

LOCAL TIME STEPPING WITH ADAPTIVE TIME STEP CONTROL FOR A TWO-PHASE FLUID SYSTEM*

FRÉDÉRIC COQUEL^{1,2}, QUANG LONG NGUYEN³, MARIE POSTEL^{1,2} AND QUANG HUY TRAN³

Résumé. Cet article aborde la question de l'utilisation efficace des maillages adaptatifs en vue de simuler correctement le transport des fronts sur des temps longs avec un coût CPU raisonnable. Ces singularités proviennent des systèmes de lois de conservation modélisant les écoulements diphasiques à travers les conduites pétrolières. Nous proposons non seulement un cadre multirésolution afin d'ajuster le pas d'espace à la régularité locale de la solution, mais aussi une stratégie de pas de temps local afin d'alléger les contraintes dues à la condition CFL de stabilité. Nous examinerons plus particulièrement le choix optimal des micro pas de temps à l'intérieur d'un macro pas de temps. Nous présentons les simulations numériques pour illustrer l'efficacité des méthodes préconisées.

Abstract. This paper is concerned with how to make effective use of adaptive grids to correctly simulate the transportation of fronts over a long range of time under reasonable CPU time. These step gradients arise from systems of conservation laws modelling two-phase flows in pipelines. We propose not only a multi-resolution setup in order to adapt the space grid to the local smoothness of the solution, but also a local time stepping strategy in order to alleviate the constraints due to the CFL stability condition. A special focus is given to the optimal choice of micro time steps within a macro time step. Numerical simulations are presented to illustrate the efficiency of the methods put forward.

INTRODUCTION

We are interested in accurate and fast computation of two-phase flows in pipelines within the context of a drift-flux model given to us by physicists. This one-dimensional model consists of nonlinear conservation laws involving very expensive algebraic closure laws coming from experimental data. The closure laws are so intricate that they can even spoil the hyperbolicity of the PDE system. Whenever hyperbolic, this model contains two types of waves, namely, fast acoustic waves and slow kinematic waves. Only slow kinematic waves are interesting to the oil field engineers.

This setup is a good candidate for adaptive methods such as those introduced by Harten [15] and Cohen *et al* [6]. In analyzing the local smoothness of the solution using a wavelet-like transformation, we manage to design a local refinement strategy for adapting the mesh-size and for moving the mesh along with the solution. By “moving the mesh,” we mean that the fronts propagating at slow kinematic speeds

* This work was supported by the Ministère de la Recherche under grant ERT-20052274: Simulation avancée du transport des hydrocarbures and by the IFP.

¹ UPMC Univ Paris 06, UMR 7598 LJLL, Paris, F-75005 France

² CNRS, UMR 7598 LJLL, Paris, F-75005 France

³ IFP, Département Mathématiques Appliquées, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France

are suitably captured by refined cells and accurately computed. Comprehensive description and analysis of multi-resolution techniques can be found in [5] and [18]. The successful use of this type of methods for our problem was demonstrated in [1, 10].

Although multi-resolution has brought us significant speed-up factors, e.g., about 3 for a typical run, we are still hampered by the fact that the time step is the same all over the grid and therefore must be governed by a CFL condition on the smallest mesh-size. The goal of this paper is to overcome this restriction with the help of local time stepping techniques. By using different time steps in different areas of the grid, we legitimately hope to further enhance the computational performance of the code, since there will be much less calls to the expensive algebraic closure laws. Early works in this direction go back to Berger *et al* [4], Osher *et al* [22] and more recent works like [23], [3] and [12] provide alternatives to the method adopted in the present paper. Seemingly a natural enhancement to a space adaptive scheme, the actual implementation of a Local Time Stepping strategy (LTS) is usually not straightforward. In fact, the additional book-keeping inferred by the synchronization of the solution on the different discretized time grids might act as a deterrent, if the data structure is not clever enough to lead to interesting gains in computing time. Another inconvenience is the lack of analysis for such methods in the context of hyperbolic PDEs, whereas some stability results are available for parabolic equations [11, 14].

Recently a local time stepping strategy in the context of the fully adaptive scheme described in the previous paragraph has been proposed in Müller *et al* [20], and successfully implemented for various PDE systems in several dimensions [12, 17, 19]. This method was adapted in the context of the Lagrange-projection semi-implicit scheme for a simplified version (no-slip law) of our diphasic model in [8, 9]. We now carry it over to the full model in the explicit setting. The novelty on which we wish to lay emphasis is the optimal choice of the micro time steps within a macro time step.

This paper is organized into two parts. We first recall the model and the basic numerical scheme before any adaptive treatment. Then, we show how to formalize the multi-resolution and local time stepping philosophy to the problem at hand. Finally, some numerical results are supplied.

1. PHYSICAL MODEL AND NUMERICAL SCHEME

1.1. Physical model in Eulerian coordinates

Over the state space

$$\Omega_{\mathbf{w}} = \{\mathbf{w} = (\rho, \rho v, \rho Y)^T \in \mathbb{R}^3 \mid \rho > 0, Y \in [0, 1]\}, \quad (1)$$

we consider the problem of finding a function $\mathbf{w} : \mathbb{R}^+ \times \mathbb{R} \rightarrow \Omega_{\mathbf{w}}$ such that $\mathbf{w}(t, x)$ solves

$$\partial_t(\rho) + \partial_x(\rho v) = 0 \quad (2a)$$

$$\partial_t(\rho v) + \partial_x(\rho v^2 + P) = s \quad (2b)$$

$$\partial_t(\rho Y) + \partial_x(\rho Y v + \sigma) = 0. \quad (2c)$$

In (2), ρ is the density of the two-phase mixture, v is the velocity and Y is the gas mass-fraction. The quantities $P = P(\mathbf{w})$, $\sigma = \sigma(\mathbf{w})$ and $s = s(\mathbf{w})$ are given by

$$\sigma = \rho Y(1 - Y)\phi \quad (3a)$$

$$P = \rho Y(1 - Y)\phi^2 + p \quad (3b)$$

$$s = -\rho g \sin \theta - \frac{1}{2} C \rho v |v| \quad (3c)$$

in which $p = p(\rho, Y)$ denotes the pressure and $\phi = \phi(\rho, Y, v)$ denotes the slip velocity between the two phases. These algebraic closure laws are ‘‘sophisticated’’ functions used by oilfield engineers to match experimental data within a large scope of operating conditions. In practice, they are piecewise defined, even tabulated, and therefore highly expensive to evaluate.

System (2), referred to as the *Drift-Flux Model* (DFM), expresses the conservation laws associated with the mixture density, the total momentum and the gas density. It can be put under the abstract form

$$\partial_t \mathbf{w} + \partial_x \mathbf{f}(\mathbf{w}) = \mathbf{s}(\mathbf{w}), \quad (4)$$

using the vector notations

$$\mathbf{f}(\mathbf{w}) = \mathbf{w}v + \mathbf{g}(\mathbf{w}) \quad (5a)$$

$$\mathbf{g}(\mathbf{w}) = (0, P(\mathbf{w}), \sigma(\mathbf{w}))^T \quad (5b)$$

$$\mathbf{s}(\mathbf{w}) = (0, s(\mathbf{w}), 0)^T. \quad (5c)$$

Except for a few specific instances of the closure laws p and σ , nothing can be said in general about the *hyperbolicity* of (2). For $\sigma \equiv 0$, most notably, the DFM model (2) degenerates to the standard Euler model. The latter is known to be strictly hyperbolic provided that

$$\partial_\rho p(\rho, Y)|_Y > 0. \quad (6)$$

By analogy with this case, we require that

$$\partial_\rho P(\mathbf{w})|_{Y,v} > 0, \quad (7)$$

with a little abuse of notations regarding the arguments of P . When strict hyperbolicity occurs, it is usually observed that the three real eigenvalues

$$\lambda_-(\mathbf{w}) < \lambda_0(\mathbf{w}) < \lambda_+(\mathbf{w}), \quad (8)$$

of the Jacobian matrix $\nabla_{\mathbf{w}} \mathbf{f}(\mathbf{w})$ satisfy

$$|\lambda_0(\mathbf{w})| \ll |\lambda_-(\mathbf{w})| \approx |\lambda_+(\mathbf{w})|. \quad (9)$$

In other words, there are two scales of wave velocities with distinctive orders of magnitude: the fast acoustic wave λ_\pm and the slow kinematic wave λ_0 . To put (9) differently, the flow regime that most often prevails in the context of oil transportation corresponds to low Mach numbers.

1.2. Physical model in Arbitrary Lagrange-Euler coordinates

It is judicious to separate fast and slow waves in order to make positivity properties easier to be guaranteed. The full details of this approach can be found in [7]. Here, we simply give the core ideas.

We introduce a new referential frame, in which the coordinates are denoted by χ . This frame is neither the material (Lagrangian) configuration nor the laboratory (Eulerian) configuration x . Instead, it moves at the imposed speed $v - w$ with respect to the laboratory. Let $x = x(\chi, t)$ be the correspondence between the moving frame and the laboratory frame, and define

$$J = \partial_\chi x|_t \quad (10)$$

to be the dilatation rate. Then, it can be proven [13, 16] that system (4) is equivalent to

$$\partial_t(J) + \partial_\chi(w) - \partial_\chi(v) = 0 \quad (11a)$$

$$\partial_t(\mathbf{w}J) + \partial_\chi(\mathbf{w}w) + \partial_\chi(\mathbf{g}(\mathbf{w})) = J\mathbf{s}(\mathbf{w}) \quad (11b)$$

The above form highlights the role of the arbitrary velocity w , thanks to which we can now apply a splitting technique.

Let $\Delta t > 0$ denote the time step. Each iteration from time t^n to $t^{n+1} = t^n + \Delta t$ is made up of two steps

$$n \rightarrow n^\sharp \rightarrow n + 1. \quad (12)$$

(1) **The Lagrange step.** Starting from $J^n = 1$, we solve

$$\partial_t(J) - \partial_\chi(v) = 0 \quad (13a)$$

$$\partial_t(\mathbf{w}J) + \partial_\chi(\mathbf{g}(\mathbf{w})) = J\mathbf{s}(\mathbf{w}). \quad (13b)$$

Writing out all components, the above system reads

$$\partial_t(J) - \partial_\chi(v) = 0 \quad (14a)$$

$$\partial_t(\rho J) = 0 \quad (14b)$$

$$\partial_t(\rho v J) + \partial_\chi(P) = J s \quad (14c)$$

$$\partial_t(\rho Y J) + \partial_\chi(\sigma) = 0. \quad (14d)$$

By doing so, we have temporarily disregarded the convection due to w . However, as was the case for the initial system (4), no claim can be made about the hyperbolicity of the Lagrangian system (13). On the other hand, the difficulties associated with the closure laws remain identical.

In a series of previous works [1, 2, 10], it has been demonstrated that *relaxation* can be a handy tool to help us solve (13). In essence, relaxation amounts to replacing (14) by

$$\partial_t(J) - \partial_\chi(v) = 0 \quad (15a)$$

$$\partial_t(\rho J) = 0 \quad (15b)$$

$$\partial_t(\rho v J) + \partial_\chi(\Pi) = J s \quad (15c)$$

$$\partial_t(\rho \Pi J) + \partial_\chi(a^2 v) = 0 \quad (15d)$$

$$\partial_t(\rho Y J) + \partial_\chi(\Sigma) = 0 \quad (15e)$$

$$\partial_t(\rho \Sigma J) + \partial_\chi(b^2 Y) = 0, \quad (15f)$$

in which Π, Σ are the auxiliary variables representing $P(\mathbf{w})$ and $\sigma(\mathbf{w})$, while a, b are two positive parameters. For stability reasons, a and b must be subject to the Whitham conditions

$$a > \sqrt{-\partial_\tau P(\mathbf{u})|_{v,Y} + \left(\partial_v P(\mathbf{u})|_{\tau,Y}\right)^2} \quad (16a)$$

$$b > \left|\partial_Y \sigma(\mathbf{u})|_{\tau,v}\right|, \quad (16b)$$

where

$$\mathbf{u} = (\tau := \rho^{-1}, v, Y). \quad (17)$$

The interest of (15) lies in the fact that it is always hyperbolic. Indeed, its characteristic speeds are

$$-a\tau, -b\tau, 0, 0, b\tau, a\tau. \quad (18)$$

(2) **The projection step.** Starting from the intermediate state n^\sharp , we set $w = v$ and solve

$$\partial_t(J) + \partial_\chi(v) = 0 \quad (19a)$$

$$\partial_t(\mathbf{w}J) + \partial_\chi(\mathbf{w}v) = 0. \quad (19b)$$

By doing so, we recover the motion of the particles. Choosing $w = u$ yields $J^{n+1} = 1$ at the end of the step, and the mesh returns to its initial configuration. Multiplying (19a) by \mathbf{w} and subtracting to (19b), we end up with

$$\partial_t \mathbf{w} + \frac{v}{J} \partial_x \mathbf{w} = 0. \quad (20)$$

Since all components of \mathbf{w} are transported at the same characteristic speed v/J , this step is also called the *remap* step.

1.3. Numerical scheme

The two steps need to be fully explicit at the discrete level. We decompose the space domain into cells labeled by subscripts $j \in \mathbb{Z}$. The edges of the cells are designated by subscripts $j + 1/2$, while the size of cell j is Δx_j .

(1) **The Lagrange step.** Taking as granted that $J_j^n = 1$, we discretize (13) by

$$\frac{J_j^{n\sharp} - J_j^n}{\Delta t} - \frac{\tilde{v}_{j+1/2}^n - \tilde{v}_{j-1/2}^n}{\Delta x_j} = 0 \quad (21a)$$

$$\frac{(\mathbf{w}J)_j^{n\sharp} - (\mathbf{w}J)_j^n}{\Delta t} + \frac{\tilde{\mathbf{g}}_{j+1/2}^n - \tilde{\mathbf{g}}_{j-1/2}^n}{\Delta x_j} = (J\mathbf{s})_j^n, \quad (21b)$$

with the notation

$$\tilde{\mathbf{g}}_{j+1/2}^n = (0, \tilde{\sigma}_{j+1/2}^n, \tilde{P}_{j+1/2}^n)^T. \quad (22)$$

The flux values are retrieved by solving the Riemann problem associated with the relaxation system (15) at edge $j + 1/2$. This results in

$$\tilde{v}_{j+1/2}^n = \tilde{v}(\mathbf{w}_j^n, \mathbf{w}_{j+1}^n) = \frac{v_j^n + v_{j+1}^n}{2} - \frac{P_{j+1}^n - P_j^n}{2a_{j+1/2}} \quad (23a)$$

$$\tilde{P}_{j+1/2}^n = \tilde{P}(\mathbf{w}_j^n, \mathbf{w}_{j+1}^n) = \frac{P_j^n + P_{j+1}^n}{2} - a_{j+1/2} \frac{v_{j+1}^n - v_j^n}{2} \quad (23b)$$

$$\tilde{\sigma}_{j+1/2}^n = \tilde{\sigma}(\mathbf{w}_j^n, \mathbf{w}_{j+1}^n) = \frac{\sigma_j^n + \sigma_{j+1}^n}{2} - b_{j+1/2} \frac{Y_{j+1}^n - Y_j^n}{2}, \quad (23c)$$

where the positive parameters $a_{j+1/2}, b_{j+1/2}$ are subject to the discrete Whitham conditions

$$a_{j+1/2} > \max\{\sqrt{-P_\tau(\mathbf{u}_j^n) + (P_v(\mathbf{u}_j^n))^2}, \sqrt{-P_\tau(\mathbf{u}_{j+1}^n) + (P_v(\mathbf{u}_{j+1}^n))^2}\} \quad (24a)$$

$$b_{j+1/2} > \max\{|\sigma_Y(\mathbf{u}_j^n)|, |\sigma_Y(\mathbf{u}_{j+1}^n)|\}. \quad (24b)$$

Note that we have used the relaxation system (15) only as a virtual tool in order to derive the flux values (23). In reality, the additional terms Π and Σ are not computed by (15) but simply reset by $\Pi_j^{n\sharp} = P(\mathbf{w}_j^{n\sharp})$ and $\Sigma_j^{n\sharp} = \sigma(\mathbf{w}_j^{n\sharp})$.

(2) **The projection step.** Taking as granted that $w = \tilde{v}$, we discretize (19) by

$$\frac{J_j^{n+1} - J_j^{n\sharp}}{\Delta t} + \frac{\tilde{v}_{j+1/2}^n - \tilde{v}_{j-1/2}^n}{\Delta x_j} = 0 \quad (25a)$$

$$\frac{(\mathbf{w}J)_j^{n+1} - (\mathbf{w}J)_j^{n\sharp}}{\Delta t} + \frac{(\mathbf{w}\tilde{v})_{j+1/2}^{n\sharp} - (\mathbf{w}\tilde{v})_{j-1/2}^{n\sharp}}{\Delta x_j} = 0, \quad (25b)$$

with the upwind flux

$$(\mathbf{w}\tilde{v})_{j+1/2}^{n\sharp} = \mathbf{w}_j^{n\sharp}(\tilde{v}_{j+1/2})^+ + \mathbf{w}_{j+1}^{n\sharp}(\tilde{v}_{j+1/2})^-. \quad (26)$$

The velocity to be prescribed at the edges for the projection step is an outcome of the Lagrange step.

To see how the overall scheme combining both steps gives rise to a consistent approximation of the initial model (2), let us add (13) and (19). We obtain

$$\frac{J_j^{n+1} - J_j^n}{\Delta t} = 0 \quad (27a)$$

$$\frac{(\mathbf{w}J)_j^{n+1} - (\mathbf{w}J)_j^n}{\Delta t} + \frac{\tilde{\mathbf{f}}_{j+1/2}^{n\sharp} - \tilde{\mathbf{f}}_{j-1/2}^{n\sharp}}{\Delta x_j} = (J\mathbf{s})_j^n, \quad (27b)$$

where

$$\tilde{\mathbf{f}}_{j+1/2}^{n\sharp} = \mathbf{w}_j^{n\sharp}(\tilde{v}_{j+1/2}^n)^+ + \mathbf{w}_{j+1}^{n\sharp}(\tilde{v}_{j+1/2}^n)^- + \tilde{\mathbf{g}}_{j+1/2}^n. \quad (28)$$

Since $J_j^n = 1$, we have $J_j^{n+1} = 1$ and the resulting scheme

$$\frac{\mathbf{w}_j^{n+1} - \mathbf{w}_j^n}{\Delta t} + \frac{\tilde{\mathbf{f}}_{j+1/2}^{n\sharp} - \tilde{\mathbf{f}}_{j-1/2}^{n\sharp}}{\Delta x_j} = \mathbf{s}_j^n \quad (29)$$

clearly appears to be a discretization of the Drift-Flux Model via a consistent flux $\tilde{\mathbf{f}}_{j+1/2}^{n\sharp}$. This flux is about the only thing that will be passed to the multi-resolution and local time stepping subroutines.

1.4. CFL conditions and positivity properties

Besides the Whitham conditions (24), we impose various other CFL conditions to ensure stability.

(1) Since the non-zero characteristic speeds of the Lagrangian step are $\pm a\tau$ and $\pm b\tau$, we require

$$\frac{a_{j\pm 1/2}\Delta t}{\rho_j^n \Delta x_j} < \frac{1}{2} \quad (30a)$$

$$\frac{b_{j\pm 1/2}\Delta t}{\rho_j^n \Delta x_j} < \frac{1}{2}. \quad (30b)$$

In practice, $b \ll a$ (which is a consequence of $|\lambda_0| \ll |\lambda_{\pm}|$). Therefore, it is expected that only (30a) needs to be met.

(2) Since the characteristic speed of the projection step is $w = \tilde{v}$, we require

$$\frac{|\tilde{v}_{j\pm 1/2}^n|\Delta t}{\Delta x_j} < \frac{1}{2}. \quad (31)$$

The edge velocities $\tilde{v}_{j+1/2}^n$ are computable before updating the variables. Again, it is observed for most test cases in the low Mach regime that

$$|\tilde{v}_{j\pm 1/2}^n| \ll a_{j\pm 1/2}\tau_j^n. \quad (32)$$

Hence, it is likely that only (30a) would be saturated. However, the computational procedure takes into account the three conditions (30a), (30b), (31).

From (31), we infer that

$$\rho_j^{n\sharp} > 0. \quad (33)$$

Indeed, $\rho_j^{n\sharp} = (\tau_j^{n\sharp})^{-1}$ and

$$\tau_j^{n\sharp} = \tau_j^n - \frac{\Delta t}{\rho_j^n \Delta x_j} (\tilde{v}_{j+1/2}^n - \tilde{v}_{j-1/2}^n) \quad (34a)$$

$$= \tau_j^n \left(1 - \frac{\tilde{v}_{j+1/2}^n \Delta t}{\Delta x_j} + \frac{\tilde{v}_{j-1/2}^n \Delta t}{\Delta x_j} \right) \quad (34b)$$

$$> \tau_j^n \left(1 - \frac{|\tilde{v}_{j+1/2}^n| \Delta t}{\Delta x_j} - \frac{|\tilde{v}_{j-1/2}^n| \Delta t}{\Delta x_j} \right). \quad (34c)$$

The last right-hand side is obviously positive under (31). Furthermore, we have

$$\rho_j^{n+1} > 0. \quad (35)$$

Indeed, multiplying (25a) by $\mathbf{w}_j^{n\sharp}$ and subtracting the product to (25b), we end up with

$$\mathbf{w}_j^{n+1} = \mathbf{w}_j^{n\sharp} - \frac{\Delta t}{\Delta x_j} ((\tilde{v}_{j-1/2}^n)^+ (\mathbf{w}_j^{n\sharp} - \mathbf{w}_{j-1}^{n\sharp}) + (\tilde{v}_{j+1/2}^n)^- (\mathbf{w}_{j+1}^{n\sharp} - \mathbf{w}_j^{n\sharp})) \quad (36a)$$

$$= \left(1 - \frac{(\tilde{v}_{j-1/2}^n)^+ \Delta t}{\Delta x_j} + \frac{(\tilde{v}_{j+1/2}^n)^- \Delta t}{\Delta x_j} \right) \mathbf{w}_j^{n\sharp} + \frac{(\tilde{v}_{j-1/2}^n)^+ \Delta t}{\Delta x_j} \mathbf{w}_{j-1}^{n\sharp} - \frac{(\tilde{v}_{j+1/2}^n)^- \Delta t}{\Delta x_j} \mathbf{w}_{j+1}^{n\sharp}. \quad (36b)$$

But

$$1 - \frac{(\tilde{v}_{j-1/2}^n)^+ \Delta t}{\Delta x_j} + \frac{(\tilde{v}_{j+1/2}^n)^- \Delta t}{\Delta x_j} \geq 1 - \frac{|\tilde{v}_{j-1/2}^n| \Delta t}{\Delta x_j} - \frac{|\tilde{v}_{j+1/2}^n| \Delta t}{\Delta x_j} > 0. \quad (37)$$

This implies that (36b) is a convex combination. Looking at the first component, this means that ρ_j^{n+1} is a convex combination of $\rho_{j-1}^{n\sharp}$, $\rho_j^{n\sharp}$ and $\rho_{j+1}^{n\sharp}$, all of which are positive by (33). By a similar (but lengthier) technique, we are in a position to establish that

$$Y_j^{n+1} \in [0, 1]. \quad (38)$$

2. FULLY ADAPTIVE MULTIREOLUTION SCHEME

We start this section with a brief description of the space adaptive scheme designed in [6]. We refer to [10] and [1] for details and numerical examples in the context of diphasic flows. A more thorough study of the adaptive method for the Lagrange projection formulation of the simplified model (without drift) is proposed in [8]. In the second paragraph we describe how we adapt the local time stepping strategy proposed by [20] to our problem where the duration of the local time step is controlled by the solution.

2.1. Space adaptivity

We consider a uniform mesh with step size Δx_0 , whose cells are denoted by $\Omega_{0,j}$. Starting from this coarsest discretization labeled 0 we define a hierarchy of $K + 1$ nested grids $(S_k)_{k=0,\dots,K}$ by dyadic refinement, with cells $\Omega_{k,j} = \Omega_{k+1,2j} \cup \Omega_{k+1,2j+1}$ of size $\Delta x_k = 2^{-k} \Delta x_0$.

Initially, the piecewise constant vector-valued function \mathbf{w} is defined on the finest grid where it is represented by the sequence of its mean values $\mathbf{w}_K = (\mathbf{w}_{K,j})_j$ on the cells $\Omega_{K,j}$. The coarsening or projection operator P_{k+1}^k consists in cell averaging from one grid to the coarser one. The inverse operator P_k^{k+1} predicts the mean values on grid level $k + 1$, given the mean values on the coarser level k . The reconstruction or prediction

operator enjoys several useful properties. It is local, its stencil width is $2s + 1$. It is exact for polynomials of degree lower or equal to $2s$. By construction it preserves the mean values, *i.e.* $P_{k+1}^k P_k^{k+1} = Id$. Thanks to this consistency property the difference between the exact and predicted values on two subdivisions of a coarse cell are symmetric. We define the detail vector \mathbf{d}_{k+1} as the value of this difference $\mathbf{w}_{k+1} - P_k^{k+1} \mathbf{w}_k$ on the left-hand side subdivisions on level $k + 1$.

The two vectors \mathbf{w}_{k+1} and $(\mathbf{w}_k, \mathbf{d}_{k+1})$ are of same length and we can use \mathbf{d}_{k+1} along with \mathbf{w}_k and the prediction operator P_k^{k+1} to recover \mathbf{w}_{k+1} entirely. Iterating this encoding operation from the finest level down to the coarsest provides the multiscale representation $\mathbf{m} = (\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_K) = \mathcal{M} \mathbf{w}_K$, where the notation has been extended to $\mathbf{d}_0 = \mathbf{w}_0$. The indices of the multiscale representation \mathbf{m} vary in

$$\nabla^K = \{[0, \dots, N_0], [0, \dots, N_0], \dots, [0, \dots, 2^{K-1} N_0]\}.$$

Thanks to the polynomial exactness of the prediction operator P_k^{k+1} , a small detail in absolute value means that the solution can be locally approached by a polynomial, quadratic in our case ($s = 1$). We can use this property to design a grid where the size of the mesh varies and is adapted to the local smoothness of the solution. Given level-dependent threshold values $\varepsilon = (\varepsilon_k)_{k=0, \dots, K}$, we introduce the subset of ∇^K

$$\Gamma_\varepsilon = \Gamma(\varepsilon_0, \dots, \varepsilon_K) := \{\gamma \text{ s.t. } |\mathbf{d}_\gamma| \geq \varepsilon_{|\gamma|}\}, \quad \text{where } |\gamma| = |(k, j)| = k. \quad (39)$$

For practical purposes it is necessary to enlarge this set of significant details into a *gradual tree*, which is defined recursively by

$$(i, k) \in \Gamma \rightarrow \left(\left\lfloor \frac{i}{2} \right\rfloor + l, k - 1 \right) \in \Gamma, \quad |l| \leq g.$$

Choosing the grading parameter $g = 1$ ensures that the stencil of the prediction operator P_k^{k+1} always belongs to the tree. We can then define an adaptive grid S_ε where the local size of the cell will be the grid step corresponding to the finest non negligible detail

$$S_\varepsilon = \{(k, j), k \in \{0, \dots, K\}, j \in \{0, \dots, 2^k N_0\}, \text{ s.t. } (k, \lfloor j/2 \rfloor) \in \Gamma_\varepsilon \\ \text{and } (k + 1, j) \notin \Gamma_\varepsilon\}, \quad (40)$$

where $\lfloor j \rfloor$ denotes the integer part of j . The reverse decoding operator \mathcal{M}^{-1} is then partially applied to obtain a discretized representation on the adaptive grid S_ε . The idea is to apply the finite volume scheme (29) discretizing the evolution operator on this object. Relying on the hyperbolicity of the system of PDEs, it is possible to predict the propagation of the fronts which move at finite speed, and also their formation to design a grid suitable to represent the solution at two consecutive times steps t^n and t^{n+1} — that is satisfying the thresholding condition (39). The prediction of the refined tree $\tilde{\Gamma}_\varepsilon^{n+1}$ is actually the bottleneck of the method. In practice we use the heuristic rules of Harten [15] and the complete adaptive algorithm goes as follows:

Algorithm 1 Adaptive algorithm

Initialization: encoding of the initial solution using P_{k+1}^k and definition of Γ_ε^0 using (39).

for time step $n = 0, \dots, N - 1$ **do**

 Prediction of $\tilde{\Gamma}_\varepsilon^{n+1} \supset \Gamma_\varepsilon^n \cup \tilde{\Gamma}_\varepsilon^{n+1}$.

 Partial decoding of $\mathbf{w}^{\varepsilon, n}$ on $\tilde{S}_\varepsilon^{n+1}$ using P_k^{k+1} on (40).

 Evolution of $\mathbf{w}^{\varepsilon, n}$ to $\mathbf{w}^{\varepsilon, n+1}$ on the adaptive grid $\tilde{S}_\varepsilon^{n+1}$ by applying (29).

 Partial encoding of $\mathbf{w}^{\varepsilon, n+1}$ using P_{k+1}^k and definition of Γ_ε^{n+1} using (39) $\mathbf{w}^{\varepsilon, n+1}$ and thresholding.

end for

Decoding of $\mathbf{w}^{\varepsilon, N}$ on the finest grid S_K .

In this adaptive algorithm the time step Δt appearing in the evolution scheme (29) is controlled by stability conditions and to emphasize its dependency on the solution we now denote it as

$$\Delta t^n \leq \min_{(k,j) \in \tilde{S}_\varepsilon^{n+1}} \left(\frac{1}{2} \frac{\Delta x_k}{|\tilde{v}_{k,j\pm 1/2}^n|}, \frac{\Delta x_k \rho_{k,j}^n}{a_{k,j\pm 1/2}} \right), \quad (41)$$

where $\tilde{v}_{k,j\pm 1/2}^n$ is the speed at the interface computed by (23a) and $a_{k,j\pm 1/2}$ satisfies the Whitham condition (24). The time step Δt^n is set globally for the whole mesh. It therefore depends on the size of the smallest cell belonging to $\tilde{S}_\varepsilon^{n+1}$. In practical applications using a hierarchy of K grids, the ratio between the largest and smallest cells is 2^K . This means that in the smooth areas of the domain where the solution is discretized on coarsest cells it is updated with a time step 2^K times smaller than the limit set locally by the stability conditions (30,31). This obviously infers a loss in computing efficiency. In the next section, we describe how to implement a local time stepping strategy to use a spatially varying time step that is locally adapted to the size of the cell.

2.2. Local time stepping and time step control

In our case the multiresolution scheme is based on a hierarchy of nested dyadic grids. The major ingredient of a Local Time Stepping (LTS) method is therefore the design of a macro time step, suitable for the coarsest level cells, along with a cascade of intermediate time steps at which the solution defined on finer levels is updated in turns.

There are three important issues to consider in the design of such an algorithm:

- (1) Control and update the length of the local time step so that the stability condition (41) is locally satisfied.
- (2) Ensure the synchronization of the solution in a conservative way at interfaces between cells of different size.
- (3) Perform partial regridding at intermediate time steps in order to minimize the predicted refinement of the adaptive grid.

The first point is motivated by the first section, where a lot of effort has been spent in the design of a stability condition on the time step ensuring natural physical bounds of the density and the gas mass fraction. We will illustrate in the numerical simulations in the next section that it is important to make the intermediate time steps variable and dependent on the solution evolution. The way we propose to do this is based on a generalization of the algorithm proposed in [20], which assumes a dyadic subdivision of the macro time step to be used on the coarsest level. The two other points are treated in [20] and we will briefly rephrase them in the context of our variable time step implementation.

The Figure 1 illustrates the macro time step subdivision in the case of a four-level hierarchy. In the general case, the macro time step $\Delta t^n = t^{n+1} - t^n$ is subdivided in a hierarchical way mimicking the spatial dyadic subdivision. Starting from $\Delta t^n = \Delta t_0^{n,0}$, the time steps to be used at the different levels are defined by recursive subdivision

$$\Delta t_k^{n,i} = \Delta t_{k+1}^{n,2i} + \Delta t_{k+1}^{n,2i+1}. \quad (42)$$

The intermediate time steps $\Delta t_k^{n,i}$ will be used to update the cells active at level k at the intermediate times $t_k^{n,i}$

$$t_k^{n,i+1} = t_k^{n,i} + \Delta t_k^{n,i}, \text{ for } i = 0, \dots, 2^k - 1,$$

with the convention

$$t^n = t_k^{n,0} \text{ and } t^{n+1} = t_k^{n+1,2^k}.$$

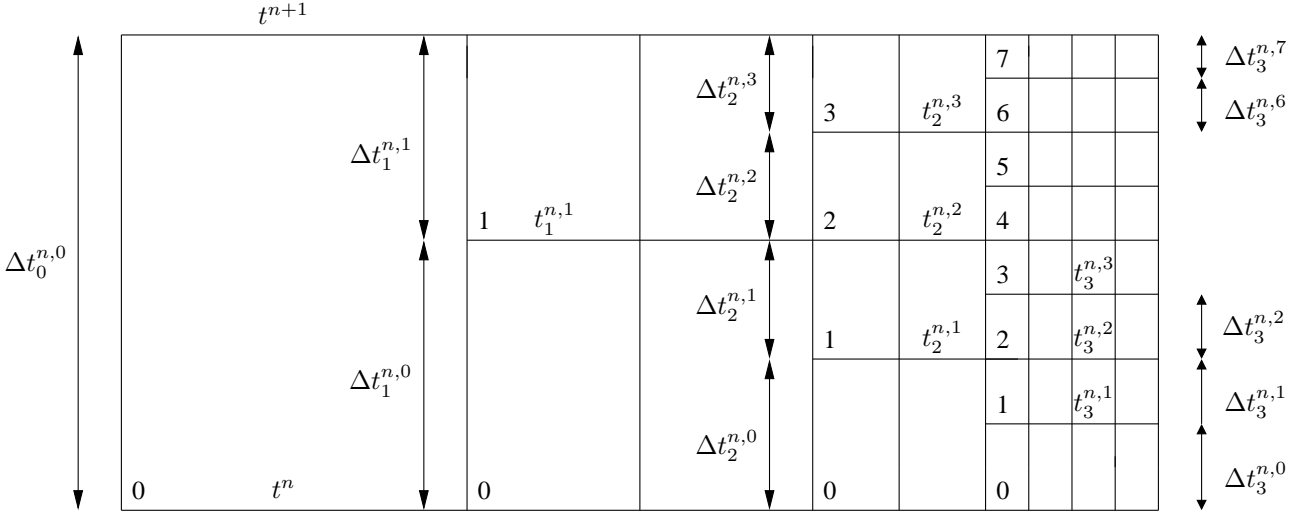


FIGURE 1. A four-level adaptive grid, with the level dependent intermediate time step numbering

Note here that not all levels are concerned by the updating process at a given intermediate time $t_K^{n,i}$. To make this precise, we define at each intermediate step $i = 1, \dots, 2^K$ a synchronization level k_i

$$k_i := \min\{k, 0 \leq k \leq K, i \bmod 2^{K-k} = 0\}. \quad (43)$$

It indicates the coarsest level above which cells are synchronized at intermediate time $t_K^{n,i}$. The cells at level k , with $k_i \leq k \leq K$ are updated from their previous state at time $t_k^{n,q}$ using the appropriate time step $\Delta t_k^{n,q}$ with

$$q(i, k) = \left\lfloor \frac{i-1}{2^{K-k}} \right\rfloor.$$

We now address the actual evaluation of the micro time step $\Delta t_K^{n,i}$ for $i = 1, \dots, 2^K$ which must vary in time, that is within the macro time step, so that the bound (41) is satisfied by all intermediate times steps $\Delta t_k^{n,i}$ for $i = 1, \dots, 2^k$.

We denote by $\mathcal{F}_{n,i}$ the set of indices of cells where fluxes must be updated before synchronization of the solution at the intermediate time $t_K^{n,i}$

$$\mathcal{F}_{n,i} = \left\{ (k, j) \in \tilde{S}_\varepsilon^{n+1}, \text{ for } k \geq k_{i-1} \right\}.$$

Then the first micro time step is bounded by

$$\Delta t_K^{n,0} \leq \Delta x_K \min_{\substack{(k,j) \in \mathcal{F}_{n,0} \\ 0 \leq k \leq K}} \min \left(\frac{1}{2 |\tilde{v}_{k,j \pm 1/2}^n|}, \frac{\rho_{k,j}^n}{a_{k,j \pm 1/2}} \right) \quad (44)$$

and the subsequent micro time steps are designed to be non increasing

$$\Delta t_K^{n,i} = \min \left(\Delta t_K^{n,i-1}, \Delta x_K \min_{\substack{(k,j) \in \mathcal{F}_{n,i} \\ 0 \leq k \leq K}} \min \left(\frac{1}{2 |\tilde{v}_{k,j \pm 1/2}^n|}, \frac{\rho_{k,j}^n}{a_{k,j \pm 1/2}} \right) \right).$$

The intermediate time steps $\Delta t_k^{n,i}$ are then built up adaptively by summing the micro time steps with the recursion (42). With these notations we illustrate the synchronization method on an example in the case of

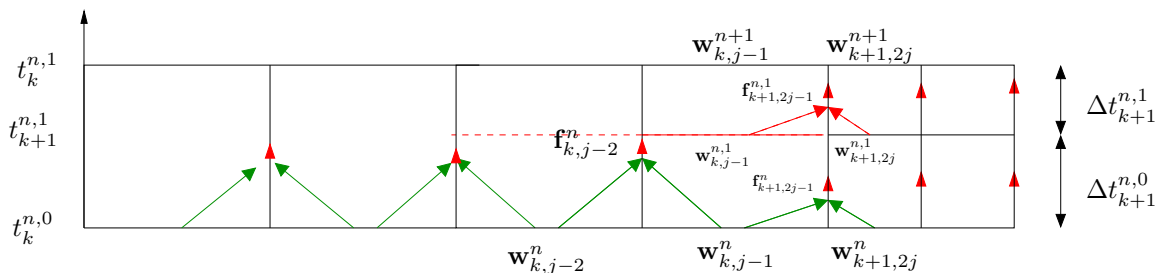


FIGURE 2. Local Time Step method for two refinement levels. The solution discretized on level $k + 1$ is updated in two micro time steps. The solution discretized on level k is updated in a single macro time step everywhere except in the transition zone (here of width 2).

a two-level hierarchy represented in Figure 2. The little arrows on the vertical lines indicate which fluxes must be computed. The part of the solution discretized on the fine level (numbered $k + 1$) is first updated from $\mathbf{w}_{k+1,2j}^{n,0}$ to $\mathbf{w}_{k+1,2j}^{n,1}$ using the micro time step $\Delta t_{k+1}^{n,1}$. The data necessary to compute the flux $\mathbf{f}_{k+1,2j-1}^{n,1}$ must be synchronized at time $t_{k+1}^{n,1}$. The neighbor cells $(k, j - 1)$ and $(k, j - 2)$ on the left of the cell $2j$ must therefore also be updated at the intermediate time $t_{k+1}^{n,1}$ even though they belong to the coarser level k .

The intermediate values $\mathbf{w}_{k,j-1}^{n,1}$ and $\mathbf{w}_{k,j-2}^{n,1}$ are then used along with $\mathbf{w}_{k+1,2j}^{n,1}$ and $\mathbf{w}_{k+1,2j+1}^{n,1}$ to compute the flux $\mathbf{f}_{k+1,2j-1}^{n,1}$.

The fluxes $\mathbf{f}_{k,j-2}^n$ and $\mathbf{f}_{k,j-1}^n$ remain valid for the macro time step at the interface of cells $(k, j - 2)$ and $(k, j - 1)$ and are used one more time to update $\mathbf{w}_{k,j-2}^{n,1}$ and $\mathbf{w}_{k,j-1}^{n,1}$ to their final value $\mathbf{w}_{k,j-2}^{n+1}$ and $\mathbf{w}_{k,j-1}^{n+1}$.

This is formalized by designing for each level $k = 0, \dots, K - 1$ a *transition zone* \bar{C}^k containing the cells on level k which belong to the stencil of the evolution scheme (29) of cells on level $k + 1$.

Note that this scheme is consistent but induces an additional $O(\Delta t^2)$ error, except for first order two-point stencil fluxes. The costly part in the algorithm is the flux evaluation itself, since the update of the solution in the form (29) consists in algebraic operations. A simpler way of implementing the LTS algorithm, suggested in [19], consists therefore in updating all the cells at each intermediate time step using the last computed flux value, and not only the cells belonging to the transition zone. The extra computations done in updating the coarse cells at each intermediate time instead of a single time step should be balanced by the economy in book-keeping provided by this simplification. Our implementation is based on the original strategy proposed in [20] where the cells outside transition zones are updated only when needed with the time step adapted to their size.

The overall LTS algorithm can be summarized as follows

Algorithm 2 Local Time Stepping inner loop at time t_n

for intermediate time step $t_K^{n,i}$, $i = 1, \dots, 2^K$ **do**
 Update fluxes on cells that have been updated at $t_K^{n,i-1}$.
 Compute $\Delta t_k^{n,q(i,k)}$ for $k = K \setminus k_i$.
for levels $k = K \setminus k_i$ **do**

$$\mathbf{w}_{k,j}^{n,i} = \mathbf{w}_{k,j}^{n,q(i,k+1)} - \frac{\Delta t_{k+1}^{n,q(i,k+1)}}{\Delta x_k} \left(\mathbf{f}_{k,j+1/2}^{n,q(i,k+1)} - \mathbf{f}_{k,j-1/2}^{n,q(i,k+1)} \right)$$
 if $(k,j) \in \bar{C}_k$

$$\mathbf{w}_{k,j}^{n,i} = \mathbf{w}_{k,j}^{n,q(i,k)} - \frac{\Delta t_k^{n,q(i,k)}}{\Delta x_k} \left(\mathbf{f}_{k,j+1/2}^{n,q(i,k)} - \mathbf{f}_{k,j-1/2}^{n,q(i,k)} \right)$$
 otherwise
end for

for $(k,i) \in \bar{C}_{k_i-1}$ **do**

$$\mathbf{w}_{k,j}^{n,i} = \mathbf{w}_{k,j}^{n,q(i,k+1)} - \frac{\Delta t_{k+1}^{n,q(i,k+1)}}{\Delta x_k} \left(\mathbf{f}_{k,j+1/2}^{n,q(i,k+1)} - \mathbf{f}_{k,j-1/2}^{n,q(i,k+1)} \right)$$

end for
end for

The important point in this algorithm is the direction of the inner loop on the discretization levels to perform the solution update. The solution on the coarser levels is updated last, and this makes it possible to compute the corresponding time step so that it adds up to the sum of its intermediate time subdivisions. This adaptive building up of the macro time step is made possible by the synchronization method original to the chosen approach. Other LTS schemes are based on predictor corrector methods like the one proposed by Osher *et al* [22] or by Domingues *et al* [11] in the context of multiple stages Runge-Kutta schemes. In these approaches the solution in the transition zones on level k is updated first to its own synchronization time (that would be $\mathbf{w}_{k,j-1}^{n+1}$ and $\mathbf{w}_{k,j-2}^{n+1}$ in our two-level example), and the data needed to update the solution on level $k+1$ is interpolated at time $t_{k+1}^{n,1}$ ($\mathbf{w}_{k,j-1}^{n,1}$ and $\mathbf{w}_{k,j-2}^{n,1}$). These strategies require to know the complete time step decomposition of the macro time step before starting to update the solution.

It is easy to check from the definition (43) or by looking at the time-space hierarchy displayed in Figure 1 that $k_i = K$ when i is odd. The partial regridding mentioned at the beginning of the paragraph is therefore performed only after even time steps on levels between $k_i + 1$ and K .

3. NUMERICAL SIMULATIONS

We illustrate the numerical method designed in the previous paragraph on a realistic test case modeling a 4km long pipeline. A constant flow mixture of gas and oil is injected at the inlet for 100s. Then the gas flow increases linearly for 100s from $10\text{kg/m}^2/\text{s}$ to 20kg/m^2 . After $t = 200\text{s}$ it is kept constant until the end of simulation which lasts 2500s. The oil flow is kept constant equal to $1000\text{kg/m}^2/\text{s}$ throughout the simulation. The pressure at the outlet of the pipe is also kept constant equal to 10 bars. We use the thermodynamical and hydrodynamical closure laws (incompressible liquid and Zuber-Findlay) described in [21]. We compare the numerical solutions obtained using the standard global time multiresolution and the local time stepping scheme, on a 5-level hierarchy starting with 256 cells on the finest level ($\Delta x = 15.625\text{m}$). The threshold parameter ε is set to 10^{-3} . The curves on Figure 3 display the physical quantities of interest: gas mass fraction, gas mass flow rate, liquid mass flow rate and pressure as a function of time. The 5 different curves on the same graph correspond to 5 different measurement points positioned at $x = 0, \dots, 4\text{km}$ in the pipe. The two schemes give qualitatively the same results. We compare them from the point of view of the time step monitoring and gain in computing time.

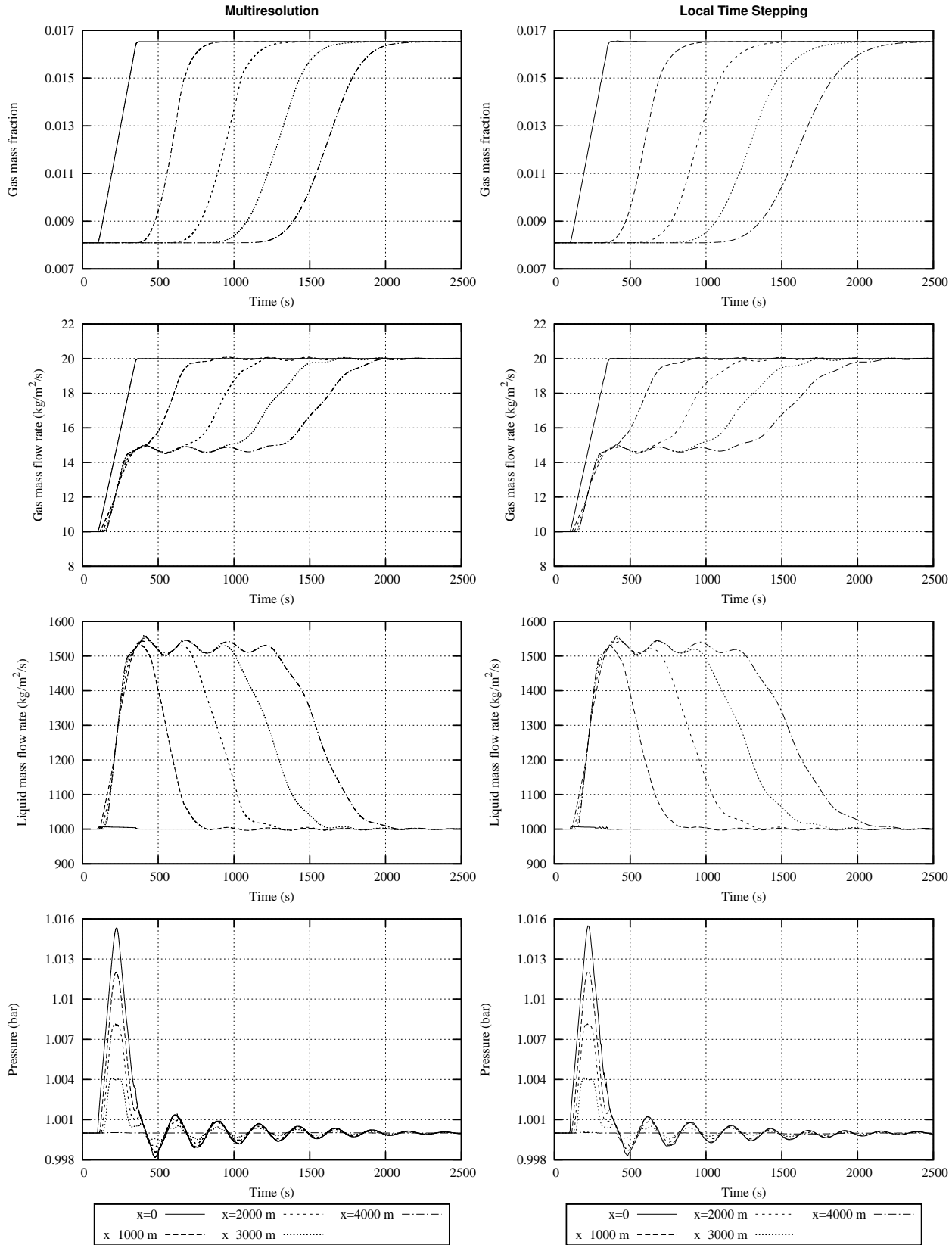


FIGURE 3. Time variations of the gas mass fraction, flow rates and pressure at five measurement points in the pipeline. Multiresolution (left) and Local Time Stepping (right). Explicit 1st order scheme, 256 cells, 5 levels, $\varepsilon = 10^{-3}$

3.1. Time step evolution

Due to the strong nonlinearity of the model and its coupled vector structure there are no invariant regions in the phase space of the solution. It is therefore impossible to predict a time step from the initial condition which would remain suitable throughout the computation. Actually, in the chosen test case, the stiff changes in input flow at the inlet of the pipe transmits itself to the flow inside the pipe and induces changes in the stability condition on the time step. The evolution of the time step monitored by the stability condition is displayed in Figure 4. It exhibits a region where it decreases, corresponding to the increase in gas input. Towards the end of the computation it starts to increase gently. We take a closer look at the evolution of the time step in those two regions. In the decreasing region between 180s and 200s, thanks to the control on the micro time step, the time step of both adaptive schemes stays close to the time step of the uniform scheme. For sake of comparison we also display the time step evolution if the micro time step is kept constant during each macro time step. This would obviously lead to violate the stability condition. In the other region of interest — after 1200s — the time step monitored by the stability condition increases gently in the uniform and global time stepping multiresolution case. The micro-time step is forbidden to increase during a macro time step and this leads to a gentler increase than in the uniform case. Yet the LTS scheme time step eventually catches up with the uniform evolution which further corroborates the robustness of our time stepping strategy.

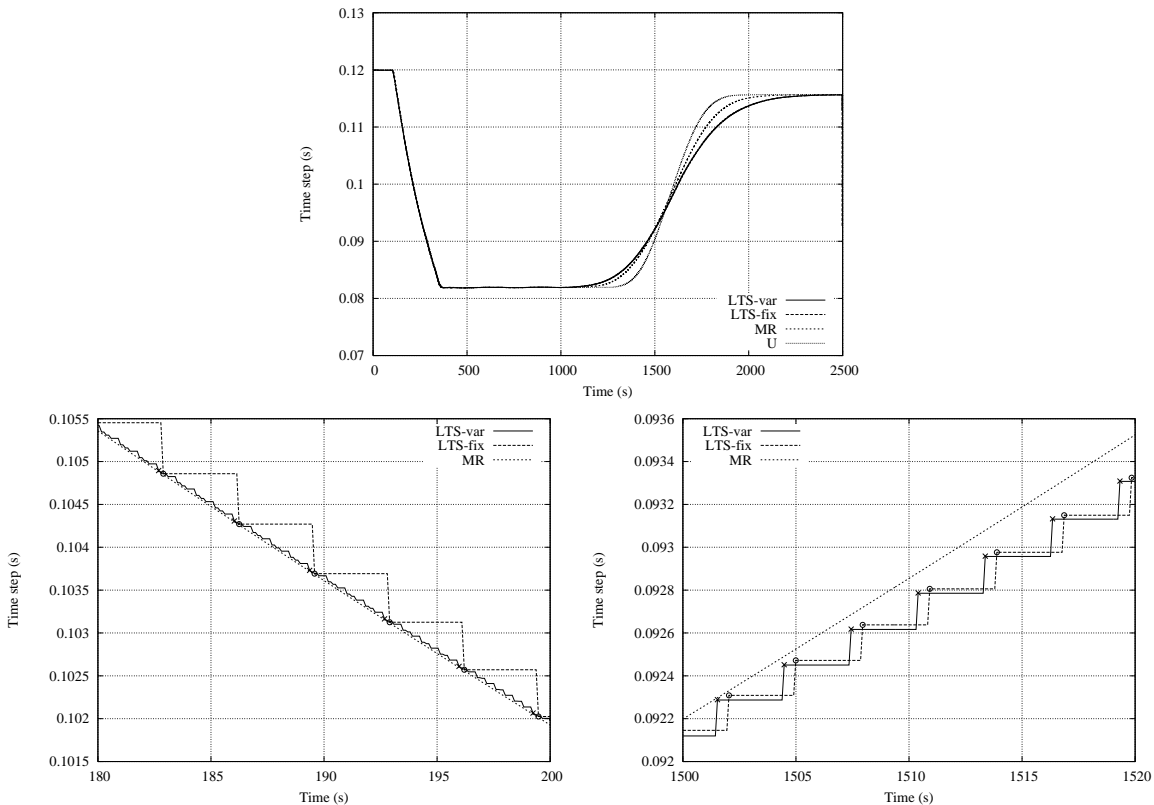


FIGURE 4. Evolution of the time step during the simulation with the different strategies.

3.2. Gains in CPU time

To demonstrate the performance of the multiresolution schemes both with global and local time stepping strategies we compute the gains in computing time. We also compute the ratio between the number of calls to the state laws using the different strategies. This indicator is a measure of the adaptive grid complexity and should be correlated to the gain in CPU time. Another equivalent indicator could be the ratio between the grid sizes. Computations corresponding to 256, 512, and 1024 cells on the finest level and 8 cells on the coarsest one are compared. The *simple* Zuber-Findlay hydrodynamical law used to obtain the results in Figure 3 is compared with a more *realistic* one. This so-called “synthetic “ state law reproduces the experimental tabular law provided by oil field measurements but is much more practicable to use in a transient computation. In Table 1, which summarizes the performances, N is the number of cells on the finest level and K is the number of levels in the MR hierarchy. The gain in CPU is denoted by R_{CPU} while R_{law} is the ratio between the number of calls to state laws. As expected, the gain in computing time increases with the number of cells and the number of levels. The improvement due to the local time stepping strategy is present for all simulations but is much higher when using the costly state law. The most interesting feature

$N(K)$	Ratio UNI/GTS				Ratio GTS/LTS			
	<i>Simple</i> law		<i>realistic</i> law		<i>Simple</i> law		<i>realistic</i> law	
	R_{CPU}	R_{law}	R_{CPU}	R_{law}	R_{CPU}	R_{law}	R_{CPU}	R_{law}
256(5)	2.98	2.90	3.17	3.07	6.61	11.67	10.41	10.82
512(6)	4.19	4.64	5.02	4.85	7.77	18.79	16.13	17.06
1024(7)	5.47	6.99	7.48	7.13	8.39	29.20	24.58	25.88

TABLE 1. Ratio between CPU times and number of calls to state laws - Uniform (U) versus Global Time Stepping (GTS) and Global Time Stepping (GTS) versus Local Time Stepping strategy (LTS).

which arises from this table is the correlation between R_{CPU} and R_{law} . It is good in the GTS versus uniform case for both state laws because the overlay due to adaptive grid generation is light enough. In the LTS versus GTS case the correlation is bad for the simple state law and good for the costly synthetic law. Actually, the additional book-keeping due to LTS is compensated in terms of CPU gain only when the cost of the actual numerical scheme is sufficiently high.

Notwithstanding the influence of the state laws, the CPU gains provided by successive scheme enhancements Uniform \rightarrow GTS and GTS \rightarrow LTS are always significant. These are important criteria when dealing with industrial applications where the investment necessary to introduce adaptivity in the existing methods and codes must be justified and motivated by strong efficiency expectations.

REFERENCES

- [1] N. ANDRIANOV, F. COQUEL, M. POSTEL, AND Q. H. TRAN, *A relaxation multiresolution scheme for accelerating realistic two-phase flows calculations in pipelines*, Int. J. Numer. Meth. Fluids, 54 (2007), pp. 207–236.
- [2] M. BAUDIN, C. BERTHON, F. COQUEL, R. MASSON, AND Q. H. TRAN, *A relaxation method for two-phase flow models with hydrodynamic closure law*, Numer. Math., 99 (2005), pp. 411–440.
- [3] M. BENDAHDANE, B. R., R. RUIZ, AND K. SCHNEIDER, *Appl. Num. Math.*, 59 (2009), pp. 1668–1692.
- [4] M. J. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys., 53 (1984), pp. 484–512.
- [5] A. COHEN, *Numerical analysis of wavelet methods*, vol. 32 of Studies in Mathematics and its Applications, North-Holland Publishing Co., Amsterdam, 2003.
- [6] A. COHEN, M. S. KABER, S. MÜLLER, AND M. POSTEL, *Fully adaptive multiresolution finite volume schemes for conservation laws*, Math. Comp., 72 (2003), pp. 183–215.
- [7] F. COQUEL, Q. L. NGUYEN, M. POSTEL, AND Q. H. TRAN, *Entropy-satisfying relaxation method with large time-steps for Euler IBVPs*. To appear in Math. of Comp. , prepublication LJLL R08004, 2007.

- [8] ———, *Local time stepping applied to implicit-explicit methods for hyperbolic systems*. To appear in Multiscale Modeling and Simulation, prepublication LJLL R07058, 2007.
- [9] F. COQUEL, Q. L. NGUYEN, M. POSTEL, AND Q. H. TRAN, *Local time stepping for implicit-explicit methods on time varying grids*, in Proceedings of the 7th European Conference on Numerical Mathematics and Advanced Applications (ENUMATH), Graz, septembre 2007, O. S. K. Kunisch, G. Of, ed., Berlin, 2008, Springer-Verlag, pp. 257–264.
- [10] F. COQUEL, M. POSTEL, N. POUSSINEAU, AND Q. H. TRAN, *Multiresolution technique and explicit-implicit scheme for multicomponent flows*, J. Numer. Math., 14 (2006), pp. 187–216.
- [11] M. O. DOMINGUES, S. M. GOMES, O. ROUSSEL, AND K. SCHNEIDER, *An adaptive multiresolution scheme with local time stepping for evolutionary PDEs*, J. Comput. Phys., 227 (2008), pp. 3758–3780.
- [12] ———, *Space-time adaptive multiresolution methods for hyperbolic conservation laws: Applications to compressible euler equations*, Appl. Num. Math., 59 (2009), pp. 2303–2321.
- [13] J. DONEA, A. HUERTA, J. P. PONTOT, AND A. RODRÍGUEZ-FERRAN, *Arbitrary Lagrangian-Eulerian methods*, in Encyclopedia of Computational Mechanics, E. Stein, R. de Borst, and T. Hughes, eds., vol. 1, John Wiley & Sons, 2004, ch. 14, pp. 413–437.
- [14] I. FAILLE, F. NATAF, F. WILLIEN, AND S. WOLF, *Local time steps for a finite volume scheme*. Submitted 2007, prepublication LJLL R08051, <http://www.ann.jussieu.fr/prepublication.php3?M=prepub>, 2008.
- [15] A. HARTEN, *Multiresolution algorithms for the numerical solutions of hyperbolic conservation laws*, Comm. Pure Appl. Math., 48 (1995), pp. 1305–1342.
- [16] C. W. HIRT, A. A. AMSDEN, AND J. L. COOK, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, J. Comput. Phys., 14 (1974), pp. 227–253.
- [17] P. LAMBY, S. MÜLLER, AND Y. STIRIBA, *Solution of shallow water equations using fully adaptive multiscale schemes*, Int. J. Numer. Meth. Fluids, 49 (2005), pp. 417–437.
- [18] S. MÜLLER, *Adaptive Multiscale Schemes for Conservation Laws*, vol. 27 of Lecture Notes on Computational Science and Engineering, Springer, 2002.
- [19] S. MÜLLER, P. HELLUY, AND J. BALLMANN, *Numerical simulation of a single bubble by compressible two-phase fluids*, Int. Journal for Numerical Methods in Fluids, (2009). DOI: 10.1002/fld.2033. Preprint : report No. 273, February 2007 IGPM, RWTH Aachen.
- [20] S. MÜLLER AND Y. STIRIBA, *Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping*, J. Sci. Comput., 30 (2007), pp. 493–531.
- [21] Q. L. NGUYEN, *Adaptation dynamique de maillage pour les écoulements diphasiques en conduites pétrolières. Application à la simulation de phénomènes de terrain slugging et severe slugging*, PhD thesis, Université Pierre et Marie Curie (Paris VI), 2009.
- [22] S. OSHER AND R. SANDERS, *Numerical approximations to nonlinear conservation laws with locally varying time and space grids*, Math. Comp., 41 (1983), pp. 321–336.
- [23] H.-Z. TANG AND G. WARNECKE, *A class of high resolution schemes for hyperbolic conservation laws and convection-diffusion equations with varying time and space grids*, SIAM J. Sci. Comput., 26 (2005), pp. 1415–1431.