

A high-order implicit–explicit Runge–Kutta type scheme for the numerical solution of the Kuramoto–Sivashinsky equation

H. P. Bhatt & A. Chowdhury

To cite this article: H. P. Bhatt & A. Chowdhury (2020): A high-order implicit–explicit Runge–Kutta type scheme for the numerical solution of the Kuramoto–Sivashinsky equation, International Journal of Computer Mathematics, DOI: [10.1080/00207160.2020.1814262](https://doi.org/10.1080/00207160.2020.1814262)

To link to this article: <https://doi.org/10.1080/00207160.2020.1814262>



Accepted author version posted online: 24 Aug 2020.
Published online: 11 Sep 2020.



Submit your article to this journal [↗](#)



Article views: 15



View related articles [↗](#)



View Crossmark data [↗](#)



A high-order implicit–explicit Runge–Kutta type scheme for the numerical solution of the Kuramoto–Sivashinsky equation

H. P. Bhatt^a and A. Chowdhury^b

^aDepartment of Mathematics, Utah Valley University, Orem, UT, USA; ^bDepartment of Mathematics, Savannah State University, Savannah, GA, USA

ABSTRACT

This manuscript is concerned with the development and the implementation of a numerical scheme to study the spatio-temporal solution profile of the well-known Kuramoto–Sivashinsky equation with appropriate initial and boundary conditions. A fourth-order Runge–Kutta based implicit–explicit scheme in time along with compact higher-order finite difference scheme in space is introduced. The proposed scheme takes full advantage of the method of line (MOL) and partial fraction decomposition techniques, therefore, it just needs to solve two backward Euler-type linear systems at each time step to get the solution. Performance of the scheme is investigated by testing it on some test examples and by comparing numerical results with relevant known schemes. The numerical results showed that the proposed scheme is more accurate and reliable than existing schemes to solve Kuramoto–Sivashinsky equation.

ARTICLE HISTORY

Received 10 July 2020
Revised 27 July 2020
Accepted 14 August 2020

KEYWORDS

Implicit–explicit scheme; Kuramoto–Sivashinsky equation; partial fraction splitting technique; compact finite difference; exponential time differencing

2010 MATHEMATICS

SUBJECT CLASSIFICATION
65

1. Introduction

The nonlinear time-dependent partial differential equations naturally arise from mathematical modelling of chemical, physical or biological problems and these include dispersive and dissipative nonlinear equations. One of the well-known dissipative equations is the Kuramoto–Sivashinsky equation (KSE). The KSE is the equation governing in the frame of the weakly nonlinear approximation for the shape of the free surface of thin film of viscous liquid falling down a vertical plane when the capillary forces are substantial. Kapitza in his pioneering work [20] first investigated the role of viscosity on the capillary flow in thin layers. Benny [6] carried out the boundary-layer simplifications for the viscous case and Homsy [18] developed the long-wave weakly nonlinear approximation whose consistent application allows one to derive the following dimensionless equation for the evolution of the scaled film thickness u , after rescaling the variables

$$u_t + uu_x + \alpha u_{xx} + \beta u_{xxxx} = 0, \quad (1)$$

where α and β are non-zero constants. Equation (1) is usually referred as KSE which is of great fundamental interest just in the same way as its famous counterparts Korteweg–de Vries (KdV) and Burgers equations are. The alternative form of KSE was obtained in [25] while deriving it as a phase equation for the complex Ginzburg–Landau equation for the evolution of reaction fronts. The KSE also describes flame-front instabilities [13,37,38] as well as the dynamics of viscous-fluid films flowing along walls [36,39]. For more details associated with the rich phenomenology of this flow, we

refer the reader to [4,34] for comprehensive reviews of the experimental and theoretical approaches, respectively.

The KSE is a nonlinear evolution equation which is capable of demonstrating chaotic behaviour in both time and space. It contains nonlinearity, fourth-order dissipative term u_{xxxx} and second-order source term u_{xx} (anti-dissipation). In fact, the KSE represents the extreme case when the dispersion is negligible in comparison with dissipation while the KdV equation [23] demonstrates the other extreme situation. The sole nonlinearity in the KSE is the convective term, which is also known as ‘eikonal’ nonlinearity [12,19] in the framework of deriving Equation (1).

Over the years, large numbers of numerical studies have been devoted to the KSE; the readers are referred to the review paper [19] where a study consists of a thorough investigation of different regimes is presented. Among the earlier works, two numerical studies are worth mentioning – one by Frisch *et al.* [16] where a detailed multiple-scale analysis of the KSE with 2π -periodic boundary conditions is presented, and the other by Christov and Bekyarov [11] where a new Fourier–Galerkin method with a complete orthonormal system of functions in $L^2(-\infty, \infty)$ is applied to the solitary wave problem for the KSE. At about the same time (1992 onwards), the attention has also been focused on developing theoretically sound space and time discretization. In [1], Akrivis applied a finite difference scheme to Equation (1) with periodic boundary condition. Akrivis also reported in [2] a consistent numerical approach to solve the KSE by employing a finite element Galerkin method with extrapolated Crank–Nicolson scheme. In both cases, rigorous error analysis has been carried out in order to derive a refined error bound.

Over the last few decades, a computational approach based on orthogonal spline collocation (OSC) method while seeking a numerical solution to the KSE has turned out to be quite popular. It was Manickam *et al.* [29] who first attempted to solve the KSE by using the orthogonal cubic spline collocation method in conjunction with a second-order splitting method. Later, different types of OSC methods, such as quintic B-spline collocation (QBSC) method [30], Septic B-spline collocation (SBSC) method [44] have been successfully implemented to seek a numerical solution to the KSE. Furthermore, a numerical scheme based on the B-spline functions is introduced in [27] for solving the generalized Kuramoto–Sivashinsky Equation (gKSE) where one test example is devoted to discussing the nonlinear stability with convergence for Equation (1) subjected to Gaussian initial condition.

In addition, numerous other methods including the discontinuous Galerkin method [43], the implicit–explicit BDF method [3], radial basis function (RBF) based mesh-free method [42], etc. have been proposed to find the numerical solutions of the KSE. In another study [33] reported very lately, a lattice Boltzmann model for the KSE is modified to achieve an enhanced level of accuracy and stability. Here, the model’s enhanced stability enables one to use larger time increments which is more than enough to compensate the extra computational cost due to high lattice speeds – a substantial improvement over the existing model. Another lattice Boltzmann model (LBM) with the Chapman–Enskog expansion has been proposed for the gKSE in [26] where numerically obtained results are found to be in good agreement with the analytical results. The gKSE is also studied with the aid of Chebyshev spectral collocation methods in [22] where the resultant reduced system of ordinary differential equations have been solved by employing the implicit–explicit BDF method depicted in [3]. Very recently, Mouloud *et al.* [32] have numerically studied the KSE equation with an additional dispersive term, arising in turbulent gas flow over laminar liquid and Singh *et al.* [35] have introduced a compact finite difference scheme in space and optimal four-stages, third-order strong stability preserving (SSP) time-stepping Runge–Kutta scheme to solve the KSE.

1.1. Contribution

In this manuscript, a fourth-order Runge–Kutta based implicit–explicit scheme in time along with a compact fourth-order finite difference scheme in space is proposed to study the spatio-temporal solution profile of the KSE with the periodic and Dirichlet boundary conditions. The proposed scheme takes full benefit of MOL and partial fraction decomposition techniques, therefore, it just needs to

solve two backward Euler-type linear systems at each time step to get the solution. In addition, for the efficient implementation of the scheme it just requires to compute two LU decompositions outside the time loop. Several numerical experiments on the KSE were run in order to study an empirical convergence analysis and the accuracy of the proposed scheme by comparing with other existing schemes. The numerical results exhibit that the proposed scheme provides better accuracy in most of the cases than the existing schemes considered in this manuscript. In addition, the linear truncation error and stability analysis of the proposed scheme is also discussed.

1.2. Organization of the manuscript

The remainder of the paper is organized as follows: in Section 2, compact fourth-order schemes are described to approximate u_x , u_{xx} and u_{xxxx} . In Section 2, the fourth-order Runge–Kutta based implicit–explicit scheme is briefly explained. The linear truncation error and stability analysis of the proposed scheme are discussed in Section 3. In Section 4, numerical experiments are performed on KSE to test the accuracy and reliability of the proposed scheme. The conclusions are presented in Section 5.

2. Fourth-order compact finite differencing schemes

In order to approximate spatial derivatives in the KSE, we partitioned the computational domain $\Omega \times [0, T] = \{(x, t) | a \leq x \leq b, 0 \leq t \leq T\}$ into uniform grids described by the set of nodes $\{(x_i, t_j)\}$, in which $x_i = a + (i - 1)h$, $i = 1, \dots, N + 1$, $h = \frac{(b-a)}{N}$, $t_j = jk$, $j = 0, 1, \dots, M$, and $k = \frac{T}{M}$, where h and k are spatial and temporal step sizes, respectively.

There are several methods which are used to generate a compact finite difference formula to approximate the first-, second- and fourth-order spatial derivatives. The readers are referred to [28] and references therein for more details on how to generate compact finite difference formula (CFDF). In this study, the spatial derivatives in the KSE are approximated by utilizing the following CFDF:

2.1. Approximation of the first derivative with periodic boundary conditions

If u'_i represents an approximation of the first derivative of $u(x)$ at x_i , then an approximation of first derivative may be written as

$$u'_{i-1} + 4u'_i + u'_{i+1} = \frac{3}{h}(u_{i+1} - u_{i-1}), \quad i = 1, \dots, N. \quad (2)$$

The truncation error for formula (2) is $\mathcal{O}(h^4)$.

The matrix representation of the scheme (2) is given as

$$L_1 \mathbf{U}' = M_1 \mathbf{U}, \quad (3)$$

where

$$L_1 = \begin{bmatrix} 4 & 1 & 0 & \cdots & 1 \\ 1 & 4 & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & 1 & 4 & 1 \\ 1 & \cdots & 0 & 1 & 4 \end{bmatrix}_{N \times N}, \quad M_1 = \frac{3}{h} \begin{bmatrix} 0 & 1 & 0 & \cdots & -1 \\ -1 & 0 & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 0 & 1 \\ 1 & \cdots & 0 & -1 & 0 \end{bmatrix}_{N \times N}, \quad \mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix}_{N \times 1}.$$

Hence the fourth-order CFDF with periodic boundary conditions for u_x is given by

$$\mathbf{U}' = L_1^{-1} M_1 \mathbf{U}. \quad (4)$$

2.2. Approximation of the second derivative with periodic boundary conditions

If u''_i represents an approximation of the second derivative of $u(x)$ at x_i , then an approximation of second derivatives of $u(x)$ may be written as

$$u''_{i-1} + 10u''_i + u''_{i+1} = \frac{12}{h^2}(u_{i-1} - 2u_i + u_{i+1}), \quad i = 1, 2, \dots, N. \quad (5)$$

The matrix representation of the scheme (5) is given as

$$L_2 \mathbf{U}'' = M_2 \mathbf{U}, \quad (6)$$

where

$$L_2 = \begin{bmatrix} 10 & 1 & 0 & \cdots & 1 \\ 1 & 10 & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & 1 & 10 & 1 \\ 1 & \cdots & 0 & 1 & 10 \end{bmatrix}_{N \times N}, \quad M_2 = \frac{12}{h^2} \begin{bmatrix} -2 & 1 & 0 & \cdots & 1 \\ 1 & -2 & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & 1 & -2 & 1 \\ 1 & \cdots & 0 & 1 & -2 \end{bmatrix}_{N \times N}.$$

Hence the fourth-order CFDF with periodic boundary conditions for u_{xx} is given by

$$\mathbf{U}'' = L_2^{-1} M_2 \mathbf{U}. \quad (7)$$

2.3. Approximation of the fourth derivative with periodic boundary conditions

If $u^{(iv)}_i$ represents an approximation of the fourth derivative of $u(x)$ at x_i , then an approximation of fourth derivative of $u(x)$ may be obtained by replacing $\mathbf{U}$ in (6) by $\mathbf{U}''$, that is

$$L_2 \mathbf{U}^{(iv)} = M_2 \mathbf{U}'',$$

where L_2 and M_2 are coefficient matrices defined above.

Hence the fourth-order CFDF with periodic boundary conditions for u_{xxxx} is given by

$$\mathbf{U}^{(iv)} = L_2^{-2} M_2^2 \mathbf{U}. \quad (8)$$

2.4. Approximation of first derivative with Dirichlet boundary conditions

In this case, uniform grid $x_i = a + (i-1)h$, $i = 1, \dots, N$, $h = \frac{(b-a)}{N-1}$ is assumed. The standard compact finite difference formula for first derivatives of $u(x, t)$ at interior points is

$$u'_{i-1} + 4u'_i + u'_{i+1} = \frac{3}{h}(u_{i+1} - u_{i-1}), \quad i = 2, \dots, N-1, \quad (9)$$

where $u_i \approx u(x_i)$ and $u'_i \approx \frac{du(x_i)}{dx}$. The truncation error for formula (9) is $\mathcal{O}(h^4)$. At boundary, when $i = 1$ the formula is

$$4u'_1 + 12u'_2 = \frac{3}{h} \left(-\frac{34}{9}u_1 + 2u_2 + 2u_3 - \frac{2}{9}u_4 \right), \quad (10)$$

and when $i = N$, the formula is

$$4u'_N + 12u'_{N-1} = \frac{3}{h} \left(\frac{34}{9}u_N - 2u_{N-1} - 2u_{N-2} + \frac{2}{9}u_{N-3} \right). \quad (11)$$

The truncation errors in the formula (10) and (11) are $\mathcal{O}(h^4)$.

We can write (9)–(11) into matrix form as

$$\mathcal{L}_1 \mathbf{U}' = \mathcal{M}_1 \mathbf{U}, \quad (12)$$

where

$$\mathcal{L}_1 = \begin{bmatrix} 4 & 12 & 0 & \cdots & 0 \\ 1 & 4 & 1 & \vdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & 1 & 4 & 1 \\ 0 & \cdots & 0 & 12 & 4 \end{bmatrix}_{N \times N}, \quad \mathcal{M}_1 = \frac{3}{h} \begin{bmatrix} -\frac{34}{9} & 2 & 2 & -\frac{2}{9} & 0 & \cdots & 0 \\ -1 & 0 & 1 & 0 & 0 & & \vdots \\ 0 & -1 & 0 & 1 & 0 & \ddots & \vdots \\ 0 & 0 & -1 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \frac{2}{9} & -2 & -2 & \frac{34}{9} \end{bmatrix}_{N \times N}.$$

Hence the fourth-order CFDF with Dirichlet boundary conditions for u_x is given by

$$\mathbf{U}' = (\mathcal{L}_1)^{-1} \mathcal{M}_1 \mathbf{U}. \quad (13)$$

2.5. Approximation of second derivative with Dirichlet boundary conditions

The standard compact finite difference formula for second derivatives of $u(x, t)$ at interior points is

$$u''_{i-1} + 10u''_i + u''_{i+1} = \frac{12}{h^2}(u_{i-1} - 2u_i + u_{i+1}), \quad i = 2, 3, \dots, N-1, \quad (14)$$

where $u''_i \approx \frac{d^2 u(x_i)}{dx^2}$. The truncation error for formula (14) is $\mathcal{O}(h^4)$.

At boundary, when $i = 1$, apply

$$10u''_1 + 100u''_2 = \frac{12}{h^2} \left(\frac{725}{72}u_1 - \frac{190}{9}u_2 + \frac{145}{12}u_3 - \frac{10}{9}u_4 + \frac{5}{72}u_5 \right), \quad (15)$$

and when $i = N$,

$$10u''_N + 100u''_{N-1} = \frac{12}{h^2} \left(\frac{725}{72}u_N - \frac{190}{9}u_{N-1} + \frac{145}{12}u_{N-2} - \frac{10}{9}u_{N-3} + \frac{5}{72}u_{N-4} \right). \quad (16)$$

The truncation error in both of the formulae is also $\mathcal{O}(h^4)$.

Writing (14)–(16) in matrix form yields

$$\mathcal{L}_2 \mathbf{U}'' = \mathcal{M}_2 \mathbf{U}, \quad (17)$$

where

$$\mathcal{L}_2 = \begin{bmatrix} 10 & 100 & 0 & \cdots & 0 \\ 1 & 10 & 1 & \vdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & 1 & 10 & 1 \\ 0 & \cdots & 0 & 100 & 10 \end{bmatrix}_{N \times N},$$

$$\mathcal{M}_2 = \frac{12}{h^2} \begin{bmatrix} \frac{725}{72} & -\frac{190}{9} & \frac{145}{12} & -\frac{10}{9} & \frac{5}{72} & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & & \vdots \\ 0 & 1 & -2 & 1 & 0 & \ddots & 0 \\ & \vdots & \vdots & \vdots & & & \\ \vdots & & 0 & 1 & -2 & 1 & 0 \\ & & 0 & 0 & 1 & -2 & 1 \\ 0 & \cdots & \frac{5}{72} & -\frac{10}{9} & \frac{145}{12} & -\frac{190}{9} & \frac{725}{72} \end{bmatrix}_{N \times N}.$$

Hence the fourth-order CFDF with Dirichlet boundary conditions for u_{xx} is given by

$$\mathbf{U}'' = (\mathcal{L}_2)^{-1} \mathcal{M}_2 \mathbf{U}. \quad (18)$$

2.6. Approximation of the fourth derivative with Dirichlet boundary conditions

If $u_i^{(iv)}$ represents an approximation of the fourth derivative of $u(x)$ at x_i , then an approximation of the fourth derivative of $u(x)$ may be obtained by replacing $\mathbf{U}$ in (17) by $\mathbf{U}''$, that is

$$\mathcal{L}_2 \mathbf{U}^{(iv)} = \mathcal{M}_2 \mathbf{U}'', \quad (19)$$

where $\mathcal{L}_2$ and $\mathcal{M}_2$ are coefficient matrices defined above.

Hence the fourth-order CFDF with Dirichlet boundary conditions for u_{xxxx} is given by

$$\mathbf{U}^{(iv)} = \left((\mathcal{L}_2)^{-1} \mathcal{M}_2 \right)^2 \mathbf{U}. \quad (20)$$

3. Fourth-order time-stepping scheme

This section presents a brief derivation procedure of the fourth-order implicit–explicit (IMEX4) scheme based on a fourth-order exponential time differencing Runge–Kutta (ETDRK4-B) [24] time integrator to solve the following equation with periodic boundary conditions:

$$\frac{\partial u}{\partial t} + \mathcal{L}u = \mathcal{F}(u, t), \quad u(x, 0) = g(x), \quad (21)$$

where $\mathcal{L}$ and $\mathcal{F}$ represent the linear and the nonlinear operators of the KSE, respectively. Approximating $\mathcal{L}u = \alpha u_{xx} + \beta u_{xxxx}$ with schemes (7) and (8), and $\mathcal{F}(u, t)$ with scheme (4), we arrive at the following system of ODEs:

$$\frac{\partial \mathbf{U}}{\partial t} + L\mathbf{U} = F(\mathbf{U}, t), \quad (22)$$

where $L = L_2^{-2}(\alpha L_2 M_2 + \beta M_2^2)$ and $F(\mathbf{U}, t) = -\frac{1}{2} L_1^{-1} M_1 \mathbf{U}^2$. Here, L is a sparse matrix having only a few diagonals occupied with non-zero elements and easily generated by using ‘spdiags’, an inbuilt function in Matlab for the computational purpose. The term $F(\mathbf{U})$ in (22) is a nonlinear term which we do not want to integrate implicitly because the Jacobian of $F(\mathbf{U})$ is non-symmetric and non-definite, therefore an iterative solution of the implicit equation is desired. To avoid this issue, the simplest solution would be integrating $F(\mathbf{U})$ explicitly. The linear term $L\mathbf{U}$ in general is a stiff term which should be integrated implicitly to avoid sufficiently small time steps. Hence, for problems of the form (22), it usually makes sense to integrate $L\mathbf{U}$ implicitly and $F(\mathbf{U})$ explicitly. In order to achieve that, we obtain the following recurrence formula by using the variation of constant formula by letting $k = t_{n+1} - t_n$:

$$\mathbf{U}_{n+1} = e^{-kL} \mathbf{U}_n + k \int_0^1 e^{-kL(1-\tau)} F(\mathbf{U}(t_n + \tau k), t_n + \tau k) d\tau, \quad (23)$$

where $\mathbf{U}_n = \mathbf{U}(t_n)$. Expression (23) is an exact solution of system (22) and the approximation of its integral term leads to various ETD schemes. This paper considers a popular ETD scheme of Runge–Kutta type, namely ETDRK4-B [24] and presents its implicit–explicit version for its general applicability.

The ETDRK4-B [24] scheme is given as

$$\begin{aligned}\mathbf{U}_{n+1} = & \varphi_0(kL)\mathbf{U}_n + k\varphi_1(kL)F_n + k\varphi_2(kL) \left(-3F_n + 2F_n^a + 2F_n^b - F_n^c \right) \\ & + 4k\varphi_3(kL) \left(F_n - F_n^a - F_n^b + F_n^c \right),\end{aligned}\quad (24)$$

where

$$\begin{aligned}F_n &= F(\mathbf{U}_n, t_n), \quad F_n^a = F\left(\mathbf{a}_n, t_n + \frac{k}{2}\right), \quad F_n^b = F\left(\mathbf{b}_n, t_n + \frac{k}{2}\right) \quad \text{and} \quad F_n^c = F(\mathbf{c}_n, t_n + k), \\ \mathbf{a}_n &= \varphi_0(kL/2)\mathbf{U}_n + \frac{k}{2}\varphi_1(kL/2)F_n, \\ \mathbf{b}_n &= \varphi_0(kL/2)\mathbf{U}_n + \frac{k}{2}\varphi_1(kL/2)F_n + k\varphi_2(kL/2) \left(F_n^a - F_n \right), \\ \mathbf{c}_n &= \varphi_0(kL)\mathbf{U}_n + k\varphi_1(kL)F_n + 2k\varphi_2(kL) \left(F_n^b - F_n \right).\end{aligned}$$

Note that the scheme (24) contains matrix functions of the form

$$\varphi_0(kL) = e^{-kL}, \quad \varphi_\mu(kL) = (-kL)^{-\mu} \left(e^{-kL} - \sum_{j=0}^{\mu-1} \frac{(-kL)^j}{j!} \right), \quad \mu = 1, 2, 3. \quad (25)$$

If eigenvalues of L are close to zero, the direct computation of φ functions in (25) is a challenging problem in numerical analysis due to disastrous cancellation error in the computation. To overcome this and other numerical issues associated with it, many researchers have proposed different techniques to compute these φ functions. Some of the well-known techniques include (i) the rational approximation of $\varphi_\mu(kL)$ [8,9], (ii) the Krylov subspace method [17,41], and (iii) the polynomial approximation of $\varphi_\mu(kL)$ [10,40].

This study focuses on the rational approximation of $\varphi_\mu(kL)$ and introduces implicit–explicit version of ETDRK4-B schemes utilizing a partial fraction decomposition technique is given in [8]. In order to alleviate computational difficulties associated in direct computation of $\varphi_\mu(kL)$, at first a fourth-order (2, 2)-Padé approximation to e^{-kL} is utilized, which helps to avoid the direct computations of the matrix exponential and higher powers of a matrix inverse. In addition, another advantage we found in utilizing (2, 2)-Padé approximation is that the factors L^{-1} and L^{-3} cancel out in ETDRK4-B scheme.

Implementation of (2, 2)-Padé approximation $R_{2,2}(kL) = (12I + 6kL + k^2L^2)^{-1}(12I - 6kL + k^2L^2)$ into (24) to approximate matrix exponential functions yields:

$$\begin{aligned}\mathbf{U}_{n+1} = & R_{2,2}(kL)\mathbf{U}_n + P_1(kL)F_n + P_2(kL) \left(-3F_n + 2F_n^a + 2F_n^b - F_n^c \right) \\ & + P_3(kL) \left(F_n - F_n^a - F_n^b + F_n^c \right),\end{aligned}\quad (26)$$

where

$$\begin{aligned}P_1(kL) &= 12k(12I + 6kL + k^2L^2)^{-1}, \\ P_2(kL) &= k(6I + kL)(12I + 6kL + k^2L^2)^{-1},\end{aligned}$$

$$P_3(kL) = 2k(4I + kL)(12I + 6kL + k^2L^2)^{-1}.$$

In addition

$$\begin{aligned} \mathbf{a}_n &= \tilde{R}_{2,2}(kL)\mathbf{U}_n + \tilde{P}_1(kL)F_n, \\ \mathbf{b}_n &= \tilde{R}_{2,2}(kL)\mathbf{U}_n + \tilde{P}_1(kL)F_n + \tilde{P}_2(kL)(F_n^a - F_n), \\ \mathbf{c}_n &= R_{2,2}(kL)\mathbf{U}_n + P_1(kL)F_n + 2P_2(kL)(F_n^b - F_n), \end{aligned}$$

with

$$\begin{aligned} \tilde{R}_{2,2}(kL) &= (48I - 12kL + k^2L^2)(48I + 12kL + k^2L^2)^{-1}, \\ \tilde{P}_1(kL) &= 24k(48I + 12kL + k^2L^2)^{-1}, \\ \tilde{P}_2(kL) &= 2k(12I + kL)(48I + 12kL + k^2L^2)^{-1}. \end{aligned}$$

3.1. Fourth-order implicit-explicit Runge-Kutta type scheme

Since the scheme (26) consists of high-order matrix polynomials to invert, its direct implementation would be computationally burdensome and numerically unstable if the matrices have high condition numbers. In addition, the round-off error while computing the power of the matrices can produce bad approximations [31]. In order to overcome this difficulty, $R_{2,2}(kL)$ and $\tilde{R}_{2,2}(kL)$ will not be computed directly. Instead, the problem of stably computing the inverse of matrix polynomials inherent in (26) is handled by utilizing a partial fraction decomposition technique as suggested in [8]. This decomposition does reduce the computational complexity to just two LU decompositions over the entire time interval (provided space step h , and the time step k are held constant). In addition, this decomposition approach is important in alleviating ill-conditioning problems because only implicit Euler-type solvers are required. A description of the scheme upon implementing a partial fraction decomposition technique is presented in the following algorithm and from this point the algorithm is referenced as IMEXRK4 scheme. We named it IMEXRK4 because it solves the linear term in (22) implicitly and the nonlinear term explicitly by using Runge-Kutta methods.

In order to implement this IMEXRK4 scheme, poles and corresponding weights were computed for $R_{2,2}(kL)$, $\{P_i(kL)\}_{i=1}^3$, $\tilde{R}_{2,2}(kL)$, and $\{\tilde{P}_i(kL)\}_{i=1}^2$ by using Maple. We called the Maple sequence ‘Convert(f , parfrac, K)’, where f is rational function, ‘parfrac’ represents partial fraction, and K is real or complex to compute the following poles and corresponding weights:

$$\begin{aligned} c_1 &= -3.0 + i1.7320508075688772935, & w_1 &= -6.0 - i10.39230484541326376, \\ w_{11} &= -i3.4641016151377545871, & w_{21} &= 0.5 - i0.8660254037844386467, \\ w_{31} &= 1.0 - i0.57735026918962576452, & \tilde{c}_1 &= 2c_1, \\ \tilde{w}_1 &= 2w_1, & \tilde{\Omega}_1 &= w_{11}, & \tilde{\Omega}_2 &= 2w_{21} \quad . \end{aligned}$$

Remark: Since the matrices $(kL - c_1I)$ and $(kL - \tilde{c}_1I)$ are diagonally dominant, we do not require Gauss elimination for the LU decomposition. We can utilize the well-known Thomas algorithm directly for their LU decomposition. For the efficient implementation of the Algorithm 1, LU decomposition of the matrices are carried and stored outside the time loop. Here, we have illustrated how to implement the steps given in the Algorithm 1 efficiently by showing the work involved in step 1 as an example

$$\begin{aligned} (kL - \tilde{c}_1I)R_a &= \tilde{w}_1\mathbf{U}_n + k\tilde{\Omega}_1F_n, \\ (\alpha L_2M_2 + \beta M_2^2 - \tilde{c}_1L_2^2)R_a &= M_2^2(\tilde{w}_1\mathbf{U}_n + k\tilde{\Omega}_1F_n). \end{aligned}$$

Algorithm 1 IMEXRK4 scheme.**Step 1:**

Solve the linear system $(kL - \tilde{c}_1 I)R_a = \tilde{w}_1 \mathbf{U}_n + k\tilde{\Omega}_1 F_n$.

Define $\mathbf{a}_n = \mathbf{U}_n + 2 \operatorname{Re}(R_a)$,

and $F_n^{\mathbf{a}} = F\left(\mathbf{a}_n, t_n + \frac{k}{2}\right)$.

Step 2:

Solve the linear system $(kL - \tilde{c}_1 I)R_b = \tilde{w}_1 \mathbf{U}_n + k(\tilde{\Omega}_1 - \tilde{\Omega}_2)F_n + k\tilde{\Omega}_2 F_n^{\mathbf{a}}$,

Define $\mathbf{b}_n = \mathbf{U}_n + 2 \operatorname{Re}(R_b)$,

and $F_n^{\mathbf{b}} = F\left(\mathbf{b}_n, t_n + \frac{k}{2}\right)$.

Step 3:

Solve the linear system $(kL - c_1 I)R_c = w_1 \mathbf{U}_n + k(w_{11} - 2w_{21})F_n + 2kw_{21}F_n^{\mathbf{b}}$,

Define $\mathbf{c}_n = \mathbf{U}_n + 2 \operatorname{Re}(R_c)$,

and $F_n^{\mathbf{c}} = F(\mathbf{c}_n, t_n + k)$.

Step 4:

Solve the linear system $(kL - c_1 I)R_u = w_1 \mathbf{U}_n + k(w_{11} - 3w_{21} + w_{31})F_n$
 $+ k(2w_{21} - w_{31})(F_n^{\mathbf{a}} + F_n^{\mathbf{b}}) - k(w_{21} - w_{31})F_n^{\mathbf{c}}$.

Evaluate $\mathbf{U}_{n+1} = \mathbf{U}_n + 2 \operatorname{Re}(R_u)$.

To solve the above system we computed the LU decomposition of the matrix $(k(\alpha L_2 M_2 + \beta M_2^2) - \tilde{c}_1 L_2^2)$ by utilizing the Thomas algorithm.

4. Linear analysis

The linear truncation error and stability analysis of the scheme (26) are presented in this section.

4.1. Truncation error analysis

It is obvious that the overall spatial discretization is of order four because a fourth-order CFDF is applied to linear parts of the KSE. To analyse the overall local temporal truncation error of the scheme (26) for KSE, the following linear semi-discretization system is considered:

$$\frac{\partial \mathbf{U}}{\partial t} = -L\mathbf{U} + R\mathbf{U}, \quad (27)$$

where L represents matrices derived from the linear spatial discretization of the second and the fourth-order spatial derivatives, R represents the linear spatial discretization of first-order derivative term of a linear KSE, and $\mathbf{U}$ is a vector of unknowns.

Applying scheme (26) to Equation (27) yields

$$\begin{aligned} \mathbf{U}_{n+1} = & \left(I + \frac{kL}{2} + \frac{k^2 L^2}{12} \right)^{-1} \left(\left(I - \frac{kL}{2} + \frac{k^2 L^2}{12} + Rk \right) \mathbf{U}_n \right. \\ & + \frac{Rk}{2} \left(I + \frac{kL}{6} \right) (-3 + 2\mathbf{a}_n + 2\mathbf{b}_n - \mathbf{c}_n) \\ & \left. + \frac{2Rk}{3} \left(I + \frac{kL}{4} \right) (1 - \mathbf{a}_n - \mathbf{b}_n + \mathbf{c}_n) \right), \end{aligned} \quad (28)$$

where

$$\begin{aligned} \mathbf{a}_n &= \left(I + \frac{1}{4}kL + \frac{1}{48}k^2 L^2 \right)^{-1} \left(I - \frac{1}{4}kL + \frac{1}{48}k^2 L^2 + \frac{1}{2}Rk \right) \mathbf{U}_n, \\ \mathbf{b}_n &= \left(I + \frac{1}{4}kL + \frac{1}{48}k^2 L^2 \right)^{-1} \left(\left(I - \frac{1}{4}kL + \frac{1}{48}k^2 L^2 + \frac{1}{2}Rk \right) \mathbf{U}_n \right. \\ & \quad \left. + \frac{1}{2}Rk \left(I + \frac{1}{12}kL \right) (\mathbf{a}_n - \mathbf{U}_n) \right), \\ \mathbf{c}_n &= \left(I + \frac{1}{2}kL + \frac{1}{12}k^2 L^2 \right)^{-1} \left(\left(I - \frac{1}{2}kL + \frac{1}{12}k^2 L^2 + Rk \right) \mathbf{U}_n + Rk \left(I + \frac{1}{6}kL \right) (\mathbf{b}_n - \mathbf{U}_n) \right). \end{aligned}$$

Taylor expansion of Equation (28) yields

$$\begin{aligned} \mathbf{U}_{n+1} = & \left(I + (R-L)k + \left(\frac{L^2}{2} - LR + \frac{R^2}{2} \right) k^2 + \left(\frac{L^2 R}{2} - \frac{LR^2}{2} - \frac{L^3}{6} + \frac{R^3}{6} \right) k^3 \right. \\ & \left. + \left(\frac{L^4}{24} + \frac{L^2 R^2}{4} - \frac{LR^3}{6} - \frac{L^3 R}{6} + \frac{R^4}{24} \right) k^4 + \dots \right) \mathbf{U}_n. \end{aligned} \quad (29)$$

Since the exact solution of Equation (27) is

$$\mathbf{U}(t_{n+1}) = e^{(R-L)k} \mathbf{U}(t_n),$$

the local truncation error of the scheme (26) is

$$\begin{aligned} \mathbf{e}_{n+1} = & \left(I + (R-L)k + \left(\frac{L^2}{2} - RL + \frac{R^2}{2} \right) k^2 + \left(\frac{RL^2}{2} - \frac{LR^2}{2} - \frac{L^3}{6} + \frac{R^3}{6} \right) k^3 \right. \\ & \left. + \left(\frac{L^4}{24} + \frac{L^2 R^2}{4} - \frac{LR^3}{6} - \frac{L^3 R}{6} + \frac{R^4}{24} \right) k^4 + \dots \right) \mathbf{U}_n - e^{(R-L)k} \mathbf{U}(t_n) = \mathcal{O}(k^5). \end{aligned} \quad (30)$$

Hence the scheme (26) is fourth-order in time discretization.

4.2. Stability analysis

4.2.1. Amplification symbol

In this section, the behaviour of the fourth-order (2, 2)-Padé approximation $R_{2,2}(z) = \frac{12-6z+z^2}{12+6z+z^2}$ is compared to e^{-z} and illustrated in Figure 1.

Definition: A rational approximation $R_{r,s}(z)$ of the exponential e^z is said to be A-acceptable if $|R_{r,s}(z)| < 1$, whenever $\Re(z) < 0$ and L-acceptable if, in addition $|R_{r,s}(z)| \rightarrow 0$ as $\Re(z) \rightarrow -\infty$.

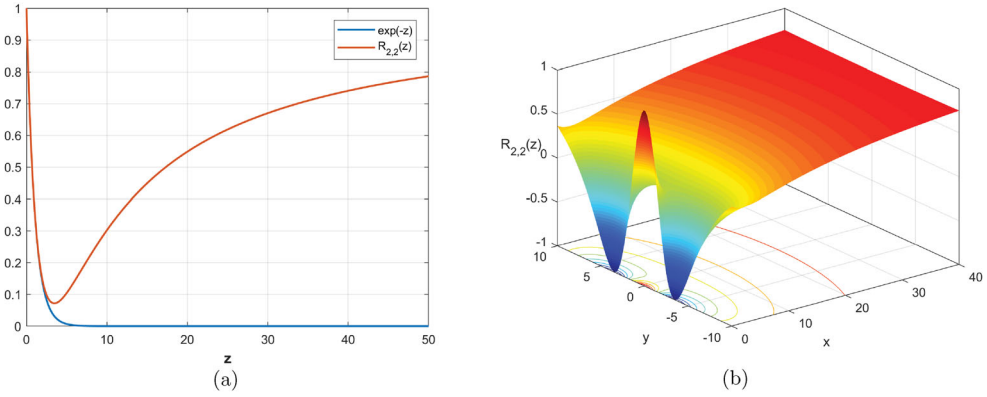


Figure 1. (a) Behaviour of function e^{-z} , and $R_{2,2}(z)$ for $z \in [0, 50]$. (b) Surface plot of the behaviour of function $R_{2,2}(z)$ for $z \in [0, 40] \times [-10, 10]$.

4.2.2. Stability regions

The linear stability of the scheme (26) was analysed by plotting its stability regions (see in [7] and references therein) for the nonlinear autonomous ODE:

$$u_t = -cu + F(u), \quad (31)$$

where $F(u)$ is a nonlinear part and c represents the approximation of the linear parts of KSE. Let us assume that there exists a fixed point u_0 such that $-cu_0 + F(u_0) = 0$. Linearizing about this fixed point, we thus obtain

$$u_t = -cu + \gamma u, \quad (32)$$

where u is the perturbation of u_0 , and $\gamma = F'(u_0)$. If (31) represents a system of ODEs, then γ is a diagonal or a block diagonal matrix containing the eigenvalues of F . To keep the fixed point stable, we need that $\text{Re}(\gamma - c) < 0$, for all γ (see [14]). This approach only provides an indication to how stable a numerical method is, since in general, one cannot linearize both terms simultaneously [24].

In general, the parameters c and γ may both be complex-valued. The stability region of the scheme (26) is four-dimensional and therefore difficult to represent [14]. The two-dimensional stability region is obtained, if both c and γ are purely imaginary or purely real [15], or if γ is complex and c is fixed and real [7].

Utilization of the scheme (26) to the linearized Equation (32) leads to a recurrence relation involving u_n , and u_{n+1} . By letting $r = \frac{u_{n+1}}{u_n}$, $x = \gamma k$, and $y = -ck$, we come up with the following amplification factor (Figure 1):

$$r(x, y) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4, \quad (33)$$

where

$$\begin{aligned} c_0 &= 1 + y + \frac{1}{2}y^2 + \frac{1}{6}y^3 + \frac{1}{24}y^4 + \frac{1}{144}y^5 + O(y^6), \\ c_1 &= 1 + y + \frac{1}{2}y^2 + \frac{1}{6}y^3 + \frac{7}{192}y^4 + \frac{5}{2304}y^5 + O(y^6), \\ c_2 &= \frac{1}{2} + \frac{1}{2}y + \frac{1}{4}y^2 + \frac{23}{288}y^3 + \frac{11}{768}y^4 - \frac{37}{27648}y^5 + O(y^6), \\ c_3 &= \frac{1}{6} + \frac{1}{6}y + \frac{47}{576}y^2 + \frac{77}{3456}y^3 + \frac{5}{9216}y^4 - \frac{515}{165888}y^5 + O(y^6), \end{aligned}$$

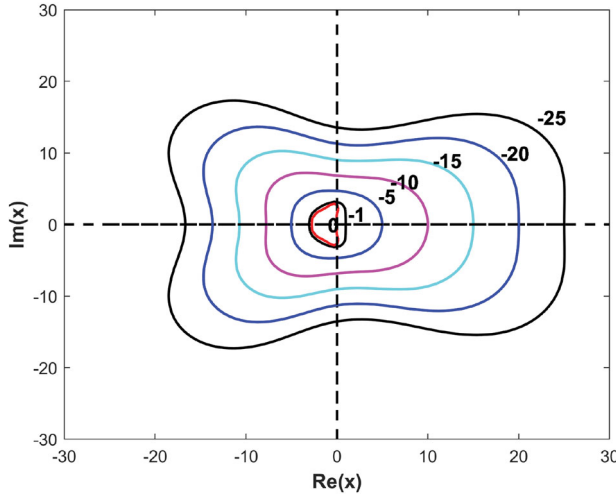


Figure 2. Stability regions for different values of $\gamma \in \mathbb{R}^-$.

$$c_4 = \frac{1}{24} + \frac{1}{32}\gamma + \frac{35}{3456}\gamma^2 - \frac{1}{13824}\gamma^3 - \frac{169}{82944}\gamma^4 - \frac{457}{331776}\gamma^5 + O(\gamma^6).$$

The boundaries of the stability regions of the scheme (26) are obtained by substituting $r = e^{i\theta}$, $\theta \in [0, 2\pi]$ into Equation (33) and solving for x , but unfortunately we do not know the explicit expression for $|r(x, y)| = 1$. We will only be able to plot it and in this paper, we have plotted the stability regions for the two cases. At first, this study focuses on the case where γ is complex and c is fixed and real. The analysis begins by selecting several real negative values of γ and looking for a region of stability in the complex x plane where $|r(x, y)| = 1$. The corresponding families of stability regions of the scheme (26) in the complex x are plotted in Figure 2. According to Beylkin *et al.* [7], for scheme (26) to be applicable, it is important that stability regions grow as $\gamma \rightarrow -\infty$. As we can see in Figure 2, the stability regions for the scheme grow larger as $\gamma \rightarrow -\infty$. These regions give an indication of the stability of the proposed scheme.

In the second case, we assume γ is complex and c is purely imaginary and stability regions for different values of $\gamma = -5i, 5i, -20i$, and $20i$ are depicted in Figure 3(a-d).

5. Numerical experiments and discussions

In this section, we present the results of extensive numerical experiments carried out by implementation of the proposed scheme on four test problems in order to demonstrate the efficiency and accuracy of the scheme. All the numerical experiments are conducted in MATLAB 9.3 platforms based on an Intel Core i52410M 2.30 GHz workstation. The accuracy of the scheme is measured in terms of maximum norm errors $\|\cdot\|_\infty$ and global relative error (GRE) at the final time $t = T$ which are defined as

$$\|\cdot\|_\infty = \max_{1 \leq i \leq N} |u(x_i, T) - U_i^M|,$$

$$\text{GRE} = \frac{\sum_i |u(x_i, T) - U_i^M|}{\sum_i |u(x_i, T)|},$$

where u and U are the exact and numerical solutions, respectively.

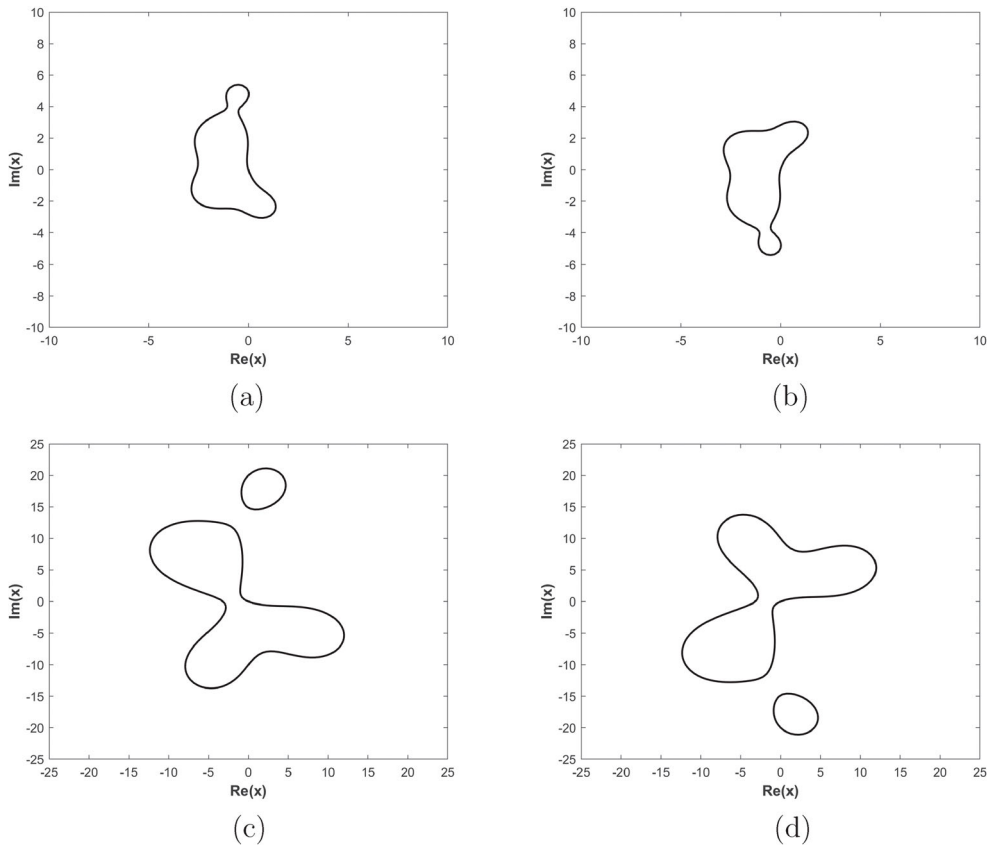


Figure 3. Stability regions for different values of $y = -5i, 5i, -20i$, and $20i$. (a) $y = -5i$. (b) $y = 5i$. (c) $y = -20i$. (d) $y = 20i$.

When the exact solution of the considered problem(s) is/are available, we compute the spatial convergence rate with

$$\text{order} = \frac{\log_{10} (\|u - U_h\|_{\infty} / \|u - U_{h/2}\|_{\infty})}{\log_{10}(2)}$$

and the temporal convergence rate with

$$\text{order} = \frac{\log_{10} (\|u - U_k\|_{\infty} / \|u - U_{k/2}\|_{\infty})}{\log_{10}(2)},$$

where $\|u - U_h\|_{\infty}$ and $\|u - U_{h/2}\|_{\infty}$ are the maximum error norms with spatial step sizes equal to h and $h/2$, respectively. Similar description is valid for temporal convergence rate.

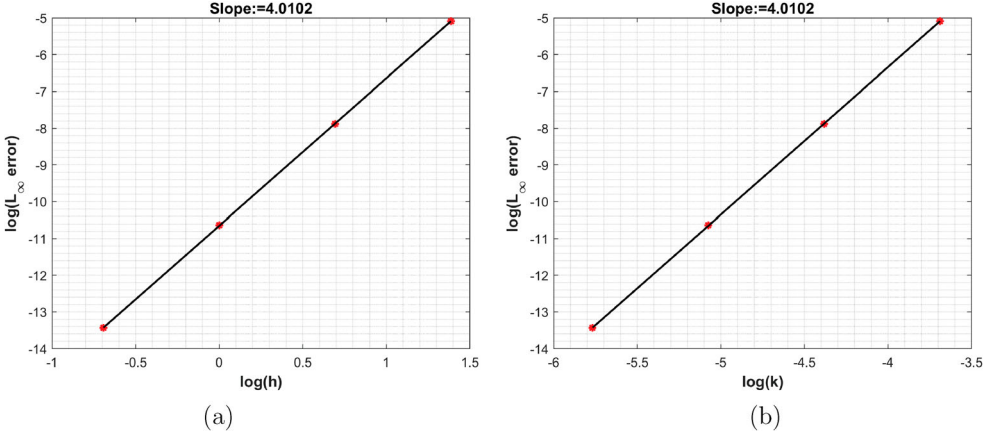
On the other hand, when the exact solution of the problem(s) is/are unavailable, we utilize

$$\text{order} = \frac{\log_{10}(E_k/E_{k/2})}{\log_{10}(2)}, \quad (34)$$

where $E_k = \|U_k - U_{2k}\|_{\infty}$ and $E_{k/2} = \|U_{k/2} - U_k\|_{\infty}$ are the maximum error norms at k and $k/2$, respectively, to measure the temporal convergence rate of the scheme.

Table 1. The maximum error, rates of convergence, and CPU time of IMEXRK4 for Example 5.1 at $T = 2.0$.

h & k	4 & 0.025	2 & 0.0125	1 & 0.00625	0.5 & 0.003125
$\ \cdot\ _\infty$	6.157E-03	3.775E-04	2.396E-05	1.461E-06
Order	—	4.0278	3.9777	4.0359
CPU(s)	0.2004	0.5011	1.6728	8.0996

**Figure 4.** Log-log plots of spatial and time rates of convergence of the proposed method. (a) Spatial rate of convergence. (b) Time rate of convergence.

Example 5.1 (Benchmark Problem): In this example, the KSE with $\alpha = -1$ and $\beta = 1$ over a domain $\Omega = [-50, 50]$, with the analytical solution

$$u(x, t) = \mu + \frac{15 \tanh^3(v(x - \mu t - x_0)) - 45 \tanh(v(x - \mu t - x_0))}{19^{3/2}} \quad (35)$$

is considered.

The initial and boundary conditions are extracted from the exact solution (35). This example is considered as a benchmark problem (solved in [26, 43] among others) in order to investigate the performance in terms of accuracy and efficiency of the proposed method. The parameters in (35) are chosen as $\mu = 5$, $v = \frac{1}{2\sqrt{19}}$ and $x_0 = -25$.

In order to investigate the order of accuracy and computational efficiency of the proposed scheme IMEXRK4 for solving the KSE, a numerical test on Example 5.1 was performed. In the computation, a simulation was run up to $T = 2.0$ by initially setting $h = 4$ and $k = 0.025$, then by reducing both of them by a factor of 2 in each refinement. The maximum error and rates of convergence are listed in Table 1. As can be seen from Table 1 that the computed convergence rates of the proposed scheme apparently demonstrate the expected fourth-order accuracy in both time and space.

In order to visualize the space and time rates of convergence of the proposed scheme, we illustrated them in Figure 4 with log-log scale graph. From Figure 4, it can be seen that the slopes of the regression line for maximum errors both in time and spatial directions are close to four, which corresponds to the fourth-order scheme both in time and space.

We ran another sets of experiment in Example 5.1 with $h = 0.5$, $k = 0.01$ until time $T = 10$ and captured the 3D view of the solution profile in Figure 5(a). From Figure 5(a, b), we can see that the solution obtained via using the proposed scheme is close to the exact solution.

In Table 2, we compare the scheme IMEXRK4 with other existing schemes: SBSC [44], QBSC [30], and LBM [26] by listing GRE of $u(x, t)$ with $N = 200$, $k = 0.01$ at different time levels $t \leq 12$. Table 2

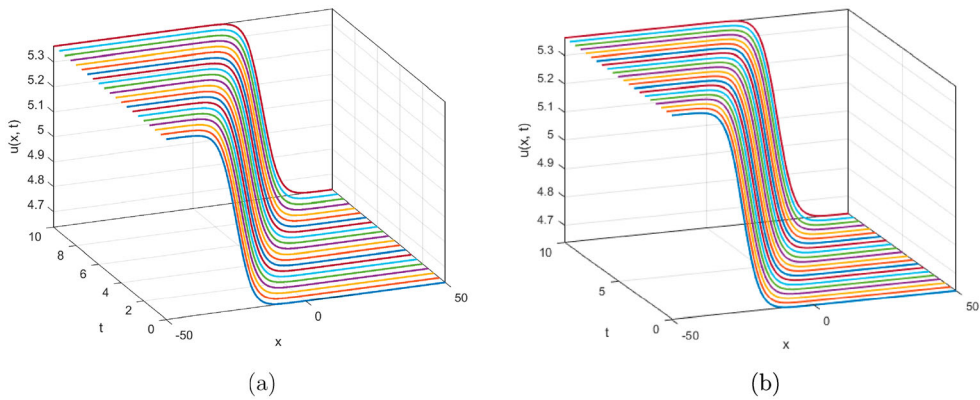


Figure 5. Numerical solution vs. exact solution for various time $t \in [0, 10]$. (a) Numerical solution. (b) Exact solution.

Table 2. Comparison of GRE at different time with $N = 200$ and $k = 0.01$ for Example 5.1.

	Time(t)	6	8	10	12
IMEXRK4	GRE	7.624E-08	8.092E-08	8.589E-08	3.188E-07
	Time(t)	6	8	10	12
SBSC	GRE	1.625E-07	1.940E-07	2.229E-07	5.314E-07
	Time(t)	6	8	10	12
QBSC	GRE	6.509E-06	7.132E-06	7.310E-06	8.776E-06
	Time(t)	6	8	10	12
LBM	GRE	7.881E-06	9.532E-06	1.089E-05	1.179E-05
	Time(t)	6	8	10	12

Table 3. Comparison of GRE at different time with $N = 100$ and $k = 0.01$ for Example 5.1.

	Time(t)	6	8	10	12
IMEXRK4	GRE	7.935E-08	8.444E-08	8.716E-08	9.988E-08
	Time(t)	6	8	10	12
CFDS	GRE	8.434E-08	8.912E-08	9.254E-08	1.692E-07
	Time(t)	6	8	10	12

clearly indicates that the numerical results obtained via proposed scheme are more accurate than that obtained via SBSC [44], QBSC [30] and LBM [26].

In the another set of experiments, we compare the accuracy of the proposed scheme with scheme introduced in [35]. In this experiment, we used the sixth-order compact scheme given in [35] for first-order derivative term and fourth-order scheme for the second and the fourth-order derivative terms in order to compare our results with results given in [35]. From the table, we can note that our proposed scheme yields better accuracy than the scheme introduced in [35].

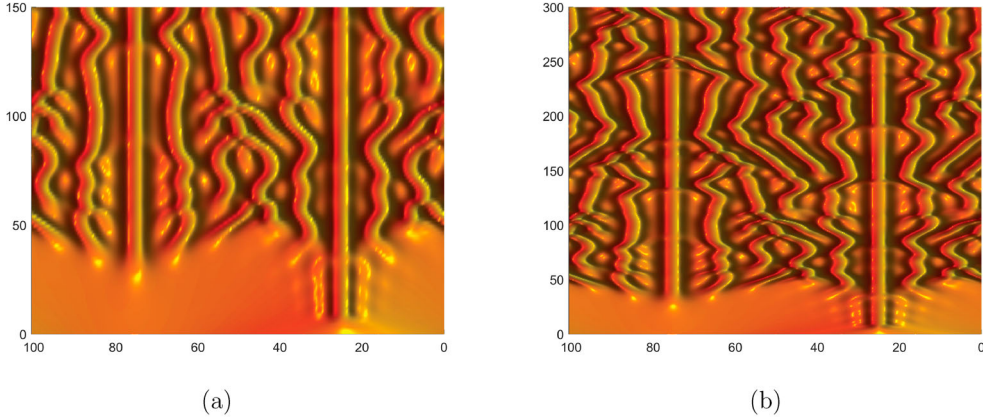
Example 5.2 (The KSE with periodic boundary conditions): In this example, we consider the KSE with $\alpha = 1$ and $\beta = 1$ over a domain $\Omega = [0, 32\pi]$ along with periodic boundary conditions and following initial condition:

$$u(x, 0) = \cos\left(\frac{x}{16}\right) \left(1 + \sin\left(\frac{x}{16}\right)\right). \quad (36)$$

Two sets of numerical experiments on Example 5.2 were conducted. In the first set of the experiment, the order of accuracy in the temporal direction of the proposed method with periodic boundary conditions was measured by running an experiment until time $T = 10.0$ with $N = 256$. Initially, $k = \frac{1}{2}$ was set and repeatedly halved it at each time and numerical results are presented in Table 4. The error values E_k listed for IMEXRK4 scheme in Table 4 are again calculated through a maximal

Table 4. The maximum error, time rates of convergence, and CPU time of IMEXRK4 for Example 5.2 with $N = 256$ at $T = 10.0$.

k	1/4	1/8	1/16	1/32
E_k	9.031E-04	6.291E-05	3.922E-06	2.442E-07
Order	—	3.8436	4.0034	4.0052
CPU(s)	1.1791	2.1269	3.3537	5.2132

**Figure 6.** Chaotic solution profile of the component $u(x, t)$ at various time obtained via IMEXRK4. (a) $t \in [0, 150]$, $N = 256$ and $k = \frac{1}{4}$. (b) $t \in [0, 300]$, $N = 512$ and $k = \frac{1}{8}$.

difference between each simulation. From Table 4, it is clear that the proposed scheme is able to achieve the expected fourth-order accuracy in time with periodic boundary conditions.

To better understand the applicability of the IMEXRK4 scheme while simulating the long-time behaviour of the KSE, a second set of experiments was conducted on Example 5.2 until the long-time $t = 150$ with $N = 256$, $k = \frac{1}{4}$ and $t = 300$ with $N = 512$, $k = \frac{1}{8}$. The chaotic solution profiles of the component $u(x, t)$ for $t \in [0, 150]$ and for $t \in [0, 300]$ were captured in Figure 6(a,b), respectively. Chaotic solution profile corresponds for $t \leq 150$ in the Figure 6(a) are in good agreement with the results depicted in [21] – therefore, we are confident that the profile corresponds for $t \leq 300$ is correct and reliable.

In [5], Asante-Asanami *et al.* argued that for the case involving periodic boundary conditions, the matrices of the form $(kL - \tilde{c}_1 I)$ are cyclic Toeplitz matrices and LU decomposition of such matrices is computationally expensive in comparison to the utilization of Fourier transformation for solving the linear system involving the above mentioned matrix. In order to validate their observation, we ran another set of experiments in Example 5.2 applying the fast Fourier transform and compared the computational efficiency with compact finite difference scheme. We can see from Table 5 that fast Fourier transform is computationally more efficient than the compact finite difference scheme. Therefore, the readers are encouraged to utilize Fourier transform in spatial direction to solve problems involving periodic boundary conditions.

Example 5.3 (The KSE with Gaussian initial condition): Here, we consider the KSE with $\alpha = 1$ and $\beta = 1$ which exhibits chaotic behaviour over a finite domain $\Omega = [-30, 30]$ with homogeneous Dirichlet boundary conditions and the following Gaussian initial condition:

$$u(x, 0) = \exp(-x^2). \quad (37)$$

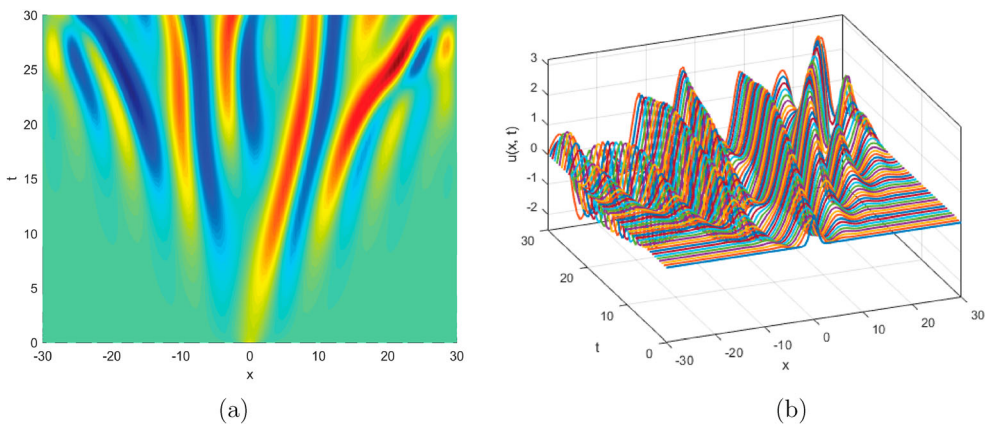
In what follows, again on Example 5.3 two sets of numerical experiments were performed. In the first set of the experiments, the order of accuracy in the temporal direction of the proposed scheme

Table 5. Comparison of computational efficiency with $N = 256$ for Example 5.2.

	k	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
Fourier transform	CPU(s)	0.7093	0.8073	1.2973	1.9206
	k	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
Compact finite difference	CPU(s)	5.8523	12.8448	22.5964	44.4866

Table 6. The maximum error, time rates of convergence, and CPU time of IMEXRK4 for Example 5.3 with $N = 101$ at $T = 1$.

k	0.01/2	0.01/4	0.01/8	0.01/16
E_k	2.723E-08	1.976E-09	1.324E-10	8.613E-12
Order	—	3.7847	3.8995	3.9422
CPU(s)	0.5543	1.0044	1.9298	3.4635

**Figure 7.** The aerial and 3D chaotic solution profile of the KSE with $N = 101$, $k = 0.1$ at $t \in [0, 30]$ for Example 5.3. (a) Aerial view. (b) 3D view.

with homogeneous Dirichlet boundary conditions were examined by running an experiment until time $T = 1.0$ with fixed $N = 101$. Again the error values reported in Table 6 are calculated through a maximal difference between each simulation which is obtained by repeatedly halving an initial time step size $k = \frac{1}{2}$ at each time. From the results, it can be seen that the proposed scheme is able to achieve the expected fourth-order accuracy in time.

In the second sets of experiment, the chaotic behaviour of the component $u(x, t)$ is simulated for Gaussian initial condition. The simulations are accomplished in $t \in [0, 30]$ with the parameters $N = 101$ and $k = 0.1$ and captured in Figure 7 with aerial and 3D views. From Figure 7, it can be clearly seen that the result shows same behaviour as reported in [27,30].

Example 5.4 (The KSE with different values of β): The KSE with $\alpha = 1$ and different values of β over a finite domain $\Omega = [-1, 1]$ along with homogeneous Dirichlet boundary conditions and the following initial condition is considered here:

$$u(x, 0) = -\sin(\pi x). \quad (38)$$

For the empirical convergence analysis in the temporal direction of the proposed scheme, we ran an experiment until time $T = 1.0$ with $\beta = 1.1$ and fixed $h = 0.05$. The error values reported in

Table 7. The maximum error, time rates of convergence, and CPU time of IMEXRK4 for Example 5.4 with $\beta = 1.1$, $h = 0.05$ at $T = 1$.

k	0.005/2	0.005/4	0.005/8	0.005/16
E_k	1.431E-08	9.7926E-10	6.532E-11	3.320E-12
Order	—	3.8692	3.9060	4.2983
CPU(s)	1.0360	2.1329	3.7341	5.6047

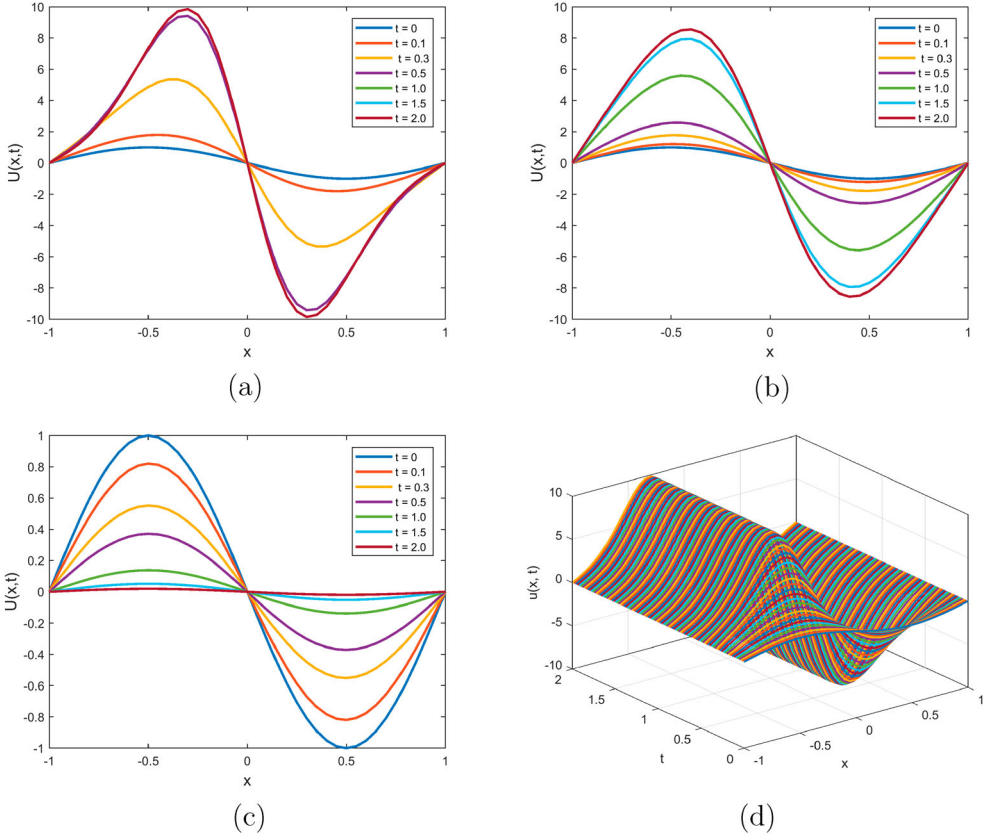
**Figure 8.** Space-time evolution profile of the component $u(x, t)$ obtained via IMEXRK4 scheme with various values of β at different time levels. (a) $\beta = 0.4/\pi^2$. (b) $\beta = 0.6/\pi^2$. (c) $\beta = 0.8/\pi^2$. (d) $\beta = 0.4/\pi^2$ and $t \in [0, 2]$.

Table 7 are computed through a maximal difference between each simulation which is obtained by repeatedly halving an initial time step size $k = 0.005$ at each time. From the results, it can be seen that the proposed scheme is able to exhibit the expected fourth-order accuracy in time.

The space-time evolution profile of $u(x, t)$ via IMEXRK4 scheme with different values of β , $h = 0.05$, $k = 0.001$ at different time levels were depicted in Figure 8. The results in Figure 8 clearly exhibit good agreement with results reported in [29,30].

6. Conclusions

This manuscript introduced a fourth-order scheme both in time and space to solve the KSE. The proposed method utilized a compact fourth-order finite difference scheme for a spatial discretization and the fourth-order Runge–Kutta based implicit–explicit scheme for time discretization. A Compact finite difference scheme is used to transform the KSE to a system of ordinary differential

equations (ODEs) in time, and then, fourth-order time integrator is implemented to solve the resulting ODEs. Calculation of local truncation error and an empirical convergence analysis exhibited the fourth-order accuracy of the proposed scheme. The performance and applicability of the scheme have been investigated by testing it on several test problems. The computed numerical solutions maintain good accuracy compared with the exact solution. In addition, the numerical results exhibited that the proposed scheme provides better accuracy in comparison with other existing schemes.

In future, the parallel implementation of the scheme will be considered to solve multi-dimensional nonlinear evolution equations subject to different boundary conditions.

Acknowledgments

The authors are grateful to anonymous referees for their valuable comments and suggestions which improved the quality of this manuscript. A. Chowdhury acknowledges support from the US National Science Foundation under grant number 1800798.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

A. Chowdhury acknowledges support from the US National Science Foundation [grant number 1800798].

References

- [1] G.D. Akrivis, *Finite difference discretization of the Kuramoto–Sivashinsky equation*, Numer. Math. 63 (1992), pp. 1–11.
- [2] G.D. Akrivis, *Finite element discretization of the Kuramoto–Sivashinsky equation*, Numer. Anal. Math. Model. 29(1) (1994), pp. 155–163.
- [3] G. Akrivis and Y.-S. Smyrlis, *Implicit-explicit BDF methods for the Kuramoto–Sivashinsky equation*, Appl. Numer. Math. 51 (2004), pp. 159–161.
- [4] S.V. Alekseenko, V.E. Nakoryakov, and B.G. Pokusaev, *Wave formation on vertical falling liquid films*, Int. J. Multiphase Flow. 11 (1985), pp. 607–627.
- [5] E.O. Asante-Asanami, A. Kleefeld, and B.A. Wade, *A second-order exponential time differencing scheme for non-linear reaction–diffusion systems with dimensional splitting*, J. Comput. Phys. 415 (2020), p. 109490.
- [6] D.J. Benny, *Long waves in liquid film*, J. Math. Phys. 45 (1966), pp. 150–155.
- [7] G. Beylkin, J.M. Keiser, and L. Vozovoi, *A new class of time discretization schemes for the solution of nonlinear PDEs*, J. Comput. Phys. 147 (1998), pp. 362–387.
- [8] H.P. Bhatt and A.Q.M. Khaliq, *Higher order exponential time differencing scheme for system of coupled nonlinear Schrödinger equations*, Appl. Math. Comput. 228 (2014), pp. 271–291.
- [9] H.P. Bhatt and A.Q.M. Khaliq, *The locally extrapolated exponential time differencing LOD scheme for multidimensional reaction–diffusion systems*, J. Comput. Appl. Math. 285 (2015), pp. 256–278.
- [10] M. Caliarì and A. Ostermann, *Implementation of exponential Rosenbrock type integrals*, App. Numer. Math. 59 (2009), pp. 568–581.
- [11] C.I. Christov and K.L. Bekyarov, *A Fourier-series method for solving solution problems*, SIAM J. Sci. Stat. Comput. 11(4) (1990), pp. 631–647.
- [12] C.I. Christov and M.G. Velarde, *Dissipative solitons*, Phys. D. 86 (1995), pp. 323–347.
- [13] P. Clavin, *Dynamic behavior of premixed flame fronts in laminar and turbulent flows*, Prog. Energy Combust. Sci. 11(1) (1985), pp. 1–59.
- [14] S.M. Cox and P.C. Matthews, *Exponential time differencing for stiff systems*, J. Comput. Phys. 176 (2002), pp. 430–455.
- [15] B. Fornberg and T.A. Driscoll, *A fast spectral for nonlinear wave equations with linear dispersion*, J. Comput. Phys. 155(2) (1999), pp. 456–467.
- [16] U. Frisch, Z.S. She, and O. Thual, *Viscoelastic behaviour of cellular solutions to the Kuramoto–Sivashinsky model*, J. Fluid. Mech. 168 (1986), pp. 221–240.
- [17] M. Hochburck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal. 34 (1997), pp. 1911–1925.
- [18] G.M. Homsy, *Model equations for wavy viscous film flow*, Lect. Appl. Math. 15 (1974), pp. 191–194.

- [19] J.M. Hyman and B. Nicolaenko, *The Kuramoto–Sivashinsky equation: A bridge between PDE's and dynamical systems*, Phys. D 18 (1986), pp. 113–126.
- [20] P.L. Kapitza, *Wave flow in thin layer of viscous liquid*, Zh. Exp. Theor. Fiz. 18 (1948), pp. 3–18.
- [21] A.K. Kassam and L.N. Trefethen, *Fourth-order time stepping for stiff PDEs*, SIAM J. Sci. Comput. 26(4) (2005), pp. 1214–1233.
- [22] A.H. Khater and R.S. Temsah, *Numerical solutions of the generalized Kuramoto–Sivashinsky equation by Chebyshev spectral collocation methods*, Comput. Math. Appl. 56 (2008), pp. 1465–1472.
- [23] D.J. Korteweg and G. de Vries, *On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves*, Philos. Mag. 39 (1895), pp. 422–443.
- [24] S. Krogstad, *Generalized integrating factor methods for stiff PDEs*, J. Comput. Phys. 203 (2005), pp. 72–88.
- [25] Y. Kuramoto and T. Tsuzuki, *Persistent propagation of concentration waves in dissipative media far from thermal equilibrium*, Prog. Theor. Phys. 55 (1976), p. 356–369.
- [26] H. Lai and C.F. Ma, *Lattice Boltzmann method for the generalized Kuramoto–Sivashinsky equation*, Phys. A 388 (2009), pp. 1405–1412.
- [27] M. Lakestania and M. Dehghan, *Numerical solutions of the generalized Kuramoto–Sivashinsky equation using B-spline functions*, Appl. Math. Model. 36 (2012), pp. 605–617.
- [28] S.K. Lele, *Compact finite difference schemes with spectral-like resolution*, J. Comput. Phys. 103(1) (1992), pp. 16–42.
- [29] A.V. Manickam, K.M. Moudgalya, and A.K. Pani, *Second-order splitting combined with orthogonal cubic spline collocation method for the Kuramoto–Sivashinsky equation*, Comput. Math. Appl. 35 (1998), pp. 5–25.
- [30] R.C. Mittal and G. Arora, *Quintic B-spline collocation method for numerical solution of the Kuramoto–Sivashinsky equation*, Comm. Nonlinear Sci. Numer. Simul. 15 (2010), pp. 2798–2808.
- [31] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev. 45 (2003), pp. 3–49.
- [32] A. Mouloud, H. Fellouah, B.A. Wade, and M. Kessal, *Time discretization and stability regions for dissipative-dispersive Kuramoto–Sivashinsky equation arising in turbulent gas flow over laminar liquid*, J. Comput. Appl. Math. 330 (2018), pp. 605–617.
- [33] H. Otomo, B.M. Boghosian, and F. Dubois, *Efficient lattice Boltzmann models for Kuramoto–Sivashinsky equation*, Comput. Fluids. 172 (2018), pp. 683–688.
- [34] A. Pumir, P. Manneville, and Y. Pomeau, *On the solitary waves running down an inclined plane*, J. Fluid Mech. 135 (1983), pp. 27–50.
- [35] B.K. Singh, G. Arora, and P. Kumar, *A note on solving the fourth-order Kuramoto–Sivashinsky equation by the compact finite difference scheme*, Ain Shams Eng. J. 9 (2018), pp. 1581–1589.
- [36] T. Shlang and G.I. Sivashinsky, *Irregular flow of a liquid film down a vertical column*, J. Phys. Paris 43 (1982), pp. 459–466.
- [37] G.I. Sivashinsky, *Nonlinear analysis of hydrodynamics instability in laminar flames – I. derivation of basic equations*, Acta Astronaut. 4 (1977), pp. 1177–1206.
- [38] G.I. Sivashinsky, *Instabilities, pattern-formation, and turbulence in flames*, Ann. Rev. Fluid Mech. 15 (1983), pp. 179–199.
- [39] G.I. Sivashinsky and D.M. Michelson, *On irregular wavy flow of a liquid film down a vertical plane*, Prog. Theor. Phys. 63 (1980), pp. 2112–2114.
- [40] Y.A. Suhov, *A spectral method for the time evolution in parabolic problems*, J. Sci. Comput. 29 (2006), pp. 201–217.
- [41] H. Tal-Ezer, *On restart and error estimation for Krylov approximation of $w = f(A)v$* , SIAM J. Sci. Comput. 29 (2007), pp. 2426–2441.
- [42] M. Uddin, S. Haq, Siraj-ul-Islam, *A mesh-free numerical method for solution of the family of Kuramoto–Sivashinsky equations*, Appl. Math. Comput. 212(2) (2009), pp. 458–469.
- [43] Y. Xu and C.W. Shu, *Local discontinuous Galerkin methods for the Kuramoto–Sivashinsky equations and the Ito-type coupled KdV equations*, Comput. Meth. Appl. Mech. Eng. 195 (2006), pp. 3430–3447.
- [44] M. Zarebnia and R. Parvaz, *Septic B-spline collocation method for numerical solution of the Kuramoto–Sivashinsky equation*, Int. J. Math. Comput. Sci. Eng. 2(1) (2013), pp. 55–61.