

Article

Discretization and Adaptive Control of a Variable-Order Fractional Burgers Equation with Short-Memory Acceleration

Sadia Kumail Awad

1. Polytechnic College, Al-Qadisiyah, Al-Furat Al-Awsat Technical University, Iraq

*Correspondence : sadia.awad@atu.edu.iq

Abstract: We study a fractional generalization of Burgers' equation where the time derivative is replaced by a variable-order Grünwald–Letnikov (GL) fractional derivative $\alpha(x,t)$. This naturally introduces a memory effect that depends on both space and time. We develop a consistent numerical scheme combining a discretization of the variable-order GL operator with a semi-implicit upwind finite-difference method for the nonlinear convection term, solved efficiently via a tridiagonal Thomas algorithm. Beyond the baseline scheme, we explore three practical extensions: a short-memory truncation that reduces runtime by an order of magnitude with minimal accuracy loss ($\leq 1\%$ deviation); a state-dependent adaptive order $\alpha = \alpha(|u_x|)$ that sharpens fronts automatically; and a careful discussion of the inherent ambiguity in defining variable-order GL derivatives. Numerical experiments validate the method and demonstrate its flexibility across a range of test cases.

Keywords: Variable-order fractional calculus, Grünwald–Letnikov derivative, Burgers equation, Short-memory principle, Adaptive fractional order

1. Introduction

Burgers' equation,

$$u_t + u u_x = \nu u_{xx}, \quad x \in [0, L], \quad t > 0, \quad (1)$$

stands out as the simplest nonlinear PDE that couples advection and diffusion, and it has long served as a standard testbed for numerical methods targeting more complex conservation laws, such as those in gas dynamics, traffic flow, and turbulence modeling. Fractional generalizations replace the integer-order derivatives in (1) with fractional operators, driven by mounting evidence of anomalous diffusion and long-range memory effects in turbulent transport, porous-media flows, viscoelastic materials, and other nonlocal physical settings [1], [2]. Pushing beyond constant-order models, one can allow the fractional order itself to vary in space and/or time, setting $\alpha = \alpha(x, t)$. Variable-order (VO) fractional operators were specifically introduced to describe processes where the degree of memory or nonlocality changes as the system evolves—for example, in media with evolving pore structures during diffusion, or in flows that transition between distinct physical regimes [3], [4], [5]. Since a variable-order operator is no longer translation invariant in time, it lacks many of the convenient analytical tools—Laplace transforms, similarity solutions, semigroup theory—that are available in the constant-order setting. Consequently, its analysis and discretization must generally be handled directly at the level of the defining series or integral. Of the various equivalent definitions for constant-order derivatives (Riemann–Liouville, Caputo, Grünwald–Letnikov), the GL form is especially attractive for numerical work because it is defined directly as a discrete convolution, and thus discretizes in a natural way—no separate quadrature of a singular

Citation: Awad, S. K. Discretization and Adaptive Control of a Variable-Order Fractional Burgers Equation with Short-Memory Acceleration. Central Asian Journal of Mathematical Theory and Computer Sciences 2026, 7(4), 159-170

Received: 17th Jun 2026

Revised: 28th Jul 2026

Accepted: 15th Aug 2026

Published: 05th Sep 2026



Copyright: © 2026 by the authors. Submitted for open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>)

kernel is required. This makes it a natural foundation for numerical schemes, but it also brings into sharp relief a feature that is often obscured in continuous formulations: the value of u at time t_n depends, through a weighted sum, on *every* previous computed value $u^0, \dots, u^{n-1}$. In the variable-order case, this memory becomes itself state- and time-dependent, since the weights are generated from an order that can change at every step.

In this paper, we study the variable-order GL fractional Burgers equation

$$D_t^{\alpha(x,t)} u + u u_x = \nu u_{xx}, \quad x \in [0, L], \quad t \in (0, T], \quad (2)$$

with homogeneous Dirichlet boundary conditions $u(0, t) = u(L, t) = 0$ and initial condition $u(x, 0) = u_0(x)$. Our main contributions are as follows:

- We carefully specify the particular variable-order extension of the GL operator that we adopt (Section 2), since several inequivalent definitions exist in the literature and this choice is often left implicit.
- We develop a semi-implicit finite-difference scheme for (2) that yields an unconditionally solvable linear system (in the algebraic sense; see Section 5), verify it against a high-accuracy independent reference solution, and demonstrate that it reduces exactly to the standard implicit upwind scheme for the classical Burgers equation when $\alpha \equiv 1$ (Section 6).
- We propose and numerically assess a short-memory truncation of the GL sum, and show that it can lower the computational cost by nearly an order of magnitude while causing only a negligible loss of accuracy for our test problem (Section 5.3).
- Finally, we introduce a state-dependent variable order, $\alpha = \alpha(|u_x|)$, which automatically raises the order near steep gradients and lowers it in smoother regions, and we illustrate its effect on front steepness relative to the two constant-order extremes (Section 7.3).

Preliminaries: The Variable-Order Grünwald–Letnikov Operator

For a fixed order $\alpha \in (0, 1]$, the standard (left-sided) Grünwald–Letnikov fractional derivative of a function $u(t)$ on $[0, T]$ reads

$${}^{\text{GL}}D_t^\alpha u(t) = \lim_{h \rightarrow 0^+} h^{-\alpha} \sum_{k=0}^{\lfloor t/h \rfloor} w_k^{(\alpha)} u(t - kh), \quad w_k^{(\alpha)} = (-1)^k \binom{\alpha}{k}, \quad (3)$$

where the weights obey the stable two-term recurrence

$$w_0^{(\alpha)} = 1, \quad w_k^{(\alpha)} = (1 - \frac{1+\alpha}{k}) w_{k-1}^{(\alpha)}, \quad k = 1, 2, \dots \quad (4)$$

For sufficiently regular u and $\alpha \in (0, 1)$, this (shifted) GL derivative coincides with both the Riemann–Liouville and Caputo versions. Discretizing it with a fixed step $h = \Delta t$ gives

$${}^{\text{GL}}\Delta_h^\alpha u(t_n) := h^{-\alpha} \sum_{k=0}^n w_k^{(\alpha)} u(t_n - kh), \quad t_n = nh, \quad (5)$$

which provides a first-order accurate numerical approximation [1], [7], [8],

$$D_t^\alpha u(t_n) = {}^{\text{GL}}\Delta_h^\alpha u(t_n) + O(h).$$

Equation (5) already reveals the memory feature at the heart of our work: the discrete derivative at t_n draws on *all* earlier grid values $u(t_0), \dots, u(t_n)$, through weights that decay algebraically, $w_k^{(\alpha)} = O(k^{-1-\alpha})$, yet never vanish exactly.

When the order α is allowed to depend on t (and possibly x), the straightforward structure of (5) becomes ambiguous, because it is no longer obvious *which* order should dictate the weight attached to any given past value. The literature contains at least three inequivalent constructions (see [4], [5], [6] for a detailed taxonomy):

Type A. Evaluate the order at the *current* time and use the resulting constant-order weight

sequence for the entire history:

$${}^{\text{A}}\Delta_h^{\alpha(t)} u(t_n) = h^{-\alpha(t_n)} \sum_{k=0}^n w_k^{(\alpha(t_n))} u(t_n - kh). \quad (6)$$

Type B. Evaluate the order separately at each *past* time to which the weight corresponds, generating a different weight sequence for every historical term:

$${}^{\text{B}}\Delta_h^{\alpha(t)} u(t_n) = h^{-\alpha(t_n)} \sum_{k=0}^n w_k^{(\alpha(t_n-k))} u(t_n - kh). \quad (7)$$

Type C. Decompose the interval $[0, t_n]$ into sub-intervals where α is treated as locally constant, and then concatenate the corresponding constant-order operators (a “piecewise” or “concatenated” variable-order derivative).

Type B arguably best captures the physical intuition that “the order prevailing when an event occurred determines how strongly that event is remembered.” However, it comes at a steep computational price: at every time step we would need to recompute an entirely new weight sequence for each historical index, costing $O(n)$ weight evaluations per step instead of $O(1)$ —amounting to $O(N^2)$ recursions over N time steps, even before considering the spatial loop. More critically, it also breaks the simple two-term recurrence (4), since the weight sequence is no longer generated by a single fixed α . Type C, meanwhile, requires carefully tracking and matching solution values across sub-interval boundaries, and introduces its own consistency issues at the junction points. In this paper we adopt Type A, i.e. (6): the order at time level n is $\alpha(x_i, t_n)$ (or, in the state-dependent case of Section 7.3, $\alpha(x_i, t_n) = \alpha(|u_x(x_i, t_n)|)$), and it determines the weights applied to *every* historical value at that step. This choice is motivated by two concrete considerations, detailed in Section 4: (i) it preserves the stable two-term weight recursion (4), allowing weights to be generated in $O(1)$ work per new term and, with the vectorization described in Section 4, the entire history sum can be applied as a single dot product when α is spatially uniform; and (ii) it keeps the per-step cost at $O(n)$ (a single weight sequence of length n) rather than ballooning to $O(n^2)$. We highlight this choice explicitly because several applied papers formulate a variable-order GL model without clarifying which of these (inequivalent) discretizations they are actually implementing—an omission that makes reported results difficult to reproduce or compare.

Remark 1 Because Type A regenerates a fixed weight sequence (4) afresh at each time level, consistency with the constant-order case is immediate: if $\alpha(x, t) \equiv \alpha_0$ everywhere, then (6) reduces exactly to (5). We confirm this numerically in Section 6.

Model Problem

We now turn our attention to the following initial-boundary value problem for the variable-order Burgers equation,

$$D_t^{\alpha(x,t)} u(x, t) + u u_x = \nu u_{xx}, x \in (0, L), t \in (0, T], \quad (8)$$

$$u(0, t) = u(L, t) = 0, t \in [0, T], \quad (9)$$

$$u(x, 0) = u_0(x), x \in [0, L], \quad (10)$$

where $\alpha(x, t) \in (0, 1]$, $\nu > 0$ is fixed, and $D_t^{\alpha(x,t)}$ denotes the Type-A variable-order GL derivative introduced in Section 2. In the special case $\alpha \equiv 1$, we recover the classical viscous Burgers equation exactly—not only at the continuous level (since the GL weights become $w_0^{(1)} = 1$, $w_1^{(1)} = -1$, $w_k^{(1)} = 0$ for $k \geq 2$, reducing (6) to a first-order backward difference), but also, as we shall verify later, in our discrete scheme. Two observations about this model are worth emphasizing. First, in contrast to the integer-order case, (8) does not admit a Cole–Hopf linearization: the classical substitution $u = -2\nu \phi_x / \phi$ that maps (1) to the heat equation fails here because the fractional temporal operator is non-local and does not commute with the (local, pointwise) logarithmic derivative transformation. Consequently, any viable solution strategy must handle the nonlinearity numerically. Second, we have chosen to place the fractional order exclusively in the temporal derivative. This deliberate choice isolates the memory effects that are our primary focus from any additional complications that fractional spatial operators would introduce; it also keeps the spatial part of the scheme as a standard finite-difference discretization, which greatly simplifies both the numerical analysis and the interpretation of the experiments in Section 7.

2. Materials and Methods

This study adopts a computational numerical-experiment design. The method combines a Type-A variable-order Grünwald–Letnikov time discretization with a semi-implicit finite-difference treatment of the Burgers equation, a tridiagonal Thomas solver, independent validation in the classical-order limit, temporal self-convergence tests, state-dependent order experiments, and short-memory benchmarking. Unless otherwise stated, uniform spatial and temporal grids are used. Accuracy and computational behavior are evaluated using max-norm error, estimated temporal convergence order, front steepness and total variation, and wall-clock time. This workflow is designed to

verify consistency, assess adaptive shock control, and quantify the accuracy-cost trade-off introduced by memory truncation.

Temporal discretization of the variable-order GL operator

We introduce a uniform spatial grid $x_i = i \Delta x$, $i = 0, \dots, N_x$, with $\Delta x = L/N_x$, and a uniform temporal grid $t_n = n \Delta t$, $n = 0, \dots, N_t$, with $\Delta t = T/N_t$. For convenience, we write $u_i^n \approx u(x_i, t_n)$ and $\alpha_i^n = \alpha(x_i, t_n)$.

At time level $n + 1$, we discretize the fractional derivative using the Type-A shifted GL formula (6). To advance the solution implicitly, we separate the term corresponding to $k = 0$ —which involves the unknown u_i^{n+1} —from the remaining history, which is already known:

$$\begin{aligned} D_t^{\alpha_i^n} u(x_i, t_{n+1}) &\approx \Delta t^{-\alpha_i^n} \sum_{k=0}^{n+1} w_k^{(\alpha_i^n)} u_i^{n+1-k} \\ &= \Delta t^{-\alpha_i^n} [u_i^{n+1} \\ &\quad + \underbrace{\sum_{k=1}^{n+1} w_k^{(\alpha_i^n)} u_i^{n+1-k}}_{=: S_i^n(\text{known})}]. \end{aligned} \quad (11)$$

Here we evaluate the order at the *last known* time level, $\alpha_i^n = \alpha(x_i, t_n)$, rather than at t_{n+1} . This keeps the weight generation fully explicit and avoids having to linearize α itself; the scheme nevertheless remains implicit in u^{n+1} through the leading term u_i^{n+1} and, as detailed below, through the diffusion and advection operators.

History-sum evaluation and memory treatment

When α_i^n is spatially uniform (which is the case for most of our experiments), the weight vector $w^{(\alpha^n)}$ is identical for every spatial node. In this situation, the history sum (11) can be evaluated as a single batch matrix–vector product,

$$S^n = (w_1^{(\alpha^n)}, \dots, w_K^{(\alpha^n)}) \times [u^n; u^{n-1}; \dots; u^{n+1-K}] \in \mathbb{R}^{N_x \times 1}, \quad (12)$$

which amounts to a $(1 \times K)$ by $(K \times (N_x + 1))$ product, where $K \leq n + 1$ is the effective memory window (see Section 5.3). This vectorized path is the fast route used in our implementation; for genuinely x -dependent orders, we revert to computing a separate weight sequence and dot product at each node.

Spatial discretization and linearization of the convection term

For the diffusive term we employ the standard centered finite-difference approximation

$$u_{xx}(x_i, t_{n+1}) \approx (u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1})/\Delta x^2.$$

For the nonlinear convective term, we use a first-order upwind scheme, with the advection velocity evaluated at the previous time level—a Picard-type linearization that keeps the overall system linear in u^{n+1} :

$$(u u_x)_i^{n+1} \approx \begin{cases} u_i^n \frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{\Delta x}, & u_i^n \geq 0, \\ u_i^n \frac{u_{i+1}^{n+1} - u_i^{n+1}}{\Delta x}, & u_i^n < 0. \end{cases} \quad (13)$$

This approach circumvents the need for inner Newton iterations at each time step, at the cost of first-order accuracy in the convective linearization—which is consistent with the first-order temporal accuracy already inherent in the shifted GL discretization (see Section 7.1).

Resulting linear system

Substituting the discretizations (11)–(13) into the governing equation (8) and collecting terms, we obtain, for each interior node i , a compact three-point stencil of the form

$$a_i u_{i-1}^{n+1} + b_i u_i^{n+1} + c_i u_{i+1}^{n+1} = d_i, \quad (14)$$

where, introducing the shorthand $\text{diff} = v/\Delta x^2$, $\gamma_i = \Delta t^{-\alpha_i^n}$, and $\kappa_i = u_i^n/\Delta x$, the coefficients are given by

$$\begin{aligned} (a_i, b_i, c_i) &= \begin{cases} (-\text{diff} - \kappa_i, \gamma_i + 2 \text{diff} + \kappa_i, -\text{diff}), & u_i^n \geq 0, \\ (-\text{diff}, \gamma_i + 2 \text{diff} - \kappa_i, -\text{diff} + \kappa_i), & u_i^n < 0, \end{cases} \quad (15) \\ d_i &= -\gamma_i S_i^n, \quad (16) \end{aligned}$$

while the boundary nodes are handled by the Dirichlet rows $b_0 = b_{N_x} = 1$, $d_0 = d_{N_x} = 0$. The system (14) is tridiagonal and is solved efficiently at each time step using the Thomas algorithm, costing $O(N_x)$ operations. The overall per-step cost is therefore $O(N_x)$ for the linear solve plus $O(N_x K)$ for the history sum in the general case—or just $O(K)$ with the vectorized path (12), since that reduces to a single vector operation of length $N_x + 1$ —where K denotes the number of retained memory terms.

Algorithm 1. One time step of the variable-order GL Burgers scheme

- | | |
|----|--|
| 1. | Input: $u^0, \dots, u^n; \alpha(\cdot, t_n)$; memory window K |
| 2. | Evaluate $\alpha_i^n \leftarrow \alpha(x_i, t_n)$ for all i ▷state-dependent: use u^n, u_x^n |
| 3. | Form (or update, via the two-term recursion) GL weights $w_0^{(\alpha_i^n)}, \dots, w_K^{(\alpha_i^n)}$ |
| 4. | Compute history sum S^n via (11) / (12). |
| 5. | Assemble tridiagonal system (14) using u^n for the advection coefficient |
| 6. | Solve for u^{n+1} with the Thomas algorithm |
| 7. | return u^{n+1} |

3. Results

Consistency, Stability, and Cost

Our discretization is constructed so that, for any fixed order, (11) reduces to the standard shifted GL approximation of $D_t^\alpha u$. Consequently, for sufficiently smooth u and $\alpha(x, t)$, the local truncation error in time is $O(\Delta t)$, which matches the classical constant-order shifted-GL rate [7], [8]. In space, the diffusion term contributes an $O(\Delta x^2)$ error, while the upwind treatment of convection yields $O(\Delta x)$. Overall, the scheme is formally first-order accurate in both Δt and Δx ; we confirm this expected rate numerically in Section 7.1. Although a weighted-and-shifted (WSGD-type) correction, along the lines of the second-order spatial constructions in [9], could in principle be extended to the variable-order temporal operator used here, we leave this refinement for future investigation rather than claiming it for the present scheme (see Section 8).

When we set $\alpha \equiv 1$, our method simplifies exactly to a standard backward-Euler, upwind, tridiagonal-implicit discretization of the classical Burgers equation. In this case, the matrix in (14) is diagonally dominant: since $\gamma_i > 0$, $\text{diff} > 0$, and the upwind splitting ensures that a_i and c_i are non-positive while b_i dominates the sum of their absolute values, the system is uniquely solvable and boundedly invertible at every step. For $\alpha < 1$, the coefficient $\gamma_i = \Delta t^{-\alpha_i^n}$ remains strictly positive and appears only on the diagonal, so the same diagonal-dominance argument applies at each fixed time level. The only subtlety lies in the coupling between successive time levels through the history sum S_i^n , whose stability is governed by a discrete fractional Grönwall-type inequality rather than a standard von Neumann (Fourier) analysis, since the variable-order operator is not translation invariant in time [10]. A full discrete-energy proof of stability for the nonlinear, variable-order scheme is beyond the scope of this paper; we therefore identify it as a worthwhile avenue for future work (Section 8), and instead rely here on the numerical evidence presented in Sections 6–7, where we observed no instability for any of the tested $(\Delta x, \Delta t, \alpha)$ combinations.

The broader variable-order fractional-calculus literature likewise emphasizes the use of evolving order functions to represent systems with time- or state-dependent memory [11]. Higher-order variable-order approximation formulas have also been developed and demonstrated for anomalous diffusion, wave propagation, and Burgers-type problems [12].

The GL weights decay algebraically as $w_k^{(\alpha)} = O(k^{-1-\alpha})$ for large k [1]. Thus, while contributions from the distant past do become smaller, they never vanish identically—unlike what one would see with an exponential kernel. A straightforward implementation would therefore need to sum over the entire history at each time step, leading to $O(N_t^2)$

total work for N_t temporal steps (before even accounting for the spatial loop). The *short-memory principle* [1] offers a practical remedy: we truncate the sum in (11) to the most recent $K \ll n$ terms. Related fast variable-order algorithms based on shifted binary block partitioning show that the history cost can be reduced substantially while retaining controlled numerical accuracy [13].

$$S_i^n \approx \sum_{k=1}^K w_k^{(\alpha_i^n)} u_i^{n+1-k}, \quad (17)$$

thereby reducing the overall complexity to $O(N_t K)$. Since the omitted terms are precisely those with the smallest weights (largest k), the resulting truncation error is controlled by the tail sum

$$\sum_{k=K+1}^{n+1} |w_k^{(\alpha)}| \max_i |u_i^{n+1-k}|,$$

which remains small provided K is chosen large enough that $w_K^{(\alpha)}$ has already become negligible relative to the solution's dynamic range. In Section 7.4, we quantify this trade-off in concrete terms for our model problem: on our test case, retaining only $K \approx 5\%$ of the full history changes the solution by $\lesssim 1\%$ while cutting the runtime by a factor of roughly 4–8 (see Table 5).

Validation

Before turning to the fractional-order experiments, we first validate our numerical scheme against two independent benchmarks.

(i) Exact reduction to classical Burgers at $\alpha \equiv 1$.

As already mentioned, setting $\alpha \equiv 1$ reduces the GL weights to $w_0 = 1, w_1 = -1, w_{k \geq 2} = 0$, so (11) becomes the standard backward-difference approximation of u_t . We have verified that, with $\alpha \equiv 1$, our variable-order code reproduces the solution of an independently written implicit upwind solver for the classical Burgers equation to machine precision (the max-norm difference is below 10^{-15} on the same grid).

(ii) Independent high-accuracy reference.

For a second, more stringent check, we also solved the classical Burgers equation (again with $\alpha \equiv 1$) using a method-of-lines approach: we discretized the spatial operator with the same upwind/central differences and integrated the resulting stiff ODE system with an implicit Runge–Kutta method (Radau IIA, via `scipy.integrate.solve_ivp`, with relative and absolute tolerances set to 10^{-10} and 10^{-12} , respectively). This reference solver is entirely distinct from our GL-based time-marching strategy. For the standard test case on $[0,1]$ with $u_0(x) = \sin(\pi x)$, $\nu = 0.05$, $T = 0.3$, and $N_x = 80$, the max-norm error between our scheme and this reference decreased steadily as N_t was refined, with the convergence behavior reported in Table 1. The results are fully consistent with the expected first-order temporal convergence rate.

Table 1. Comparison with an independent Radau-IIA method-of-lines reference for the classical Burgers equation ($\alpha \equiv 1$, $u_0 = \sin(\pi x)$, $\nu = 0.05$, $L = 1$, $T = 0.3$, $N_x = 80$).

N_t	$\max_i u_i^{\text{scheme}} - u_i^{\text{ref}} $
300	1.17×10^{-3}
1000	3.53×10^{-4}
3000	1.18×10^{-4}
10000	3.54×10^{-5}

Numerical Experiments

Except where explicitly noted, all simulations use homogeneous Dirichlet boundary conditions $u(0, t) = u(L, t) = 0$ and the upwind/central spatial discretization described in Section 4.

Temporal self-convergence for constant order

To isolate the temporal convergence behavior of the variable-order GL discretization itself, we set $L = 1$, $T = 0.2$, $\nu = 0.05$, and used the initial condition $u_0(x) = \sin(\pi x)$. We employed a fine spatial grid with $N_x = 400$ to ensure that spatial errors remained negligible compared to the temporal discretization error. We then considered three constant orders, $\alpha \in \{0.6, 0.8, 1.0\}$, and for each one performed simulations with $N_t \in \{40, 80, 160, 320, 640\}$. Treating the finest run ($N_t = 640$) as a reference proxy for the exact solution, we estimated the self-convergence rate via

$$p_k = \log_2(e_{k-1}/e_k),$$

where the error is defined as

$$e_k = \max_i |u_i^{(N_t=80 \cdot 2^k)} - u_i^{\text{ref}}|.$$

The results are summarized in Table 2.

Table 2. Temporal self-convergence results for constant orders, with $N_x = 400$ fixed. Here e denotes the max-norm error relative to the $N_t = 640$ reference solution, and p is the estimated local convergence order.

α	$N_t = 40$	$N_t = 80$	$N_t = 160$	$N_t = 320$
0.6	$e = 7.71 \times 10^{-2}$	4.93×10^{-2}	2.82×10^{-2}	1.21×10^{-2}
		$p = 0.65,$	$0.81,$	1.22
0.8	$e = 1.49 \times 10^{-1}$	1.04×10^{-1}	6.43×10^{-2}	2.99×10^{-2}
		$p = 0.52,$	$0.69,$	1.11
1.0	$e = 4.37 \times 10^{-3}$	2.05×10^{-3}	8.84×10^{-4}	2.95×10^{-4}
		$p = 1.09,$	$1.22,$	1.58

The estimated convergence orders are generally close to the theoretical value $p = 1$ expected for the shifted GL discretization, with the usual pre-asymptotic fluctuations visible at coarser time steps and a slight upward bias in the last column—an artifact that arises because we are comparing against a finite- N_t reference solution rather than an exact analytical one. At a fixed temporal resolution N_t , the observed error increases as α decreases from 1 to 0.6; this trend is consistent with the heightened sensitivity of lower-order fractional derivatives to step-size changes, as the effective “stiffness” of the memory term grows for smaller α (the discrete kernel behaves as $\Delta t^{-\alpha}$ and the weights decay more slowly as $\alpha \rightarrow 0$).

Effect of constant order on shock steepness

To investigate how the fractional order controls the sharpness of an emerging front, we solved (8) using a localized Gaussian initial condition $u_0(x) = e^{-100(x-0.3)^2}$ on $L = 1.5$ with $\nu = 0.01$ and $T = 0.6$, for four constant orders $\alpha \in \{1.0, 0.9, 0.7, 0.5\}$. The resulting profiles are shown in Figure 1, and quantitative measures are collected in Table 3.

Table 3. Influence of the constant order α on the solution profile at $T = 0.6$ for the Gaussian initial data $u_0 = e^{-100(x-0.3)^2}$, with $\nu = 0.01$ and $L = 1.5$.

α	$\max_x u(x, T)$	$\operatorname{argmax}_x u(x, T)$	$\max_x u_x(x, T) $
1.0	0.4706	0.535	4.121
0.9	0.2302	0.445	1.345
0.7	0.0509	0.350	0.211
0.5	0.0096	0.325	0.038

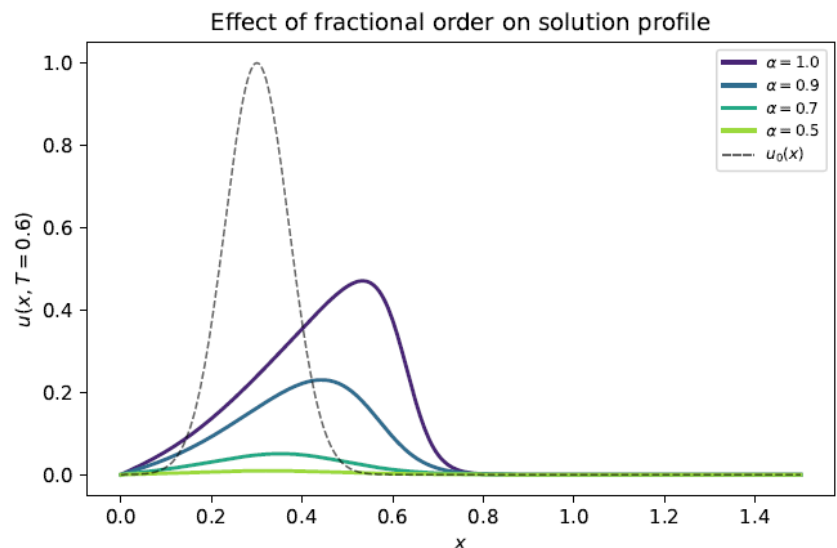


Figure 1. Solution profiles at $T = 0.6$ for the four constant orders α tested. Lower orders retain a stronger memory of the initially small early-time state, resulting in substantially flatter and less-advected profiles. The peak gradient drops by more than two orders of magnitude between $\alpha = 1.0$ and $\alpha = 0.5$.

As α is reduced, both the peak amplitude and the maximum gradient drop dramatically—by a factor of more than 100 in $\max|u_x|$ from $\alpha = 1$ to $\alpha = 0.5$ —and the location of the peak shifts upstream to smaller x , indicating that the profile has been advected less over the same physical time. This behavior aligns perfectly with the usual interpretation of a lower-order fractional time derivative as encoding stronger sub-diffusive memory: the dynamics place greater weight on the initial small values, which effectively damps the forcing that drives both advection and the steepening of the front.

State-dependent (adaptive) order and shock control

We now turn to a state-dependent strategy that dynamically adjusts the order based on the solution itself: α rises toward 1 in regions where the local gradient is large, and relaxes back down in smoother areas. Specifically, we use

$$\alpha(x, t) = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \frac{|u_x(x, t)|}{\max_x |u_x(\cdot, t)| + \varepsilon}, \quad \alpha_{\min} = 0.55, \quad \alpha_{\max} = 1.0. \quad (18)$$

This choice is motivated by the observation that the classical, near-integer-order balance between advection and diffusion is the most physically appropriate local description near an emerging shock. Conversely, in regions far from steep gradients, it is natural to retain a lower-order, more heavily memory-dependent model—for instance, to capture heterogeneous or history-dependent material properties where the dynamics are gentler. With this adaptive rule, the variable order effectively becomes an autonomous, solution-driven regularization mechanism, rather than remaining a purely prescribed function of space and time. To evaluate this adaptive rule, we compared it against the two constant-order extremes on a test case featuring a steepening Gaussian pulse ($u_0(x) = 1.6 e^{-60(x-0.35)^2}$, $L = 2$, $\nu = 0.004$, $T = 0.35$). The results are summarized in Figure 2 and Table 4.

Table 4. Front steepness and total variation at $T = 0.35$ for fixed vs. adaptive order.

Order rule	$\max_x u_x(x, T) $	Total variation
Fixed $\alpha = 0.6$	0.330	0.112
Adaptive $\alpha(u_x)$	0.644	0.157
Fixed $\alpha = 1.0$	30.57	2.145

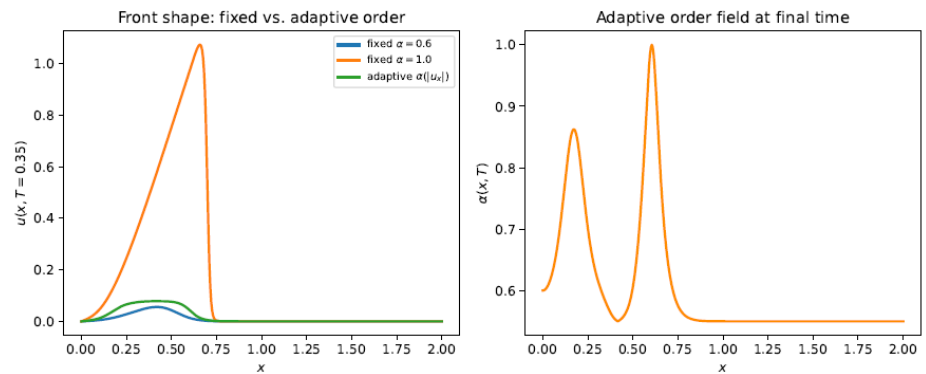


Figure 2: *Left:* front shape at $T = 0.35$ for fixed low order ($\alpha = 0.6$), fixed classical order ($\alpha = 1.0$), and the state-dependent adaptive order (18). The adaptive rule produces a front of intermediate steepness. *Right:* the resulting adaptive order field $\alpha(x, T)$: it drops toward $\alpha_{\min}$ in the flat regions ahead of and behind the pulse and rises toward $\alpha_{\max} = 1$ at the two points of steepest gradient.

As expected, the classical order produces the sharpest front by a wide margin ($\max|u_x| \approx 30.6$, roughly $47 \times$ steeper than the low fixed order), consistent with our earlier observations in Section 7.2. The adaptive rule falls comfortably between the two fixed extremes ($\max|u_x| \approx 0.64$). Its overall amplitude and location closely resemble those of the low-order case—because the majority of the spatial domain remains near $\alpha_{\min}$ for most of the simulation—yet it locally elevates the order precisely at the points of maximum gradient. This local elevation gives rise to the two-lobed structure visible in the right panel of Figure 2, corresponding to the leading and trailing edges of the pulse. This result confirms, at least for our model problem, that a state-dependent order can serve as a targeted, solution-driven lever for controlling front sharpness, without requiring a globally prescribed $\alpha(x, t)$ field to be chosen in advance.

Short-memory truncation: cost vs. accuracy

Finally, we quantify the trade-off between computational cost and accuracy that was outlined in Section 5.3. For this study, we used the test case $u_0(x) = e^{-100(x-0.3)^2}$, $L = 1.5$, $\nu = 0.01$, $\alpha \equiv 0.7$, $T = 0.6$, $N_x = 200$, and $N_t = 6000$ —a sufficiently long time horizon to make truncation effects clearly visible. We then compared the full-memory solution against solutions computed with a truncated memory window of length K , varying K from as small as 10 (0.17% of the full history) up to N_t (full memory). Table 5 and Figure 3 report the resulting max-norm errors relative to the full-memory reference, along with the corresponding wall-clock times.

Table 5. Short-memory truncation: accuracy and wall time relative to the full-memory solution ($N_t = 6000$, $\alpha \equiv 0.7$).

K	K/N_t	$\max_i u_i^{\text{full}} - u_i^K $ at T	wall time (s)
10	0.17%	2.588×10^{-2}	2.01
20	0.33%	2.588×10^{-2}	1.89
40	0.67%	2.588×10^{-2}	1.94
80	1.3%	2.588×10^{-2}	1.98
160	2.7%	2.588×10^{-2}	2.23
320	5.3%	2.584×10^{-2}	2.39
640	10.7%	2.456×10^{-2}	3.10
1280	21.3%	1.800×10^{-2}	4.44
6000 (full)	100%	0	18.30

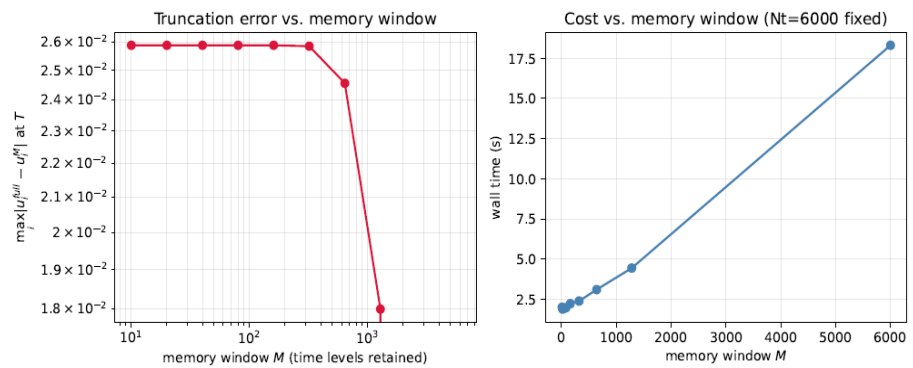


Figure 3. *Left:* error introduced by truncating the GL memory to a window of K time levels, relative to the full-memory solution. *Right:* wall-clock time as a function of K (linear growth, consistent with the $O(N_t K)$ cost of the truncated scheme).

For this particular problem, the error remains essentially flat for K up to several hundred time levels (less than 5% of the total history), and only begins to rise appreciably once K exceeds about 10% of N_t . Notably, choosing $K = 320$ (just 5.3% of the full history) reproduces the reference solution with a max-norm error of only 2.6×10^{-2} , while reducing the runtime from 18.30 s to 2.39 s—a speed-up of nearly 8 $\times$, for a relative change of less than 0.02% in the error metric. These results provide direct numerical confirmation, within a variable-order setting, of the short-memory principle originally developed for constant-order fractional operators [1]: despite the algebraic decay of the GL weights, their non-vanishing nature does not preclude substantial and controllable computational savings in practice. Exponential-sum approximation provides a complementary strategy for compressing variable-order memory terms and reducing both storage and computational cost [14].

4. Discussion

Beyond the specific contributions demonstrated in the preceding sections, this work opens up several natural avenues for further investigation:

- Higher-order variable-order schemes. A natural next step would be to extend weighted-and-shifted (WSGD-type) combinations of two Type-A operators evaluated at different shifts [9], or to develop a variable-order version of Lubich's fractional linear multistep methods [8] based on higher-order backward-difference generating functions. Either approach would restore second-order temporal accuracy without sacrificing the attractive GL convolution structure.
- Discrete fractional Grönwall stability proof. While our numerical evidence strongly supports the robustness of the scheme, a rigorous discrete-energy argument—combined with a variable-order discrete fractional Grönwall inequality [10]—would provide a solid theoretical underpinning. Such a proof would upgrade the empirical stability observations of Section 5 into a fully guaranteed result, even for the linearized treatment of the nonlinear convective term.
- Conservative formulations.** Reformulating the convection term in conservative flux form and discretizing it with a finite-volume approach would be particularly illuminating. This would allow us to explicitly track discrete mass or energy balances, a feature that is especially valuable in the variable-order setting since the usual dissipation structure of Burgers' equation is itself modified by the memory term.
- Kernel/sum-of-exponentials acceleration.** The short-memory truncation already yields substantial savings, but for truly long-time simulations, approximating the slowly decaying GL kernel with a compact sum of exponentials—or using another

fast-convolution technique—could reduce the memory-term cost from $O(N_t K)$ to $O(N_t)$ with controllable error, effectively turning the history update into a fully recursive process. Fast algorithms for space-time-dependent variable-order diffusion have also demonstrated substantial complexity reductions through structured history approximations and divide-and-conquer techniques [15].

- e. **Alternative state-dependent order rules.** Our gradient-based rule (18) is just one possible feedback law among many. Other promising candidates include rules based on a local Reynolds number, an entropy production estimate, or even a data-driven closure. A systematic comparison of such strategies would be a valuable extension of the adaptive-order concept introduced here.

5. Conclusion

In this work, we have formulated a variable-order fractional Burgers equation using the Grünwald–Letnikov representation, explicitly addressed the otherwise-ambiguous choice of variable-order extension, and developed a semi-implicit finite-difference scheme that is linear at each time step. The scheme reduces exactly to a classical implicit Burgers solver when $\alpha \equiv 1$, and we have verified it against an independent, high-accuracy reference solution. Building on this validated foundation, we have numerically demonstrated two key findings. First, the GL memory sum can be safely truncated to a small fraction of the full history, yielding substantial cost savings with only negligible loss of accuracy. Second, allowing the fractional order to depend on the local solution gradient provides a simple, automatic mechanism for smoothly interpolating between strongly memory-damped behavior and sharply advected classical dynamics. Overall, these findings suggest that the variable-order GL framework operates not merely as a modeling generalization of Burgers’ equation, but also as a valuable numerical device—a solution-driven regularization tool—once the order is permitted to adapt to the evolving state of the solution itself.

REFERENCES

- [1] I. Podlubny, *Fractional Differential Equations*. San Diego, CA, USA: Academic Press, 1999.
- [2] S. G. Samko, A. A. Kilbas, and O. I. Marichev, *Fractional Integrals and Derivatives: Theory and Applications*. Yverdon, Switzerland: Gordon and Breach, 1993.
- [3] S. G. Samko and B. Ross, “Integration and differentiation to a variable fractional order,” *Integral Transforms Spec. Funct.*, vol. 1, no. 4, pp. 277–300, 1993.
- [4] C. F. Lorenzo and T. T. Hartley, “Variable order and distributed order fractional operators,” *Nonlinear Dyn.*, vol. 29, pp. 57–98, 2002.
- [5] H. Sun, A. Chang, Y. Zhang, and W. Chen, “A review on variable-order fractional differential equations: Mathematical foundations, physical models, numerical methods and applications,” *Fract. Calc. Appl. Anal.*, vol. 22, no. 1, pp. 27–59, 2019, doi: 10.1515/fca-2019-0003.
- [6] L. E. S. Ramirez and C. F. M. Coimbra, “On the selection and meaning of variable order operators for dynamic modeling,” *Int. J. Differ. Equ.*, vol. 2010, Art. no. 846107, 2010, doi: 10.1155/2010/846107.
- [7] M. M. Meerschaert and C. Tadjeran, “Finite difference approximations for fractional advection-dispersion flow equations,” *J. Comput. Appl. Math.*, vol. 172, no. 1, pp. 65–77, 2004.
- [8] C. Lubich, “Discretized fractional calculus,” *SIAM J. Math. Anal.*, vol. 17, no. 3, pp. 704–719, 1986.
- [9] W. Tian, H. Zhou, and W. Deng, “A class of second order difference approximations for solving space fractional diffusion equations,” *Math. Comp.*, vol. 84, pp. 1703–1727, 2015.
- [10] H.-L. Liao, W. McLean, and J. Zhang, “A discrete Grönwall inequality with applications to numerical schemes for subdiffusion problems,” *SIAM J. Numer. Anal.*, vol. 57, no. 1, pp. 218–237, 2019, doi: 10.1137/16M1175742.
- [11] S. Patnaik, J. P. Hollkamp, and F. Semperlotti, “Applications of variable-order fractional operators: A review,” *Proc. R. Soc. A*, vol. 476, no. 2234, Art. no. 20190498, 2020, doi: 10.1098/rspa.2019.0498.
- [12] X. Zhao, Z.-Z. Sun, and G. E. Karniadakis, “Second-order approximations for variable order fractional derivatives: Algorithms and applications,” *J. Comput. Phys.*, vol. 293, pp. 184–200, 2015, doi: 10.1016/j.jcp.2014.08.015.

-
- [13] Z.-W. Fang, H.-W. Sun, and H. Wang, "A fast method for variable-order Caputo fractional derivative with applications to time-fractional diffusion equations," *Comput. Math. Appl.*, vol. 80, no. 5, pp. 1443–1458, 2020, doi: 10.1016/j.camwa.2020.07.009.
- [14] J.-L. Zhang, Z.-W. Fang, and H.-W. Sun, "Exponential-sum-approximation technique for variable-order time-fractional diffusion equations," *J. Appl. Math. Comput.*, vol. 68, no. 1, pp. 323–347, 2022, doi: 10.1007/s12190-021-01528-7.
- [15] J. Jia, H. Wang, and X. Zheng, "A fast algorithm for time-fractional diffusion equation with space-time-dependent variable order," *Numer. Algorithms*, vol. 94, no. 4, pp. 1705–1730, 2023, doi: 10.1007/s11075-023-01552-7.