INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **54**:207–236
Published online 7 December 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1397

A relaxation multiresolution scheme for accelerating realistic two-phase flows calculations in pipelines

N. Andrianov[1], F. Coquel[2], M. Postel[2,*,†] and Q. H. Tran[1]

[1] *Département Mathématiques Appliquées, Institut Français du Pétrole, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex, France*
[2] *Laboratoire Jacques-Louis Lions, CNRS et Université Pierre et Marie Curie, B.C. 187, 75252 Paris Cedex 05, France*

SUMMARY

We wish to demonstrate that it is judicious to combine various existing computational techniques that appeared for academic cases in seemingly unrelated areas, namely, semi-implicit relaxation schemes for hyperbolic systems and adaptive multiresolution algorithms, in order to achieve fast and accurate simulations of realistic two-phase flows problems in oil transportation. By 'realistic' we mean problems that are modelled by partial differential equation (PDE) systems closed by sophisticated thermodynamics and hydrodynamics laws, set out over a terrain-induced geometry and associated with time-dependent boundary conditions. Although the combination of these techniques is not a straightforward matter, it is made possible *via* a careful examination of the objectives of the simulation problem and suitable adaptations of which we shall give the details. Significant benchmarks demonstrate the efficiency of the proposed method. Copyright © 2006 John Wiley & Sons, Ltd.

Received 23 June 2006; Revised 4 October 2006; Accepted 5 October 2006

KEY WORDS: finite volumes; multiresolution; two-phase flows

1. INTRODUCTION

We are interested in accelerating the computation of realistic two-phase flows in pipelines, while keeping an acceptable degree of accuracy. Our primary application is oil transportation. Before

*Correspondence to: M. Postel, Laboratoire Jacques-Louis Lions, CNRS et Université Pierre et Marie Curie, B.C. 187, 75252 Paris Cedex 05, France.
†E-mail: postel@ann.jussieu.fr

Contract/grant sponsor: Ministère de la Recherche; contract/grant number: ERT-20052274
Contract/grant sponsor: Marie Curie Fellowship Association for the European Community; contract/grant number: HPMI-CT-1999-00012
Contract/grant sponsor: Institut Français du Pétrole

Copyright © 2006 John Wiley & Sons, Ltd.

talking about the proposed acceleration technique, let us elaborate on what actually lies behind the word ‘realistic’.

Traditionally, such flows are modelled by a 1-D system of partial differential equations (PDEs), obtained by averaging the 3-D conservation equations over cross-sections. This operation is justified, since a typical pipeline length is of the order of several kilometers, whereas its diameter is of the order of a meter, so the flow can be considered as 1-D. We shall consider a class of models known as *drift–flux* model [1], which consists of two-phase-mass balance equations and one total-momentum balance equation. Drift–flux models are used in industrial codes such as TACITE (IFP). Its main advantage over models with two momentum equations [2] is that it provides us with a system in *conservative* form. In return, we have to supply it with an additional algebraic *slip law*.

The degree to which the flow is considered ‘realistic’ depends, of course, on the type of closure laws in use. The slip law, also called the *hydrodynamics* law, reflects indeed the empirical knowledge of physicists about the flow pattern. The *thermodynamics* law, the purpose of which is to give the pressure, looks more standard but could be troublesome too. An evaluation of either the hydrodynamics law or the thermodynamics law may involve iterative procedures and thus be computationally expensive.

But the ‘realism’ of the simulation also depends heavily on two other features. On one hand, we have to take into account the *scenario* that the operator-user wants to apply. In mathematical terms, a scenario is expressed by a set of time-depending values of parameters that are to be imposed as boundary conditions for the PDE system. Since time-varying boundary conditions are the main source of inflow–outflow information for the physical problem, it is therefore capital to cope with them in an effective and accurate way. On the other hand, the geometry of the pipeline has a tremendous influence over the flow. For instance, the liquid can accumulate in the low points of the pipeline, thus blocking the flow. Consequently, the gas pressure just before this bottleneck increases and some point pushes the stopper liquid forward. This phenomenon is known as *slugging*. Observe that slugging is a highly non-stationary flow regime, even if the conditions at the pipeline’s inflow and outflow do not change in time. Due to violent pressure changes in the pipeline, slugging causes faster pipeline fatigue and non-constant oil production on the platform. Needless to say, it is also crucial to be able to deal with changes in inclination.

Another problem we have to face with, in the context of ‘realistic’ oil production flows, is the existence of waves whose speeds are several orders of magnitude apart: fast acoustic (pressure) waves and slow kinematic (transport) waves. Of these waves, only the slow kinematic ones are of concern to us, since these correspond to the displacement of the gas mass-fraction in the mixture. The fast acoustic waves, although uninteresting for the physical application considered, nevertheless impose a severe stability condition (*via* the classical CFL restriction) if treated explicitly. Previous works by Faille and Heintzé [3] and by Baudin *et al.* [4] have shown that it is possible to design an explicit–implicit time integration strategy, so that fast components of the solution are evolved in time with an implicit scheme, thus eliminating the most severe part of the CFL condition, and slow components are evolved in time with an explicit scheme, therefore, ensuring a high degree of accuracy required for the computation of transport waves. This strategy, also called *semi-implicit* scheme, meets the very important requirement of conservativity. Furthermore, extensive test cases show that suitable tuning of the explicit and implicit CFL conditions provide relevant accuracy for the complete solution.

According to engineers, it is essential that, in the neighbourhood of a gas mass-fraction discontinuity or an abrupt change in inclination or the inflow–outflow boundaries, a very small cell size

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

be used in order to capture the motion with enough accuracy. If a uniform mesh were used, this would lead to a huge number of cells in the overall mesh, which is quite unaffordable. Away from a discontinuity, the fluctuations in the solution are due to the supposedly uninteresting acoustic waves. Since these acoustic waves are treated implicitly, the possible zones of stiff transitions are smoothed out from the very first time-steps on. Therefore, it is reasonable to think that their computation can be carried out over a much coarser grid. This observation naturally leads to multiresolution techniques. However, the question that remains to be answered is how can the standard multiresolution techniques, that have been introduced for explicit schemes by Harten [5] and developed by Cohen *et al.* [6, 7] in the context of conservation laws, be properly extended to the semi-implicit scheme.

In [8], we made a first attempt in this direction, by considering the simplified situation where the pipeline is uniformly horizontal and the slip velocity vanishes identically. These hypotheses give rise to models with good mathematical properties. We then succeeded in devising a grid adaptation paradigm, based on local smoothness indicators, that yields good speed-up factors with acceptable accuracy in the results. In the present paper, we gather similar ingredients and some new ideas in order to tackle the most general setting, i.e. problems with complex closure laws and variations in inclination. The closure laws we shall be using are much more 'realistic' than the zero-drift law used in [8].

The paper is outlined as follows. In Section 2 we introduce the drift–flux model and discuss its mathematical properties. In Section 3, we briefly recall the semi-implicit version of the relaxation method [4]. In Section 5, we provide some details about the multiresolution algorithm. The emphasis will be laid on the way we put together the multiresolution algorithm and the semi-implicit scheme. Finally, Section 6 contains numerical results to illustrate the method, including in particular the so-called *severe slugging* phenomenon.

2. THE DRIFT–FLUX MODEL

The pipeline is characterized by its length L and diameter D. In this work, we only consider the *plane* pipelines, i.e., the geometry of the pipeline is determined by the angle $\theta=\theta(x)$, where the x-axis is oriented along the pipeline. Assuming that $D\ll L$, we can consider the flow as 1-D.

The phases are denoted by the subscripts ℓ (liquid) and g (gas). For $k=\ell, g$, the phase densities are denoted by ρ_k, the fraction surfaces (the part of the pipeline cross-section, occupied by the phase k) by R_k, and the velocities by v_k. We suppose the pressure equilibrium between the two phases, $p_\ell=p_g=: p$.

2.1. Conservation laws

The so-called *drift–flux model* consists of three PDEs, expressing the conservation of the masses of both phases and the total momentum of the mixture. It is usually written by physicists as

$$\begin{aligned}\partial_t(\rho_\ell R_\ell)+\partial_x(\rho_\ell R_\ell v_\ell)&=0\\ \partial_t(\rho_g R_g)+\partial_x(\rho_g R_g v_g)&=0\\ \partial_t(\rho_\ell R_\ell v_\ell+\rho_g R_g v_g)+\partial_x(\rho_\ell R_\ell v_\ell^2+\rho_g R_g v_g^2+p)&=S\end{aligned}\qquad(1)$$

where

$$S = -(\rho_\ell R_\ell + \rho_g R_g)[g \sin\theta + Cv|v|] \tag{2}$$

is the source term. Here, $C = 2C_\mathrm{f}/D$, with C_f the friction coefficient, and

$$v = \frac{\rho_\ell R_\ell v_\ell + \rho_g R_g v_g}{\rho_\ell R_\ell + \rho_g R_g} \tag{3}$$

the mass-averaged velocity. Subscripts t and x stand for partial derivatives with respect to t and x.

For mathematical purposes, it is much more convenient to put (1) under a form that is close to gas dynamics. Let us introduce

- the total mixture density $\rho := \rho_\ell R_\ell + \rho_g R_g$
- the gas mass fraction $Y := \rho_g R_g/\rho$
- the slip velocity $\phi := v_\ell - v_g$

so that

$$v_\ell = v + Y\phi \quad \text{and} \quad v_g = v - (1-Y)\phi \tag{4}$$

Then, a little algebra shows that (1) is equivalent to

$$\begin{aligned} \partial_t(\rho) + \partial_x(\rho v) &= 0 \\ \partial_t(\rho Y) + \partial_x(\rho Y v - \sigma) &= 0 \\ \partial_t(\rho v) + \partial_x(\rho v^2 + P) &= S \end{aligned} \tag{5}$$

with

$$\sigma = \rho Y(1-Y)\phi \quad \text{and} \quad P = p + \rho Y(1-Y)\phi^2 \tag{6}$$

Note that the source term (2) now becomes $S = -\rho g \sin\theta - C\rho v|v|$.

The partial differential system (5) has the abstract form

$$\partial_t \mathbb{U} + \partial_x \mathscr{F} = \mathbb{T} \quad \text{for } x \in [0, L], \quad t \geqslant 0 \tag{7}$$

with

$$\begin{aligned} \mathbb{U} &= (\rho, \rho Y, \rho v)^\mathrm{T} \\ \mathscr{F} &= (\rho v, \rho Y v - \Sigma, \rho v^2 + P)^\mathrm{T} \\ \mathbb{T} &= (0, 0, S)^\mathrm{T} \end{aligned} \tag{8}$$

The unknown $\mathbb{U}$ belongs to an open convex subset $\Omega \subset \mathbb{R}^3$, the definition of which will be given later on. The source term $\mathbb{T}$ is easily expressible as a function of $\mathbb{U}$. In order for the flux $\mathscr{F}$ to be a function of the conservative variables $\mathbb{U}$, namely, $\mathscr{F} = \mathscr{F}(\mathbb{U})$, we need to be given two additional algebraic relations to compute p and ϕ. These will be referred to as *closure laws*.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

2.2. Closure laws

2.2.1. Thermodynamics. In the most general setting, we are given two equations of state, one for each phase, that is,

$$\rho_\ell = \rho_\ell(p) \quad \text{and} \quad \rho_g = \rho_g(p) \tag{9}$$

Then, from the obvious equation $R_\ell + R_g = 1$, which is now written as

$$\frac{\rho(1-Y)}{\rho_\ell(p)} + \frac{\rho Y}{\rho_g(p)} = 1 \tag{10}$$

we can infer a value for the pressure $p = p(\rho, Y)$.

In the context of this paper, we use the ideal gas equation of state for the gas

$$\rho_g(p) = \frac{p}{a_g^2} \tag{11}$$

where a_g is the constant gas sound speed. The equation of state for the liquid is

$$\rho_\ell(p) = \rho_\ell^0 + \frac{p - p^0}{a_\ell^2} \tag{12}$$

where p^0 is the atmospheric pressure, ρ_ℓ^0 is the liquid density at $p = p^0$, and a_ℓ is the constant liquid sound speed. Those three parameters are subject to the inequality

$$\rho_\ell^0 > \frac{p^0}{a_\ell^2} \tag{13}$$

Under such conditions, Equation (10) can be transformed into a quadratic equation for p, i.e.

$$\frac{1}{a_\ell^2} p^2 + Dp - E = 0 \tag{14}$$

with

$$D = \left(\rho_\ell^0 - \frac{p^0}{a_\ell^2}\right) - \rho(1-Y) - \rho Y \frac{a_g^2}{a_\ell^2} \quad \text{and} \quad E = a_g^2 \rho Y \left(\rho_\ell^0 - \frac{p^0}{a_\ell^2}\right) \tag{15}$$

Equation (14) has two roots. If $E>0$, only one of the roots is positive. This positive root is associated with the branch

$$p(\mathbb{U}) = \frac{-D + \sqrt{D^2 + 4E/a_\ell^2}}{2/a_\ell^2} = \frac{2E}{D + \sqrt{D^2 + 4E/a_\ell^2}} \tag{16}$$

Note that we also have to require $D>0$, so that in the limit case of an incompressible liquid, we have

$$p(\mathbb{U}) = \frac{2E}{D + |D|} = \frac{E}{D} = a_g^2 \frac{\rho Y}{1 - (\rho/\rho_\ell^0)(1-Y)} > 0 \tag{17}$$

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

In summary, for the pressure to be well-defined, we must have $D>0$ and $E>0$, which restricts the domain to

$$\Omega_{\text{th}} = \left\{ \mathbb{U} \in \mathbb{R}^3 | 0<Y<1, 0<\rho(1-Y) + \rho Y \frac{a_g^2}{a_\ell^2} < \rho_\ell^0 - \frac{p^0}{a_\ell^2} \right\} \tag{18}$$

2.2.2. Hydrodynamics. The slip law $\phi = \phi(\mathbb{U})$ is meant to reflect the flow regime of the two-phase flow. In this work, we shall be using two hydrodynamics laws. The first one is the so-called Zuber–Findlay law [9]. This original Zuber–Findlay is an experimental result for intermittent flows, which states that

$$v_g = c_0(R_g v_g + R_\ell v_\ell) + c_1 \tag{19}$$

where $c_0 = 1+\mu>1$ and $c_1 = v$, with μ and v two arbitrary constants. Inserting (4) into (19), we end up with

$$\phi(\mathbb{U}) = -\frac{(c_0-1)v + c_1}{1-Y+c_0\kappa(\mathbb{U})} = \frac{\mu v + v}{1-Y+(1+\mu)\kappa(\mathbb{U})} \tag{20}$$

where

$$\kappa(\mathbb{U}) = (1-Y)\left(\frac{\rho}{\rho_\ell(p(\mathbb{U}))} - 1\right) \tag{21}$$

In numerical tests, we use $\mu = 0.07$ and $v = 0.2162$. Note that this slip law is highly nonlinear with respect to the unknown $\mathbb{U}$, since it involves the thermodynamics. From now on, and with a little abuse of notation, we will drop the dependence on $\mathbb{U}$.

It can be shown [10] that system (5), equipped with the laws (16) and (20), is hyperbolic over the domain

$$\Omega = \left\{ \mathbb{U} \in \Omega_{\text{th}} | 0<R_g<\frac{1}{c_0}, \phi^2 < \frac{1+\varepsilon}{\varepsilon} W \right\} \tag{22}$$

where

$$\varepsilon = \frac{\rho_g}{\rho_\ell}\frac{c_0 R_g}{1-c_0 R_g} \quad \text{and} \quad W = \frac{p}{\rho_\ell R_g (1-c_0 R_g)} \tag{23}$$

Then, the characteristic speeds are of the form

$$\lambda^- = \hat{v} - a_m, \quad \lambda^0 = v_g, \quad \lambda^+ = \hat{v} + a_m \tag{24}$$

where

$$\hat{v} = \frac{v_\ell + \varepsilon v_g}{1+\varepsilon} \quad \text{and} \quad a_m = \frac{\sqrt{(1+\varepsilon)W - \varepsilon\phi^2}}{1+\varepsilon} \tag{25}$$

Physically, $a_m \gg \hat{v}$ so that $\lambda^- < 0$, $\lambda^+ > 0$. Similarly, in the case of the Euler equations, the *fast* characteristic speeds λ^- and λ^+ are associated with the acoustic waves, whereas the *slow* speed λ^0 is associated with the transport. In our applications we are mainly interested in the accurate resolution of the slow waves, since they correspond to the motion of material interfaces.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

Note that this original Zuber–Findlay law (20) is not valid for $R_g > 1/c_0$, where system (5) fails to be hyperbolic. An easy way to see why this original Zuber–Findlay law behaves badly in the neighbourhood of a pure gas phase is the following. Substituting $(R_g, R_\ell) = (1, 0)$ into Equation (19), we have $v_g = c_0 v_g + c_1$ if v_ℓ were to remain bounded. Unless $(c_0, c_1) = (1, 0)$, this would impose a specific value for v_g, which makes no physical sense. To carry out simulations involving pure gas states, we will use a modified Zuber–Findlay law [11], in which c_0 and c_1 now depend on R_g. More specifically, we choose to work with

$$c_0 = 1 + \mu R_\ell \quad \text{and} \quad c_1 = \nu R_\ell \tag{26}$$

In terms of slip value, we have

$$\phi = -\frac{\mu v + \nu}{1 + \mu\kappa} \tag{27}$$

Due to relative complexity of the law (27), it is impossible to provide the characteristic analysis of system (5), closed with the law (27). In numerical tests, we shall take $\mu = 0.2\sin^2\theta$ and $\nu = 0.35\sqrt{gD}\sin\theta$.

2.3. Boundary conditions and initial data

We define the mass flow-rates $q_\ell = \rho_\ell R_\ell v_\ell$, $q_g = \rho_g R_g v_g$, and $q = q_\ell + q_g$. At the entrance of the pipeline $x = 0$, we impose the mass flows for the liquid $q_\ell = \bar{q}_\ell(t)$, and for the gas $q_g = \bar{q}_g(t)$. An equivalent way of expressing things is to say that, at the entrance, we impose the total flow-rate $q = \bar{q}(t) = \bar{q}_g(t) + \bar{q}_\ell(t)$ and the gas flow-rate $q_g = \bar{q}_g(t)$. At the exit $x = L$, we impose the pressure $p := \bar{p}(t)$. These boundary conditions $\bar{q}$, $\bar{q}_g$, and $\bar{p}$ depend on time.

The initial data $\hat{\mathbb{U}}(x) = \mathbb{U}(x, t = 0)$ are always set as the steady-state solution of (1) with the boundary conditions taken at time $t = 0$, that is,

$$\begin{aligned} \mathrm{d}_x(\hat{\rho}\hat{v}) &= 0 \\ \mathrm{d}_x(\hat{\rho}\hat{Y}\hat{v} - \hat{\sigma}) &= 0 \\ \mathrm{d}_x(\hat{\rho}\hat{v}^2 + \hat{P}) &= \hat{S} \end{aligned} \tag{28}$$

with $\hat{\rho}\hat{v}(x = 0) = \bar{q}(t = 0)$, $(\hat{\rho}\hat{Y}\hat{v} - \hat{\sigma})(x = 0) = \bar{q}_G(t = 0)$, and $\hat{p}(x = L) = \bar{p}(t = 0)$. Note that the achievement of this solution may require an iterative numerical resolution.

3. THE SEMI-IMPLICIT RELAXATION METHOD

In this section we recall briefly the main ideas behind the semi-explicit version of the relaxation method. For details, we refer to Baudin *et al.* [4, 12] and Coquel *et al.* [8].

3.1. Basics of relaxation

3.1.1. Highlighting nonlinearities. In Lagrangian co-ordinates (t, m), with $\mathrm{d}m = \rho\,\mathrm{d}x - \rho v\,\mathrm{d}t$, the drift–flux model (5) boils down to

$$\begin{aligned} \partial_t \tau - \partial_m v &= 0 \\ \partial_t Y - \partial_m \sigma &= 0 \\ \partial_t v + \partial_m P &= \tau S \end{aligned} \tag{29}$$

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

where $\tau=1/\rho$ is the specific volume. This Lagrangian form highlights the nonlinearities σ and P. The idea of relaxation consists in replacing those highly complex functions by unknowns Σ and Π, and to work out evolution equations for these unknowns. More specifically, we consider the new system

$$\begin{aligned}\partial_t\tau-\partial_m v&=0\\ \partial_t Y-\partial_m\Sigma&=0\\ \partial_t v+\partial_m\Pi&=\tau S\\ \partial_t\Pi+a^2\partial_m v&=\lambda[P(\tau,Y,v)-\Pi]\\ \partial_t\Sigma-b^2\partial_m Y&=\lambda[\sigma(\tau,Y,v)-\Sigma]\end{aligned}\tag{30}$$

in which a, b are two positive coefficients, and $\lambda>0$ is the relaxation parameter. Let us note that in the regime of an infinite relaxation parameter $\lambda\to\infty$, the last two PDEs in (30) reduce to $\Pi=P(\tau,Y,v)$ and $\Sigma=\sigma(\tau,Y,v)$. This observation is only formal, since after the works of [13], it is known that, the eigenvalues of the relaxation system (30) must be properly interlaced with those of the equilibrium system (29) to prevent the relaxation procedure from instability in the regime of large λ. In our setting this requirement implies that the coefficients a and b are appropriately set. Precise bounds on these coefficients will be given later on. Let us emphasize that a has the dimension of the Lagrangian acoustic speed while b refers to the kinematic speed. As a consequence, a is much larger than b.

3.1.2. Properties of the relaxation system. Transforming (30) back to Eulerian co-ordinates, we have

$$\begin{aligned}\partial_t(\rho)+\partial_x(\rho v)&=0\\ \partial_t(\rho Y)+\partial_x(\rho Yv-\Sigma)&=0\\ \partial_t(\rho v)+\partial_x(\rho v^2+\Pi)&=S\\ \partial_t(\rho\Pi)+\partial_x(\rho\Pi v+a^2v)&=\lambda\rho[P(\mathbb{U})-\Pi]\\ \partial_t(\rho\Sigma)+\partial_x(\rho\Sigma v-b^2Y)&=\lambda\rho[\sigma(\mathbb{U})-\Sigma]\end{aligned}\tag{31}$$

This is called the *relaxation system* for (5). In abstract form, it reads

$$\partial_t\mathbb{V}+\partial_x\mathscr{G}(\mathbb{V})=\lambda\mathbb{L}(\mathbb{V})+\mathbb{S}(\mathbb{V})\tag{32}$$

where

$$\begin{aligned}\mathbb{V}&=(\rho,\rho Y,\rho v,\rho\Pi,\rho\Sigma)^{\mathrm{T}}=(\mathbb{U},\rho\Pi,\rho\Sigma)^{\mathrm{T}}\\ \mathscr{G}(\mathbb{V})&=(\rho v,\rho Yv-\Sigma,\rho v^2+\Pi,\rho\Pi v+a^2v,\rho\Sigma v-b^2Y)^{\mathrm{T}}\\ \mathbb{S}(\mathbb{V})&=(0,0,S(\mathbb{U}),0,0)^{\mathrm{T}}\\ \mathbb{L}(\mathbb{V})&=(0,0,0,\rho(P(\mathbb{U})-\Pi),\rho(\sigma(\mathbb{U})-\Sigma))^{\mathrm{T}}\end{aligned}\tag{33}$$

Copyright © 2006 John Wiley & Sons, Ltd.

Of course, P and σ are given by Equation (6). The unknown $\mathbb{V}$ belongs to the open set

$$\Omega_{\mathbb{V}} = \{\mathbb{V} = (\mathbb{U}, \rho\Pi, \rho\Sigma) \in \Omega \times \mathbb{R}^2\} \tag{34}$$

The relaxation system (31), at least for the homogeneous part, has several advantages over the original system (5). First of all, its mathematical structure appears to be quite simple. Indeed, the homogeneous system (32) is strictly hyperbolic, and all of its eigenvalues

$$v - \frac{a}{\rho} \ll v - \frac{b}{\rho} \leqslant v \leqslant v + \frac{b}{\rho} \ll v + \frac{a}{\rho} \tag{35}$$

correspond to linearly degenerated fields. Analogously to (24), and since, as already claimed $a \gg b$, we can distinguish between the slow transport waves v and $v \pm b/\rho$ and the fast acoustic waves $v \pm a/\rho$ of (32). Moreover, we can also solve exactly the Riemann problem for (32) (see [12] and below). Therefore, we can use Godunov's method for the numerical solution of (32).

3.1.3. Whitham conditions. When is (31) a 'good' approximation of (5)? The question is equivalent to when is (30) a 'good' approximation of (29). The answer relies on the so-called Chapman–Enskog expansion. We introduce the first-order approximations Π^λ and Σ^λ of the equilibrium values $P(\mathbb{U})$ and $\sigma(\mathbb{U})$

$$\Pi^\lambda = P(\mathbb{U}^\lambda) + \frac{1}{\lambda} P^1_{a,b}(\mathbb{U}^\lambda, \partial_x \mathbb{U}^\lambda) + O\left(\frac{1}{\lambda^2}\right)$$

$$\Sigma^\lambda = \sigma(\mathbb{U}^\lambda) + \frac{1}{\lambda} \sigma^1_{a,b}(\mathbb{U}^\lambda, \partial_x \mathbb{U}^\lambda) + O\left(\frac{1}{\lambda^2}\right)$$

where the first-order correctors $P^1_{a,b}$ and $\sigma^1_{a,b}$ depend on the coefficients a and b. Plugging these expansions in the first three equations of the relaxation system (30) leads to the following system for describing near equilibrium evolution at the first-order in $1/\lambda$:

$$\partial_t \mathbb{U}^\lambda + \partial_x \mathscr{F}(\mathbb{U}^\lambda) = \mathbb{T}(\mathbb{U}^\lambda) + \frac{1}{\lambda} \partial_x (\mathscr{D}_{a,b}(\mathbb{U}^\lambda) \partial_x \mathbb{U}^\lambda)$$

where $\mathscr{D}_{a,b}(\mathbb{U}^\lambda)$ is a matrix which must have non-negative eigenvalues since it must be a diffusion term. The coefficients a and b have to be properly defined to meet this requirement. For this, it is easier to consider P and σ as functions of τ, Y, and v. In [4], it was shown that an appropriate set of conditions to ensure that the relaxation system is a stable approximation to the original system are

$$a^2 > [\partial_v P]^2 - \partial_\tau P \quad \text{and} \quad b > |\partial_Y \sigma| \tag{36}$$

which we will refer to as the Whitham conditions.

3.2. First-order explicit scheme

Consider a computational mesh in the domain $[0, L]$, consisting of cells j with mesh sizes Δx_j, $j = 1, \ldots, N$. In general, the mesh sizes Δx_j can be different. We will describe the construction of the mesh in Section 5. Denote by $\mathbb{V}^n_j$ the average values at the mesh cell j at time t^n, $j = 1, \ldots, N$. The time step Δt is the same for all cells and bounded by the CFL condition on the smallest cell.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

3.2.1. Two-step relaxation procedure. Given some approximation of the equilibrium solution at time t^n, say,

$$\mathbb{V}^n(x) = (\mathbb{U}^n, \rho P(\mathbb{U}^n), \rho\sigma(\mathbb{U}^n))^{\mathrm{T}}(x)$$

this one is evolved at the next time level $t^{n+1} = t^n + \Delta t$ into two steps.

1. *Evolution step*. For times $t \in (0, \Delta t)$, we solve the Cauchy problem for the relaxation system (32) when setting the relaxation parameter λ to zero

$$\begin{aligned}\partial_t(\rho) + \partial_x(\rho v) &= 0\\ \partial_t(\rho Y) + \partial_x(\rho Y v) &= 0\\ \partial_t(\rho v) + \partial_x(\rho v^2 + \Pi) &= S\\ \partial_t(\rho\Pi) + \partial_x(\rho\Pi v + a^2 v) &= 0\\ \partial_t(\rho\Sigma) + \partial_x(\rho\Sigma v - b^2 Y) &= 0\end{aligned} \tag{37}$$

At the discrete level, this step is carried out as

$$\mathbb{V}_j^{n+1} := \mathbb{V}_j^n - \frac{\Delta t}{\Delta x_j}(\mathbb{G}_{j+1/2}^n - \mathbb{G}_{j-1/2}^n) + \Delta t \mathbb{S}_j^n \tag{38}$$

where

$$\mathbb{G}_{j+1/2}^n = \mathbb{G}(\mathbb{V}_j^n, \mathbb{V}_{j+1}^n) = \mathscr{G}(\omega(0^+, \mathbb{V}_j^n, \mathbb{V}_{j+1}^n)) \tag{39}$$

is the numerical flux, based on the exact Riemann problem for the relaxation system and $\mathbb{S}_j^n = \mathbb{S}(\mathbb{U}_j^n)$. The self-similar solution to the Riemann problem with initial data

$$\mathbb{V}(x, t=0) = \begin{cases}\mathbb{V}_j & \text{if } x<0\\ \mathbb{V}_{j+1} & \text{if } x>0\end{cases} \tag{40}$$

is denoted by $\omega(., \mathbb{V}_j, \mathbb{V}_{j+1})$. Its detailed structure reads (see [12])

$$\omega(x/t, \mathbb{V}_j, \mathbb{V}_{j+1}) = \begin{cases}\mathbb{V}_j & \text{if } x/t \leqslant v_j - a/\rho_j\\ \mathbb{V}_1 & \text{if } v_j - a/\rho_j < x/t \leqslant v^* - b/\rho_L^*\\ \mathbb{V}_2 & \text{if } v^* - b/\rho_L^* < x/t \leqslant v^*\\ \mathbb{V}_3 & \text{if } v^* < x/t \leqslant v^* + b/\rho_R^*\\ \mathbb{V}_4 & \text{if } v^* + b/\rho_R^* < x/t \leqslant v_{j+1} + a/\rho_{j+1}\\ \mathbb{V}_{j+1} & \text{if } v_{j+1} + a/\rho_{j+1} < x/t\end{cases} \tag{41}$$

Copyright © 2006 John Wiley & Sons, Ltd.

with

$$\begin{aligned}
\mathbb{V}_1 &= (\rho_L^*, \rho_L^* Y_j, \rho_L^* v^*, \rho_L^* \Pi^*, \rho_L^* \Sigma_j)^{\mathrm{T}}\\
\mathbb{V}_2 &= (\rho_L^*, \rho_L^* Y^*, \rho_L^* v^*, \rho_L^* \Pi^*, \rho_L^* \Sigma^*)^{\mathrm{T}}\\
\mathbb{V}_3 &= (\rho_R^*, \rho_R^* Y^*, \rho_R^* v^*, \rho_R^* \Pi^*, \rho_R^* \Sigma^*)^{\mathrm{T}}\\
\mathbb{V}_4 &= (\rho_R^*, \rho_R^* Y_{j+1}, \rho_R^* v^*, \rho_R^* \Pi^*, \rho_R^*, \Sigma_{j+1})^{\mathrm{T}}
\end{aligned} \tag{42}$$

where

$$\begin{aligned}
v^* &= \frac{1}{2}(v_j + v_{j+1}) - \frac{1}{2a}(\Pi_{j+1} - \Pi_j)\\
\Pi^* &= \frac{1}{2}(\Pi_j + \Pi_{j+1}) - \frac{a}{2}(v_{j+1} - v_j)\\
Y^* &= \frac{1}{2}(Y_j + Y_{j+1}) + \frac{1}{2b}(\Sigma_{j+1} - \Sigma_j)\\
\Sigma^* &= \frac{1}{2}(\Sigma_j + \Sigma_{j+1}) + \frac{b}{2}(Y_{j+1} - Y_j)
\end{aligned} \tag{43}$$

while

$$\frac{1}{\rho_L^*} = \frac{1}{\rho_j} + \frac{\Pi_j - \Pi^*}{a^2} \quad \text{and} \quad \frac{1}{\rho_R^*} = \frac{1}{\rho_{j+1}} + \frac{\Pi_{j+1} - \Pi^*}{a^2} \tag{44}$$

2. *Equilibrium step*. At time $t = \Delta t$, we solve the ODE problem

$$\begin{aligned}
\partial_t(\rho) &= 0\\
\partial_t(\rho Y) &= 0\\
\partial_t(\rho v) &= 0\\
\partial_t(\rho \Pi) &= \lambda \rho [P(\mathbb{U}) - \Pi]\\
\partial_t(\rho \Sigma) &= \lambda \rho [\sigma(\mathbb{U}) - \Sigma]
\end{aligned} \tag{45}$$

with initial data $\mathbb{V}(x, t = \Delta t)$ while letting the relaxation parameter λ go to $+\infty$. This second step in the method clearly amounts to setting

$$\Pi_j^{n+1} = P(\mathbb{U}_j^{n+1}) \quad \text{and} \quad \Sigma_j^{n+1} = \sigma(\mathbb{U}_j^{n+1}) \tag{46}$$

at the discrete level.

3.2.2. Interpretation as a Roe method. As observed in [4], the linear degeneracy property of all the fields in (37) allows us to re-express the Godunov numerical flux function (39) as a Roe-type numerical flux function

$$\mathbb{G}(\mathbb{V}_j, \mathbb{V}_{j+1}) = \tfrac{1}{2}[\mathscr{G}(\mathbb{V}_j) + \mathscr{G}(\mathbb{V}_{j+1})] - \tfrac{1}{2}|\mathscr{A}_{j+1/2}|(\mathbb{V}_{j+1} - \mathbb{V}_j) \tag{47}$$

Copyright © 2006 John Wiley & Sons, Ltd.

where the Roe linearization $\mathscr{A}(\mathbb{V}_j, \mathbb{V}_{j+1}) : \Omega_{\mathbb{V}} \times \Omega_{\mathbb{V}} \to \mathrm{Mat}(\mathbb{R}^5)$ is defined in its diagonalized form $\mathscr{A}(\mathbb{V}_j, \mathbb{V}_{j+1}) = \mathscr{A}_{j+1/2} = \mathscr{R}_{j+1/2}\Delta_{j+1/2}\mathscr{L}_{j+1/2}$ by

$$\mathscr{A}_{j+1/2} = \mathscr{R}_{j+1/2}\Delta_{j+1/2}\mathscr{L}_{j+1/2} \tag{48}$$

with

$$\begin{aligned}
\Delta_{j+1/2} &= \mathrm{diag}(v_j - a/\rho_j,\, v^* - b/\rho_L^*,\, v^*,\, v^* + b/\rho_R^*,\, v_{j+1} + a/\rho_{j+1})\\
\mathscr{R}_{j+1/2} &= \begin{bmatrix} 1 & 0 & 1 & 0 & 1\\ Y_j & 1 & Y^* & 1 & Y_{j+1}\\ v_j - a/\rho_j & 0 & v^* & 0 & v_{j+1} + a/\rho_{j+1}\\ \Pi_j + a^2/\rho_j & 0 & \Pi^* & 0 & \Pi_{j+1} + a^2/\rho_{j+1}\\ \Sigma_j & b & \Sigma^* & -b & \Sigma_{j+1} \end{bmatrix}\\
\mathscr{L}_{j+1/2} &= [\mathscr{R}_{j+1/2}]^{-1}
\end{aligned} \tag{49}$$

For the sake of completeness, we recall that, by definition, a Roe linearization $\mathscr{A}(\mathbb{V}_j, \mathbb{V}_{j+1}) : \Omega_{\mathbb{V}} \times \Omega_{\mathbb{V}} \to \mathrm{Mat}(\mathbb{R}^5)$ should satisfy the three standard conditions

$$\begin{aligned}
&\text{(R1)}\quad \mathscr{A}(\mathbb{V}, \mathbb{V}) = \nabla\mathscr{G}(\mathbb{V})\\
&\text{(R2)}\quad \mathscr{A}(\mathbb{V}_L, \mathbb{V}_R) = (\mathscr{R}\Delta\mathscr{R}^{-1})(\mathbb{V}_L, \mathbb{V}_R) \text{ is } \mathbb{R}\text{-diagonalizable}\\
&\text{(R3)}\quad \mathscr{G}(\mathbb{V}_R) - \mathscr{G}(\mathbb{V}_L) = \mathscr{A}(\mathbb{V}_L, \mathbb{V}_R)(\mathbb{V}_R - \mathbb{V}_L)
\end{aligned} \tag{50}$$

for all states $\mathbb{V}$, $\mathbb{V}_L$ and $\mathbb{V}_R$ in $\Omega_{\mathbb{V}}$. When such a linearization exists, it is common to replace the Riemann problem associated with the exact system

$$\partial_t\mathbb{V} + \partial_x\mathscr{G}(\mathbb{V}) = 0 \tag{51}$$

by that associated with the linearized system

$$\partial_t\mathbb{V} + \mathscr{A}(\mathbb{V}_j, \mathbb{V}_{j+1})\partial_x\mathbb{V} = 0 \tag{52}$$

at each edge $j + \frac{1}{2}$. This yields form (47) for the numerical flux. The ability to re-express the first-order explicit numerical flux (39) as a Roe-type linearization is the most crucial feature that enables us to design an implicit version of the relaxation method.

3.2.3. Practical choice for relaxation parameters. So far, we have considered a and b as global parameters, that is, independent of the space variable. In practice, it is interesting to take them as local parameters $a_{j+1/2}$ and $b_{j+1/2}$ in order to minimize the numerical dissipation.

The Whitham conditions (36) are taken into account by requiring that

$$\begin{aligned}
a_{j+1/2} > a^{\flat}_{j+1/2} &:= \max\left(\sqrt{[\partial_v\tilde{P}]^2 - \partial_\tau\tilde{P}}_j, \sqrt{[\partial_v\tilde{P}]^2 - \partial_\tau\tilde{P}}_{j+1}\right)\\
b_{j+1/2} > b^{\flat}_{j+1/2} &:= \max\left(|\partial_Y\tilde{\sigma}|_j, |\partial_Y\tilde{\sigma}|_{j+1}\right)
\end{aligned} \tag{53}$$

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

Moreover, as shown in [4, 12] it is possible to guarantee that the first-order explicit scheme is positivity-preserving, i.e., it ensures $\rho_j^{n+1}>0$ and $Y_j^{n+1}\in[0,1]$. In order to do so, we must choose the relaxation parameters appropriately, that is,

$$\begin{aligned}
a_{j+1/2} > a^{\sharp}_{j+1/2} &:= \frac{v_i - v_{i+1} + \sqrt{(v_i - v_{i+1})^2 + 8\max(\tau_i,\tau_{i+1})|P_i - P_{i+1}|}}{4\min(\tau_i,\tau_{i+1})} \\
b_{j+1/2} > b^{\sharp}_{j+1/2} &:= \frac{|\rho_i\sigma_i + \rho_{i+1}\sigma_{i+1}| + |\rho_i\sigma_i - \rho_{i+1}\sigma_{i+1}|}{2}
\end{aligned} \tag{54}$$

Therefore, the final tuning for the local relaxation parameters is

$$a_{j+1/2} = \max(a^{\flat}_{j+1/2}, a^{\sharp}_{j+1/2}) \quad \text{and} \quad b_{j+1/2} = \max(b^{\flat}_{j+1/2}, b^{\sharp}_{j+1/2}) \tag{55}$$

3.3. Treatment of boundary conditions

In order to take the boundary conditions into account, we introduce two ghost states: $\mathbb{V}_0^n$ at the inlet of the pipe and $\mathbb{V}_{N+1}^n$ at the outlet. These states are computed as nonlinear functions

$$\mathbb{V}_0^n = \mathbb{V}_0(\mathbb{V}_1^n) \quad \text{and} \quad \mathbb{V}_{N+1}^n = \mathbb{V}_{N+1}(\mathbb{V}_N^n) \tag{56}$$

so that, in the weak sense, they allow us to impose the boundary conditions at the interfaces $x_{1/2}$ and $x_{N+1/2}$.

At the inlet, the ghost state $\mathbb{V}_0^n$ is defined in order to satisfy the five following conditions. First, the relaxation state $\mathbb{V}_0^n$ is at equilibrium which leads to the two identities (46). Second, we specify the total mass flow-rate (which corresponds to the first component ρv) and the gas mass flow-rate (which corresponds to the second component $\rho Y v - \Sigma$) which leads to the two identities

$$\mathbb{G}_1(\mathbb{V}_0^n, \mathbb{V}_1^n) = \bar{q}_\ell(t^n) + \bar{q}_g(t^n), \quad \mathbb{G}_2(\mathbb{V}_0^n, \mathbb{V}_1^n) = \bar{q}_g(t^n) \tag{57}$$

Third, let us introduce ℓ_1 the left eigenvector corresponding to the first eigenvalue of $\nabla\mathscr{G}(\mathbb{V}_1^n)$. We then specify the characteristic projection

$$\ell_1.\mathbb{V}_0^n = \ell_1.\mathbb{V}_1^n \tag{58}$$

The nonlinear system made up of the five equations (46), (57) and (58) is solved by a classical nonlinear solver (for instance, Newton's method).

At the outlet, the ghost state $\mathbb{V}_{N+1}^n$ is defined in order to satisfy the five following conditions. First, the relaxation state $\mathbb{V}_{N+1}^n$ is at equilibrium which leads to the two identities (46). Second, we specify the pressure which leads to the identity

$$p(\mathbb{V}_{N+1}^n) = \bar{p}(t^n) \tag{59}$$

Third, we write the identities

$$Y_{N+1}^n = Y_N^n, \quad v_{N+1}^n = v_N^n \tag{60}$$

The nonlinear system made up of the five equations (46), (59) and (60) is solved exactly.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

3.4. Semi-implicit strategy

3.4.1. Linearly implicit scheme. Thanks to the Roe form (47), we now envisage the implicit version of the evolution step (38)

$$\mathbb{V}_j^{n+1} := \mathbb{V}_j^n - \frac{\Delta t}{\Delta x_j}(\mathbb{G}_{j+1/2}^{n+1} - \mathbb{G}_{j-1/2}^{n+1}) + \Delta t\, \mathbb{S}_j^{n+1} \tag{61}$$

using

$$\mathbb{G}_{j+1/2}^{n+1} = \tfrac{1}{2}[\mathscr{G}(\mathbb{V}_j^{n+1}) + \mathscr{G}(\mathbb{V}_{j+1}^{n+1})] - \tfrac{1}{2}|\mathscr{A}(\mathbb{V}_j^n, \mathbb{V}_{j+1}^n)|(\mathbb{V}_{j+1}^{n+1} - \mathbb{V}_j^{n+1}) \tag{62}$$

Note that we have 'frozen' the Roe matrix $\mathscr{A}_{j+1/2}^n = \mathscr{A}(\mathbb{V}_j^n, \mathbb{V}_{j+1}^n)$ at time n. This implicit numerical flux is still too much involved, so let us Taylor expand it to have a new approximation, namely,

$$\mathbb{G}_{j+1/2}^{n+1} = \mathbb{G}_{j+1/2}^n + \tfrac{1}{2}[\nabla\mathscr{G}(\mathbb{V}_j^n) + |\mathscr{A}_{j+1/2}^n|]\delta\mathbb{V}_j + \tfrac{1}{2}[\nabla\mathscr{G}(\mathbb{V}_{j+1}^n) - |\mathscr{A}_{j+1/2}^n|]\delta\mathbb{V}_{j+1} \tag{63}$$

with $\delta\mathbb{V}_j := \mathbb{V}_j^{n+1} - \mathbb{V}_j^n$. Likewise, we linearize the source term with the Taylor expansion as

$$\mathbb{S}_j^{n+1} = \mathbb{S}_j^n + \nabla\mathbb{S}(\mathbb{V}_j^n)\delta\mathbb{V}_j \tag{64}$$

Concerning the implicit treatment of the boundary conditions at the first order of approximation, we freeze the ghost states $\mathbb{V}_0^n$ and $\mathbb{V}_{N+1}^n$. Exhaustive numerical tests support this strategy (see [4]).

We end up with the linearly implicit scheme

$$\begin{bmatrix} \mathbf{D}_1 & \mathbf{E}_2 & 0 & \cdots & 0 \\ \mathbf{C}_1 & \mathbf{D}_2 & \mathbf{E}_3 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \mathbf{C}_{N-2} & \mathbf{D}_{N-1} & \mathbf{E}_N \\ 0 & \cdots & 0 & \mathbf{C}_{N-1} & \mathbf{D}_N \end{bmatrix} \begin{bmatrix} \delta\mathbb{V}_1 \\ \delta\mathbb{V}_2 \\ \cdots \\ \delta\mathbb{V}_{N-1} \\ \delta\mathbb{V}_N \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \cdots \\ R_{N-1} \\ R_N \end{bmatrix} \tag{65}$$

with

$$\begin{aligned} \mathbf{C}_{j-1} &= -\frac{\Delta t}{2\Delta x_j}[\nabla\mathscr{G}(\mathbb{V}_{j-1}^n) + |\mathscr{A}_{j-1/2}^n|] \\ \mathbf{D}_j &= \mathbf{I} + \frac{\Delta t}{2\Delta x_j}[|\mathscr{A}_{j+1/2}^n| + |\mathscr{A}_{j-1/2}^n|] - \Delta t\nabla\mathbb{S}(\mathbb{V}_j^n) \\ \mathbf{E}_{j+1} &= \frac{\Delta t}{2\Delta x_j}[\nabla\mathscr{G}(\mathbb{V}_{j+1}^n) - |\mathscr{A}_{j+1/2}^n|] \\ R_j &= -\frac{\Delta t}{\Delta x_j}[\mathbb{G}_{j+1/2}^n - \mathbb{G}_{j-1/2}^n] + \Delta t\,\mathbb{S}_j^n \end{aligned} \tag{66}$$

and $R_0 = R_{N+1} = 0$. In (66), $\mathbf{I}$ denotes the unity matrix. System (65) is block-tridiagonal 5×5. As for the R_j's, they are called *explicit residues*.

3.4.2. Linearly semi-implicit scheme. The above scheme (65) is implicit with respect to all waves. As stated in the Introduction, we are mostly interested in the accurate computation of slow transport wave. This leads us to work out a semi-implicit strategy, in which slow waves are explicited while fast waves are implicited.

In order to achieve this objective, we cancel out the diagonal entries corresponding to slow transport waves in $\Delta^n_{j+1/2}$, defined by (49). This amounts to considering

$$\tilde{\Delta}^n_{j+1/2} = \mathrm{diag}(v^n_j - a/\rho^n_j, 0, 0, 0, v^n_{j+1} + a/\rho^n_{j+1}) \tag{67}$$

and to use $\tilde{\mathscr{A}}^n_{j+1/2} = \mathscr{R}^n_{j+1/2}\tilde{\Delta}^n_{j+1/2}\mathscr{L}^n_{j+1/2}$ instead of $\mathscr{A}^n_{j+1/2}$. In a similar fashion, we eliminate the contribution of slow transport waves to the implicit part by modifying the Jacobian matrices: (i) define $\widetilde{\nabla\mathscr{G}}(\mathbb{V})$ by diagonalizing $\nabla\mathscr{G}(\mathbb{V})$ first, then by canceling its slow diagonal entries, and finally by returning to the initial basis; (ii) use $\widetilde{\nabla\mathscr{G}}(\mathbb{V})$ instead of $\nabla\mathscr{G}(\mathbb{V})$ in (66). This leads to the matrices

$$\begin{aligned}
\tilde{\mathbf{C}}_{j-1} &= -\frac{\Delta t}{2\Delta x_j}[\widetilde{\nabla\mathscr{G}}(\mathbb{V}^n_{j-1}) + |\tilde{\mathscr{A}}^n_{j-1/2}|] \\
\tilde{\mathbf{D}}_j &= \mathbf{I} + \frac{\Delta t}{2\Delta x_j}[|\tilde{\mathscr{A}}^n_{j+1/2}| + |\tilde{\mathscr{A}}^n_{j-1/2}|] - \Delta t\nabla\mathbb{S}(\mathbb{V}^n_j) \\
\tilde{\mathbf{E}}_{j+1} &= \frac{\Delta t}{2\Delta x_j}[\widetilde{\nabla\mathscr{G}}(\mathbb{V}^n_{j+1}) - |\tilde{\mathscr{A}}^n_{j+1/2}|]
\end{aligned} \tag{68}$$

to be plugged into the block linear system (65), in place of $\mathbf{C}_{j-1}$, $\mathbf{D}_j$ and $\mathbf{E}_{j+1}$.

Actually, for the semi-implicit scheme to behave well, the experience of Baudin *et al.* [4] shows that it is necessary to take into account the equilibrium condition

$$\Pi^{n+1}_j := P(\mathbb{U}^{n+1}_j) \tag{69}$$

to be satisfied at the equilibrium step (46), already in the implicit evolution step from t^n to $t^n + \Delta t$. During this step, we propose to force a linearized version of the equilibrium condition

$$\delta(\rho\Pi)_j - \left.\frac{\partial\rho P}{\partial\rho}\right|^n_j \delta(\rho)_j - \left.\frac{\partial\rho P}{\partial\rho Y}\right|^n_j \delta(\rho Y)_j - \left.\frac{\partial\rho P}{\partial\rho v}\right|^n_j \delta(\rho v)_j = 0 \tag{70}$$

This amounts to modify the fourth line of $\tilde{\mathbf{D}}_j$ in (68) as

$$\left(\left.\frac{\partial\rho P}{\partial\rho}\right|^n_j, \left.\frac{\partial\rho P}{\partial\rho Y}\right|^n_j, \left.\frac{\partial\rho P}{\partial\rho v}\right|^n_j, -1\right)$$

and to set the fourth lines of $\tilde{\mathbf{C}}_{j-1}$, $\tilde{\mathbf{E}}_{j+1}$ and R_j to zero.

The time step is subject to the two CFL constraints: the explicit and the implicit one. We demand that

- the slow characteristic speeds satisfy

$$\Delta t \max_j \frac{\max(|v^\star - b/\rho^\star_L|_{j+1/2}, |v^\star + b/\rho^\star_R|_{j+1/2})}{\min(\Delta x_j, \Delta x_{j+1})} < \mathrm{CFL}_{\mathrm{exp}} \tag{71}$$

with $\mathrm{CFL}_{\mathrm{exp}} \approx 0.5$;

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

- the fast characteristic speeds satisfy

$$\Delta t \max_j \frac{\max(|v_j - a_{j+1/2}/\rho_j|, |v_{j+1} + b_{j+1/2}/\rho_{j+1}|)}{\min(\Delta x_j, \Delta x_{j+1})} < \mathrm{CFL}_{\mathrm{imp}} \tag{72}$$

  with $\mathrm{CFL}_{\mathrm{imp}} \approx 20$.

See Coquel *et al.* [8] for an in-depth discussion on stability conditions and CFL ratios.

4. EXTENSION TO THE SECOND ORDER

The second-order enhancement in space is performed only on the explicit part of the scheme, namely, the residues R_j. We use the MUSCL piecewise linear data reconstruction of Engquist and Osher [14]. Given piecewise constant values $\mathbb{V}_j^n$ on a *non-uniform* grid where in general $\Delta x_{j_1} \neq \Delta x_{j_2}$ with $j_1 \neq j_2$, we obtain the linearly extrapolated values

$$\mathbb{W}_{j-1/2}^{+} = \mathbb{W}_j^n - \frac{\Delta x_j}{2} \bar{\Delta} \mathbb{W}_j \quad \text{and} \quad \mathbb{W}_{j+1/2}^{-} = \mathbb{W}_j^n + \frac{\Delta x_j}{2} \bar{\Delta} \mathbb{W}_j \tag{73}$$

The experience of Baudin *et al.* [4] shows that the best results are obtained if this step is performed in primitive variables

$$\mathbb{W} = (p, Y, v)^{\mathrm{T}} \tag{74}$$

We limit the slopes $\bar{\Delta}\mathbb{W}_j$ according to

$$\bar{\Delta}\mathbb{W}_j = \mathrm{Limiter}(\Delta\mathbb{W}_{j-1/2}, \Delta\mathbb{W}_{j+1/2}) \tag{75}$$

(to be understood component-wise), where

$$\Delta\mathbb{W}_{j+1/2} = \frac{2}{\Delta x_{j+1} + \Delta x_j} (\mathbb{W}_{j+1}^n - \mathbb{W}_j^n) \tag{76}$$

The limiter function could be any known limiter, such as minmod, Van Leer or superbee.

Once the reconstructed values $\mathbb{W}_{j+1/2}^{-}$, $\mathbb{W}_{j+1/2}^{+}$ have been obtained at edge $j+\frac{1}{2}$, we recompute the corresponding conservative variables $\mathbb{V}_{j+1/2}^{-}$, $\mathbb{V}_{j+1/2}^{+}$ so that these are in equilibrium. We now apply the ideas presented in Section 3, but with $\mathbb{V}_{j+1/2}^{-}$, $\mathbb{V}_{j+1/2}^{+}$ instead of $\mathbb{V}_j^n$, $\mathbb{V}_{j+1}^n$. In other words, the explicit numerical flux is now

$$\mathbb{G}_{j+1/2}^n = \tfrac{1}{2}[\mathscr{G}(\mathbb{V}_{j+1/2}^{-}) + \mathscr{G}(\mathbb{V}_{j+1/2}^{+})] - \tfrac{1}{2}|\mathscr{A}(\mathbb{V}_{j+1/2}^{-}, \mathbb{V}_{j+1/2}^{+})|(\mathbb{V}_{j+1/2}^{+} - \mathbb{V}_{j+1/2}^{-}) \tag{77}$$

The matrix of the implicit system is kept unchanged.

Since realistic field problems are mainly driven by strongly varying *boundary conditions* (see benchmarks in Section 6) it is mandatory to perform the second-order enhancement up to the boundaries. A consistent handling of the boundary conditions would require the knowledge of the ghost state $\mathbb{V}_0(\mathbb{V}_{1/2}^{+,n})$ (respectively, $\mathbb{V}_{N+1}(\mathbb{V}_{N+1/2}^{-,n})$) in order to apply the limited reconstruction (75)–(76). Since $\mathbb{V}_{1/2}^{+,n}$ (respectively, $\mathbb{V}_{N+1/2}^{-,n}$) are the sought-after quantities, we approximate them

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

in the limiter at the previous time step by

$$\mathbb{W}^{+,n}_{1/2} = \mathbb{W}^n_1 - \frac{\Delta x_1}{2}\bar{\Delta}\mathbb{W}_1 \quad \text{and} \quad \mathbb{W}^{-,n}_{N+1/2} = \mathbb{W}^n_N + \frac{\Delta x_N}{2}\bar{\Delta}\mathbb{W}_N$$

with

$$\bar{\Delta}\mathbb{W}_1 = \text{Limiter}(\Delta\mathbb{W}_{1/2}, \Delta\mathbb{W}_{3/2}) \quad \text{and} \quad \bar{\Delta}\mathbb{W}_N = \text{Limiter}(\Delta\mathbb{W}_{N-1/2}, \Delta\mathbb{W}_{N+1/2})$$

where

$$\Delta\mathbb{W}_{1/2} = \frac{1}{\Delta x_1}(\mathbb{W}^n_1 - \mathbb{W}^{n-1}_0) \quad \text{and} \quad \Delta\mathbb{W}_{N+1/2} = \frac{1}{\Delta x_N}(\mathbb{W}^{n-1}_{N+1} - \mathbb{W}^n_N)$$

5. ADAPTIVE MULTIRESOLUTION

Thanks to the implicit treatment of the fast-acoustic-wave components, the time step of the simulation is now mainly ruled by the stability condition derived from the slow transport speed. Despite this major improvement, the overall computing time is still high and leads the engineers to design grids which are too coarse to capture correctly the phenomena—typically slugging—arising near the steep phase transitions. At the same time, because of the field set-up, the computational domain and time length of the simulations are quite large and the solution is most of the time quite smooth in a significant portion of the domain. This makes our problem a good candidate for using an adaptive grid.

In the context of systems of conservation laws, with highly unsteady phenomena, classical adaptive mesh refinement techniques relying on error estimators [15] have been competed with by multiresolution-based adaptive schemes introduced in the early nineties by Harten [5]. In this later framework, multiscale analysis of the transient solution is used to evaluate its time-dependent local smoothness. Following the original papers, the first method consists of defining smoothness indicators selecting the adequate level of discretization among a hierarchy of nested grids. The solution is discretized everywhere on the finest grid and the indicators are used to locally select between the costly numerical scheme and a cheaper linear interpolation. This approach has been successfully developed for complex flows in multidimensional problems [16].

A more recent method introduced by Dahmen *et al.* [17] and formalized in [6] consists in using the multiscale analysis to design a time-varying adaptive grid by selecting the cells at the level among the hierarchy of nested grids according to the local smoothness of the solution. This so-called fully adaptive scheme has been analysed in the explicit scalar case on Cartesian grids in [6] and further implemented in the system cases, for instance, in [7, 18, 19].

In [8], we fully describe how to apply this method in the case of a semi-implicit scheme. Numerical simulations were performed on a simplified version of the system studied here: the drift is neglected, the pipeline being considered horizontal, which leads to a system of four equations only. We will briefly recall the main points of the algorithm and refer the reader to this paper for a full description. The novelties here are the computation of the flux, the treatment of time-varying boundary conditions and of the varying geometry, all of which will be mentioned and described in Section 5.2.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

5.1. Basics of multiresolution analysis

We consider a uniform mesh with step size Δx and starting from this finest discretization, we define a hierarchy of $K+1$ nested grids $(S_k)_{k=0,\dots,K}$ by dyadic coarsening, with cell interfaces $x_j^k = 2^{K-k} j \Delta x$. Initially, the unknown vector-valued function $\mathbb{V}$ is defined on the finest grid, numbered K, where it is represented by the sequence of its mean values $\mathbb{V}^K = (\mathbb{V}_j^K)_j$ on the cells $[x_j^K, x_{j+1}^K]$. The coarsening operator P_k^{k-1} consists of cell averaging from one grid to the coarser one, i.e.

$$\mathbb{V}^{k-1} = P_k^{k-1} \mathbb{V}^k \quad \text{with } \mathbb{V}_j^{k-1} = \tfrac{1}{2}(\mathbb{V}_{2j}^k + \mathbb{V}_{2j+1}^k) \tag{78}$$

The inverse operator consists in recovering the mean values on grid level k, given the mean values on the coarser level $k-1$. This involves an approximation—or prediction—operator P_{k-1}^k. We select here a linear reconstruction based on a quadratic polynomial which obeys adequate locality and consistency rules, therefore, allowing for some analysis (see [6])

$$\hat{\mathbb{V}}^k = P_{k-1}^k \mathbb{V}^{k-1} \quad \text{with} \begin{cases} \hat{\mathbb{V}}_{2j}^k = \mathbb{V}_j^{k-1} - \frac{1}{8}(\mathbb{V}_{j+1}^{k-1} - \mathbb{V}_{j-1}^{k-1}) \\ \hat{\mathbb{V}}_{2j+1}^k = \mathbb{V}_j^{k-1} + \frac{1}{8}(\mathbb{V}_{j+1}^{k-1} - \mathbb{V}_{j-1}^{k-1}) \end{cases} \tag{79}$$

We define the prediction error $\mathbb{E}^k$ at a given level k as the difference between the solution on level k and its reconstruction $\hat{\mathbb{V}}^k$ using the solution at level $k-1$. Thanks to the consistency property $\mathbb{E}_{2j}^k = -\mathbb{E}_{2j+1}^k$, we define the detail vector $\mathbb{D}^{k-1}$ with $\mathbb{D}_j^{k-1} = \mathbb{V}_{2j}^k - \hat{\mathbb{V}}_{2j}^k$ and use it along with $\mathbb{V}^{k-1}$ to recover $\mathbb{V}^k$ entirely.

The two vectors $\mathbb{V}^k$ and $(\mathbb{V}^{k-1}, \mathbb{D}^{k-1})$ are of the same length. Iterating this encoding operation from the finest level down to the coarsest provides the multiscale representation $\mathbb{M}^K = (\mathbb{V}^0, \mathbb{D}^0, \dots, \mathbb{D}^{K-1})$. The double indices of the components of $\mathbb{M}^K$ vary in $\nabla^K = \{(k, j), k = 0, \dots, K-1, j = 0, \dots, 2^k - 1\}$. Using the local structure of the operators P_k^{k-1} and P_{k-1}^k, the multiscale transformation $\mathscr{M} : \mathbb{V}^K \mapsto \mathbb{M}^K$ and its inverse $\mathscr{M}^{-1}$ can be implemented with an optimal $0(N_K)$ complexity, where N_K represents the dimension of the finest grid S^K. The interest of the multiscale representation lies in the fact that the local regularity of the function is reflected by the size of its details. We can use this property to compress the function in the multiscale domain by dropping all details smaller than a given threshold. To clarify this idea, we first define a threshold operator $\mathscr{T}_\Gamma$ acting on the multiscale representation M^K, depending on a subset $\Gamma \subset \nabla^K$ of indices $\gamma = (k, j)$, by

$$\mathscr{T}_\Gamma(\mathbb{D}_\gamma) = \begin{cases} 0 & \text{if } \gamma \in \Gamma \\ \mathbb{D}_\gamma & \text{otherwise} \end{cases} \tag{80}$$

Given level-dependent threshold values $\varepsilon = (\varepsilon_k)_k$, we introduce the subset

$$\Gamma_\varepsilon = \Gamma(\varepsilon_0, \varepsilon_1, \dots, \varepsilon_K) := \{\gamma \text{ s.t. } |\mathbb{D}_\gamma| \geqslant \varepsilon_{|\gamma|}\} \quad \text{where } |\gamma| = |(k, j)| = k \tag{81}$$

This completes the definition of the threshold operator $\mathscr{T}_\varepsilon := \mathscr{T}_{\Gamma_\varepsilon}$, and gives rise to an approximating operator $\mathscr{A}_\varepsilon := \mathscr{M}^{-1} \mathscr{T}_\varepsilon \mathscr{M}$ acting on the physical domain representation. In practice, we take advantage of the fact that the remaining fine-scale details will be concentrated near singularities. This is not such a trivial result because the operator $\mathscr{A}_\varepsilon$ is nonlinear since Γ_ε depends on

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

$\mathbb{V}^K$ throughout the threshold scheme. We refer to [20] for a thorough investigation of nonlinear approximation and the proof of the main result

$$\|\mathbb{V}^K - \mathscr{A}_\varepsilon \mathbb{V}^K\|_{L^1} < C\varepsilon_K \tag{82}$$

valid when $\varepsilon_k = 2^{k-K}\varepsilon_K$. This allows us to 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. The representation $\mathbb{V}_\varepsilon$ of the solution on this adaptive grid is intermediate between the physical representation $\mathbb{V}_\varepsilon^K$ on the finest grid and the encoded multiscale representation $\mathbb{M}_\varepsilon^K = \mathscr{A}_\varepsilon \mathbb{V}^K$

$$\begin{aligned} S_\varepsilon = \{(k, j) \in \{0, \dots, K\} \times \{0, \dots, 2^k N_0\} \setminus \Gamma_\varepsilon \quad \text{s.t. } (k-1, \lfloor j/2 \rfloor) \in \Gamma_\varepsilon \\ \text{or } k = 0\} \end{aligned} \tag{83}$$

where $\lfloor j \rfloor$ denotes the integer part of j. Note, in particular, that the representation of its mean value $\mathbb{V}_j^k$ on an intermediate level k does not mean that the function is locally constant on this cell of width $2^k h$, but simply that its mean values on the finest grid in this area can be recovered—within the ε accuracy—using the mean values on this intermediate level and the reconstruction operators P_{l-1}^l for $l = k+1, \dots, K$. In the sequel, we will call *partial decoding* the algorithm that computes $\mathbb{V}_\varepsilon$ on this adaptive grid from $\mathbb{M}^K$, and *partial encoding* the reverse transformation.

5.2. Prediction strategy for the tree

In the context of the semi-implicit scheme presented in Section 3.4, the singularities of the solution can vary in strength and position with time. They can actually appear or disappear completely, which means that the tree Γ_ε depends on time and that its computation must be performed at each time step. Of course, we wish to compute this time-dependent tree without having to decode the solution back to the finest grid at each time step, since this would destroy all the benefits of the adaptive computation. This is possible, thanks to the hyperbolic nature of the PDEs system, which ensures that the singularities of the solution move at finite speed.

More specifically, if we denote by Γ_ε^n the graded tree obtained by applying $\mathscr{A}_\varepsilon$ to $\mathbb{V}_K^n$, then Γ_ε^n can be inflated into $\tilde{\Gamma}_\varepsilon^{n+1}$ containing Γ_ε^{n+1} as well as Γ_ε^n, ensuring that estimation (82) is valid at both times n and $n+1$ when using $\mathscr{A}_\varepsilon = \mathscr{M}^{-1}\mathscr{T}_{\tilde{\Gamma}_\varepsilon^{n+1}}\mathscr{M}$. Setting $\tilde{\Gamma}_\varepsilon^{n+1}$ to ∇^K does the trick but it is not very interesting in practice. The inflated tree $\tilde{\Gamma}^{n+1}$ should be as small as possible.

The inflation strategy proposed by Harten [5] consists of adding immediate neighbours of cells where the detail is above the level-dependent threshold and the two subdivisions of cells where the detail is more than twice this threshold. In practice this is done with the following algorithm:

Algorithm 5.1

```
PREDICTION OF THE TREE $\tilde{\Gamma}_\varepsilon^{n+1}$
    1. Loop on levels $k$ from fine to coarse
        • if $(k, j) \in \Gamma_\varepsilon^n$ put $(k, j-1)$, $(k, j)$ and $(k, j+1)$ in $\tilde{\Gamma}_\varepsilon^{n+1}$.
        • if $(k, j) \in \Gamma_\varepsilon^n$ and $|\mathbb{D}_{k,j}| \geqslant 2\varepsilon_k$ put $(k+1, 2j)$ and $(k+1, 2j+1)$ in $\tilde{\Gamma}_\varepsilon^{n+1}$.
```

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

2. Loop on levels k from fine to coarse. Ensure gradedness: if $(k, j) \in \tilde{\Gamma}_\varepsilon^{n+1}$, then
   - put $(k, 2\lfloor j/2 \rfloor)$ and $(k, 2\lfloor j/2 \rfloor + 1)$ in $\tilde{\Gamma}_\varepsilon^{n+1}$,
   - put $(k-1, \lfloor j/2 \rfloor - 1)$, $(k-1, \lfloor j/2 \rfloor)$ and $(k-1, \lfloor j/2 \rfloor + 1)$ in $\tilde{\Gamma}_\varepsilon^{n+1}$.

This relies strongly on the CFL condition which is less than one in the case of an explicit scheme. In [8], we have extended this strategy to the time explicit–implicit scheme, thanks to the following heuristic argument sustained by numerical experiments. The implicit treatment of the fast waves ensures a proper behavior—from the point of view of weak solutions—of the fast waves. The chosen CFL, on the other hand, ensures that the initial discontinuous profiles for the fast waves are immediately smoothed out. The only sharp profiles that remain to be captured at the finest discretization levels are associated with the slow wave components. The adaptive computation with the prediction of the tree based on the slow wave CFL condition ensures that these profiles remain as sharp as with the fine uniform grid computation. Below is the actual adaptive algorithm we implement:

Algorithm 5.2
TIME ADAPTIVE MULTIRESOLUTION ALGORITHM.

1. Initialization: encoding of the initial solution using (78) and definition of Γ_ε^0 according to (80–81)
2. Loop over time steps $n = 0, \ldots, N-1$:
   - Prediction of $\tilde{\Gamma}_\varepsilon^{n+1}$ (Algorithm 5.1) and partial decoding of $\mathbb{V}^n$ using (79)
   - Derivation of the adaptive grid $\tilde{S}^{n+1}$ from $\tilde{\Gamma}_\varepsilon^{n+1}$ using (83).
   - Evolution of $\mathbb{V}^n$ to $\mathbb{V}^{n+1}$ on $\tilde{S}^{n+1}$ using scheme described in paragraph 3.4
   - Definition of Γ_ε^{n+1} by partial encoding of $\mathbb{V}^{n+1}$.
3. Decoding of $\mathbb{V}^N$ on the finest grid S^K.

5.2.1. Flux evaluation in the evolution step. An important point in this algorithm is the evaluation of the numerical fluxes between adjacent cells of the adaptive grid, which must be performed in order to update the solution $\mathbb{V}^n$ into $\mathbb{V}^{n+1}$. Theoretically, to ensure the same order of accuracy as the original scheme on the fine uniform scheme, the numerical flux $\mathbb{G}(\mathbb{V}_{j+1/2}^{n,-}, \mathbb{V}_{j+1/2}^{n,+})$ should be fed with reconstructed values on the finest level. These can be obtained by applying the reconstruction operator P_{k-1}^k as many times as necessary, starting from the available values on the adaptive grid. Even though this can be done quite efficiently, at least in the 1-D case [6], this can lead to local violation of the maximum principle. Indeed the thresholding procedure gives an ε-approximation of a positivity preserving scheme using the non-convex linear reconstruction (79). In our case, this can become a severe drawback since the scheme should ensure that the densities remain positive and that the gas mass fractions remain between zero and one for the closure laws to be evaluated.

This leads us to prefer the alternative method consisting of applying the nonlinear limited local reconstruction described in Section 4 directly on the unknowns available on the adaptive grid. A comparison of the two strategies—with or without local MR reconstruction on the finest grid—is presented in [21] for the simple model described in [8], and corroborates our choice.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

5.2.2. Geometry of the pipeline. The pipeline is made of piecewise linear sections. In such geometries, a smooth initial condition—say, constant—can very well develop a singularity at an angle, therefore, requiring small cells in this area. This cannot be correctly captured by the standard MR analysis which designs the initial grid by looking at the smoothness of the initial condition and will find need for coarse cells only. We circumvent this limitation by treating the geometry, in other words the function $\theta(x)$, as a component of the solution as far as the multiresolution is concerned. Angles of the pipeline are seen as stationary singularities in the solution and small cells are imposed at these locations however smooth the solution can be.

5.2.3. Time-varying boundary conditions. Realistic scenarios involve inlet and outlet conditions which can have sharp variations in time. A typical operating set-up consists for instance, in injecting mostly liquid then a sharp linear increase in gas mass fraction until reaching a constant—higher—input level. If nothing is done near the boundary besides the standard MR analysis of the current solution inside the pipeline, since the initial condition is smooth (28), it will require only coarse cells. The sharp increase in injected gas will therefore be badly captured. The MR analysis must be aware of the coming change in the boundary conditions and add small cells accordingly.

In order to play it safe, we do this by testing the gradient in time of the boundary conditions, approximated by a forward finite difference. If the gradient is higher than tolerance, based on the ε tolerance of the MR scheme, the finest-level cell is activated at the corresponding edge of the grid. The adaptive grid is then graded in the usual manner and the solution—however, smooth it may be at the current time—is locally reconstructed using P_{k-1}^{k}.

6. NUMERICAL RESULTS

In order to compare the performance of the numerical scheme of Section 3 we have chosen three test cases which are typical for the offshore oil production. The first two tests simulate the variation of gas inflow in V- and W-shaped pipelines, see Sections 6.1 and 6.2. The last one illustrates the severe slugging phenomenon, see Section 6.3.

A particular emphasis is made on the comparison of the performances of the scheme on uniform and adaptive grids. All computations are made with the second-order scheme of Section 4 using the minmod limiter. The time step is subject to the explicit CFL condition (71) with $\mathrm{CFL}_{\mathrm{exp}} = 0.5$ and the implicit one (72) with $\mathrm{CFL}_{\mathrm{imp}} = 20$.

We introduce the following notions: the relative L^1 error is defined as the arithmetic average of the relative errors in density, gas mass fraction, velocity, and pressure at given time

$$e := \frac{1}{4}\left(\frac{\|\rho_{\mathrm{a}} - \rho_{\mathrm{u}}\|}{\max(\|\rho_{\mathrm{a}}\|, \|\rho_{\mathrm{u}}\|)} + \frac{\|Y_{\mathrm{a}} - Y_{\mathrm{u}}\|}{\max(\|Y_{\mathrm{a}}\|, \|Y_{\mathrm{u}}\|)} + \frac{\|v_{\mathrm{a}} - v_{\mathrm{u}}\|}{\max(\|v_{\mathrm{a}}\|, \|v_{\mathrm{u}}\|)} + \frac{\|p_{\mathrm{a}} - p_{\mathrm{u}}\|}{\max(\|p_{\mathrm{a}}\|, \|p_{\mathrm{u}}\|)}\right) \tag{84}$$

where the subscript a denotes the values of the solution on an adaptive grid, reconstructed up to the finest grid, and the subscript u is related to the values of the uniform solution on the finest grid. The method or the solution is called *adaptive* if it is related to the adaptive grid, and *uniform*

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

if it is related to the uniform grid. The ratio of the execution times of a uniform computation over that of an adaptive one is called the *CPU gain*.

6.1. Varying inflow in a V-shaped pipeline

Consider the following pipeline geometry:

$$\theta(x) = \begin{cases} -30^\circ, & 0 \leqslant x \leqslant L/2 \\ +30^\circ, & L/2 < x \leqslant L \end{cases} \tag{85}$$

with the length $L = 4000$ m and the diameter $D = 0.146$ m.

The sound speed is $a_g = 316.22777$m/s. We consider the liquid as compressible with the equation of state (16), where $\rho_\ell^0 = 1000$ kg/m^3, $p^0 = 10^5$ Pa, and $a_\ell = 500$ m/s. The friction coefficient is $C_\mathrm{f} = 0.005$ and the gravity acceleration is $g = 9.81$ m/s^2.

The boundary conditions at the inflow are variable in time

$$\bar{q}_g(t) = \begin{cases} q_g^0 := 0.1/A, & 0 \leqslant t \leqslant t^0 \\ (q_g^0(t^1 - t) + q_g^1(t - t^0))/(t^1 - t^0), & t^0 < t \leqslant t^1 \\ q_g^1 := 10/A, & t > t^1 \end{cases} \tag{86}$$

where $t^0 := 100$ s and $t^1 := 130$ s determine the stabilization period, and $A = \pi D^2/4$ is the cross-section of the pipeline. The other boundary conditions are constant

$$\bar{q}_\ell := \frac{20}{A}\ \mathrm{kg/m^2\,s}, \quad \bar{p} := 10^5\ \mathrm{Pa} \tag{87}$$

The numerical results on 256 mesh cells, using the modified Zuber–Findlay slip law (27), are presented on Figure 1. The density, velocity and pressure components of the flow, as well as the total mass flow ρv, are represented *versus* the abscissa along the pipeline at time $t = 2000$ s. Two curves are superimposed on each graph and indistinguishable. They correspond to the solution obtained with the uniform grid and to the adaptive one, obtained with $\varepsilon = 0.01$ and three refinement levels. The square symbols indicate the level of refinement used locally, scaled between 0 and 3 on the right vertical axis. We see that the mesh is refined in the neighbourhood of discontinuities and near the low point of the pipeline $x = L/2 = 2000$ m.

Note that this test case is favourable for the MR method. Indeed, the MR method is especially proficient if the solution can be computed on a coarse mesh in most of the domain. The speed-up is directly related to the ratio between the numbers of mesh cells on the uniform and adaptive grids. This ratio is particularly large for this test case: the pipeline is long enough, and the solution has the form of a single propagating front along with a steady perturbation coming from the pipeline low point. Figure 1 shows that only very few cells on the finest level are used in the MR computation. The dependence of the CPU gain on the size of the adaptive grid is shown on Figure 2. Each point on these three curves corresponds to a MR computation made with a different value of the threshold parameter ε. Definition of the thresholding strategy (81) indicates that the larger the ε value is, the more details are discarded and the coarser the resulting adaptive grid is. This corresponds to a small value of the ratio between the number of cells in the adaptive grid and the number of cells on the uniform finest grid. Best CPU gains are obtained by allowing a 5-level coarsening of the discretization. Whatever the number of levels, high CPU gains correspond to

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

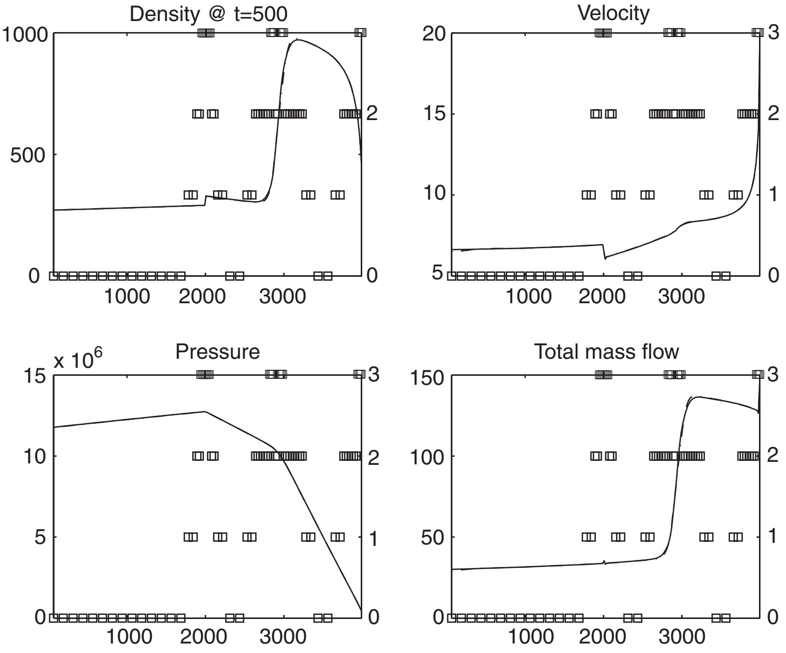

Figure 1. The adaptive and uniform solutions for the flow in the V-shaped pipeline. The squares indicate the refinement levels.

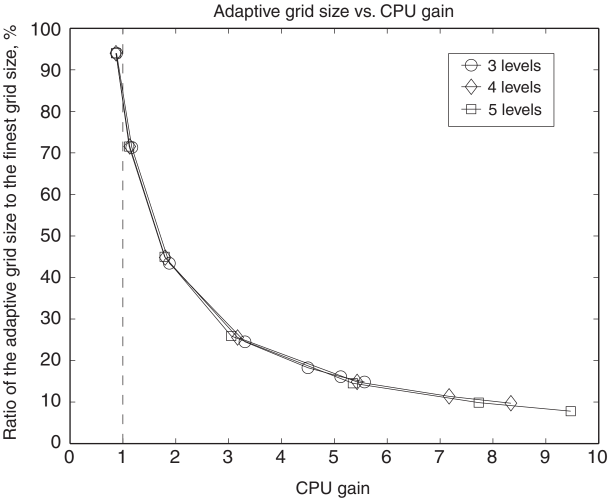

Figure 2. The dependence of the CPU gain on the size of the adaptive grid, 1024 cells on the finest level.

coarse adaptive grids, or large value of the threshold parameter. Discarding relatively not so small details obviously has a price in terms of accuracy, quantified by the error estimate (82) which is illustrated in the next figure.

Copyright © 2006 John Wiley & Sons, Ltd.

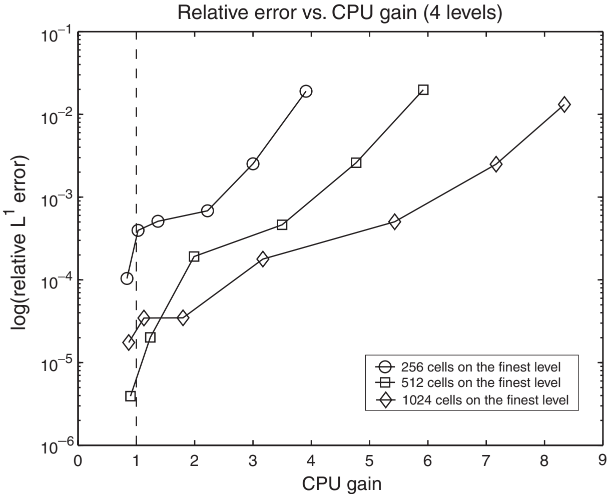

Figure 3. The numerical relative error for several adaptive computations and the corresponding CPU gains.

Figure 3 illustrates the compromise between the precision and the speed-up. The relative L^1 error between the uniform and adaptive solutions is displayed *versus* the CPU gain.

The error is actually computed with respect to the solution obtained using the finest uniform discretization. It behaves in $\mathcal{O}(\varepsilon)$ as predicted by (82). This is standard in performance evaluation of adaptive scheme (see references [6, 16, 17] for instance). The total error with respect to the exact solution is bounded by the sum of this error plus the discretization error. The latter one is in our case behaving in $\mathcal{O}(\Delta x)$ or $\mathcal{O}(\Delta x^2)$ in smooth regions. Again, each point corresponds to a computation made with a different value of ε. For instance, the last point $\diamond$ on the curve displaying the results computed with 1024 cells on the finest level, indicates that the solution can be obtained nine times faster than on the uniform grid. This speed-up is obtained within a 1% error and Figure 2 indicates that the size of the adaptive grid was on average 10% that of the uniform one.

The dependence of the CPU gain on the number of levels for the fixed number of cells on the finest level is shown in Figure 4. It is obvious that increasing the number of coarsening levels does not necessarily lead to speed-up in the computation. Actually, for very small threshold values up to $\varepsilon = 10^{-4}$ the CPU gain diminishes as the number of refinement levels increases. This happens because the precision requirements prevent the coarsest levels to be used. In fact the adaptive grid contains roughly the same number of active cells. On the other hand, the overhead cost due to the MR analysis of the solution increases.

6.2. Varying inflow in a W-shaped pipeline

Consider the following pipeline geometry:

$$\theta(x) = \begin{cases} -30^\circ, & 0 \leqslant x \leqslant L/4 \\ +30^\circ, & L/4 < x \leqslant L/2 \\ -30^\circ, & L/2 \leqslant x \leqslant 3L/4 \\ +30^\circ, & 3L/4 < x \leqslant L \end{cases} \tag{88}$$

with the length $L = 4000$ m and the diameter $D = 0.146$ m.

Copyright © 2006 John Wiley & Sons, Ltd.

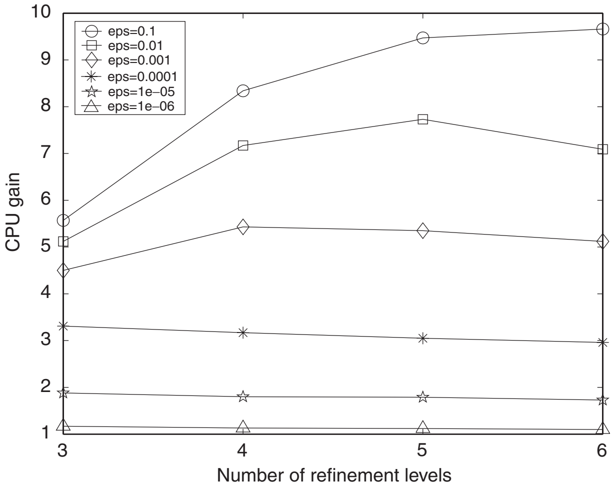

Figure 4. The CPU gain *versus* the number of refinement levels for 1024 cells on the finest level.

The thermodynamic properties and the boundary conditions are the same as for the previous test case.

The numerical results on 256 mesh cells, using the modified Zuber–Findlay slip law (27), are presented on Figure 5. The adaptive solution is obtained with $\varepsilon = 0.01$ and four refinement levels. Here again, we see that the mesh is refined in a neighbourhood of the discontinuities and near the low points of the pipeline.

Table I presents the speed-up of the MR method using 512 cells on the finest grid over the uniform method on the grid of 512 cells. The numbers in parenthesis indicate the ratio of the adaptive grid size over the finest uniform grid size, averaged over the total simulation. Observe that the speed-up is directly related to this ratio which in turn depends on the number of active cells on the finest level. As also shown in Figure 2 for the V-shape pipeline test, it does not depend on the number of discretization levels.

6.3. Severe slugging

The severe slugging phenomenon is a particularly violent flow, caused by the presence of a vertical segment in the pipeline (the *riser*). Slugging occurs in a certain range of pipeline geometries, boundary conditions, and physical material properties. In particular, severe slugging takes place under the following conditions, introduced in [11]: consider a pipeline with the length $L = 74$ m and diameter $D = 0.05$ m. The pipeline inclination is

$$\theta(x) = \begin{cases} 0^\circ, & 0 \leqslant x \leqslant 60 \\ +90^\circ, & 60 < x \leqslant L \end{cases} \tag{89}$$

The sound speed is $a_g = 316.22777\,\text{m/s}$. We consider the liquid as compressible with the equation of state (16), where $\rho_\ell^0 = 1000\,\text{kg/m}^3$, $p^0 = 10^5$ Pa, and $a_\ell = 500\,\text{m/s}$. The friction coefficient is $C_\text{f} = 0.005$ and the gravity acceleration is $g = 9.81\,\text{m/s}^2$. We use the modified Zuber–Findlay slip law (27).

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

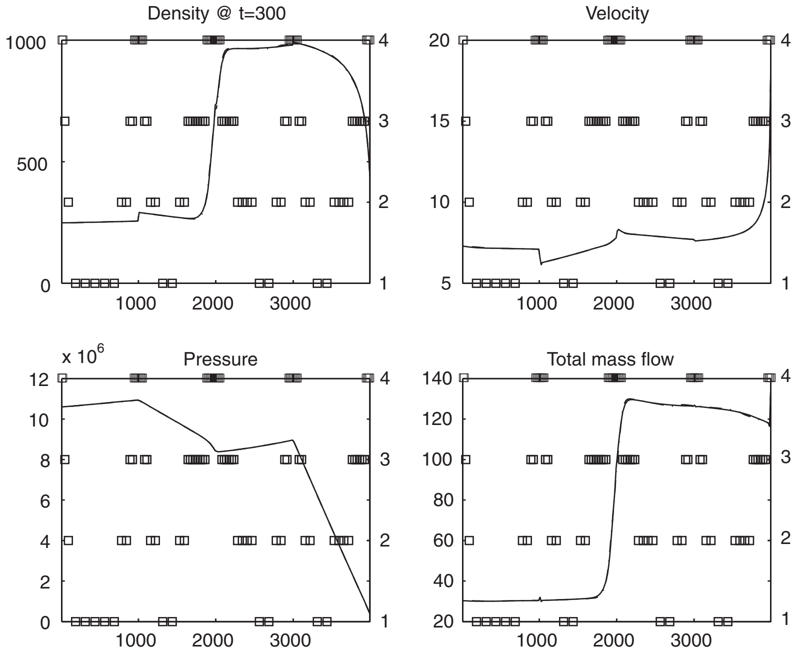

Figure 5. The adaptive and uniform solutions for the flow in the W-shaped pipeline. The squares indicate the refinement levels.

Table I. Speed-up of the MR method *versus* uniform scheme for the W-shaped pipeline test case.

	$\varepsilon = 0.01$	$\varepsilon = 0.001$	$\varepsilon = 0.0001$
3 levels	3.32 (25%)	2.66 (31%)	1.60 (52%)
4 levels	3.48 (23%)	2.48 (32%)	1.57 (52%)
5 levels	3.32 (24%)	2.44 (33%)	1.55 (52%)

The boundary conditions at the inflow and outflow are constant in time

$$\bar{q}_\ell := \frac{0.07854}{A}\,\text{kg/m}^2\,\text{s}, \quad \bar{q}_g := \frac{0.000196}{A}\,\text{kg/m}^2\,\text{s}, \quad \bar{p} := 10^5\,\text{Pa} \tag{90}$$

The time evolution of the solution at fixed position $x = 60\,\text{m}$, using an adaptive grid with three refinement levels, $\varepsilon = 0.0001$, and 128 cells on the finest grid is displayed in Figure 6. The solution at the beginning of the computation exhibits moderate oscillations in the mass flow, caused by the ‘stabilization’ of the initial solution according to the given boundary conditions. Around time $t \approx 180$ the liquid starts to accumulate at the corner $x = 60$, thus, blocking the flow. The pressure in the horizontal part of the pipeline starts to increase and finally pushes the mixture towards the

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

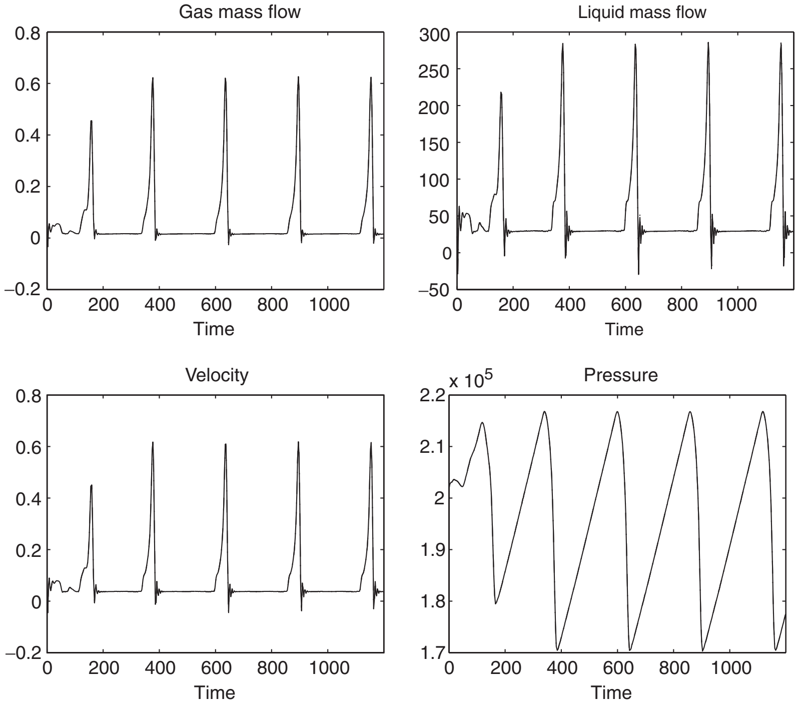

Figure 6. The time evolution of the solution at $x = 60$ for the severe slugging phenomenon. The solutions on the adaptive grid and the uniform one coincide.

Table II. Speed-up of the MR method *versus* the uniform scheme. Severe slugging test case.

	$\varepsilon = 0.01$	$\varepsilon = 0.001$	$\varepsilon = 0.0001$
3 levels	2.24 (36%)	2.08 (39%)	1.94 (43%)
4 levels	2.40 (34%)	2.18 (38%)	1.86 (44%)
5 levels	2.32 (35%)	2.11 (38%)	1.83 (44%)

pipeline exit. The gas and liquid mass flows at the corner grow almost instantaneously by an order of magnitude. Then the flow rarefies and the pressure decreases to a point where the part of the mixture which was at the riser falls down and causes small oscillations in velocity. The liquid blocks the flow again and the process repeats, causing the solution to be periodic. Table II presents the speed-up of the MR method using 128 cells on the finest grid, over the uniform method on the grid of 128 cells. The numbers in parenthesis indicate the ratio of the adaptive grid size over the corresponding uniform grid size, averaged over the full computation. Observe that the speed-up is directly related to the number of the cells on the finest level, cf. Figure 2.

It is extremely difficult to provide numerical evidence of convergence for severe-slugging solutions, because unlike steady-state solutions, we are dealing with an intrinsically unstable phenomenon. Since the initial data are in an unstable region, the state of the system progressively

Copyright © 2006 John Wiley & Sons, Ltd.

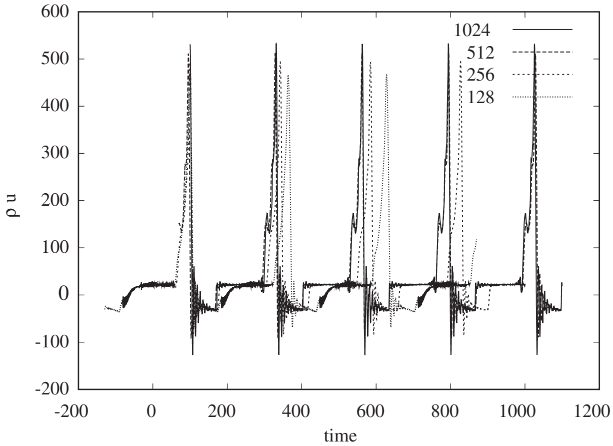

Figure 7. Variation of the total mass flow at the bottom of the riser. Uniform grid computations with varying number of cells.

evolves into a periodic path. By changing the the mesh size, we also change the initial data. As a result, the transitional path towards the periodic solution can be totally different.

The only kind of convergence we are able to observe numerically is the convergence in period and in magnitude of the severe-slugging solutions. In Figure 7, we display the variation of total mass flow at the bottom of the riser *versus* time, for varying uniform discretizations, labeled with the number of cells. The curves have been shifted along the time axis so that the first 'big jumps' all occur at the same time. The convergence in period and in magnitude is then obvious. In the industrial world, the oil-company operators are mostly interested in those two quantities for a severe-slugging problem, instead of the exact locations in time of spikes

7. CONCLUSION

Throughout this paper, we have demonstrated the possibility of combining various existing computational techniques, namely, semi-implicit relaxation schemes for hyperbolic systems and adaptive multiresolution algorithms, in order to speed up numerical simulations of two-phase flows in oil transportation problems. The new dish we cooked up from (not so) old ingredients is meant not only to have good taste, but also to meet strong needs in a realistic industrial context.

At the most fundamental level, the key point that makes things work properly is the conservative form of the PDEs model under consideration, as well as the conservative form of the basic numerical scheme over a uniform grid. This allows us to extend the method quite straightforwardly to other conservative models. At a more practical level, expertise for each ingredient is required, and a large number of additional technical ideas are necessary.

We are pleased with the overall performance of this acceleration method, although it is fair to say that the actual speed-up ratio is dependent on many factors, such as the threshold parameter or the type of data used in the run. One conclusion of the parametric study presented in the last section of this paper is a criterion for achieving interesting CPU gains. The relation between this

Copyright © 2006 John Wiley & Sons, Ltd.

property and the compression property of the adaptive scheme, measured as the percentage ratio between the number of cells in the adaptive and uniform grids, corroborates an observation valid for most adaptive strategies: good performances require that the solution singularities are localized enough to ensure a high compression ratio.

To achieve better gains, we are planning to design adaptive algorithms including local time stepping. This next step of our project appears to be a promising, but far more difficult, challenge.

ACKNOWLEDGEMENTS

This work was supported by the Ministère de la Recherche under grant ERT-20052274: *Simulation avancée du transport des hydrocarbures* and by the Institut Français du Pétrole. We are also grateful to the Marie Curie Fellowship Association for the European Community financial aid HPMI-CT-1999-00012 during Nikolay Andrianov's post-doctoral stay at the IFP.

REFERENCES

1. Viviand H. Étude de modèles d'écoulements diphasiques en approximation monodimensionnelle. *Rapport Principia RSY.77.066.01*, 1997.
2. Stewart HB, Wendroff B. Two-phase flow: models and methods. *Journal of Computational Physics* 1984; **56**:363–409.
3. Faille I, Heintzé E. A rough finite volume scheme for modeling two-phase flow in a pipeline. *Computers and Fluids* 1999; **28**:213–241.
4. Baudin M, Coquel F, Tran Q-H. A semi-implicit relaxation scheme for modeling two-phase flow in a pipeline. *SIAM Journal on Scientific Computing* 2005, in press.
5. Harten A. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Communications on Pure and Applied Mathematics* 1995; **48**(12):1305–1342.
6. Cohen A, Kaber SM, Müller S, Postel M. Fully adaptive multiresolution finite volume schemes for conservation laws. *Mathematics of Computation* 2003; **72**(241):183–225 (electronic).
7. Cohen A, Kaber SM, Postel M. Multiresolution analysis on triangles: application to conservation laws. In *Finite Volumes for Complex Applications*, Benkhaldoun F, Vielsmeier R, Hänel D (eds), vol. II. Hermes: Paris, 1999; 841–848.
8. Coquel F, Postel M, Poussineau N, Tran Q-H. Multiresolution technique and explicit–implicit scheme for multicomponent flows, publication LJLL R05026. *Journal of Numerical Mathematics* 2005 (Special Issue on 'Breaking Complexity: Multiscale Methods for Efficient PDE Solvers', in press).
9. Zuber N, Findlay J. Average volumetric concentration in two-phase flow systems. *Journal of Heat Transfer* 1965; **65**:453–468.
10. Benzoni-Gavage S. Analyse numérique des modèles hydrodynamiques d'écoulements diphasiques instationnaires dans les réseaux de production pétrolière. *Ph.D. Thesis*, Ecole Normale Supérieure de Lyon, 1991.
11. Patault S, Tran Q-H. Modèle et schéma numérique du code TACITE-NPW. *Rapport IFP 42415*, 1996.
12. Baudin M, Berthon C, Coquel F, Masson R, Tran Q-H. A relaxation method for two-phase flow models with hydrodynamic closure law. *Numerische Mathematik* 2005; **99**(3):411–440.
13. Chen GQ, Levermore CD, Liu T-P. Hyperbolic conservation laws with stiff relaxation terms and entropy. *Communications on Pure and Applied Mathematics* 1994; **47**(6):787–830.
14. Engquist B, Osher S. One-sided difference approximations for nonlinear conservation laws. *Mathematics of Computation* 1981; **36**:321–352.
15. Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* 1984; **53**:484–512.
16. Chiavassa G, Donat R. Point value multiscale algorithms for 2D compressible flows. *SIAM Journal on Scientific Computing* 2001; **23**(3):805–823 (electronic).
17. Dahmen W, Gottschlich-Müller B, Müller S. Multiresolution schemes for conservation laws. *Numerische Mathematik* 2001; **88**(3):399–443.
18. Roussel O, Schneider K, Tsigulin A, Bockhorn H. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. *Journal of Computational Physics* 2003; **188**(2):493–523.

Copyright © 2006 John Wiley & Sons, Ltd.

19. Bramkamp F, Lamby Ph, Müller S. An adaptive multiscale finite volume solver for unsteady and steady state flow computations. *Journal of Computational Physics* 2004; **197**(2):460–490.
20. Cohen A. *Numerical Analysis of Wavelet Methods*. North-Holland: Amsterdam, 2003.
21. Andrianov N, Coquel F, Postel M, Tran Q-H. Semi-implicit multiresolution for multiphase flows. *Proceedings ENUMATH 2005*, in press.
22. Andrianov N. Performance of numerical methods on the non-unique solution to the Riemann problem for the shallow water equations. *International Journal for Numerical Methods in Fluids* 2005; **47**:825–831.
23. Andrianov N, Warnecke G. On the solution to the Riemann problem for the compressible duct flow. *SIAM Journal on Applied Mathematics* 2004; **64**(3):878–901.
24. Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics* 1989; **82**:64–84.
25. Dal Maso G, LeFloch PG, Murat F. Definition and weak stability of nonconservative products. *Journal de Mathematiques Pures et Appliquees* 1995; **74**:483–548.
26. Ferré D, Bouvier V, Pauchon C. TACITE Physical Model Description Manual. *Rapport IFP 42385*, 1995.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/fld