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 In this work, we address the construction of an automatic selection of a suitable mesh refinement threshold in adaptive mesh refinement methods. Usually, the mesh refinement threshold is a problem dependent parameter whose choice might be a critical weakness of the mesh adaptation method. The automatic selection of a suitable mesh refinement threshold is constructed from the decreasing rearrangement function of the mesh refinement criterion. It allows to detect almost all relevant regions to refine without any hand-calibration as usually done for mesh adaptation algorithms. The efficiency of the method is illustrated for a one-dimensional test case. More complex one-dimensional and multi-dimensional test cases can be found in the second part of this paper.

1 Introduction

A major issue of many modelling challenges is to solve accurately processes over very large ranges in spatial scale. Solving accurately those processes with high resolution inexorably leads to heavy computational time [21, 22, 2].

In principle, Adaptive Mesh Refinement (AMR) [4] method allow to solve in a reasonable CPU time equations adjusting the computational effort locally to maintain a quite uniform level of accuracy. It relies on macro cells which can be refined (and then possibly coarsened).

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

The zones where the mesh needs to be refined are determined thanks to a suitable *mesh refinement criterion* and a *mesh refinement threshold*. Depending on the application, equations and numerical methods, a variety of criteria might be used based on an error estimation procedure or a feature detection technique. Basically, the zones where a given threshold is exceeded are refined. Usually, *the mesh refinement threshold is a test case dependent parameter*. If this parameter is set too small, the results will be accurate but the computational cost will be expensive. On the contrary, if this threshold is set too large, pertinent regions are not detected and the mesh adaptation algorithm will not be efficient. Generally speaking, if this parameter is not well-calibrated, it may yield to unnecessary refined cells, and sometimes the numerical solution can develop spurious oscillations. Therefore, the way to choose the mesh refinement threshold is very important.

In this paper, we propose a method to set automatically the threshold parameter to detect almost all relevant regions to refine. The *construction of the automatic selection of the threshold is based on the decreasing rearrangement function* of the mesh refinement criterion. The decreasing rearrangement provides a description of the mesh refinement criterion in terms of the local maxima which are sorted from the smallest to the largest. Thus, in principle, this method allows to set the threshold automatically small enough to detect the most relevant cells to refine. In practice, we are able to reach a good balance between the accuracy and the computational time without any hand-calibration of the mesh refinement threshold as usually performed in mesh adaptation algorithms.

In the sequel, we focus on h-AMR method. As a non-exhaustive list of references, one can refer to [4, 3, 30, 10, 16, 8].

The paper is organised as follows. The Section 2 is devoted to the presentation of the finite volume method and the principle of the h-AMR approach. We also present some mesh refinement criteria used in this paper. In Sect. 3, we discuss about advantages and drawbacks of some classical thresholding methods. Then, we introduce a new method based on the decreasing rearrangement of the mesh refinement criterion to construct a suitable automatic threshold. Finally, for a given one dimensional Riemann problem for the non-linear shallow water equations, we illustrate the efficiency of the method to detect simple waves without hand-calibration. In the second part of this paper [20], more complex multi-dimensional test cases are considered.

2 Finite volume approximation and h-AMR algorithm

For the sake of simplicity, we consider here the one-dimensional case (we refer to the second part of this paper [20] for the multi-dimensional case and applications). This section summarises the main features of the finite volume approximation and the adaptive mesh refinement method for a general non-linear hyperbolic system

$$\begin{cases} \partial_t \mathbf{w} + \partial_x \mathbf{f}(\mathbf{w}) = 0 \\ \qquad \mathbf{w}(x,0) = \mathbf{w}_0(x) \end{cases} \tag{1}$$

where $x \in \mathbb{R}$ is the coordinate, $t > 0$ is the time, $\mathbf{w}(x,t) \in \mathbb{R}^d$ is the unknown vector with $d \geq 1$, $\mathbf{f}(\mathbf{w}) \in \mathbb{R}^d$ is the flux at a space-time point (x,t) and $\mathbf{w}_0$ is the initial data.

In particular, in this work, we will consider the Saint-Venant equations as a prototype model with $\mathbf{w} = \begin{pmatrix} h \\ hu \end{pmatrix}$ and $\mathbf{f}(\mathbf{w}) = \begin{pmatrix} hu \\ hu^2 + g\frac{h^2}{2} \end{pmatrix}$. Here the unknowns $h(x,t) \geq 0$ and $u(x,t) \in \mathbb{R}$ stand respectively for the water height and the depth-averaged velocity of the water at a space-time point (x,t) and $g \approx 9.81$ m/s^2 is the gravitational constant. To illustrate the method of automatic thresholding proposed in Sect. 3.2, we will consider the Riemann problem (c.f. Sect. 3.3):

$$\mathbf{w}_0(x) = \begin{cases} \mathbf{w}_l & \text{if } x < 0 \\ \mathbf{w}_r & \text{if } x > 0 \end{cases} \tag{2}$$

Interested readers can found multi-dimensional test cases in the second part of this paper [20].

2.1 First order finite volume approximation

We suppose that the computational domain $\Omega \subset \mathbb{R}$ is split into a set of cells $C_k = \left]x_{k-1/2}, x_{k+1/2}\right[$ of size δx_k with $x_{k\pm1/2} = x_k \pm \delta x_k/2$ such that $\Omega = \cup_k C_k$. We assume that $k \in \mathbb{Z}$ to simplify the presentation. We define the discrete time 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

$$\boldsymbol{w}_k(t) \simeq \frac{1}{\delta x_k} \int_{C_k} \boldsymbol{w}(x,t)\, dx.$$

the approximation of the mean value of the unknown $\boldsymbol{w}(x,t)$ on C_k at time t_n, integrating (1) over $C_k \times (t_n, t_{n+1})$, and dividing by δx_k, we obtain the first order finite volume approximation of Eqs. (1) (see for instance [11, 29, 9]):

$$\mathbf{w}_k^{n+1} = \mathbf{w}_k^n - \frac{\delta t_n}{\delta x_k} \left(\boldsymbol{F}_{k+1/2}(\mathbf{w}_k^n, \mathbf{w}_{k+1}^n) - \boldsymbol{F}_{k-1/2}(\mathbf{w}_{k-1}^n, \mathbf{w}_k^n) \right)$$

where $\boldsymbol{F}_{k+1/2}(\mathbf{w}_k^n, \mathbf{w}_{k+1}^n)$ is the approximation of the flux $\boldsymbol{F}_{k+1/2}(\mathbf{w}_k^n, \mathbf{w}_{k+1}^n) \approx \frac{1}{\delta t_n} \int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{w}(x_{k+1/2}, t))\, dt$. $\boldsymbol{F}_{k+1/2}(\mathbf{w}_k^n, \mathbf{w}_{k+1}^n)$ is defined via any three-point scheme (see for instance [19, 28, 29]). In this paper, we consider a first order space-time discretisation using the Godunov scheme, *i.e.* the numerical flux $\boldsymbol{F}_{k+1/2}$ is computed with the exact solution of the 1D Riemann problem at the interface $x_{k+1/2}$ with the left state $\mathbf{w}_k^n$ and the right state $\mathbf{w}_{k+1}^n$ (for further details see, for instance, [29]).

2.2 Principle of h-AMR method

For the convenience of the reader, we recall the procedure for the first order scheme in the one dimensional case. We refer to [8] for the second order scheme and to [2, 20] for the multi-dimensional case. Let us note that in h-AMR method the mesh evolves in time, thus each cell C_k should be written C_k^n. To simplify the notation, we omit here the time dependency.

In order to reduce the time necessary to manage the refinement, we use "macro cells" which could be refined by generating hierarchical grids. Each cell can be split in two sub-cells. We thus produce a dyadic cells graph, whose numbering (in basis 2) allows a quick computing scan to determine the adjacent cells. We use the following notations: let k_b be the index associated to the macro cell numbered k with b a binary number which contains the hierarchical information of a sub-cell. In particular, the level of a sub-cell C_{k_b} is defined as the length$(b) - 1$. By convention, the coarsest cells are of level $l = 1$, while the finest cells are of level $l_{\max} \in \mathbb{N}^*$. For instance, a macro cell C_{k_0} of level 1 will be split into two sub-cells $C_{k_{00}}$ and $C_{k_{01}}$ of level 2. A mesh refinement example is proposed in Fig. 1.

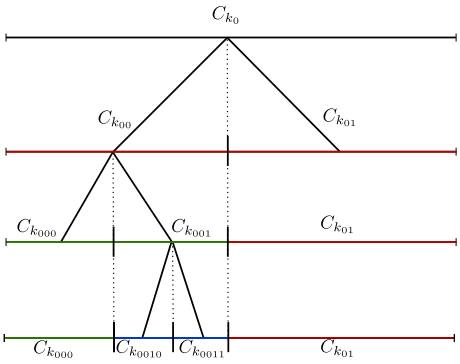


Fig. 1 Example of hierarchical dyadic tree.

The mesh refinement procedure can be simply expressed as follows. We introduce a non negative mesh refinement criterion $S_{k_b}^n$ computed on each cell C_{k_b} at time t_n and compared, for instance, to the average S_m

$$S_m = \frac{1}{|\Omega|} \sum_{k_b} S_{k_b}^n \tag{3}$$

This approach is the so-called mean method (*c.f.* Sect. 3.1 for more details and a presentation of some other approaches).

We then define two coefficients $0 < \beta_{\min} \leq \beta_{\max}$, which determine the ratio of the cells to be refined or coarsened. Thus, for each cell C_{k_b}:

- if $S_{k_b}^n > \alpha_{\max} = S_m \beta_{\max}$, the cell is refined and split into two sub-cells $C_{k_{b0}}$ and $C_{k_{b1}}$,
- if $S_{k_{b0}}^n < \alpha_{\min} = S_m \beta_{\min}$ and $S_{k_{b1}}^n < \alpha_{\min}$, the cell is coarsened into a cell C_{k_b}.

Therefore, the adaptive algorithm stops when the level l of a cell C_{k_b} reaches the maximum level $l_{\max}$.

Remark 1. The threshold parameters $\beta_{\min}$ and $\beta_{\max}$ allow to set a percentage of mesh refinement and mesh coarsening with respect to the quantity S_m. The grid is not refined or coarsened between $\alpha_{\min}$ and $\alpha_{\max}$. It is not surprising that these settings will deteriorate or improve the accuracy of the numerical solution. For instance, the more $\beta_{\min}$ and $\beta_{\max}$ is small, the more accurate are the results to the expense of the computational time. In many AMR method, the choice of these thresholds parameters is therefore a critical weakness.

On one hand, if a cell C_{k_b} is split into two sub-cells $C_{k_{b0}}$ and $C_{k_{b1}}$, averaged values at time t_n are projected on each sub-cell:

$$\boldsymbol{w}_{k_{b0}}^n = \boldsymbol{w}_{k_{b1}}^n = \boldsymbol{w}_{k_b}^n$$

On the other hand, if two sub-cells $C_{k_{b0}}$ and $C_{k_{b1}}$ are coarsened, we initialise the new cell C_{k_b} as:

$$\boldsymbol{w}_{k_b}^n = \frac{1}{2}\left(\boldsymbol{w}_{k_{b0}}^n + \boldsymbol{w}_{k_{b1}}^n\right)$$

Moreover, as done in [8, Section 4.2], we use a numerical smoothing grid technique which prevents two adjacent cells to have a level difference greater than two for stability purpose.

To simplify the notations, we do not use the binary subscript notation and *we will assume that* $\beta_{\min} = \beta_{\max} = \beta$ *and therefore* $\alpha_{\min} = \alpha_{\max} = \alpha$ for the sake of simplicity. By abuse of notations, we write the adaptive algorithm as follows: for each cell C_k:

- if $S_k^n > \alpha$, the cell is refined and split into two sub-cells,
- if $S_k^n < \alpha$, the cell is coarsened (in the sense that two cells $C_{k_{b0}}$ and $C_{k_{b1}}$ are coarsened into a cell C_{k_b} as described before).

If the threshold is not correctly chosen, it is well-known that such a strategy may yields to spurious oscillations since locally, the grid incessantly changes between coarse and fine cells. However, even in this settings (*i.e.* $\alpha_{\min} = \alpha_{\max} = \alpha$), *we will see that the method proposed for the automatic selection of the threshold does not develop, in general, spurious oscillations*. If the threshold parameter is automatically well-chosen at each time step, one can always define thereafter $\alpha_{\min}$ and $\alpha_{\max}$ to allow to set a percentage of mesh refinement and mesh coarsening to reinforce the stability of the algorithm.

2.3 Mesh refinement criteria

The choice of a suitable mesh refinement criterion is crucial to perform accurate mesh adaptation. Indeed, ideally, one would like to set up a mesh that minimises the discretisation error and maximises the computational efficiency of the algorithm. The construction of an efficient criterion remains a difficult subject, since it depends on physical phenomena and on numerical methods. Multi-resolution analysis [12, 18] provides some robust tools to estimate and control rigorously the error at each refinement level. Based upon Richardson-type estimates of the local truncation error [4], one can also refine or coarsen to reach a given accuracy for a minimum amount of work. However, near discontinuous solutions, Richardson extrapolation is invalid [24]. It will work but it is no more reliable than heuristic methods and the extra cost cannot be justified. The entropy indicator for hyperbolic systems is also an efficient mesh refinement criterion for smooth and non smooth solutions [25]. Finally, the mesh criterion can be also constructed, as simple as possible, as the combination of heuristic gradients based on physical quantities to detect pertinent regions to refine or to coarsen [26]. We also refer to other methods based on *a posteriori* error estimates based on a measure of the local residual [30] and [15].

In this paper, we will consider the following three mesh refinement criteria.

Criterion 0: Since the exact solution of the Riemann problem (1)-(2) is known, we consider the criterion as the "best[1] mesh refinement criterion"

$$S_k^n := |h_k^n - h_{\text{ex}}(x_k, t_n)| \tag{4}$$

where h_{ex} stands for the exact solution.

Criterion 1: One can show that System (1) admits a mathematical entropy E

$$E(\mathbf{w}) = \frac{q^2}{2h} + \frac{gh^2}{2}$$

with $q = hu$ which satisfies the entropy inequality

$$\mathscr{E}(x,t) = \partial_t E(\mathbf{w}(x,t)) + \partial_x \left(G(\mathbf{w}(x,t))\right) \leq 0 \,. \tag{5}$$

with

$$G(\mathbf{w}) = \left(E(\mathbf{w}) + \frac{gh^2}{2}\right) u \,.$$

The entropy satisfies a conservation equation only in regions where the solution is smooth and an inequality when the solution develops shocks. In simple cases, it can be proved that the term missing in (5) to make it an equality is a Dirac mass. Following [25, 8, 2], we define the numerical density of entropy production

[1] The criterion 0 is qualified as "the best" in the *a posteriori* sense. Indeed it gives the exact numerical error at time t^n. However it will not necessary give the best convergence rate at the end because the best mesh refinement should be done where large errors are going to be done (at time t^{n+1}).

$$\mathscr{E}_k^n = \frac{E(\mathbf{w}_k^{n+1}) - E(\mathbf{w}_k^n)}{\delta t_n} + \frac{\left(\mathscr{G}_{k+1/2}(\mathbf{w}_k^n, \mathbf{w}_{k+1}^n) - \mathscr{G}_{k-1/2}(\mathbf{w}_{k-1}^n, \mathbf{w}_k^n)\right)}{\delta x_k}$$

where the numerical flux $\mathscr{G}_{k+1/2}$ is computed with the exact solution of the 1D Riemann problem at the interface $x_{k+1/2}$ with the left state $\mathbf{w}_k^n$ and the right state $\mathbf{w}_{k+1}^n$ as done before in Sect. 2.1.
At the discrete level, as shown in [25], $\mathscr{E}_k^n$ converges to zero with the same rate of the local truncation error in smooth regions. Therefore, $\mathscr{E}_k^n$ can be regarded as an *a posteriori* error indicator. Moreover, as proved in [25], the entropy production built on first order monotone schemes is essentially negative definite, in the sense that positive overshoots in the cell entropy inequality are possible, but only on non monotone data and their amplitudes decrease fast under grid refinement for smooth solutions. Thus, in the following, we define

$$S_k^n = |\mathscr{E}_k^n| \ . \tag{6}$$

Criterion 2: Finally, the last mesh refinement criterion is based on the gradient method. It can be constructed as a combination of heuristic gradients based on physical quantities to detect pertinent regions to refine or to coarsen [26]. To make it as simple as possible, we consider the gradient of h which can be an efficient mesh refinement criterion if the threshold is well-chosen,

$$S_k^n = \left| \frac{h_{k+1}^n - h_k^n}{x_{k+1} - x_k} \right| \ . \tag{7}$$

In what follows, we define the piecewise mesh refinement criterion as follows :

$$S(x,t) = \sum_k S_k^n \mathbb{1}_{C_k \times [t_n, t_{n+1}[}(x) \tag{8}$$

where $\mathbb{1}_A(x) = \begin{cases} 1 \ \text{ if } x \in A \ , \\ 0 \ \text{otherwise.} \end{cases}$

3 Mesh refinement thresholding methods

Generally speaking, as already described before, the mesh adaptation process is constructed as follows: if $S(x,t) > \alpha$, the cell is refined and split into two sub-cells and otherwise the cell is coarsened. The quantity α is the so-called mesh refinement threshold. As presented before, if we set $\alpha = \beta S_m$ for some $\beta > 0$, we obtain the mean method. We propose here to review some of the classical thresholding methods. Then, we present a new automatic thresholding approach constructed with the decreasing rearrangement function of the mesh refinement criterion S.

3.1 A brief review of the classical thresholding methods

In general, the mesh refinement criterion of error indicator type can be written as a function of the local discretisation error ε

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

where $f : \mathbb{R} \to \mathbb{R}^+$. If the mesh refinement criterion is an *a priori* or *a posteriori* error estimates, then the function f is in general linear as for the criterion (4), say $S(x,t) = \kappa\varepsilon(x,t)$, with $\kappa > 0$. In this settings, it is easy to define the mesh refinement threshold $\alpha = \varepsilon(x,t)/\kappa$ which almost corresponds to the desired local accuracy. Let us note that in general it is very difficult to construct such a mesh refinement criterion. The criterion (6) is an *a posteriori* error indicator. In the case of the mesh refinement criterion (6), the function is not necessarily linear and therefore it is not easy to define a suitable threshold.

There exists several thresholding methods. We recall the classical ones: mean method, mean method with standard deviation, two-steps filtering, wavelets, local maxima, Dannenhoffer and Powell cumulative distribution methods and we finally present our method. The last one is the only method, up to our knowledge, yielding to an efficient automatic threshold almost parameterless contrary to the previous ones.

Mean method.

It is the simplest and the cheapest method. It can be an efficient method to detect homogeneous smooth solutions, *i.e.* solutions for which the local maxima of S have almost the same amplitudes. In what follows, we refer to smooth solutions for homogeneous smooth solutions. As done before, for a given time $t > 0$, the mesh refinement criterion S is compared to

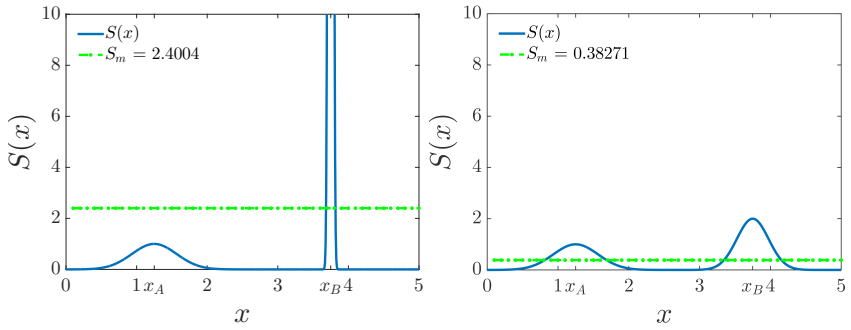
$$\alpha = \beta \frac{1}{|\Omega|} \int_\Omega S(x,t)\, dx$$

where β is a user-calibrated dimensionless parameter.

For a discontinuous solution, the mean method is not able to detect the relevant smooth regions to refine if the threshold β is not set small enough. However, the mean method is, in general, suitable for smooth solutions for which setting $\beta = 1$ yields to an efficient refinement. In Fig. 2, we have illustrated formally a discontinuous solution, see Fig. 2(a), and a smooth one, see Fig. 2(b). In Fig. 2(a), the peak localised around at $x = x_B = 3.75$ shows the presence of a discontinuity of the solution while the remainder region corresponds to the smooth part of the solution.

On one hand, for a discontinuous solution, one can see in Fig. 2(a) that if the parameter $\beta \geq 1$ (*i.e.* $\alpha \geq S_m$) then the smooth part of the solution will not be marked for refining. On the other hand, for a smooth solution, one can observe in

Fig. 2(b) that setting $\beta = 1$ allows to detect relevant part of the mesh to refine. *In general, to be efficient* β *should be calibrated such that* $\beta \in (0,1]$ *or equivalently* $\alpha \in (0, S_m]$.



(a) $S(x)$ with $a = 200$, $b = 1000$ (b) $S(x)$ with $a = 2$, $b = 10$

Fig. 2 Comparison of the mean value S_m for two given criteria $S(x) = a\exp(-b(x-x_B)^2) + \exp(-5(x-x_A)^2)$ where $x_A = 1.25$ and $x_B = 3.75$. S represents either a discontinuous solution (left) or a smooth solution (right).

Mean method with standard deviation.

The mean method can be improved by taking into account the fluctuations of the mesh refinement criterion. For a given time $t > 0$, the mesh refinement criterion S is now compared to

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

where σ is the standard deviation of $S(x,t)$ and $\delta \in \mathbb{R}$. These methods are largely used due to their simplicities, see for instance [14, 31, 13, 27] however β and δ are problem dependent and therefore user-calibrated parameters.

Filtering: two steps method.

The method of "filtering" is constructed to detect the smooth part of a discontinuous solution, see for instance [31, 1]. This method improves the classical mean method by applying two successive filtering.

For a given time $t > 0$ and for a given fixed β, as a first step, the mean method is applied to S to identify discontinuities and steep gradients in the solution, like in the example in Fig. 2(a). Then, we redefine the mesh refinement criterion as $S_1(x,t) = S(x,t)$ if $S(x,t) < \alpha = \beta \frac{1}{|\Omega|} \int_\Omega S(x,t)\, dx$, otherwise $S_1(x,t) = \beta \frac{1}{|\Omega|} \int_\Omega S(x,t)\, dx$. We now apply the mean method to the mesh refinement criterion S_1. As a consequence,

this method allows to filter the large peaks into the function S and try to detect also the smooth part of the solution. However, the two steps may be not enough to efficiently capture the smooth part of the solution. Moreover, the second drawback concerns still the parameter β which is again test-case dependent.

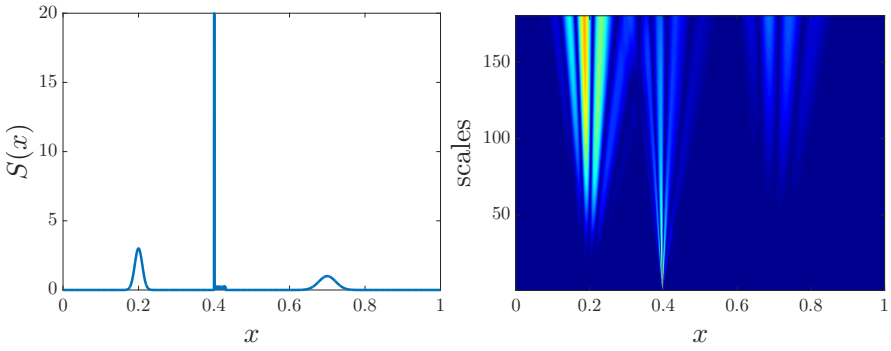
Filtering: wavelet method.

In contrast with the previous method, the detection of smooth and discontinuous solutions can be efficiently captured with a space-time localisation based on a wavelet transform, see for instance [17]. 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 smooth regions [2] as displayed in Fig. 3(b) for a given signal

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

where $x \mapsto r_{0,1}(x) \in [0,1]$ returns a random number, see Fig. 3(a). High variations of the signal yield to large absolute values of wavelet coefficients centered around the discontinuity at all regions. Depending on the wavelet's support, the larger the region 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 has the smallest region. On the contrary, smooth signal produces relatively large wavelet coefficients at large regions 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. 3, for a given region (which corresponds to a threshold parameter α), 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 refine. However, as for the method described before, we are still confronted with the choice of a threshold parameter (y). Moreover, even if this method is based on fast wavelet transform, it increases the global cost of the numerical method.

[2] the wavelet coefficients in this example are computed with the Daubechies wavelet with four vanishing moments [6, 7]



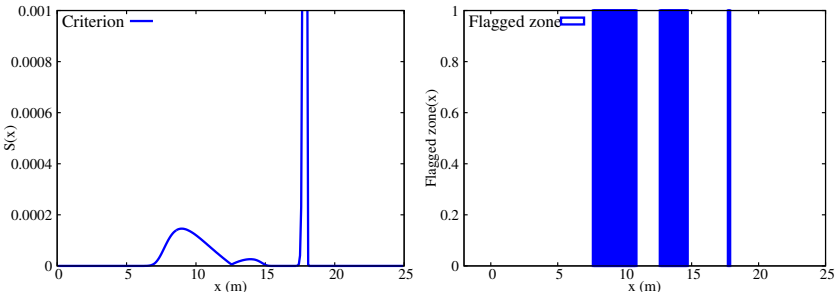
(a) Mesh refinement criterion (b) Wavelet coefficients ($x =$ space, $y =$ region)

Fig. 3 Illustration of the wavelet transformation for a given mesh refinement criterion computed with the Daubechies wavelets with four vanishing moments (warm colors correspond to large coefficients and cold colors to small coefficients).

Local maxima approach.

It consists in looking for the local maxima of the mesh refinement criterion. As a first drawback, the local maxima of the mesh refinement criterion are not necessarily those of the local error. Once a local maximum is detected, say x^*, the surrounding area to refine should be large enough. It can be defined without parameter by constructing the interval $[x^* - \zeta(x^*), x^* + \zeta(x^*)]$ where $\zeta(x^*)$ is the distance between the local maxima position x^* and the closest inflexion point of the criterion. As a second drawback, such a method can be quite costly in a multi-dimensional settings. Moreover, for a noisy criterion function (like for instance in Fig. 5(a)), it can lead to an over refined domain.

In Fig. 4, 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 point (see Fig. 4(b) and Fig. 4(a)).



(a) Mesh refinement criterion S (b) Region to refine

Fig. 4 Illustration of the local maxima method for a mesh refinement criterion involving multiple regions.

Dannenhoffer cumulative distribution method and Powell method.

The method proposed by Dannenhoffer [5] is simple and *a priori* efficient to automatically set the threshold parameter.

Let us consider a given discrete mesh refinement criterion (8) for a given time $t > 0$. Let $(\alpha_j)_{0 \leq j \leq M}$ be an increasing sequence of $M+1$ threshold parameters such that $\alpha_0 = 0$ and $\alpha_M = \max_x S(x,t_n)$ where α_j, is given by

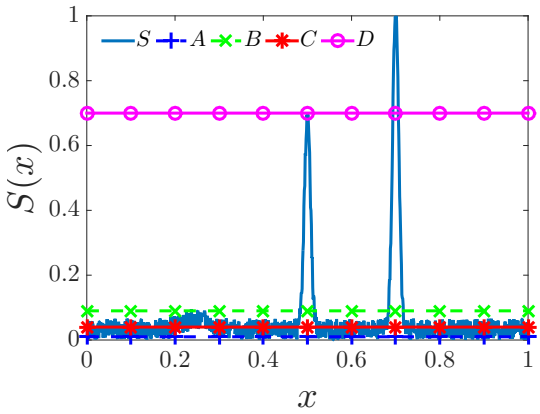
$$\alpha_j = S_M \left(\frac{j}{M} \right)^{\gamma}, \; 0 \leq j \leq M \tag{9}$$

with $\gamma \geq 1$. Following Dannenhoffer [5], we construct the piecewise constant distribution function $d_j = d(\alpha_j)$ as:

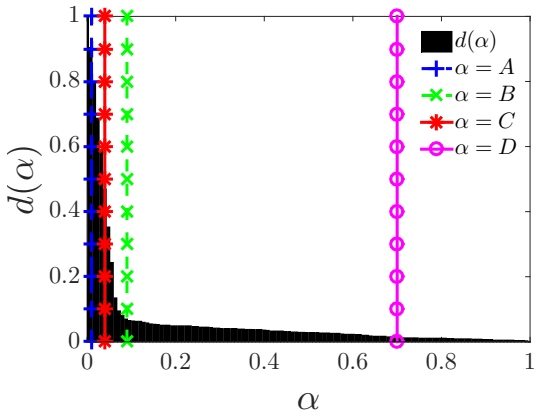
$$d(\alpha) = \sum_{j=0}^{M-1} d_j \mathbb{1}_{(\alpha_j, \alpha_{j+1})}(\alpha) \text{ with } d_j = \#\{k \; ; \; S_k^n > \alpha_j\} \tag{10}$$

where # is the number of elements in the set $\{S_k^n > \alpha_j\}$. *This function is useful to detect the local maxima*.

We display the shape of the cumulative distribution function in Fig. 5(b) (black area) for a given perturbed mesh refinement criterion (see Fig. 5(a)) as in [5]. In this illustration, the first and the second peaks around $x = 0.5$ and $x = 0.7$ can be associated to a steep gradient or a discontinuity in the solution. The peak around $x = 0.25$ can be associated to a region where the solution is smooth. Ideally, we would like to detect those cells to mark for refinement.



(a) Mesh refinement criterion



(b) Cumulative distribution function

Fig. 5 Dannenhoffer cumulative distribution function for a given mesh refinement criterion.

One can define several threshold parameters denoted $\alpha = A$, $\alpha = B$, $\alpha = C$ and $\alpha = D$ in Fig. 5. The threshold $\alpha = A$ corresponds to the area for which the mesh refinement criterion oscillates. Thus, setting $\alpha = A$ yields to detect almost all the cells to refine and it is not a good strategy. The threshold D is too large and almost all regions are not detected. Thus, Dannenhoffer proposes to set a threshold between $\alpha = C$ and $\alpha = B$ which is localised at the "knee" of the cumulative distribution (see Fig. 5(b)). In particular, following [5], the "knee" of the cumulative distribution is defined as the smallest $\alpha = \alpha_{\text{Dan}}$ such that

$$d'(\alpha) = -1 \, .$$

Ideally, if the location of the "knee" can be computed then one can construct an automatic threshold which allows to refine the pertinent areas. However, as mentioned

in [5], to compute correctly the threshold α_{Dan}, a construction of an accurate interpolation of the derivative is required which increases the computational cost of the AMR algorithm. Otherwise, the threshold α_{Dan} computed is often too small and the method detects too many cells to refine as mentioned in [5].

Based on this cumulative distribution function, Powell [23] proposes to automatically set the threshold at the lowest value of the refinement parameter that produces a local maximum in the curvature $\kappa = \frac{|d''|}{(1+d'^2)^{3/2}}$. The threshold $\alpha = \alpha_{\mathrm{P}}$ is localised before the first inflexion point of the cumulative distribution function but still an accurate interpolation of the second and third derivatives are required.

As a conclusion, in general, the construction of a suitable mesh refinement threshold is either user-calibrated or automatically but with an extra numerical cost. In the following section, we propose a new automatic mesh refinement threshold α_{PE}, only based on the cumulative distribution function and not on its derivatives. The threshold α_{PE} is defined such that $\max(\alpha_{\mathrm{Dan}}, \alpha_{\mathrm{P}}) < \alpha_{\mathrm{PE}}$. The computational cost of the construction of the threshold α_{PE} is less expensive than the Dannenhoffer [5] or the Powell [23] method.

3.2 A distribution function for automatic thresholding

In this section, we present the new method to select the threshold automatically to detect relevant cells to refine.

For the sake of simplicity and readability, we present our new threshold method in the one-dimensional case and we refer to the second part of this paper [20] for the generalisation to the multi-dimensional case.

Let us consider a smooth (at least twice differentiable) mesh refinement criterion $S(x,t) \in \mathbb{R}^+$, $x \in [0,L]$ 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 $S_\infty = 0$ the numerical solution is identically equal to zero).

Remark 2. The meaning of the assumption $S(0) = S(L) = S'(0) = S'(L) = 0$ depends on the mesh refinement criterion and simplifies the following computations. For the mesh refinement criterion (7), it means that the free surface is flat at the boundaries. Generally speaking, if for instance at $x = 0$, $S(0) \neq 0$ and $S'(0) \neq 0$, one can always arbitrarily extend the function S on the interval $(-\varepsilon, 0)$ such that $S(-\varepsilon) = S'(-\varepsilon) = 0$ for a given $\varepsilon > 0$. So without loss of generality, one can consider that $S(0) = S(L) = S'(0) = S'(L) = 0$.

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. Indeed, for each $0 < \alpha < S_\infty$, since $S_\infty > 0$, 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

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

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

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

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

With these settings, we define the cumulative distribution function

$$\alpha \in [0, S_\infty] \mapsto\ d(\alpha) := \begin{cases} L & \text{if } \alpha = 0 , \\ \sum_{k=1}^{p_\alpha} x_{2k+1}(\alpha) - x_{2k}(\alpha) & \text{if } 0 < \alpha < S_\infty , \\ 0 & \text{if } \alpha = S_\infty . \end{cases} \tag{11}$$

Remark 3. The function (11) corresponds to the well-known 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 $\{S(x) > \alpha\}$.

Remark 4. From a theoretical viewpoint, in general, the presence of a Dirac mass in the mesh refinement criterion indicates a discontinuous solution. The automatic thresholding being based on the Lebesgue measure, the measure of a singleton being zero, the method will be not able to detect the presence of discontinuities. However, from a numerical viewpoint, due to the numerical diffusion, the Dirac mass is often associated to a steep peak and globally the mesh refinement criterion can be assumed smooth enough. As a consequence, the method works to detect large peaks which are associated to numerical approximation of discontinuous solutions. Therefore, we can assume that the mesh refinement criterion is at least twice differentiable. Even if the smoothness assumption on S is rough, the method developed works to detect the discontinuity, the steep gradient and the smooth part of the solution.

Then, one has:

Theorem 1. *Let $l \in \mathbb{N}_*$, $S \in C^l([0,L], \mathbb{R}^+)$ be a smooth function such that $S(0) = S(L) = S'(0) = S'(L) = 0$, $0 < S_\infty < \infty$ 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*
   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 function (11), the first and the second properties hold. For $\alpha \in [0, S_\infty]$, the derivative of d is given by

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

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 last property is obtained as a straightforward consequence of the above inequality. □

In order to illustrate the previous notations and results, we display in Fig. 6, for a given mesh refinement criterion S (see Fig. 6(a)), the corresponding distribution function (11) (see Fig. 6(b)), the first (see Fig. 6(c)) and the second derivative (see Fig. 6(d)). For better readability the distribution has been normed by L. In this example, we consider a function S with $p = 3$. In Fig. 6(b), the points for which the distribution function has vertical asymptotes are

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

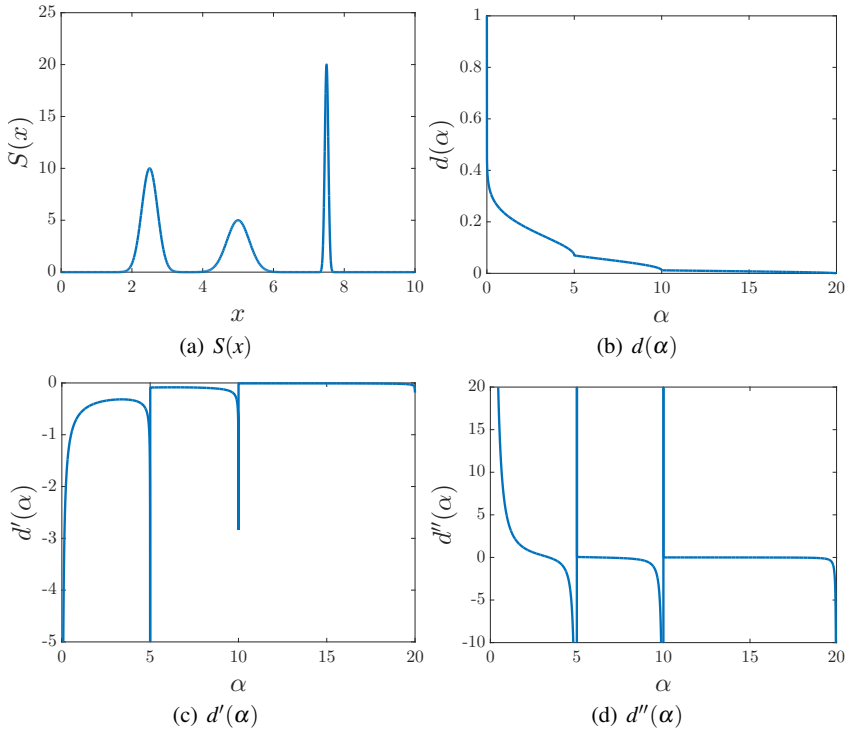
and correspond to the local maxima of S which are sorted, by construction, from the smallest to the largest. In Fig. 6(a), for $\alpha = 2$, the set Z_α is composed of $6 = 2p_\alpha$ elements which are approximately

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

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

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

$$d(\alpha) = \frac{(x_1(\alpha) - x_0(\alpha)) + (x_3(\alpha) - x_2(\alpha)) + (x_5(\alpha) - x_4(\alpha))}{L} \, .$$



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

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

Fig. 6 The distribution function d, the first d' and the second derivative d'' 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 $L = 10$.

Remark 5. The cumulative distribution function (11) is the continuous version of the Dannenhoffer cumulative distribution function [5] (see also Eq. (10)).

In view of Theorem 1, the distribution function (11) is useful because it provides a complete description of the mesh refinement criterion S, in particular, it allows to localise the smallest and the largest local maximum. Therefore, the construction of an automatic threshold based on the function d can be relevant.

However, on one hand, the method proposed by Dannenhoffer (resp. by Powell) is based on the first derivative (resp. the second and the third derivatives) of the distribution function (10) which requires an accurate interpolation. On the other hand, using only the distribution function (11) or equivalently (10) to construct a relevant mesh refinement threshold is numerically difficult since a local maximum corresponds to detect an area for which the derivative of d is infinite. Therefore, one way to overpass these drawbacks is to use the distribution function (11), without using any derivatives of d, and simply weighting the function d as follows

$$f(\alpha) = \alpha d(\alpha)\ . \tag{12}$$

This function f transforms a local point of d between the first inflexion point of d and the first local maximum of S (*i.e.* the first local vertical asymptote of d) into a local maximum. More precisely, we have the following useful properties:

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

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

and the function f (12) 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 function. As a consequence, there exists $\overline{\alpha}_1 \in (\alpha^{**}_{1}, \alpha^{*}_{1})$ which maximises 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 $\alpha = \overline{\alpha}_k$ for a given k yields to

$$\max(\alpha_{\mathrm{Dan}}, \alpha_{\mathrm{P}}) < \overline{\alpha}_k \ .$$

On one hand, as mentioned in Sect. 3.1, for smooth solutions, the mesh refinement threshold $\alpha = S_m$ is in general relevant ($S_m = \frac{1}{|\Omega|}\int_\Omega S(x)\ dx$). On the other hand, we have also pointed out that α should be less than S_m for discontinuous solutions. Thus, the threshold $\alpha = \overline{\alpha}_k$ if $\overline{\alpha}_k < S_m$, otherwise $\alpha = S_m$ is a good candidate. More precisely, we define our automatic threshold as follows:

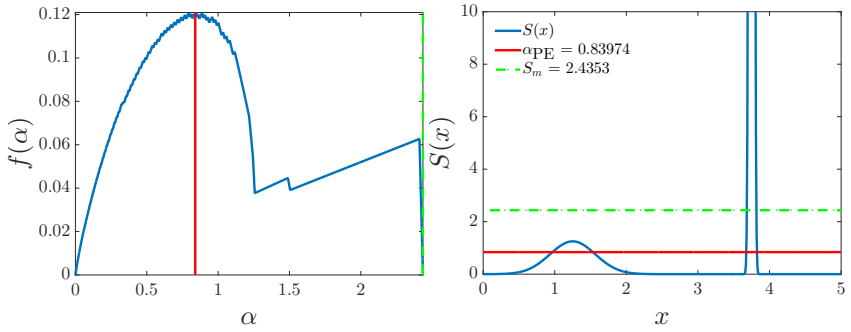
$$\alpha_{\mathrm{PE}} \text{ such that } f(\alpha_{\mathrm{PE}}) = \max_{0<\alpha\leq S_m} f(\alpha) \ . \tag{13}$$

One important consequence of this choice is that the threshold is a global maximum of a regular function, and therefore easy and numerically reliable to determine. By construction, if $\alpha_{\mathrm{PE}} = S_m$, we are able to predict that the solution is almost everywhere smooth and if $\alpha_{\mathrm{PE}} < S_m$ that the solution possesses at least one discontinuity or a steep gradient.

Let us now illustrate the threshold α_{PE} through an example. We represent in Fig. 7 a discontinuous solution and in Fig. 8 a smooth solution for a given mesh refinement criterion S.

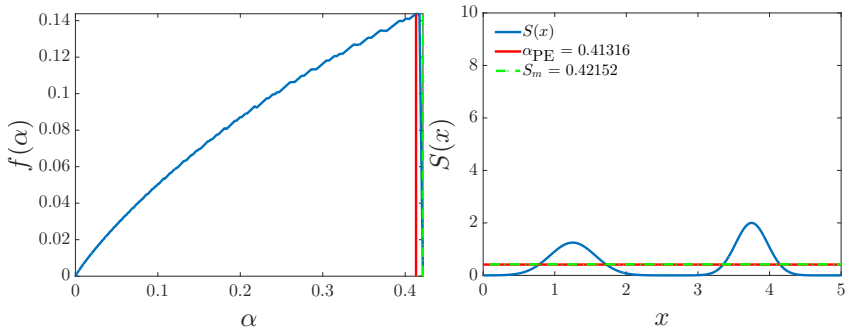
- For the discontinuous solutions, as mentioned before, the threshold $\alpha = S_m$ is not small enough to detect the region where the solution is smooth. However, if we set $\alpha = \alpha_{\mathrm{PE}} < S_m$ as a mesh refinement threshold, one can detect this region (see Fig. 7(b)).

- For smooth solutions, as emphasised in Sect. 3.1, the threshold $\alpha = S_m$ allows to detect almost all smooth regions to refine as displayed in Fig. 8(b).



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

Fig. 7 The function f for the mesh refinement criterion $S(x) = 200\exp(-1000(x-3.75)^2) + 1.25\exp(-5(x-1.25)^2)$ representing a shock type solution.



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

Fig. 8 The function f for the mesh refinement criterion $S(x) = 2\exp(-10(x-3.75)^2) + 1.25\exp(-5(x-1.25)^2)$ representing a smooth solution.

3.3 Numerical illustration using α_{PE} as an automatic mesh refinement threshold.

In this section we show the efficiency of the proposed method in the selection of an automatic threshold. To this end, we consider the Riemann problem (1)-(2) with

$$\forall x \in [0,80],\ \mathbf{w}(x,0) = (h(x,0),u(x,0)) = \begin{cases} (5.64,8) & \text{if } x < 20\,, \\ (0.6,8) & \text{if } x > 20\,. \end{cases} \tag{14}$$

for which the exact solution is composed of a left-going rarefaction and a right-going shock wave (see, for instance, Fig. 9(a), black line).

Each numerical experiment starts initially with 100 cells, the maximum level of refinement is set to $l_{\max} = 3$, $\gamma = 2$, $M = 1000$ (see Eq. (9)) and we prescribe free boundary conditions.

The objective is to show that, for any given mesh refinement criterion, the threshold α_{PE} is selected automatically to adapt the mesh without hand-calibration. In particular, we will confront our method to the mean method (*i.e.* $\alpha = \beta S_m$) for several values of β. We recall that in this method, the threshold has to be calibrated. In this numerical experiment, we have used the following strategy: for each cell C_k:

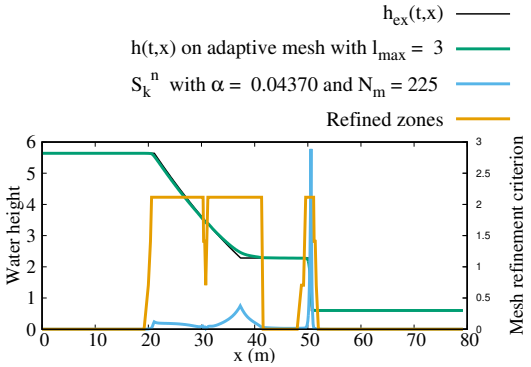
- if $S_k^n > \alpha$, the mesh is refined and split,
- if $S_k^n < \alpha$ the mesh is coarsened.

which is known to be a source of spurious oscillations. We will see that our method leads to non-oscillating numerical solution without hand-calibration. For the mean method, as said before, if the threshold parameter β is not well-calibrated, then the numerical solution can yield to spurious oscillations whatever the mesh refinement criterion is. To be non redundant, we show the results only for the criterion (6).

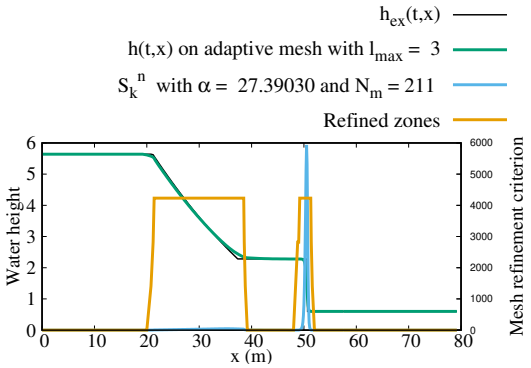
In Fig. 9, we have displayed the numerical water height at time $t = 2$ s using the method of automatic selection of the threshold with the mesh refinement criteria (4), (6) and (7). With our method, the threshold parameter α_{PE}, being adapted at each time step, allows to detect the pertinent region of the domain to refine as one can observe in Figs. 9 and 11. It is important to note that, in spite of the mesh adaptation strategy (refined or coarsen), the numerical solution at time $t = 2$ s does not contain spurious oscillations. This also true at different times, as shown in Figs. 10(a)–10(d), for instance for the mesh refinement criterion (4). Furthermore, it is also interesting to highlight that, whatever the mesh refinement criterion, we obtain almost the same numerical solution with almost the same order of accuracy (as observed in Fig. 9 and Table 1).

Criterion	$N_m(l) \to \|h - h_{ex}\|_1$	$N_m(l) \to \|u - u_{ex}\|_1$
0	2.0136	2.1342
1	2.0491	2.0901
2	2.1109	2.1789

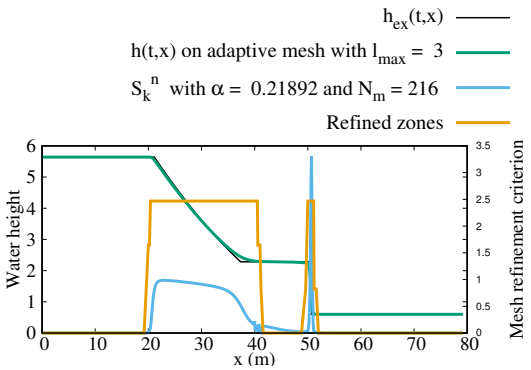
Table 1 Convergence rate of the L^1 discretisation error obtained with the refinement level $l = 1$, 2, 3 and 4.



(a) criterion (4)

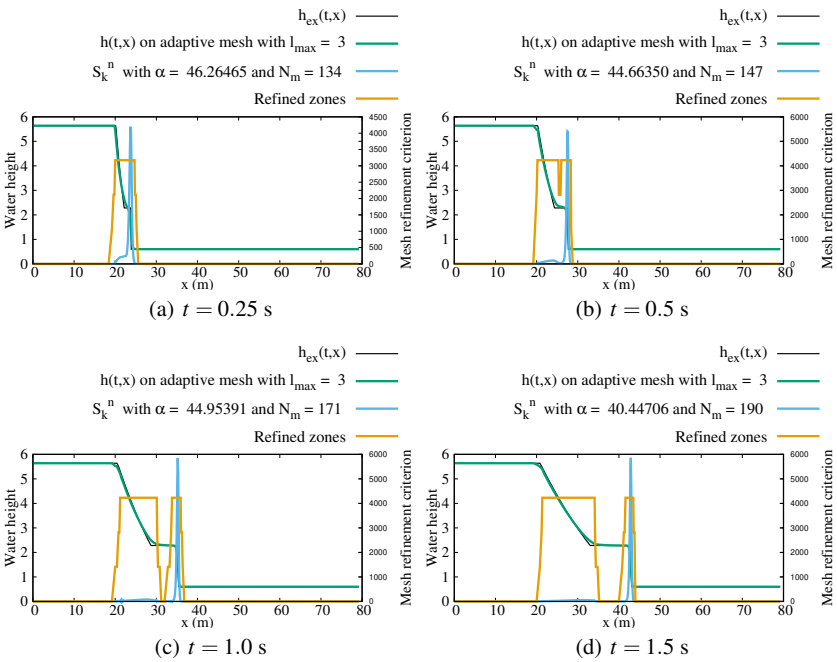


(b) criterion (6)



(c) criterion (7)

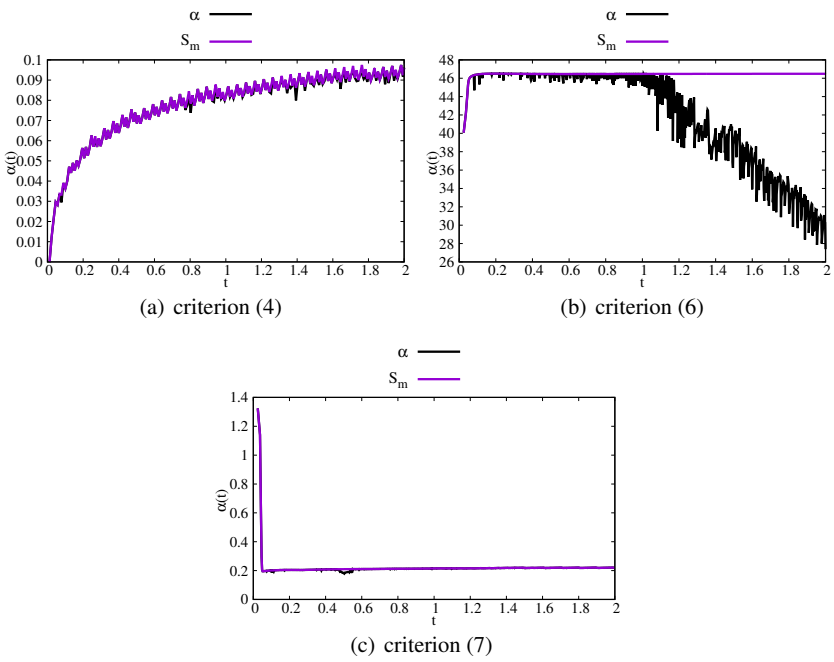
Fig. 9 Numerical results for the water height at time $t = 2$ s. N_m is the mean number of cells used during the simulation.



(a) $t = 0.25$ s (b) $t = 0.5$ s

(c) $t = 1.0$ s (d) $t = 1.5$ s

Fig. 10 Numerical results for the water height obtained with the criterion (6) at time $t = 0.25$ s, $t = 0.5$ s, $t = 1$ s and $t = 1.5$ s. N_m is the mean number of cells used during the simulation.



(a) criterion (4) (b) criterion (6)

(c) criterion (7)

Fig. 11 Time evolution of the threshold parameter α_{PE}.

In Fig. 12, we have displayed the numerical water height at time $t = 2$ s using the mean method with the mesh refinement criterion (6). Obviously, if the threshold parameter β is not well-chosen, we get:

- in Fig. 12(a), only the shock region is refined since β is set too large,
- in Fig. 12(b), the numerical solution oscillates,
- in Fig. 12(c), β is well-chosen, the pertinent regions to refine are well-detected and the numerical solution is well-computed (as in Fig. 9(b) with α_{PE}),
- in Fig. 12(d), the domain is refined almost everywhere since β is set too small.

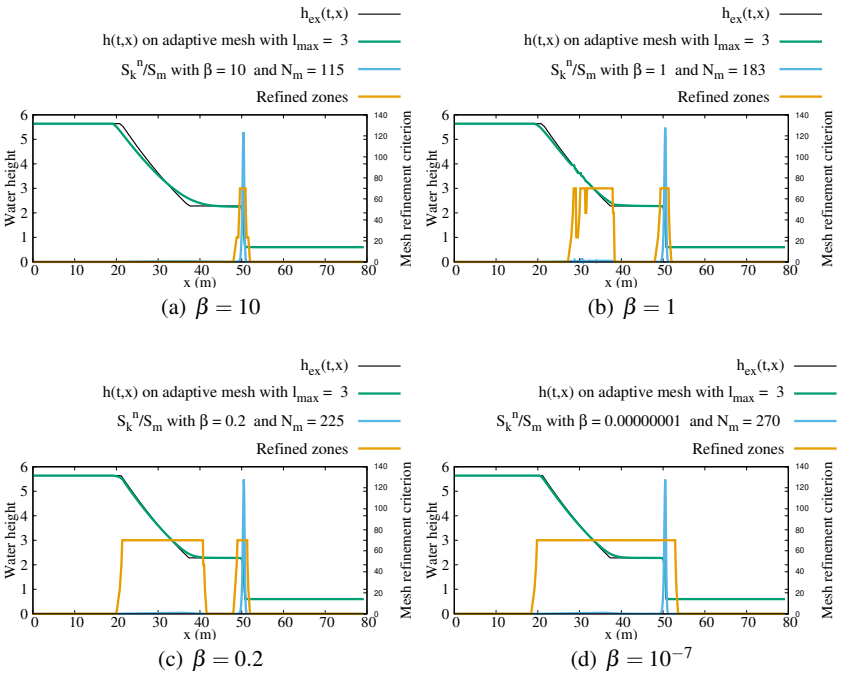


Fig. 12 Numerical results for the water height at time $t = 2$ s using the criterion (6) with the mean method. N_m is the mean number of cells used during the simulation.

4 Conclusion

In this paper, we have introduced a new method to set the threshold parameter automatically to get an h-AMR algorithm almost parameterless. This method allows, in principle, to refine without hand-calibration the pertinent regions.

The automatic selection is constructed from a weighted function $\alpha \mapsto f(\alpha) = \alpha d(\alpha)$ where d is decreasing rearrangement function of the mesh refinement criterion. The function f allows to compute an automatic threshold without an extra computational cost compared to the method by Dannehoffer or by Powell. Obviously other weighted functions are possible but it is important to keep in mind that a perfect automatic thresholding is *a priori* unreachable without any knowledge of the link between the real error and the used mesh refinement criterion. Therefore we can only focus to construct a method which has a good behaviour depending of the criterion function (with heterogeneous or homogeneous amplitudes, with smooth or discontinuous solutions, *etc.*).

Moreover, the method works well for any mesh refinement criterion and yields to a numerical solution without spurious oscillations. Whatever the mesh refinement

criterion is, we obtain almost the same numerical solution with almost the same order of accuracy. In particular, combined with a simple gradient method as a mesh refinement criterion which is less expensive than other criteria, the h-AMR algorithm can be very efficient.

In this part, we have illustrated the efficiency on a one dimensional Riemann problem to show that our method is able to detect simple waves. In the second part of this paper [20], we show through several multi-dimensional test cases, including multiple waves interactions, the efficiency of the method.

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. The author also wish to thanks the referees for their valuable remarks which led to substantial improvement of the first version of this paper.

References

1. Aftosmis, M.: Upwind method for simulation of viscous flow on adaptively refined meshes. AIAA journal **32**(2), 268–277 (1994)
2. 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)
3. 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
4. Berger, M.J., Oliger, J.: Adaptive mesh refinement for hyperbolic partial differential equations. Journal of computational Physics **53**(3), 484–512 (1984)
5. Dannenhoffer, J.F.: Grid adaptation for complex two-dimensional transonic flows. Ph.D. thesis, Massachusetts Institute of Technology (1987)
6. Daubechies, I., et al.: Ten lectures on wavelets, vol. 61. SIAM (1992)
7. Ersoy, M.: A simple and efficient new algorithm to increase the regularity and vanishing moments of biorthogonal wavelets. preprint (2008)
8. 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
9. 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)
10. 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)
11. Godlewski, E., Raviart, P.: Numerical approximation of hyperbolic systems of conservation laws, *Applied Mathematical Sciences*, vol. 118. Springer-Verlag, New York (1996)
12. Harten, A.: Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. Communications on Pure and Applied Mathematics **48**(12), 1305–1342 (1995)
13. 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)

14. Kallinderis, Y.G., Baron, J.R.: Adaptation methods for a new navier-stokes algorithm. AIAA journal **27**(1), 37–43 (1989)
15. Karni, S., Kurganov, A., Petrova, G.: A smoothness indicator for adaptive algorithms for hyperbolic systems. J. Comp. Phys. **178**(2), 323–341 (2002)
16. 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)
17. Leonard, S., Terracol, M., Sagaut, P.: A wavelet-based adaptive mesh refinement criterion for large-eddy simulation. Journal of Turbulence **7**(64) (2006)
18. Müller, S.: Adaptive multiscale schemes for conservation laws, vol. 27. Springer Science & Business Media (2012)
19. Perthame, B., Simeoni, C.: A kinetic scheme for the Saint-Venant system with a source term. Calcolo **38**(4), 201–231 (2001)
20. Pons, K., Ersoy, M., Golay, F., Marcer, R.: Adaptive mesh refinement method. Part 2: Application to tsunamis propagation (2016). URL https://hal.archives-ouvertes.fr/hal-01330680. Working paper or preprint
21. Popinet, S.: Quadtree-adaptive tsunami modelling. Ocean Dynamics **61**(9), 1261–1285 (2011)
22. Popinet, S.: Adaptive modelling of long-distance wave propagation and fine-scale flooding during the tohoku tsunami. Nat. Hazards Earth Syst. Sci **12**(4), 1213–1227 (2012)
23. 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)
24. Powell, K.G., Roe, P.L., Quirk, J.: Adaptive-mesh algorithms for computational fluid dynamics. In: Algorithmic trends in computational fluid dynamics, pp. 303–337. Springer (1993)
25. Puppo, G., Semplice, M.: Numerical entropy and adaptivity for finite volume schemes. Commun. Comput. Phys. **10**(5), 1132–1160 (2011). DOI 10.4208/cicp.250909.210111a. URL http://dx.doi.org/10.4208/cicp.250909.210111a
26. Quirk, J.J.: A parallel adaptive grid algorithm for computational shock hydrodynamics. Applied numerical mathematics **20**(4), 427–453 (1996)
27. 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)
28. Simeoni, C.: Remarks on the consistency of upwind source at interface schemes on nonuniform grids. SIAM J. Sci. Comput. **48**(1), 333–338 (2011)
29. 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
30. Verfürth, R.: A posteriori error estimation and adaptive mesh-refinement techniques. Journal of Computational and Applied Mathematics **50**(1), 67–83 (1994)
31. Warren, G.P., Anderson, W.K., Thomas, J.L., Krist, S.L.: Grid convergence for adaptive methods. AIAA paper **1592**, 1991 (1991)