# A High-Order Monotonicity-Preserving Scheme for Linear Scalar Advection on 3-D Irregular Meshes

Quang Huy Tran, Bruno Scheurer

# A High-Order Monotonicity-Preserving Scheme for Linear Scalar Advection on 3-D Irregular Meshes

Quang Huy Tran* Bruno Scheurer†

April 2005

**Abstract**

In [*J. Comput. Phys.* **175** (2002), 454–486], we proposed a new scheme for the linear nonconservative transport equation on 2-D irregular meshes. We now extend this scheme to the most general case, i.e., linear conservative transport equation over 3-D distorted meshes, the velocity fields of which may be a function of space and time. Since conservativity is now a major issue, we will thoroughly discuss about the trade-offs between accuracy, monotonicity and conservativity. The greatest advantage of this new scheme, which is compact in space and multilevel in time, lies in the fact that it does conserve mass, preserve monotonicity while ensuring high-order accuracy in smooth regions. Its effectiveness is illustrated by numerical tests.

## Introduction

This paper is a sequel to a previous work [31], published in this Journal. We are concerned with a new numerical method for linear scalar advection over distorted meshes. As was mentioned in [31], our motivation comes from the fact that industrial fluid mechanics ALE codes, such as [2, 36], make heavy use of linear scalar advection as a step within a process for solving the full nonlinear system. In such a context, it has been observed that the grid's orientation has a tremendous effect on the quality of the results, especially when engineers resort to a scheme based on the traditional strategy of dimensional splitting.

We refer the readers to [31] for a discussion about various approaches and a tentative bibliographical review about the quest for the "perfect" scheme for multidimensional advection. In this Introduction, we will briefly recall our search path in order to highlight what remains to be done in the rest of the paper.

At the beginning, we set about implementing and comparing most of the "genuinely multidimensional" methods enumerated and commented below:

*Institut Français du Pétrole, 1 et 4, avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex, France (Q-Huy.Tran@ifp.fr)

†Commissariat à l'Énergie Atomique, DIF, DCSA, B.P. 12, 91680 Bruyères-le-Châtel, France (Bruno.Scheurer@cea.fr)

1. *Corner Transport Upwind* (CTU) by Colella [6], LeVeque [21] and Bell et al. [3] in 2-D, van Leer [35] in 3-D. Equipped with slope-limitation to become high-order, CTU methods do not always ensure a suitable accuracy when the mesh is irregular. In addition, CTU methods are extremely difficult to implement for irregular unstructured meshes.

2. *Discontinuous Galerkin* (DG) by Cockburn et al. [5] in a series of papers. DG methods still involve one-dimensional Riemann problems at each edge (face, in 3-D). Moreover, these turn out to be very expensive.

3. *Residual Distribution* (RD) by Deconinck et al. [8] in a series of VKI courses, Paillère et al. [25]. RD methods have been designed for steady calculations. For unsteady problems, they exhibit first order behaviors. Lately, Abgrall and Mezine [1] put forward a second order scheme for unsteady flows. We do not know about the actual efficiency of this method.

4. *Narrow Schemes* (N) by Sidilkover and Roe [26, 29]. Same remark as for RD.

After unsuccesful attempts to apply any ready-to-use method that would bring a satisfactory answer to typical IFP run tests, we undertook to construct ourselves a method specifically dedicated to the problem at issue. The constraints imposed to us were that the method be (i) monotonicity-preserving; (ii) high-order accurate for smooth data; (iii) suitable to distorted and/or unstructured meshes; (iv) conservative. Because of constraint (iii), we turned our attention toward stencils which are compact in space but possibly multilevel in time. For this purpose, we first went back to the 1-D case and proceeded to several changes in Iserles-Roe's non-dissipative scheme [17, 27] to make it monotonicity-preserving. The price to be paid for is the violation of condition (iv). However, simulations testify to the fact that the lack of conservativity is very small and is somehow "affordable" considering the high accuracy of the results. Postponing the settlement of (iv), we extended this new scheme to the nonconservative 2-D case by applying it along flowlines. Additional tricks were required to solve technical details associated with interpolation over each edge. Shortly after [31] appeared, we have heard of a contribution by Kim [18], who also generalizes Iserles-Roe's original stencil to the 2-D case, yet in another direction. Kim's approach suffers from the disadvantage of not preserving monotonicity.

In this paper, we wish to reach the last milestone of the journey by tackling with the conservativity issue (iv) and by working out a 3-D version of the new scheme for a variable velocity field. Indeed, in real-life problems, the advection step is always governed by a conservative transport equation, and mass-conservation, for instance, turns out to be crucial for physicists and engineers at the numerical level. Since the modified Iserles-Roe scheme of [31] is suitable only for the nonconservative form of the transport equation, we now need to take into account the divergence part of the velocity field by considering it as a source term and by applying a well-balanced fractional-step strategy. Moreover, a mass-correction procedure will also be necessary to ensure conservativity.

The paper is outlined as follows. First, we go back to the 1-D case and study the conservative transport equation. Once a solution has been proposed for the 1-D case, we generalize it directly to the 3-D case, where additional tricks are required. Finally, numerical results are given and commented on.

# 1 The 1-D case

Our objective is to numerically solve the conservative transport equation

$$u_t + [a(x,t)u]_x = 0, \tag{1}$$

where $a(x,t)$ represents the velocity field. In subsections 1.1, 1.2 and 1.3, starting from the easiest case to the hardest one, a scheme is constructed in such a way to meet the first three requirements stated in the Introduction, that is, (i) monotonicity-preserving; (ii) high-order accurate for smooth data; (iii) extendable to distorted and unstructured meshes. In subsection 1.4, an additional procedure is imposed so as to meet the fourth requirement, that is, (iv) mass conservation.

## 1.1 Uniform velocity

By "uniform" we mean $a(x,t) = a$. Among the many non-dissipative schemes studied by Roe in [27] for the advection equation

$$u_t + au_x = 0, \qquad \text{with } \ a > 0, \tag{2}$$

the one that will be extremely helpful to us is

$$u_i^{n+1} = u_{i-1}^{n-2} + 2(1-3\lambda)(u_i^n - u_{i-1}^{n-1}) + \frac{(1-3\lambda)(1-2\lambda)}{1+\lambda}(u_{i-1}^n - u_i^{n-1}), \tag{3}$$

that from now on we refer to as the *Iserles-Roe* original scheme. In (3), standard notations are used for subscript $i$, superscript $n$ and the CFL ratio

$$\lambda = \frac{a\Delta t}{\Delta x}. \tag{4}$$

It can be shown [27] that the Iserles-Roe scheme is fourth-order accurate for sufficiently smooth data. It is exact for $\lambda = \frac{1}{3}$ and stable for $\lambda \leq \frac{1}{2}$. Furthermore, it satisfies a discrete mass conservation rule. Unfortunately, it does not preserve monotonicity, which prevents it from being widely used in industrial applications.

In [31], Tran and Scheurer showed that the Iserles-Roe scheme can be slightly altered so as to become monotonicity-preserving. The scheme we came up with can be put under the following predictor-corrector form.

1. PREDICTOR

- if $\frac{1}{4} \leq \lambda < \frac{1}{2}$, the predicted value is given by the Iserles-Roe scheme (3), i.e.,

$$u_i^\star = u_{i-1}^{n-2} + 2(1-3\lambda)(u_i^n - u_{i-1}^{n-1}) + \frac{(1-3\lambda)(1-2\lambda)}{1+\lambda}(u_{i-1}^n - u_i^{n-1}); \tag{5}$$

- if $0 < \lambda < \frac{1}{4}$, it should rather be computed by what we call the *small-CFL* formula

$$\begin{aligned} u_i^\star = &-\frac{6\lambda^2(1-3\lambda)}{1-\lambda^2}u_{i-1}^{n-1} + \frac{6\lambda^2}{1-\lambda}u_{i-1}^{n-2} + \frac{3(1-3\lambda)}{1-\lambda}u_i^n \\ &- \frac{3(1-3\lambda)(1-2\lambda)}{1-\lambda}u_i^{n-1} + \frac{(1-2\lambda)(1-3\lambda)}{1+\lambda}u_i^{n-2} \end{aligned} \tag{6}$$

2. CORRECTOR

- if $\frac{1}{3} < \lambda < \frac{1}{2}$, the predicted value is changed to

$$u_i^{n+1} = \Pi_{|u_{i-1}^{n-1}, u_{i-1}^{n-2}|}(u_i^\star) \tag{7}$$

- if $0 < \lambda < \frac{1}{3}$, the new value is

$$u_i^{n+1} = \Pi_{|u_{i-1}^{n-2}, u_i^{n}|}(u_i^\star) \tag{8}$$

The notation $\Pi_{|v,w|}(u)$ stands for the projected image of $u$ onto the convex hull spanned by $v$ and $w$. More specifically,

$$\Pi_{|v,w|}(u) = \begin{cases} \min(v,w) & \text{if } u < \min(v,w) \\ u & \text{if } u = \theta v + (1-\theta)w,\ \theta \in [0,1] \\ \max(v,w) & \text{if } u > \max(v,w) \end{cases} \tag{9}$$

Figure 1 recapitulates the 3 different situations that can occur. The sloped lines are the characteristic curves associated to $a$. The predictor step can be interpreted as a Lagrange interpolation process from the values of $u$ at the X-points to obtain a value $u^\star$ at the $\star$-point. The projection interval involved in the corrector step is depicted by a bounding box.

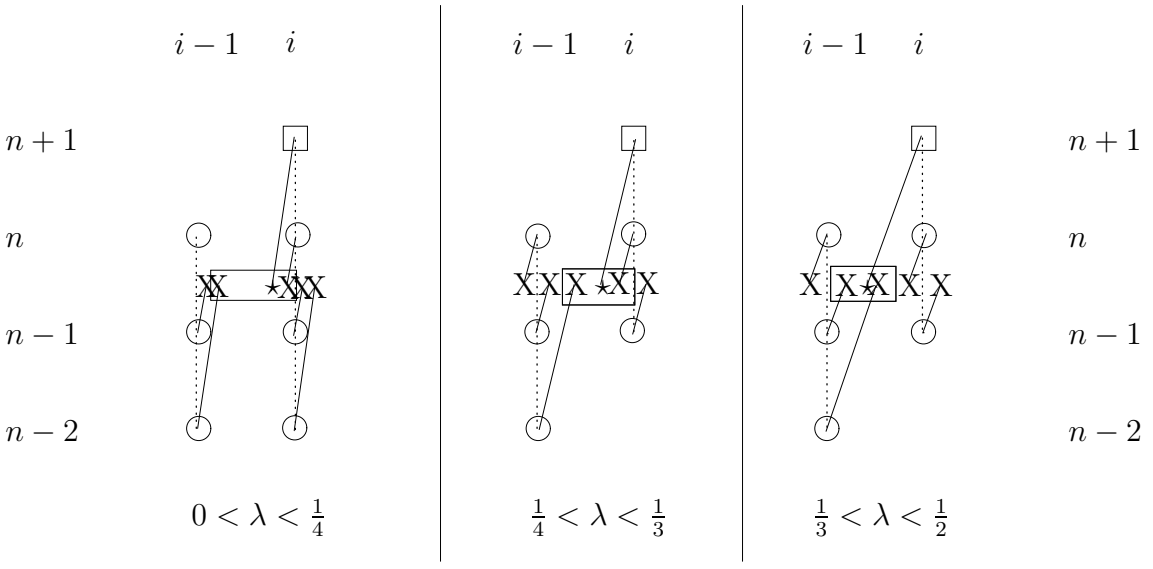


Figure 1: 1-D stencils for various values of the CFL ratio.

The reasons why we chose to do so have been explained in [31]. In a nutshell, the operator $\Pi$ in the corrector step acts as a truncation procedure to force a maximum principle. Here, the key idea is to select the $u$-values at the two *closest* X-neighbors of the $\star$-target in order to define

the projection interval. This "minimizes" the violence of the truncation. As for the small-CFL formula (6) in the predictor step, it comes from the observation that when $\lambda < \frac{1}{4}$, the X-point corresponding to $u_{i+1}^{n-2}$ is closer to the $\star$-target than the grid point $(i,n)$. Therefore, sticking to the proximity principle, we replace the latter by the former in the interpolation process. Numerical simulations evidence the fact that this small-CFL switch is necessary to avoid a staircase effect that would otherwise spoil the computed solution. The overall scheme proves to be very competitive in terms of accuracy, even though it suffers from two minor shortcomings. On one hand, the switch around $\frac{1}{4}$ in the predictor step is not continuous with respect to $\lambda$. On the other hand, there no longer holds any discrete mass conservation property. However, the mass defect can be numerically assessed and turns out to be very small, provided that the characteristic length of the initial data is sufficiently sampled.

### 1.2 Space-variable velocity

By "space-variable" we mean $a(x,t) = a(x)$. In such a case, the conservative form (1) can be rewritten as

$$u_t + a(x)u_x = -a'(x)u, \tag{10}$$

where $a'$ denotes the derivative of $a$ with respect to $x$. Naturally, this leads us to consider, in the first place, the constant-coefficient equation

$$u_t + au_x = bu, \qquad \text{with } \ a > 0 \ \text{ (for simplicity).} \tag{11}$$

Following the ideas advocated by Kim [19], we can prove that one of the "correct" ways to extend the Iserles-Roe scheme (3) to (11) is to write

$$u_i^{n+1}e^{-3\nu} = u_{i-1}^{n-2}e^{3\nu} + 2(1-3\lambda)(u_i^n e^{-\nu} - u_{i-1}^{n-1}e^{\nu}) + \frac{(1-3\lambda)(1-2\lambda)}{1+\lambda}(u_{i-1}^n e^{-\nu} - u_i^{n-1}e^{\nu}), \tag{12}$$

with

$$\lambda = \frac{a\Delta t}{\Delta x} \quad \text{and} \quad \nu = \frac{b\Delta t}{2}. \tag{13}$$

By "correct" we mean a way that ensures Fourier stability for the discretized equation. Actually, (12) can be obtained by applying the original Iserles-Roe formula (3) to the new unknown variable $v(t,x) = u(t,x)e^{bt}$, which satisfies $v_t + av_x = 0$. Likewise, the counterpart of the small-CFL formula (6) reads

$$\begin{aligned} u_i^{n+1}e^{-3\nu} = &-\frac{6\lambda^2(1-3\lambda)}{1-\lambda^2}u_{i-1}^{n-1}e^{\nu} + \frac{6\lambda^2}{1-\lambda}u_{i-1}^{n-2}e^{3\nu} + \frac{3(1-3\lambda)}{1-\lambda}u_i^n e^{-\nu} \\ &- \frac{3(1-3\lambda)(1-2\lambda)}{1-\lambda}u_i^{n-1}e^{\nu} + \frac{(1-2\lambda)(1-3\lambda)}{1+\lambda}u_i^{n-2}e^{3\nu} \end{aligned} \tag{14}$$

If $|\nu| \ll 1$, that is, if the time-step $\Delta t$ is sufficiently small with respect to the source term, then by inserting the approximate expansions

$$e^{\pm\nu} \simeq 1 \pm \nu \quad \text{and} \quad e^{\pm 3\nu} \simeq 1 \pm 3\nu, \tag{15}$$

into formulas (12) and (14), it is possible to express them under a fractional-step form, i.e.,

$$u_i^{n+1} = u_i^\star + s_i^{n+1}, \tag{16}$$

where $u_i^\star$ is the value predicted by (5) or (6), as if there were no source term. As for the source term $s_i^{n+1}$, it is given by

$$s_i^{n+1} = 3\nu(u_{i-1}^{n-2} + u_i^{n+1}) - 2\nu(1-3\lambda)(u_{i-1}^{n-1} + u_i^n) - \nu\frac{(1-3\lambda)(1-2\lambda)}{1+\lambda}(u_{i-1}^n + u_i^{n-1}) \tag{17}$$

for $\lambda \geq \frac{1}{4}$, and

$$s_i^{n+1} = 3\nu(\beta u_{i-1}^{n-2} + \epsilon u_i^{n-2} + u_i^{n+1}) + \nu(\alpha u_{i-1}^{n-1} + \delta u_i^{n-1} + \gamma u_i^n) \tag{18}$$

for $\lambda < \frac{1}{4}$, where $(\alpha, \beta, \gamma, \delta, \epsilon)$ are rational fractions of $\lambda$. We see that the fractional-step interpretation (16) is in fact implicit with respect to the unknown $u_i^{n+1}$, but since the equation to be solved is linear, this unknown can be computed easily via a division. The value obtained for $u_i^{n+1}$ at this stage will be designated by $u_i^{\star\star}$.

We now need to investigate how the projection steps (7) and (8) should be modified to be consistent with the new equation. Keeping in mind that the basic idea amounts to apply the uniform-velocity scheme to the new unknown $v = ue^{-bt}$, it is straightforward to see that what we have to do now is

$$u_i^{n+1} = \Pi_{|u_{i-1}^{n-1}e^{4\nu}, u_{i-1}^{n-2}e^{6\nu}|}(u_i^{\star\star}) \simeq \Pi_{|u_{i-1}^{n-1}(1+4\nu), u_{i-1}^{n-2}(1+6\nu)|}(u_i^{\star\star}) \tag{19}$$

for $\frac{1}{3} < \lambda < \frac{1}{2}$, and

$$u_i^{n+1} = \Pi_{|u_{i-1}^{n-2}e^{6\nu}, u_i^n e^{2\nu}|}(u_i^{\star\star}) \simeq \Pi_{|u_{i-1}^{n-2}(1+6\nu), u_i^n(1+2\nu)|}(u_i^{\star\star}) \tag{20}$$

for $\lambda < \frac{1}{3}$. Let us recapitulate the scheme for (11) as a three-step algorithm:

1. PREDICTOR [identical to the uniform-velocity case]
   - if $\frac{1}{4} \leq \lambda < \frac{1}{2}$, the predicted value $u_i^\star$ is given by the (5);
   - if $0 < \lambda < \frac{1}{4}$, it should rather be computed by (6)
2. SOURCE CORRECTOR [modify $u_i^\star$ with (16) to obtain $u_i^{\star\star}$]
   - if $\frac{1}{4} \leq \lambda < \frac{1}{2}$, use (17) for $s_i^{n+1}$
   - if $\lambda < \frac{1}{4}$, use (18) for $s_i^{n+1}$
3. MONOTONICITY CORRECTOR [truncate $u_i^{\star\star}$ to get $u_i^{n+1}$]
   - if $\frac{1}{3} < \lambda < \frac{1}{2}$, apply (19)
   - if $\lambda < \frac{1}{3}$, apply (20)

Let us introduce the abstract notation

$$u_i^{n+1} = \mathcal{S}_{\lambda,\nu}(\{u_{i-1}\}, \{u_i\}) \tag{21}$$

to designate the 3-step scheme previously described. The symbol $\{u\}$ represents the *ordered list* of values of $u$ at time-levels $n$, $n-1$ and $n-2$. As for the letter $\mathcal{S}$, it stands for *stencil*. This computer-like notation will be very convenient later on.

Now, we return to (10). Assume the velocity $a$ is given at each node $i$ of the grid. Then, we consider

$$\lambda_{i+1/2} = \frac{a_i + a_{i+1}}{2}\frac{\Delta x}{\Delta t} \quad \text{and} \quad \nu_{i+1/2} = \frac{a_i - a_{i+1}}{2}\frac{\Delta x}{\Delta t} \tag{22}$$

The velocity field is thought of as a constant $a_{i-1/2} = \frac{1}{2}(a_i + a_{i+1})$ over each interval $[i-1, i]$. Unlike the uniform case, we cannot assume positivity for $a_{i-1/2}$ because the velocity is allowed to change sign. If the CFL ratios $\lambda_{i-1/2}$ and $\lambda_{i+1/2}$ are both positive, then, accordingly with [31], $u_i$ is updated by

$$u_i^{n+1} = \mathcal{S}_{\lambda_{i-1/2},\nu_{i-1/2}}(\{u_{i-1}\}, \{u_i\}), \tag{23}$$

using the notation (21). If the CFL ratios are both negative, then

$$u_i^{n+1} = \mathcal{S}_{-\lambda_{i+1/2},\nu_{i+1/2}}(\{u_{i+1}\}, \{u_i\}). \tag{24}$$

Should there be a sign disagreement between the two CFL ratios, we consider the local quantities

$$\lambda_i = \frac{a_i \Delta t}{\Delta x} \quad \text{and} \quad \nu_i = \frac{a_{i-1} - a_{i+1}}{4}\frac{\Delta t}{\Delta x} \tag{25}$$

Then, we decide that

$$u_i^{n+1} = \begin{cases} u_i^n(1 + 2\nu_i) & \text{if } \lambda_i = 0 \\ \mathcal{S}_{|\lambda_i|,\nu_i}(\{u_{i-\mathrm{sgn}(\lambda_i)}\}, \{u_i\}) & \text{otherwise.} \end{cases} \tag{26}$$

### 1.3 Time- and space-variable velocity

As a preliminary to the fully variable velocity field, we consider the equation

$$u_t + a(t)u_x = b(t)u. \tag{27}$$

At the discrete level, both the velocity $a$ and the opposite of its derivative $b$ can naturally be thought of as constant $a^{n+1/2}$ and $b^{n+1/2}$ over each interval $[n, n+1]$. The time-step from $n$ to $n+1$ is equal to $\Delta t^{n+1/2}$. The situation starts getting more intricated because the characteristic curves are now broken lines instead of straight lines, as examplified in Fig. 2. Consequently, formulas (5)–(8) are no longer valid, even for positive velocities, unless we have a uniform field.

However, this difficulty is purely technical. In accordance with the spirit of the scheme $\mathcal{S}$, let us introduce the generic symbol

$$\lambda^k = \frac{a^k \Delta t^k}{\Delta x} \quad \text{and} \quad \nu^k = \frac{b^k \Delta t^k}{2} \tag{28}$$

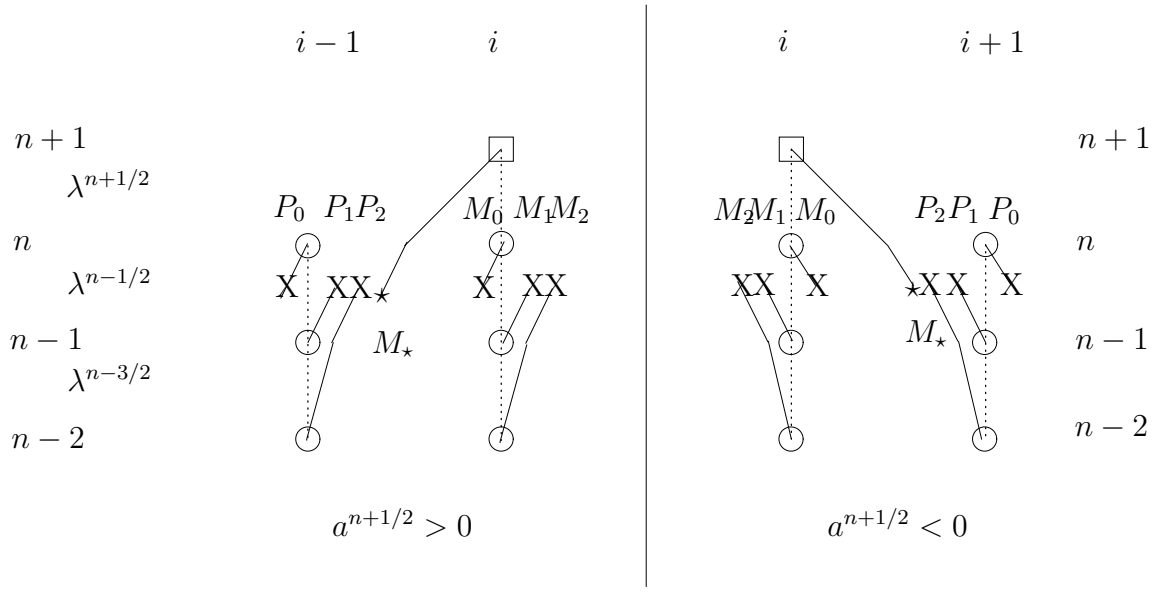


Figure 2: 1-D stencils for time-dependent velocity fields

for any time superscript $k$. Consider the target point $M_\star$, the location of which is now given by

$$M_\star = M_0 - \lambda^{n+1/2}\Delta x, \tag{29}$$

and consider the 6 points X (Fig. 2) where the characteristic curves meet the axis $t = n - \frac{1}{2}$. Let us call them $M_0$, $M_1$, $M_2$ (downwind points) and $P_0$, $P_1$, $P_2$ (upwind points). First, we assume $a^{n+1/2} \neq 0$ and set

$$\ell = \operatorname{sgn}(a^{n+1/2}) = \operatorname{sgn}(\lambda^{n+1/2}). \tag{30}$$

Then, using the same kind of notations as in (29), we define

$$\begin{array}{ll} M_0 = x_i - \frac{1}{2}\lambda^{n-1/2}\Delta x & P_0 = M_0 - \ell\Delta x \\ M_1 = M_0 + \lambda^{n-1/2}\Delta x & P_1 = P_0 + \lambda^{n-1/2}\Delta x \\ M_2 = M_1 + \lambda^{n-3/2}\Delta x & P_2 = P_1 + \lambda^{n-3/2}\Delta x \end{array} \tag{31}$$

Note that we might have $P_2 = P_0$ (if $a$ changes sign) or $P_2 = M_0$ (if $a$ changes too quickly), so that those 6 points are not necessarily distinct from each other. This redundancy phenomenon is a little annoying, but will be solved appropriately in a few moments.

Interpolation and projection lie at the heart of the scheme $\mathcal{S}$. These are the building blocks to be extended. Therefore, what we ought to do is:

1. PREDICTOR & SOURCE CORRECTOR
   To use the values of $u$ (modified by the source term) at *some* of the 6 points $M_{0,1,2}$ and $P_{0,1,2}$ to compute the Lagrange-interpolated value of $\tilde{u}$ at $M_\star$; recall that to each $M$ or

$P$ point, there is an associated value of $u$, namely

$$\begin{array}{ll} M_0 \leftarrow u_i^n e^{-\nu^{n-1/2}} & P_0 \leftarrow u_{i-\ell}^n e^{-\nu^{n-1/2}} \\ M_1 \leftarrow u_i^{n-1} e^{\nu^{n-1/2}} & P_1 \leftarrow u_{i-\ell}^{n-1} e^{\nu^{n-1/2}} \\ M_2 \leftarrow u_i^{n-2} e^{2\nu^{n-3/2}+\nu^{n-1/2}} & P_2 \leftarrow u_{i-\ell}^{n-2} e^{2\nu^{n-3/2}+\nu^{n-1/2}} \end{array} \tag{32}$$

As for the interpolated value $\tilde{u}$ at $M_\star$, it actually represents an approximation of

$$\tilde{u} \simeq u_i^{n+1} e^{2\nu^{n+1/2}+\nu^{n-1/2}} \simeq u_i^{\star\star} e^{2\nu^{n+1/2}+\nu^{n-1/2}} \tag{33}$$

The outcome of this predictor-source corrector step is, as before, denoted by $u_i^{\star\star}$. Of course, with small time-steps, it is possible to replace the exponentials by the less expensive corresponding first-order expansions. Anyhow, there is a degree of implicitness in this calculation, and the value of $u_i^{\star\star}$ would be obtained via a division.

2. MONOTONICITY CORRECTOR
   To project $u_i^{\star\star}$, the predicted and partially corrected value, onto an interval defined by the values of $u$ (modified by the source term) at the 2 closest neighbors X to the target, one on the left and one on the right.

In the predictor step, the number of points actually involved in the interpolation process cannot be fixed in advance. Indeed, because of possible redundancy, it may be necessary to eliminate some of the points. For instance, if $P_2 = P_0$ (in which case $M_2 = M_0$ as well), we remove $P_2$ and $M_2$. If $P_2 = M_0$, we remove $P_2$. Once redundancy has been settled, we select *at most 5* among the remaining points based on the criterion of proximity to the target $M_\star$ and proceed to interpolate. To carry out the Lagrange interpolation effectively, we opt for the use of Newton's divided differences [7]. As for the corrector step, it does not raise any particular problem.

If $a^{n+1/2} = 0$, it trivially comes that $u_i^{n+1} = u_i^n e^{2\nu^{n+1/2}}$. As before, it is convenient to introduce the abstract notation

$$u_i^{n+1} = \mathcal{T}_{\{\lambda\},\{\mu\}}(\{u_{i-\ell}\},\{u_i\}) \qquad \text{or} \qquad u_M^{n+1} = \mathcal{T}_{\{\lambda\},\{\mu\}}(\{u_P\},\{u_M\}) \tag{34}$$

to encapsulate various steps of the scheme just described for a time-dependent velocity. The symbol $\{\lambda\}$ represents the ordered list $\lambda^{n+1/2}$, $\lambda^{n-1/2}$, $\lambda^{n-3/2}$. The similar notation holds for $\{\mu\}$. As for the letter $\mathcal{T}$, it stands for *time-dependent*. This abstract notation includes the trivial situation $a^{n+1/2} = 0$, for which we arbitrarily set $\ell = 0$ and $P_r = M_r$ for $r \in \{0, 1, 2\}$. It can now be checked that if the velocity is uniform over three time-steps, as well as the source factor, that is, if

$$\lambda^{n+1/2} = \lambda^{n-1/2} = \lambda^{n-3/2} = \bar{\lambda} \quad \text{and} \quad \nu^{n+1/2} = \nu^{n-1/2} = \nu^{n-3/2} = \bar{\nu} \tag{35}$$

then the scheme for time-dependent velocity degenerates consistently toward the scheme for uniform velocity. In other words,

$$\mathcal{T}_{\{\lambda\},\{\nu\}}(\{u_{i-\ell}\},\{u_i\}) = \mathcal{S}_{\bar{\lambda},\bar{\nu}}(\{u_{i-\ell}\},\{u_i\}). \tag{36}$$

Finally, it is easy to combine the two cases (space-variable and time-variable velocity field) in order to obtain a scheme for linear conservative equation in which the velocity field is a function of space and time. It suffices to apply $\mathcal{T}$ over each intervall $[i-1, i]$ with suitables values for the $a_{i-1/2}$'s and $b_{i-1/2}$'s by formulas similar to (22).

### 1.4 Conservativity-correction procedure

So far, we have built a new scheme for (1) that meets requirements (i), (ii) and (iii) stated in the Introduction. At first, it may seem odd that, instead of taking advantage of the conservative form of (1), we have chosen to work with (10), which amounts to splitting (1) into a pure color equation [22] $u_t + a(x)u_x = 0$ and a linear source term $-a'(x)u$. The fact is that this unusual manner of seeing things allows us to make use of the (appropriately modified) Iserles-Roe basic stencil, which is a highly accurate one. Talking about "fractional-step" —as we did for equation (16)— is just a way of translating equations into words in order to make the former easier to understand. In reality, for a uniform velocity field, the speed $a$ and the amplification factor $b$ are simultaneously involved in the exact stencil (12) and (14). For a time-dependent velocity, it is not advisable to attempt any separation between the predictor step and the source corrector step, even when exponentials are approximated by first-order expansions.

The question remains, though, to know whether or not the overall scheme is conservative, as mass-conservation, for instance, is a crucial matter for engineers. As was said in subsection 1.1, the answer is negative: as soon as the truncation function $\Pi$ is activated (so as to preserve monotonicity), there appears a defect of conservativity. The latter is very small (less then 1%) when the initial data is sufficiently well-sampled (say, more than 10 points per characteristic length), but can turn out to be impressively large (more than 10%) if the initial is roughly sampled (say, less then 3 points per characteristic length).

Let $M^n$ be the total discrete "mass" (to be understood in an abstract way as the sum or integral of the advected quantity $u$) at time $n$ For instance, over a uniform mesh, we have

$$M^n = \Delta x \sum_i u_i^n. \tag{37}$$

Let us assume that boundary conditions are not involved, so that we would like $M^n$ to remain equal to some constant value $M = M^0 = \int u(x,0)dx$ all the time. Furthermore, in typical run cases, we know *a priori* bounds on $u$, based upon its physical meaning. For instance, it is usually possible to predict

$$0 \leq u(t,x) \leq u_{\max}(t) \tag{38}$$

where $u_{\max}(t)$ is a given function. In such a context, it is possible to work out a conservativity-correction procedure as follows. After time step $n \to n+1$, we compute $M^{n+1}$ and compare it to the theoretical value $M$ by considering the ratio

$$\kappa^{n+1} = \frac{M}{M^{n+1}}. \tag{39}$$

If we multiply every value of the sequence $u_i^{n+1}$ by $\kappa^{n+1}$, i.e.,

$$u_i^{n+1} = \kappa^{n+1} u_i^{n+1}, \tag{40}$$

then the new total "mass" will have the correct value. However, although the homothetic correction (40) does respect the lower-bound $u_i^{n+1} \geq 0$, it may violate the upper-bound $u_i^{n+1} \leq u_{\max}(t^{n+1})$. This is why we set

$$u_i^{n+1} = \min\left\{\kappa^{n+1} u_i^{n+1}, u_{\max}(t^{n+1})\right\}. \tag{41}$$

On one hand, we realize that preserving monotonicity by means of this truncation leads to nonconservation. In other words, we will not be always able to ensure $M^{n+1} = M$. On the other hand, in order to better achieve conservation, we are tempted to carry out an iterative process based on the sequence of formulas (37)–(41). However, this process does not always converge, and does not bring about a significant improvement in terms of accuracy. Regardless of whether or not we iterate over (37)–(41), the conservativity restoration may fail at some time iterations and succeed at a later iteration. Anyhow, numerical simulations show that most of the time, the *a posteriori* correction (41) alone gives rise to much better results, as will be demonstrated in Section 3. This correction procedure is a global step, insofar as it involves simultaneously every nodes of the grid.

# 2 The 3-D case

Since the basic ideas for the 2-D case are already presented in [31] for the nonconservative transport equation, we find it more interesting to go directly the general 3-D conservative equation. Our objective is to numerically solve

$$u_t + \mathrm{div}[\mathbf{a}(\mathbf{p},t)u] = 0 \tag{42}$$

where $\mathbf{a} = (a_x, a_y, a_z)$ represents the velocity vector field, and $\mathbf{p} = (x, y, z)$ are the coordinates. The split form of (42) reads

$$u_t + \mathbf{a}(\mathbf{p},t) \cdot \mathbf{grad} u = -\mathrm{div}[\mathbf{a}(\mathbf{p},t)]u. \tag{43}$$

The key idea here is to make use of the 1-D case by expressing (43) under the "1-D" form, i.e.,

$$u_t + \|\mathbf{w}(s,\sigma,t)\| u_s = -\mathrm{div}[\mathbf{w}(s,\sigma,t)]u, \tag{44}$$

where $s$ is the Euclidean curvilinear coordinate along the *flowline* tagged by $\sigma \in \mathbb{R}^2$. This flowline is rigorously defined as the trajectory $(x_\sigma(s), y_\sigma(s), z_\sigma(s))$ of the differential system

$$\frac{\mathrm{d}x_\sigma}{\mathrm{d}s} = \frac{a_x}{\|\mathbf{a}\|}, \quad \frac{\mathrm{d}y_\sigma}{\mathrm{d}s} = \frac{a_y}{\|\mathbf{a}\|} \quad \text{and} \quad \frac{\mathrm{d}z_\sigma}{\mathrm{d}s} = \frac{a_z}{\|\mathbf{a}\|} \tag{45}$$

along with the Cauchy conditions

$$x_\sigma(s=0) = x_0(\sigma), \quad y_\sigma(s=0) = y_0(\sigma) \qquad \text{and} \qquad z_\sigma(s=0) = z_0(\sigma). \tag{46}$$

There is a technical requirement for the problem (45)–(46) to be well-posed, that is, the surface $\sigma \longmapsto (x_0(\sigma), y_0(\sigma), z_0(\sigma)$ be nowhere tangent to the local vector $\mathbf{a}(x_0, y_0, z_0)$. Under this condition, at fixed $\sigma$, we are brought back to the 1-D case, at least as far as the left-hand side of (44) is concerned. This enables us to apply stencils $\mathcal{S}$ or $\mathcal{T}$ along each flowline.

## 2.1 Uniform velocity

In the uniform case, flowlines are straight lines. Because of uniformity, the divergence source term vanishes. Let $M$ be the vertex at which we wish to update $u$. The backward flowline,

originating from $M$ and directed by $-\mathbf{a}$, meets the grid at a point $P$ called *parent point* of $M$. In a general layout, $P$ belongs to a face, called *parent face*, of a cell containing the backward flowline, called *parent cell*. Introduce the local CFL ratio

$$\lambda(MP) = \frac{\|\mathbf{a}\|\Delta t}{\|MP\|}. \tag{47}$$

If the values of $\{u\}$ were known at $P$, then we would be in a good position to apply the scheme $\mathcal{S}$ between $P$ and $M$. The resulting update formula would be

$$u_M^{n+1} = \mathcal{S}_{\lambda(MP),0}(\{u_P\}, \{u_M\}). \tag{48}$$

We remind that $\{u\}$ denotes the ordered list of $u$ at time-levels $n$, $n-1$, $n-2$. Unfortunately, except for very rare situations when $P$ coincide with another vertex of the mesh, no information about $\{u_P\}$ is available. To get around this difficulty, we replace the missing informations $\{u_P\}$ by some approximated values $\{v_P\}$, so that the scheme becomes

$$u_M^{n+1} = \mathcal{S}_{\lambda(MP),0}(\{v_P\}, \{u_M\}). \tag{49}$$

The hardest part of the job, however, is how to obtain a good value for $v_P$.

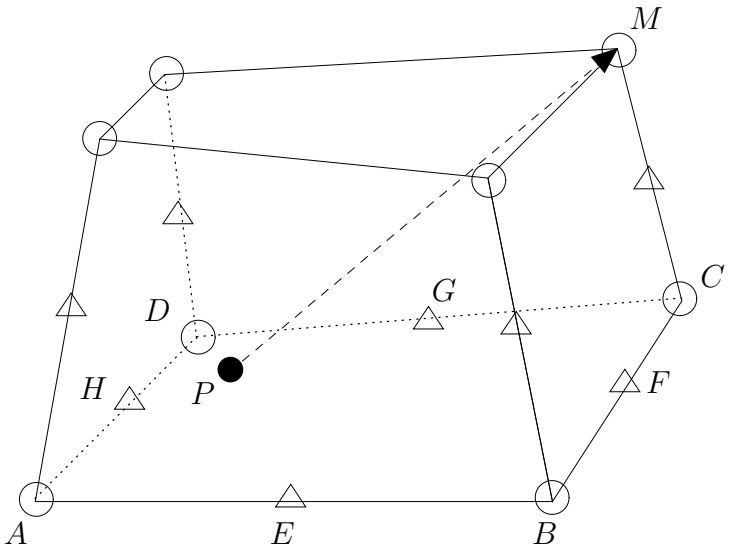


Figure 3: 3-D stencil for uniform velocity fields.

We are going to dwell into the details of estimating $u$ at $P$ for a hexahedrical cell, the faces of which are isoparametric quadrilaterals. This geometrical restriction is assumed in order to fix ideas, and also because in the KIVA code [2, 36], the grid is made up of hexahedrical cells with non-planar quadrilateral faces. The guidelines of the method are actually valid for any kind of cells.

As in the 2-D case [31], we first need to insert extra unknowns, associated with the midpoints of the edges. These auxiliary points have to be updated in the same manner as the vertex points.

The justification for working with the midpoints of edges —and not the centers of faces— is that we want the 3-D scheme to degenerate consistently toward the 2-D scheme whenever $P$ falls upon an edge. In Fig. 3, the vertices of the parent face are $A$, $B$, $C$, $D$, while the midpoints of the edges are $E$, $F$, $G$, $H$. We must have in mind that this face is an isoparametric surface[1] in space. The question is to find a formula

$$v_P = \mathcal{F}_{\eta(P),\zeta(P)}(u_A, \ldots, u_H), \tag{50}$$

in which $(\eta(P), \zeta(P))$ are some local coordinates of $P$ with respect to the face, such that

- $v_P$ is a "good" approximation of $u$ at $P$;
- $v_P$ does not exceed the upper and lower bounds defined by the 8 values $(u_A, \ldots, u_H)$;
- if $P$ lies on an edge, then $\mathcal{F}$ yields the same output as $\mathcal{E}$, the *edge* interpolation operator defined in [31]; for instance, if $P \in [AB]$, then we demand that

$$\mathcal{F}_{\eta(P),\zeta(P)}(u_A, u_B, u_C, u_D; u_E, u_F, u_G, u_H) = \mathcal{E}_{\chi(P;[AB])}(u_A, u_E, u_B) \tag{51}$$

  regardless of the values $u_C$, $u_D$, $u_F$, $u_G$, $u_H$.

The letter $\mathcal{F}$ stands for *face*. The algorithm we are proposing below for $\mathcal{F}$ is composed of two steps. The readers not interested in the details may skip the next two subsections.

### 2.1.1 Predictor

Every point $P$ belonging to the *interior* of the isoparametric surface $(ABCD)$ can be classically expressed as a special affine combination of its vertices. More accurately, for any $P \in \mathrm{Int}(ABCD)$, there exists a unique 4-uplet $(\mu_A, \mu_B, \mu_C, \mu_D) \in [0,1]^4$, which depends on $P$, such that

$$\left\{ \begin{array}{l} 1 = \mu_A \quad + \mu_B \quad + \mu_C \quad + \mu_D \\ 0 = \mu_A\mu_C - \mu_B\mu_D \\ P = \mu_A A \; + \mu_B B \; + \mu_C C + \mu_D D. \end{array} \right. \tag{52}$$

The numbers $(\mu_A, \mu_B, \mu_C, \mu_D)$ are none other than the $Q_1$ finite element basis functions [11]. We also refer to them as *normalized barycentric coordinates* of $P$. These 4 quantities are not independent. As a matter of fact, they can be parameterized by 2 independent variables $(\eta, \zeta) \in [-1,1]^2$, also known as coordinates of the image of $P$ in the reference square [11].

We take it for granted that the normalized barycentric coordinates of $P$ can be computed easily in some way. The formulas we use for predicting $v_P$ is inspired from $Q_2$-serendipity finite elements [4]. This amounts to saying that

$$v_P^\sharp = \sum_{i \in \{A,B,C,D\}} (\nu_i - \tfrac{1}{4}\nu_0) u_i + \sum_{j \in \{E,F,G,H\}} (\nu_j + \tfrac{1}{2}\nu_0) u_j \tag{53}$$

[1] It may be useful to recall that an isoparametric surface leaning on 4 points always contains the straight edges connecting 2 consecutive points.

with

$$\left|\begin{array}{l}\nu_A = \mu_A(\mu_A - \mu_B + \mu_C - \mu_D)\\ \nu_B = \mu_B(\mu_B - \mu_C + \mu_D - \mu_A)\\ \nu_C = \mu_C(\mu_C - \mu_D + \mu_A - \mu_B)\\ \nu_D = \mu_D(\mu_D - \mu_A + \mu_B - \mu_C)\\ \nu_E = 4\mu_A(\mu_B - \mu_C)\\ \nu_F = 4\mu_B(\mu_C - \mu_D)\\ \nu_G = 4\mu_C(\mu_D - \mu_A)\\ \nu_H = 4\mu_D(\mu_A - \mu_B)\\ \nu_0 = 8(\mu_A\mu_C + \mu_B\mu_D).\end{array}\right. \tag{54}$$

If $P$ happens to lie on an edge, e.g., $P \in [AB]$, then $\mu_C = \mu_D = 0$, and it can checked that the value provided by (53) is exactly equal to that given by the edge interpolation predictor in the 2-D case [31].

### 2.1.2 Corrector

Let us define the 4 *feet* of $P$ with respect to the non-planar quadrilateral $(ABCD)$ by

$$\begin{array}{r}I = (\mu_A + \mu_D)A + (\mu_B + \mu_C)B\\ J = (\mu_B + \mu_A)B + (\mu_C + \mu_D)C\\ K = (\mu_C + \mu_B)C + (\mu_D + \mu_A)D\\ L = (\mu_D + \mu_C)D + (\mu_A + \mu_B)A.\end{array} \tag{55}$$

It can be checked that $P = [IK] \cap [JL]$ and that, moreover, the segments $[IK]$ and $[JL]$ entirely belong to the isoparametric surface. At the feet of $P$, we consider the values

$$\begin{array}{r}\overline{v}_I = \mathcal{E}_{\chi(I;[AB])}(u_A, u_E, u_B)\\ \overline{v}_J = \mathcal{E}_{\chi(J;[BC])}(u_B, u_F, u_C)\\ \overline{v}_K = \mathcal{E}_{\chi(K;[CD])}(u_C, u_G, u_D)\\ \overline{v}_L = \mathcal{E}_{\chi(L;[DA])}(u_D, u_H, u_A)\end{array} \tag{56}$$

obtained via the edge interpolation process $\mathcal{E}$ (see [31] for details).

The next move is to select 2 values out of those 4 candidates, based on a criterion of proximity to $P$. To this purpose, we define

$$\overline{v}(P; IK) = \begin{cases}\overline{v}_I & \text{if } \chi(P;[IK]) < 0\\ \overline{v}_K & \text{if } \chi(P;[IK]) > 0\\ \frac{1}{2}(\overline{v}_I + \overline{v}_K) & \text{if } \chi(P;[IK]) = 0\end{cases} \tag{57}$$

and

$$\overline{v}(P; JL) = \begin{cases}\overline{v}_J & \text{if } \chi(P;[JL]) < 0\\ \overline{v}_L & \text{if } \chi(P;[JL]) > 0\\ \frac{1}{2}(\overline{v}_J + \overline{v}_L) & \text{if } \chi(P;[JL]) = 0\end{cases} \tag{58}$$

In other words, along each segment $[IK]$ and $[JL]$, we retain the value $\overline{v}$ of the foot that is closer to $P$. If $P$ is the midpoint of the segment, we simply take the half-sum the $\overline{v}$-values. In Fig. 4, for instance, $\overline{v}(P; IK) = \overline{v}_I$ and $\overline{v}(P; JL) = \overline{v}_J$.

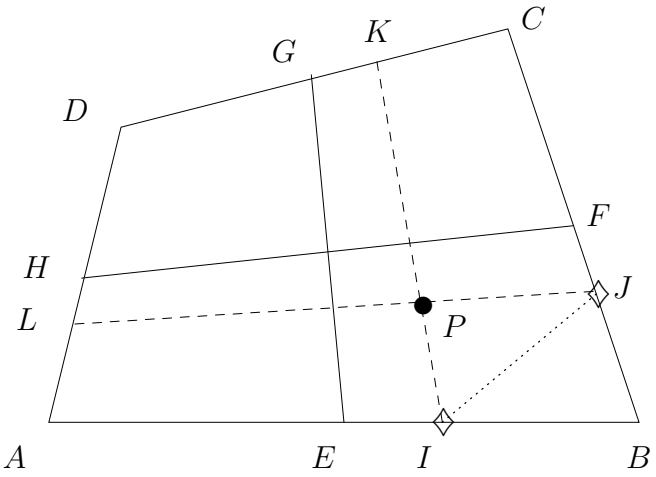


Figure 4: Monotonicity-preserving device on a quadrilateral face

Finally, the corrected value is set to

$$v_P = \Pi_{|\overline{v}(P;IK),\overline{v}(P;JL)|}(v_P^\sharp). \tag{59}$$

It is not difficult, although a little tedious, to verify that the process $\mathcal{F}$ defined by (53) and (59) does comply with all of the requirements enumerated at the beginning of the section.

## 2.2 Space-variable velocity

The velocity vector in the equation

$$u_t + \mathbf{a}(\mathbf{p}) \cdot \mathbf{grad} u = -\mathrm{div}[\mathbf{a}(\mathbf{p})]u \tag{60}$$

can be thought of as a constant vector $\mathbf{a}_K$ over each cell $K$. Likewise, the opposite of its divergence $b(\mathbf{p}) = -\mathrm{div}[\mathbf{a}(\mathbf{p})]$ can also be thought of as a constant $b_K$ over each cell $K$. Various formulas could be proposed for $\mathbf{a}_K$ and $b_K$ from their values on the vertices of $K$.

Let $M$ be the vertex to be updated. If there is no information conflict between the vectors $\mathbf{a}_{K'}$ for the cells $K'$ containing $M$, that is, if these vectors all give rise to the same parent cell $K$ for $M$, then $\mathbf{a}_K$ is used to determine the parent edge and point. In case there is a conflict about the parent cell, we need to compute a local velocity $\mathbf{a}_M$ (see [31] for more details). This $\mathbf{a}_M$ is then used to determine the parent cell, edge and point with $\mathbf{a}_M$. In such a conflictual case, another discrete value for $b_M$ can also be worked out.

Once the point $P$ have been found on the parent face $(ABCD)$ with middle-points $(EFGH)$, belonging to the parent cell $K$, we define

$$\lambda(MP) = \frac{\|\mathbf{a}_K\|\Delta t}{\|MP\|} \quad \text{and} \quad \nu_K = \frac{b_K \Delta t}{2} \tag{61}$$

and apply the scheme

$$u_M^{n+1} = \mathcal{S}_{\lambda(MP),\nu_K}(\{v_P\},\{u_M\}), \tag{62}$$

where the $v_P$'s are determined by the face interpolation operator symbolized by (50).

### 2.3 Time- and space-variable velocity

As a preliminary to the fully variable velocity field, we consider the equation

$$u_t + \mathbf{a}(t) \cdot \mathbf{grad} u = b(t) u. \tag{63}$$

When the velocity $\mathbf{a}$ and the opposite of its divergence $b$ depend only on the time variable $t$, it can naturally be thought of as a constant vector $\mathbf{a}^{n+1/2}$ over each interval $[n, n+1]$. If the direction of $\mathbf{a}$ remains constant in time, the parent point $P$ does not move as time iterations go on. Then, it makes sense to apply the 1-D stencil $\mathcal{T}$ between $P$ and $M$. But in the most general case, the 3 vectors $\mathbf{a}^{n+1/2}$, $\mathbf{a}^{n-1/2}$ and $\mathbf{a}^{n-3/2}$ do not share the same direction. As a consequence, the parent point $P$ is not fixed in time. We should then distinguish $P^{n+1/2}$ (computed with $\mathbf{a}^{n+1/2}$), $P^{n-1/2}$ (computed with $\mathbf{a}^{n-1/2}$), and $P^{n-3/2}$ (computed with $\mathbf{a}^{n-3/2}$).

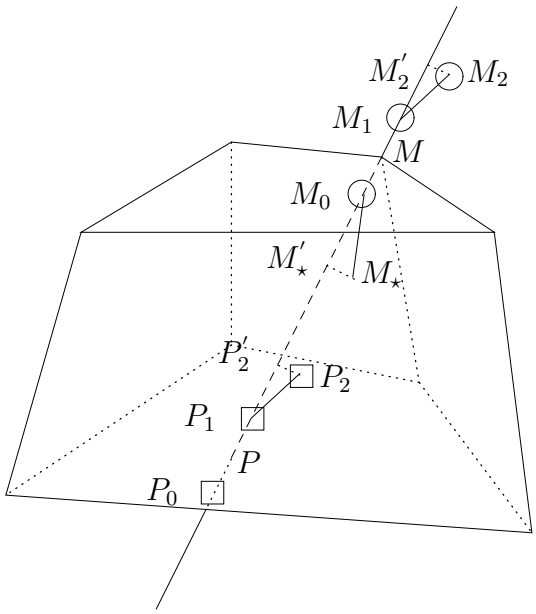


Figure 5: 3-D stencil for time-dependent velocity fields, at time $t = n - 1/2$.

Before solving this turning direction problem, let us see it from a different viewpoint. As usual, let $M$ be the vertex where we wish to update $u$. As has been noticed several times, there is a natural symmetry of the scheme with respect to the time level $t = n - 1/2$. This is why we will compute the backward flowline originating from $M$ with $\mathbf{w}^{n-1/2}$. This backward flowline cuts the rest of the grid at the parent point $P$ ($= P^{n-1/2}$). Analogously to (31), in the 3-D space, consider the following points

$$\begin{array}{ll} M_0 = M - \frac{1}{2}\Delta t^{n-1/2}\mathbf{a}^{n-1/2} & P_0 = P - \frac{1}{2}\Delta t^{n-1/2}\mathbf{a}^{n-1/2} \\ M_1 = M_0 + \Delta t^{n-1/2}\mathbf{a}^{n-1/2} & P_1 = P_0 + \Delta t^{n-1/2}\mathbf{a}^{n-1/2} \\ M_2 = M_1 + \Delta t^{n-3/2}\mathbf{a}^{n-3/2} & P_2 = P_1 + \Delta t^{n-3/2}\mathbf{a}^{n-3/2} \end{array} \tag{64}$$

and the target point

$$M_\star = M_0 - \Delta t^{n+1/2}\mathbf{a}^{n+1/2}. \tag{65}$$

Geometrically speaking, since these points are not aligned, it does not make sense to ask for an interpolated value at $M_\star$, given the values at the six other points.

From this observation, it can be suggested a device to get around the difficulty, at the expense of a further level of approximation. We are going to project every point on $(MP)$ so that to ensure alignment before performing interpolation. In other words, let

$$\mathbf{e} = \frac{\mathbf{a}^{n-1/2}}{\|\mathbf{a}^{n-1/2}\|} \tag{66}$$

be the unit vector associated to $\mathbf{a}^{n-1/2}$ and project the above points onto its direction. This yields

$$\begin{array}{ll} M_0^{'} = M_0 & P_0^{'} = P_0 \\ M_1^{'} = M_1 & P_1^{'} = P_0 \\ M_2^{'} = M_1 + \Delta t^{n-3/2}(\mathbf{a}^{n-3/2}\cdot\mathbf{e})\mathbf{e} & P_2^{'} = P_1 + \Delta t^{n-3/2}(\mathbf{a}^{n-3/2}\cdot\mathbf{e})\mathbf{e} \end{array} \tag{67}$$

as well as

$$M_\star^{'} = M_0 - \Delta^{n+1/2}(\mathbf{a}^{n+1/2}\cdot\mathbf{e})\mathbf{e}. \tag{68}$$

We are now ready to summarize what needs to be done:

1. PREDICTOR & SOURCE CORRECTOR
   To use the values of $u$ (modified by the source term) at *some* of the 6 points $M_{0,1,2}^{'}$ and $P_{0,1,2}^{'}$ to compute an interpolated value of $\tilde{u}$ at $M_\star^{'}$; recall that to each $M^{'}$ or $P^{'}$ point, there is an associated value of $u$, namely

$$\begin{array}{ll} M_0^{'} \leftarrow u_M^n e^{-\nu^{n-1/2}} & P_0^{'} \leftarrow v_{P_0}^n e^{-\nu^{n-1/2}} \\ M_1^{'} \leftarrow u_M^{n-1} e^{\nu^{n-1/2}} & P_1^{'} \leftarrow v_{P_0}^{n-1} e^{\nu^{n-1/2}} \\ M_2^{'} \leftarrow u_M^{n-2} e^{2\nu^{n-3/2}+\nu^{n-1/2}} & P_2^{'} \leftarrow v_{P_0}^{n-2} e^{2\nu^{n-3/2}+\nu^{n-1/2}} \end{array} \tag{69}$$

   where the $v_{P_0}$'s have been assessed by the face interpolation-truncation procedure $\mathcal{F}$. As for the interpolated value $\tilde{u}$ at $M_\star^{'}$, it actually represents an approximation of

$$\tilde{u} \simeq u_M^{n+1} e^{2\nu^{n+1/2}+\nu^{n-1/2}} \simeq u_M^{\star\star} e^{2\nu^{n+1/2}+\nu^{n-1/2}} \tag{70}$$

   The outcome of this predictor-source corrector step is, as before, denoted by $u_i^{\star\star}$. It is possible to replace the exponentials by the corresponding first-order expansions.

2. MONOTONICITY CORRECTOR
   To truncate $u_i^{\star\star}$, the predicted and partially corrected value, onto an interval defined by the values of $u$ (modified by the source term) at the 2 closest neighbors X enclosing the target on the line formed by the points $M'$ and $P'$.

Once this is done, we formally proclaim that

$$u_M^{n+1} = \mathcal{T}_{\{\lambda(MP)\},\{\nu_K\},\{M'\},\{P'\}}(\{v_{P'}\},\{u_{M'}\}), \tag{71}$$

where the subscripts $\{M'\}$ and $\{P'\}$ are specified in the abstract operator $\mathcal{T}$ in order to emphasis that, this time, the Lagrange interpolation in the predictor step must use absolute coordinates of the $M'$s and the $P'$s, while all elements of the list $\{\lambda(MP)\}$ have to be computed base on the distance $\|MP\|$, corresponding to time $t = n - 1/2$. Therefore, contrary to the uniform case, we cannot store and reuse some of the $\lambda(MP)$'s or some of the $v_{P'}$'s that were computed at previous time-levels.

It is now not difficult to combine the elements in this subsection and those in the previous one to get a scheme for a time- and space-variable velocity.

### 2.4 Conservativity-correction procedure

As for the 1-D case, when boundary conditions are not involved and when we have some natural bounds for $u$, we proceed to a global homothetic correction step after each time-iteration. The details are identical to subsection 1.4. The only difference is that now we have to agree upon some rule for evaluating the total discrete "mass" of the advected quantity $u$ in domains with a distorted mesh.

## 3 Numerical results

From now on, the scheme built up so far will be referred to as `ISE` (for Iserles). In [31], we have compared a 2-D version of `ISE` to two other 2-D schemes named `DON` (for Donor) [2, 36] and `CTU` (for Corner Transport Upwind) [6, 21, 35]. In this section, we are going to compare the 3-D versions of `ISE` (nonconservative and conservative) to their `DON` counterparts. The reason why we left out `CTU` is that its practical implementation for a 3-D deformed mesh is extremely painful, not to say impossible. We do not recall details about the Donor cell scheme (`DON`), since it is relatively well-known in the literature. It is enough for us to know that we use a second-order version with slope limitation over each cell. The gradient reconstruction method is either Ultrabee [9] (for Cartesian meshes) or Dukowicz and Kodis [12] (for irregular meshes).

### 3.1 An "industrial" expansion test

We consider the following test case, which is close to a real simulation. Over the domain $\bar{\Omega} = [0, 045] \times [0, 090] \times [0, 018]$ sketched out in Fig. 6, we consider:

1. The initial data $u^0$ whose support is

$$\mathcal{S} = [0.003, 0.006] \times [0.006, 0.009] \times [0.003, 0.006] \tag{72}$$

   The center of $\mathcal{S}$ is located at $C = (0.0045, 0.0075, 0.0045)$. There can be two types of initial data, namely,

   - a `SQUARE`, defined by

$$u^0(x, y, z) = \begin{cases} 1 & \text{if } (x, y, z) \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \tag{73}$$

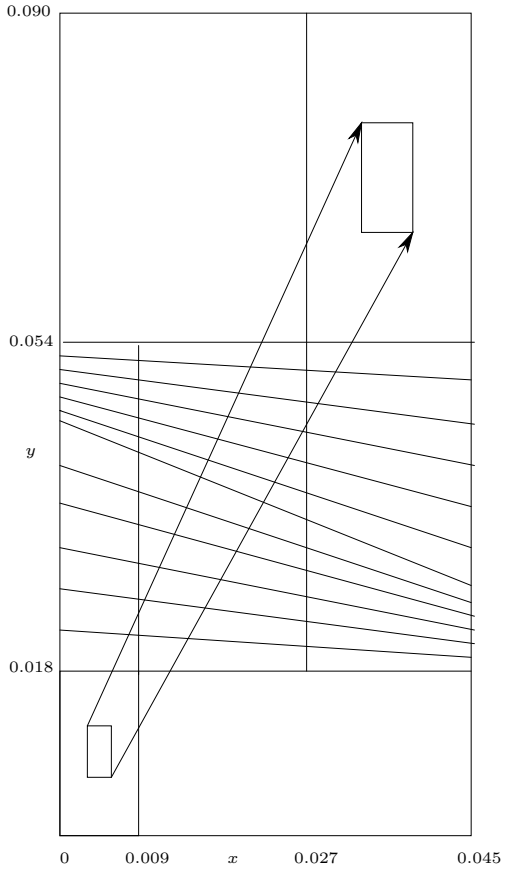


Figure 6: Expansion experiment.

- a `WAVELET`, defined by

$$u^0(x,y,z) = \begin{cases} \cos^2[4\pi(x-x_C)]\cos^2[4\pi(y-y_C)]\cos^2[4\pi(z-z_C)] & \text{if } (x,y,z) \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \tag{74}$$

2. The radial velocity field

$$\mathbf{a}(x,y,z) = 400\ln 2 \times \begin{pmatrix} x+0.027 \\ y+0.054 \\ z \end{pmatrix} \tag{75}$$

   The center of this velocity field lies outside $\bar{\Omega}$. Furthermore, we have div$\mathbf{a} = 1200\ln 2$. This divergence is constant, which makes it easy for us to determine the analytical solution.

If we carry out the simulation over $T = \frac{1}{400}$, then at the end of the simulation, the stretching factor in each direction will be equal to 2, while the attenuation factor in amplitude will

be equal to $8 = 2^3$. What was initially in $\mathcal{S}$ will be translated and expanded into $\mathcal{S}' = [0.033, 0.039] \times [0.066, 0.078] \times [0.006, 0.012]$, the center of which is $C' = (0.036, 0.072, 0.009)$. The domain $\bar{\Omega}$ itself can be meshed by a deformed mesh of size $45 \times 30 \times 18$, with distorted cells in the region $y \in [0.018, 0.054]$. In uniform regions, we naturally have

$$\Delta x = 0.001,\ \Delta y = 0.003,\ \Delta z = 0.001. \tag{76}$$

The maximal CFL ratio is set to 0.5, which gives $\Delta t = 1.13498 \times 10^{-5}$. The display window for the results is $[0.036, 0.045] \times [0.073, 0.090] \times [0, 0.018]$.

The main interest of this expansion experiment lies in the fact that it is inspired from a real industrial case, for which the initial data is sampled at a very coarse rate. Here, $\mathcal{S}$ merely contains $3 \times 2 \times 3$ cells, which falls below the traditional Nyquist rate. This is, however, a good test to compare various schemes in typically harsh conditions.

Figure 7 depicts the error distribution in the plane $z = 0.009$ of the display cube, for the conservative transport of a `SQUARE`. In the `ISE_S` panel, there is no conservativity-correction applied at each time iteration, while in the `ISE_Z` panel, the conservativity-correction procedure is activated. We clearly see that the error is much bigger for the `DON` scheme. To be convinced of this fact, let us proceed to 1-D cuts along the 3 principal directions of the display cube. In Fig. 8, we compare the 3 schemes in each cut panel. As far as the amplitude is concerned, `ISE_S` and `ISE_Z` are quite similar. As for `DON`, it is obviously worse by one order of magnitude.

Figures 9 and 10 plot the same results but associated with the `WAVELET` data. The observations are even more impressive here, insofar as no scheme is good enough to predict to maximal value, but `DON` remains the most diffusive. One can wonder why the situation is worse for `WAVELET`, a smooth function, than for `SQUARE`, a discontinuous data. The answer is: although we are working with highly accurate schemes, we are still very far from convergence, because the (deformed) mesh we are using is very coarse with respect to the data.

It is instructive to watch the behavior of the conservativity-correction procedure as time evolves. In each panel of Fig. 11, this behavior is represented by two curves: one for the total "mass" before correction, one for the total "mass" after correction. The left column corresponds to the `SQUARE` data, the right column to the `WAVELET`. The top row is associated with the coarse grid currently used, while the bottom row is associated with a sligthly refined version of it, that is, the same grid with 3 times more cells in the $y$-direction. It is interesting to see that the mass-correction procedure gets into some trouble only at the beginning of the simulation, but progressively, it manages to achieve a form of convergence. The unstable period is not related to neither the moment when the data starts entering the deformed zone ($t = 3.138 \times 10^{-4}$) nor on the moment when it comes out of the deformed ($t = 2.119 \times 10^{-3}$). The more the grid is refined, the milder the mass oscillations are. This testifies to the fact that the coarseness of the grid is the main reason for the mass defect encountered at the very first time iterations. Note that, at the end of the simulation, the total mass is fully conserved.

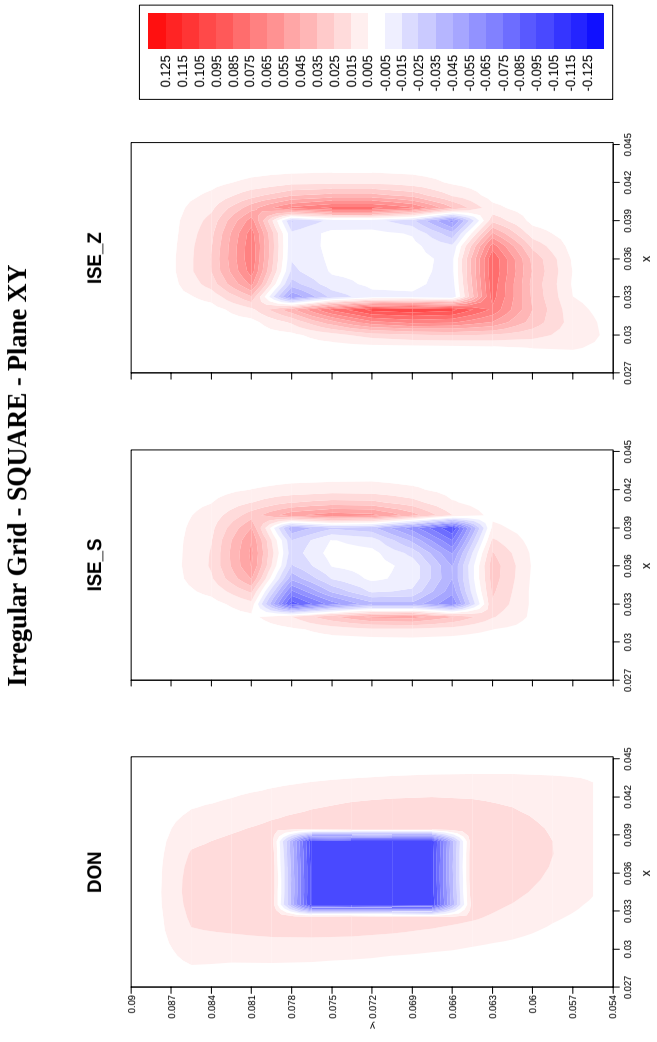


Figure 7: Expansion of `SQUARE` on grid A. 2-D cuts in plane $(x, y)$.

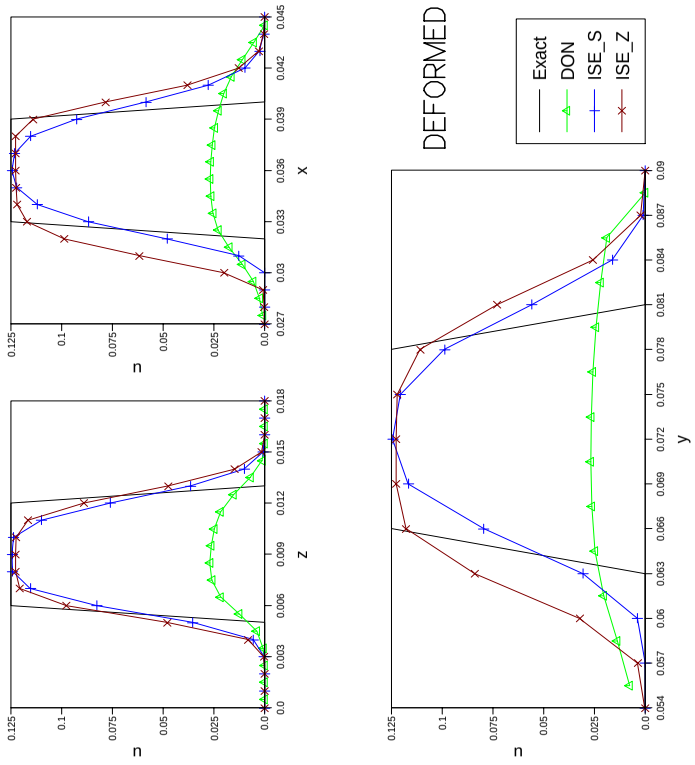


Figure 8: Expansion of `SQUARE` on grid A. 1-D cuts along 3 directions.

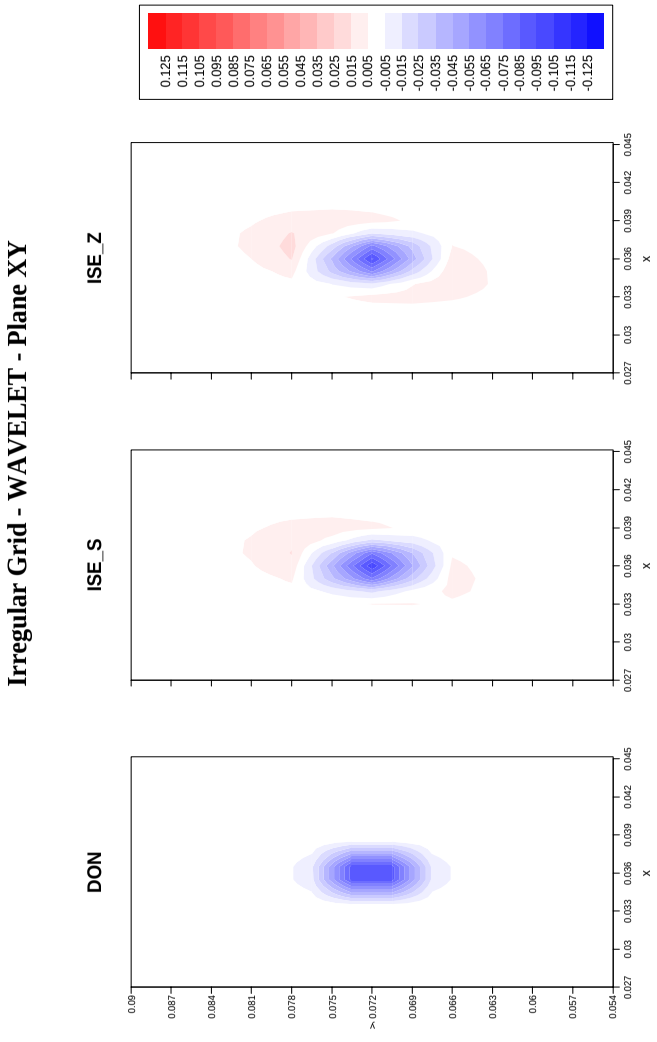


Figure 9: Expansion of `WAVELET` on grid B. 2-D cuts in plane $(x, y)$.

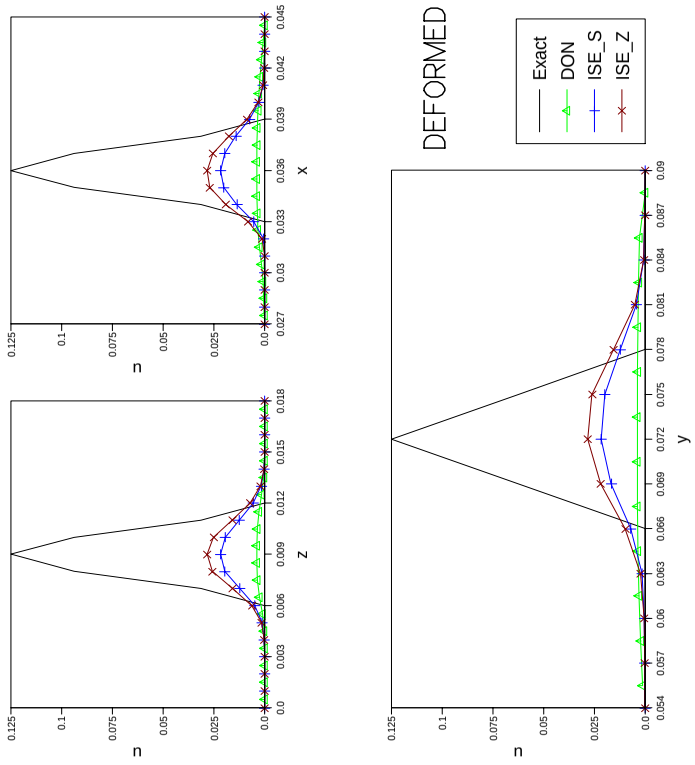


Figure 10: Expansion of `WAVELET` on grid B. 1-D cuts along 3 directions.

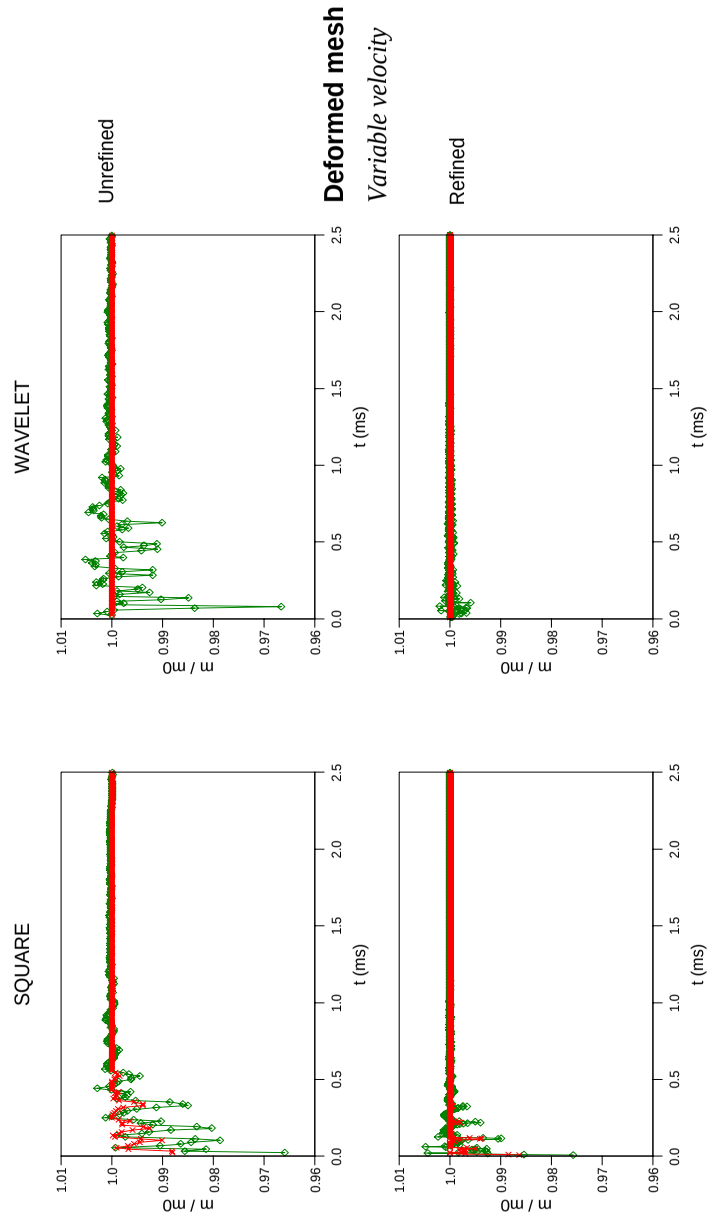


Figure 11: Expansion experiment. Total mass before and after correction.

### 3.2 An "academic" vortex test

We work with a velocity that depends only on time $t$. Let us consider

$$\mathbf{a}(t) = \pi \begin{pmatrix} \alpha\cos(\pi t) \\ \beta\sin(\pi t) \\ \gamma\sin(2\pi t) \end{pmatrix}. \tag{77}$$

This is a divergence-free velocity field. The trajectory of a point located at $(x_0, y_0, z_0)$ when $t = 0$ can be integrated analytically. We arrive at

$$\left| \begin{array}{l} x(t) = x_0 + \ \alpha\sin(\pi t) \\ y(t) = y_0 + \ \beta[1-\cos(\pi t)] \\ z(t) = z_0 + \frac{1}{2}\gamma[1-\cos(2\pi t)]. \end{array} \right. \tag{78}$$

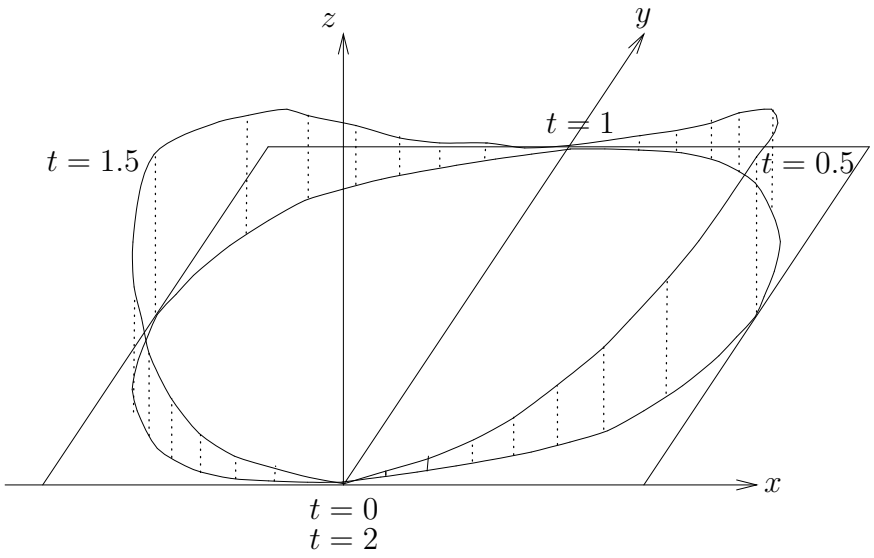


Figure 12: Vortex experiment on a regular Cartesian grid

Upon elimination of $t$, this 2-periodic trajectory can be seen as the intersection curve of two surfaces, namely: (1) the cylinder oriented in the $z$-direction, the basis of which is the ellipsis

$$\left(\frac{x-x_0}{\alpha}\right)^2 + \left(\frac{y-y_0-\beta}{\beta}\right)^2 = 1; \tag{79}$$

and (2) the cylinder oriented in the $y$-direction, the basis of which is the parabola

$$z - z_0 = \frac{\gamma}{\alpha^2}(x-x_0)^2. \tag{80}$$

These properties account for the name *vortex* that we attribute to the field defined by (77).

### 3.2.1 Regular grid

Let us consider the set of parameters

$$\bar{\Omega} = [0,1] \times [0,1] \times [0,1], \qquad S = [\tfrac{1}{4}, \tfrac{3}{4}] \times [\tfrac{1}{8}, \tfrac{5}{8}] \times [\tfrac{1}{8}, \tfrac{5}{8}], \qquad (\alpha, \beta, \gamma) = (\tfrac{1}{8}, \tfrac{1}{8}, \tfrac{1}{4}), \tag{81}$$

where the notations $\bar{\Omega}$ and $S$ bear the same meaning as before. The experiment is run until $T = 4$, the time for the initial data to make 2 complete rounds.

The mesh size is $\Delta x = \Delta y = \Delta z = \frac{1}{80}$. The largest CFL ratio is 0.4. Figure 13 display 2-D cuts of error distribution along the three orthogonal planes $z = \frac{3}{4}$, $x = \frac{1}{2}$, $y = \frac{3}{4}$. The fronts are sharper for `ISE` than for `DON`. We do not show 1-D cuts of computed solutions, since the conclusion is the same as in the uniform velocity case. From the standpoint of $L^1$-errors, the total error due to `ISE` is 3 times lesser that that of `DON`. If we conduct the same experiment with the `WAVELET` data, this factor goes up to 10.

### 3.2.2 Irregular grid

We now take

$$\bar{\Omega} = [0,6] \times [0, \frac{3}{2}] \times [0, \frac{3}{4}], \qquad S = [\tfrac{5}{2}, \tfrac{7}{2}] \times [\tfrac{1}{4}, \tfrac{5}{4}] \times [\tfrac{1}{8}, \tfrac{3}{8}], \qquad (\alpha, \beta, \gamma) = (\tfrac{9}{4}, 0, \tfrac{1}{4}), \tag{82}$$

For $x \in [0, \frac{9}{4}] \cup [\frac{15}{4}, 6]$, the cells are not cubes but trapezoidal prisms in the $z$-direction. This trapezoid mesh is a copy of that used in [31].

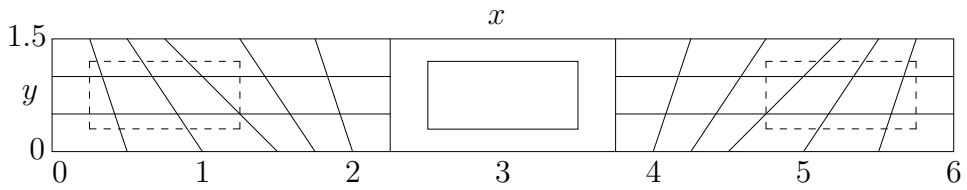


Figure 14: Vortex experiment on a Kershaw-like grid.

Since $\beta = 0$, the trajectories defined by (78) are pieces of parabola lying in the $(x, z)$-plane. The choice $\beta = 0$ is meant to minimize the size of the computational domain. It does not harm the relevancy of the experiment, insofar as the trajectories actually cross the irregularities of the mesh. In Fig. 14, the dotted rectangles represent the extreme positions of the data during the motion.

Figure 15 corresponds to 2-D cuts of error distribution along the three orthogonal planes $z = \frac{1}{4}$, $x = 3$, $y = \frac{3}{4}$. The conjunction of irregular mesh and time-dependent velocity appears to be a difficult challenge for both schemes. The results are not as clean as in the uniform velocity case or for the regular grid. However, `ISE` remains unquestionably an order of magnitude better that `DON`. The ratios between the $L^1$-errors is about 2.5 for the `SQUARE` data and 8.5 for the `WAVELET` data.

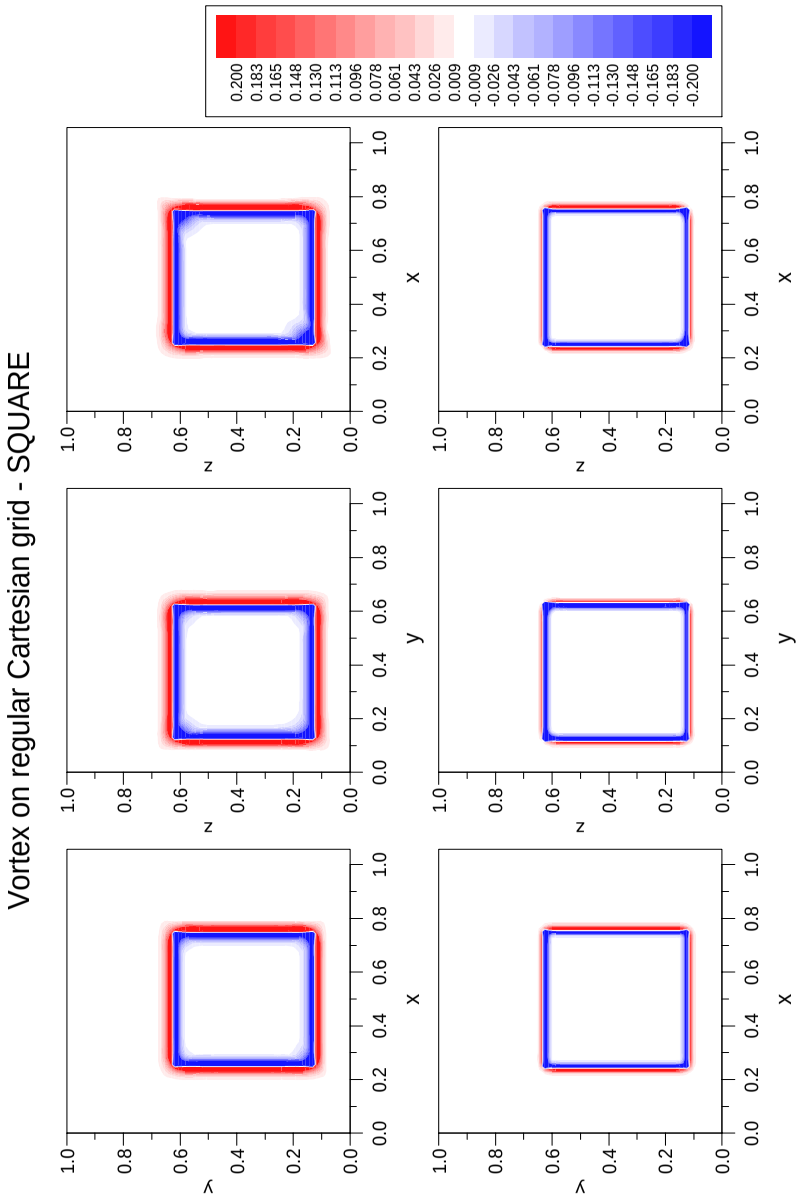


Figure 13: Vortex of `SQUARE` by `DON` (upper row) et `ISE` (lower row) on a regular Cartesian mesh. 2-D cuts of error distribution.

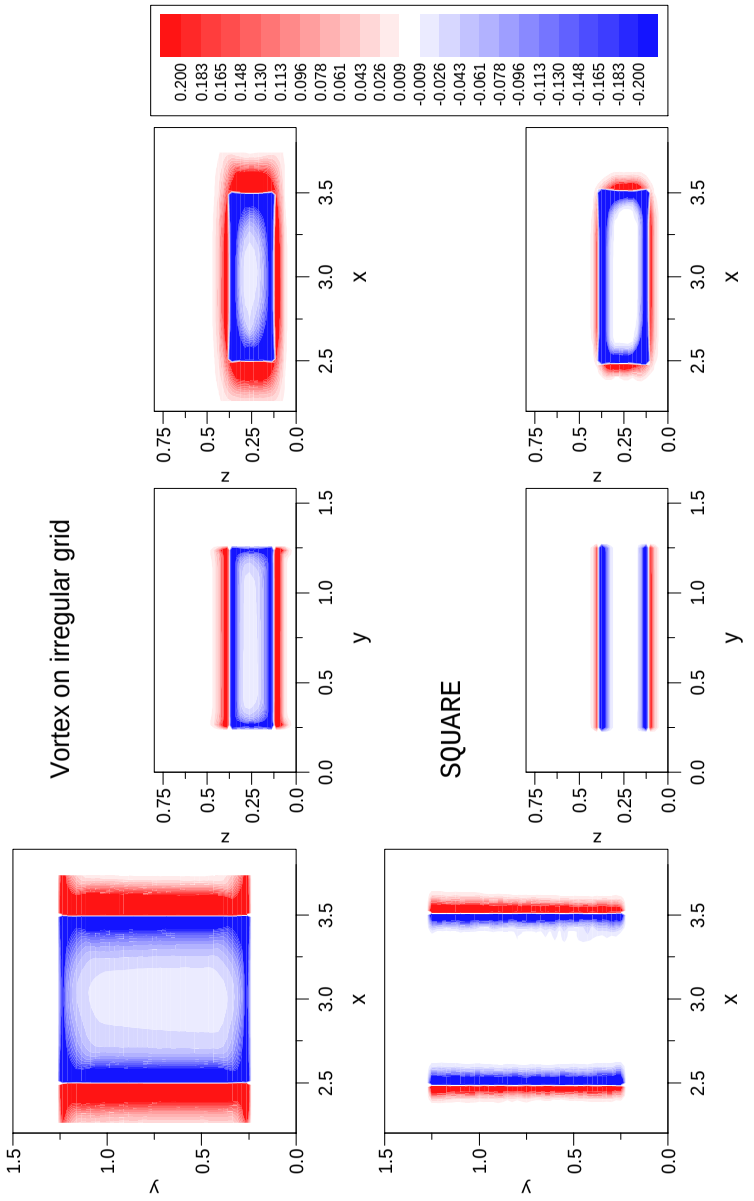


Figure 15: Vortex of `SQUARE` by `DON` (upper row) et `ISE` (lower row) on a Kershaw-like mesh. 2-D cuts of error distribution.

## Conclusion

Despite its apparent simplicity, linear advection is far from being a trivial matter, especially when it comes to multidimensional numerical schemes. The new scheme we have been putting forward in [31] and in the present paper is, hopefully, a possible answer to the quest for accuracy over distorted meshes. This answer is based on the one-dimensional nature of advection (at least for a divergence-free velocity field or for the color equation), which suggests one to apply one-dimensional schemes along flowlines. The suitability of this scheme for unstructured meshes is ensured by the compactness of the corresponding one-dimensional stencil, while its accuracy relies on the multiple time-level feature.

A relatively heavy machinery is nevertheless necessary to support this key idea. This machinery, which aims at assessing missing informations with a high degree of accuracy while maintaining monotonicity and at dealing with time-dependent velocity fields, turns out to be quite expensive. For time-dependent velocities, it turns out to be twice more memory-demanding and 8–10 times more time-consuming than the original Donor counterpart. As was mentioned in [31], however, this project was undertaken with the assumption that we are willing to pay the price.

Another tricky —not to say controversial— aspect of the scheme we propose is conservativity. This aspect was deliberately not investigated in [31], since at that time, we were above all concerned with the possibility of extending the Iserles-Roe stencil to various cases of the nonconservative transport equation. It now appears, and this should not come as a surprise to anybody, that it is almost impossible to reconcile various good theoretical properties of a scheme, namely: (i) monotonicity; (ii) accuracy; (iii) compacity; (iv) conservativity. Depending on the type problems we have to deal with, there is a choice we have to make regarding how we want the results to behave. In [31], priority was given to (i), (ii) and (iii). In this paper, based on practical run tests, we require a "weak" version of (iv), that is, conservativity almost-everywhere. The drawback we then have to consent is a worsening of (ii). Moreover, the conservativity-correction procedure we introduce here lies on some peculiar features related to the nature of the unknown variable $u$, and we do not claim it to be extendable to every kind of problems.

Notwithstanding cost considerations, if our primary objective is to improve the accuracy of the results, it is a good news to learn that the $L^1$-errors due to the new scheme are systematically smaller that the errors due to Donor, even for very coarse meshes. Thus, the conservativity-correction procedure does not seem to destroy too much (ii). As a matter of fact, if we take an industrial 3-D test case with its typical coarse mesh, the new scheme always performs better than Donor.

## Acknowledgments

The authors wish to thank Arnaud Torrès, Julien Bohbot and Marc Zolver for many helpful discussions and suggestions.

## References


[1] Abgrall R. and Mezine M. (2003), "Construction of Second Order Accurate Monotone and Stable Residual Distributive Schemes for Unsteady Flow Problems," *J. Comput. Phys.* **188**, 16–55.

[2] Amsden A.A., O'Rourke P.J. and Butler T.D., *KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays*, Report LA-11560-MS, Los Alamos National Laboratory, 1989.

[3] Bell J.B., Dawson C.N. and Shubin G.R. (1988), "An Unsplit, Higher Order Godunov Method for Scalar Conservation Laws in Multiple Dimensions," *J. Comput. Phys.* **74**, 1–24.

[4] Ciarlet P.G. (1991), "Introduction to the Finite Element Method," in *Handbook of Numerical Analysis* **II**, 17–351, eds. Ciarlet P.G. and Lions J.L., Elsevier Science Publishers, North-Holland.

[5] Cockburn B., Hou S. and Shu C.W. (1990), "The Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws IV: The Multidimensional Case," *Math. Comp.* **54**, 545–581.

[6] Colella P. (1990), "Multidimensional Upwind Methods for Hyperbolic Conservation Laws," *J. Comput. Phys.* **87**, 171–200.

[7] Davis P., *Interpolation and Approximation*, Dover, New-York, 1975.

[8] Deconinck H., Struijs R. and Bourgois G. (1993), "Compact Advection Schemes on Unstructured Grids," in *VKI Lecture Series 1993-04 on CFD*, von Karman Institute, Belgium.

[9] Després B. and Lagoutière F. (1999), "Un schéma non-linéaire anti-dissipatif pour l'équation d'advection linéaire," *C. R. Acad. Sci. Paris* **238**, I, 939–944.

[10] Després B. and Loubère R. (2005), "Convergence of Repair Algorithms in 1-D," preprint. `http://www.ann.jussieu.fr/publications/2005/R05013.pdf`

[11] Dhatt G. and Touzot G., *Une Présentation de la Méthode des Éléments Finis*, Maloine, Paris, 1984.

[12] Dukowicz J.K. and Kodis J.W. (1987), "Accurate Conservative Remapping for Arbitrary Lagrangian-Eulerian Computations," *SIAM J. Sci. Stat. Comp.* **8**, 305–321.

[13] Engquist B., Lötstedt P. and Sjögreen B. (1989), "Nonlinear Filters for Efficient Shock Computation," *Math. Comp.* **52**, 509–537.

[14] Fauvin D. *Étude de Schémas Non-Dissipatifs pour Certaines Équations Hyperboliques*, CEA/DAM/CELV Technical Report, 1995.

[15] Godlewski E. and Raviart P.A., *Hyperbolic Systems of Conservation Laws*, Mathématiques et Applications, SMAI, Ellipses, Paris, 1991.

[16] Harten A., Lax P.D. and Van Leer B. (1983), "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws," *SIAM Review* **25**, 35–61.

[17] Iserles A. (1986), "Generalised Leapfrog Schemes," *IMA J. Numer. Anal.* **6**, 381–392.

[18] Kim C.W. (2003), "Accurate Multi-Level Schemes for Advection," *Int. J. Numer. Meth. Fluids* **41**, 471–494.

[19] Kim C.W., *Multi-Dimensional Upwind Leapfrog Schemes and their Applications*, partial PhD dissertation, University of Michigan, 1997.

[20] Kucharik M., Shashkov M. and Wendroff B. (2003), "An Efficient Linearity-and-Bound-Preserving Remapping Method," *J. Comput. Phys.* **188**, 462–471.

[21] LeVeque R.J. (1996), "High-Resolution Conservative Algorithms for Advection in Incompressible Flow," *SIAM J. Numer. Anal.* **33**, 627–665.

[22] LeVeque R.J., *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.

[23] Loubère R., Staley M. and Wendroff B., *The Repair Paradigm: New Algorithms and Applications to Compressible Flows,* Los Alamos Technical Report LAUR-04-0795, 2004.

[24] Morel J.E., Dendy J.E., Jr., Hall M.L. and White S.W. (1992), "A Cell-Centered Lagrangian-Mesh Diffusion Differencing Scheme," *J. Comput. Phys.* **103**, 286–299.

[25] Paillère H., Boxho J., Degrez G. and Deconinck H. (1996), "Multidimensional Upwind Residual Distribution Schemes for the Convection–Diffusion Equation," *Int. J. Numer. Meth. Fluids* **23**, 923–936.

[26] Roe P.L. and Sidilkover D. (1992), "Optimum Positive Linear Schemes for Advection in Two and Three Dimensions," *SIAM J. Numer. Anal.* **29**, 1542–1568.

[27] Roe P.L. (1998), "Linear Bicharacteristic Schemes without Dissipation," *SIAM J. Sci. Comp.* **19**, 1405–1427.

[28] Shashkov M. and Wendroff B. (2004), "The Repair Paradigm and Application to Conservation Laws," *J. Comput. Phys.* **198**, 265–277.

[29] Sidilkover D. and Roe P.L., *Unification of Some Advection Schemes in Two Dimensions*, ICASE Report 95-10, 1995.

[30] Sweby P.K. (1984), "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws," *SIAM J. Numer. Anal.* **21**, 995–1011.

[31] Tran Q.H. and Scheurer B. (2002), "High-Order Monotonicity-Preserving Compact Schemes for Linear Scalar Advection on 2-D Irregular Meshes," *J. Comput. Phys.* **175**, 454–486.

[32] Tran Q.H. and Scheurer B. (2002), "High-Order Monotonicity-Preserving Compact Schemes for Linear Advection on 2-D Irregular Meshes," in *Finite Volumes for Complex Applications* **III**, 437–444, editors Herbin R. and Kröner D., Hermes Penton Science, London.

[33] Tran Q.H., *Schémas de Type Multidimensionnel en Maillage Déformé Structuré pour l'Advection Scalaire Linéaire. IV: Passage en 3-D*, IFP Technical Report 57295, 2003.

[34] Tran Q.H., *Schémas de Type Multidimensionnel en Maillage Déformé Structuré pour l'Advection Scalaire Linéaire. III: Champs de vitesse variables en temps*, IFP Technical Report 56740, 2002.

[35] Van Leer B. (1984), "Multidimensional Explicit Difference Schemes for Hyperbolic Conservation Laws," in *Computing Methods in Applied Sciences and Engineering* **VI**, 493–497, editors Glowinski R. and Lions J.-L., Elsevier Science Publishers, North-Holland.

[36] Zolver M., Klahr D., Bohbot J., Laget O. and Torrès A. (2003), "Reactive CFD in Engines with a New Unstructured Parallel Solver," *Oil & Gas Science and Technology* **58**, 33–46.