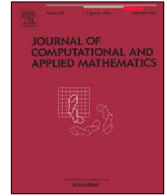




Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Numerical analysis of a time filtered scheme for a linear hyperbolic equation inspired by DNA transcription modeling

K. Boatman^a, L. Davis^b, F. Pahlevani^{a,*}, T. Susai Rajan^a^a Division of Science & Engineering, Penn State University-Abington, Abington, PA, United States of America^b Department of Mathematical Sciences, Montana State University, Bozeman, MT, United States of America

ARTICLE INFO

Article history:

Received 30 September 2022

Received in revised form 1 February 2023

Keywords:

Advection equation

Lighthill–Whitham–Richards model

RNA polymerases

Time filter

Transcription time

Upwind scheme

ABSTRACT

The focus of this paper is the development and analysis of a time filtering process for a linear hyperbolic equation motivated by the modeling of the transcription of ribosomal RNA in bacteria Davis et al. (2021). We demonstrate that a time filter technique can be combined with the classical upwind to produce a new explicit scheme with virtually no dissipation introduced by the method, and the filter can be implemented with minimal computational cost. The analysis shows that the filtered scheme gives the practitioner the ability to adjust the filtering so the dissipation can be made arbitrarily small over a range of time step choices. The analysis also indicates that the filtered scheme has a smaller local truncation error when compared to that of the original upwind method. A CFL condition for the new algorithm is derived, and it is shown to depend explicitly on the filter parameter. Numerical computations illustrate stability and convergence as well as dissipation and dispersion assessments of the filtered upwind scheme.

Published by Elsevier B.V.

1. Introduction

This paper describes one step in a larger research effort to develop and analyze an efficient simulation framework for a comprehensive model of ribosome abundance control in bacteria. The full model combines a mathematical description of three bio-polymerization processes: *transcription* of ribosomal RNA, transcription of mRNA of ribosomal proteins (r-proteins) and *translation* of r-proteins with a characterization of feedback mechanisms that regulate initiation rates, processing rates and abundance of key molecules.

1.1. Motivating application

The current version of the full mathematical model consists of a system of six nonlinear ordinary differential equations (ODEs) that describe the concentration of several protein building blocks, and the ODE system is coupled to three compartment models describing the processes of transcription and translation, see Fig. 1 for a graphical representation. Compartments 1 and 2 represent transcription processes, and compartment 3 represents translation. Both processes are fundamentally characterized by the motion of a molecular motor copying a segment of DNA or mRNA. The molecular motors are represented by the blue and red objects in the diagram. In transcription, the molecular motor is the RNA polymerase (RNAP), and in translation, the motor is the ribosome. Transcription begins when the RNAP binds to a promoter sequence on a gene (*initiation*) and then proceeds through the *elongation* process. During elongation the motor

* Corresponding author.

E-mail address: fxp10@psu.edu (F. Pahlevani).

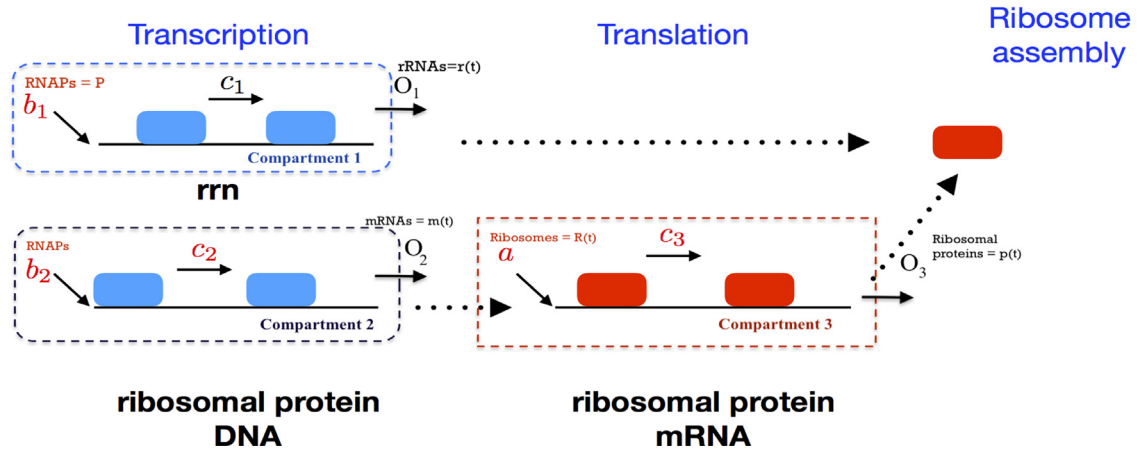


Fig. 1. Diagram describing the three compartments of the ribosome assembly process including transcription of the *rrn* operon, transcription of ribosomal protein and translation of the ribosomal mRNA. Ribosomes are then assembled from the ribosomal proteins. The blue objects represent RNAP on DNA strands, and the red objects represent ribosomes translating the mRNA strand.

travels along the coding region of the gene, reading the genetic information on the strand and generating a single-stranded copy. Transcription ends at the gene's *termination* region, where the motor releases the nascent copy of the strand and unbinds from it. Translation is a similar process with the ribosome playing the role of the molecular motor. The ribosome translates the genetic information contained in the mRNA strand. The speed at which these motors travel along the strand while reading the gene is referred to as the *elongation velocity*, and the parameters c_1 , c_2 and c_3 denote the elongation velocities in Fig. 1.

Each of the three compartment processes is described by a continuum model taking the form of a hyperbolic conservation law, and this system of three PDEs is weakly coupled as the output of one of the transcription compartments is related to the input of the translation compartment. Within the full model, the coupling between the PDE system and the ODE model occurs through the inflow and elongation velocity parameters labeled as b_1 , b_2 , a , c_1 , c_2 , and c_3 in Fig. 1. The values of these parameters are informed by the dependent variables in the ODE system. The outputs of the three compartments are also coupled to the ODE system through the production of rRNA, mRNA molecules and r-proteins.

For the purposes of the current work, we assume that each compartment model is of the same prototype. The mathematical model for a prototypical compartment describing either the transcription or the translation processes is the advection equation. This is a linear hyperbolic partial differential equation (PDE), and it has been used to describe traffic flow based on the well-known Lighthill–Whitham–Richards (LWR) continuum model for the case where the flow velocity is taken to be constant. Previous work using continuum models and parameter studies can be found in [1–3], and the work in [4] illustrates the mathematical modeling of the DNA transcription process that leads to the partial differential equation in (1)–(3), after non-dimensionalizing the variables and quantities involved in the equations. For the biological model, we assume that the density, $\rho(x, t)$, of molecular motors on the strand obeys the following PDE

$$\rho_t + \bar{v}\rho_x = 0, \quad 0 < x \leq 1, \quad t > 0 \quad (1)$$

$$\rho(x, 0) = \rho_0 \quad (2)$$

$$\rho(0, t) = \rho_l \quad (3)$$

Here $\bar{v} > 0$ is a constant elongation velocity, corresponding to the parameter values c_1 , c_2 and c_3 in the compartments of Fig. 1. The parameters labeled ρ_0 and ρ_l are initial and boundary conditions respectively, and the ρ_l parameter is associated with the parameters labeled b_1 , b_2 and a in Fig. 1. The variable $\rho(x, t)$ denotes the density of the molecular motors at the spatial point x along the strand at time t (roughly like an ensemble average). The initial density of RNAPs (or ribosomes) on the strand is determined by the ρ_0 parameter. The inflow boundary is at $x = 0$ and represents the initiation site or the promoter site for the gene under consideration. The outflow boundary is denoted as $x = 1$, and this corresponds to the gene's termination site.

The prototypical model in (1)–(3) is easily solved analytically and numerically, but here we focus on the situation where three of these PDEs with different elongation velocities will be coupled with a system of ODEs when one simulates the full model. It is well-known that low order explicit methods applied to (1)–(3), such as the classical upwind and Lax–Friedrichs schemes, exhibit significant dissipation unless the time step is coupled to the spatial grid size so that the Courant number is close to 1. When a computed solution exhibits a corner or a discontinuity, dissipative schemes require a large number of spatial grid points in order to accurately resolve the behavior. When considering simulation techniques for the PDE system alone, once a spatial discretization is selected, then a time step must be chosen that maintains the

stability of the calculation while minimizing the amount of dissipation and dispersion introduced by the PDE solver. Hence the time step must be chosen small enough to maintain stability but large enough so that the Courant number is close to 1. In order to simulate the full ribosome assembly model, one needs to couple the ODE system to the PDE system. Often the stability requirements of the nonlinear system of ODEs requires a time step that may be *too small*. That is, the time step that is small enough to maintain stable ODE simulations may introduce too much dissipation into the coupled PDE approximation scheme. The goal of this paper is to demonstrate that a simple filtering technique can be used to augment an existing PDE solver in order to satisfy the stability requirements of both systems while minimizing the amount of dissipation and dispersion introduced by the PDE solver and achieving the desired accuracy of the simulation.

This work demonstrates that a simple time filter technique can be combined with the first-order upwind scheme to control the numerical dissipation over a range of *small* time step values for a given choice of spatial grid size. That is, on a fixed spatial grid, careful choice of the filter parameter allows one to control the dissipation of the numerical scheme over a range of time step values that would typically introduce too much numerical dissipation into an upwind approximation. This allows for model simulations that use a coarse mesh for the spatial grid paired with a time step size that would be considered *too small* (relative to the grid size) to give accurate approximations when using the classical upwind scheme.

Although we do not address simulation efforts for the full model here, we stress that the time-filtered approach is chosen with full model simulation as the ultimate goal. The filtering technique allows for straightforward coupling between the ODE system and the three compartments governed by the PDEs. The algorithm allows the user to pair the PDE solver with a black-box ODE solver with very little additional computational effort and in a manner that controls dissipation. A discussion of the pairing of these two solver techniques is left for a subsequent work. In the current paper, the focus is the derivation and analysis of the time-filtered scheme.

One advantage of this approach is that existing (legacy) codes can be augmented with the time filter with minimal effort and computational costs. For the case described here, the filter requires one additional line of code, and one previous iteration of the solution approximation must be stored. One can use the filter technique in two ways. First, given a spatial grid size and a filter parameter value, one can choose the largest time step to maintain stability of the scheme. Or, given a spatial grid size and a time step value, one can choose a filter parameter so that the dissipation of the numerical scheme is very small when compared to using the same grid and time step pairing with the un-filtered scheme. Although it is the latter approach that we intend to use for the full ribosome assembly model, we focus on the former in the context of the numerical analysis of the filtered scheme for the current work.

1.2. Time filter background

For several decades time filters have been explored as computational tools for improving accuracy and controlling the influence of non-physical modes introduced by discretization techniques. The first version of a time filter was proposed by Robert in 1966 [5]. It was originally applied to the leapfrog scheme for weather forecasting using meteorological models. Later in 1972, an analysis by Asselin showed that the filter removes leapfrog's computational mode and damps out high frequency waves [6]. Williams developed several improvements to the amplitude accuracy of the filter by using higher-order time derivative approximations [7–10] to arrive at what is known as the RAW filter and its variants. Recently Williams and coauthors shows that the filtered leapfrog scheme compares favorably with the accuracy of the third order Runge Kutta (RK3) scheme while being computationally less expensive [11]. In other recent work, Li and Trenchea derive the higher-order RA (hoRA) time filter that improves the stability and accuracy of the leapfrog scheme [12], and Guzel and Trenchea were able to show that when the hoRA scheme is combined with the Williams step, both accuracy and stability could be improved even further [13].

Early analysis of filter schemes focused on the treatment of ordinary differential equations, and the application of time filters to discretizations of PDEs is still being explored with some early work found here [14–16]. Other methods have been developed for achieving higher order accuracy for PDE simulations. One class of methods involves the use of weighted essentially nonoscillatory (WENO) schemes for the spatial discretization and then paired with a time discretization scheme, such as Runge Kutta or Adams–Bashforth methods. The review paper by Shu [17] notes that care must be taken when combining the spatial and time discretizations for the case of hyperbolic conservation laws since stability of the algorithm is not guaranteed. For example, when a high order WENO spatial discretization is combined with a forward Euler method or a second order Runge–Kutta method, the resulting scheme is linearly unstable. Using a finite volume method with a flux limiter is another option for achieving higher order accuracy when solving hyperbolic conservation laws [18,19]. In each of these approaches, the algorithm is constructed with advection in mind, and a balance between spatial accuracy and stability of the time stepping method must be addressed in order to guarantee convergence of the scheme.

The work presented here is an analysis of a PDE simulation scheme that is combined with a time filter technique in order to improve accuracy. We focus on a time filter approach since it is non-intrusive; that is, it can be viewed as a post processing step that is amended to an existing approximation scheme, such as a finite difference scheme or a finite volume method, without making any changes to the original solver. This approach results in an algorithm that is easy to implement for existing legacy codes, and the accuracy and stability of the existing solver can be improved for minimal computational costs. In our case, the original solver is an explicit PDE discretization scheme, and a full convergence analysis of the filtered scheme is straightforward.

The work presented here is motivated by the original RA filter, which was developed to numerically solve the first order system

$$y' = f(t, y(t)), \quad t > 0 \quad y(0) = y_0$$

with the leapfrog centered at t_n and applying the filter as a second step. This process can be written as follows

Step 1:

$$\frac{y_{n+1}^* - y_{n-1}}{2\Delta t} = f(t_n, y_n^*)$$

Step 2:

$$y_n = y_n^* + \frac{\gamma}{2}(y_{n+1}^* - 2y_n^* + y_{n-1}), \quad \text{with } \gamma \approx 0.1.$$

where γ is the time filter parameter, y_n^* and y_{n+1}^* denote unfiltered values for an approximation to $y(t_n)$ and $y(t_{n+1})$ respectively and y_n is a filtered approximated value. In this algorithm, the value of y_{n+1}^* from step 1 is required in the calculation of the filtered value for y_n^* indicating that step 1 is computed twice before moving to the next step.

In recent years, Guzel and Layton [20] proposed shifting the filter to t_{n+1} for use with single step methods. The filter in this case takes the following form

$$y_{n+1} = y_{n+1}^* - \frac{\gamma}{2}(y_{n+1}^* - 2y_n + y_{n-1}) \quad (4)$$

As shown in [20], the term $y_{n+1}^* - 2y_n + y_{n-1}$ in (4) is the discrete curvature that is calculated from a quadratic interpolant polynomial ϕ_n given by

$$\phi_n(t) = y_{n+1}l_{n+1}(t) + y_n l_n(t) + y_{n-1}l_{n-1}(t), \quad t_{n-1} \leq t \leq t_{n+1}$$

with the standard Lagrangian basis functions $l_{n-1}(t)$, $l_n(t)$, and $l_{n+1}(t)$. Thus the discrete curvature centered at level n is defined as

$$\kappa_n = \Delta t^2 \phi_n''(t) = y_{n+1} - 2y_n + y_{n-1}. \quad (5)$$

It is claimed that $0 < \gamma < 2$ in (4) results in reduced curvature for the filtered approximation. The analysis contained in [20] shows that combining the filter with the fully implicit or Backward Euler (BE) method in two steps leads to a linear multistep method inducing an increased accuracy from first to second order for $\gamma = \frac{2}{3}$. In addition, the method is shown to be strongly A-stable, and this filter parameter value gives an immediate error estimator based on the pre- and post-filter difference that has the advantage of easily extending the combination to variable time steps.

In this work the curvature reduced time filter introduced in (4) is combined with the classical upwind scheme, and we analyze the influence of the value for the filter parameter, γ , in the stability and convergence of the simulation using the method of modified equation. This technique results in a CFL condition that is a function of γ , and the local truncation calculation yields an improved accuracy by a factor of $|\frac{\gamma}{2} - 1|$. The method of modified equation is further used to analyze the dissipation and dispersion behavior of the filtered scheme versus classical upwind. In addition, we investigate the optimal value for γ in the context of computational cost of simulations, reducing dissipation as well as improving accuracy of approximated solutions.

The outline of the paper is as follows. Section 2 gives the derivation of the time filtered upwind scheme. Section 3 uses the modified equation analysis approach to analyze both the dissipation and dispersion for the filtered method. The local truncation error and a CFL condition are derived. Section 4 illustrates the numerical convergence results for the model problem. This includes test problems highlighting the dissipation and dispersion results given in Section 3. Finally the last part of this section is dedicated to illustrating how the filtered scheme can be used for improved accuracy in estimating the Average Transcription Time (ATT) in the biological application.

2. A time filter scheme for the linear problem

This section introduces a time filter combined with the fully explicit upwind method in (7). The core idea of adding a time filter, as introduced in [20], to a classical finite difference scheme is to increase the accuracy of the scheme by reducing the discrete curvature of the solution. This result is simply accomplished by adding a single line of code that utilizes a linear combination of approximated solutions at time levels t_{n-1} , t_n and t_{n+1} with a scaling factor γ known as the time filter parameter. In our application, the algorithm can be viewed as a two-step method where the unfiltered approximation at each time step is computed using the upwind method as step 1, and the approximation is then filtered in step 2. The filtered approximation becomes the next iterate in the scheme.

The method is derived by discretizing the PDE at the following mesh grid points $x_j = jh$, for $j = 0, 1, 2, \dots, M$ and $t_n = nk$, for $n = 0, 1, 2, \dots, N$. Define the approximation $P_j^n \approx \rho(x_j, t_n)$, and Eq. (1) in its discretized form reads as

$$\frac{P_j^{n+1} - P_j^n}{\Delta t} + \bar{v} \frac{(P_j^n - P_{j-1}^n)}{\Delta x} = 0, \quad (6)$$

and the typical upwind scheme is expressed as

$$P_j^{n+1} = P_j^n - \nu (P_j^n - P_{j-1}^n). \quad (7)$$

The parameter ν , known as the Courant number, is defined to simplify the statement of the method

$$\nu = \bar{v} \frac{\Delta t}{\Delta x}, \quad (8)$$

and for a fixed choice of the parameters, the value of ν is constant. The upwind method is known to be stable when

$$0 < \nu \leq 1 \quad (9)$$

see [18,21–23] as well as many other textbooks on the subject. The above condition indicates the choice of $\Delta t \leq \frac{1}{\bar{v}} \Delta x$ providing a proper rate of resolution in both space and time while preserving the stability.

A fully explicit discretization of the upwind scheme combined with a time filter at a typical point (x_j, t_{n+1}) reads as

Step 1:

$$\tilde{P}_j^{n+1} = (1 - \nu)P_j^n + \nu P_{j-1}^n \quad (10)$$

Step 2:

$$P_j^{n+1} = \tilde{P}_j^{n+1} - \frac{\gamma}{2}(\tilde{P}_j^{n+1} - 2P_j^n + P_{j-1}^{n-1}) \quad (11)$$

where $\tilde{P}_j^{n+1}$ and P_j^{n+1} denotes the unfiltered and filtered density approximations, respectively. The parameter γ in the second step is the time filter parameter whose value is considered to be non-zero. One can note that $\gamma = 0$ reduces the scheme to the classical upwind. Due to the linearity of the filter, the entire process can be simplified into a single step method. Solving Eq. (11) for the filtered variable and substituting into the upwind scheme yields the one-step scheme.

Algorithm 2.1. The upwind method with the time filter assuming a constant time step with γ as the filter parameter is described as follows. Given P_{j-1}^n, P_j^n and P_j^{n-1} , find P_j^{n+1} satisfying the following

$$P_j^{n+1} = [\gamma + (1 - \nu)(1 - \frac{\gamma}{2})]P_j^n - \frac{\gamma}{2}P_j^{n-1} + \nu(1 - \frac{\gamma}{2})P_{j-1}^n \quad (12)$$

Proof. The Eq. (12) is simply obtained by solving (11) for $\tilde{P}_j^{n+1}$ in Step 2 that yields

$$\tilde{P}_j^{n+1} = \left(\frac{2}{2 - \gamma} \right) \left[P_j^{n+1} - \gamma P_j^n + \frac{\gamma}{2} P_j^{n-1} \right] \quad (13)$$

and replacing (13) into the upwind scheme in Step 1 above. $\square$

Remark. Based on the definition in (5), at any given point x , the discrete curvature centered at level n is defined by

$$\kappa_n = \Delta t^2 \phi_n''(t) = P^{n+1} - 2P^n + P^{n-1}, \quad (14)$$

then adding $-2P^n + P^{n-1}$ to both sides of the filter expression in (11) gives $\kappa_n = (1 - \frac{\gamma}{2})\tilde{\kappa}_n$ for the curvature post- and pre-filter, κ_n and $\tilde{\kappa}_n$. This indicates that the curvature reduction holds for $0 < \gamma < 2$. Next, we derive the convergence results for the filtered scheme (12).

3. Analysis of the time filtered upwind scheme

In this section, the local truncation error for the filtered scheme is derived, and the stability of the method is addressed using a modified equation analysis. If $\mathcal{N}(\cdot)$ denotes the numerical operator mapping the approximate solution at one time step to the next, the one-step error is defined by $\mathcal{N}(\rho^n) - \rho^{n+1}$. The local truncation error at t_n is defined by

$$\tau^n = \frac{1}{\Delta t} (\mathcal{N}(\rho^n) - \rho^{n+1}) \quad (15)$$

indicating the magnitude of the global error as well as the order of accuracy for when the method is stable. The main result describing consistency and stability is given below.

Theorem 3.1. The scheme (12) in Algorithm 2.1 is consistent of first order in both $\mathcal{O}(\Delta t)$ and $\mathcal{O}(\Delta x)$. When $0 < \nu \leq CFL(\gamma) = \frac{2-\gamma}{2+\gamma}$, the method is stable and convergent.

Proof. Let $\rho_j^n = \rho(x_j, t_n)$. Then $\mathcal{N}(\rho_j^n)$ represents the right side of (12) evaluated at the true solution and is given by

$$\mathcal{N}(\rho_j^n) = [\gamma + (1 - \nu)(1 - \frac{\gamma}{2})]\rho_j^n - \frac{\gamma}{2}\rho_j^{n-1} + \nu(1 - \frac{\gamma}{2})\rho_{j-1}^n. \quad (16)$$

Thus the local truncation error is formulated as

$$\begin{aligned}\tau^n &= \left(\frac{\gamma}{2} - 1\right)[\rho_t + \bar{v}\rho_x + \Delta t\rho_{tt} - \frac{1}{2\bar{v}}\Delta x\rho_{xx} + \mathcal{O}(\Delta t^2)] \\ &= \left(\frac{\gamma}{2} - 1\right)\left[\Delta t\rho_{tt} - \frac{1}{2\bar{v}}\Delta x\rho_{xx} + \mathcal{O}(\Delta t^2)\right]\end{aligned}$$

This shows that the local truncation error for the method is $\mathcal{O}(\Delta t)$, and that the method is first order in both time and space. One should also observe that for values of γ that are close to 2, the local truncation is very small.

For stability of the method, we follow the modified equation analysis in order to derive a CFL condition, see Chapter 8 of [18,24]. Define a function $u(x, t)$ that agrees with P_j^n in (12) at the grid points in the mesh, where we identify $x = x_j$ and $t = t_n$. Then $u(x, t)$ satisfies

$$u(x, t + \Delta t) = [\gamma + (1 - \nu)(1 - \frac{\gamma}{2})]u(x, t) - \frac{\gamma}{2}u(x, t - \Delta t) + \nu(1 - \frac{\gamma}{2})u(x - \Delta x, t) \quad (17)$$

For the modified equation analysis one uses Taylor expansions for each term of the difference scheme in order to derive a PDE for which the difference scheme is a solution. This PDE is referred to as the modified equation, and for the general time filtered upwind scheme it is given by

$$u_t + \bar{v}u_x = \frac{\bar{v}\Delta x}{2}(1 - \nu\alpha)u_{xx} + \frac{\bar{v}\Delta x^2}{6}[(1 - 3\alpha^2)\nu^2 + 3\alpha\nu - 1]u_{xxx} + \mathcal{O}(\Delta x^4) \quad (18)$$

where ν is defined in (8), and the parameter α is the reciprocal of $CFL(\gamma)$. That is,

$$\alpha = \frac{2 + \gamma}{2 - \gamma}$$

The details of the calculations required to derive Eq. (18) are described in Appendix. The filtered scheme is stable if the diffusion coefficient is positive or zero; that is, when

$$\frac{\bar{v}\Delta x}{2}(1 - \nu\alpha) \geq 0,$$

which leads to the stability requirement

$$\nu\alpha \leq 1 \Leftrightarrow \nu \leq \frac{1}{\alpha} \Leftrightarrow \nu \leq \frac{2 - \gamma}{2 + \gamma} = CFL(\gamma) \quad (19)$$

This yields a precise stability limit for the filtered method in Algorithm 2.1. Since the method is shown to be consistent and since stability holds when the time step Δt is chosen according to the CFL condition in the above equation, then by the Lax Equivalence Theorem, the filtered upwind method is convergent. $\square$

3.1. Dissipation and dispersion

Both dissipation and dispersion are of central importance in finite difference methods. Although many PDEs are non-dissipative and/or non-dispersive, their discrete approximation schemes almost invariably exhibit both phenomena. To characterize these properties for the filtered upwind scheme, we refer to the modified equation in (18), and we note that the amount of dissipation and dispersion exhibited by the scheme depends on the filter parameter $0 < \gamma < 2$. The filtered scheme can be viewed as a first order accurate approximation to the true solution of (18), and this indicates that the solutions to this advection-diffusion equation travel at the proper speed $\bar{v}$. The dissipation in the method is characterized by the coefficient of u_{xx} in (18). This indicates that solutions diffuse and sharp corners in solutions are smeared out when $(1 - \nu\alpha) > 0$.

Remark. For the special case where we choose $\nu = CFL(\gamma) = \frac{2-\gamma}{2+\gamma}$, the diffusion coefficient in (18) vanishes, and there is no dissipation in the method. In order for the method to be stable, we must have $(1 - \nu\alpha) \geq 0$. Larger values of $(1 - \nu\alpha) > 0$ result in greater dissipation, and when γ is fixed, one can choose the time step Δt so that the dissipation can be made close to zero while stability is maintained.

Following the discussion in [24], the dispersion is identified by the coefficient of u_{xxx} in (18) defined as below,

$$\mu(3) = \frac{\bar{v}\Delta x^2}{6}[(1 - 3\alpha^2)\nu^2 + 3\alpha\nu - 1], \quad (20)$$

and the status of wave speed for the approximated solution depends on the sign and the magnitude of $\frac{-1}{\bar{v}}\mu(3)$. A Fourier mode is known to have a lower wave speed than the exact mode if $\frac{-1}{\bar{v}}\mu(3) > 0$ and to have a higher wave speed than the exact mode if $\frac{-1}{\bar{v}}\mu(3) < 0$. Given the following expression,

$$\frac{-1}{\bar{v}}\mu(3) = \frac{-\Delta x^2}{6}[(1 - 3\alpha^2)\nu^2 + 3\alpha\nu - 1]$$

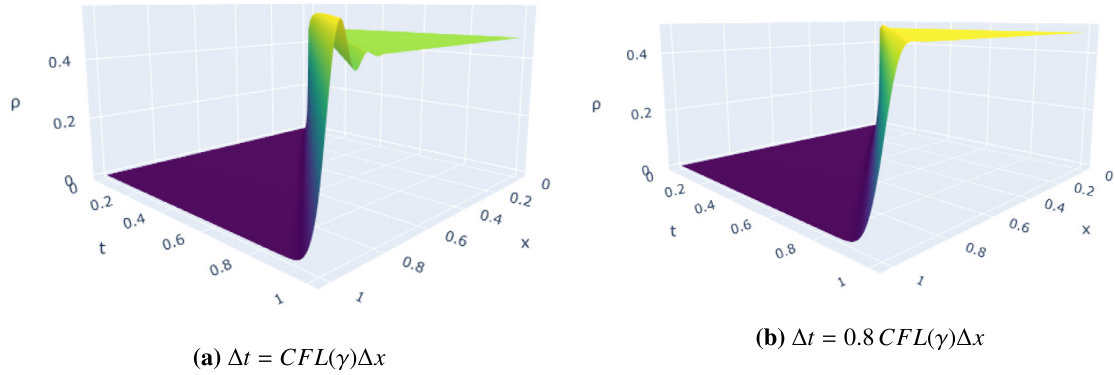


Fig. 2. The density surface plot for filtered upwind with $\gamma = 1.75$ in Eq. (12) of Algorithm 2.1 and different choices of Δt for a fixed mesh size of $\Delta x = \frac{1}{128}$.

we denote $p(v) = (1 - 3\alpha^2)v^2 + 3\alpha v - 1$. It is easy to show that $p(v)$ is a downward parabola as long as $0 \leq \gamma < 2$. One can calculate the range of values for the leading coefficient of $p(v)$ as a function of γ as given below,

$$-3 < 1 - 3\alpha^2 = 1 - 3 \left[\frac{2 + \gamma}{2 - \gamma} \right]^2 \leq -2 \quad (21)$$

Furthermore the maximum value of $p(v)$ happens at $v_0 = -\frac{3\alpha}{2(1-3\alpha^2)}$, and $p(v_0) = \frac{-4+3\alpha^2}{4(1-3\alpha^2)}$. One can obtain that for $\gamma \approx 0.144$, $p(v_0) = 0$ and for $0.144 < \gamma < 2$, $-\frac{1}{4} < p(v_0) < 0$. This results in $\frac{-1}{v}\mu(3) > 0$ indicating that the approximated solution moves with a lower speed than the true solution.

4. Numerical results

In this section, we present numerical illustrations in three themes. First the stability and convergence for the filtered upwind schemes with different values of filter parameter γ are examined, and a comparison versus the classical upwind scheme is presented. Second dissipation and dispersion of the filtered upwind schemes versus upwind are tested using two benchmark problems with periodic boundary conditions. Lastly estimations of the biology measure for ATT using the filtered and unfiltered upwind schemes are presented in presence of perturbations in elongation velocity and the inflow condition respectively. All the computations in this paper are carried out using the programming language Python.

The results in this section presented under the numerical error and the biology application are given for constant initial density and inflow condition with values $\rho_0 = 0$ and $\rho_l = 0.47$ respectively. Choice for these conditions are made according to the biology measures due to the application of modeling the DNA transcription process. We note that in this case of constant initial and boundary conditions, the general solution of Eqs. (1) has characteristics in the form of $x(t) = \bar{v}t + x_0$ and the analytical solution to (1)–(3) is given by,

$$\rho(x, t) = \begin{cases} \rho_l, & 0 < x < \bar{v}t \\ \rho_0, & \bar{v}t < x < 1. \end{cases} \quad (22)$$

This expression is used to evaluate numerical errors in the following section.

4.1. Numerical error and convergence

In order to estimate accuracy and demonstrate convergence in this section, we examine the computational error for approximated density using upwind and filtered upwind with different values of filter parameter in the range of $0 \leq \gamma < 2$ over the finite time interval $[0, 1]$ or final time $T = 1$. The error at the grid point (x_j, t_n) is denoted as $E_j^n = \rho_j^n - P_j^n$, and our choice for the error norm is the L^1 -norm defined as $\|E^n\| = \Delta x \sum_{j=0}^M |E_j^n|$ for $n = 0, 1, 2, \dots, N$. In the error calculations for (7) and (12), we assume a fixed ratio $\nu = \frac{\Delta t}{\Delta x} = 0.8 CFL(\gamma)$ for the case of $\bar{v} = 1$. The factor of 0.8 is to warranty a numerical solution free of oscillations. Fig. 2 illustrates the effect of this factor resulting in a smooth approximated solution surface without oscillations for filtered upwind with the choice of $\gamma = 1.75$. Letting $\Delta t \rightarrow 0$ by refining the spatial grid, i.e. Δx , the errors and their rate of convergence at different times within the interval $[0, 1]$ for $\gamma = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5$ and 1.75 at $t = 1$ are calculated and graphically presented in Figs. 3(a) and 3(b). As one observes for each fixed γ value, the errors using all schemes become smaller converging to zero as $\Delta x \rightarrow 0$. Comparing the error plots for different γ values the approximated density, P_j^n , shows a better accuracy for larger values of γ . The

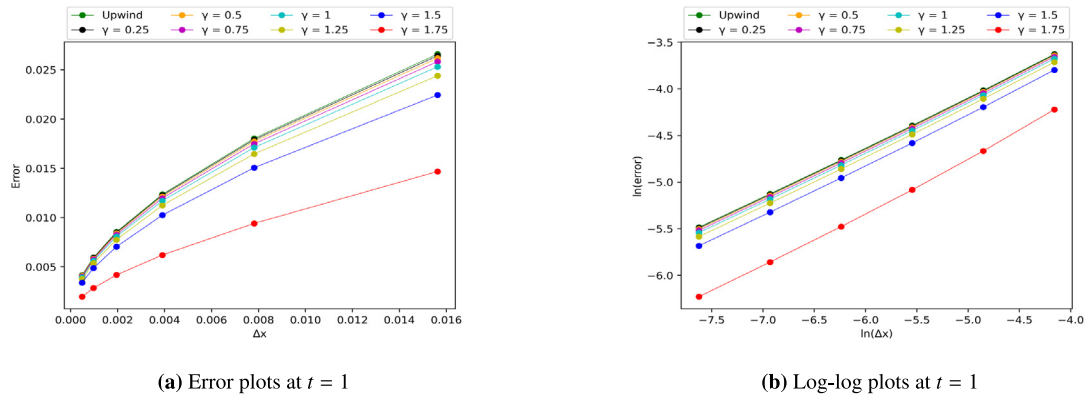


Fig. 3. The L^1 -error and log-log plots for the filtered and upwind schemes at $t = 1$ with $\Delta t = 0.8 \text{CFL}(\gamma)\Delta x$ for $\gamma = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5$ and 1.75 .

Table 1

The density approximation error for upwind and filtered upwind with $\gamma = 1.75$ using L^1 -norm and $\Delta t = 0.8 \text{CFL}(\gamma)\Delta x$.

L^1 Errors using upwind					
Δx	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$	$t = 1$
2^{-6}	2.0502843 e-2	2.8801752 e-2	3.5201096 e-2	4.0135472 e-2	2.6596661 e-2
2^{-7}	1.4400876 e-2	2.0348151 e-2	2.4983407 e-2	2.8763555 e-2	1.8032184 e-2
2^{-8}	1.0174076 e-2	1.4408416 e-2	1.765423 e-2	2.040246 e-2	1.23596 e-2
2^{-9}	7.204208 e-3	1.0201711 e-2	1.2484444 e-2	1.4421344 e-2	8.542645 e-3
2^{-10}	5.100856 e-3	7.210672 e-3	8.830084 e-3	1.0196868 e-2	5.941618 e-3
2^{-11}	3.605336 e-3	5.098434 e-3	6.245251 e-3	7.210903 e-3	4.151713 e-3
L^1 Errors using filtered upwind with $\gamma = 1.75$					
Δx	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$	$t = 1$
2^{-6}	1.1132714 e-2	1.4705811 e-2	1.7473431 e-2	1.9902732 e-2	1.4680994 e-2
2^{-7}	7.352905 e-3	9.952463 e-3	1.1986604 e-2	1.366 e-2	9.408003 e-3
2^{-8}	4.976232 e-3	6.830003 e-3	8.271503 e-3	9.512124 e-3	6.199179 e-3
2^{-9}	3.415001 e-3	4.756062 e-3	5.789412 e-3	6.665715 e-3	4.177955 e-3
2^{-10}	2.378031 e-3	3.332857 e-3	4.073224 e-3	4.700162 e-3	2.857089 e-3
2^{-11}	1.666429 e-3	2.350081 e-3	2.875603 e-3	3.318734 e-3	1.971929 e-3

error rate of convergence presents a similar pattern. The filtered scheme with $\gamma = 1.75$ shows the most significant error reduction while retaining the same rate of convergence compared to other schemes, see Fig. 3(b).

Error values for the approximated density, P_i^n , computed by upwind and by filtered upwind with $\gamma = 1.75$ are listed in Table 1. Comparing the errors in the table, the latter scheme shows a better accuracy at all computed times through all the mesh refinements by almost a factor of $\frac{1}{2}$. Table 2 contains the rate of convergence estimations at $t = 1$ for both schemes. Given a grid size defined as $h_i = 2^{-(i+5)}$ for $i = 1, 2, 3, \dots, 6$, the convergence rate of the error is determined by calculating the exponent a in the following expression,

$$\left(\frac{h_i}{h_{i+1}}\right)^a = \frac{\|E(h_i)\|}{\|E(h_{i+1})\|} \quad (23)$$

The convergence rate values indicate that the L^1 -norm of the errors decay at the rate of about $\frac{1}{2}$ in all cases. This means that error decreases like $(\Delta x)^{\frac{1}{2}}$. Although the methods are all first order based on the local truncation error analysis in Section 3, these results are valid for smooth solutions only. One might expect a lower rate of convergence for discontinuous solutions like the one we explore here, as indicated in Section 8.7 of [18].

The numerical solutions obtained by upwind and filtered upwind with different γ values versus the analytical solution over the time interval $[0, 1]$ are demonstrated in Figs. 4 and 5. These figures demonstrate a comparison of the numerical solutions as functions of one variable at a time. In Fig. 4, the approximated density is calculated at different spatial locations as a function of time and in Fig. 5(a), it is shown at $t = 1$ as a function of space. Fig. 5(b) illustrates the numerical solutions along the characteristic line $x = t$. As one can observe, the upwind with time filter drops down to zero faster than upwind thus capturing the jump discontinuity more quickly. All the plots present a better accuracy of the numerical solution via filtered upwind. Overall our numerical results are in favor of using the time filter. Although all the numerical schemes that we have tested share the same rate of convergence, the calculated errors indicate a smaller error and therefore a better estimation on a coarse mesh when a time filter is applied.

Table 2

The convergence rate estimate for upwind and filtered upwind with $\gamma = 1.75$ at $t = 1$ according to the error values in Table 1 using the values of i to estimate the value of a in Eq. (23).

Rate of convergence		
i	$\gamma = 0$	$\gamma = 1.75$
1	0.560671043	0.641989278
2	0.544942108	0.601811426
3	0.532877323	0.569280247
4	0.523826975	0.548250738
5	0.517149093	0.534938674

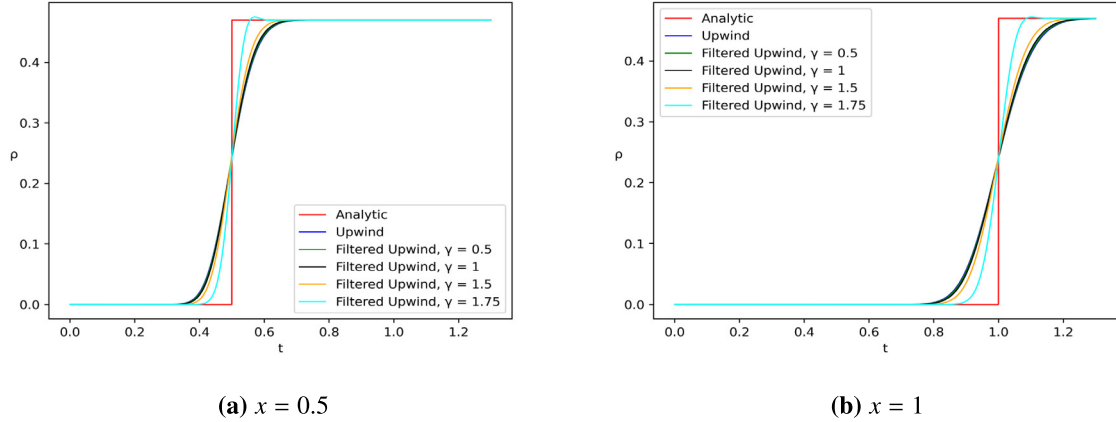


Fig. 4. The density plots for $0 < t < 1$ at $x = 0.5$ and $x = 1$ respectively with the choice of $\Delta x = \frac{1}{128}$ and $\Delta t = 0.8 \text{CFL}(\gamma) \Delta x$.

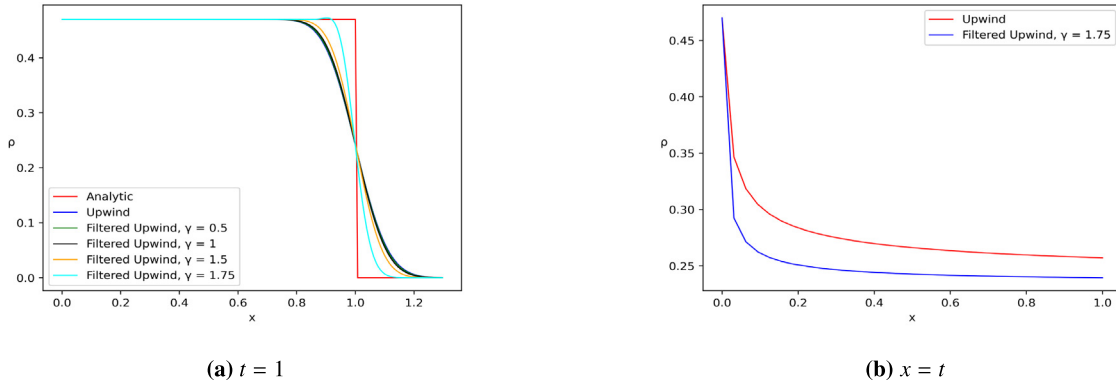


Fig. 5. Figure (a) is the density plot for $0 < x < 1$ at $t = 1$ respectively and figure (b) is the density plot for upwind and filtered upwind with $\gamma = 1.75$ along the characteristic line $x = t$ with the choice of $\Delta x = \frac{1}{128}$ and $\Delta t = 0.8 \text{CFL}(\gamma) \Delta x$.

4.2. Dissipation and dispersion test problems

To provide a comparison in dissipation and dispersion behavior of the upwind versus the filtered upwind scheme with different γ values, we utilize a periodic boundary condition test problem generally formulated as,

$$\begin{aligned} \rho_t + \rho_x &= 0, & 0 < x \leq 1, \quad t > 0 \\ \rho(x, 0) &= \rho_0(x) \\ \rho(0, t) &= \rho(1, t) \end{aligned}$$

where two different density initial condition profiles listed below are considered for the purpose of examining the dissipation and dispersion respectively. For *Case 1*, we choose $\rho_0(x) = \sin 2\pi x$, and for *Case 2*, we use $\rho_0(x) = e^{-100(x-0.5)^2} \sin(80x)$. These examples can be found in [18,22].

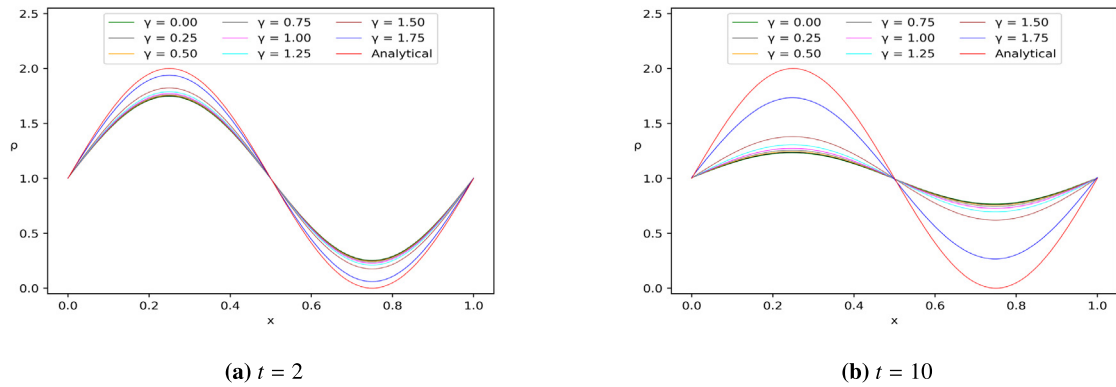


Fig. 6. Case 1: The density plots for one period of the initial density, $0 < x < 1$, at $t = 2$ and $t = 10$ for different γ values with the choice of $\Delta x = \frac{1}{128}$ and $\Delta t = 0.8 \text{ CFL}(\gamma) \Delta x$.

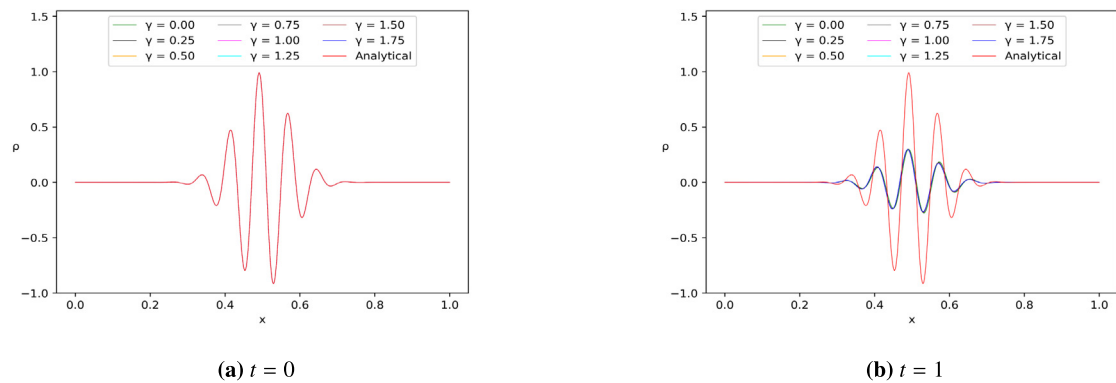


Fig. 7. Case 2: The density plots for one period of the initial density, $0 < x < 1$, at $t = 0$ and final time $t = 1$ for different γ values with the choice of $\Delta x = \frac{1}{512}$ and $\Delta t = 0.8 \text{ CFL}(\gamma) \Delta x$.

Fig. 6 shows dissipation in density as time progresses from 0 to 10. A greater dissipation is observed for smaller γ values indicating $\gamma = 1.75$ as an optimal filter parameter for obtaining the least dissipation while maintaining a time step of a reasonable size. In Fig. 7, one can observe that the dispersion happens at the same rate for all the tested schemes in the time interval of $[0, 1]$. Overall, there is a very small dispersion in the form of a phase shift seen in Fig. 7(b) for all γ values. One can notice that the dispersion is of a negligible magnitude.

4.3. Biology application: Average transcription time estimation

Returning to the motivating biological model of the compartments describing transcription and translation, the computations in this section are focused on estimating ATT, that is, we estimate the amount of time a molecular motor spends in a compartment. In each compartment of the full model, the key parameters are those that control the initiation rate and the elongation rate of RNA polymerase on the strand. In genes where the density of RNAPs is observed to be very high, transcriptional elongation of RNAPs is not uniform over the extent of the strand or over the time horizon of a cell cycle. Indeed, bacteria are capable of adjusting their growth rate according to the available resources in their environment, and they can respond quickly and efficiently to sudden changes in these resources. Environmental changes can affect both the initiation and elongation rates of the RNAPs transcribing the gene. The numerical study in this section considers estimation of ATT and the effects of small perturbations in the elongation velocity and initiation rate on this measure.

4.3.1. Perturbations in the average elongation velocity

It is known that the elongation rate of individual RNAPs can vary, and a range for the elongation rate is reported as 70 - 200nts per second, see page 3744 of [25]. We consider perturbations in the velocity of the linear model problem once it has been nondimensionalized. That is done by replacing $\bar{v}$ with $1 + \epsilon$ in Eq. (1). Taking into account rescaling the model in the Appendix section of [4], the perturbations correspond to values of average elongation rate that range from 90nts

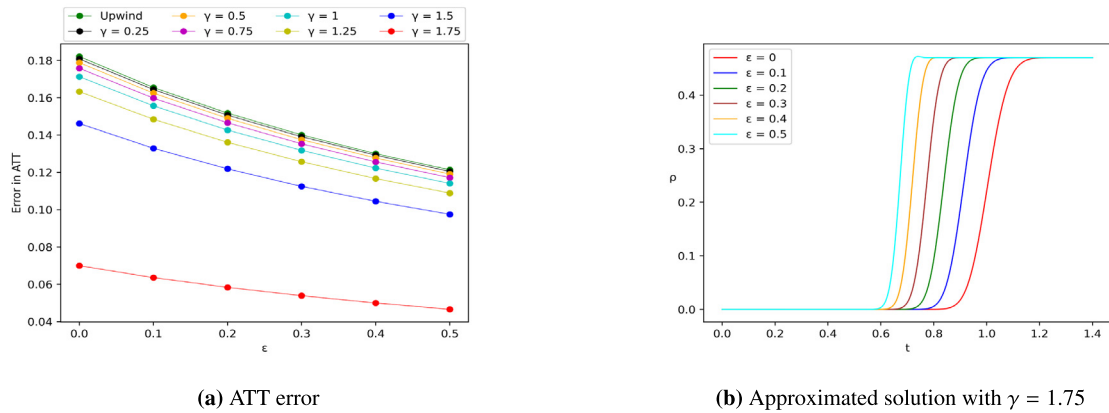


Fig. 8. The ATT error plots for different γ values and the approximated solution at $x = 1$ with filter parameter $\gamma = 1.75$ for $\epsilon = 0, 0.1, 0.2, 0.3, 0.4$ and 0.5 with the choice of $\Delta x = \frac{1}{128}$ and $\Delta t = \frac{1}{1+\epsilon}(0.8 CFL(\gamma)\Delta x)$.

Table 3

The first four columns contain error values for the average transcription time approximation using the filtered upwind with $\gamma = 1.75$ and upwind ($\gamma = 0$) along line $x = 1$ with perturbed elongation velocity for $\epsilon = 0.1, 0.2, 0.3, 0.4, 0.5$ and the grid size of $\Delta x = \frac{1}{128}$ and $\Delta t = \frac{1}{1+\epsilon}(0.8 CFL(\gamma)\Delta x)$. The last three columns contain the errors in ATT estimation for filtered upwind and upwind along line $x = 1$ for the case of perturbed inflow for $\epsilon = \pm 0.01, \pm 0.05, \pm 0.1$ and the grid size of $\Delta x = \frac{1}{128}$ and $\Delta t = 0.8 CFL(\gamma)\Delta x$.

Perturbations in elongation velocity				Perturbations in initiation rate		
ϵ	Analytic	Error in ATT estimation		ϵ	Error in ATT estimation	
		$\gamma = 1.75$	$\gamma = 0$		$\gamma = 1.75$	$\gamma = 0$
0	1	0.07	0.1821	-0.1	0.0675	0.1725
0.1	0.9092	0.0466	0.165	-0.05	0.0688	0.1775
0.2	0.8333	0.0338	0.1509	-0.01	0.0696	0.1813
0.3	0.7696	0.0275	0.1387	0.01	0.07	0.1829
0.4	0.7146	0.0262	0.1283	0.05	0.07	0.1862
0.5	0.6667	0.0248	0.1195	0.1	0.071	0.19

per second to 136nts per second. In the results that follow, we report on only the positive perturbations of the elongation velocity. Similar results were obtained with negative perturbations of the velocity, but we omit those for brevity.

Given a small perturbation in the advection velocity, say $\bar{v} = 1 + \epsilon$ where $0 \leq \epsilon \leq 0.5$, the analytical solution is constant along the characteristic lines of $x(t) = (1 + \epsilon)t + x_0 = \bar{v}t + x_0$. Fig. 8(a) graphically presents the error estimation in ATT using different filter parameter values in the numerical scheme, $\gamma = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5$ and 1.75 , and perturbation values $\epsilon = 0, 0.1, 0.2, 0.3, 0.4$ and 0.5 . As shown, larger γ values result in a better accuracy for ATT estimation leading to the best estimation with $\gamma = 1.75$. It is also notable that the errors for all the schemes are smaller for larger values of the perturbation parameter ϵ . In Fig. 8(b), one observes the approximated solution using $\gamma = 1.75$ travels faster for larger ϵ values.

Table 3 gives the analytical values of the ATT along with the ATT estimations and the errors in the approximations calculated using both the filtered scheme in Algorithm 2.1 with $\gamma = 1.75$ and the upwind scheme when $\gamma = 0$. Both schemes consistently overestimate the ATT; however, the filtered scheme is more accurate than the upwind scheme for all the ϵ values, reducing the error by a factor of at least about $1/3$. As shown in [4], both numerical schemes more accurately estimate the jump in the function $\rho(1, t)$ for larger perturbations indicating that the wave travels more quickly across the domain, and the approximation schemes have less time for the discontinuity to smear.

4.3.2. Perturbations in the initiation rate

We study perturbations in the initiation rate while the elongation velocity is fixed at $\bar{v} = 1$. The inflow ρ_l in the equations corresponds to the initiation rate of the biological model. The parameter $\rho_l = 0.47 \pm \epsilon$ is perturbed by $0 \leq \epsilon \leq 0.1$. Note that the ATT is independent of any perturbation in this parameter and stays the same as one unit of time for various values of ϵ . In addition, the density along the right boundary at $x = 1$ attains the value $\rho_l = 0.47 \pm \epsilon$ at that time. Table 3 reports the error estimations of ATT calculated via the filtered scheme using $\gamma = 1.75$ and the standard upwind. The ATT estimations by the filtered scheme presents about 40% better accuracy compared to that of the traditional upwind for all the ϵ values using the same grid size. It is notable that the filtered scheme delivers increased accuracy over the range of perturbations and with the same computational overhead.

5. Conclusion

In this work, we present the numerical analysis for a curvature reduced time filter applied to the classical upwind scheme for advection. The method is first order in both time and space, and an approximately optimal choice of the filter parameter is identified. [Theorem 3.1](#) shows that when γ is chosen so that $|\frac{\gamma}{2} - 1|$ is small, then the local truncation of the method is small, and the accuracy of the filtered upwind scheme is an improvement over the original upwind. This improvement is obtained with very little adjustments to a code with the only major intrusion being that the solution at one more previous time step must be stored. The local truncation error is computed, and the CFL condition is derived using the modified equation method in [Theorem 3.1](#). These results indicate that for γ close to 2, the time step must be reduced significantly from that of the original upwind scheme. Therefore, there must be a balance between improving accuracy and maintaining a time step that is of a reasonable size for long time computations. For the cases considered here, we recommend that $\gamma = 1.75$ is an optimal choice for that balance.

This work also shows that the relationship between the filter parameter and both the local truncation error and the dissipation of the method is the key to identifying a reasonable range for the filter parameter. The analysis also demonstrates that the filtered upwind scheme exhibits very little dispersion over the full range of parameters considered here. The numerical results validate these observations, and it is clear that choosing the filter parameter to minimize dissipation should be the priority whenever advection is the dominant physical process. Finally, we show that the filtered upwind scheme can be used to improve calculations for the ATT in the presence of perturbations for a mathematical model of transcription of ribosomal RNA.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to acknowledge the support of the National Science Foundation, United States under Award DMS-1951510. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. The authors offer special thanks to Prof. Zafer Hatahet, Embry-Riddle Aeronautical University, for his continued support providing lectures, consultations and many useful references on *rrn* operon polymerization process. The authors also acknowledge Prof. T. Gedeon, Montana State University, for the many helpful conversations about ribosomal assembly mechanisms and for the ongoing collaboration. Finally, the authors are grateful to the anonymous reviewers for their time and their constructive comments that have improved the quality of this paper.

Appendix. Derivation of modified equation

This section presents the details of the modified equation analysis given in Section 3. We follow the process described in [24]. The modified PDE represents the actual PDE that the computed solution satisfies, assuming that the computed solution is obtained using the filtered scheme in (12). In order to derive the modified equation, one uses Taylor expansions of the computed solution to eliminate all higher order time derivatives (past the first order) leaving only the terms involving spatial derivatives. This includes eliminating all partial derivatives of mixed order.

Note that $\rho(x, t)$ satisfies the original advection PDE in (1)–(3). In contrast, the computed solution for the filtered upwind scheme, P_j^n , is a grid function defined at the grid points (x_j, t_n) of the mesh, and it satisfies (12) at the grid points. We assume that there exists a continuously differentiable function $u(x, t)$ that agrees with P_j^n at the grid points. That is, $u(x_j, t_n) = P_j^n$ for all j, n . Once the Taylor expansions of u are completed and substituted into (12), the function $u(x_j, t_n)$ must satisfy the expression below at all grid points. All grid notation is suppressed for the sake of brevity.

$$u_t + \bar{v}u_x + \alpha \frac{\Delta t}{2} u_{tt} - \bar{v} \frac{\Delta x}{2} u_{xx} + \frac{\Delta t^2}{6} u_{ttt} + \bar{v} \frac{\Delta x^2}{6} u_{xxx} + \alpha \frac{\Delta t^3}{24} u_{tttt} - \bar{v} \frac{\Delta x^3}{24} u_{xxxx} + \dots = 0 \quad (24)$$

The next step is to perform a sufficient number of algebraic manipulations using (24) in order to eliminate all terms involving a time derivative, with the exception of u_t , up to a desired order. At each step, the equation in (24) is multiplied by various coefficients, and then that result is added to (24) in order to eliminate a term with a time derivative. To make the process more clear, we define the expressions below as they are used in the following table. Define the following parameters $\alpha = \frac{2+\gamma}{2-\gamma}$, $\beta = \bar{v}\Delta t$, $\chi = 1 - 3\alpha^2$, along with the four parameters given below.

$$\begin{aligned} \mu_0 &= \frac{\beta}{12} [-2\beta\chi - 3\alpha\Delta x] & \mu_1 &= \beta \frac{\Delta t}{24} [3\alpha\beta(3 - 5\alpha^2) - 2\Delta x\chi] \\ \mu_2 &= \frac{\beta}{24} [4\beta\Delta x + 2\alpha\Delta x^2 - 9\alpha\beta^2 - 12\alpha^2\beta\Delta x + 15\alpha^3\beta^2] & \mu_3 &= -\frac{\bar{v}\Delta x^2}{6} [\chi v^2 + 3\alpha v - 1] \end{aligned}$$

Table A.4

Modified equation table of coefficients.

Multiplier with derivative of [*]	Coefficient of each partial derivative									
	$\frac{\partial u}{\partial t}$	$\frac{\partial u}{\partial x}$	$\frac{\partial^2 u}{\partial t^2}$	$\frac{\partial^2 u}{\partial t \partial x}$	$\frac{\partial^2 u}{\partial x^2}$	$\frac{\partial^3 u}{\partial t^3}$	$\frac{\partial^3 u}{\partial t^2 \partial x}$	$\frac{\partial^3 u}{\partial t \partial x^2}$	$\frac{\partial^3 u}{\partial x^3}$	$\frac{\partial^4 u}{\partial t^4}$
Coefficients of [*]	1	$\bar{v}$	$\alpha \frac{\Delta t}{2}$	0	$-\bar{v} \frac{\Delta x}{2}$	$\frac{\Delta t^2}{6}$	0	0	$\bar{v} \frac{\Delta x^2}{6}$	$\frac{\alpha \Delta t^3}{24}$
$-\alpha \frac{\Delta t}{2} [*]_t$	0	0	$-\alpha \frac{\Delta t}{2}$	$-\alpha \bar{v} \frac{\Delta t}{2}$	0	$-\alpha^2 \frac{\Delta t^2}{4}$	0	$\alpha \bar{v} \frac{\Delta t \Delta x}{4}$	0	$-\frac{\alpha \Delta t^3}{12}$
$\frac{\alpha \beta}{2} [*]_x$	0	0	0	$\frac{\alpha \beta}{2}$	$\frac{\bar{v} \alpha \beta}{2}$	0	$\frac{\alpha^2 \beta \Delta t}{4}$	0	$-\bar{v} \alpha \beta \frac{\Delta x}{4}$	0
$-\frac{\Delta t^2}{12} (x+1) [*]_{tt}$	0	0	0	0	0	$-\frac{\Delta t^2}{12} (x+1)$	$-\bar{v} \frac{\Delta t^2}{12} (x+1)$	0	0	$-\frac{\alpha \Delta t^3}{24} (x+1)$
$\beta \frac{\Delta t}{6} x [*]_{tx}$	0	0	0	0	0	0	$\beta \frac{\Delta t}{6} x$	$\bar{v} \beta \frac{\Delta t}{6} x$	0	0
$\mu_0 [*]_{xx}$	0	0	0	0	0	0	0	μ_0	$\bar{v} \mu_0$	0
$\alpha \frac{\Delta t^3}{8} (1-\alpha^2) [*]_{ttt}$	0	0	0	0	0	0	0	0	0	$\frac{\alpha \Delta t^3}{8} (1-\alpha^2)$
$\alpha \beta \frac{\Delta t^2}{24} (9\alpha^2 - 7) [*]_{ttx}$	0	0	0	0	0	0	0	0	0	0
$\mu_1 [*]_{bxx}$	0	0	0	0	0	0	0	0	0	0
$\mu_2 [*]_{bxxx}$	0	0	0	0	0	0	0	0	0	0
...	...									
Sum of coefficients	1	$\bar{v}$	0	0	$\frac{\bar{v} \Delta x}{2} (v\alpha - 1)$	0	0	0	μ_3	0

Each row in Table A.4 represents an equation that has been derived by differentiating Eq. (24) by the appropriate partial derivative and multiplying by a coefficient as indicated in the first column. For example, the second row under the headings of Table A.4 denotes the equation derived by differentiating (24) with respect to t and then multiplying it by the coefficient $-\alpha \frac{\Delta t}{2}$. The headings at the top of the tables denote the position of each partial derivative in the new equation, and the expression in row 2 under each column denotes the coefficient of that partial derivative within the new equation.

The table is constructed one row at a time so that it is clear how to choose the multiplier for each subsequent row. Once the equations are completed for each row, the final step is to sum all the equations. The coefficient of each partial derivative within the final sum is denoted at the bottom of Table A.4. The last row contains the final terms for the modified equation in (18). In order to save space, we have referenced Eq. (24) as an asterisk in brackets with the appropriate partial derivative in this fashion $[*]_a = [Eq.(24)]_a$, where a represents the operation of partial differentiation with respect to $a = t$ or $a = x$.

References

- [1] L. Davis, T. Gedeon, J. Gedeon, J. Thorenson, A traffic flow model for bio-polymerization processes, *J. Math. Biol.* (2013).
- [2] L. Davis, T. Gedeon, J. Thorenson, Discontinuous Galerkin calculations for a nonlinear PDE model of DNA transcription with short, transient and frequent pausing, *J. Comput. Math.* 32 (6) (2014) 601–629.
- [3] L. Davis, T. Gedeon, J. Thorenson, Parameter studies for a nonlinear continuum model of transcription, in: *Proceedings of 21st International Symposium on the Mathematical Theory of Networks and Systems*, 2014.
- [4] L. Davis, F. Pahlevani, T.S. Rajan, An accurate and stable filtered explicit scheme for biopolymerization processes in the presence of perturbations, *Appl. Comput. Math.* 10 (6) (2021) 121–137.
- [5] A.J. Robert, The integration of a low order spectral form of the primitive meteorological equations, *J. Meteorol. Soc. Jpn. Ser. II* 44 (5) (1966) 237–245.
- [6] R. Asselin, Frequency filter for time integrations, *Mon. Weather Rev.* 100 (6) (1972) 487–490.
- [7] P.D. Williams, A proposed modification to the Robert–Asselin time filter, *Mon. Weather Rev.* 137 (8) (2009) 2538–2546.
- [8] P.D. Williams, The RAW filter: An improvement to the Robert–Asselin filter in semi-implicit integrations, *Mon. Weather Rev.* 139 (6) (2011) 1996–2007.
- [9] P.D. Williams, Achieving seventh-order amplitude accuracy in leapfrog integrations, *Mon. Weather Rev.* 141 (9) (2013) 3037–3051.
- [10] J. Amezcua, P.D. Williams, The composite-tendency Robert–Asselin–Williams (RAW) filter in semi-implicit integrations, *Q. J. R. Meteorol. Soc.* 141 (688) (2015) 764–773.
- [11] P.D. Williams, J.M. Straka, K.M. Kanak, The performance of filtered leapfrog schemes in benchmark simulations, *Q. J. R. Meteorol. Soc.* 148 (743) (2022) 784–808.
- [12] Y. Li, C. Trenchea, A higher-order Robert–Asselin type time filter, *J. Comput. Phys.* 259 (2014) 23–32.
- [13] A. Guzel, C. Trenchea, The Williams step increases the stability and accuracy of the hoRA time filter, *Appl. Numer. Math.* 131 (2018) 158–173.
- [14] R.E. Schlesinger, W. Uccellini, D.R. Johnson, The effects of the Asselin time filter on numerical solutions to the linearized shallow-water wave equations, *Mon. Weather Rev.* 111 (3) (1983) 455–467.
- [15] M. Déqué, D. Cariolle, Some destabilizing properties of the Asselin time filter, *Mon. Weather Rev.* 114 (5) (1986) 880–884.
- [16] E. Cordero, A. Staniforth, A problem with the Robert–Asselin time filter for three-time-level semi-implicit semi-Lagrangian discretizations, *Mon. Weather Rev.* 132 (2) (2004) 600–610.
- [17] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, *SIAM Rev.* 51 (1) (2009) 82–126.
- [18] R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, in: *Cambridge Texts in Applied Mathematics*, Cambridge University Press, 2002.
- [19] J.S. Hesthaven, *Numerical Methods for Conservation Laws*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2018.
- [20] A. Guzel, W. Layton, Time filters increase accuracy of the fully implicit method, *BIT Numer. Math.* 58 (2) (2018) 301–315.

- [21] K.W. Morton, D.F. Mayers, Numerical Solution of Partial Differential Equations: An Introduction, second ed., Cambridge University Press, 2005.
- [22] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, second ed., Society for Industrial and Applied Mathematics, Philadelphia, 2004.
- [23] J.W. Thomas, Numerical Partial Differential Equations: Conservation Laws and Elliptic Equations, Texts in Applied Mathematics, vol. 33, Springer, New York, NY, 2013.
- [24] R. Warming, B. Hyett, The modified equation approach to the stability and accuracy analysis of finite-difference methods, J. Comput. Phys. 14 (2) (1974) 159–179.
- [25] P. Dennis, M. Ehrenberg, F. D., H. Bremer, Varying rate of RNA chain elongation during RRN transcription in *Escherichia coli*, J. Bacteriol. 191 (11) (2009) 3740–3746.