Explicit and Implicit TVD Schemes for Conservation Laws with Caputo Derivatives

Jian-Guo Liu[1] · **Zheng Ma**[2] · **Zhennan Zhou**[3]

Abstract In this paper, we investigate numerical approximations of the scalar conservation law with the Caputo derivative, which introduces the memory effect. We construct the first order and the second order explicit upwind schemes for such equations, which are shown to be conditionally ℓ^1 contracting and TVD. However, the Caputo derivative leads to the modified CFL-type stability condition, $(\Delta t)^\alpha = O(\Delta x)$, where $\alpha \in (0, 1]$ is the fractional exponent in the derivative. When α is small, such strong constraint makes the numerical implementation extremely impractical. We have then proposed the implicit upwind scheme to overcome this issue, which is proved to be unconditionally ℓ^1 contracting and TVD. Various numerical tests are presented to validate the properties of the methods and provide more numerical evidence in interpreting the memory effect in conservation laws.

Keywords TVD schemes · Conservation law · Caputo derivative · Memory effect

1 Introduction

In this paper, we study numerical approximations to the scalar conservation law with Caputo derivatives. The governing equation is the following conservation law with fractional time derivative:

✉ Zhennan Zhou
zhennan@math.duke.edu

Jian-Guo Liu
jliu@phy.duke.edu

Zheng Ma
mayuyu@sjtu.edu.cn

1   Department of Mathematics and Department of Physics, Duke University, Box 90320, Durham, NC 27708, USA

2   Department of Mathematics, Shanghai Jiao Tong University, 800 Dongchuan RD, Shanghai 200240, China

3   Department of Mathematics, Duke University, Box 90320, Durham, NC 27708, USA

$$\partial_t^\alpha u(x, t) + f(u(x, t))_x = 0, \quad x \in \mathbb{R}, \ t > 0, \tag{1}$$

where $u(x, t)$ is a certain density function or concentration, and $f(u)$ is the flux function. This equation is completed by an initial condition:

$$u(x, 0) = u_0(x), \quad x \in \mathbb{R}. \tag{2}$$

Here, when fractional exponent $\alpha \in (0, 1)$, ∂_t^α denotes the Caputo derivative:

$$\partial_t^\alpha u(t) = C_\alpha \int_0^t (t - s)^{-\alpha} \partial_s u(s) ds,$$

where $C_\alpha = 1/\Gamma(1 - \alpha)$. When $\alpha = 1$, $\partial_t^\alpha u(t)$ is the standard time derivative $\partial_t u(t)$.

The Caputo derivative was introduced in [3] for diffusion of fluids in porous media with the memory effect. It has been shown that the Caputo derivative is effective in modeling plasma transport (see [4,5]), and fractional functions have been made use of to construct physics models for long–range or nonlocal interactions in space and time, see for e.g., [13,19]. Recently in [1], Allen, Caffarelli and Vasseur proved the existence of the weak solutions and the Hölder continuity for such solutions when with both a fractional potential pressure and fractional time derivative. The scalar conservation law with Caputo derivatives (1) models the time evolution of a distribution of interest with general flux function and memory effect. However, while the physical interpretation is clear, not much mathematical research has been done for such equations.

In the aspect of numerical approximation, many numerical methods have been introduced and analyzed for ODE's or diffusion equations with the Caputo derivative, see e.g., [2,7,10,22], and most of which can be extended to fractional space-time advection–diffusion equations (see [11]) with additional treatment in the space (fractional) derivatives. Also, there have been a few worthy studies in numerical approximations of fractional space-time advection-dispersion equation, see [21]. It has been shown in different equations that, a consistent discretization of the Caputo derivative will introduce dissipation in time, which can help stabilize a numerical scheme. Therefore, at least for linear equations, the Caputo derivative does not cause extra challenges in numerical approximations. In [22], Zhao, Sun and Karniadakis proposed a numerical method to approximate viscous Burgers' equation, however the methods suffered from the Gibbs phenomenon at jumps since the Fourier collocation method was used for the space discretization. To our best knowledge, this was still the only successful attempt for nonlinear conservation laws, although the presence of the diffusion terms helped avoid the issue of weak solution.

In this paper, we aim to construct and analyze explicit and implicit upwind schemes for the scalar conservation law with the Caputo derivative (1). We propose the first order and the second upwind schemes to Eq. (1), and show that with modified CFL conditions, the numerical schemes are TVD. However, the modified CFL conditions are becoming more restricted as $\alpha \to 0$, which makes the explicit schemes not feasible for small α. Motivated by this, we further design an implicit upwind method for the conservation law which is shown to be ℓ^1 contracting and thus TVD. And in particular, for the linear advection case, we also show that the implicit scheme is also energy stable and satisfies the entropy condition.

The rest of the paper is outline as follows. We summarize some preliminary knowledge of the scalar conservation law with the Caputo derivative in the second part of this section. In Sect. 2, we briefly summarize existing results on numerical approximation of the Caputo derivative. We first propose and show stability analysis of the first order and the second order explicit upwind scheme in Sect. 3, which is followed by the introduction of an implicit upwind scheme to avoid the CFL conditions. We carry out various numerical tests in Sect. 4, not only

to verify the properties of the proposed schemes, but also investigate the interpretations of the Caputo derivative by conducting some designed tests.

1.1 Preliminaries

In this part, we start by introducing the Mittag-Leffler function for the time fractional linear advection equation and conclude with discussions on weak solutions to the nonlinear conservation law with the Caputo derivative. We consider the following equation with fractional time derivative

$$\partial_t^\alpha u = -Au, \quad u(x, 0) = g(x). \tag{3}$$

where A is an operator to be specified. By the standard Laplace transformation,

$$\mathcal{L}u(s) := \hat{u}(s) = \int_0^\infty e^{-st} u(t) dt,$$

the Eq. (3) becomes

$$(s^\alpha + A)\hat{u} = s^{\alpha-1} g.$$

Application of the Laplace inversion gives

$$u(t) = E_\alpha(-t^\alpha A)g, \tag{4}$$

where $E_\alpha(z)$ is the Mittag-Leffler function

$$E_\alpha(z) = \sum_{n=0}^\infty \frac{z^n}{\Gamma(\alpha n + 1)}. \tag{5}$$

If we set $A = -a\partial_x$, where a is a constant, the corresponding advection equation reads,

$$\partial_t^\alpha u + a\partial_x u = 0, \tag{6}$$

and its solution can be represented as

$$u(x, t) = \sum_{n=0}^\infty \frac{\partial_x^n g}{\Gamma(\alpha n + 1)} a^n t^{\alpha n}. \tag{7}$$

Notice that, when $\alpha = 1$, the equation reduces to the normal scalar conservation law, and from the above equation we get nothing but

$$u(x, t) = g(x + at) = \sum_{n=0}^\infty \frac{\partial_x^n g}{n!} a^n t^n. \tag{8}$$

However, for general conservation laws (1), there is lack of representations, and it is more convenient to work with its weak solutions due to the nonlinear flux. We give the following definition:

Definition 1.1 $u(x, t)$ is a *weak solution* of the Eq. (1), if $\partial_t^\alpha u \in L_{\text{loc}}^1(\mathbb{R})$, $f(u) \in L_{\text{loc}}^1(\mathbb{R})$ and if for every test function $\phi \in \mathcal{D}(\mathbb{R})$, we have

$$\int_\mathbb{R} \left(\partial_t^\alpha u\phi - f(u)\partial_x\phi\right) dx = 0. \tag{9}$$

One can easily verify that the notion of a weak solution extends that of a classical solution: every classical solution of (1) is also a weak solution.

However, like standard conservation laws, the weak solutions to (1) may not be unique. To construct a Cauchy problem which admits more than one weak solution, we choose the fractional Burgers' equation

$$\partial_t^\alpha u + \partial_x \left(\frac{1}{2}u^2\right) = 0, \tag{10}$$

with initial data

$$u(x, 0) = \begin{cases} 1, & x > 0, \\ -1, & x < 0. \end{cases} \tag{11}$$

First we claim that the initial data itself serves as one weak solution, which means we have a static discontinuous solution, (see Fig. 1 left)

$$u(x, t) = \begin{cases} 1, & x > 0, \\ -1, & x < 0. \end{cases} \tag{12}$$

We now verify that this solution satisfies definition of the weak solution. First, we have

$$\partial_t^\alpha u \equiv 0 \quad \text{for } \forall x \neq 0,$$

and we also have

$$f(u) = \frac{1}{2}u^2 \equiv \frac{1}{2} \quad \text{for } \forall x \neq 0.$$

According to **Definition** (1.1), we have for every test function $\phi \in \mathcal{D}(\mathbb{R})$

$$\int_\mathbb{R} \left(\partial_t^\alpha u \phi - f(u)\partial_x \phi\right) dx = 0 - \frac{1}{2}\int_\mathbb{R} \phi_x \, dx = 0.$$

So it's a weak solution of Eq. (10). But obviously, this is not an entropy solution since the characteristics are moving outwards from the discontinuous jump.

To construct another solution, we use the numerical method we will introduce later in this paper, and the numerical result is plotted in Fig. 1 right for $t = 0.02$ and $\alpha = 0.8$. We see that, intuitively, the solution is between the static discontinuous solution and the standard rarefaction solution, which manifests the memory effect.
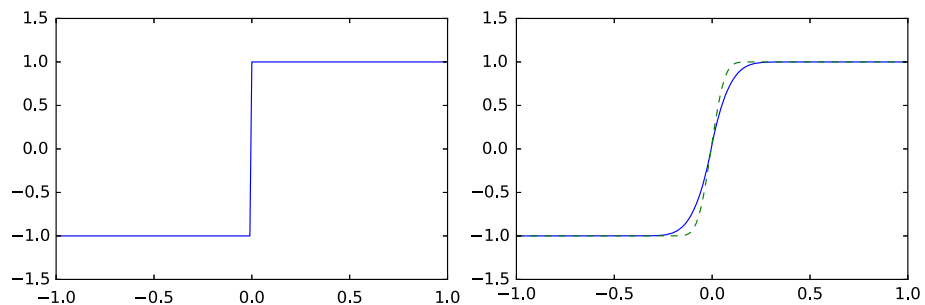

Fig. 1 Non-uniqueness of the solution. *Left*: the static discontinuous solution. *Right*: *solid line* is the rarefaction solution with memory effect; *dashed line* is the rarefaction solution for the Burgers' equation with the standard time derivative

In order to define the entropy solutions, we consider the conservation law with artificial viscosity

$$\partial_t^\alpha u + \partial_x f(u) = \varepsilon \partial_{xx} u. \tag{13}$$

Here, $\varepsilon > 0$ is a diffusion coefficient, $\varepsilon \ll 1$. The Cauchy problem for (13) can be shown to have one and only one classical solution u^ε which satisfies the maximum principle. If the sequence $\{u^\varepsilon\}$ converges almost everywhere to a function u when $\varepsilon \to 0$, we call that u is an entropy solution of (1).

We conclude this section by the following remark. It is entirely possible for one to define the entropy solutions by constructing the convex entropy function and entropy flux pairs and link that to the vanishing viscosity limit, but that will be quite analysis involved and is off the focus of the current paper. We will save the rigorous analysis of the entropy solutions to (1) as one of the possible future directions.

2 Numerical Approximation of the Caputo Derivative

While there has been thorough understanding on numerical approximation of standard ordinary differential equations, the investigation of numerical methods for fractional ODEs (FODEs) is quite limited since rigorous numerical analysis of numerical methods of FODEs has met additional difficulties. However, there has been a growing interest in the research of this area.

Langlands and Henry [8] considered the diffusion equation with fractional order time derivatives, and introduced an L_1-stable scheme for this equation. Sun and Wu [15] constructed a finite difference scheme with L_1 approximation for the fractional time derivative. Lin and Xu [10] analyzed a finite difference scheme for the time discretization of the time fractional diffusion equation, and proved that the convergence in time is of $2 - \alpha$ order. Lv and Xu [12] improved the error estimates by giving a more accurate coefficient. Zhao, Sun and Karniadakis [22] derived two second-order approximation formulas for fractional time derivatives involved in anomalous diffusion and wave propagation. Lin and Liu [9] analyzed a linear multistep method and proved the stability and convergence of the method. Kumar and Agrawal [7] proposed another numerical approach for a class of FODEs, which can be reduced into a Volterra type integral equation. Based on this approach, Cao and Xu [2] presented a general technique to construct high order numerical schemes for FODEs.

In this paper, we use the following numerical approximation of the Caputo derivative, which basically follows the work done by Lin and Xu in [10]. We assume uniform time step size τ, and denote $t^n = n\tau$, $n = 0, 1, 2, \ldots$. We denote numerical approximation of $u(t^n)$ by U^n.

To construct a first order method, for $t = t^{n+1}$, we assume uniform partition in time,

$$0 = t_0 < t_1 < \cdots < t_n < t_{n+1} = t.$$

Afterwards, each standard time derivative is approximated by the forward difference. That is,

$$\partial_t^\alpha u(t^{n+1}) \approx \frac{1}{\Gamma(1-\alpha)} \sum_{k=0}^n \int_{t_k}^{t_{k+1}} \frac{U^{k+1} - U^k}{\tau(t_{n+1} - s)^\alpha} \, ds$$

$$= \frac{1}{\Gamma(1-\alpha)(1-\alpha)} \sum_{k=0}^n \frac{(n+1-k)^{1-\alpha} - (n-k)^{1-\alpha}}{\tau^\alpha} (U^{k+1} - U^k)$$

$$= \frac{1}{\Gamma(2-\alpha)\tau^\alpha} \left(U^{n+1} - \sum_{n=0}^{k} c_k^{n+1} U^k \right) := D_t^\alpha U^{n+1}, \tag{14}$$

where

$$c_k^{n+1} = 2(n+1-k)^{1-\alpha} - (n+2-k)^{1-\alpha} - (n-k)^{1-\alpha}, \quad k = 1, \ldots, n.$$
$$c_0^{n+1} = (n+1)^{1-\alpha} - n^{1-\alpha}.$$

By direct calculation, and note that $y = x^{1-\alpha}$ is an increasing and concave function for $x > 0$, we get,

$$\sum_{k=0}^{n} c_k^{n+1} = 1, \quad c_k^{n+1} > 0, \quad k = 0, \ldots, n. \tag{15}$$

Hence, while the standard time derivative gives the instantaneous rate of change, from its numerical approximation, from (14), the Caputo derivative can be interpreted as the rate of change of a quantity from a convex combination of its history values, and the coefficients c_k^{n+1} specifies the influence strength due to the memory effect. And the influence strength of a history values decreases in time since its memory effect becomes weaker.

It is worth remarking that, we have for fixed α,

$$c_n^{n+1} = 2 - 2^{1-\alpha} - 0^{1-\alpha} = 2 - 2^{1-\alpha},$$

which is independent of n. Hence, we further denote c_n^{n+1} by $\tilde{c}$, which plays a significant role in the CFL conditions for explicit upwind schemes, as we shall show later.

The consistency error has been proved by Lv and Xu [12], which is the following Theorem,

Lemma 2.1 *For any $\alpha \in (0, 1)$, the truncation error of this scheme given by*

$$\partial_t^\alpha u(t^n) = D_t^\alpha u(t^n) + r_\tau^n, \tag{16}$$

satisfied the following error estimate

$$|r_\tau^k| \leqslant C M(u) \tau^{2-\alpha}, \quad \forall k = 0, 1, \ldots, n, \tag{17}$$

where C is independent of u and τ, $M(u) = \max_{t \in (0, t^n]} |\partial_t^2 u(t)|$.

It is worth remarking that, the focus of the current paper is to construct and analyze numerical methods for conservation laws with the Caputo derivative, so we choose to use a prevailing method for numerical approximation of the Caputo derivative. Actually, since many higher order approximation are available for the Caputo derivative, (see, e.g., [2,22]), the accuracy in time of the numerical methods proposed in the following can easily be improved with these results.

3 Numerical Methods and Stability Analysis

In this section, we utilize the numerical approximation of the Caputo derivative introduced in the previous section to design numerical schemes for ODE's and conservation laws. We focus on the stability condition for each scheme, especially how they differ from the models with standard time derivatives.

3.1 Backward Euler Method for an ODE Model

We consider the following ODE model,

$$\partial_t^\alpha u(t) = \lambda u(t). \tag{18}$$

Here, λ is a complex number with $\mathbf{Re}(\lambda) \leqslant 0$, which is reminiscent of the eigenvalue of an (discrete) operator.

The stability analysis of several numerical schemes for ODE models with Caputo derivatives have been shown in some previous works, see [2,6,10,12,18]. However, in this work, since we aim to design fully implicit schemes for nonlinear conservations laws in Sect. 3.3, we need to utilize the numerical methods for the ODE models where the time discretization is given by a backward differentiation formula (BDF), which have not been studied yet. Thus, we start by considering the backward Euler method for such ODE models.

$$D_t^\alpha U^{n+1} = \lambda U^{n+1}. \tag{19}$$

Multiply each side by $\tau^\alpha \Gamma(2-\alpha)$, we reformulate the above equation to

$$\left(1 - \lambda \tau^\alpha \Gamma(2-\alpha)\right) U^{n+1} = \sum_{k=0}^{n} c_k^{n+1} U^k.$$

If we denote $z = \lambda \tau^\alpha \Gamma(2-\alpha)$, the stability polynomial $\pi(\xi; z)$ for this numerical method is

$$\pi(\xi; z) = (1 - z) \xi^{n+1} - \sum_{k=0}^{n} c_k^{n+1} \xi^k.$$

In the following, we discuss two different situations, i.e., when $\lambda \neq 0$ and when $\lambda = 0$.

When $\lambda \neq 0$, then $\mathbf{Re}(z) \leqslant 0$ and $z \neq 0$, then we obtain that, $|1 - z| > 1$. If we assume ξ_0 with $|\xi_0| \geqslant 1$ is a root to $\pi(\xi; z)$, then for $k \leqslant n$, we have

$$|\xi_0^k| \leqslant |\xi_0|^k \leqslant |\xi_0^{n+1}|.$$

Then, we get,

$$\left| (1-z)\xi_0^{n+1} \right| = |1-z||\xi_0|^{n+1} = \left| \sum_{k=0}^{n} c_k^{n+1} \xi_0^k \right|$$

$$\leqslant \sum_{k=0}^{n} c_k^{n+1} |\xi_0|^k \leqslant \left(\sum_{k=0}^{n} c_k^{n+1} \right) |\xi_0|^{n+1} = |\xi_0|^{n+1},$$

which is a contradiction. This means, the stability polynomial only has roots with modulus less than 1, and the method is absolute stable.

When $\lambda = 0$, then $z = 0$, and the stability analysis reduces to the zero stability of the time discretization. If the modulus of the root of the stability polynomial is strictly larger than 1, then the analysis above still carries out, and one can show that no such roots exist.

If the modulus of the root of the stability polynomial is 1, we can assume that root is $\xi_0 = e^{i\theta}$. If $\theta = 0$, then $\xi_0 = 1$, we get

$$\pi(1; 0) = 1^{n+1} - \sum_{k=0}^{n} c_k^{n+1} = 0.$$

Then, we compute

$$\frac{d\pi(\xi; 0)}{d\xi} = (n+1)\xi^n - \sum_{k=0}^{n} c_k^{n+1} k \xi^{k-1}.$$

Note that, the coefficients $\{c_k^{n+1}\}_{k=0}^{n}$ satisfy condition (15), therefore,

$$\left| \sum_{k=0}^{n} c_k^{n+1} k \right| < \left| \sum_{k=0}^{n} c_k^{n+1} n \right| = n.$$

From this, we conclude that,

$$\frac{d\pi(1; 0)}{d\xi} = (n+1) - \sum_{k=0}^{n} c_k k > n + 1 - n = 1 \neq 0.$$

Hence, 1 is not a repeat root of the stability polynomial.

If $\theta \neq 0$, then we get the following equation,

$$e^{i(n+1)\theta} = \sum_{k=0}^{n} c_k^{n+1} e^{ik\theta}.$$

By dividing each side by $e^{i(n+1)\theta}$, we get

$$1 = \sum_{k=0}^{n} c_k^{n+1} e^{i(k-1-n)\theta}.$$

Since $\theta \neq 0$, at least one $e^{i(k-1-n)\theta}$ is not real-valued. Therefore, the right hand side of the above equation is a convex combination of $n+1$ unit complex numbers. So, we conclude that the right hand side never add up to 1. Therefore, $e^{i\theta}$ with $\theta \neq 0$ is not a root to the stability polynomial.

Finally, we conclude that, when $\mathbf{Re}(\lambda) \leqslant 0$, the backward Euler method for the ODE model is unconditionally stable. In other words, we have proved that

Theorem 3.1 *The backward Euler method (19) for the time fractional ODE model (18) is A-stable.*

This result agrees with the stability results in [10] for ODE's and parabolic equations, which is not very surprising since it is believed that the fractional time derivative adds dissipation in time [10,22]. However, stability analysis for fractional time hyperbolic problems is quite open, and in the next section, we aim to focus on the scalar conservation laws.

To provide some intuitive knowledge of the stability region of the Backward Euler method (19), we use the boundary locus method for linear multistep methods to numerically plot the boundary points of this method for different α and n in the complex plane of the z variable. As shown in Fig. 2, the stability regions are the exteriors to the closed curves. We observe that, the stability region not only depends on α, but also, it depends on n, which modifies the coefficients and is in proportion to the length of memory effect. As $n \to \infty$, we see that the boundary is asymptotically approaching a limit curve.

Notice that, our stability region include the imaginary axis $\mathbf{Re}(\lambda) = 0$, which is the same case for backward Euler method with standard time derivatives, and thus it has the potential to apply for hyperbolic equations. Although, for hyperbolic problems with standard time derivatives, explicit methods are in general preferred, we shall see that the Caputo time
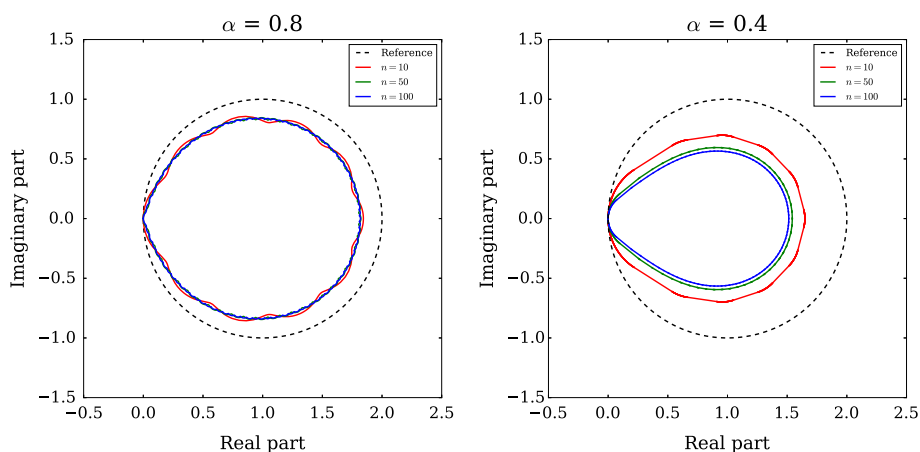
Fig. 2 *Left*: Absolute stability zone for $\alpha = 0.8$, $n = 10, 50, 100$. *Right*: Absolute stability zone for $\alpha = 0.4$, $n = 10, 50, 100$. Reference: backward Euler for the standard derivative

derivatives cause additional constraint in stability, which motivates us to design implicit schemes for (1).

3.2 Explicit Upwind Method for the Scalar Conservation Law

3.2.1 The First Order Scheme

We consider the one dimensional conservation law

$$\partial_t^\alpha u + (f(u))_x = 0, \tag{20}$$

where the flux function can be decomposed as

$$f = f^+ + f^-, \quad (f^+)' \geqslant 0, \quad (f^-)' \leqslant 0. \tag{21}$$

We make this technique assumption to simplify our analysis. Note that, the flux decomposition is not essential in designing such numerical schemes. The readers may refer to [14,17] for general discussion on this issue. Without loss of generality, the flux function $f(u)$ might be nonlinear. But obviously, it also includes the linear advection case, when $f = au$.

Again, we assume uniform time step size τ, and denote $t^n = n\tau$, $n = 0, 1, 2, \ldots$. Also, on the computation domain $[a, b]$, we assume uniform spatial grids $x_j = a + jh$, for $j = 0, 1, \ldots, M$, where the spatial grid size $h = \frac{b-a}{M}$. We denote the numerical approximation of $u(x = x_j, t = t^k)$ by U_j^k. Then, the first order upwind method for the nonlinear conservation law is,

$$D_t^\alpha U_j^{n+1} + \frac{1}{h}\left(f^+\left(U_j^n\right) - f^+\left(U_{j-1}^n\right)\right) + \frac{1}{h}\left(f^-\left(U_{j+1}^n\right) - f^-\left(U_j^n\right)\right) = 0. \tag{22}$$

If we denote

$$\lambda_j^{+,n} = \frac{a_j^{+,n}\tau^\alpha\Gamma(2-\alpha)}{h}, \quad \lambda_j^{-,n} = \frac{a_j^{-,n}\tau^\alpha\Gamma(2-\alpha)}{h},$$

where for some ξ_j^n between U_{j-1}^n and U_j^n and some η_j^n between U_j^n and U_{j+1}^n, we have

$$a_j^{+,n} = \frac{f^+\left(U_j^n\right) - f^+\left(U_{j-1}^n\right)}{U_j^n - U_{j-1}^n} = (f^+)'\left(\xi_j^n\right) \geqslant 0,$$

$$a_j^{-,n} = \frac{f^-\left(U_{j+1}^n\right) - f^-\left(U_j^n\right)}{U_{j+1}^n - U_j^n} = (f^-)'\left(\eta_j^n\right) \leqslant 0,$$

then, the numerical method can be rewritten as

$$U_j^{n+1} = \left(\tilde{c} - \lambda_j^{+,n} + \lambda_j^{-,n}\right) U_j^n + \lambda_j^{+,n} U_{j-1}^n - \lambda_j^{-,n} U_{j+1}^n + \sum_{k=0}^{n-1} c_k^{n+1} U_j^k. \quad (23)$$

Therefore, we propose the following CFL condition for the first order upwind method:

$$\frac{\tau^\alpha \Gamma(2-\alpha)}{h} \left(\max |(f^+)'| + \max |(f^-)'|\right) \leqslant \tilde{c}. \quad (24)$$

We observe that, the CFL condition essentially agrees with the conservation law with standard time derivatives except that the time step τ gains an exponent α due to the Caputo derivative.

With the CFL condition, we have

$$\tilde{c} - \lambda_j^{+,n} + \lambda_j^{-,n} \geqslant 0.$$

Hence, we can easily conclude the maximal principle for the upwind method, $\forall n \in \mathbb{N}^+$

$$\max_j |U_j^n| \leqslant \max_j |U_j^0|.$$

Also, we can show that this method is TVD if the CFL condition (24) is satisfied. Actually, we can rewrite (23) as

$$U_j^{n+1} = \tilde{c} U_j^n - \delta f^+\left(U_j^n\right) + \delta f^+\left(U_j^n\right) + \delta f^+\left(U_{j-1}^n\right) - \delta f^+\left(U_{j+1}^n\right)$$
$$+ \sum_{k=0}^{n-1} c_k^{n+1} U_j^k, \quad (25)$$

where $\delta = \frac{\tau^\alpha \Gamma(2-\alpha)}{h}$. And we consider another solution V_j^n, which satisfies the same difference equation,

$$V_j^{n+1} = \tilde{c} V_j^n - \delta f^+\left(V_j^n\right) + \delta f^+\left(V_j^n\right) + \delta f^+\left(V_{j-1}^n\right) - \delta f^+\left(V_{j+1}^n\right)$$
$$+ \sum_{k=0}^{n-1} c_k^{n+1} V_j^k. \quad (26)$$

Subtract these two equations, and we can get

$$U_j^{n+1} - V_j^{n+1} = \tilde{c}\left(U_j^n - V_j^n\right) - \delta\left(f^+\left(U_j^n\right) - f^+\left(V_j^n\right)\right) + \delta\left(f^-\left(U_j^n\right) - f^-\left(V_j^n\right)\right)$$
$$+ \delta\left(f^+\left(U_{j-1}^n\right) - f^+\left(V_{j-1}^n\right)\right) - \delta\left(f^-\left(U_{j+1}^n\right) - f^-\left(V_{j+1}^n\right)\right)$$
$$+ \sum_{k=0}^{n-1} c_k^{n+1}\left(U_j^k - V_j^k\right).$$

Then, by mean value theorem, we have

$$
\begin{aligned}
U_j^{n+1} - V_j^{n+1} &= \left(\tilde{c} - \delta(f^+)' \left(\xi_j^+ \right) + \delta(f^-)' \left(\xi_j^- \right) \right) \left(U_j^n - V_j^n \right) \\
&\quad + \delta(f^+)' \left(\xi_{j-1}^+ \right) \left(U_{j-1}^n - V_{j-1}^n \right) \\
&\quad - \delta(f^-)' \left(\xi_{j+1}^- \right) \left(U_{j+1}^n - V_{j+1}^n \right) + \sum_{k=0}^{n-1} c_k^{n+1} \left(U_j^k - V_j^k \right),
\end{aligned}
$$

where ξ_j^+ and ξ_j^- are respectively some numbers between U_j^n and V_j^n. Note, when the CFL condition (24) is satisfied, we have

$$
\tilde{c} - \delta(f^+)'(\xi_j^+) + \delta(f^-)'(\xi_j^-) \geqslant 0,
$$

Then, by triangle inequality, we have

$$
\begin{aligned}
\left| U_j^{n+1} - V_j^{n+1} \right| &= \left(\tilde{c} - \delta(f^+)' \left(\xi_j^+ \right) + \delta(f^-)' \left(\xi_j^- \right) \right) \left| U_j^n - V_j^n \right| \\
&\quad + \delta(f^+)' \left(\xi_{j-1}^+ \right) \left| U_{j-1}^n - V_{j-1}^n \right| \\
&\quad - \delta(f^-)' \left(\xi_{j+1}^- \right) \left| U_{j+1}^n - V_{j+1}^n \right| + \sum_{k=0}^{n-1} c_k^{n+1} \left| U_j^k - V_j^k \right|.
\end{aligned}
$$

Then, we sum the equation over j and get

$$
\sum_j \left| U_j^{n+1} - V_j^{n+1} \right| \leqslant \sum_{k=0}^{n} c_k^{n+1} \sum_j \left| U_j^k - V_j^k \right|.
$$

Note here, the flux terms have all been canceled out. By induction, we get,

$$
\| U^n - V^n \|_{\ell^1} \leqslant \| U^0 - V^0 \|_{\ell^1},
$$

namely, we have proved the following theorem

Theorem 3.2 *The first order upwind method (22) for the scalar conservation law (1) is ℓ^1 contracting, when the CFL condition (24) is satisfied.*

As an immediate consequence, if we set V_j^n as U_{j+1}^n, then we get that for $n \in \mathbb{N}^+$,

$$
\mathrm{TV}\left[U^n \right] \leqslant \mathrm{TV}\left[U^0 \right],
$$

where $\mathrm{TV}\left[U^n \right] = \sum_j |U_{j+1}^n - U_j^n|$. In other words, we have shown that,

Corollary 3.1 *The first order upwind method (22) for the scalar conservation law (1) is TVD, when the CFL condition (24) is satisfied.*

3.2.2 MUSCL Scheme

To construct a second order method in space, the positive and negative fluxes can be approximated by the piecewise linear function,

$$
f^{\pm,n}(x) = f_j^{\pm,n} + s_j^{\pm,n}(x - x_j), \quad x_{j-\frac{1}{2}} < x < x_{j+\frac{1}{2}},
$$

where we denote $f_j^{\pm,n} = f^{\pm,n}(U_j^n)$. The slope functions are determined by a limiter function,

$$s_j^{\pm,n} = \frac{f_j^{\pm,n} - f_{j-1}^{\pm,n}}{h} \phi^0 \left(\frac{f_{j+1}^{\pm,n} - f_j^{\pm,n}}{f_j^{\pm,n} - f_{j-1}^{\pm,n}} \right).$$

In this paper, we only consider the minmod limiter

$$\phi^0(\theta) = \max(0, \min(1, \theta)),$$

or the Van Leer limiter,

$$\phi^0(\theta) = \frac{|\theta| + \theta}{1 + \theta}.$$

Note that, both limiters are symmetric, i.e., $a\phi^0(b/a) = b\phi^0(a/b)$. Consequently, the second-order flux splitting method is given by

$$D_t^\alpha U_j^{n+1} + \frac{1}{h} \left(f^{+,n} \left(x_{j+\frac{1}{2}} - \right) - f^{+,n} \left(x_{j-\frac{1}{2}} - \right) \right)$$
$$+ \frac{1}{h} \left(f^{-,n} \left(x_{j+\frac{1}{2}} + \right) - f^{-,n} \left(x_{j-\frac{1}{2}} + \right) \right) = 0.$$

Note that, the numerical method above can be rewritten as

$$D_t^\alpha U_j^{n+1} + \psi_j^{+,n} \frac{f_j^{+,n} - f_{j-1}^{+,n}}{h} + \psi_j^{-,n} \frac{f_{j+1}^{-,n} - f_j^{-,n}}{h} = 0.$$

with the coefficients

$$\psi_j^{+,n} = 1 + \frac{1}{2} \phi^0 \left(\frac{f_{j+1}^{+,n} - f_j^{+,n}}{f_j^{+,n} - f_{j-1}^{+,n}} \right) - \frac{1}{2} \phi^0 \left(\frac{f_{j-1}^{+,n} - f_{j-2}^{+,n}}{f_j^{+,n} - f_{j-1}^{+,n}} \right),$$

$$\psi_j^{-,n} = 1 + \frac{1}{2} \phi^0 \left(\frac{f_{j+2}^{-,n} - f_{j+1}^{-,n}}{f_{j+1}^{-,n} - f_j^{-,n}} \right) - \frac{1}{2} \phi^0 \left(\frac{f_j^{-,n} - f_{j-1}^{-,n}}{f_{j+1}^{-,n} - f_j^{-,n}} \right).$$

Again, one can apply the mean value theorem for f^+ and f^-, and the second order flux splitting scheme for the nonlinear conservation law problem can still be reformulated in the form of (23), with

$$a_j^{+,n} = \psi_j^{+,n}(f^+)' \left(\xi_j^n \right), \quad a_j^{-,n} = \psi_j^{-,n}(f^-)' \left(\eta_j^n \right).$$

Since both limiters satisfy the conditions,

$$0 \leqslant \frac{\phi^0(\theta)}{\theta} \leqslant 2, \quad 0 \leqslant \phi^0(\theta) \leqslant 2,$$

so the two coefficients $\psi_j^{\pm,n} \in [0, 2]$, which implies $\pm a_j^{\pm,n}$ are both nonnegative. Also, we can propose the following CFL condition for the second order flux splitting method,

$$2\frac{\tau^\alpha \Gamma(2 - \alpha)}{h} \left(\max |(f^+)'| + \max |(f^-)'| \right) \leqslant \tilde{c}. \tag{27}$$

Clearly, with the above condition, the coefficients on the right hand side of the form (23) are all nonnegative. Therefore, we can conclude conditional TVD properties and stability properties by the argument similar to the first order cases.

We observe that, for explicit methods of conservation laws, although the stability constraint only changes to $\tau^\alpha = O(h)$ due to the Caputo derivative, in practice, it makes these methods infeasible. For example, when $\alpha = \frac{1}{2}$, this constraint is already as restricted as explicit methods for parabolic equations. And, when $\alpha \to 0$, the choice of time steps suffers severely from the CFL conditions.

3.3 Implicit Upwind Method for the Scalar Conservation Law

3.3.1 Stability Analysis

As we see in the previous section, albeit we are able to derive modified CFL condition for the scalar conservation law, the stability constraint implies that the time step Δt is a higher order small quantity of the spatial size Δx. Therefore, we are motivated to analyze the implicit upwind scheme for Eq. (20) with the flux function f satisfies the condition (21), which is given by

$$D_t^\alpha U_j^{n+1} + \frac{1}{h}\left(f^+\left(U_{j-1}^{n+1}\right) - f^+\left(U_{j-2}^{n+1}\right)\right) + \frac{1}{h}\left(f^-\left(U_{j+1}^{n+1}\right) - f^-\left(U_j^{n+1}\right)\right) = 0. \tag{28}$$

Next, we introduce similar notations to reformulate the scheme. If we denote

$$\lambda_j^{+,n} = \frac{a_j^{+,n}\tau^\alpha\Gamma(2-\alpha)}{h}, \quad \lambda_j^{-,n} = \frac{a_j^{-,n}\tau^\alpha\Gamma(2-\alpha)}{h},$$

where for some ξ_j^{n+1} between U_{j-1}^{n+1} and U_j^{n+1} and some η_j^{n+1} between U_j^{n+1} and U_{j+1}^{n+1}, we have

$$a_j^{+,n+1} = \frac{f^+\left(U_j^{n+1}\right) - f^+\left(U_{j-1}^{n+1}\right)}{U_j^{n+1} - U_{j-1}^{n+1}} = (f^+)'\left(\xi_j^{n+1}\right) \geqslant 0,$$

$$a_j^{-,n+1} = \frac{f^-\left(U_{j+1}^{n+1}\right) - f^-\left(U_j^{n+1}\right)}{U_{j+1}^{n+1} - U_j^{n+1}} = (f^-)'\left(\eta_j^{n+1}\right) \leqslant 0,$$

then, the numerical method can be rewritten as

$$U_j^{n+1} - \sum_{k=0}^n c_k^{n+1}U_j^k = -\lambda_j^{+,n+1}\left(U_j^{n+1} - U_{j-1}^{n+1}\right) - \lambda_j^{-,n+1}\left(U_{j+1}^{n+1} - U_j^{n+1}\right). \tag{29}$$

Now, we show the following result,

Theorem 3.3 *The implicit upwind scheme (28) for the conservation law (1) is unconditionally ℓ^1 contracting.*

Proof We further rewrite the scheme (29) as

$$U_j^{n+1} - \sum_{k=0}^n c_k^{n+1}U_j^k = -\delta\left(f^+\left(U_{j-1}^{n+1}\right) - f^+\left(U_{j-2}^{n+1}\right)\right)$$

$$-\delta\left(f^-\left(U_{j+1}^{n+1}\right) - f^-\left(U_j^{n+1}\right)\right). \tag{30}$$

We recall for convenience that $\delta = \frac{\tau^\alpha}{h}$. And we assume V_j^n also gives the numerical solution to the same conservation law, which is also given by the implicit upwind scheme

$$V_j^{n+1} - \sum_{k=0}^{n} c_k^{n+1} V_j^k = -\delta \left(f^+ \left(V_{j-1}^{n+1} \right) - f^+ \left(V_{j-2}^{n+1} \right) \right)$$
$$-\delta \left(f^- \left(V_{j+1}^{n+1} \right) - f^- \left(V_j^{n+1} \right) \right). \tag{31}$$

By subtracting the above formula from the original scheme, and by applying mean value theorem, we get

$$U_j^{n+1} - V_j^{n+1} + \delta (f^+)' \left(\xi_j^+ \right) \left(U_j^{n+1} - V_j^{n+1} \right) - \delta (f^-)' \left(\xi_j^- \right) \left(U_j^{n+1} - V_j^{n+1} \right)$$
$$= \sum_{k=0}^{n} c_k^{n+1} \left(U_j^k - V_j^k \right) - \delta (f^-)' \left(\xi_{j-1}^- \right) \left(U_{j+1}^{n+1} - V_{j+1}^{n+1} \right)$$
$$- \delta (f^+)' \left(\xi_{j-1}^+ \right) \left(U_{j-1}^{n+1} - V_{j-1}^{n+1} \right),$$

where ξ_j^+ and ξ_j^- are respectively some numbers between U_j^{n+1} and V_j^{n+1}.

By multiplying each side by $\mathrm{Sgn} \left(U_j^{n+1} - V_j^{n+1} \right)$, and summing over j, we get

$$\text{L.H.S.} = \sum_j \left| U_j^{n+1} - V_j^{n+1} \right| + \sum_j \delta (f^+)' \left(\xi_j^+ \right) \left| U_j^{n+1} - V_j^{n+1} \right|$$
$$- \sum_j \delta (f^-)' \left(\xi_j^- \right) \left(U_j^{n+1} - V_j^{n+1} \right)$$
$$= \sum_j \left| U_j^{n+1} - V_j^{n+1} \right| + \delta \sum_j \left(\left| \delta (f^+)' \left(\xi_j^+ \right) \left(U_j^{n+1} - V_j^{n+1} \right) \right| \right.$$
$$\left. + \left| (f^-)' \left(\xi_j^- \right) \left(U_j^{n+1} - V_j^{n+1} \right) \right| \right)$$
$$= \sum_j \left| U_j^{n+1} - V_j^{n+1} \right| + \delta \sum_j \left[\left| f^+ \left(U_j^{n+1} \right) - f^+ \left(V_j^{n+1} \right) \right| \right.$$
$$\left. + \left| f^- \left(U_j^{n+1} \right) - f^+ \left(V_j^{n+1} \right) \right| \right].$$

For the right hand side, by triangle inequality, we obtain,

$$\text{R.H.S.} \leqslant \sum_j \sum_{k=0}^{n} c_k^{n+1} \left| U_j^k - V_j^k \right| + \delta \sum_j \left(\left| (f^+)' \left(\xi_{j-1}^+ \right) \left(U_{j-1}^{n+1} - V_{j-1}^{n+1} \right) \right| \right.$$
$$\left. + \left| (f^-)' \left(\xi_{j+1}^- \right) \left(U_{j+1}^{n+1} - V_{j+1}^{n+1} \right) \right| \right)$$
$$= \sum_j \sum_{k=0}^{n} c_k^{n+1} \left| U_j^k - V_j^k \right| + \delta \sum_j \left[\left| f^+ \left(U_j^{n+1} \right) - f^+ \left(V_j^{n+1} \right) \right| \right.$$
$$\left. + \left| f^- \left(U_j^{n+1} \right) - f^+ \left(V_j^{n+1} \right) \right| \right].$$

Thus, we conclude that

$$\sum_j \left| U_j^{n+1} - V_j^{n+1} \right| \leqslant \sum_j \sum_{k=0}^{n} c_k^{n+1} \left| U_j^k - V_j^k \right|.$$

With an induction argument, we get,

$$\|U^n - V^n\|_{\ell^1} \leqslant \|U^0 - V^0\|_{\ell^1},$$

namely, the implicit upwind method is ℓ^1 contracting. □

Similar to the previous case, as an immediate result, we obtain

Corollary 3.2 *The implicit upwind scheme (28) for the conservation law (1) is unconditionally TVD.*

3.3.2 Energy Estimate and Entropy Solutions

To further investigate the numerical dissipation introduced when approximating the Caputo derivative, we apply the following energy method, and show that the implicit upwind method is unconditionally l^2 stable for the linear advection equation, i.e. $f = au$. For simplicity, we take $a > 0$, then the right hand side of (29) reduces to

$$\text{R.H.S.} = -\lambda \left(U_j^{n+1} - U_{j-1}^{n+1} \right),$$

where $\lambda = a\tau^\alpha \Gamma(2 - \alpha)/h$.

Multiply Eq. (29) by U_j^{n+1}, and sum over j, we get that

$$
\begin{aligned}
\text{L.H.S.} &= \sum_j U_j^{n+1} \left(U_j^{n+1} - \sum_{k=0}^n c_k^{n+1} U_j^k \right) \\
&= \sum_j \sum_{k=0}^n c_k^{n+1} \left[\left(U_j^{n+1} \right)^2 - U_j^{n+1} U_j^k \right] \\
&= \frac{1}{2} \sum_j \sum_{k=0}^n c_k^{n+1} \left[\left(U_j^{n+1} \right)^2 + \left(U_j^{n+1} - U_j^k \right)^2 - \left(U_j^k \right)^2 \right] \\
&= \frac{1}{2} \sum_j \left(U_j^{n+1} \right)^2 + \frac{1}{2} \sum_j \sum_{k=0}^n c_k^{n+1} \left(U_j^{n+1} - U_j^k \right)^2 - \frac{1}{2} \sum_j \sum_{k=0}^n c_k^{n+1} \left(U_j^k \right)^2 \\
&= \frac{1}{2} \|U^{n+1}\|_{l^2}^2 - \frac{1}{2} \sum_{k=0}^n c_k^{n+1} \|U^k\|_{l^2}^2 + \frac{1}{2} \sum_j \sum_{k=0}^n c_k^{n+1} \left(U_j^{n+1} - U_j^k \right)^2.
\end{aligned}
$$

and

$$
\begin{aligned}
\text{R.H.S.} &= \sum_j -\lambda U_j^{n+1} \left(U_j^{n+1} - U_{j-1}^{n+1} \right) \\
&= -\frac{\lambda}{2} \sum_j \left[\left(U_j^{n+1} \right)^2 - 2 U_j^{n+1} U_{j-1}^{n+1} + \left(U_{j-1}^{n+1} \right)^2 \right] \\
&= -\frac{\lambda}{2} \sum_j \left(U_j^{n+1} - U_{j-1}^{n+1} \right)^2.
\end{aligned}
$$

Therefore, we obtain that,

$$\|U^{n+1}\|_{l^2}^2 + \sum_j \sum_{k=0}^n c_k^{n+1} \left(U_j^{n+1} - U_j^k \right)^2 + \lambda \sum_j \left(U_j^{n+1} - U_{j-1}^{n+1} \right)^2 = \sum_{k=0}^n c_k^{n+1} \|U^k\|_{l^2}^2.$$

With an induction argument, we can easily show that for $n \in \mathbb{N}^+$, $\|U^n\|_{l^2}^2 \leqslant \|U^0\|_{l^2}^2$, and hence, we conclude the following estimate,

$$\|U^n\|_{l^2}^2 + \sum_j \sum_{k=0}^{n-1} c_k^n \left(U_j^n - U_j^k\right)^2 + \lambda \sum_j \left(U_j^n - U_{j-1}^n\right)^2 \leqslant \|U^0\|_{l^2}^2.$$

We remark that the second term of the left hand side corresponds to the damping effect of the fractional time derivative, and the third term of the left hand side corresponds to the numerical dissipation by the upwind method.

At last, we want to verify that, the implicit upwind method also satisfies the entropy condition for the linear advection equation. Assume that $\eta(u)$ is a convex entropy function and $\psi(u)$ is its entropy flux function. For the linear advection equation, we have clearly $\psi(u) = a\eta(u)$. Without loss of generality, we take $a > 0$, and we rewrite the implicit upwind scheme in the following way,

$$U_j^{n+1} + \lambda U_j^{n+1} = \sum_{k=0}^{n} c_k^{n+1} U_j^k + \lambda U_{j-1}^{n+1},$$

where $\lambda = a\tau^\alpha/(hC_\alpha)$. Then, by dividing each side by $1 + \lambda$, we get,

$$U_j^{n+1} = \sum_{k=0}^{n} \frac{c_k^{n+1}}{1+\lambda} U_j^k + \frac{\lambda}{1+\lambda} U_{j-1}^{n+1}.$$

Clearly, the right hand side gives a convex combination since

$$\sum_{k=0}^{n} c_k^{n+1} + \lambda = 1 + \lambda, \quad c_k^{n+1} > 0.$$

Then, the convexity of the entropy function implies,

$$\eta\left(U_j^{n+1}\right) = \eta\left(\sum_{k=0}^{n} \frac{c_k^{n+1}}{1+\lambda} U_j^k + \frac{\lambda}{1+\lambda} U_{j-1}^{n+1}\right) \leqslant \sum_{k=0}^{n} \frac{c_k^{n+1}}{1+\lambda} \eta\left(U_j^k\right) + \frac{\lambda}{1+\lambda} \eta\left(U_{j-1}^{n+1}\right).$$

Then, if sum up the above inequality over j and denote $\eta(U^n) = \sum_j \eta(U_j^N)$, then we obtain that,

$$\eta\left(U^{n+1}\right) \leqslant \sum_{k=0}^{n} c_k^{n+1} \eta\left(U^k\right).$$

Then, by induction, we conclude that

$$\eta\left(U^{n+1}\right) \leqslant \eta\left(U^0\right),$$

which implies the discrete entropy does not increase in time.

For general conservation laws with Caputo derivatives, it is possible to show similar results. However, the analytical aspect of entropy solutions to conservation law with the Caputo derivatives is not completely understood yet. Therefore, we would leave the related numerical analysis as one of the possible future directions. But in this work, we would instead conduct various numerical tests.

4 Numerical Examples

4.1 Examples for Explicit Scheme

In this section, we will give numerical examples of our first and second order explicit schemes for scalar advection equations.

First, we consider the following advection equation:

$$\partial_t^\alpha u + \partial_x u = 0, \tag{32}$$

with a discontinuous initial data

$$u(x, 0) = \begin{cases} 2, & \text{if } x < 0, \\ 1, & \text{if } x \geq 0. \end{cases} \tag{33}$$

4.1.1 Convergence and Stability Test for the First Order Scheme

For the convergence tests, we will fix $\Delta t = 0.0001$ and compute the solution at time $T = 0.2$. The reference solution is obtained by using a fine mesh with $\Delta x = 0.001$ and $\Delta t = 0.0001$. The measure of error here we use is the ℓ^1 error which is:

$$\text{error} = \|u(x_j, T) - u_{\text{ref}}\|_{\ell^1}. \tag{34}$$

To make a comparison, we will test with $\alpha = 0.8$ and $\alpha = 0.9$ respectively. The result is shown in Fig. 3: We can easily verify that it is a first order scheme from this log-log plot.

For the stability tests we will fix $\Delta x = 0.01$ and let Δt increase from fine to coarse, we show the empirical critical value of Δt which makes our scheme diverges as shown in the lower row of Fig. 4, which is compared with results computed with the largest Δt satisfying the proposed CFL condition (24). We observe that, the stability conditions we derived are basically sharp.

We observe that the constriction on Δt is severely strict when using explicit methods, which makes the computation really time consuming. In particular, the stability condition is extremely restricted as $\alpha \to 0$, and makes the implementation impractical.
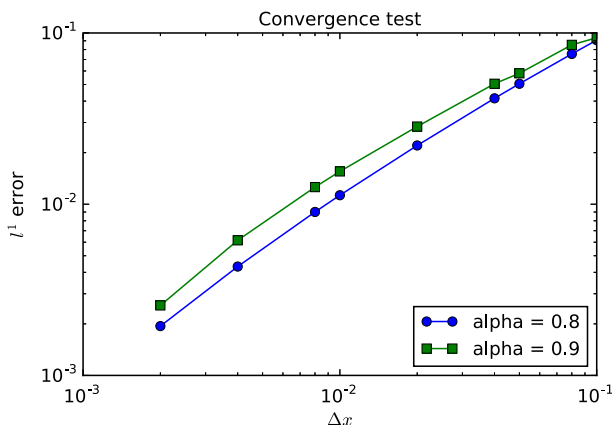
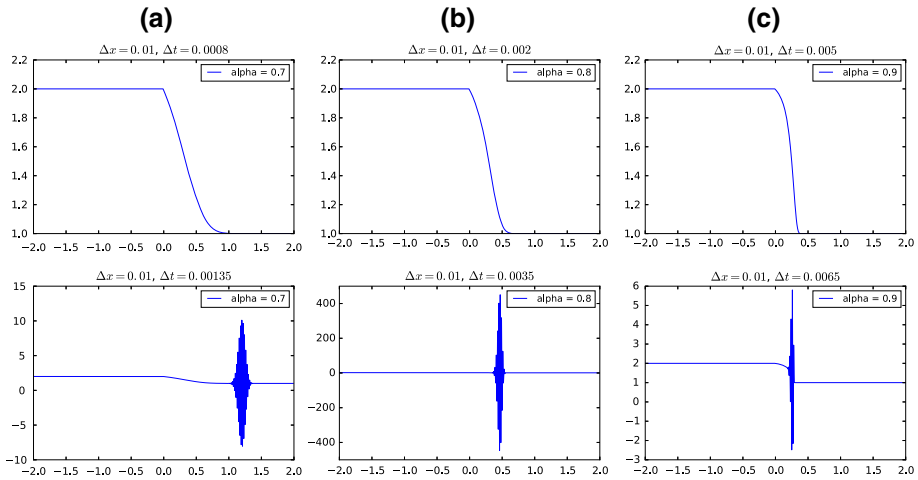**Fig. 3** Convergence test shows it is a first order scheme in Δx

Fig. 4 The stability condition test. **a** $\alpha = 0.7$. *Up*: scheme converges when $\Delta t = 0.0008$. *Below*: scheme diverges when $\Delta t = 0.00135$. **b** $\alpha = 0.8$. *Up*: scheme converges when $\Delta t = 0.002$. *Below*: scheme diverges when $\Delta t = 0.0035$. **c** $\alpha = 0.9$. *Up*: scheme converges when $\Delta t = 0.005$. *Below*: scheme diverges when $\Delta t = 0.0065$
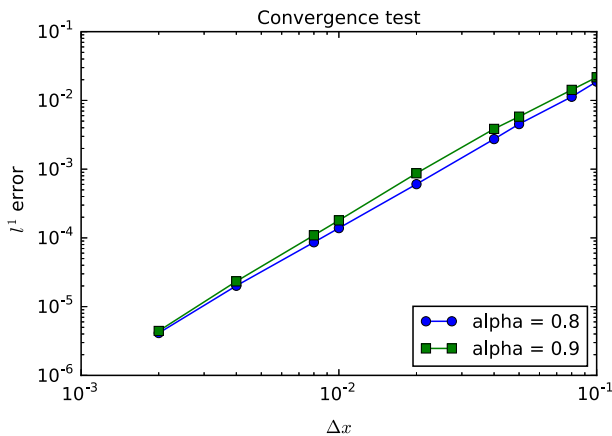

Fig. 5 Convergence test shows it is a second order scheme in Δx

4.1.2 Convergence and Stability Tests for the Second Order Scheme

In this part we basically repeat the numerical tests we did in the first order case. For the convergence test, we still fix $\Delta t = 0.0001$ and compute the solution at time $T = 0.2$. Due to the use of the limiters, the accuracy of MUSCL schemes may degrade to first order locally. For simplicity, we choose to use a continuous initial condition for the accuracy test,

$$u(x, 0) = e^{-10x^2} + 1 \tag{35}$$

We can easily verify that it is a second order scheme for this log-log plot (Fig. 5).
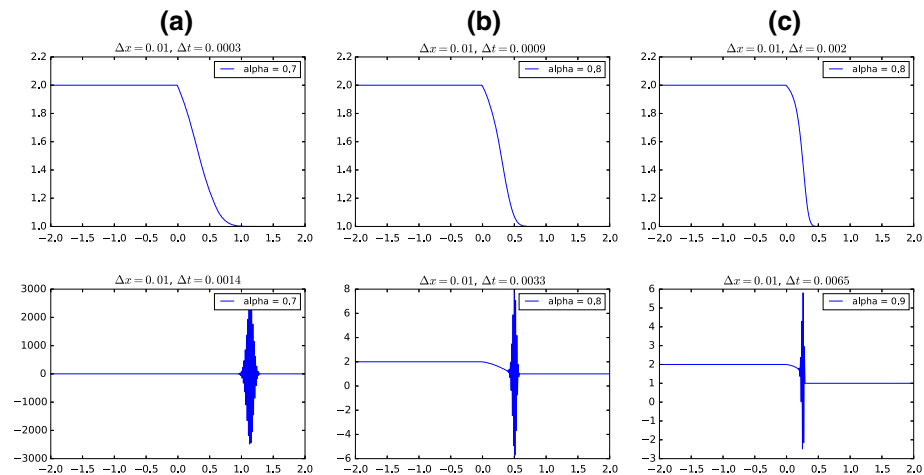
Fig. 6 The stability condition test. **a** $\alpha = 0.7$. *Up*: scheme converges when $\Delta t = 0.0003$. *Below*: scheme diverges when $\Delta t = 0.0014$. **b** $\alpha = 0.8$. *Up*: scheme converges when $\Delta t = 0.0009$. *Below*: scheme diverges when $\Delta t = 0.0033$. **c** $\alpha = 0.9$. *Up*: scheme converges when $\Delta t = 0.002$. *Below*: scheme diverges when $\Delta t = 0.0065$

For the stability test we repeat the same experiments as in the previous section. The results is shown in Fig. 6. Also We can find the strict constriction on Δt when using explicit method which makes the computation really inefficient.

To conclude this section, we remark that, we have carried out the same tests for the Burgers' equation, and similar results have been obtained, which we would skip in this paper.

4.2 Examples for Implicit Scheme

In the previous section, we show that it is nearly infeasible to use an explicit scheme for small α. Due to the restricted CFL conditions, an explicit scheme is extremely inefficient especially when $\alpha \to 0$. which motivates us to use an implicit scheme instead. By using an implicit scheme, we can conduct more numerical tests to explore more about the conservation law with the Caputo derivative.

4.2.1 Convergence Test for the Implicit Scheme

In this subsection, we will test the convergence of implicit upwind scheme. All the configurations for scalar convection equation are the same as previous section. We still fix $\Delta x = 0.01$ and since we expect this scheme to be unconditionally stable, we also choose $\alpha = 0.2$ which is the case we cannot afford in the explicit case. For the stability test, we choose $\Delta t = 0.01, 0.02, 0.04, 0.06, 0.08$ which is $O(\Delta x)$ as shown in Fig. 7 left. For the convergence test, we can now fix $\Delta t = 0.01$, thanks to the unconditionally stable feature and a first order convergence in space is observed (see Fig. 7 right).

This constraint-free stability feature also allows us to run a test of solving nonlinear fractional equations. Here we test a Burgers' equation with the Caputo derivative:

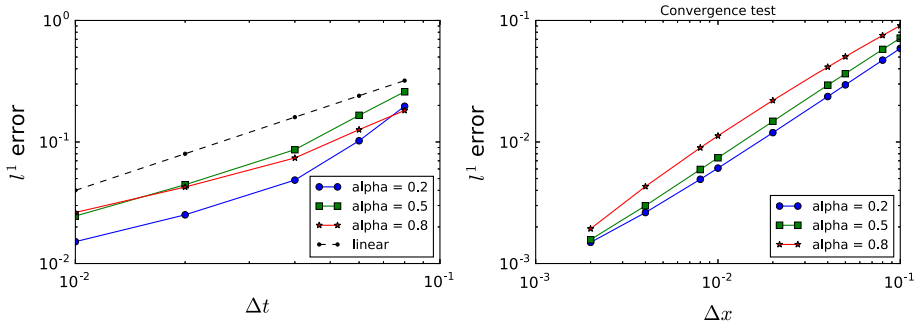$$\begin{cases} \partial_t^\alpha u + u \partial_x u = 0, \\ u(x, 0) = -\sin(\pi x). \end{cases} \tag{36}$$

Fig. 7 Implicit upwind scheme for the linear advection equation with different α. *Left*: stability test. *Right*: convergence test in Δx
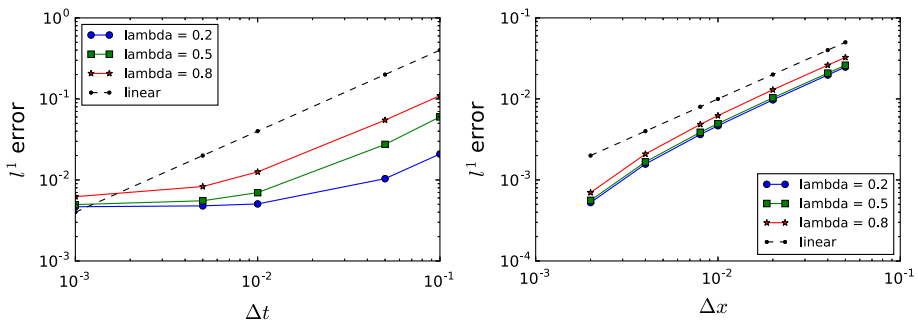

Fig. 8 Implicit upwind scheme for the Burgers' equation. *Left*: stability test. *Right*: convergence test in Δx

with fixed $\alpha = 0.2, 0.5, 0.8$. Same as above, for the stability test we fix $\Delta x = 0.01$ and increase Δt; for the convergence test we fix $\Delta t = 0.01$ and increase Δx. The results are shown in Fig. 8. We remark that, in the stability test, the numerical error decreases in proportion to Δt until the spatial error becomes dominant, which explains the flat error curve for small Δt.

4.2.2 Numerical Experiments Towards Understanding the Memory Effect

As we can see, since the implicit scheme is efficient and stable for conservation laws with the Caputo derivative, we will use this scheme to investigate these equations. We will give several tests in this section.

First we show the solutions with different α's for the advection equation and the Burgers' equation respectively. In Fig. 9 left, we observe that the solutions at discontinuous point exhibit convergent behavior as $\alpha \to 1$, finally converges to the solution when $\alpha = 1$ which is the standard convection equation. For the Burgers' equation, the same behavior is shown as in Fig. 9 right.

Next we consider the case of inhomogeneous memory effect, i.e., when α depends on x and t. In the linear convection case, we consider

$$\alpha(x, \lambda) = 1 - \lambda \exp(-30x^2 - 7000 \times (0.5)^{12}), \tag{37}$$
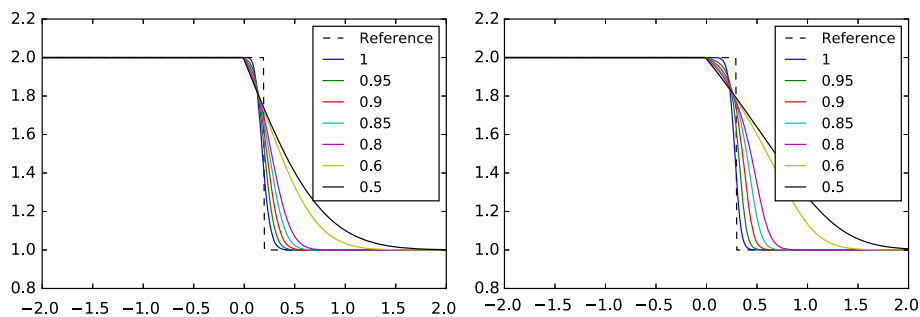
Fig. 9 Solutions at $T = 0.2$ with different α, $a = 1$, $\Delta t = \Delta x = 0.01$, by the implicit upwind method. *Left*: linear advection equation. *Right*: Burgers' equation
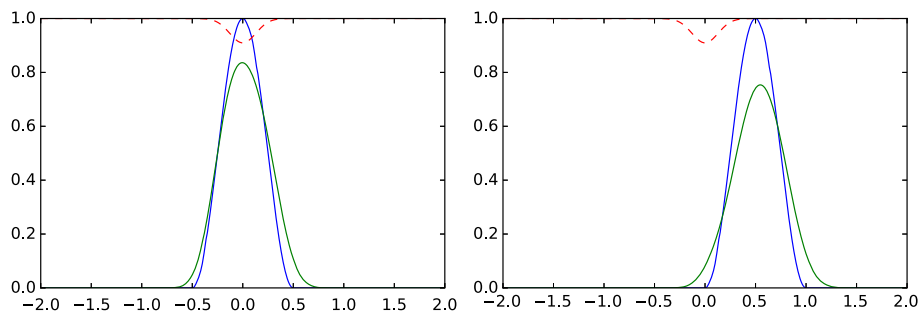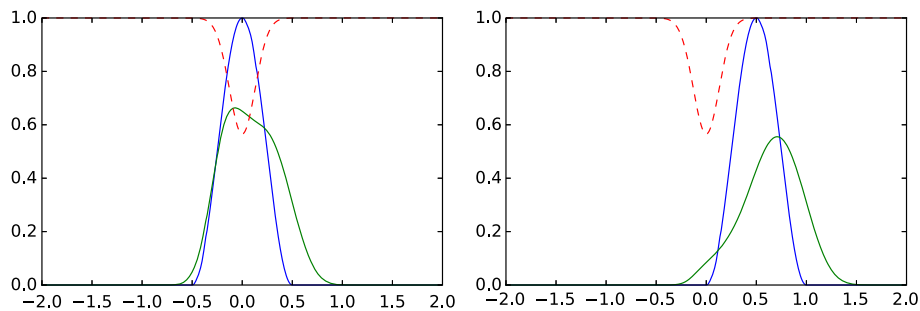

Fig. 10 *Left*: solution (*green line*) at $T = 1$ with $\lambda = 0.5$ (*dash line*) and exact solution with $T = 1$, $\alpha = 1$ (*blue line*). *Right*: solution (*green line*) at $T = 1.5$ with $\lambda = 0.5$ (*dash line*) and exact solution with $T = 1.5$, $\alpha = 1$ (*blue line*). *Dash line* is $\alpha(x, \lambda)$ with corresponding λs (Color figure online)

Fig. 11 *Left*: solution (*green line*) at $T = 1$ with $\lambda = 2.4$ (*dash line*) and exact solution with $T = 1$, $\alpha = 1$ (*blue line*). *Right*: solution (*green line*) at $T = 1.5$ with $\lambda = 2.4$ (*dash line*) and exact solution with $T = 1.5$, $\alpha = 1$ (*blue line*). *Dash line* is $\alpha(x, \lambda)$ with corresponding λs (Color figure online)

with initial data

$$u(x, 0) = \begin{cases} 0.5 \cos(\pi(2x + 4)) + 0.5, & x \in [-1.5, -0.5] \\ 0, & \text{otherwise} \end{cases} \tag{38}$$

In Figs. 10, 11, and 12, we show snapshots of solutions with different $\alpha(x, t)$ functions. We observe a vertical suppress and horizontal spread of the profile due to the memory effect.
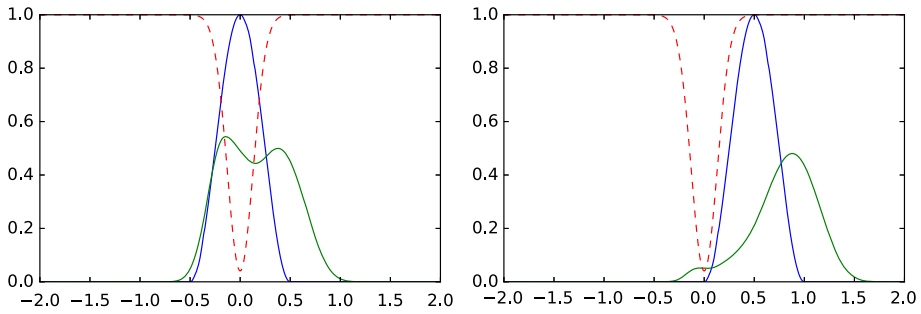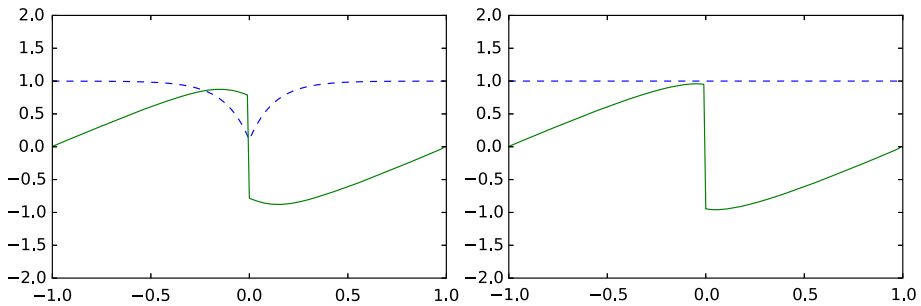
Fig. 12 *Left*: solution (*green line*) at $T = 1$ with $\lambda = 5.3$ (*dash line*) and exact solution with $T = 1, \alpha = 1$ (*blue line*). *Right*: solution (*green line*) at $T = 1.5$ with $\lambda = 5.3$ (*dash line*) and exact solution with $T = 1.5$, $\alpha = 1$ (*blue line*). *Dash line* is $\alpha(x, \lambda)$ with corresponding λs (Color figure online)
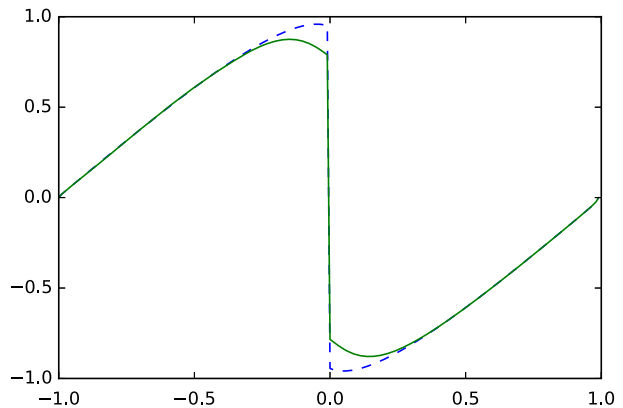

Fig. 13 *Left*: solution (*solid line*) at $T = 0.5$ with $\alpha = \alpha_1(x, t)$ (*dash line*), $\Delta t = \Delta x = 0.01$. *Right*: solution (*solid line*) at $T = 0.5$ with $\alpha = 1$, $\Delta t = \Delta x = 0.01$ by using the implicit upwind method with fast sweeping method [16]

Fig. 14 Solution (*green solid line*) at $T = 0.5$ with $\alpha = \alpha_1(x, t)$ and solution (*blue dash line*) at $T = 0.5$ with $\alpha = 1$, $\Delta t = \Delta x = 0.01$ by using the implicit upwind method with fast sweeping method [16] (Color figure online)

Finally, we consider the Burgers' Eq. (36) with where $\alpha_1(x, t) = 1 - 0.9 \exp(-8|x| - 7000(t - 0.8)^{12})$ and $\alpha_2(x, t) = 1$, which is the same example as in Karniadakis' paper [20]. However, in their paper, they got a solution with Gibbs phenomenon at the discontinuity point due to the pseudo-spectral method they use for spatial derivative. Here, our results are free of oscillation and also the damping effect can be observed. See Figs. 13 and 14.

References

1. Allen, M., Caffarelli, L., Vasseur, A.: Porous medium flow with both a fractional potential pressure and fractional time derivative. arXiv preprint (2015)
2. Cao, J., Xu, C.: A high order schema for the numerical solution of the fractional ordinary differential equations. J. Comput. Phys. **238**, 154–168 (2013)
3. Caputo, M.: Diffusion of fluids in porous media with memory. Geothermics **28**(1), 113–130 (1999)
4. del Castillo-Negrete, D., Carreras, B.A., Lynch, V.E.: Fractional diffusion in plasma turbulence. Phys. Plasmas **11**(8), 3854–3864 (2004)
5. del Castillo-Negrete, D., Carreras, B.A., Lynch, V.E.: Nondiffusive transport in plasma turbulence: a fractional diffusion approach. Phys. Rev. Lett. **94**(6), 065003 (2005)
6. Gao, G., Sun, H.: Three-point combined compact alternating direction implicit difference schemes for two-dimensional time-fractional advection-diffusion equations. Commun. Comput. Phys. **17**(02), 487–509 (2015)
7. Kumar, P., Agrawal, O.P.: An approximate method for numerical solution of fractional differential equations. Signal Process. **86**(10), 2602–2610 (2006). Special Section: Fractional Calculus Applications in Signals and Systems
8. Langlands, T.A.M., Henry, B.I.: The accuracy and stability of an implicit solution method for the fractional diffusion equation. J. Comput. Phys. **205**(2), 719–736 (2005)
9. Lin, R., Liu, F.: Fractional high order methods for the nonlinear fractional ordinary differential equation. Nonlinear Anal. **66**(4), 856–869 (2007)
10. Lin, Y., Xu, C.: Finite difference/spectral approximations for the time-fractional diffusion equation. J. Comput. Phys. **225**(2), 1533–1552 (2007)
11. Liu, F., Zhuang, P., Anh, V., Turner, I., Burrage, K.: Stability and convergence of the difference methods for the space-time fractional advection-diffusion equation. Appl. Math. Comput. **191**(1), 12–20 (2007)
12. Lv, C., Xu, C.: Improved error estimates of a finite difference/spectral method for time-fractional diffusion equations. Int. J. Numer. Anal. Model. **12**(2), 384–400 (2015)
13. Metzler, R., Klafter, J.: The random walk's guide to anomalous diffusion: a fractional dynamics approach. Phys. Rep. **229**(1), 1–77 (2000)
14. Shu, C.: A numerical method for systems of conservation laws of mixed type admitting hyperbolic flux splitting. J. Comput. Phys. **100**(2), 424–429 (1992)
15. Sun, Z., Wu, X.: A fully discrete difference scheme for a diffusion-wave system. Appl. Numer. Math. **56**(2), 193–209 (2006)
16. Tsai, R., Cheng, L., Osher, S., Zhao, H.: Fast sweeping algorithms for a class of hamilton-jacobi equations. SIAM J. Numer. Anal. **41**(2), 673–694 (2003)
17. Wang, C., Liu, J.: Positivity property of second-order flux-splitting schemes for the compressible euler equations. Discret. Contin. Dyn. Syst. Ser. B **3**(2), 201–228 (2003)
18. Xu, D.: Alternating direction implicit galerkin finite element method for the two-dimensional time fractional evolution equation. Numer. Math. Theory Methods Appl. **7**(01), 41–57 (2014)
19. Zaslavsky, G.M.: Chaos, fractional kinetics, and anomalous transport. Phys. Rep. **371**(6), 461–580 (2002)
20. Zayernouri, M., Karniadakis, G.E.: Fractional Sturm-Liouville eigen-problems: theory and numerical approximation. J. Comput. Phys. **252**(C), 495–517 (2013)
21. Zhang, H., Liu, F., Phanikumar, M.S., Meerschaert, M.M.: A novel numerical method for the time variable fractional order mobile-immobile advection-dispersion model. Comput. Math. Appl. **66**(5), 693–701 (2013)
22. Zhao, X., Sun, Z., Karniadakis, G.E.: Second-order approximations for variable order fractional derivatives: algorithms and applications. J. Comput. Phys. **293**(C), 184–200 (2015)