Journal of Computational Physics 526 (2025) 113784

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A fully semi-Lagrangian technique for viscous and dispersive conservation laws

R. Ferretti

Dipartimento di Matematica e Fisica, Università Roma Tre, Largo S. Leonardo Murialdo, 1, Roma, Italy

A R T I C L E I N F O

MSC:
65M06
65M12
65M25

Keywords:
Conservation laws
Dispersive equations
Semi-Lagrangian schemes

A B S T R A C T

We consider a scheme of Semi-Lagrangian type for the approximation of conservation laws with convex or concave flux. Following a previously proposed approach, the scheme treats in explicit form the diffusive/dispersive terms, although it still requires an implicit phase to determine the correct characteristic. We prove consistency and one-sided Lipschitz stability in the viscous case, and show that, despite being nonconservative, the scheme has good performances and negligible conservation error with large Courant numbers as far as the diffusion scales are resolved. Then, the approach is generalized to conservation laws with dispersive terms (in particular, the Korteweg–de Vries equation). We also validate the scheme with a set of numerical examples.

1. Introduction

Born in the 50s in the framework of environmental fluid dynamics, semi-Lagrangian (SL) schemes have become in recent years a useful tool to treat various PDE models, mainly of hyperbolic type. In its basic formulation, a SL scheme works by discretizing a characteristics-based representation formula for the solution of a hyperbolic equation. For example, in the easiest case of the constant-coefficient advection equation,

$$\begin{cases} u_t + a u_x = 0 \qquad (x,t) \in \mathbb{R} \times \mathbb{R}^+, \\ u(x,0) = u_0(x), \end{cases}$$

the solution may be represented in the well-known form

$$u(x,t) = u_0(x - at),$$

or, using a discrete timestepping,

$$u(x, t + \Delta t) = u(x - a\Delta t, t), \tag{1}$$

with a time discretization step Δt. This representation formula is discretized in SL form as

$$\begin{cases} v_j^{n+1} = I[V^n](x_j - a\Delta t), \\ v_j^0 = u_0(x_j). \end{cases} \tag{2}$$

E-mail address: roberto.ferretti@uniroma3.it.

https://doi.org/10.1016/j.jcp.2025.113784
Received 25 May 2024; Received in revised form 22 January 2025; Accepted 23 January 2025

Available online 27 January 2025
0021-9991/© 2025 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

In (2), $I[V](x)$ represents an operator of interpolation on the values v_i of the vector V, computed at the point x, which replaces the exact computation of the solution in (1). In this paper, the values v_i will be associated to an infinite regular space grid of points $x_i = i\Delta x$. Moreover, given the time step Δt, the notation v_j^n will denote the approximation of the solution at the node x_j and time $t_n = n\Delta t$, V^n denoting the vector collecting all such values.

This approximation strategy was first proposed in [5], then extensively used in the field of environmental fluid dynamics [14], and, independently, in computational plasma physics (see the seminal paper [6], and, among the wide later literature, [13]). In addition, it has been applied to a variety of situations, like level set methods and Hamilton–Jacobi equations (we refer the reader to the extensive review in [8]). The crucial, appealing feature of SL schemes is to allow, by exploiting the representation formula of the solution, for Courant numbers far beyond the unity, still preserving stability.

In presence of a diffusion term, i.e., considering the advection–diffusion equation

$$\begin{cases} u_t + au_x = \nu u_{xx} \quad (x,t) \in \mathbb{R} \times [0,T], \\ u(x,0) = u_0(x), \end{cases} \tag{3}$$

a possible way of extending this approach is via the scheme

$$v_j^{n+1} = \frac{1}{2} I[V^n](x_j - a\Delta t - \sqrt{2}\,\delta) + \frac{1}{2} I[V^n](x_j - a\Delta t + \sqrt{2}\,\delta), \tag{4}$$

in which

$$\delta = \sqrt{\nu \Delta t}. \tag{5}$$

An explicit treatment of diffusion like (4) has an established tradition, which originates from the Feynman–Kac stochastic representation formula for the solution of (3), see [4]. Among the vast literature, we refer the reader to [9] for a general review, and to [3,2] for more recent developments. This approach has a strong link with the probabilistic representation for solutions of parabolic problems [12], as well as with the so-called *Markov chain approximations*, see [10].

The aim of this work is to apply this numerical framework to conservation laws of the form

$$\begin{cases} u_t + f(u)_x = \nu \dfrac{\partial^l u}{\partial x^l} \quad (x,t) \in \mathbb{R} \times [0,T], \\ u(x,0) = u_0(x), \end{cases} \tag{6}$$

and, possibly, to more general versions in multiple dimensions and/or with vector-valued solutions. This requires to extend the technique in the direction of both a nonlinear advection and a dispersive right-hand side.

We start by listing some standing assumptions on the problem, to be kept along the whole paper.

Basic assumptions

- The flux $f(u)$ is either convex or concave, i.e., its derivative $f'(u)$ is monotone. In what follows we will assume f to be convex, the reverse case being an obvious modification;
- $f(u) \in C^2$ and $0 < m_f \le |f''(u)| \le M_f$;
- $\nu \neq 0$;
- The interpolation I is non-expansive in $W^{1,\infty}$.

The last assumption is satisfied, for example, by a piecewise linear ($\mathbb{P}_1$) interpolation. Note that if an interpolation is monotone and invariant for translations of multiples of Δx, then it satisfies this assumption, as proved in [7]. In the numerical tests, we will show that nonmonotone, higher order interpolations can also be valuable; in particular, we will use spline interpolation.

The paper is structured as follows. In Sec. 2, we construct and analyze the scheme for (6) in the special case of viscous conservation laws ($l = 2$). Section 3 replies this study for the case of dispersive operators, with a special focus on the Korteweg–de Vries (KdV) equation. Last, in Sections 4–5 we present a numerical validation for the scheme and draw some conclusions.

2. Construction and analysis of the scheme: the viscous case

First, we start with a second-order r.h.s. as in (3), so that $l = 2$. Recall that (6) may be rewritten in the advective form

$$u_t + f'(u)u_x = \nu u_{xx}, \tag{7}$$

which shows, as it is well-known, that the advection speed for a given value u is $f'(u)$.

The most straightforward idea to adapt (4) to the nonlinear case (7) is to consider the value v_j^{n+1} (and hence its propagation speed $f'(v_j^{n+1})$) as an unknown, to be determined so as to satisfy

$$\begin{cases} v_j^{n+1} = \frac{1}{2} I[V^n](x_j - f'(v_j^{n+1})\Delta t - \sqrt{2}\,\delta) + \frac{1}{2} I[V^n](x_j - f'(v_j^{n+1})\Delta t + \sqrt{2}\,\delta) \\ v_j^0 = u_0(x_j). \end{cases} \tag{8}$$

Here, δ is defined by (5), and $V^n = (v_1^n \cdots v_N^n)$. Note that the equations to be solved at each step are decoupled: the j-th equation contains only the unknown v_j^{n+1}. Note also that (8) always admits a solution if V^n is bounded. In fact, the function $\phi_j(v) = v - T_j(v)$ is continuous and satisfies $v - \|V^n\|_\infty \le \phi(v) \le v + \|V^n\|_\infty$, so that

$$\lim_{v\to\pm\infty} \phi(v) = \pm\infty,$$

which implies the existence of at least one solution $\bar v = T_j(\bar v)$.

Approximations in the form (8) are not new. In particular, a similar technique is proposed in [11] for a general form of semilinear parabolic equations which includes (7). However, the present study differs from [11] in various respects. First, we avoid here the probabilistic approach, which limits the applicability of this technique to the second-order case; as a result, we are able to extend the technique to higher order operators. Second, the computation of the advection speed $f'(u)$ is carried out in [11] in explicit form, using the values of the solution at the previous time step. While computing the speed in the implicit form (8) may be in principle more complex, it also avoids crossing of characteristics and may treat steep transitions of the solution with less severe restrictions on the time step.

2.1. Consistency

To prove consistency, assume that the elements of the vector V^n are the samples of a smooth solution of (6), so that $v_j^n = u(x_j, t_n)$. Assuming that the interpolation is accurate with order r on smooth functions, i.e.,

$$\|I[V^n] - u(t_n)\|_\infty \le C\Delta x^r,$$

we can write the consistency condition for (8) as

$$\begin{aligned} u(x_j,t_{n+1}) = {} & \frac{1}{2}u(x_j - f'(u(x_j,t_{n+1}))\Delta t - \sqrt{2}\,\delta, t_n) \\ & + \frac{1}{2}u(x_j - f'(u(x_j,t_{n+1}))\Delta t + \sqrt{2}\,\delta, t_n) + O(\Delta x^r) + \frac{1}{\Delta t}L_{\Delta x,\Delta t}(x_j,t_n), \end{aligned} \tag{9}$$

where $L_{\Delta x,\Delta t}(x_j,t_n)$ is the local truncation error of the scheme, and the interpolation error has been taken into account.

Rewriting $u(x_j - f'(u(x_j,t_{n+1}))\Delta t \pm \sqrt{2}\,\delta, t_n)$ as

$$\begin{aligned} u(x_j - f'(u(x_j,t_{n+1}))\Delta t \pm \sqrt{2}\,\delta, t_n) &= u(x_j,t_n) + u_x(x_j,t_n)\left(-f'(u(x_j,t_{n+1}))\Delta t \pm \sqrt{2}\,\delta\right) \\ &\quad + \frac{1}{2}u_{xx}(x_j,t_n)\left(-f'(u(x_j,t_{n+1}))\Delta t \pm \sqrt{2}\,\delta\right)^2 \\ &= u(x_j,t_n) - u_x(x_j,t_n)f'(u(x_j,t_n))\Delta t \\ &\quad \pm u_x(x_j,t_n)\sqrt{2}\,\delta + O(\Delta t^2) + u_{xx}(x_j,t_n)\delta^2 \\ &= u(x_j,t_n) - u_x(x_j,t_n)f'(u(x_j,t_n))\Delta t \\ &\quad \pm u_x(x_j,t_n)\sqrt{2}\,\delta + O(\Delta t^2) + \nu u_{xx}(x_j,t_n)\Delta t \end{aligned}$$

in which we have used the definition (5) for δ, approximated $u(x_j,t_{n+1})$ as $u(x_j,t_n) + O(\Delta t)$ and neglected the higher order terms. Using this expansion in (9), we get therefore:

$$\begin{aligned} \frac{u(x_j,t_{n+1}) - u(x_j,t_n)}{\Delta t} + u_x(x_j,t_n)f'(u(x_j,t_n)) &= \nu u_{xx}(x_j,t_n) \\ &\quad + O(\Delta t) + O\left(\frac{\Delta x^r}{\Delta t}\right) + L_{\Delta x,\Delta t}(x_j,t_n), \end{aligned}$$

which results, considering that the incremental ratio in the left-hand side is a first-order approximation of u_t, in a truncation error of the form

$$L_{\Delta x,\Delta t}(x_j,t_n) = O(\Delta t) + O\left(\frac{\Delta x^r}{\Delta t}\right). \tag{10}$$

In particular, in the simplest case of a piecewise linear ($\mathbb{P}_1$) interpolation, the consistency error is $O(\Delta t) + O(\Delta x^2/\Delta t)$ and the scheme is formally first-order under linear $\Delta t/\Delta x$ relationships.

Note that this analysis is targeted at first-order consistency. Achieving higher orders in time discretization requires to consider a more complex interaction between the advection and the diffusion term. In the linear case, some ideas in this direction are given in [9], [2].

2.2. Stability

As a first remark, note that the ℓ^∞ stability of the scheme is obvious under the basic assumptions since, from (8),

$$|v_j^{n+1}| \le \frac{1}{2}\|V^n\|_\infty + \frac{1}{2}\|V^n\|_\infty = \|V^n\|_\infty,$$

and hence,

$$\|V^n\|_\infty \le \|V^0\|_\infty.$$

Moreover, we can prove that, for all times and independently of the viscosity ν, one-sided Lipschitz stability (as used, e.g., in [15]) holds. Let

$$L_n^+ = \sup_j \frac{v_{j+1}^n - v_j^n}{\Delta x}.$$

To prove an upper bound on L_n^+, we write (8) at the nodes j and $j+1$, obtaining

$$\begin{aligned} v_{j+1}^{n+1} - v_j^{n+1} &= \frac{1}{2}\left(I[V^n](z_{j+1}^-) - I[V^n](z_j^-)\right) + \frac{1}{2}\left(I[V^n](z_{j+1}^+) - I[V^n](z_j^+)\right) \\ &\le \frac{1}{2}L_n^+\left(\left(z_{j+1}^- - z_j^-\right) + \left(z_{j+1}^+ - z_j^+\right)\right). \end{aligned} \tag{11}$$

On the other hand, via the Lagrange theorem, we get, for some unknown value $\bar v$,

$$\begin{aligned} z_{j+1}^\pm - z_j^\pm &= \Delta x - \left(f'(v_{j+1}^{n+1}) - f'(v_j^{n+1})\right)\Delta t \\ &\le \Delta x - f''(\bar v)\left(v_{j+1}^{n+1} - v_j^{n+1}\right)\Delta t, \end{aligned}$$

which, used into (11), gives

$$v_j^{n+1} - v_j^n \le L_n^+\Delta x - L_n^+ f''(\bar v)\Delta t\left(v_j^{n+1} - v_j^n\right),$$

and hence,

$$\left(1 + L_n^+ f''(\bar v)\Delta t\right)\left(v_j^{n+1} - v_j^n\right) \le L_n^+\Delta x.$$

This finally provides the bound

$$L_{n+1}^+ \le \frac{L_n^+}{1 + m_f\Delta t L_n^+} < L_n^+,$$

that is,

$$L_{n+1}^+ \le L_0^+. \tag{12}$$

Note that this estimate holds regardless of the value of ν, and that it ensures an entropic behaviour of the scheme. For example, according to (12), (8) cannot generate a shock (or, more precisely, a steep transition) with outgoing characteristics (i.e., with a positive jump).

2.3. Iterative solution of the implicit form

Define now

$$z_j^\pm(v) = x_j - f'(v)\Delta t \pm \sqrt{2}\,\delta,$$

so that the two points at which the interpolation is computed in (8) will be denoted shortly as $z_j^\pm(v_j^{n+1})$, and (8) will be rewritten as

$$v_j^{n+1} = \frac{1}{2}I[V^n](z_j^-(v_j^{n+1})) + \frac{1}{2}I[V^n](z_j^+(v_j^{n+1})),$$

which shows that solving the scheme for v_j^{n+1} amounts to find a fixed point for the equation $v = T_j(v)$, where

$$T_j(v) = \frac{1}{2}I[V^n](z_j^-(v)) + \frac{1}{2}I[V^n](z_j^+(v)).$$

In order to find conditions under which T is a contraction (and therefore, (8) might be solved by a fixed-point iteration), we compute

$$T_j'(v) = -\frac{\Delta t f''(v)}{2}\left(I'[V^n](z_j^-(v)) + I'[V^n](z_j^+(v))\right)$$

and note that, whenever the derivative $I'[V^n]$ is bounded, then

$$\|T_j'\|_\infty \le \Delta t M_f \|I'[V^n]\|_\infty$$

and T_j is a global contraction (therefore, (8) has a unique solution v_j^{n+1}) for

$$\Delta t < (M_f \|I'[V^n]\|_\infty)^{-1}. \tag{13}$$

If we consider, for example, a $\mathbb{P}_1$ space interpolation, we have $\|I'[V^n]\|_\infty \le \|V^n\|_\infty/\Delta x$ and therefore

$$\Delta t < \frac{\Delta x}{M_f \|V^n\|_\infty},$$

which shows that a linear $\Delta t/\Delta x$ relationship suffices to ensure contractivity of T_j. In practice, the iterative solution is obtained under a far less restrictive condition on Δt. In particular, if both $z_j^-(v)$ and $z_j^+(v)$ fall in an interval in which the numerical solution is increasing, then, by (12),

$$-\Delta t M_f L_0^+ \le -\Delta t f''(v) L_n^+ \le T_j'(v) \le 0,$$

which shows that T_j is a contraction for $\Delta t < \frac{1}{M_f L_0^+}$.

In the numerical examples of this paper, V^{n+1} has always been computed using V^n as an initial approximation. Note that, since the contraction coefficient of T_j is of order Δt, the global consistency rate is preserved even in case (8) is solved with a very small number of fixed point iterations; the lack of precision in solving (8) may result in higher conservation errors, though. For simplicity, in the numerical tests, we have used a fixed number of 5 to 10 iterations, without detecting relevant inaccuracies in this phase. Anyway, convergence might be accelerated by solving (8) via a secant method.

2.4. Treating systems of conservation laws

The approach outlined above can also be applied in the case of systems of viscous conservation laws, whenever they are solved in terms of Riemann invariants, as it happens, for example, in the one-dimensional Shallow Water Equations – in this case, the diffusive term should be rewritten in terms of the transformed variables, which is a slight additional difficulty.

An example which can be treated in this form is provided by the two-dimensional (or multi-dimensional) viscous Burgers' equation:

$$\begin{cases} u_t + uu_{x_1} + vu_{x_2} = \nu \Delta u \\ v_t + uv_{x_1} + vv_{x_2} = \nu \Delta v. \end{cases} \tag{14}$$

In this case, the characteristic field is (u, v) and the scheme can be written, with an obvious extension of the one-dimensional case, as

$$\begin{cases} u_j^{n+1} = \dfrac{1}{4}\displaystyle\sum_{k=1}^{4} I[U^n](x_j - u_j^{n+1}\Delta t - v_j^{n+1}\Delta t + \delta_k) \\ v_j^{n+1} = \dfrac{1}{4}\displaystyle\sum_{k=1}^{4} I[V^n](x_j - u_j^{n+1}\Delta t - v_j^{n+1}\Delta t + \delta_k) \end{cases} \tag{15}$$

where

$$\delta_k = \pm 2\delta e_i, \quad (k = 1, \dots, 4),$$

(with e_i $(i = 1, 2)$ denoting the canonical basis in $\mathbb{R}^2$) for all combination of both the sign and the index $i = 1, 2$, and δ defined again by (5). In this case, the equations defining u_j^{n+1} and v_j^{n+1} are coupled, but they remain decoupled among different nodes, i.e., for different values of the index j. Instead of the fixed point iteration in (15), it is possible to obtain u_j^{n+1} and v_j^{n+1} with a quasi-Newton solver like the Broyden method.

We refer to [9,3] for a more in-depth analysis of the diffusive term in the two-dimensional case. We also remark that (15) has been used in [1] to treat the advection–diffusion step in a Chorin–Temam type scheme for the Navier–Stokes equation.

Note that the diffusion substep could be completely separated from the advection substep by turning (14) into a fractional steps form, for example in the form

$$\begin{cases} u_j^{n+1/2} = \dfrac{1}{4}\displaystyle\sum_{k=1}^{4} I[U^n](x_j+\delta_k) \\ v_j^{n+1/2} = \dfrac{1}{4}\displaystyle\sum_{k=1}^{4} I[V^n](x_j+\delta_k) \\ u_j^{n+1} = I[U^{n+1/2}](x_j - u_j^{n+1}\Delta t - v_j^{n+1}\Delta t) \\ v_j^{n+1} = I[V^{n+1/2}](x_j - u_j^{n+1}\Delta t - v_j^{n+1}\Delta t). \end{cases} \tag{16}$$

3. Construction and analysis of the scheme: the dispersive case

In order to treat dispersive operators as in the general form (6), we first construct the abstract difference operator associated to the linear dispersive equation

$$u_t = \nu \frac{\partial^l u}{\partial x^l}.$$

In particular, we look for a time-discrete approximation in the form of a linear combination of N_δ values at displaced points $x+\delta_k$, from the previous time level:

$$\begin{cases} w(x,t_{n+1}) = \displaystyle\sum_{k=1}^{N_\delta} a_k w(x+\delta_k, t_n) \\ w(x,0) = u_0(x), \end{cases} \tag{17}$$

with $\delta_k \to 0$ for $\Delta t \to 0$. We will first examine consistency and stability for this abstract difference operator.

3.1. Consistency

The minimal consistency condition for (17) reads

$$\sum_{k=1}^{N_\delta} a_k u(x+\delta_k, t_n) = u(x,t_n) + \Delta t\, \nu \frac{\partial^l u}{\partial x^l} + o(\Delta t). \tag{18}$$

Rewriting the values $u(x+\delta_k,t_n)$ in (18) via a Taylor expansion centred at x, and collecting the homologous terms, we get

$$\begin{aligned} \sum_{k=1}^{N_\delta} a_k u(x+\delta_k,t_n) = &\sum_{k=1}^{N_\delta} a_k\, u(x,t_n) + \sum_{k=1}^{N_\delta} a_k \delta_k\, u_x(x,t_n) + \cdots \\ &+ \frac{1}{l!}\sum_{k=1}^{N_\delta} a_k \delta_k^l\, \frac{\partial^l u}{\partial x^l}(x,t_n) + O\left(\left(\max_k |\delta_k|\right)^{l+1}\right). \end{aligned} \tag{19}$$

Setting now $\delta_k = b_k \delta$, with

$$\delta = (|\nu|\Delta t)^{\frac{1}{l}}, \tag{20}$$

we can rewrite (19) as

$$\begin{aligned} \sum_{k=1}^{N_\delta} a_k u(x+\delta_k,t_n) = &\sum_{k=1}^{N_\delta} a_k\, u(x,t_n) + (\nu\Delta t)^{\frac{1}{l}} \sum_{k=1}^{N_\delta} a_k b_k\, u_x(x,t_n) + \cdots \\ &+ |\nu|\Delta t \frac{1}{l!}\sum_{k=1}^{N_\delta} a_k b_k^l\, \frac{\partial^l u}{\partial x^l}(x,t_n) + O\left(\Delta t^{\frac{l+1}{l}}\right). \end{aligned} \tag{21}$$

Comparing (18) and (21), consistency is recast as the following set of conditions on the coefficients a_k and b_k:

$$\begin{cases} \sum_k a_k = 1 \\ \sum_k a_k b_k = 0 \\ \vdots \\ \sum_k a_k b_k^{l-1} = 0 \\ \sum_k a_k b_k^l = \mathrm{sgn}(\nu)\, l! \end{cases} \tag{22}$$

and, accordingly, the consistency error is

$$O\left(\Delta t^{\frac{l+1}{l}-1}\right) = O\left(\Delta t^{\frac{1}{l}}\right).$$

Remark 1. In order to improve accuracy, one could consider more terms in (21) and enforce more conditions in (22) to obtain a remainder of higher order, so that

$$\begin{cases} \sum_k a_k = 1 \\ \sum_k a_k b_k = 0 \\ \vdots \\ \sum_k a_k b_k^{l-1} = 0 \\ \sum_k a_k b_k^{l} = \mathrm{sgn}(\nu)\, l! \\ \sum_k a_k b_k^{l+1} = 0 \\ \vdots \\ \sum_k a_k b_k^{l+m-1} = 0 \\ \sum_k a_k b_k^{l+m} \neq 0, \end{cases} \tag{23}$$

and in this case the consistency error is

$$O\left(\Delta t^{\frac{l+m}{l}-1}\right) = O\left(\Delta t^{\frac{m}{l}}\right).$$

We will provide an example as such in the following subsection; however, this improvement occurs naturally if l is even and the two sets of parameters a_k and b_k are symmetric, i.e., if to any displacement $\delta_k \neq 0$ there corresponds another displacement $\delta_{k'}$ such that $\delta_{k'} = -\delta_k$ and $a_{k'} = a_k$. This happens, for example, in the treatment of the second derivative in (4): in this specific case, all odd terms in (21) vanish, the first neglected term is zero, and the leading error term in (21) becomes $O(\Delta t^2)$, which implies an $O(\Delta t)$ consistency error.

Passing from the time-discrete scheme (17) to a fully discrete one requires to replace the values $w(x+\delta_k, t_n)$ with an interpolation of the numerical solution at the previous time level. We obtain therefore a scheme in the form

$$\begin{cases} v_j^{n+1} = \displaystyle\sum_{k=1}^{N_\delta} a_k I[V^n](x_j + \delta_k) \\ v_j^0 = u_0(x_j). \end{cases} \tag{24}$$

Since the interpolation is assumed to have an accuracy of order r on smooth functions, following the arguments of §2.1 and of Remark 1, we obtain for (24) a consistency estimate of the form

$$L_{\Delta x, \Delta t}(x_j, t_n) = O\left(\Delta t^{\frac{m}{l}}\right) + O\left(\frac{\Delta x^r}{\Delta t}\right).$$

Remark 2. As it is stated, the consistency condition (18) is unsuitable to enforce rates of consistency higher than first order. Moreover, since $r \geq 2$ and m is expected to be small, in the dispersive case (i.e., for $l > 2$) the leading error term comes typically from time discretization. This suggests that, despite its unconditional stability, the scheme might work at its best for small time steps, and possibly with $\Delta t = o(\Delta x)$. We will provide a sharper evaluation of efficiency in the numerical test section.

3.2. Stability

It is clear that two given set of parameters a_k and b_k satisfying (22) (or (23)) do not necessarily provide a stable approximation. In addition, this system of conditions is not expected to have a unique solution; therefore, both consistency and stability can only be checked *a posteriori* once the two sets of coefficients have been determined. It is therefore crucial to have a procedure to practically build the scheme.

A general technique to find a consistent and stable scheme is to start from an explicit, stable and consistent eulerian scheme, and rescale the stencil of the method. We illustrate the idea with some examples.

Example 1. We start by constructing the approximation for the dispersive operator of the KdV equation. We consider therefore the so-called *Airy equation*

$$u_t = \nu u_{xxx}, \tag{25}$$

with $\nu < 0$. Using the right-biased third incremental ratio (with a generic increment Δ) to approximate u_{xxx}, the forward Euler method for time discretization, and working in the time-discrete setting of (17), we obtain the scheme

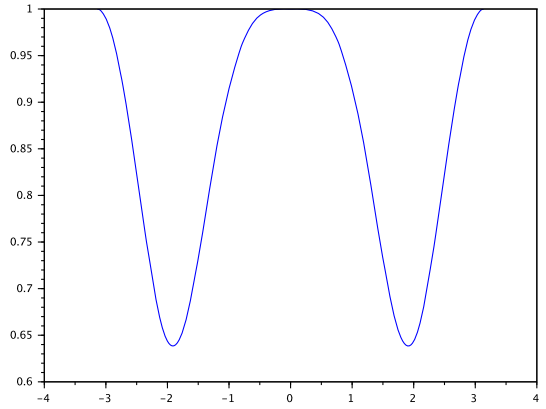

Fig. 1. Modulus of the transfer function of the scheme (26) for $\Delta = 1$, $\lambda = -0.25$ and $\omega \in [-\pi, \pi]$.

$$\begin{aligned} w(x,t_{n+1}) &= w(x,t_n) + \frac{\nu\Delta t}{\Delta^3}\left(-w(x-\Delta,t_n) + 3w(x,t_n) - 3w(x+\Delta,t_n) + w(x+2\Delta,t_n)\right) \\ &= -\lambda w(x-\Delta,t_n) + (1+3\lambda)w(x,t_n) - 3\lambda w(x+\Delta,t_n) + \lambda w(x+2\Delta,t_n) \end{aligned} \tag{26}$$

where

$$\lambda = \frac{\nu\Delta t}{\Delta^3} < 0. \tag{27}$$

Note that, on a standard uniform grid and setting the increment to $\Delta = \Delta x$, this corresponds to the scheme

$$v_j^{n+1} = -\lambda v_{j-1}^n + (1+3\lambda)v_j^n - 3\lambda v_{j+1}^n + \lambda v_{j+2}^n. \tag{28}$$

The scheme (26) is stable when taking $\lambda \in [-1/4, 0)$. We plot in Fig. 1 the modulus of the transfer function of the scheme,

$$H(\omega) = -\lambda e^{i\omega\Delta} + (1+3\lambda) - 3\lambda e^{-i\omega\Delta} + \lambda e^{-i\omega 2\Delta},$$

with $\Delta = 1$, $\lambda = -1/4$ and $\omega \in [-\pi, \pi]$, showing that the stability condition $|H(\omega)| \le 1$ is satisfied.

With this choice for λ, and using (27) and (20), we obtain

$$\Delta^3 = -4\nu\Delta t = 4\delta^3,$$

so that $\Delta = \sqrt[3]{4}\,\delta$ and (26) is rewritten as

$$w(x,t_{n+1}) = \frac{1}{4}w(x - \sqrt[3]{4}\,\delta, t_n) + \frac{1}{4}w(x,t_n) + \frac{3}{4}w(x + \sqrt[3]{4}\,\delta, t_n) - \frac{1}{4}w(x + 2\sqrt[3]{4}\,\delta, t_n).$$

The resulting time-discrete scheme is in the form (17), once we set

$$\begin{cases} a_1 = 1/4, & b_1 = -\sqrt[3]{4} \\ a_2 = 1/4, & b_2 = 0 \\ a_3 = 3/4, & b_3 = \sqrt[3]{4} \\ a_4 = -1/4, & b_4 = 2\sqrt[3]{4}. \end{cases} \tag{29}$$

It is easy to check that the coefficients (29) satisfy conditions (22)–(23) for $l = 3$ and $m = 1$, so that the final scheme is of order 1/3. Note that the smaller the magnitude of λ, the more dispersed the stencil is. To keep the stencil as compact as possible, in this work we always use the maximum magnitude of λ ensuring stability.

Finally, setting $x = x_j$ and introducing the space interpolation I, the fully discrete scheme for (25) reads

$$v_j^{n+1} = \frac{1}{4}I[V^n](x_j - \sqrt[3]{4}\,\delta) + \frac{1}{4}v_j^n + \frac{3}{4}I[V^n](x_j + \sqrt[3]{4}\,\delta) - \frac{1}{4}I[V^n](x_j + 2\sqrt[3]{4}\,\delta).$$

Example 2. In order to improve the order of the scheme obtained above, we start now from a more accurate eulerian scheme. In particular, while the third incremental ratio is a first-order approximation of the third derivative, we can obtain a better approximation by deriving an interpolating polynomial of fourth degree with right-biased stencil, which results in a second order approximation of u_{xxx}, and leads to a stable scheme up to $|\lambda| = 1/8$. Using the same arguments as above, we obtain the set of coefficients

$$\begin{cases} a_1 = 3/16, \quad b_1 = -2 \\ a_2 = 3/8, \quad b_2 = 0 \\ a_3 = 3/4, \quad b_3 = 2 \\ a_4 = -3/8, \quad b_4 = 4 \\ a_5 = 1/16, \quad b_5 = 6. \end{cases} \tag{30}$$

In this case, $m = 2$, the final scheme reads

$$v_j^{n+1} = \frac{3}{16} I[V^n](x_j - 2\delta) + \frac{3}{8} v_j^n + \frac{3}{4} I[V^n](x_j + 2\delta) - \frac{3}{8} I[V^n](x_j + 4\delta) + \frac{1}{16} I[V^n](x_j + 6\delta),$$

and its consistency rate is $2/3$.

Example 3. In this last example, we construct an approximation for the fourth-order equation

$$u_t = \nu u_{xxxx},$$

again with $\nu < 0$. Approximating the fourth derivative by a centred fourth incremental ratio provides a stable scheme up to $|\lambda| = 1/8$. We obtain the set of coefficients

$$\begin{cases} a_1 = -1/8, \quad b_1 = -2\sqrt[4]{8} \\ a_2 = 1/2, \quad b_2 = -\sqrt[4]{8} \\ a_3 = 1/4, \quad b_3 = 0 \\ a_4 = 1/2, \quad b_4 = \sqrt[4]{8} \\ a_5 = -1/8, \quad b_5 = 2\sqrt[4]{8}. \end{cases}$$

This is a symmetric case, so that $m = 2$ and the consistency rate is $1/2$. The final scheme is

$$v_j^{n+1} = -\frac{1}{8} I[V^n](x_j - 2\sqrt[4]{8}\delta) + \frac{1}{2} I[V^n](x_j - \sqrt[4]{8}\delta) + \frac{1}{4} v_j^n + \frac{1}{2} I[V^n](x_j + \sqrt[4]{8}\delta) - \frac{1}{8} I[V^n](x_j + 2\sqrt[4]{8}\delta).$$

3.3. Resolution of the smaller scales

As it has been noted elsewhere (see [9]), this technique may lead, by enlarging the stencil, to a relatively uneven numerical domain of dependence. In practice, there exist "holes" between the various points $x_j + \delta_k$, which may cause the smaller scales to be underresolved, unless some suitable relationship between Δt and Δx is enforced. Generalizing the analysis carried out in [9], we note that around each of the points $x_j + \delta_k$ the numerical domain of dependence has a radius of size $O(\Delta x)$. Since the hole in between is of size $O\left((\nu \Delta t)^{1/l}\right)$, it is filled in a number of steps

$$K = O\left(\frac{(\nu \Delta t)^{1/l}}{\Delta x}\right).$$

Imposing now that at the time of interest T the holes should be completely covered by the numerical domain of dependence, we obtain that $T > K \Delta t$, so that $\Delta t < T/K$, that is, recasting this condition in asymptotic terms,

$$\Delta t = o\left(\frac{T \Delta x}{(\nu \Delta t)^{1/l}}\right).$$

Solving this relationship w.r.t. Δt, we finally obtain

$$\Delta t = o\left(\frac{(T \Delta x)^{\frac{l}{l+1}}}{\nu^{\frac{1}{l+1}}}\right),$$

which shows in particular that keeping a linear $\Delta t / \Delta x$ relationship is a viable refinement strategy.

3.4. The nonlinear case

Finally, applying the technique outlined above to generalize the scheme (8) to the general case of equation (6), we obtain

$$\begin{cases} v_j^{n+1} = \sum_k a_k I[V^n](x_j - f'(v_j^{n+1})\Delta t + \delta_k) \\ v_j^0 = u_0(x_j). \end{cases} \tag{31}$$

For example, in the case of the KdV equation,

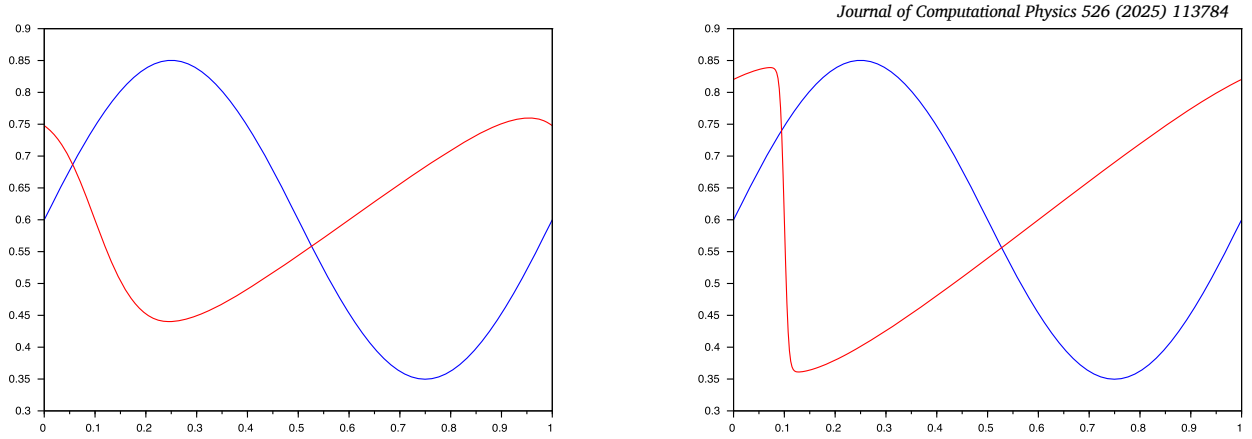

Fig. 2. Initial condition (in blue) and solutions with 400 nodes (in red) at $t = 1$ for problem (35). Viscosity $\nu = 10^{-2}$ (left) and $\nu = 10^{-3}$ (right). (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x = \nu u_{xxx} \\ u(x,0) = u_0(x) \end{cases} \tag{32}$$

(with $\nu < 0$), blending everything together we obtain the scheme

$$\begin{aligned} v_j^{n+1} = {} & \frac{1}{4} I[V^n](x_j - \Delta t v_j^{n+1} - \sqrt[3]{4}\,\delta) + \frac{1}{4} I[V^n](x_j - \Delta t v_j^{n+1}) \\ & + \frac{3}{4} I[V^n](x_j - \Delta t v_j^{n+1} + \sqrt[3]{4}\,\delta) - \frac{1}{4} I[V^n](x_j - \Delta t v_j^{n+1} + 2\sqrt[3]{4}\,\delta), \end{aligned} \tag{33}$$

in the case of a four-point stencil, and the scheme

$$\begin{aligned} v_j^{n+1} = {} & \frac{3}{16} I[V^n](x_j - \Delta t v_j^{n+1} - 2\delta) + \frac{3}{8} I[V^n](x_j - \Delta t v_j^{n+1}) + \frac{3}{4} I[V^n](x_j - \Delta t v_j^{n+1} + 2\delta) \\ & - \frac{3}{8} I[V^n](x_j - \Delta t v_j^{n+1} + 4\delta) + \frac{1}{16} I[V^n](x_j - \Delta t v_j^{n+1} + 6\delta), \end{aligned} \tag{34}$$

for the five-point case. In what follows, we will possibly refer to these time discretizations as respectively SL-4p and SL-5p.

4. Numerical examples

We provide in this section a numerical validation of the technique proposed. All the schemes have been implemented in the Matlab-like environment Scilab. In a first set of numerical tests, we treat the viscous case, focusing in particular on the Burgers' and Lighthill–Whitham–Richards (LWR) equations, in which we take a viscosity ν large enough to ensure that the solution remains smooth and the diffusion is completely resolved; in the second part, we focus on the KdV equation. Since one major source of errors in nonconservative schemes comes from the possibly incorrect speed of propagation of discontinuities, we will use the conservation error as a rough measure of accuracy, whenever an explicit solution is not available. Except for Example 3, in all the examples we will use periodic boundary conditions.

Example 1. The first example is the one-dimensional Burgers' equation

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x = \nu u_{xx} & (x,t) \in (0,1) \times (0,1), \\ u(x,0) = 0.6 + 0.25 \sin(2\pi x). \end{cases} \tag{35}$$

In (35), we use the two different values $\nu = 10^{-2}$ and $\nu = 10^{-3}$ to show the influence of the viscosity on the conservation error. The scheme (8) has been implemented with $\mathbb{P}_1$ and spline space interpolation, space grids of N nodes, with N ranging from 100 to 1600 and time step $\Delta t = 5\Delta x$, resulting in Courant numbers up to about 4; Fig. 2 shows the initial condition along with the numerical solution obtained with 400 nodes and spline interpolation, for the two different values of the viscosity. Conservation errors are reported in Table 1, which shows monotonically decreasing errors along the refinement, and larger errors at the smaller viscosity.

Example 2. The second example is the viscous LWR equation

$$\begin{cases} u_t + \left(\frac{u-u^2}{2}\right)_x = \nu u_{xx} & (x,t) \in (0,1) \times (0,1), \\ u(x,0) = 0.6 + 0.25 \sin(2\pi x). \end{cases} \tag{36}$$

Table 1
Relative conservation errors for problem (35).

N	$\nu = 10^{-2}$		$\nu = 10^{-3}$	
	$\mathbb{P}_1$	spline	$\mathbb{P}_1$	spline
100	$1.51 \cdot 10^{-11}$	$4.46 \cdot 10^{-9}$	$2.33 \cdot 10^{-5}$	$3.48 \cdot 10^{-5}$
200	$2.52 \cdot 10^{-14}$	$1.32 \cdot 10^{-9}$	$1.31 \cdot 10^{-6}$	$3.62 \cdot 10^{-6}$
400	$1.66 \cdot 10^{-15}$	$7.68 \cdot 10^{-11}$	$2.35 \cdot 10^{-8}$	$6.95 \cdot 10^{-8}$
800	$3.70 \cdot 10^{-16}$	$3.94 \cdot 10^{-12}$	$8.93 \cdot 10^{-11}$	$2.99 \cdot 10^{-10}$
1600	$1.11 \cdot 10^{-15}$	$3.88 \cdot 10^{-13}$	$1.60 \cdot 10^{-13}$	$1.82 \cdot 10^{-10}$

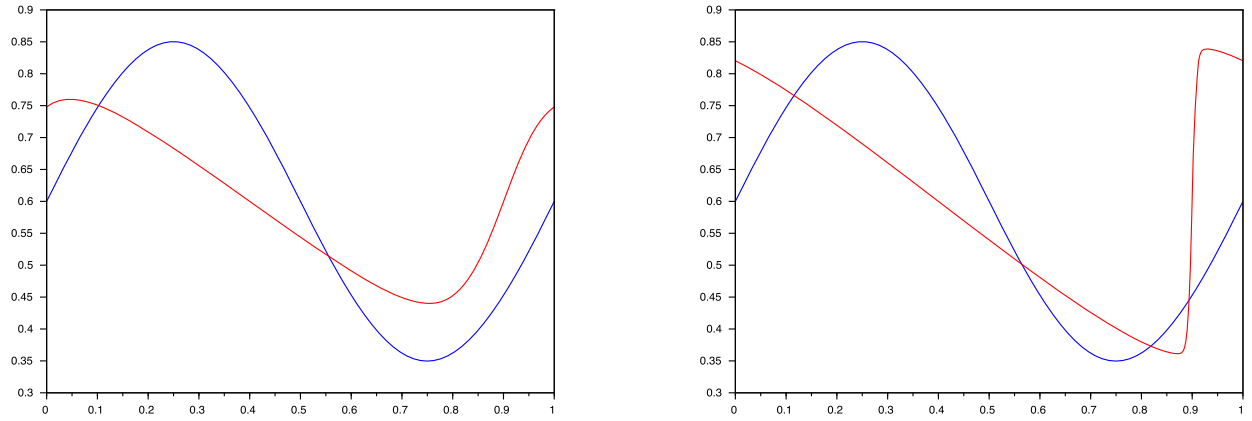

Fig. 3. Initial condition (in blue) and solutions with 400 nodes (in red) at $t = 1$ for problem (36). Viscosity $\nu = 10^{-2}$ (left) and $\nu = 10^{-3}$ (right).

Table 2
Errors in the 1-norm and convergence rates for problem (37).

N	$\mathbb{P}_1$	spline
100	$7.83 \cdot 10^{-2}$	$3.43 \cdot 10^{-2}$
200	$3.88 \cdot 10^{-2}$ (1.01)	$3.15 \cdot 10^{-3}$ (3.44)
400	$1.07 \cdot 10^{-2}$ (3.63)	$1.33 \cdot 10^{-3}$ (1.24)
800	$5.25 \cdot 10^{-3}$ (1.03)	$6.61 \cdot 10^{-4}$ (1.01)
1600	$2.7 \cdot 10^{-3}$ (0.96)	$3.31 \cdot 10^{-4}$ (0.99)

We still use in (36) the two different values $\nu = 10^{-2}$ and $\nu = 10^{-3}$, and implement the scheme (8) with the same procedure and parameters of the previous example. Conservation errors are also similar and will not be reported, while Fig. 3 shows the initial condition and the numerical solutions obtained in the two cases with 400 nodes and spline interpolation.

Example 3. To assess the accuracy of the scheme, we consider again Burgers' equation, but on the interval $[-1, 3]$ and with the initial condition

$$u_0(x) = \frac{1}{2} + \frac{1}{2} \tanh \frac{x}{4\nu}, \tag{37}$$

which corresponds to a travelling wave solution moving at speed $1/2$, so that

$$u(x,t) = u_0(x - t/2).$$

We use $\nu = 10^{-2}$, and implement the scheme (8) with $\Delta t = 5\Delta x$, giving a Courant number of 2.5. Fig. 4 shows the initial condition, the exact solution at $t = 4$ and the numerical approximation obtained with 400 nodes and spline interpolation. Due to the sharp transition, errors have been computed in the 1-norm

$$\Delta x \sum_j \left| v_j^n - u(x_j, t_n) \right|.$$

Table 2 shows the errors for both the $\mathbb{P}_1$ and the spline implementation, along with the convergence rates, which confirm the theoretical analysis. Note that, while in the case of $\mathbb{P}_1$ interpolation both terms of the consistency error (10) are first-order, in the case of splines the dominant term comes from time discretization. Although converge remains first-order, the increase of accuracy in the space discretization term causes a relevant reduction of the error (about one order of magnitude).

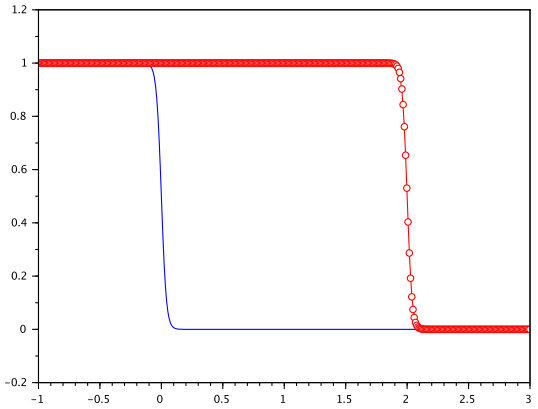

Fig. 4. Initial condition (in blue), exact and numerical solution (in red) at $t = 4$ for problem (37). Viscosity $\nu = 10^{-2}$, 400 nodes and spline interpolation.

Table 3
Relative conservation errors for problem (14), with viscosity $\nu = 10^{-2}$ and $\nu = 10^{-3}$.

N	$\nu = 10^{-2}$	$\nu = 10^{-3}$
50	$8.80 \cdot 10^{-6}$	$2.55 \cdot 10^{-2}$
100	$2.06 \cdot 10^{-5}$	$7.32 \cdot 10^{-3}$
200	$9.50 \cdot 10^{-6}$	$2.24 \cdot 10^{-3}$

Example 4. We consider here the viscous two-dimensional Burgers' equation (14), posed on $(0,1) \times (0,1)$ with doubly periodic boundary conditions, and initial conditions given by

$$u_0(x_1, x_2) = \frac{1}{4}(1 + \sin(2\pi x_1)\sin(2\pi x_2)),$$
$$v_0(x_1, x_2) = \frac{1}{2}(1 + \sin(2\pi x_1)\sin(2\pi x_2)),$$

for which the relationship $v(t) = 2u(t)$ holds for all $t > 0$.

We compute the approximate solution on three $N \times N$ grids, with N ranging from 50 to 200, with $\Delta t = 5/N$, that is, at a Courant number of about 5, up to $T = 1.5$. The initial condition and the approximate solution at $t = T$ obtained with $N = 200$ are shown in Fig. 5. Conservation errors are reported in Table 3 for the three grids considered. They are higher at the lower viscosity, but show a clear convergence to zero, with a convergence rate slightly higher than the unity.

Example 5. In this test, we approximate the KdV equation

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x = \nu u_{xxx} & (x,t) \in (0,1) \times (0,2), \\ u(x,0) = 0.6 + 0.25\sin(2\pi x), \end{cases} \tag{38}$$

at the time $T = 2$ on $[0,1]$ with $\nu = -10^{-4}$. In the inviscid Burgers' equation, a shock occurs at an intermediate time; in the dispersive case the formation of a steep transition generates a train of oscillations on the left side of this transition. We show in Fig. 6 the numerical solution obtained with $\Delta t = \Delta x = 1/400$, SL-5p scheme and a cubic spline interpolation, while Table 4 shows the conservation errors. The relatively low accuracy of $\mathbb{P}_1$ interpolation causes relevant errors, although the quasi-conservative behaviour of the scheme is confirmed. On the other hand, the conservation error benefits from a higher order space interpolation, as it can be seen by comparing the two columns.

Example 5. This last test analyses in detail a case of soliton propagation for the KdV equation. It is well-known that this equation admits closed-form solutions in the form of travelling waves. In particular, setting $\nu = -1$, $f(u) = 3u^2$ and

$$u_0(x) = \frac{1}{2\cosh^2 \frac{x - x_0}{2}},$$

we obtain a travelling wave consisting of a peak initially centred at x_0 and moving rightwards at unit speed, that is, a solution of the form

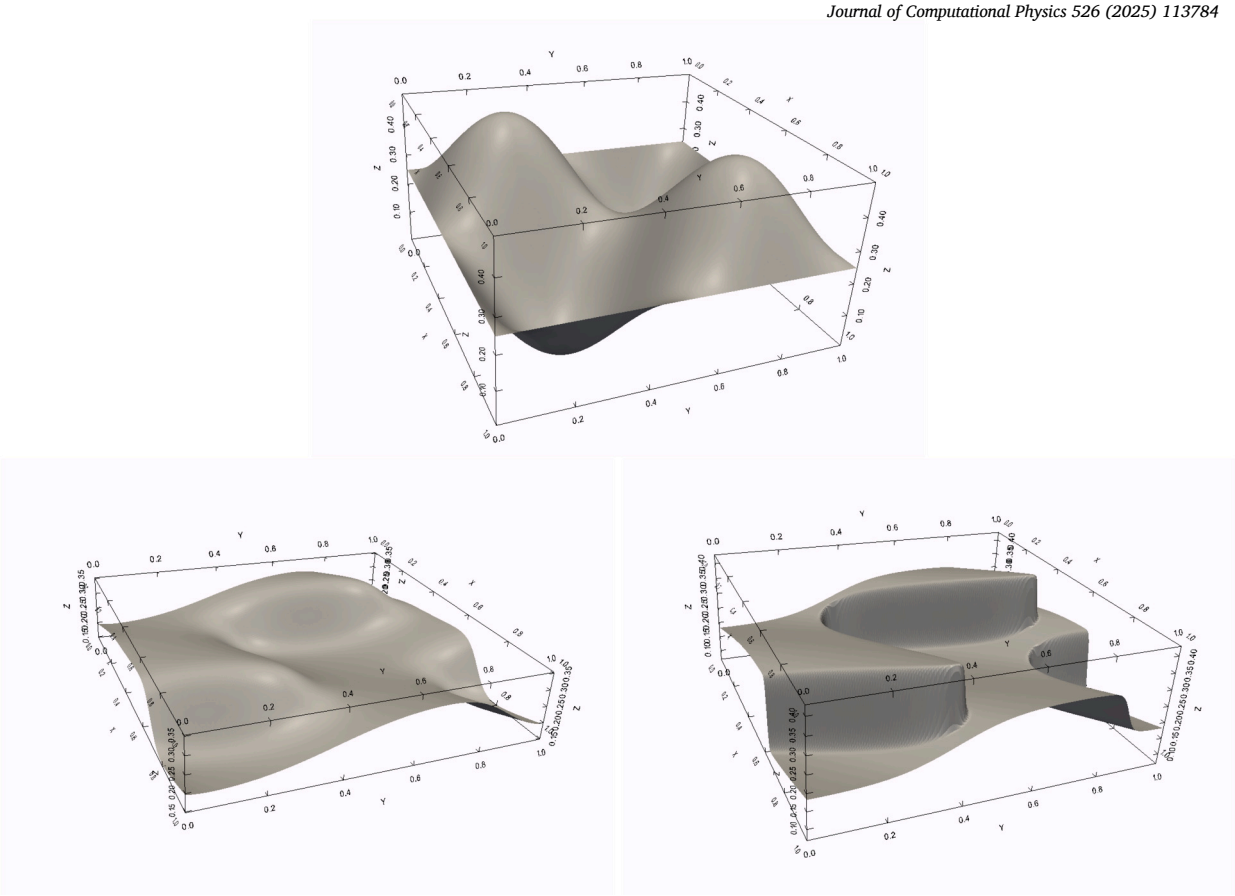

Fig. 5. Initial condition for problem (14) (upper) and numerical solutions for u at $t = 1.5$, with viscosity $\nu = 10^{-2}$ (lower left) and $\nu = 10^{-3}$ (lower right).

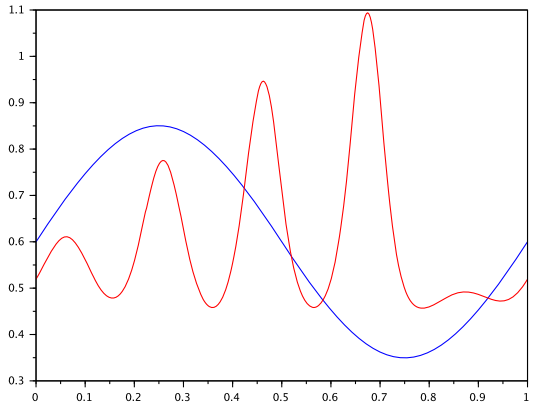

Fig. 6. Formation of a train of oscillations past a shock in the KdV equation, initial condition (in blue) and numerical solution at $T = 2$ (in red). Scheme SL-5p, 400 nodes, spline interpolation.

$$u(x,t) = u_0(x - x_0 - t) = \frac{1}{2\cosh^2 \frac{x - x_0 - t}{2}}.$$

Convergence rate We show in Fig. 7 the numerical solution at $T = 10$ for $x_0 = 15$ on the interval $[0, 40]$, with 200 nodes and time step $\Delta t = 5 \cdot 10^{-3}$. The solutions have been computed with respectively the four-point scheme (33) in the left column and the five-point scheme (34) in the right column; $\mathbb{P}_1$ interpolation has been used on the upper line and spline interpolation on the lower line. Table 5 reports the numerical error and convergence rates (in brackets) for the various schemes, along a linear refinement with $\Delta t = 1/N$, and the left plot of Fig. 8 shows the errors on a log-log scale, as compared with the orders 1/3 and 2/3 (solid lines in black). Note that, under a linear refinement, the leading error term comes from time discretization; nevertheless, higher order (in

Table 4
Relative conservation errors for problem (38), with $\nu = -10^{-4}$, scheme SL-5p.

N	$\mathbb{P}_1$	spline
100	$4.22 \cdot 10^{-2}$	$1.49 \cdot 10^{-4}$
200	$2.43 \cdot 10^{-2}$	$4.06 \cdot 10^{-5}$
400	$1.62 \cdot 10^{-2}$	$4.40 \cdot 10^{-6}$
800	$8.77 \cdot 10^{-3}$	$5.44 \cdot 10^{-7}$
1600	$1.11 \cdot 10^{-3}$	$3.61 \cdot 10^{-8}$

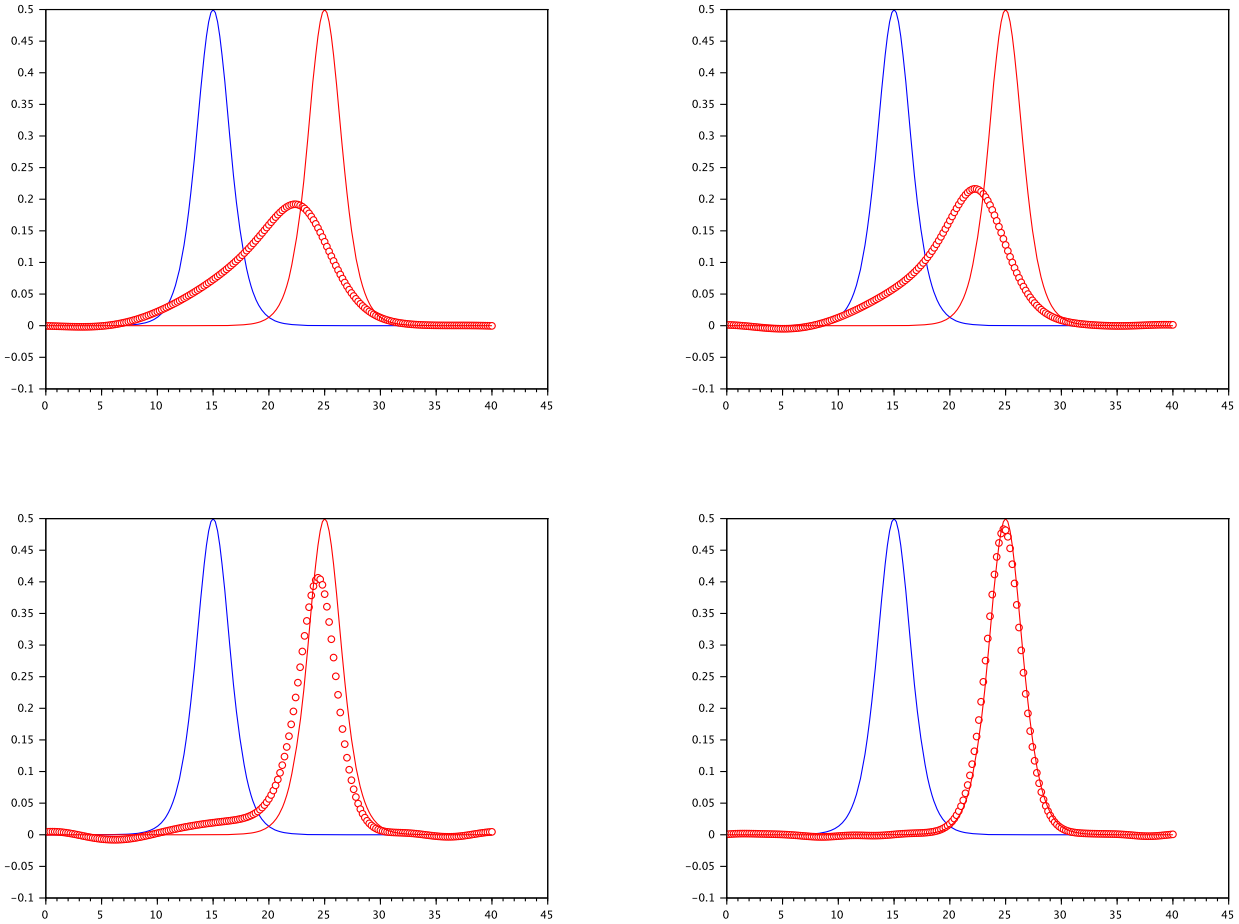

Fig. 7. Propagation of a soliton in the KdV equation: initial condition (in blue), exact and numerical solution at $T = 10$ (in red). Numerical solutions with 200 nodes and $\Delta t = 5 \cdot 10^{-3}$. Four-point scheme (left plots) and five-point scheme (right plots), linear interpolation (upper plots) and spline interpolation (lower plots).

this case, spline) interpolation reduces both the undue dissipation introduced by linear interpolation, and the decrease in the group velocity of the soliton – this results again in a relevant improvement of the accuracy. For example, although both versions of the five-point scheme converge with asymptotic rate of about 2/3, as predicted by the theoretical study, the error associated to the spline version is one and a half orders of magnitude lower.

Efficiency In the right plot of Fig. 8 we compare the efficiency of the SL-5p/spline scheme with other more conventional options, by plotting the error as a function of CPU time, for N from 100 to 1600. The schemes compared are:

- SL-5p/spline method (34), with linear refinement $\Delta t = 1/N$;
- Fully explicit method with the dispersion term treated as in (28), i.e.,

$$v_j^{n+1} = v_j^n - \Delta t f'(v_j^n) \frac{v_{j+1}^n - v_{j-1}^n}{2\Delta x} - \lambda v_{j-1}^n + (1 + 3\lambda) v_j^n - 3\lambda v_{j+1}^n + \lambda v_{j+2}^n,$$

with Δt corresponding to the maximal stability parameter $\lambda = -1/4$;

Table 5
Errors in the ∞-norm and convergence rates for Example 5.

N	SL-4p		SL-5p	
	$\mathbb{P}_1$	spline	$\mathbb{P}_1$	spline
100	0.39	0.17	0.33	$4.7 \cdot 10^{-2}$
200	0.37 (0.08)	0.15 (0.18)	0.38 (-0.2)	$3.2 \cdot 10^{-2}$ (0.55)
400	0.27 (0.45)	0.13 (0.21)	0.32 (0.25)	$2.0 \cdot 10^{-2}$ (0.68)
800	0.28 (-0.05)	0.11 (0.24)	0.21 (0.61)	$1.3 \cdot 10^{-2}$ (0.62)
1600	0.22 (0.35)	$9.3 \cdot 10^{-2}$ (0.24)	0.13 (0.69)	$8.1 \cdot 10^{-3}$ (0.68)

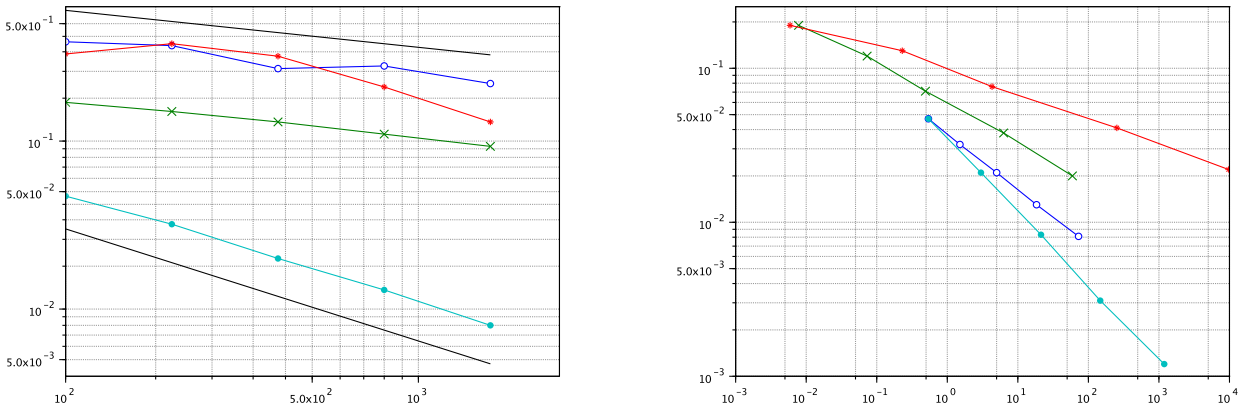

Fig. 8. Left: numerical errors as a function of the number of nodes, schemes SL-4p/$\mathbb{P}_1$ (blue circles), SL-4p/spline (green crosses), SL-5p/$\mathbb{P}_1$ (red asterisks) and SL-4p/spline (cyan dots); orders 1/3 and 2/3 (solid black lines). Right: errors as a function of the CPU time, for fully explicit scheme (red asterisks), linearly implicit scheme (green crosses), SL-5p/spline scheme with linear $\Delta x/\Delta t$ relationship (blue circles) and with $\Delta t \sim \Delta x^2$ (cyan dots).

- Linearly implicit version of the previous method, in the form

$$v_j^{n+1} = v_j^n - \Delta t f'(v_j^n) \frac{v_{j+1}^n - v_{j-1}^n}{2\Delta x} - \lambda v_{j-1}^{n+1} + (1+3\lambda) v_j^{n+1} - 3\lambda v_{j+1}^{n+1} + \lambda v_{j+2}^{n+1},$$

  again with a linear refinement $\Delta t = 1/N$;
- SL-5p/spline method (34), with a nonlinear $\Delta t/\Delta x$ refinement corresponding to a quasi-optimal order. In fact, since the scheme has a local truncation error of the form

$$O\left(\Delta t^{2/3}\right) + O\left(\frac{\Delta x^4}{\Delta t}\right),$$

  the optimal consistency is achieved by balancing the two terms, i.e., for $\Delta t = \Delta x^{12/5}$. In this case, we start from the initial grid ($N = 100$, $\Delta t = 0.01$) and use the simpler relationship $\Delta t = c\Delta x^2$, which provides a theoretical consistency rate of $\Delta x^{4/3}$ (the optimal rate is $\Delta x^{8/5}$).

While it is clear that such a comparison has a strong dependence on the implementation of the various schemes, in order to ensure a fair assessment each scheme has been optimized as much as possible for the specific platform Scilab.

The comparison shows that, at least among the cases under consideration, the best efficiency is achieved by the SL-5p/spline scheme with nonlinear refinement. In this case, the advantage of using a fully-SL strategy is related to the low complexity of time-stepping, rather than to the possibility of using large Courant numbers.

5. Conclusions

We have proposed and validated a fully semi-Lagrangian technique to treat conservation laws with viscous or dispersive terms. The technique uses the advective form of the conservation law, and may work under hyperbolic-type $\Delta t/\Delta x$ relationships. The scheme is implicit, but the system of equations to be solved at each step is decoupled, and therefore easily scalable on parallel architectures; in addition, it has moderate memory requirements. Although the consistency rate is relatively low, this technique proves to be robust enough. In particular, we find empirically that the scheme is quasi-conservative, and, for the viscous case, prove an entropic behaviour of the numerical solution.

CRediT authorship contribution statement

R. Ferretti: Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Roberto Ferretti reports financial support was provided by Francesco Severi National Institute of Higher Mathematics National Group of Scientific Calculations. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The Author thanks an anonymous reviewer for his/her constructive and useful comments. This work has been partially supported by INdAM – Gruppo Nazionale di Calcolo Scientifico.

Data availability

Data will be made available on request.

References

[1] L. Bonaventura, E. Calzola, E. Carlini, R. Ferretti, A fully semi-Lagrangian method for the Navier–Stokes equations in primitive variables, in: H. van Brummelen, A. Corsini, S. Perotto, G. Rozza (Eds.), Numerical Methods for Flows, in: Lecture Notes in Computational Science and Engineering, vol. 132, Springer, Cham, 2020.
[2] L. Bonaventura, E. Calzola, E. Carlini, R. Ferretti, Second order fully semi-Lagrangian discretizations of advection–diffusion–reaction systems, J. Sci. Comput. 88 (2021) 23.
[3] L. Bonaventura, R. Ferretti, Semi-Lagrangian methods for parabolic problems in divergence form, SIAM J. Sci. Comput. 36 (2014) A2458–A2477.
[4] F. Camilli, M. Falcone, An approximation scheme for the optimal control of diffusion processes, ESAIM: M2AN 29 (1995) 97–122.
[5] R. Courant, E. Isaacson, M. Rees, On the solution of nonlinear hyperbolic differential equations by finite differences, Commun. Pure Appl. Math. 5 (1952) 243–255.
[6] C.Z. Cheng, G. Knorr, The integration of the Vlasov equation in configuration space, J. Comput. Phys. 22 (1976) 330–351.
[7] M.G. Crandall, L. Tartar, Some relations between nonexpansive and order preserving mappings, Proc. Am. Math. Soc. 78 (1980) 385–390.
[8] M. Falcone, R. Ferretti, Semi-Lagrangian Approximation Schemes for Linear and Hamilton–Jacobi Equations, SIAM, Philadelphia, 2013.
[9] R. Ferretti, A technique for the high-order treatment of diffusion terms in semi-Lagrangian schemes, Commun. Comput. Phys. 8 (2010) 445–470.
[10] H. Kushner, P. Dupuis, Numerical Methods for Stochastic Control Problems in Continuous Time, Springer, New York, 2001.
[11] G.N. Milstein, M.V. Tretyakov, Numerical solution of the Dirichlet problem for nonlinear parabolic equations by a probabilistic approach, IMA J. Numer. Anal. 21 (2001) 887–917.
[12] G.N. Milstein, M.V. Tretyakov, Stochastic Numerics for Mathematical Physics, Springer, Berlin, 2004.
[13] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, The semi-Lagrangian method for the numerical resolution of the Vlasov equation, J. Comput. Phys. 149 (1999) 201–220.
[14] A. Staniforth, J. Coté, Semi-Lagrangian integration schemes for atmospheric models – a review, Mon. Weather Rev. 119 (1991) 2206–2223.
[15] E. Tadmor, Local error estimates for discontinuous solutions of nonlinear hyperbolic equations, SIAM J. Numer. Anal. 28 (1991) 891–906.