can force any geometrical criterion to decrease with the same rate of the local error discretization simply by multiplying it by δx_k (see for instance [8]). Thus, we also consider

$$S_k^n = \delta x_k \left|\partial_x \eta(x_k, t_n)\right| \ .$$

In what follows, we use the following notations when no ambiguities are possible: δx will represents δx_k and the (discrete) mesh refinement criterion will be noted $S(x,t) = \sum_k S_k^n \mathbb{1}_{C_k}(x)$ where $\mathbb{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \ , \\ 0 & \text{otherwise.} \end{cases}$

3.2 Sensitivity of the mesh refinement criterion with respect to N.

The first numerical study concerns the comparison of the behavior of the mesh refinement criteria (and its average (2)), as a function of the number of mesh points, to the local discretization error.

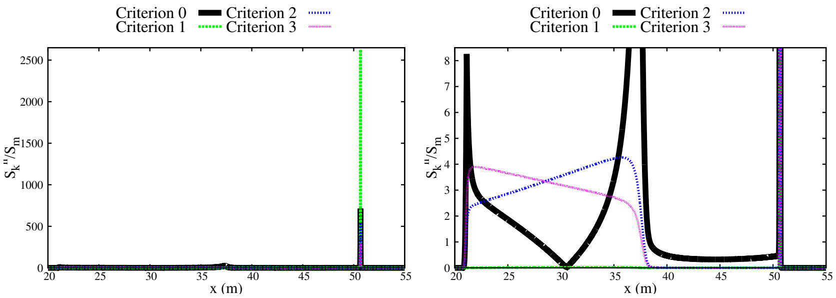
For each parameter N fixed, the mesh refinement criterion is compared to the exact error through the local efficiency index (3), and its average to the global efficiency index

$$\|\text{Efficiency}(\cdot,t)\|_1 = \frac{\|S(\cdot,t)\|_1}{\|u_{\delta x}(\cdot,t) - u_{\text{ex}}(\cdot,t)\|_1} \ .$$

For this purpose, we have computed the numerical solution at time $t = 2$ s of the Riemann problem (4)-(6) with $N = 200 \times k$, $k = 1, \ldots, 15$, 3000, 4000, 5000, 6000, 8000 and 10000 mesh points.

The Fig. 2 represents the normalized (by S_m) mesh refinement criteria 0, 1, 2 and 3 computed with 10000 mesh points. Let us note that the mesh refinement criterion 4 is not displayed since the quantity S/S_m coincides with the criterion 3. We numerically show that all the proposed mesh refinement criteria are able to locate the rarefaction wave and the shock wave (see Figs. 2(a) and 2(b)). As displayed in Fig. 2(b), theirs supports almost coincide with the reference one. However, the hierarchical order of local maxima is not representative of the local error and especially for the criterion 1 which detects almost the shock.

Therefore, from a practical viewpoint, one has to pay attention to the choice of the mesh refinement threshold. For instance, if we set $\beta = 6$ then only the shock region is refined (see Fig. 2(b)) for all of these criteria. In Table 1, we have also computed the rate of convergence of the mean value S_m for all criteria as a function of the mesh points N. The criterion 3 is not considered here since it is clear that S_m does not decrease to 0 when the number of mesh points increases. The mean value of all the criteria, except 1, decreases almost with the same rate of mean value of the criterion 0. As a consequence, up to a constant, the mean values of the criteria 2 and 4 are representative of the mean value of the error. Therefore, in view of the “relaxed” definition of “efficient” mesh refinement criterion (see Sect. 3), the criterion 2 and 4 seems to be the more robust indicators to pilot the adaptive mesh. To go further in the analysis, we now focus on the study of the global efficiency



(a) Global behavior (b) Zoom

Fig. 2 Numerical comparison of mesh refinement criteria

Criterion	rate
0	0.8585
1	0.0161
2	0.9988
4	1.001

Table 1 Convergence rate of the averaged mesh refinement criteria S_m

index. In Fig. 3, we display the global efficiency with respect to the mesh points N. On one hand, in Fig. 3(a), we numerically show that the criterion 1 is definitively not an *a posteriori* error. On the other hand, in Fig. 3(b), we deduce that, for the criteria 2 and 4, the global efficiency index $\|\text{Efficiency}(\cdot,t)\|_1$ converges to a constant $0 < C = C(\mathbf{w}, \delta x, t = 2) < +\infty$, i.e., for δx small enough, one has

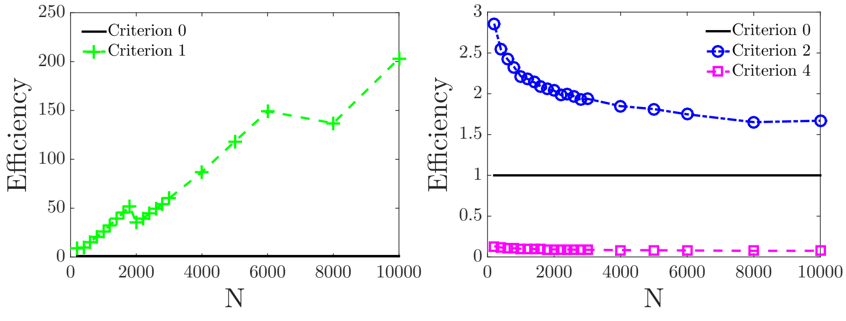
$$\|\text{Efficiency}(\cdot,t)\|_1 = \frac{\|S(\cdot,t)\|_1}{\|u_{\delta x}(\cdot,t) - u_{\text{ex}}(\cdot,t)\|_1} \approx C\ .$$

In Fig. 3(b), the computed constant C is always less than 1 for the mesh refinement criteria 4 while it is greater than 1 for the criterion 2, so that the following (numerical) error estimates holds:

$$\|u_{\delta x}(\cdot,t=2) - u_{\text{ex}}(\cdot,t=2)\|_1 < C\,\|S(\cdot,t)\|_1\ .$$

It confirms that the criterion 2 is the only *a posteriori* error estimate as expected. Even if the criterion 4 is not an *a priori* error estimate, let us highlight that the global efficiency is almost constant as displayed in Fig. 3(b). As a consequence, up to a factor β, the global efficiency index can be set close to 1, replacing $\|S(\cdot,t)\|_1$ by $\beta\,\|S(\cdot,t)\|_1$ in the definition.

As a conclusion, on one hand, we can see that the mean value of the criteria 1 and 3 do not have the good behavior in front of the mean value of the real error. Further-



(a) Efficiency (b) Efficiency

Fig. 3 Efficiency of the mesh refinement criteria

more, the more the number of mesh points is large, the more the global efficiency index is large indicating that the more these criteria are inefficients. Nevertheless, despite this weakness, these criteria are able to detect almost all the local errors. On the other hand, the mean value of the criteria 2 and 4 almost decay with the same rate of the mean value of the local error and therefore they are *a priori* more efficients than 1 and 3. However in both case, as observed in Fig. 2(b), for adaptive method, the threshold has to be carefully chosen.

Remark 1. The global efficiency index can be also regarded as a tool to improve any mesh refinement criterion. Indeed, since $\|S(\cdot,t)\|_1 = L \times S_m$ and noting $\overline{X}$ the mean value of X, $\|S(\cdot,t)\|_1 = L \times S_m$, one can write

$$\|\text{Efficiency}(\cdot,t)\|_1 = \frac{S_m}{\overline{|u_{\delta x}(\cdot,t) - u_{\text{ex}}(\cdot,t)|}} \ .$$

For instance, we have seen that the global efficiency index of the the criterion 1 or 3 increase when the number of mesh points increase (see Fig. 3(a)). Therefore, to make efficient these criteria, the quantity S_m must be modified using a decreasing function β (as a function of the number of mesh points N) such that

$$\|\text{Efficiency}(\cdot,t)\|_1 = \frac{\beta(N) S_m}{\overline{|u_{\delta x}(\cdot,t) - u_{\text{ex}}(\cdot,t)|}} \approx 1 \ .$$

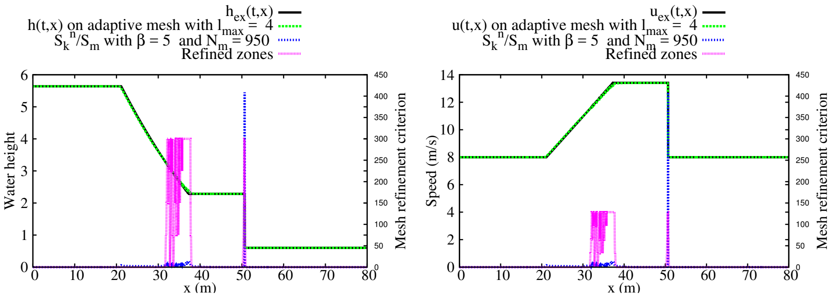
This remark also holds to improve the efficiency of the criterion 2 and 4. In adaptive method, since the number of cells evolves in time $N = N(t)$, $\beta(N)$ will be therefore a time-dependent function. This remark motivates the construction of an automatic (time-dependent) thresholding method.

3.3 Sensitivity of the mesh refinement criterion with respect to β.

The aim of this part is now to study the influence of the mesh refinement threshold β on the numerical solution. We now solve the Riemann problem for the Saint-Venant system (4)-(6) on adaptive grids. Starting with an initial coarse grid composed of 500 cells, we compute the numerical solution for each mesh refinement criterion $\beta = 50$, $\beta = 20$, $\beta = 10$, $\beta = 5$, $\beta = 2$, $\beta = 1$, $\beta = 10^{-1}$, $\beta = 10^{-1}$, $\beta = 10^{-6}$, $\beta = 10^{-9}$ following the adaptive process described in Sect. 2.2. For each numerical experiment, the maximum level of mesh refinement is fixed to $l_{\max} = 4$. During the adaptive computation, for each numerical test, we calculate the average of the number of cells $N_m(\beta,t) = N_m(\beta,t=2) = \frac{1}{M}\sum_{n=0}^{M} N(\beta)$ where M is the number of time step. N_m will be used to compute the numerical order of convergence.

The numerical results (green large dashed line) obtained for the water height and the water speed with $\beta = 5$ are shown in Figs. 4(a)–7(a) and 4(b)–7(b) and compared to the exact one (solid black line). We also display the mesh refinement level l (pink small dashed line) and the mesh refinement criterion (green medium dashed line).

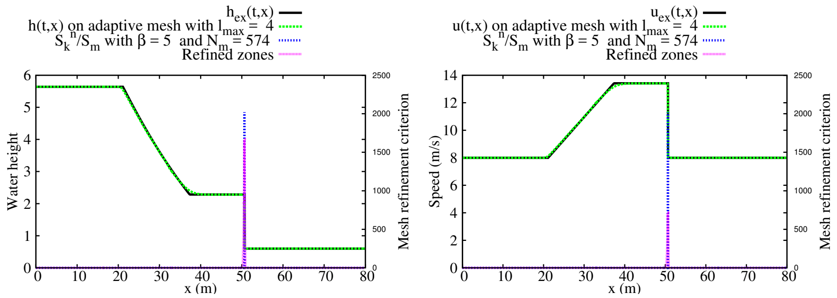
As noticed before, the criterion 1 is a shock type criterion. In this example, $\beta = 5$ is not small enough to refine in the smooth region and only the shock one is refined as displayed in Figs. 5(a) and 5(b). Concerning the criteria 0, 2, 3 and 4 (see Figs. 4, 6, 7 and 8), we clearly observe that $\beta = 5$ is not small enough and the rarefaction wave (smooth region) is partially refined. As a consequence, it yields to an oscillating mesh refinement criterion which impacts the overall quality of the numerical solution (see for instance Figs. 4(a), 6(a), 7(a) and 8(a)). Therefore, as observed in Figs. 4–8, each mesh refinement criterion exhibits a strong dependance on β. Next, we propose to study the sensitivity of the L^1 discretization error with respect to the mesh refinement threshold. In Fig. 9, we display the L^1 discretization error with respect to $\log(\beta)$ for each criteria. In Figs 9(a) and 9(b), we observe that if the mesh refinement threshold is less than some an "optimal" threshold β_{opt}, the L^1 discretization error stops to decrease since almost all cells are refined to



(a) Water height

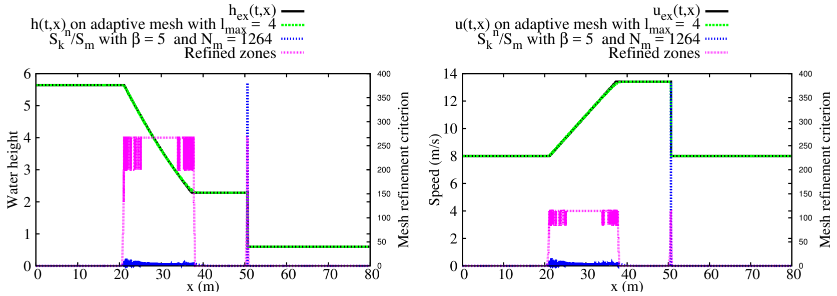
(b) Water speed

Fig. 4 Numerical results for $\beta = 5$ with the criterion 0



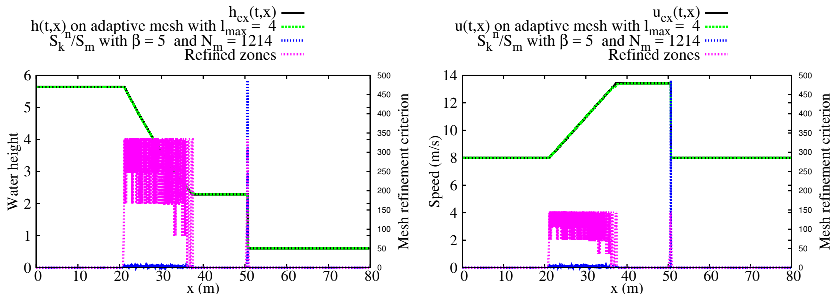
(a) Water height (b) Water speed

Fig. 5 Numerical results for $\beta = 5$ with the Criterion 1



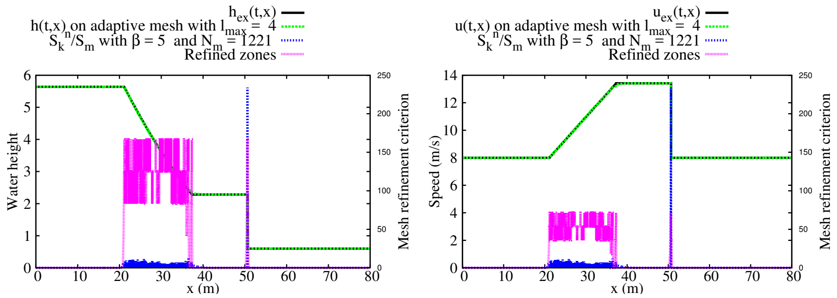
(a) Water height (b) Water speed

Fig. 6 Numerical results for $\beta = 5$ with the Criterion 2



(a) Water height (b) Water speed

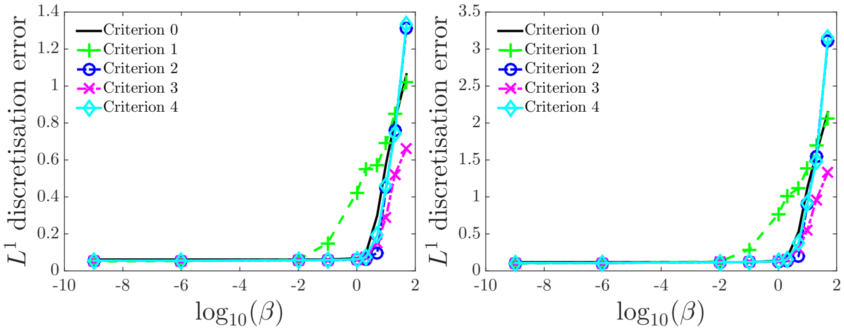
Fig. 7 Numerical results for $\beta = 5$ with the Criterion 3



(a) Water height (b) Water speed

Fig. 8 Numerical results for $\beta = 5$ with the criterion 4

the maximum level of refinement. This ”optimal” threshold depends on the mesh refinement criterion. For instance, in Fig. 9(b) we see that for almost all criteria, the ”optimal” threshold is of order $\beta_{\text{opt}} \approx 1$ while it is of order 0.01 for the criterion 1. Let us also highlight that, at least for this test case, $\beta_{\text{opt}} \in (0,1]$ for any mesh refinement criteria. In Fig. 10, we represent the numerical solution for $\beta \in (0,1]$ with $\beta = 1$ and $\beta = 0.01$ using the criterion 2. In particular, we numerically show that $\beta = \beta_{\text{opt}} = 1$ (see Fig. 10(a)) is clearly the good choice while $\beta = 0.01$ is too small and yields to unnecessary refinement.

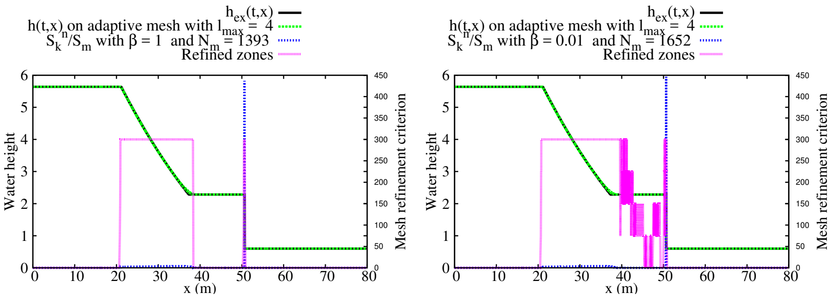


(a) Water height (b) Water speed

Fig. 9 Sensitivity of the mesh refinement criterion: L^1 discrete error vs $\log_{10}(\beta)$

Finally, let us compute the numerical order of convergence of the L^1 discretization error at time $t = 2$ s as follows

$$N_m(\beta) \to \left\| \mathbf{w}_{\delta x,\beta}(\cdot,t) - \mathbf{w}_{ex}(\cdot,t) \right\|_1 \ .$$



(a) $\beta = 1$

(b) $\beta = 0.01$

Fig. 10 An ”optimal” threshold (using the criterion 2)

The obtained results are summarized in Table 2. Despite the above remarks, it is interesting to note that all the criteria have almost the same decay of the criterion 0. In particular, these results will be confronted to the one obtained in Sect. 4.2, Table 3 using the automatic threshold method.

Criterion	$\|h - h_{ex}\|_1$ *vs* N_m	$\|u - u_{ex}\|_1$ *vs* N_m
0	1.4018	1.4035
1	1.6159	1.5971
2	1.5302	1.5698
3	1.4902	1.4689
4	1.5277	1.5596

Table 2 Convergence rate of the L^1 discretization error

As a conclusion, if the threshold is set

- too ”large”, the smooth region are not refined (see for instance Fig. 5(a)),
- not ”small enough” yields to an oscillating mesh refinement criterion impacting the overall quality of the numerical solution (see for instance Fig. 8(b)),
- too small yields to unnecessary refinement without improving the overall quality of the numerical solution (see for instance Fig. 10(b)),
- to β_{opt} yields to satisfactory results (see for instance Fig. 10(a)).

Moreover, all these assessments indicate that the mean value S_m of a mesh refinement criterion S may be relevant if the ”optimal” threshold is well-defined. More precisely, for all criteria, comparing the local mesh refinement S_k^n to $\beta_{opt}S_m$ seems to be relevant if β_{opt} is automatically (see Remark 1) well-chosen in the interval $(0,1]$. In what follows, we focus on how to define a relevant coefficient β_{opt}.

4 Mesh refinement threshold

In view of the previous discussion, it is clear that the mesh refinement threshold parameter cannot be set as a fixed constant. To this end, we first review existing thresholding strategies and secondly, we propose a new one based on the analytical distribution function which provides a detailed description of the mesh refinement criterion in terms of local maxima. It allows to define *a priori* a relevant β_{opt} threshold.

In the sequel, following the context, we make use of the notation α for a dimensional threshold and β for a dimensionless threshold as done before. In the most of the case, one can write $\alpha = \beta S_m$ and the adaptive procedure is then, for each cell C_k,

- *if $S_k^n > \alpha = \beta S_m$, the mesh is refined, and*
- *if $S_k^n < \alpha = \beta S_m$ the mesh is coarsened.*

4.1 Review of some thresholding methods

As seen before, the mesh refinement threshold is certainly the most important parameter in the adaptive process. In order to construct a new method, we first propose to revisit some classical methods.

4.1.1 Ideal case.

The perfect situation is when the local discretization error is known. In that case, if the scheme is stable then there is no *a priori* reason to propagate heterogeneous error and all local maxima detected have approximately the same order of error in terms of magnitude, see for instance [2]. As a consequence, the involved mean error S_m (2) is representative of the discretization error and it can be used as a mesh refinement threshold parameter $\alpha = 1 \times S_m$. Unfortunately, as pointed out before, in most real life models, such *a priori* error estimates are unavailable (or hard to compute) and the adaptive process requires a tunable coefficient.

4.1.2 In practice.

Let us consider a mesh refinement criterion S and let us assume that there exists a function $f : x \in \mathbb{R} \mapsto \mathbb{R}^+$ such that

$$S(x,t) = f(\varepsilon(x,t))$$

where $\varepsilon(x,t)$ stands for the local error at point x and time t. As in the ideal situation, one can calculate the mean of the mesh refinement criterion S_m. However, in a general case nothing ensures that this average will be representative of the mean error as numerically illustrated in Sects. 3.2 and 3.3. As described above, if the mesh refinement criterion S is a function of the error, see for example [20], then the mesh is refined if $S(x,t) > S_m(t)$ or coarsened otherwise. Therefore, if f is a linear function then the efficiency index is equal to a constant.

In practice, the function f is more complex than a linear function as observed in Sect. 3.2 and Fig. 3(b) for instance. These functions deforms the local error and the local magnitude of the error is no longer representative. As a consequence, the mean of the mesh refinement criterion is no longer representative of the global signal. Moreover, these functions are sensitive to any non linear transformation. For instance, the numerical density of entropy production is a well-known ”shock criterion type” (see [40]) which exploses where the physical discontinuity appears. This type of mesh refinement criterion involving a large variety of scales is therefore complex to analyse by the only knowledge of its average.

To overcome this difficulty, several strategies can be found in the literature. We propose to review a list of the most popular methods.

Mean method.

It is the simplest and cheapest method. The cells are refined if, for all $t > 0$,

$$S(x,t) > \beta \frac{1}{|\Omega|} \int_{\Omega} S(x,t)\ dx$$

where β is a tunable dimensionless threshold parameter. This strategy have been used in Sect. 3.3. A second method to take into account the fluctuations of the mesh refinement criterion is: the cells are refined if, for all $t > 0$,

$$S(x,t) > \beta \frac{1}{|\Omega|} \int_{\Omega} S(x,t)\ dx + \delta\sigma(x,t)$$

where σ is the standard deviation and $\delta \in \mathbb{R}$.

These methods are largely used due to their simplicities, see for instance [21, 40, 20, 31, 12, 17, 3] and the reference therein. A well chosen parameter couple (β,δ) yield to an efficient adaptive mesh refinement. However, (β,δ) are case-dependent, especially if the mesh refinement criterion is sensitive to multiple scales solutions (including smooth and discontinuous solutions). The main drawback of this approach is to calibrate the parameter for each test under consideration. However, one can fix the choice of the parameter if the typical behavior of the mesh refinement is well-known for any solutions. For instance, one can transform the mesh refinement criterion by applying a non-linear transformation, see for instance [40], as follows

$$S_{new}(x,t) = \left(\frac{l_l(x)}{l_g(x)}\right)^{\gamma} S_{old}(x,t)$$

where γ is a constant depending on the numerical scheme, $(l_l, l_g) \in \mathbb{R}_+^2$ are respectively a local and a global physical length of the test problem.

Controlled mesh size growth method.

Unlike the mean method, the main objective of this method is to control the mesh size growth. This approach have been used for example in [7, 35, 34]. After solving the equations on a given grid, the mesh refinement criterion is calculated for all cells. Then all values are sorted in a decreasing order in terms of local maxima and a fraction β of the total number of cells is chosen to be refined. Thereafter, as a safety margin, the neighborhood cells are also selected to be refined and closest neighborhoods as well. At the end, the new grid is composed of more than cells announced due to the safety margin (which increases the support of the local refined region). However, this method has two drawbacks. The first one concerns the sorting algorithm which increases the global computational cost and the second one is the threshold parameter β which again depends on the underlying test case.

Filtering: two steps method.

In contrast with the ”shock capturing method”, the method of ”filtering” is based on a smooth flow detection, see for instance [40, 1]. As a first step, the mean method is applied to identify strong nonlinearities. If the mesh refinement criterion is heterogeneous, then almost all of the cells with large local maxima are detected. Considering a threshold parameter β, cells for smooth flows are almost contained in $[0, \beta \max_x S]$. As a second step, the mean method is again applied to the remaining field (ie. without the previous high values) to detect smoother cells. However, in view of the discussion on the mean method, one can notice that in the case of very heterogeneous mesh refinement criterion, a two step mean method may be not enough to efficiently capture smooth solutions. The second drawback concerns the parameter β which is again dependent of the test problem.

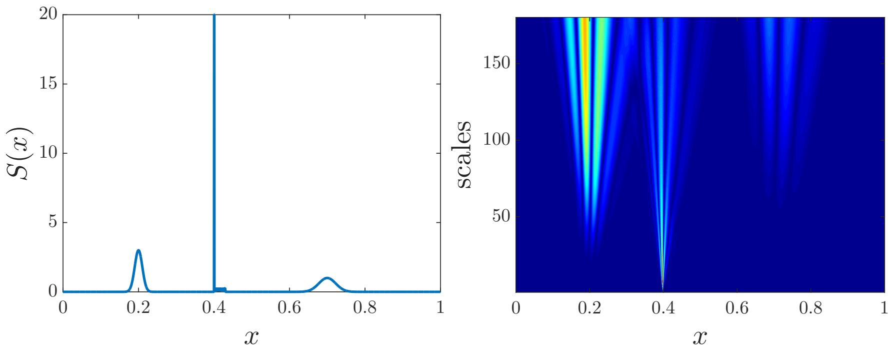
Filtering: wavelet method.

In contrast with the previous method, the detection of smooth and discontinuous flows can be efficiently captured with a space-time localisation based on a wavelet transform, see for instance [23]. A nice feature of wavelets are the detection of discontinuities and high transitions along with the resulting absolute values of wavelet coefficients are large. The localisation of such a region is generally well-captured for small scales [1] as displayed in Fig. 11(b) for a given signal

[1] the wavelet coefficients in this example are computed with the Daubechies wavelet with four vanishing moments [10, 11]

$$x \in [0,1],\ S(x) = \exp(-1000(x-0.7)^2) + 3\exp(-5000(x-0.2)^2)$$
$$+ \begin{cases} 100 & \text{if } x \in [0.4, 0.401] \\ 0.3 r_{0,1}(x) & \text{if } x \in [0.401, 0.43] \end{cases}$$

where $x \mapsto r_{0,1}(x) \in [0,1]$ returns a random number, see Fig. 11(a). High variations of the signal yield to large absolute value of wavelet coefficients centered around the discontinuity at all scales. Depending on the wavelet's support, the larger the scale is, the larger the set of coefficients affected in the wavelet transform is. It defines the so-called ”cone of influence”. As a consequence, the discontinuity have the smallest scale. On the contrary, smooth signal produces relatively large wavelet coefficients at large scales and again the definition of the cone of influence holds. The interpretation of the absolute value of wavelet coefficients as a mesh refinement criterion is based on the cone of influence and its support. For instance, in the example illustrated in Fig. 11, for a given scale (which corresponds to a threshold parameter β), say $y = 100$, the intersection with the cone of influence provides three intervals located around the point $x = 0.2$, $x = 0.4$ and $x = 0.7$ which are the regions to be refined. However, as for the method described before, we are still confronted with the choice of a threshold parameter. Moreover, even if this method is based on fast wavelet transform, it increases the global cost of the numerical method.



(a) Mesh refinement criterion (b) Wavelet coefficients ($x =$ space, $y =$ scale)

Fig. 11 Illustration of the wavelet transformation for a given mesh refinement criterion computed with the Daubechies wavelet with four vanishing moments (warm colors correspond to large coefficients and cold colors to small coefficients)

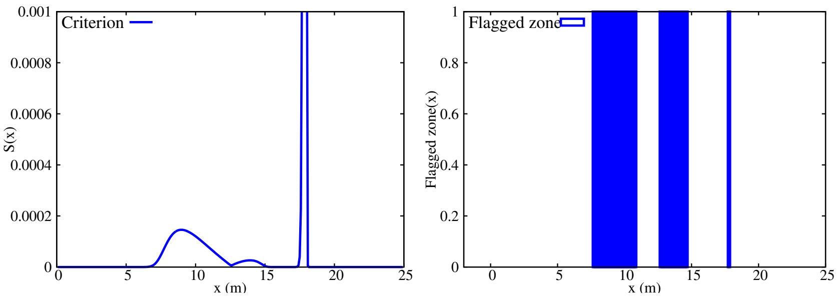
Local maxima approach.

As in the detection methods, one can also focus on the search of local maxima but without sorting cells. It can be achieved if the mesh refinement criterion is well chosen such that it only deforms the error scale but it conserves the sign of the local error. In this case, one can write, for all $t > 0$

$$\frac{\partial S(x,t)}{\partial x} = \frac{d}{dx} f(x) \frac{\partial \varepsilon(x,t)}{\partial x} \ .$$

However, as a first drawback, the local maxima of the error are not necessarily those of the mesh refinement criterion. Once a local maximum is detected, the surrounding area to be refined should be large enough. It can be defined by seeking the closest inflexions points to the local maximum or by the interval $[x - \beta_0 \max_x S, x + \beta_0 \max_x S]$ where β_0 is some parameter depending on the local maximum. As a second drawback, such a method can be quite costly in a multidimensional settings. Moreover, the parameter β_0 must be calibrated for each practical applications.

In Fig. 12, we illustrate the method for a given mesh refinement criterion. The intervals to refine around each local maxima are defined by the use of the closest inflexion points (see Fig. 12(b)). In particular, one can see the ability of the local maxima method to catch smooth flow region, poorly detected by the mean method (see Fig. 12(a)).



(a) Mesh refinement criterion S (b) Region to be refined

Fig. 12 Illustration of the local maxima method for a mesh refinement criterion involving multiple scales

Dannenhoffer cumulative distribution method.

The methodology proposed by Dannenhoffer [9] is simple and efficient to ”automatically” set the threshold parameter. Let us consider a given discrete mesh refinement criterion $S(x,t_n) = \sum_{k=1}^{N} S_k^n \mathbb{1}_{C_k}(x)$ where N is the total number of cells at time t_n. Let $(\alpha_j)_{0 \leqslant j \leqslant M}$ be an increasing sequence of $M+1$ threshold parameter such that $\alpha_0 = 0$ and $\alpha_M = \max_x S(x,t_n)$. Then, the cumulative distribution function $d_j = d(\alpha_j)$ is defined as follows:

$$d_j = \#\{k \ ; \ S_k^n > \alpha_j\} \tag{7}$$

where # is the number of elements in the set $\{S_k^n > \alpha_j\}$. For a given oscillating mesh refinement criterion illustrated in Fig. 13(a), we display the cumulative Dannenhoffer distribution function in Fig. 13(b).

Then, one can define several possible threshold denoted from A to D (see Figs. 13(a) and 13(b)). The level *D* corresponds to the zone for which the mesh refinement criterion oscillates (called background noise in [9]) as already observed in Fig. 5(a). The level *A* is too high and almost all information are ignored for smooth flows (mainly located close to the oscillating area).

Remark 2. The oscillating area is in general the consequence of a bad initial threshold as already pointed out in Sect. 3.3 (see also Fig. 8(a) for instance).

Dannenhoffer suggests to set a threshold between *B* and *C* at the "knee" of the cumulative distribution which separates disparate behaviors. As a consequence, if the location of the "knee" can be determined, then an automatic threshold can be defined. However, it is mentioned in [9] that the "knee" cannot be well-localized and for general purpose Dannenhoffer defines the "knee" localisation as the smallest α such that

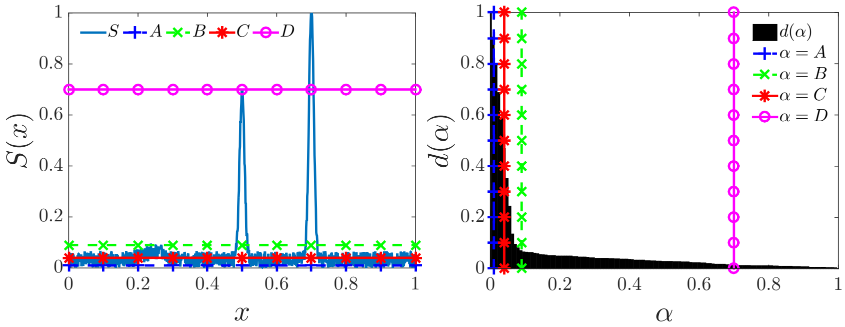
$$\frac{\partial d(\alpha)}{\partial \alpha} \approx -1 \ .$$

Moreover, to compute accurately the above approximation, a fine interpolation (or a large number of cells C_k) is required to define the "cumulative" distribution function as we will see later on. Otherwise, and as noticed in [9], the above method sometimes yields to an unacceptable threshold which detects too many unnecessary cells to be refined. Therefore, in practice, the slope "-1" is arbitrary and must be calibrated following the problem under consideration.

Let us also mention, based on this cumulative distribution function, Powell [30] proposes to automatically set the threshold as the lowest value of the threshold parameter that produces a local maximum in the curvature of the distribution. Let us note that both approach have been introduced in the context of steady problems. This approach based on the distribution function is an interesting method that we will improve in the next section.

4.2 A distribution function for automatic thresholding

In the previous section, we have revisited some of the main procedure to define the mesh refinement threshold. All the presented method can be efficient if it is combined with a mesh refinement criterion based on accurate error estimates. However, as emphasized before, such error estimates are difficult to construct, and whenever it is available it is not easy to compute and can be quite costly for multidimensional problems. Therefore, widely used mesh refinement criteria are based on the flow gradient which are easy-to-compute but require a tunable threshold parameter to be calibrated following the physical problem under consideration.



(a) Mesh refinement criterion (b) Cumulative distribution function

Fig. 13 Dannenhoffer cumulative distribution function for a given mesh refinement criterion

We propose here to construct a simple method, based on the Dannenhoffer distribution function [9], to choose easily a suitable ”optimal” mesh refinement threshold parameter for any given mesh refinement criterion.

For the sake of simplicity and readability, we present our new distribution function in the one-dimensional case. Let us assume that the solution of System (1) is smooth and let us consider a smooth (at least twice differentiable) mesh refinement criterion $S(\mathbf{w}) = S(x,t) \in \mathbb{R}^+$, $x \in [0,L]$ and $t > 0$ where L is the length of the domain. The time t being fixed, we write in the sequel $S = S(x)$. Without loss of generality, we suppose that $S(0) = S(L) = S'(0) = S'(L) = 0$ and $0 < S_\infty = \max_{x\in(0,L)} S(x) < \infty$ (if $S_\infty \equiv 0$ the numerical solution is identically equal to zero).

In view of the above assumptions, the set

$$Z_\alpha = \{x \in (0,L);\ \varphi_\alpha(x) = S(x) - \alpha = 0 \text{ and } S'(x) \neq 0\}$$

is not empty. For each $0 < \alpha < S_\infty$, since S has at least one maximum, then there exists $p_\alpha \in \mathbb{N}_*$ such that the number of elements in the set Z_α is

$$\# Z_\alpha = 2p_\alpha .$$

Thus, for all $\alpha \in (0, S_\infty)$, one can describe the set Z_α as follows

$$Z_\alpha = \left\{x_0(\alpha) < x_1(\alpha) < \cdots < x_{2p_\alpha-2}(\alpha) < x_{2p_\alpha-1}(\alpha)\right\} .$$

Let us assume that S has p local maxima. Then there exists an *increasing* sequence $\left(\alpha_k^*\right)_{1\leqslant k\leqslant p}$ and a sequence $\left(x_k^*\right)_{1\leqslant k\leqslant p}$ such that

$$\forall k = 1,\ldots,p \quad S'(x_k^*) = 0,\ S''(x_k^*) < 0 \text{ and } S(x_k^*) = \alpha_k^* .$$

By definition the sequence $\left(\alpha_k^*\right)_{1\leqslant k\leqslant p}$ sorts the local maxima from the smallest to the largest so that $\alpha_p^* = S_\infty$. Let us also emphasize that the abscissa of the local maxima $x_k^* \notin Z_\alpha$, $\forall k = 1,\dots,p$, $\forall \alpha \in (0, S_\infty)$.

With these settings, the considered distribution function reads

$$\alpha \in [0, S_\infty] \mapsto d(\alpha) := \begin{cases} L & \text{if } \alpha = 0\,, \\ \sum_{k=1}^{p_\alpha} x_{2k+1}(\alpha) - x_{2k}(\alpha) & \text{if } 0 < \alpha < S_\infty\,, \\ 0 & \text{if } \alpha = S_\infty\,. \end{cases} \tag{8}$$

Remark 3. This distribution function is also called the decreasing rearrangement function of S which is widely used in optimal transport problems. For $\alpha \in [0, S_\infty]$, $d(\alpha)$ is the Lebesgue measure of the set $\{S(x) > \alpha\}$.

Then, one has:

Theorem 1. *Let $l \in \mathbb{N}_*$, $S \in C^l([0,L], \mathbb{R}^+)$ be a smooth mesh refinement criterion such that $S(0) = S(L) = S'(0) = S'(L) = 0$, $0 < S_\infty = \max_{x\in(0,L)} S(x) < \infty$ with p local maxima.*

Then

1. $\int_0^{S_\infty} d(\alpha)\,d\alpha = \int_0^L S(x)\,dx$.
2. $d \in C^0([0, S_\infty], \mathbb{R}^+)$ *and its first derivative satisfies*

    a. $\forall \alpha \in [0, S_\infty]$, $d'(\alpha) < 0$
    b. $\forall k \in [\![0, p]\!]$, $\lim_{\alpha\to\alpha_k^*} d'(\alpha) = -\infty$ *with the convention* $\alpha_0^* := 0$

3. $d \in C^l\left(D_*, \mathbb{R}^+\right)$ *on the set*

$$D^* := \bigcup_{k=0}^{p-1} (\alpha_k^*, \alpha_{k+1}^*)\,.$$

Proof. By construction of the distribution function (8), the first and the second property holds. Let us now calculate its derivative. To this end, for $\alpha \in [0, S_\infty]$, its first derivative is

$$d'(\alpha) = \sum_{k=1}^{p_\alpha} \frac{d}{d\alpha} x_{2k+1}(\alpha) - \frac{d}{d\alpha} x_{2k}(\alpha)\,.$$

Since $\forall k \in [\![0, p_\alpha]\!]$, $x_{2k}(\alpha) \in Z_\alpha$, i.e., $\varphi_\alpha(x_{2k}(\alpha)) = S(x_{2k}(\alpha)) - \alpha = 0$, we get

$$\frac{d}{d\alpha} x_{2k}(\alpha) = \frac{1}{S'(x_{2k}(\alpha))}\,.$$

Since $S'(x_{2k}(\alpha)) > 0$ and $S'(x_{2k+1}(\alpha)) < 0$ by construction, we therefore deduce that

$$d'(\alpha) = \sum_{k=1}^{p_\alpha} \frac{S'(x_{2k}(\alpha)) - S'(x_{2k+1}(\alpha))}{S'(x_{2k+1}(\alpha))S'(x_{2k}(\alpha))} < 0\,.$$

The remaining properties are obtained as a straightforward consequence of the above inequality. □

In the sequel, for practical purpose, the distribution function (8) is normalized so that $d(\alpha)$ corresponds to a percentage of the domain $[0,L]$, i.e., $d([0,S_\infty]) \subset [0,1]$.

To illustrate the notations introduced above, we have displayed in Fig. 14, for a given mesh refinement criterion S (see Fig. 14(a)), the corresponding distribution function (8) (see Fig. 14(b)), its first (see Fig. 14(c)) and its second derivative (see Fig. 14(d)). In this example, we have considered a function S which has three local maxima, i.e. p=3. In Fig. 14(b), the points for which the distribution function has vertical asymptote are

$$\alpha_1^* = S(x_1^* = 5) = 5,\ \alpha_2^* = S(x_2^* = 2.5) = 10 \text{ and } \alpha_3^* = S(x_1^* = 7.5) = 20 = S_\infty\ .$$

and corresponds to the local maxima of S sorted from the smallest to the largest. In Fig. 14(a), for $\alpha = 2$, the set Z_α is composed of $6 = 2p_\alpha$ elements which are approximately

$$x_0(\alpha) \approx 2.0957,\ x_1(\alpha) \approx 2.9043,\ x_2(\alpha) \approx 4.566,$$

$$x_3(\alpha) \approx 5.434,\ x_4(\alpha) \approx 7.3921 \text{ and } x_5(\alpha) \approx 7.6079\ .$$

The normalized distribution function at $\alpha = 2$ is then computed as follows

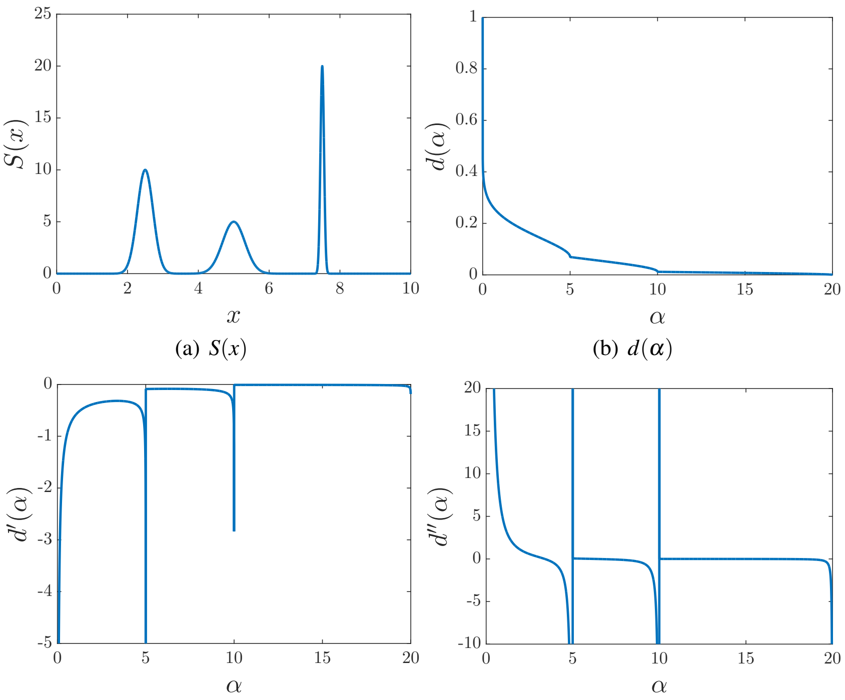
$$d(\alpha) = \frac{(x_1(\alpha) - x_0(\alpha)) + (x_3(\alpha) - x_2(\alpha)) + (x_5(\alpha) - x_3(\alpha))}{L}\ .$$

Remark 4 (General properties of the distribution function (8)*).*

1. The distribution function (8) is the continuous version of the Dannenhoffer cumulative distribution function (7). Indeed, in view of the definition of the distribution function (8), for a given interval $l_\alpha = x_{2k+1}(\alpha) - x_{2k}(\alpha) > 0$, there exists m_α cells C_k of meshsize δx such that $l_\alpha \approx m_\alpha \delta x$, therefore at the discrete level it is then equivalent to consider the sum of the length l_α or the number of cells C_k for which $d_k := d(\alpha_k) > \alpha$ with $\alpha_k = k\frac{L}{M}$ for some $M > 0$.
2. Introducing the function $\beta \mapsto \tilde{d}(\beta) = d(1-\beta) = d(\alpha)$ extended by 0 for $x < 0$ and 1 for $x > 1$. Its derivative verifies $\tilde{d}'(\beta) = -d(\alpha) > 0$ but $\int_{\mathbb{R}} \tilde{d}'(\beta) d\beta \neq 1$. Therefore, the distribution function d (8), as well as (7), is not a cumulative distribution function in the sense of the probability theory.
3. From a theoretical viewpoint, the method proposed by Dannenhoffer [9] selects more cells than the Powell method [30]. Let us recall that Dannenhoffer [9] selects the smallest α, α_{D}, such that $d'(\alpha) = -1$ while Powell [30] chooses the smallest α, α_{P}, such that d”$(\alpha) = 0$. Therefore, in view of the variations of d” (see Fig. 14(d)), one has

$$\min\{\alpha;\ d'(\alpha) = -1 \text{ and } d''(\alpha) = 0\} = \min\{\alpha;\ d'(\alpha) = -1\}$$

and therefore



(a) $S(x)$

(b) $d(\alpha)$

(c) $d'(\alpha)$

(d) $d''(\alpha)$

Fig. 14 The normalized distribution function, its first and second derivative for a smooth mesh refinement criterion $S(x) = L\exp(-10(x-L/4)^2) + L/2\exp(-5(x-L/2)^2) + 2L\exp(-200(x-3L/4)^2)$ with $L = 10$

$$\alpha_{\mathrm{D}} < \alpha_{\mathrm{P}} .$$

Remark 5 (Link with other threshold methods). The distribution function (8) can be regarded as a general method including almost all the methods presented in the previous section.

- As already noticed in Remark 4, the distribution function (8) corresponds to the continuous version of the Dannenhoffer distribution function [9].
- The main idea of filtering and local maxima are based on the localization of the smallest and the largest maximal value. By construction of the distribution function (8), these values are well known and are respectively α_1^* and α_p^* if S has p local maxima.

In view of Theorem 1, Remarks 4 and 5, the distribution function (8) provides a complete description of the mesh refinement criterion S. Therefore, if the mesh refinement criterion is constructed through an accurate error estimates, the distribution function (8) will provide an accurate description of the local error, otherwise it will helps to localize the smallest and the largest local maxima. The main and crucial question is how to automatically set the threshold parameter using the distribution function (8) when the mesh refinement criterion is not *a priori* well-linked to the local error? Setting α as Dannenhoffer [9] or Powell [30] proposes? Is there an optimal choice? The answer is of course no, there is no reason to have an optimal threshold parameter except if the mesh refinement criterion is itself the local error. However, one can define, at least, without any effort the localization of the smooth region, i.e., the region in general for which the refinement is necessary and often ignored if the threshold parameter is not small enough.

4.2.1 Strategy of the automatic thresholding: some remarks and improvements of the Dannenhoffer approach

According to Remark 4, the distribution function (8) is computed as the Dannenhoffer distribution function (7). Furthermore, in view of the conclusion in Sect. 3.3 (see also Figs. 9(a) and 9(b)), we will restrict the search of the ”optimal” threshold parameter in the interval $(0,S_m]$ where S_m is the mean value of S (see Eq. (2)).

Thus, in practice, for a given discrete mesh refinement criterion $S(x,t_n) = \sum_{k=1}^{N} S_k^n \mathbb{1}_{C_k}(x)$ where N is the total number of cells at time t_n, we define an increasing sequence $(\alpha_j)_{0\leqslant j\leqslant M}$ of $M+1$ threshold parameter such that $\alpha_0 = 0$ and $\alpha_M = S_m$. In the presence of discontinuities (large maxima) in the solution, the smooth region is in general localized for small value of α. Therefore, we suggest in practice the following non uniform mesh

$$(\alpha_j)_{0\leqslant j\leqslant M} = \left(S_m \left(\frac{j}{M}\right)^{\beta} \right)_{0\leqslant j\leqslant M}$$

with $\beta > 1$.

Then, following Dannenhoffer [9], the piecewise constant distribution function $d_j = d(\alpha_j)$ can be defined as follows:

$$d(\alpha) = \sum_{k=0}^{M-1} d_j \mathbb{1}_{(\alpha_j,\alpha_{j+1})}(\alpha) \text{ with } d_j = \#\{k\ ;\ S_k^n > \alpha_j\} \tag{9}$$

where # is the number of elements in the set $\{S_k^n > \alpha_j\}$.

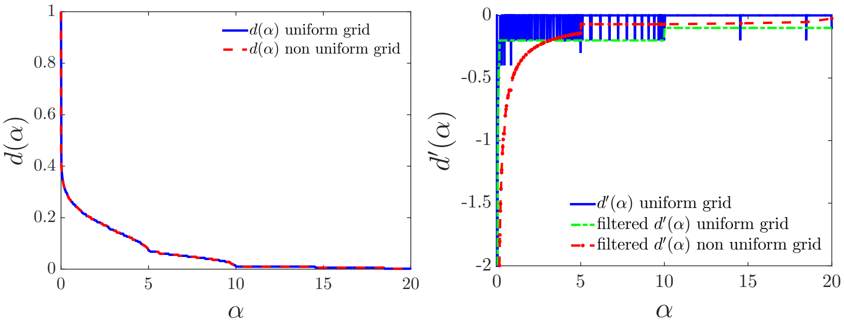
Let us notice that, by construction, Distribution (9) is a piecewise constant function. Thus, the yielding first order derivative can degenerate since d_j and d_k are not necessarily distinct producing unwanted zero value in the derivative and therefore a bad approximation. Therefore, in Algo. 1, we propose to filter the discrete derivative $d'_j = \frac{d_j - d_{j-1}}{\alpha_j - \alpha_{j-1}}$.

As displayed in Fig. 15, at least for small value of α, the analytical derivative is ”correctly” approximated on a non uniform grid with the filtrated derivative (see Fig. 15(b) and Fig. 14(c)) and badly approximated otherwise. This region is precisely the

Algorithm 1 filtered derivative

```
$l = 0$
for $j = 1, M$ do
    if $d'_j \neq 0$ then
        $\alpha'_l = \alpha_j$
        $d'_l = d_j$
        $l = l + 1$
    end if
end for
```

part of the domain in which the Dannenhoffer threshold parameter will be chosen as the smallest α_j such that $d'(\alpha) \approx -1$. To use the Dannenhoffer threshold parameter, the derivative must be computed accurately otherwise one can observe in practice too many level of unnecessary refinement. Indeed, in this example, the Dannenhoffer threshold parameter is computed and we have obtained $\alpha_{\text{D}} = 0.06$ if $\beta = 1$, 0.20 if $\beta = 2$ and 0.37 if $\beta = 4$ with $M = 1000$. This computation also shows that the approximation of the derivative has a strong dependence on β. Here and in the most of applications considered, we have observed that at least $M = 10000$ points are required to suppress the strong dependency (which increase the global cost of the numerical scheme). Due to these several remarks, the approach by Powell (based on the computation of the second derivative) will be not any more considered.



(a) $d(\alpha)$ (b) $d'(\alpha)$

Fig. 15 Distribution function and its first derivative of the mesh refinement criterion $S(x) = L\exp(-10(x-L/4)^2) + L/2\exp(-5(x-L/2)^2) + 2L\exp(-200(x-3L/4)^2)$ with $L = 10$ obtained with a uniform grid of 500 cells for x and 1000 cells uniform and non uniform with $\beta = 4$

4.2.2 Strategy of the automatic thresholding based on the distribution function

On the contrary, let us emphasize that by construction, the distribution function (9) is quite insensitive to β (see Fig. 15(a)). However, using only the distribution function (9), it is numerically difficult to catch the smallest local maxima of S. Therefore, we propose the following function

$$f(\alpha) = \alpha d(\alpha) \tag{10}$$

for which one has the following properties:

Corollary 1. *Assume that S is twice differentiable and has p local maxima. Then, by virtue of Theorem 1, there exists*

$$\forall k = 0 \dots p-1,\ \alpha^{**}_{k+1} \in (\alpha^*_k, \alpha^*_{k+1}) \textit{ such that } d''(\alpha^{**}_{k+1}) = 0$$

and the function f (10) has p local maxima $\overline{\alpha}_1, \dots, \overline{\alpha}_p$ such that

$$\forall k = 1 \dots p,\ \overline{\alpha}_k \in (\alpha^{**}_k, \alpha^*_k)\ .$$

Proof. The function f inherits the same regularity property of d as given in Theorem 1. Therefore if S is twice differentiable, then it is sufficient to show that on each interval $(\alpha^*_k, \alpha^*_{k+1})$, the function f is concave. The proof is similar for all k. Let us show the result for $k = 1$. For all $\alpha \in (\alpha^{**}_1, \alpha^*_1)$ since $d''(\alpha) < 0$ and $f''(\alpha) = \alpha d''(\alpha) + 2d'(\alpha) < 0$, we deduce that f is a strictly concave f. As a consequence, there exists $\overline{\alpha}_1 \in (\alpha^{**}_1, \alpha^*_1)$ which maximizes f. □

For each k, since $\overline{\alpha}_k \in (\alpha^{**}_k, \alpha^*_k)$, the local maximum $\overline{\alpha}_k$ of the function f does not coincide with the local maximum α^*_k of S. By construction, setting the threshold parameter $\alpha = \overline{\alpha}_1$ yields to

$$\alpha_{\mathrm{D}} < \alpha_{\mathrm{P}} < \overline{\alpha}_1\ .$$

At the numerical level, since d is a piecewise constant function, f is a linear piecewise one (with a ”sawtooth” profile as displayed in Fig. 18(b)) which makes difficult to catch the first local maximum. Nevertheless, keeping in mind that the mean value of S can be, in some situation, a relevant threshold (see Sect. 3.3), it is easy to compute the global maximum value of f in the interval $(0, S_m]$. Therefore, we propose the following thresholding method

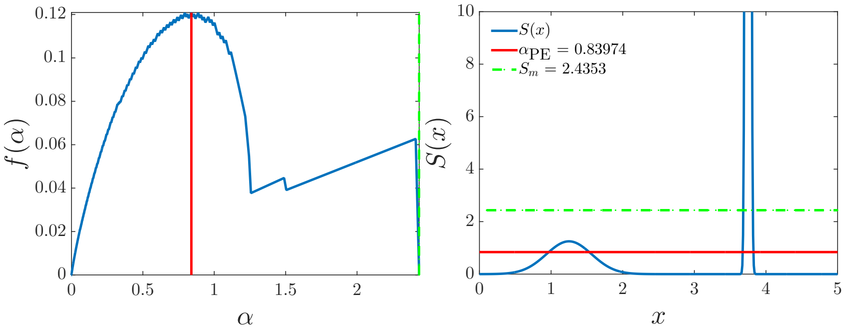
$$\alpha_{\mathrm{PE}} \text{ such that } f(\alpha_{\mathrm{PE}}) = \max_{0 < \alpha \leqslant S_m} f(\alpha)\ . \tag{11}$$

With the above selection method, the threshold α_{PE} is not necessarily the closest point to the smallest local maximum of S. This method is motivated by the following remarks.

- First, in the presence of discontinuous solutions (as displayed in Fig. 16(b)), the distribution d sharply decreases so that the smallest local maxima are in general lower than the mean value S_m. As a consequence, the use of the mean value S_m is not relevant and by construction the threshold selected is α_{PE} such that $f(\alpha_{\text{PE}}) = \max_{0<\alpha<S_m} f(\alpha)$ as illustrated in Fig. 16(a).

- Second, in the case of smooth flow, the mean value is in general a relevant mesh refinement threshold as displayed in Fig. 17(a) and 17(b).
- Third, in some cases, especially in a multi-dimensional problems, a small wave can be reflected in every directions and therefore the domain can be unnecessarily everywhere refined if we focus on the smallest local maximum.

Moreover in view of the above remarks, *this method is less sensitive with respect to β (see Figs. 18(b), 18(c) and 18(d)), less expensive and yields to less refined cells than the Dannenhoffer or the Powell approach.*



(a) $f(\alpha)$ (b) $S(x)$

Fig. 16 The function f for the mesh refinement criterion $S(x) = 200\exp(-1000(x-1.25)^2) + 1.25\exp(-5(x-3.75)^2)$ representing a shock type solution

In practice, according to the finite volume approximation defined in Sect. 2, a mesh refinement criterion S_k^n is computed on each cell at time t_n and compared to the computed mesh refinement threshold

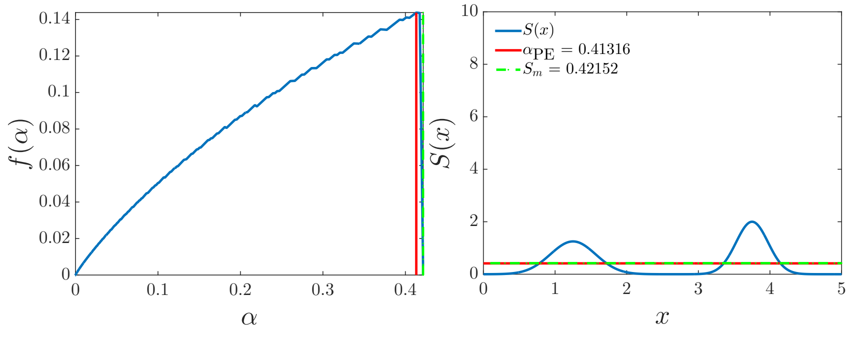
$$\alpha_{\text{PE}} \text{ such that } f(\alpha_{\text{PE}}) = \max_{0<j\leqslant M+1} f(\alpha_j)\,.$$

As a consequence, for each cell C_k:

- if $S_k^n > \alpha_{\text{PE}}$, the mesh is refined and split,
- if $S_k^n < \alpha_{\text{PE}}$ the mesh is coarsened.

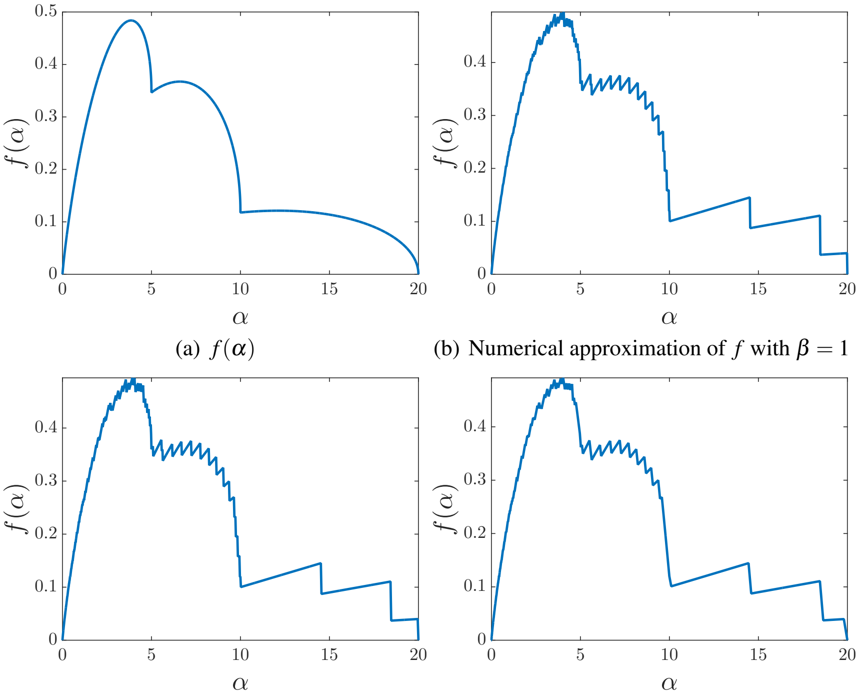
4.2.3 Numerical application using α_{PE} as an automatic mesh refinement threshold.

To illustrate the robustness of our method, we consider again the test problem described in Sect. 3.1, .i.e., the Riemann problem for Eqs. (4) with the Riemann data



(a) $f(\alpha)$ (b) $S(x)$

Fig. 17 The function f for the mesh refinement criterion $S(x) = 2\exp(-10(x-1.25)^2) + 1.25\exp(-5(x-3.75)^2)$ representing a smooth flow



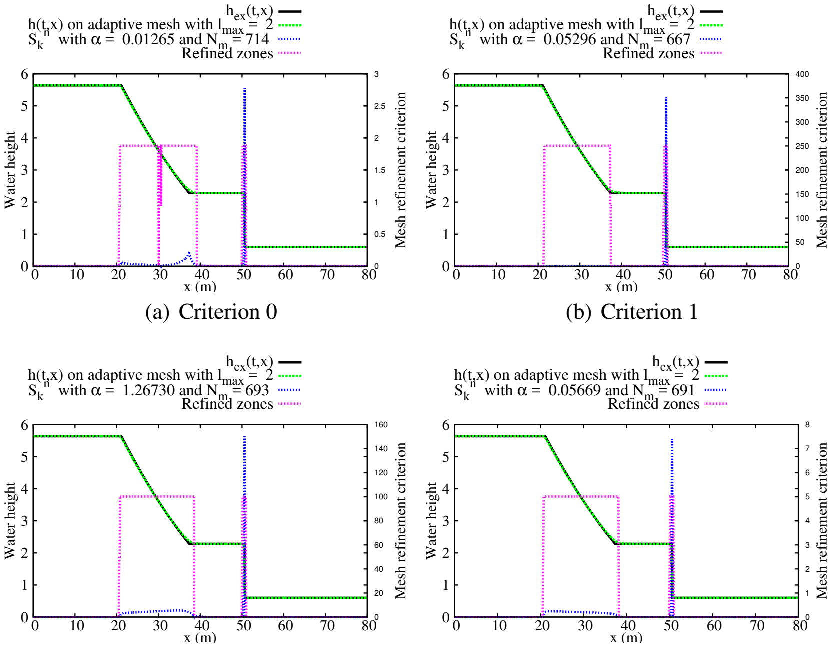
(a) $f(\alpha)$ (b) Numerical approximation of f with $\beta = 1$

(c) Numerical approximation of f with $\beta = 2$ (d) Numerical approximation of f with $\beta = 10$

Fig. 18 The modified distribution function and its numerical approximation for the mesh refinement criterion $S(x) = L\exp(-10(x-L/4)^2) + L/2\exp(-5(x-L/2)^2) + 2L\exp(-200(x-3L/4)^2)$ with $L = 10$

(6). As done before, each numerical experiment start initially with 500 mesh points. *The objective is to show that, for any given mesh refinement criterion, the threshold is adapted automatically to refine in the area of interests.* For this purpose, we consider the criterion 0–3 and we omit the criterion 4 since it provide almost the same results of the criterion 3. For each numerical experiment, we have considered $\beta = 2$ and $M = 1000$ to calculate the threshold parameter (11).

In Fig. 19, we display the water height at time $t = 2$ s for the selected mesh refinement criterion. In comparison with the numerical results obtained in Sect. 3.3, the threshold parameter (11), see Fig. 22, is now automatically calibrated at each time step depending on the distribution function (8), see Fig. 20. More precisely, independently of the mesh refinement criterion, all the area of interests (i.e., the rarefaction and the shock wave) are well-refined yielding to almost the same order of accuracy as displayed in Figs. 19(a), 19(b), 19(c) and 19(d). This test shows that even if the mesh refinement criterion is not a relevant *a priori* error estimate, adjusting the threshold parameter in a ”good manner” lead to almost the same results obtained with the exact local error as a mesh refinement criterion (see Figs. 19(a) and Remark 1).



(a) Criterion 0

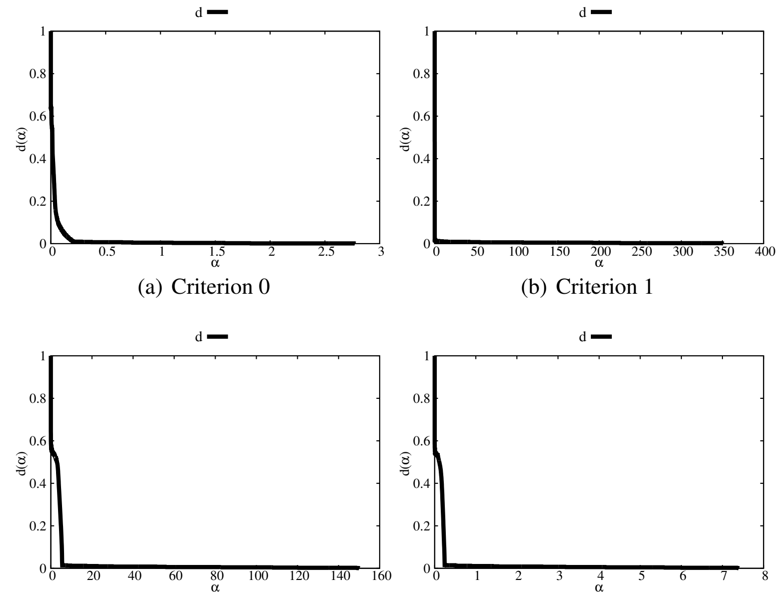
(b) Criterion 1

(c) Criterion 2

(d) Criterion 3

Fig. 19 Numerical results for the water height at time $t = 2$ s

The mesh refinement criteria being of several type, theirs behaviors are different. As a consequence, the shape of the distribution functions are also not similar as displayed in Fig. 20. Therefore, the functions f are also almost different as observed in Fig. 21. In particular, it explains the differences observed in the evolution in time of the threshold, see Fig. 22. It is automatically adapted according to the mesh refinement criterion used. As pointed out in the conclusion of Sect. 3.3, the computed threshold for each criteria is now sufficient to yield satisfactory (non oscillating) results. Even for the criterion 1 which is of shock type, the smooth region are very well-refined. This is a nice feature and a good example to assert that the present thresholding method is efficient. Indeed, for the criterion 1, the more the mesh is refined, the more the numerical density is well-computed, the larger is the global maximum (corresponding to the shock) and the less the mean value is relevant. This is why after the time $t > 0.5$, the selected threshold is not the mean value S_m but α_{PE}, see Fig. 21(b). For the criterion 2 and 3, the threshold parameter is almost of the same order of the mean value. It means that for this criterion, except few times (see Figs. 22(c) and 22(d)), the mean value as a mesh refinement threshold is relevant.



(a) Criterion 0

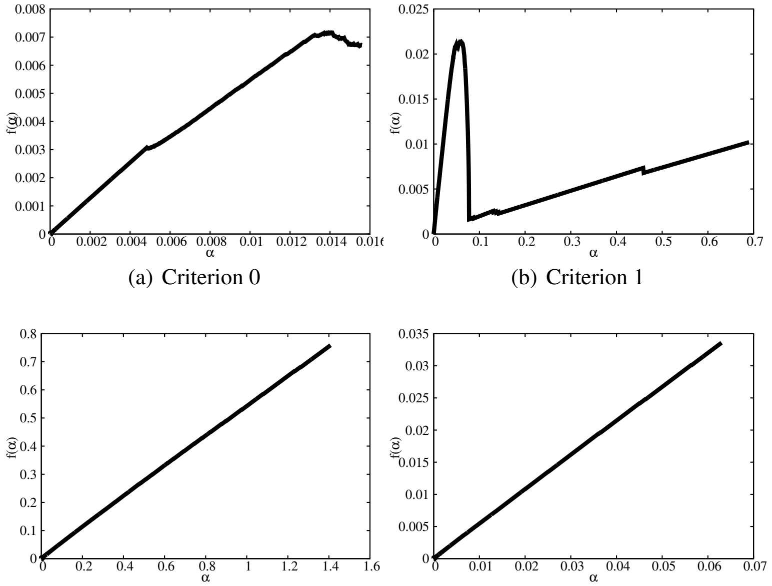
(b) Criterion 1

(c) Criterion 2

(d) Criterion 3

Fig. 20 Distribution function at time $t = 2$ s

To conclude, let us now present the new rate of convergence (to compare with those summarized in Table 2). As done in Sect. 3.3, the numerical order of convergence is computed with respect to the mean number of mesh points. The obtained



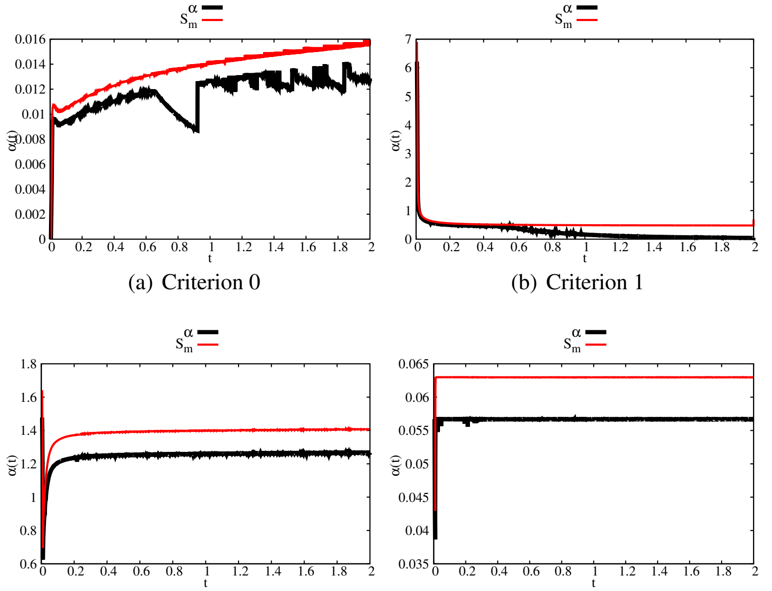
(a) Criterion 0

(b) Criterion 1

(c) Criterion 2

(d) Criterion 3

Fig. 21 Function f at time $t = 2$ s



(a) Criterion 0

(b) Criterion 1

(c) Criterion 2

(d) Criterion 3

Fig. 22 Time evolution of the threshold parameter and the mean value S_m

results are summarized in Table 3. The obtained rates are now better than the previous one, passing almost from an order 1.5 to 2. This confirms that a well-chosen automatic mesh refinement threshold can yield to accurate results even if the chosen criterion is not well-linked to the local discretization error.

Criterion	$\|h-h_{ex}\|_1$ *vs* N_m	$\|u-u_{ex}\|_1$ *vs* N_m
0	2.0477	2.1261
1	2.0363	2.0912
2	2.1175	2.1864
3	2.1067	2.1746
4	2.1070	2.1751

Table 3 Convergence rate of the L^1 discretization error.

5 Conclusion

In this paper, we have carried out a complete numerical study of some mesh refinement criteria for hyperbolic equations solved on adaptive grid. We have emphasized that despite of the lack of accurate error estimate for non trivial hyperbolic system, any ”reasonable” mesh refinement criterion may lead to accurate results if the threshold parameter is well-chosen. This mainly holds because the support of the local error are almost the same. Unfortunately, in the most of the applications, this parameter has to be calibrated. Thus, we have performed a numerical study of the behavior of some criteria with respect to the number of mesh points and its sensitivity with respect to the mesh refinement threshold parameter. In particular, we have pointed out that a such parameter should be time-dependent and ideally chosen to balance the accuracy and the computational cost by avoiding unnecessary refinement and oscillating solutions.

In order to propose an efficient method to select automatically the threshold parameter, independently of the mesh refinement criterion, we have first proposed a review of some popular thresholding methods including the mean method. It is widely used but it can fails to detect smooth region while the filtering approach is less subject to this phenomenon but still requires a parameter to calibrate. The control of the mesh size growth approach may be efficient but the mesh growing parameter must be calibrated. The local maxima method can be an alternative if high order schemes are used but it has also the drawback to tune a threshold parameter. Finally, the last method reviewed is based on the cumulative distribution function. It is the most relevant method in the detection of smooth as well as discontinuous flows. However, all the methods found in the literature are based on the numerical approximation of first or second order derivative of this function which can be quite costly and overall it depends on a tunable parameter. This distribution function was introduced by Dannenhoffer [9]. Based on the decreasing rearrangement function of the mesh refinement criterion, we have proposed a new parameterless easy to compute method. It corresponds to the continuous version of the Dannenhoffer [9] distribution function for which we have obtained several theoretical results. The proposed method is only based on the distribution function and allows to set automatically, independently of the mesh refinement criterion type and without any parameter, the threshold yielding to a well-refined solution. We have shown its robustness through a one dimensional test case. We propose as the second part of this paper [29] to show its efficiency for a large variety of real life test problems such as the propagation of tsunamis.

Finally, the obtained results are independent of the solved equations and the numerical scheme. Therefore, it can be easily used in the context of other numerical methods.

Acknowledgements This work is partially supported the Project MTM2011-29306-C01-01 from the MICINN (Spain) and the French national research project TANDEM (Tsunamis in the Atlantic and the English channel : definition of the effects through numerical modeling), the French government (Projets Investissement d'Avenir, agreement reference number ANR-11-RSNR-0023-01). The authors wish to thank J-J. Alibert for its valuable remarks and comments.

Appendix: Flux consistency error

For the sake of simplicity, let us consider the one dimensional homogeneous Saint-Venant system (1) for $x \in \mathbb{R}$, $t > 0$

$$\partial_t \mathbf{w} + \partial_x \mathbf{f}(\mathbf{w}) = 0 \tag{12}$$

where the unknown state $\mathbf{w}$ and the flux are

$$\mathbf{w}(x,t) = \begin{pmatrix} h \\ hu \end{pmatrix}, \quad \mathbf{f}(\mathbf{w}) = \begin{pmatrix} hu \\ hu^2 + \frac{g}{2}h^2 \end{pmatrix} .$$

Its first order finite volume discretization reads

$$\mathbf{w}_k^{n+1} = \mathbf{w}_k^n - \frac{\delta t_n}{\delta x}\left(\mathbf{f}_{i+1/2}^n - \mathbf{f}_{i-1/2}^n\right) \tag{13}$$

where $\mathbf{w}_k^n$ and $\mathbf{f}_{i+1/2}^n$ are defined by

$$\mathbf{w}_k^n \approx \frac{1}{\delta x}\int_{C_k} \mathbf{w}(x,t_n)\, dx, \quad \mathbf{f}_{i+1/2}^n \approx \frac{1}{\delta t_n}\int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{w}(x_{k+1/2},t))\, dt\ .$$

One can construct a mesh refinement criterion as an *a posteriori* error estimates by computing the consistency (truncation) error in the sense of finite difference scheme. Unlike this approach which yields to second order derivatives (i.e. heavy to compute numerically), we propose to write the flux consistency error by considering the consistency error at the interface.

Indeed, let us assume that $\mathbf{w}$ is a smooth solution of Eqs. (12) and suppose that for all k,

$$\mathbf{w}_k^n = \frac{1}{\delta x}\int_{C_k} \mathbf{w}(x,t_n)\, dx\,.$$

Then integrating the system of conservation laws (12) satisfied by $\mathbf{w}(x,t)$ with respect to x and t over $C_k \times (t_n,t_{n+1})$ and dividing the results by δx, we obtain:

$$\frac{1}{\delta x}\int_{C_k} \mathbf{w}(x,t_{n+1})\, dx = \frac{1}{\delta x}\int_{C_k} \mathbf{w}(x,t_n)\, dx - \frac{\delta t_n}{\delta x}\left(\mathscr{F}_{i+1/2}^n - \mathscr{F}_{i-1/2}^n\right) \tag{14}$$

where $\mathscr{F}_{i+1/2}$ stands for the exact flux. Therefore, subtracting (13) to (14), we get

$$\mathbf{w}_k^{n+1} = \frac{1}{\delta x}\int_{C_k} \mathbf{w}(x,t_{n+1})\, dx - \frac{\delta t_n}{\delta x}\left(\varepsilon_{k+1/2}^n - \varepsilon_{k-1/2}^n\right) \tag{15}$$

where

$$\varepsilon_{i+1/2} = \mathscr{F}_{i+1/2} - \boldsymbol{F}_{i+1/2}\,.$$

Equation (15) is nothing else than the definition of the consistency of system of conservation in the finite volume sense. Indeed, we say that Scheme (13) is consistent with System (12), if for all k

$$\varepsilon_{k+1/2} \to 0 \text{ as } \delta t_n \text{ and } \delta x \text{ go to } 0\,.$$

The Taylor expansion of $\mathbf{w}$ in the neighborhood of $(x_{k+1/2},t_n)$, through the above term, measures the local flux errors. Therefore, if we consider the Godunov solver, i.e., $\mathbf{f}_{i+1/2}^n = \mathbf{f}(\mathbf{w}_{k+1/2}^*(t_n))$ (where $\mathbf{w}_{k+1/2}^*(t_n) = \mathbf{w}^*(0;\mathbf{w}_k^n,\mathbf{w}_k^{n+1})$ is the exact solution of the local Riemann problem with the Riemann data $(\mathbf{w}_k^n,\mathbf{w}_{k+1}^n)$) one can write the flux error as follows:

$$\varepsilon_{k+1/2} = \underbrace{\frac{1}{\delta t_n}\int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{w}(x_{k+1/2},t)) - \mathbf{f}(\mathbf{w}(x_{k+1/2},t_n))\, dt}_{T_1}$$

$$+\underbrace{\frac{1}{\delta t_n}\int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{w}(x_{k+1/2},t_n)) - \mathbf{f}(\mathbf{w}_{k+1/2}^*(t_n))\, dt}_{T_2}\,.$$

Computation of T_1:

Let us define

$$G(s) = \int_{t_n}^{s} \mathbf{f}(\mathbf{w}(x_{k+1/2},s))\, ds\,.$$

Then,

$$\begin{aligned} G(t_{n+1}) &= G(t_n) + G'(t_n)\delta t_n + G''(t_n)\tfrac{\delta t_n^2}{2} + O(\delta t_n^3) \\ &= F(\mathbf{w}(x_{k+1/2},t_n))\delta t_n + \partial_t F(\mathbf{w}(x_{k+1/2},t_n))\tfrac{\delta t_n^2}{2} + O(\delta t_n^3) \end{aligned}$$

and therefore we get

$$\begin{aligned} T_1 &= \partial_t F(\mathbf{w}(x_{k+1/2},t_n))\tfrac{\delta t_n}{2} + O(\delta t_n^2) \\ &= -A(\mathbf{w}(x_{k+1/2},t_n))^2 \partial_x \mathbf{w}(x_{k+1/2},t_n)\tfrac{\delta t_n}{2} + O(\delta t_n^2) \end{aligned} \qquad (16)$$

where $A(\mathbf{w}) = \frac{\partial \mathbf{f}(\mathbf{w})}{\partial \mathbf{w}} = \begin{pmatrix} 0 & 1 \\ -u^2 + g\frac{h^2}{2} & 2u \end{pmatrix}$ is the convection matrix of System (12).

Computation of T_2:

Since the terms in T_2 do not depend on t, one can write

$$T_2 = \mathbf{f}(\mathbf{w}(x_{k+1/2},t_n)) - \mathbf{f}(\mathbf{w}^*_{k+1/2}(t_n)) \; .$$

and therefore, we get

$$T_2 = A(\mathbf{w}(x_{k+1/2},t_n))\partial_x \mathbf{w}(x_{k+1/2},t_n)\delta x + O(\delta x^2) \; . \qquad (17)$$

Consistency (flux consistency) error:

Gathering (16) and (17), we obtain

$$\varepsilon_{k+1/2} = \delta x A(\mathbf{w}(x_{k+1/2},t_n))\left(I_d - A(\mathbf{w}(x_{k+1/2},t_n))\frac{\delta t_n}{2\delta x}\right)\partial_x \mathbf{w}(x_{k+1/2},t_n)\delta x + O(\delta x^2)$$

where I_d is the 2×2 identity matrix. Then, introducing the Riemann Invariant $\boldsymbol{I}(h,u)$ of component $(I_1(h,u) = u + 2\sqrt{gh}\ ,\ I_2(h,u) = u - 2\sqrt{gh})$ and $P = \begin{pmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{pmatrix}$ the matrix composed by the eigenvectors, $\lambda_1(h,u) = u - \sqrt{gh} < \lambda_2(h,u) = u + \sqrt{gh}$, such that

$$\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} = P^{-1}AP \; ,$$

one can write $\mathbf{w}(x_{k+1/2},t_n)$ as

$$\mathbf{w} = \mathbf{w}(\boldsymbol{I}) \; .$$

Therefore, we deduce, for $h > 0$,

$$\begin{aligned}\varepsilon_{k+1/2} &= \delta x\, P \begin{pmatrix} \lambda_1(1-\lambda_1 \frac{\delta t_n}{2\delta x})\partial_x I_1(x_{k+1/2},t_n)) \\ \lambda_2(1-\lambda_2 \frac{\delta t_n}{2\delta x})\partial_x I_2(x_{k+1/2},t_n)) \end{pmatrix} + O(\delta x^2, \delta t_n^2) \\ &= \delta x\, P\boldsymbol{V} + O(\delta x^2, \delta t_n^2)\ .\end{aligned}$$

Finally, noting the induced norm $|||A||| := \max_{\|x\|=1} \|Ax\|$, we deduce the following consistency error estimate

$$\left\|\varepsilon_{k+1/2}\right\| \leqslant \delta x\, |||P|||\, \|V\| \tag{18}$$

where we have used $\|\cdot\| = \|\cdot\|_\infty$, i.e.

$$|||P|||_\infty = 2\max(1,|u|) \text{ and } \|V\|_\infty = \max_{i=1,2} \left| \left(\lambda_i \left(1-\lambda_i \frac{\delta t_n}{2\delta x}\right) \partial_x I_i(x_{k+1/2},t_n)) \right) \right| \ .$$

References

1. Aftosmis, M.: Upwind method for simulation of viscous flow on adaptively refined meshes. AIAA journal **32**(2), 268–277 (1994)
2. Aftosmis, M., Berger, M.: Multilevel error estimation and adaptive h-refinement for cartesian meshes with embedded boundaries. AIAA paper **863**, 2002 (2002)
3. Altazin, T., Ersoy, M., Golay, F., Sous, D., Yushchenko, L.: Numerical investigation of bb-amr scheme using entropy production as refinement criterion. Accepted in International Journal of Computational Fluid Dynamics (2016)
4. Berger, M., Colella, P.: Local adaptive mesh refinement for shock hydrodynamics. Journal of Computational Physics **82**(1), 64 – 84 (1989). DOI http://dx.doi.org/10.1016/0021-9991(89)90035-1. URL http://www.sciencedirect.com/science/article/pii/0021999189900351
5. Berger, M., Oliger, J.: Adaptive mesh refinement for hyperbolic partial differential equations. J. Comp. Phys. **53**(3), 484–512 (1984)
6. Berger, M.J., LeVeque, R.J.: Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. SIAM Journal on Numerical Analysis **35**(6), 2298–2316 (1998)
7. Biswas, R., Strawn, R.C.: A new procedure for dynamic adaption of three-dimensional unstructured grids. Applied Numerical Mathematics **13**(6), 437–452 (1994)
8. Dannenhoffer, J.F.: Computationally efficient solution based mesh adaptation: a new flag strategy. Ph.D. thesis, Aerospace Engineering at the Technical University Delft, faculty Aerospace Engineering, department aerodynamics (1987)
9. Dannenhoffer, J.F.: Grid adaptation for complex two-dimensional transonic flows. Ph.D. thesis, Massachusetts Institute of Technology (1987)
10. Daubechies, I., et al.: Ten lectures on wavelets, vol. 61. SIAM (1992)
11. Ersoy, M.: A simple and efficient new algorithm to increase the regularity and vanishing moments of biorthogonal wavelets. preprint (2008)
12. Ersoy, M., Golay, F., Yushchenko, L.: Adaptive multiscale scheme based on numerical density of entropy production for conservation laws. Cent. Eur. J. Math. **11**(8), 1392–1415 (2013). DOI 10.2478/s11533-013-0252-6. URL http://dx.doi.org/10.2478/s11533-013-0252-6
13. Eymard, R., T., G., Herbin, R.: Finite volume methods. In: Handbook of numerical analysis, Vol. VII, Handb. Numer. Anal., VII, pp. 713–1020. North-Holland, Amsterdam (2000)
14. Faccanoni, G., Mangeney, A.: Exact solution for granular flows. International Journal for Numerical and Analytical Methods in Geomechanics (2012). DOI 10.1002/nag.2124. URL https://hal.inria.fr/hal-00776614

15. Garcia, A.L., Bell, J.B., Crutchfield, W.Y., Alder, B.J.: Adaptive mesh and algorithm refinement using direct simulation monte carlo. Journal of computational Physics **154**(1), 134–155 (1999)
16. Godlewski, E., Raviart, P.: Numerical approximation of hyperbolic systems of conservation laws, *Applied Mathematical Sciences*, vol. 118. Springer-Verlag, New York (1996)
17. Golay, F., Ersoy, M., Yushchenko, L., Sous, D.: Block-based adaptive mesh refinement scheme using numerical density of entropy production for three-dimensional two-fluid flows. International Journal of Computational Fluid Dynamics **29**(1), 67–81 (2015)
18. Harten, A., Hyman, J.M.: Self adjusting grid methods for one-dimensional hyperbolic conservation laws. Journal of computational Physics **50**(2), 235–269 (1983)
19. Jasak, H.: Error analysis and estimation for the finite volume method with applications to fluid flows. Ph.D. thesis, Imperial College London (University of London) (1996)
20. Jasak, H., Gosman, A.: Automatic resolution control for the finite-volume method, part 2: Adaptive mesh refinement and coarsening. Numerical Heat Transfer: Part B: Fundamentals **38**(3), 257–271 (2000)
21. Kallinderis, Y.G., Baron, J.R.: Adaptation methods for a new navier-stokes algorithm. AIAA journal **27**(1), 37–43 (1989)
22. Kirk, B.S., Peterson, J.W., Stogner, R.H., Carey, G.F.: libmesh: a c++ library for parallel adaptive mesh refinement/coarsening simulations. Engineering with Computers **22**(3-4), 237–254 (2006)
23. Leonard, S., Terracol, M., Sagaut, P.: A wavelet-based adaptive mesh refinement criterion for large-eddy simulation. Journal of Turbulence **7**(64) (2006)
24. LeVeque, R.J., George, D.L., Berger, M.J.: Tsunami modelling with adaptively refined finite volume methods. Acta Numerica **20**, 211–289 (2011)
25. Lucier, B.J.: A moving mesh numerical method for hyperbolic conservation laws. Mathematics of computation **46**(173), 59–69 (1986)
26. Mungkasi, S.: A study of well-balanced finite volume methods and refinement indicators for the shallow water equations. Bulletin of the Australian Mathematical Society **88**(2), 351–352 (2013)
27. Mungkasi, S., Roberts, S.G.: Numerical entropy production for shallow water flows. In: W. McLean, A.J. Roberts (eds.) Proceedings of the 15th Biennial Computational Techniques and Applications Conference, CTAC-2010, *ANZIAM J.*, vol. 52, pp. C1–C17 (2011). http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/3786 [April 9, 2011]
28. Perthame, B., Simeoni, C.: A kinetic scheme for the Saint-Venant system with a source term. Calcolo **38**(4), 201–231 (2001)
29. Pons, K., Ersoy, M., Golay, F., Marcer, R.: Adaptive mesh refinement method. Part 2: Application to tsunami propagation (2016)
30. Powell, K.G., Murman, E.M.: An embedded mesh procedure for leading-edge vortex flows. In: NASA, Langley Research Center, Transonic Symposium: Theory, Application, and Experiment, vol. 1 (1989)
31. Schmidt, W., Schulz, J., Iapichino, L., Vazza, F., Almgren, A.: Influence of adaptive mesh refinement and the hydro solver on shear-induced mass stripping in a minor-merger scenario. Astronomy and Computing **9**, 49–63 (2015)
32. Simeoni, C.: Remarks on the consistency of upwind source at interface schemes on nonuniform grids. SIAM J. Sci. Comput. **48**(1), 333–338 (2011)
33. Stockie, J.M., Mackenzie, J.A., Russell, R.D.: A moving mesh method for one-dimensional hyperbolic conservation laws. SIAM Journal on Scientific Computing **22**(5), 1791–1813 (2001)
34. Syrakos, A., Efthimiou, G., Bartzis, J.G., Goulas, A.: Numerical experiments on the efficiency of local grid refinement based on truncation error estimates. Journal of Computational Physics **231**(20), 6725–6753 (2012)
35. Syrakos, A., Goulas, A.: Finite volume adaptive solutions using simple as smoother. International journal for numerical methods in fluids **52**(11), 1215–1245 (2006)
36. Tang, H., Tang, T.: Adaptive mesh methods for one-and two-dimensional hyperbolic conservation laws. SIAM Journal on Numerical Analysis **41**(2), 487–515 (2003)

37. Toro, E.F.: Riemann solvers and numerical methods for fluid dynamics, third edn. Springer-Verlag, Berlin (2009). DOI 10.1007/b79761. URL http://dx.doi.org/10.1007/b79761
38. Trangenstein, J.A.: Numerical solution of hyperbolic partial differential equations. Cambridge University Press (2009)
39. Verfürth, R.: A posteriori error estimation and adaptive mesh-refinement techniques. Journal of Computational and Applied Mathematics **50**(1), 67–83 (1994)
40. Warren, G.P., Anderson, W.K., Thomas, J.L., Krist, S.L.: Grid convergence for adaptive methods. AIAA paper **1592**, 1991 (1991)