Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 581987, 13 pages
http://dx.doi.org/10.1155/2014/581987

Research Article

A Bivariate Chebyshev Spectral Collocation Quasilinearization Method for Nonlinear Evolution Parabolic Equations

S. S. Motsa, V. M. Magagula, and P. Sibanda

School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Private Bag X01, Scottsville, Pietermaritzburg 3209, South Africa

Correspondence should be addressed to S. S. Motsa; sandilemotsa@gmail.com

Received 15 July 2014; Accepted 12 August 2014; Published 27 August 2014

Academic Editor: Hassan Saberi Nik

This paper presents a new method for solving higher order nonlinear evolution partial differential equations (NPDEs). The method combines quasilinearisation, the Chebyshev spectral collocation method, and bivariate Lagrange interpolation. In this paper, we use the method to solve several nonlinear evolution equations, such as the modified KdV-Burgers equation, highly nonlinear modified KdV equation, Fisher's equation, Burgers-Fisher equation, Burgers-Huxley equation, and the Fitzhugh-Nagumo equation. The results are compared with known exact analytical solutions from literature to confirm accuracy, convergence, and effectiveness of the method. There is congruence between the numerical results and the exact solutions to a high order of accuracy. Tables were generated to present the order of accuracy of the method; convergence graphs to verify convergence of the method and error graphs are presented to show the excellent agreement between the results from this study and the known results from literature.

1. Introduction

Nonlinearity exists everywhere and, in general, nature is nonlinear. Nonlinear evolution partial differential equations arise in many fields of science, particularly in physics, engineering, chemistry, finance, and biological systems. They are widely used to describe complex phenomena in various fields of sciences, such as wave propagation phenomena, fluid mechanics, plasma physics, quantum mechanics, nonlinear optics, solid state physics, chemical kinematics, physical chemistry, population dynamics, financial industry, and numerous areas of mathematical modeling. The development of both numerical and analytical methods for solving complicated, highly nonlinear evolution partial differential equations continues to be an area of interest to scientists whose research aim is to enrich deep understanding of such alluring nonlinear problems.

Innumerable number of methods for obtaining analytical and approximate solutions to nonlinear evolution equations have been proposed. Some of the analytical methods that have been used to solve evolution nonlinear partial differential equations include Adomian's decomposition method [1–3], homotopy analysis method [4–7], tanh-function method [8–10], Haar wavelet method [11–13], and Exp-function method [14–16]. Several numerical methods have been used to solve nonlinear evolution partial differential equations. These include the explicit-implicit method [17], Chebyshev finite difference methods [18], finite difference methods [19], finite element methods [20], and pseudospectral methods [21, 22].

Some drawbacks of approximate analytical methods include slow convergence, particularly for large time ($t > 1$). They may also be cumbersome to use as some involve manual integration of approximate series solutions and, hence, it is difficult to find closed solutions sometimes. On the other hand, some numerical methods may not work in some cases, for example, when the required solution has to be found near a singularity. Certain numerical methods, for example, finite differences require many grid points to achieve good accuracy and, hence, require a lot of computer memory and computational time. Conventional first-order finite difference methods may result in monotonic and stable solutions, but they are strongly dissipative causing the solution of the strongly convective partial differential equations to become smeared out and often grossly inaccurate. On the other hand, higher order difference methods are less dissipative but are prone to numerical instabilities.

Spectral methods have been used successfully in many different fields in sciences and engineering because of their ability to give accurate solutions of differential equations. Khater et al. [23] applied the Chebyshev spectral collocation method to solve Burgers type of equations in space and finite differences to approximate the time derivative. The Chebyshev spectral collocation method has been used together with the fourth-order Runge-Kutta method to solve the nonlinear PDEs in this study. The Chebyshev spectral collocation is first applied to the NPDE and this yields a system of ordinary differential equations, which are solved using the fourth-order Runge-Kutta method. Olmos and Shizgal [24], Javidi [25, 26], Dehghan and Fakhar-Izadi [27], Driscoll [28], and Driscoll [28] solved the Fisher, Burgers-Fisher, Burgers-Huxley, Fitzhugh-Nagumo, and KdV equations, respectively, using a combination of the Chebyshev spectral collocation method and fourth-order Runge-Kutta method. Darvishi et al. [29, 30] solved the KdV and the Burgers-Huxley equations using a combination of the Chebyshev spectral collocation method and Darvishi's preconditioning. Jacobs and Harley [31] and Tohidi and Kilicman [32] used spectral collocation directly for solving linear partial differential equations. Accuracy will be compromised if they implement their approach in solving nonlinear partial differential equations since they use Kronecker multiplication.

Chebyshev spectral methods are defined everywhere in the computational domain. Therefore, it is easy to get an accurate value of the function under consideration at any point of the domain, beside the collocation points. This property is often exploited, in particular to get a significant graphic representation of the solution, making the possible oscillations due to a wrong approximation of the derivative apparent. Spectral collocation methods are easy to implement and are adaptable to various problems, including variable coefficient and nonlinear differential equations. The error associated with the Chebyshev approximation is $\mathcal{O}(1/N^r)$ where N refers to the truncation and r is connected to the number of continuous derivatives of the function. The interest in using Chebyshev spectral methods in solving nonlinear PDEs stems from the fact that these methods require less grid points to achieve accurate results. They are computational and efficient compared to traditional methods like finite difference and finite element methods. Chebyshev spectral collocation method has been used in conjunction with additional methods which may have their own drawbacks. Here, we provide an alternative method that is not dependent on another method to approximate the solution.

The main objective of this work is to introduce a new method that uses Chebyshev spectral collocation, bivariate Lagrange interpolation polynomials together with quasilinearisation techniques. The nonlinear evolution equations are first linearized using the quasilinearisation method. The Chebyshev spectral collocation method with Lagrange interpolation polynomials are applied independently in space and time variables of the linearized evolution partial differential equation. This new method is termed bivariate interpolated spectral quasilinearisation method (BI-SQLM). We present the BI-SQLM algorithm in a general setting, where it can be used to solve any rth order nonlinear evolution equations. The applicability, accuracy, and reliability of the proposed BI-SQLM are confirmed by solving the modified KdV-Burger equation, highly nonlinear modified KdV equation, the Cahn-Hillard equation,the fourth-order KdV equation, Fisher's, Burgers-Fisher, Burger-Huxley, and the Fitzhugh-Nagumo equations. The results of the BI-SQLM are compared against known exact solutions that have been reported in the scientific literature. It is observed that the method achieves high accuracy with relatively fewer spatial grid points. It also converges fast to the exact solution and approximates the solution of the problem in a computationally efficient manner with simulations completed in fractions of a second in all cases. Tables are generated to show the order of accuracy of the method and time taken to compute the solutions. It is observed that, as the number of grid points is increased, the error decreases. Error graphs and graphs showing the excellent agreement of the exact and analytical solutions for all the nonlinear evolution equations are also presented.

The paper is organized as follows. In Section 2, we introduce the BI-SQLM algorithm for a general nonlinear evolution PDE. In Section 3, we describe the application of the BI-SQLM to selected test problems. The numerical simulations and results are presented in Section 4. Finally, we conclude in Section 5.

2. Bivariate Interpolated Spectral Quasilinearization Method (BI-SQLM)

In this section, we introduce the *Bivariate Interpolated Spectral Quasilinearization Method* (BI-SQLM) for finding solutions to nonlinear evolution PDEs. Without loss of generality, we consider nonlinear PDEs of the form

$$\frac{\partial u}{\partial \tau} = H\left(u, \frac{\partial u}{\partial \eta}, \frac{\partial^2 u}{\partial \eta^2}, \dots, \frac{\partial^n u}{\partial \eta^n}\right), \qquad (1)$$

$$\text{with the physical region } \tau \in [0, T], \quad \eta \in [a, b],$$

where n is the order of differentiation, $u(\eta, \tau)$ is the required solution, and H is a nonlinear operator which contains all the spatial derivatives of u. The given physical region, $\tau \in [0, T]$, is converted to the region $t \in [-1, 1]$ using the linear transformation $\tau = T(t + 1)/2$ and $\eta \in [a, b]$ is converted to the region $x \in [-1, 1]$ using the linear transformation

$$\eta = \frac{1}{2}(b - a)x + \frac{1}{2}(b + a). \qquad (2)$$

Equation (1) can be expressed as

$$\frac{\partial u}{\partial t} = H\left(u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \dots, \frac{\partial^n u}{\partial x^n}\right), \quad t \in [-1, 1], \ x \in [-1, 1]. \qquad (3)$$

The solution procedure assumes that the solution can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$u(x,t) \approx \sum_{i=0}^{N_x}\sum_{j=0}^{N_t} u(x_i,t_j)\, L_i(x)\, L_j(t), \tag{4}$$

which interpolates $u(x,t)$ at selected points in both the x and t directions defined by

$$\{x_i\} = \left\{\cos\left(\frac{\pi i}{N_x}\right)\right\}_{i=0}^{N_x}, \qquad \{t_j\} = \left\{\cos\left(\frac{\pi j}{N_t}\right)\right\}_{j=0}^{N_t}. \tag{5}$$

The choice of the Chebyshev-Gauss-Lobatto grid points (5) ensures that there is a simple conversion of the continuous derivatives, in both space and time, to discrete derivatives at the grid points. The functions $L_i(x)$ are the characteristic Lagrange cardinal polynomials

$$L_i(x) = \prod_{\substack{i=0\\ i\neq k}}^{N_x} \frac{x-x_k}{x_i-x_k}, \tag{6}$$

where

$$L_i(x_k) = \delta_{ik} = \begin{cases} 0 & \text{if } i \neq k \\ 1 & \text{if } i = k. \end{cases} \tag{7}$$

The function $L_j(t)$ is defined in a similar manner. Before linearizing (3), it is convenient to split H into its linear and nonlinear components and rewrite the governing equation in the form

$$F\left[u, u', \dots, u^{(n)}\right] + G\left[u, u', \dots, u^{(n)}\right] - \dot{u} = 0, \tag{8}$$

where the dot and primes denote the time and space derivatives, respectively, F is a linear operator, and G is a nonlinear operator. Assuming that the difference $u_{r+1} - u_r$ and all it's space derivative is small, we first approximate the nonlinear operator G using the linear terms of the Taylor series and, hence,

$$\begin{aligned} G\left[u, u', \dots, u^{(n)}\right] &\approx G\left[u_r, u_r', \dots, u_r^{(n)}\right] \\ &\quad + \sum_{k=0}^{n} \frac{\partial G}{\partial u^{(k)}}\left(u_{r+1}^{(k)} - u_r^{(k)}\right), \end{aligned} \tag{9}$$

where r and $r+1$ denote previous and current iterations, respectively. We remark that this quasilinearization method (QLM) approach is a generalisation of the Newton-Raphson method and was first proposed by Bellman and Kalaba [33] for solving nonlinear boundary value problems.

Equation (9) can be expressed as

$$\begin{aligned} G\left[u, u', \dots, u^{(n)}\right] &\approx G\left[u_r, u_r', \dots, u_r^{(n)}\right] \\ &\quad + \sum_{k=0}^{n} \phi_{k,r}\left[u_r, u_r', \dots, u_r^{(n)}\right] u_{r+1}^{(k)} \\ &\quad - \sum_{k=0}^{n} \phi_{k,r}\left[u_r, u_r', \dots, u_r^{(n)}\right] u_r^{(k)}, \end{aligned} \tag{10}$$

where

$$\phi_{k,r}\left[u_r, u_r', \dots, u_r^{(n)}\right] = \frac{\partial G}{\partial u^{(k)}}\left[u_r, u_r', \dots, u_r^{(n)}\right]. \tag{11}$$

Substituting (10) into (8), we get

$$\begin{aligned} F\left[u_{r+1}, u_{r+1}', \dots, u_{r+1}^{(n)}\right] + \sum_{k=0}^{n} \phi_{k,r} u_{r+1}^{(k)} - \dot{u}_{r+1} \\ = R_r\left[u_r, u_r', \dots, u_r^{(n)}\right], \end{aligned} \tag{12}$$

where

$$R_r\left[u_r, u_r', \dots, u_r^{(n)}\right] = \sum_{k=0}^{n} \phi_{k,r} u_r^{(k)} - G\left[u_r, u_r', \dots, u_r^{(n)}\right]. \tag{13}$$

A crucial step in the implementation of the solution procedure is the evaluation of the time derivative at the grid points t_j $(j = 0, 1, \dots, N_t)$ and the spatial derivatives at the grid points x_i $(i = 0, 1, \dots, N_x)$. The values of the time derivatives at the Chebyshev-Gauss-Lobatto points (x_i, t_j) are computed as (for $j = 0, 1, 2, \dots, N_t$)

$$\begin{aligned} \left.\frac{\partial u}{\partial t}\right|_{x=x_i, t=t_j} &= \sum_{p=0}^{N_x}\sum_{k=0}^{N_t} u(x_p, t_k)\, L_p(x_i)\, \frac{dL_k(t_j)}{dt} \\ &= \sum_{k=0}^{N_t} u(x_i, t_k)\, d_{jk} = \sum_{k=0}^{N_t} d_{jk} u(x_i, t_k), \end{aligned} \tag{14}$$

where $d_{jk} = dL_k(t_j)/dt$ is the standard first derivative Chebyshev differentiation matrix of size $(N_t + 1) \times (N_t + 1)$ as defined in [34]. The values of the space derivatives at the Chebyshev-Gauss-Lobatto points (x_i, t_j) $(i = 0, 1, 2, \dots, N_x)$ are computed as

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_{x=x_i, t=t_j} &= \sum_{p=0}^{N_x}\sum_{k=0}^{N_t} u(x_p, t_k)\, \frac{dL_p(x_i)}{dx} L_k(t_j) \\ &= \sum_{p=0}^{N_x} u(x_p, t_j)\, D_{ip} = \sum_{p=0}^{N_x} D_{ip} u(x_p, t_j), \end{aligned} \tag{15}$$

where $D_{ip} = dL_p(x_i)/dx$ is the standard first derivative Chebyshev differentiation matrix of size $(N_x + 1) \times (N_x + 1)$. Similarly, for an nth order derivative, we have

$$\begin{aligned} \left.\frac{\partial^n u}{\partial x^n}\right|_{x=x_i, t=t_j} &= \sum_{p=0}^{N_x} D_{ip}^n u(x_p, t_j) = \mathbf{D}^n \mathbf{U}_j, \\ & i = 0, 1, 2, \dots, N_x, \end{aligned} \tag{16}$$

where the vector $\mathbf{U}_j$ is defined as

$$\mathbf{U}_j = \left[u_j(x_0), u_j(x_1), \dots, u_j(x_{N_x})\right]^T \tag{17}$$

and the superscript T denotes matrix transpose. Substituting (16) into (12) we get

$$\begin{aligned} &F\left[\mathbf{U}_{r+1,j},\mathbf{U}'_{r+1,j},\ldots,\mathbf{U}^{(n)}_{r+1,j}\right]+\sum_{k=0}^{n}\mathbf{\Phi}_{k,r}\mathbf{U}^{(k)}_{r+1,j} \\ &\quad -\sum_{k=0}^{N_t} d_{jk}\mathbf{U}_{r+1,k} = R_r\left[\mathbf{U}_{r,j},\mathbf{U}'_{r,j},\ldots,\mathbf{U}^{(n)}_{r,j}\right] \end{aligned} \tag{18}$$

for $j = 0, 1, 2, \ldots, N_t$, where

$$\begin{gathered} \mathbf{U}^{(n)}_{r+1,j} = \mathbf{D}^{n}\mathbf{U}_{r+1,j}, \\ \mathbf{\Phi}_{k,r} = \begin{bmatrix} \phi_{k,r}\left(x_0,t_j\right) & & & \\ & \phi_{k,r}\left(x_1,t_j\right) & & \\ & & \ddots & \\ & & & \phi_{k,r}\left(x_{N_x},t_j\right) \end{bmatrix}. \end{gathered} \tag{19}$$

The initial condition for (3) corresponds to $\tau_{N_t} = -1$ and, hence, we express (18) as

$$\begin{aligned} &F\left[\mathbf{U}_{r+1,j},\mathbf{U}'_{r+1,j},\ldots,\mathbf{U}^{(n)}_{r+1,j}\right] \\ &\quad +\sum_{k=0}^{n}\mathbf{\Phi}_{k,r}\mathbf{U}^{(k)}_{r+1,j}-\sum_{k=0}^{N_t-1} d_{jk}\mathbf{U}_{r+1,k} = \mathbf{R}_j, \end{aligned} \tag{20}$$

where

$$\begin{gathered} \mathbf{R}_j = R_r\left[\mathbf{U}_{r,j},\mathbf{U}'_{r,j},\ldots,\mathbf{U}^{(n)}_{r,j}\right] + d_{jN_t}\mathbf{U}_{N_t}, \\ j = 0, 1, 2, \ldots, N_t - 1. \end{gathered} \tag{21}$$

Equation (20) can be expressed as the following $N_t(N_x+1)\times N_t(N_x+1)$ matrix system

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{N_t-1} \end{bmatrix}, \tag{22}$$

where

$$\begin{gathered} A_{i,i} = F\left[\mathbf{I},\mathbf{D},\ldots,\mathbf{D}^{(n)}\right] + \sum_{k=0}^{n}\mathbf{\Phi}_{k,r}\mathbf{D}^{(k)} - d_{i,i}\mathbf{I}, \\ A_{i,j} = -d_{i,j}\mathbf{I}, \quad \text{when } i \neq j, \end{gathered} \tag{23}$$

and $\mathbf{I}$ is the identity matrix of size $(N_x+1)\times(N_x+1)$. Solving (19) gives $u(x_i,t_j)$ and, hence, we use (4) to approximate $u(x,t)$.

3. Numerical Experiments

We apply the proposed algorithm to well-known nonlinear PDEs of the form (3) with exact solutions. In order to determine the level of accuracy of the BI-SQLM approximate solution, at a particular time level, in comparison with the exact solution, we report maximum error which is defined by

$$E_N = \max_{r}\left\{\left|u\left(x_r,t\right) - \tilde{u}\left(x_r,t\right)\right|,: 0 \le r \le N\right\}, \tag{24}$$

where $\tilde{u}(x_r,t)$ is the approximate solution and is the $u(x_r,t)$ exact solution at the time level t.

Example 1. We consider the generalized Burgers-Fisher equation [35]:

$$\frac{\partial u}{\partial t} + \alpha u^{\delta}\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u\left(1 - u^{\delta}\right), \tag{25}$$

with initial condition

$$u\left(x,0\right) = \left\{\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}x\right)\right\}^{1/\delta} \tag{26}$$

and exact solution

$$\begin{aligned} &u\left(x,t\right) \\ &\quad = \left\{\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2\left(\delta+1\right)}\right.\right. \\ &\qquad\qquad \left.\left.\times\left[x - \left(\frac{\alpha}{\delta+1} + \frac{\beta\left(\delta+1\right)}{\alpha}\right)t\right]\right)\right\}^{1/\delta}, \end{aligned} \tag{27}$$

where α, β, and δ are parameters. For illustration purposes, these parameters are chosen to be $\alpha = \beta = \delta = 1$ in this paper. The linear operator F and nonlinear operator G are chosen as

$$F\left(u\right) = u'' + u, \qquad G\left(u\right) = -uu' - u^2. \tag{28}$$

We first linearize the nonlinear operator G. We approximate G using the equation

$$G \approx G\left[u_r, u'_r, u''_r\right] + \sum_{k=0}^{2}\phi_{k,r}u^{(k)}_{r+1} - \sum_{k=0}^{2}\phi_{k,r}u^{(k)}_{r}. \tag{29}$$

The coefficients are given by

$$\begin{aligned} \phi_{0,r} &= \frac{\partial G}{\partial u}\left[u_r, u'_r, u''_r\right] = -\left(u'_r + 2u_r\right), \\ \phi_{1,r} &= \frac{\partial G}{\partial u'}\left[u_r, u'_r, u''_r\right] = -u_r, \\ \phi_{2,r} &= \frac{\partial G}{\partial u''}\left[u_r, u'_r, u''_r\right] = 0, \\ R_r &= \sum_{k=0}^{2}\phi_{k,r}u^{(k)}_r - G\left[u_r, u'_r, u''_r\right] = -u_r^2 - u_r u'_r. \end{aligned} \tag{30}$$

Therefore, the linearized equation can be expressed as

$$u''_{r+1} + \phi_{1,r}u'_{r+1} + \phi_{0,r}u_{r+1} + u_{r+1} - \dot{u} = R_r. \tag{31}$$

Applying the spectral method both in x and t and initial condition, we get

$$\begin{aligned} &\mathbf{D}^2\mathbf{U}_{r+1,i} + \mathbf{\Phi}_{1,r}\mathbf{D}\mathbf{U}_{r+1,i} + \mathbf{\Phi}_{0,r}\mathbf{U}_{r+1,i} \\ &\quad + \mathbf{U}_{r+1,i} - 2\sum_{j=0}^{N_t-1} d_{ij}\mathbf{U}_{r+1,j} = \mathbf{R}_i. \end{aligned} \tag{32}$$

Equation (32) can be expressed as

$$\begin{aligned} &\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{N_t-1} \end{bmatrix}, \end{aligned} \tag{33}$$

where

$$\begin{aligned} A_{i,i} &= \mathbf{D}^2 + \mathbf{\Phi}_{1,r}^{(i)}\mathbf{D} + \mathbf{\Phi}_{0,r}^{(i)} + \left(1 - 2d_{i,i}\right)\mathbf{I}, \\ A_{i,j} &= -2d_{i,j}\mathbf{I}, \quad \text{when } i \neq j, \\ \mathbf{R}_i &= R_r + 2d_{iN_t}\mathbf{U}_{r,N_t}. \end{aligned} \tag{34}$$

The boundary conditions are implemented in the first and last row of the matrices A_{ij} and the column vectors $\mathbf{R}_i$ for $i = 0, 1, \ldots, N_t - 1$ and $j = 0, 1, \ldots, N_t - 1$. The procedure for finding the variable coefficients ϕ_i and matrices for the remaining examples is similar.

Example 2. We consider Fisher's equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \alpha u\left(1 - u\right), \tag{35}$$

subject to the initial condition

$$u\left(x, 0\right) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6}x}\right)^2} \tag{36}$$

and exact solution [36]

$$u\left(x, t\right) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6}x - 5\alpha t/6}\right)^2}, \tag{37}$$

where α is a constant. The Fisher equation represents a reactive-diffusive system and is encountered in chemical kinetics and population dynamics applications. For this example, the appropriate linear operator F and nonlinear operator G are chosen as

$$F\left(u\right) = u'' + \alpha u, \qquad G\left(u\right) = -\alpha u^2. \tag{38}$$

Table 1: Maximum errors E_N for Fisher equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	$1.986e-008$	$1.119e-011$	$7.398e-013$	$7.171e-013$
0.2	$3.934e-008$	$3.121e-011$	$1.552e-012$	$1.561e-012$
0.3	$5.577e-008$	$4.864e-011$	$1.004e-012$	$1.005e-012$
0.4	$6.997e-008$	$6.802e-011$	$7.895e-013$	$8.124e-013$
0.5	$8.107e-008$	$7.971e-011$	$1.088e-012$	$1.027e-012$
0.6	$8.891e-008$	$8.560e-011$	$8.805e-013$	$7.847e-013$
0.7	$9.344e-008$	$8.953e-011$	$6.418e-013$	$6.463e-013$
0.8	$9.431e-008$	$8.759e-011$	$6.199e-013$	$6.164e-013$
0.9	$9.178e-008$	$8.325e-011$	$3.978e-013$	$3.695e-013$
1.0	$8.787e-008$	$7.421e-011$	$7.988e-014$	$5.596e-014$
CPU time (sec)	0.019942	0.025988	0.027756	0.029436

Table 2: Maximum errors E_N for the Burgers-Fisher equation when $\alpha = \gamma = \delta = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	$1.142e-007$	$1.369e-010$	$5.891e-012$	$6.143e-012$
0.2	$1.178e-007$	$1.373e-010$	$9.570e-012$	$1.013e-011$
0.3	$1.186e-007$	$1.479e-010$	$1.489e-011$	$1.512e-011$
0.4	$1.069e-007$	$9.450e-011$	$1.703e-011$	$1.702e-011$
0.5	$9.030e-008$	$7.944e-011$	$5.283e-012$	$5.736e-012$
0.6	$6.963e-008$	$6.618e-011$	$1.639e-011$	$1.626e-011$
0.7	$4.638e-008$	$1.579e-011$	$1.362e-011$	$1.364e-011$
0.8	$2.457e-008$	$4.030e-011$	$3.934e-012$	$3.852e-012$
0.9	$2.028e-008$	$6.006e-011$	$4.466e-012$	$4.727e-012$
1.0	$3.147e-008$	$7.708e-011$	$7.757e-013$	$7.261e-013$
CPU Time (sec)	0.010152	0.015387	0.019163	0.021564

Example 3. Consider the Fitzhugh-Nagumo equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u\left(u - \alpha\right)\left(1 - u\right) \tag{39}$$

with initial condition

$$u\left(x, 0\right) = \frac{1}{2}\left[1 - \coth\left(-\frac{x}{2\sqrt{2}}\right)\right]. \tag{40}$$

This equation has the exact solution [37]

$$u\left(x, t\right) = \frac{1}{2}\left[1 - \coth\left(-\frac{x}{2\sqrt{2}} + \frac{2\alpha - 1}{4}t\right)\right], \tag{41}$$

where α is a parameter. In this example, the linear operator F and nonlinear operator G are chosen as

$$F\left(u\right) = u'' - \alpha u, \qquad G\left(u\right) = \left(1 + \alpha\right)u^2 - u^3. \tag{42}$$

Table 3: Maximum errors E_N for the Fitzhug-Nagumo equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	5.719e − 007	1.196e − 009	2.367e − 012	9.881e − 014
0.2	6.193e − 007	1.299e − 009	2.761e − 012	3.952e − 014
0.3	6.662e − 007	1.463e − 009	3.259e − 012	8.216e − 014
0.4	6.779e − 007	1.448e − 009	3.341e − 012	8.094e − 014
0.5	6.920e − 007	1.526e − 009	3.587e − 012	5.063e − 014
0.6	7.019e − 007	1.573e − 009	3.729e − 012	3.775e − 014
0.7	6.933e − 007	1.516e − 009	3.660e − 012	8.915e − 014
0.8	6.828e − 007	81.535e − 009	3.635e − 012	7.594e − 014
0.9	6.765e − 007	1.528e − 009	3.519e − 012	3.242e − 013
1.0	6.687e − 007	1.490e − 009	3.405e − 012	1.688e − 013
CPU time (sec)	0.024281	0.024901	0.026810	0.032389

Table 4: Maximum errors E_N for the Burger-Huxley equation when $\gamma = 0.75$, $\beta = 1$, and $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	2.217e − 006	8.482e − 009	2.166e − 011	7.822e − 014
0.2	2.596e − 006	9.369e − 009	2.536e − 011	1.184e − 013
0.3	2.859e − 006	1.073e − 008	3.201e − 011	1.049e − 013
0.4	3.001e − 006	1.112e − 008	3.652e − 011	9.426e − 014
0.5	3.137e − 006	1.213e − 008	4.262e − 011	1.510e − 013
0.6	3.270e − 006	1.311e − 008	4.842e − 011	2.127e − 013
0.7	3.367e − 006	1.359e − 008	5.289e − 011	1.230e − 013
0.8	3.467e − 006	1.438e − 008	5.803e − 011	1.549e − 013
0.9	3.562e − 006	1.504e − 008	6.260e − 011	3.063e − 013
1.0	3.640e − 006	1.559e − 008	6.674e − 011	2.951e − 013
CPU time (sec)	0.023822	0.024901	0.02685	0.032806

Table 5: Maximum errors E_N for the modified KdV-Burgers equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	1.803e − 007	3.419e − 010	4.449e − 013	1.572e − 013
0.2	2.614e − 007	4.347e − 010	5.049e − 013	5.992e − 014
0.3	2.717e − 007	4.677e − 010	5.532e − 013	8.128e − 013
0.4	2.009e − 007	3.663e − 010	4.771e − 013	6.158e − 013
0.5	2.580e − 007	4.410e − 010	7.518e − 013	2.555e − 013
0.6	2.653e − 007	4.606e − 010	8.738e − 013	5.756e − 013
0.7	2.248e − 007	4.039e − 010	6.210e − 013	2.393e − 013
0.8	2.572e − 007	4.476e − 010	5.432e − 013	6.812e − 013
0.9	2.436e − 007	4.351e − 010	6.111e − 013	6.287e − 013
1.0	8.275e − 008	3.721e − 010	7.569e − 013	1.087e − 007
CPU time (sec)	0.015646	0.021226	0.030159	0.035675

Table 6: Maximum errors E_N for the highly nonlinear modified KdV equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	7.788e − 005	3.553e − 007	7.601e − 010	2.080e − 010
0.2	1.153e − 004	4.000e − 007	5.684e − 010	1.189e − 010
0.3	1.011e − 004	3.739e − 007	4.471e − 010	4.503e − 010
0.4	3.926e − 005	1.785e − 007	6.544e − 010	4.987e − 010
0.5	6.727e − 005	2.342e − 007	2.638e − 010	1.528e − 010
0.6	6.065e − 005	2.207e − 007	4.565e − 010	4.568e − 010
0.7	2.511e − 005	1.105e − 007	4.749e − 010	3.748e − 010
0.8	4.074e − 005	1.427e − 007	1.062e − 010	1.604e − 010
0.9	2.386e − 005	1.018e − 007	2.343e − 010	8.114e − 011
1.0	1.440e − 006	7.256e − 008	1.436e − 009	1.513e − 011
CPU time (sec)	0.020609	0.021241	0.030617	0.032816

Example 4. Consider the Burgers-Huxley equation

$$\frac{\partial u}{\partial t} + \alpha u^{\delta} u_x = \frac{\partial^2 u}{\partial x^2} + \beta u \left(1 - u^{\delta}\right)\left(u^{\delta} - \gamma\right), \qquad (43)$$

where $\alpha, \beta \geq 0$ are constant parameters, δ is a positive integer (set to be $\delta = 1$ in this study), and $\gamma \in (0, 1)$. The exact solution subject to the initial condition

$$u(x, 0) = \frac{1}{2} - \frac{1}{2} \tanh\left[\frac{\beta}{r - \alpha} x\right], \qquad (44)$$

is reported in [38, 39] as

$$u(x, t) = \frac{1}{2} - \frac{1}{2} \tanh\left[\frac{\beta}{r - \alpha}(x - ct)\right], \qquad (45)$$

where

$$r = \sqrt{\alpha^2 + 8\beta}, \qquad c = \frac{(\alpha - r)(2\gamma - 1) + 2\alpha}{4} \qquad (46)$$

The general solution (45) was reported in [40, 41]. In this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' - \beta\gamma u,$$
$$G(u) = -\alpha u u' + \beta(1 + \gamma) u^2 - \beta u^3. \qquad (47)$$

Example 5. We consider the modified KdV-Burgers equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} - \frac{\partial^2 u}{\partial x^2} - 6u^2 \frac{\partial u}{\partial x} \qquad (48)$$

subject to the initial condition

$$u(x, 0) = \frac{1}{6} + \frac{1}{6} \tanh\left(\frac{x}{6}\right) \qquad (49)$$

and exact solution [42]

$$u(x, t) = \frac{1}{6} + \frac{1}{6} \tanh\left(\frac{x}{6} - \frac{t}{27}\right). \qquad (50)$$

Table 7: Maximum errors E_N for Fisher equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	1.119e − 011	7.398e − 013	8.266e − 013	3.808e − 014
0.4	3.121e − 011	1.552e − 012	7.378e − 013	3.780e − 014
0.6	4.864e − 011	1.004e − 012	3.402e − 012	7.283e − 014
0.8	6.802e − 011	7.895e − 013	1.118e − 012	3.714e − 014
1.0	7.971e − 011	1.088e − 012	1.473e − 012	1.691e − 013
1.2	8.560e − 011	8.805e − 013	2.611e − 012	3.119e − 013
1.4	8.953e − 011	6.418e − 013	6.671e − 012	1.796e − 013
1.6	8.759e − 011	6.199e − 013	1.118e − 011	1.097e − 013
1.8	8.325e − 011	3.978e − 013	7.515e − 013	6.273e − 014
2.0	7.421e − 011	7.988e − 014	3.682e − 012	2.311e − 013
CPU time (sec)	0.013542	0.022967	0.023792	0.024758

Table 8: Maximum errors E_N for the Burgers-Fisher equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	1.223e − 007	1.400e − 008	1.402e − 008	1.094e − 012
0.4	1.145e − 007	1.919e − 008	1.918e − 008	3.919e − 012
0.6	9.192e − 008	2.082e − 008	2.085e − 008	1.953e − 012
0.8	2.293e − 008	1.793e − 008	1.793e − 008	6.340e − 013
1.0	2.395e − 008	1.337e − 008	1.339e − 008	2.381e − 012
1.2	5.778e − 008	1.954e − 008	1.930e − 008	1.005e − 011
1.4	6.045e − 008	1.620e − 008	1.620e − 008	3.535e − 012
1.6	5.244e − 008	7.218e − 009	7.345e − 009	5.765e − 012
1.8	4.395e − 008	6.828e − 009	6.784e − 009	3.983e − 012
2.0	2.944e − 008	9.406e − 010	8.820e − 010	3.812e − 012
CPU time (sec)	0.019942	0.025988	0.027756	0.029436

Table 9: Maximum errors E_N for the Fitzhugh-Nagumo equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	6.326e − 007	1.311e − 009	2.886e − 012	1.131e − 012
0.4	6.721e − 007	1.467e − 009	3.310e − 012	1.564e − 012
0.6	7.140e − 007	1.602e − 009	3.617e − 012	1.936e − 012
0.8	6.730e − 007	1.496e − 009	4.707e − 012	1.196e − 012
1.0	6.660e − 007	1.487e − 009	3.675e − 012	1.264e − 012
1.2	6.449e − 007	1.366e − 009	1.897e − 012	1.727e − 012
1.4	5.690e − 007	1.083e − 009	2.972e − 012	1.200e − 012
1.6	4.931e − 007	8.010e − 010	1.519e − 012	8.590e − 013
1.8	3.986e − 007	4.658e − 010	1.068e − 012	6.790e − 013
2.0	2.904e − 007	2.968e − 010	1.592e − 012	1.770e − 013
CPU time (sec)	0.041048	0.049629	0.055008	0.053863

Table 10: Maximum errors E_N for the Burgers-Huxley equation when $\gamma = 0.5$, $\beta = 1$, and $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	2.866e − 006	1.119e − 008	3.670e − 011	1.150e − 012
0.4	3.401e − 006	1.420e − 008	5.744e − 011	1.638e − 012
0.6	3.814e − 006	1.687e − 008	7.426e − 011	1.958e − 012
0.8	3.915e − 006	1.729e − 008	8.171e − 011	7.002e − 013
1.0	3.938e − 006	1.738e − 008	8.157e − 011	1.267e − 012
1.2	3.808e − 006	1.624e − 008	7.687e − 011	1.710e − 012
1.4	3.456e − 006	1.527e − 008	6.965e − 011	5.109e − 013
1.6	3.230e − 006	1.349e − 008	5.535e − 011	8.203e − 013
1.8	2.925e − 006	1.078e − 008	3.598e − 011	8.294e − 013
2.0	2.497e − 006	7.505e − 009	2.265e − 011	9.726e − 014
CPU time (sec)	0.023822	0.024901	0.02685	0.032806

The modified KdV-Burgers equation describes various kinds of phenomena such as a mathematical model of turbulence [43] and the approximate theory of flow through a shock wave traveling in viscous fluid [44]. For this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u''' - u'', \qquad G(u) = -6u'u^2. \tag{51}$$

Example 6. We consider the high nonlinear modified KdV equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} + \left(\frac{\partial u}{\partial x}\right)^2 - u^2 \tag{52}$$

subject to the initial condition

$$u(x,0) = \frac{1}{2} + \frac{e^{-x}}{4} \tag{53}$$

and exact solution

$$u(x,t) = \frac{1}{t+2} + \frac{e^{-(x+t)}}{(t+2)^2}. \tag{54}$$

For this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u''', \qquad G(u) = \left(u'\right)^2 - u^2. \tag{55}$$

4. Results and Discussion

In this section we present the numerical solutions obtained using the BI-SQLM algorithm. The number of collocation points in the space x variable used to generate the results is $N_x = 10$ in all cases. Similarly, the number of collocation points in the time t variable used is $N_t = 10$ in all cases. It was found that sufficient accuracy was achieved using these values in all numerical simulations.

Table 11: Maximum errors E_N for the modified KdV-Burgers equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	$2.137e - 007$	$3.820e - 010$	$4.846e - 013$	$9.998e - 013$
0.4	$2.480e - 007$	$4.267e - 010$	$5.596e - 013$	$8.775e - 013$
0.6	$2.691e - 007$	$4.676e - 010$	$6.565e - 013$	$2.054e - 012$
0.8	$2.214e - 007$	$3.979e - 010$	$8.776e - 013$	$1.168e - 012$
1.0	$2.538e - 007$	$4.463e - 010$	$9.650e - 013$	$8.410e - 013$
1.2	$2.650e - 007$	$4.680e - 010$	$7.450e - 013$	$5.113e - 013$
1.4	$2.383e - 007$	$4.296e - 010$	$7.500e - 013$	$1.110e - 012$
1.6	$2.568e - 007$	$4.572e - 010$	$9.704e - 013$	$2.837e - 013$
1.8	$2.520e - 007$	$4.529e - 010$	$7.443e - 013$	$5.353e - 013$
2.0	$2.370e - 007$	$4.438e - 010$	$2.719e - 013$	$8.849e - 013$
CPU time (sec)	0.062066	0.081646	0.080718	0.10775

Table 12: Maximum errors E_N for the highly nonlinear modified KdV equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	$1.986e - 008$	$1.119e - 011$	$7.398e - 013$	$7.171e - 013$
0.4	$8.010e - 005$	$3.577e - 007$	$3.902e - 008$	$1.979e - 010$
0.6	$7.235e - 005$	$2.549e - 007$	$2.016e - 008$	$4.899e - 010$
0.8	$6.284e - 005$	$1.663e - 007$	$1.155e - 007$	$2.679e - 010$
1.0	$1.642e - 005$	$1.620e - 007$	$1.243e - 007$	$2.474e - 010$
1.2	$2.753e - 005$	$1.073e - 007$	$1.073e - 007$	$1.679e - 010$
1.4	$3.738e - 006$	$8.971e - 008$	$8.598e - 008$	$4.788e - 011$
1.6	$1.223e - 005$	$2.153e - 008$	$2.503e - 008$	$2.941e - 011$
1.8	$5.836e - 006$	$2.986e - 008$	$9.127e - 009$	$5.177e - 011$
2.0	$9.310e - 006$	$6.548e - 008$	$7.277e - 008$	$1.453e - 009$
CPU time (sec)	0.020609	0.021241	0.030617	0.032816

In Tables 1, 2, 3, 4, 5, and 6 we give the maximum errors between the exact and BI-SQLM results for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, at $t \in [0.1, 1]$. The results were computed in the space domain $x \in [0, 1]$. To give a sense of the computational efficiency of the method, the computational time to generate the results is also given. Tables 1–6 clearly show the accuracy of the method. The accuracy is seen to improve with an increase in the number of collocation points N_x. It is remarkable to note that accurate results with errors of order up to 10^{-14} are obtained using very few collocation points in both the x and t variables $N_t \leq 10$, $N_x \leq 10$. This is a clear indication that the BI-SQLM is powerful method that is appropriate in solving nonlinear evolution PDEs. We remark, also, that the BI-SQLM is computationally fast as accurate results are generated in a fraction of a second in all the examples considered in this work.

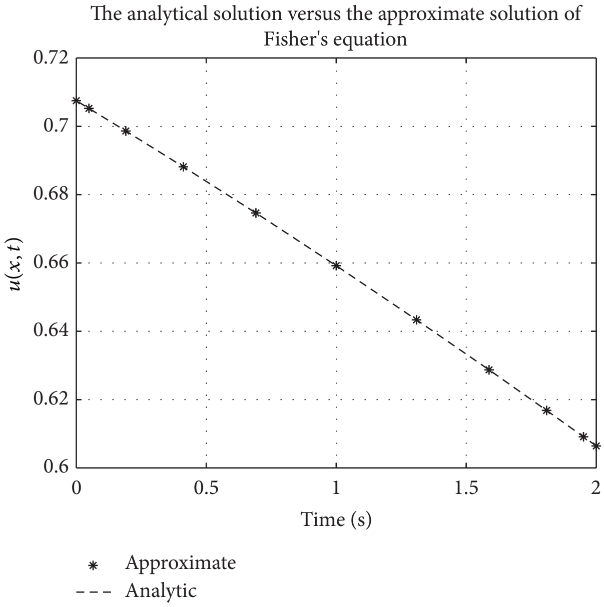


Figure 1: Fishers equation analytical solution graph.

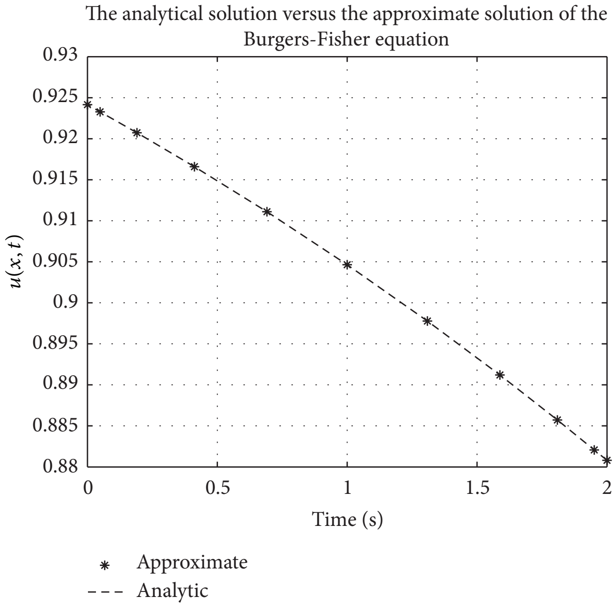


Figure 2: Burger-Fishers equation analytical solution graph.

In Tables 7, 8, 9, 10, 11, and 12 we give the maximum errors of the BI-SQLM results for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, at selected values of $t = 2$ for different collocation points, N_t, in the t-variable. The results in Tables 7–12 were computed on the space domain $x \in [0, 1]$. We note that the accuracy does not detoriate when $t > 1$ for this method as is often the case with numerical schemes such as finite differences.

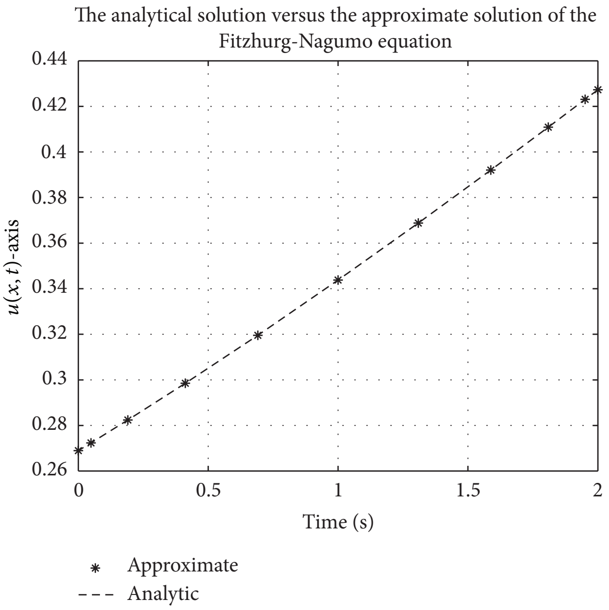


FIGURE 3: Fitzhugh-Nagumo equation analytical solution graph.

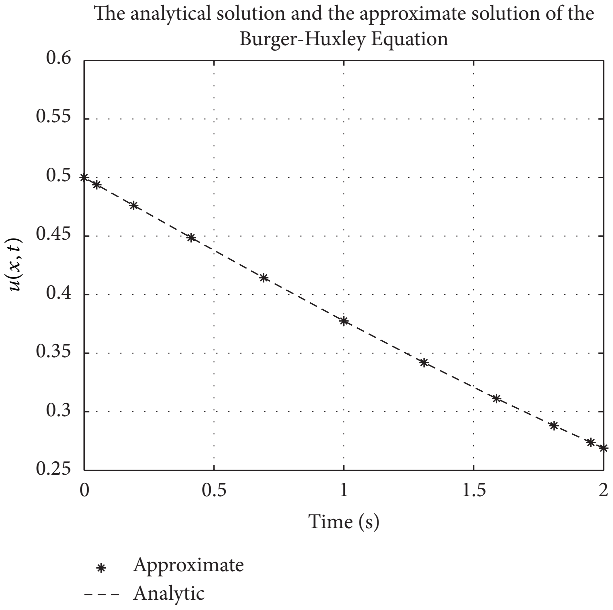


FIGURE 4: Burgers-Huxley equation analytical solution graph.

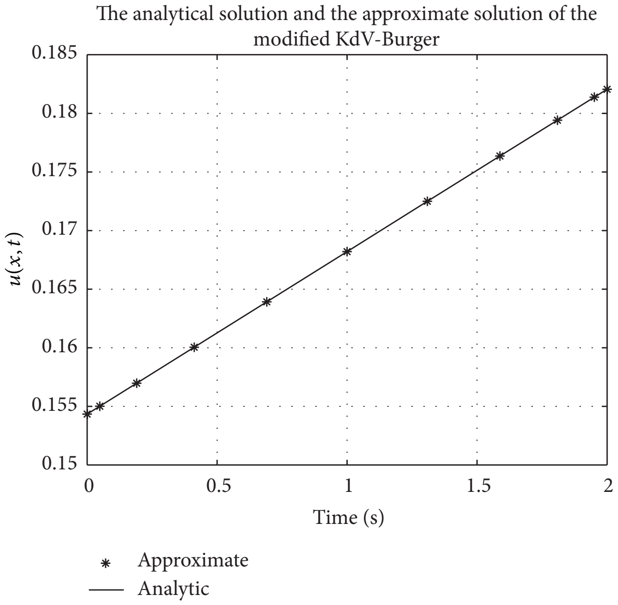


FIGURE 5: Modified KdV-Burger equation analytical solution graph.

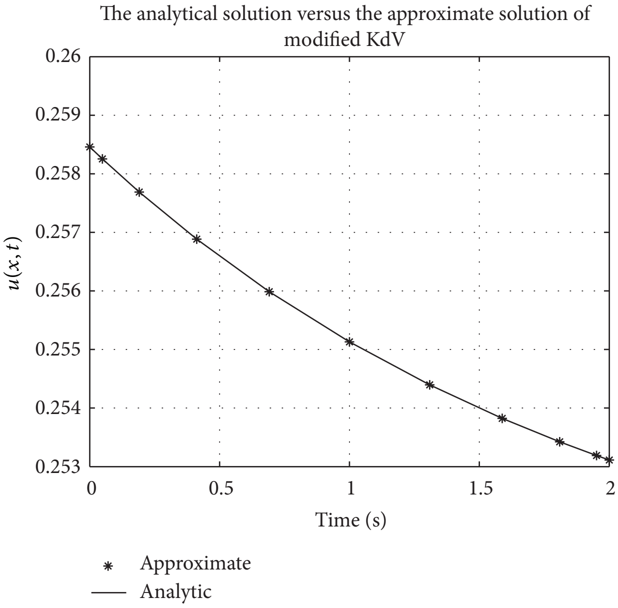


FIGURE 6: Modified KdV equation analytical solution graph.

Figures 1, 2, 3, 4, 5, and 6 show a comparison of the analytical and approximate solutions of the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, when $t = 2$. The approximate solutions are in excellent agreement with the analytical solutions, and this demonstrates the accuracy of the algorithm presented in this study.

In Figures 7, 8, 9, 10, 11, and 12, we present error analysis graphs for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, when $t = 2$.

In Figures 13, 14, 15, 16, 17, and 18, convergence analysis graphs for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively. The figures present a variation of the error norm at a fixed value of time ($t = 1$) with iterations of the BI-SQLM scheme. It can be seen that, in almost all the examples considered, the iteration scheme takes about 3 or 4 iterations to converge fully. Beyond the point where full convergence is reached, error norm levels off and does

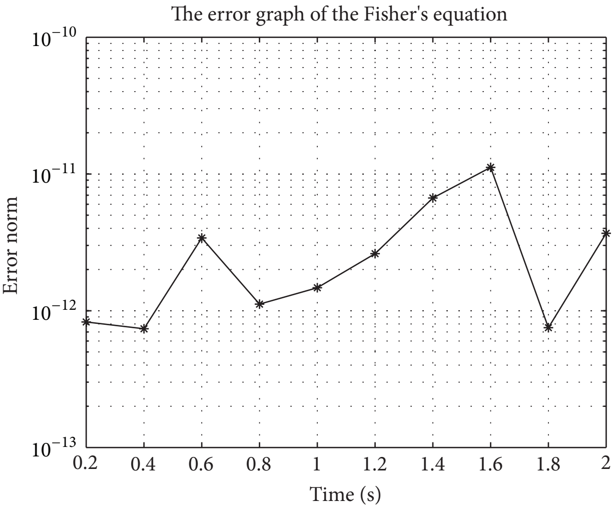


Figure 7: Fishers equation error graph.

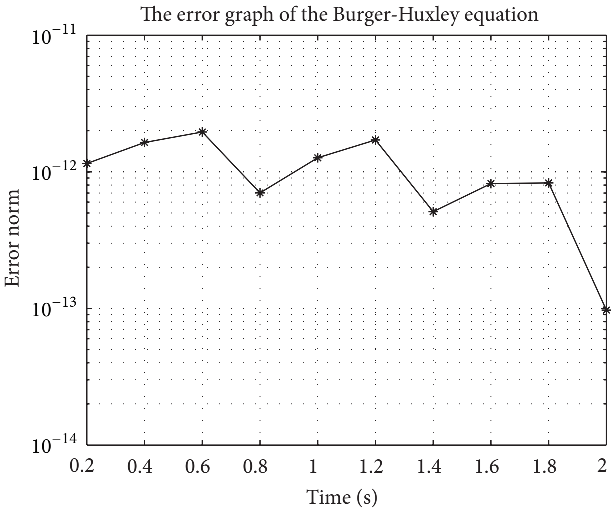


Figure 10: Burgers-Huxley equation error graph.

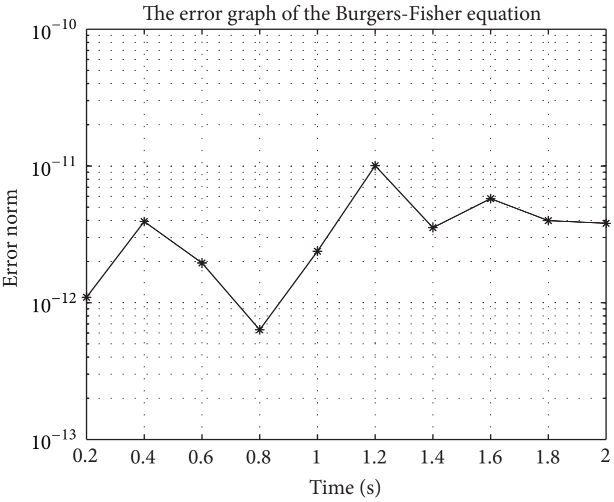


Figure 8: Burger-Fishers equation error graph.

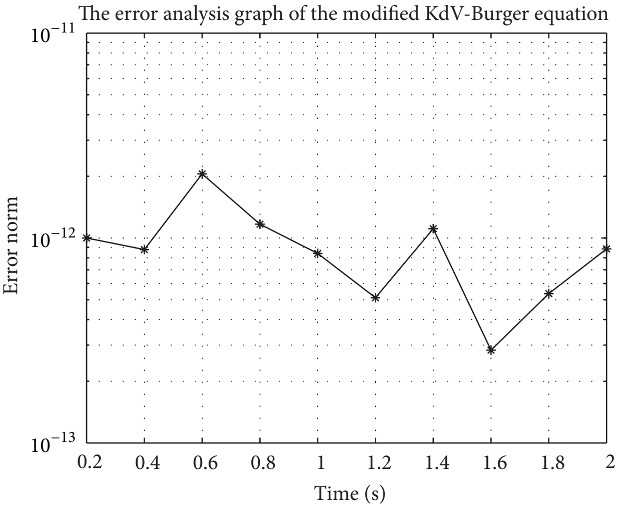


Figure 11: Modified KdV-Burger equation error graph.

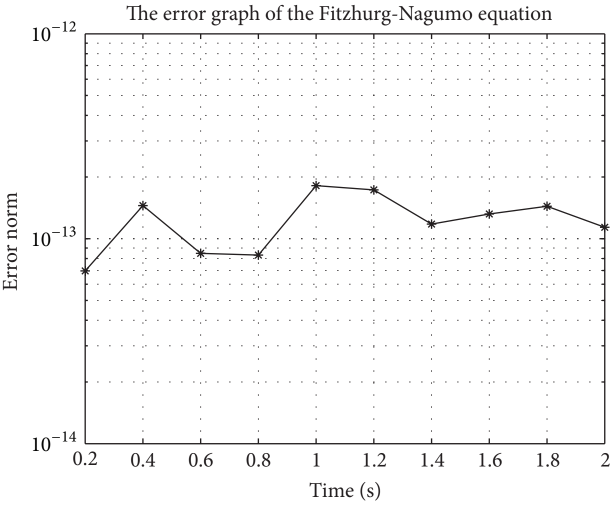


Figure 9: Fitzhugh-Nagumo equation error graph.

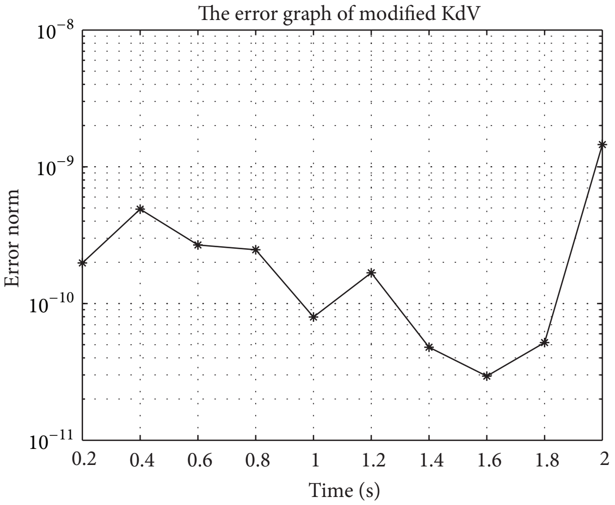


Figure 12: Modified KdV equation error graph.

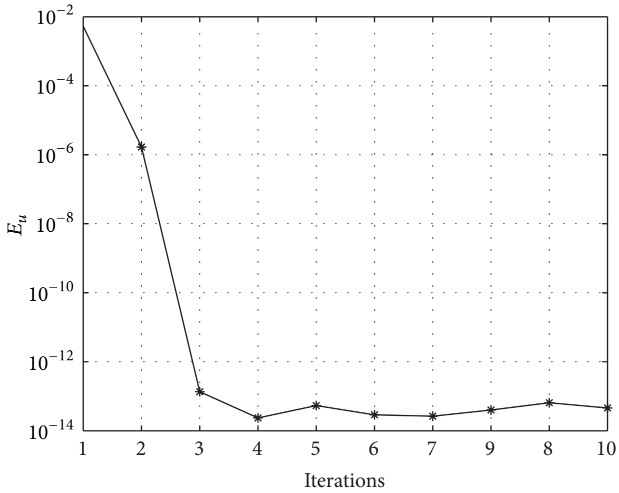


Figure 13: Fishers equation convergence graph.

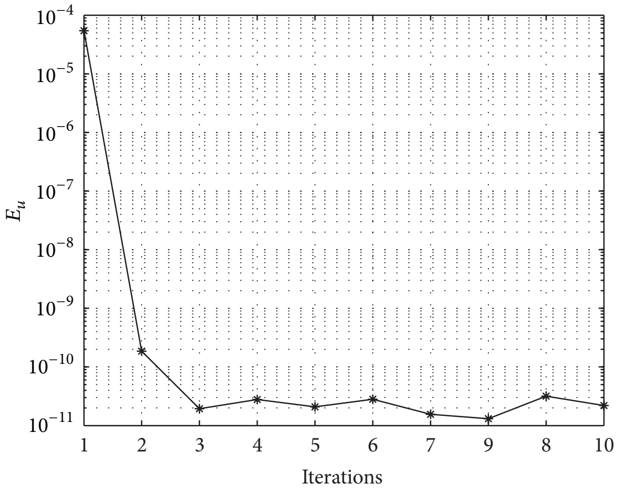


Figure 16: Burgers-Huxley equation convergence graph.

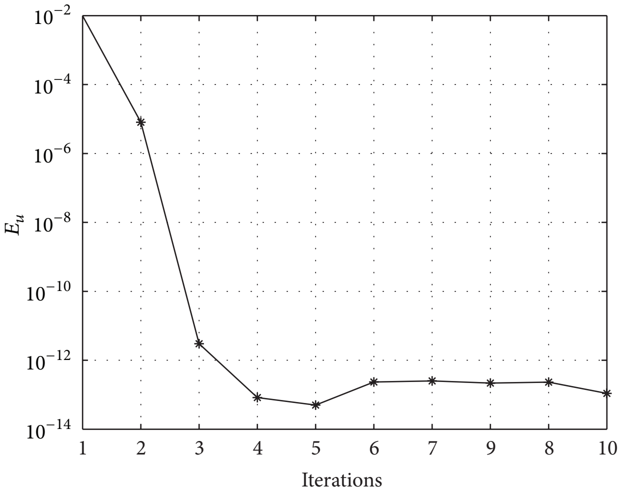


Figure 14: Burger-Fishers equation convergence graph.

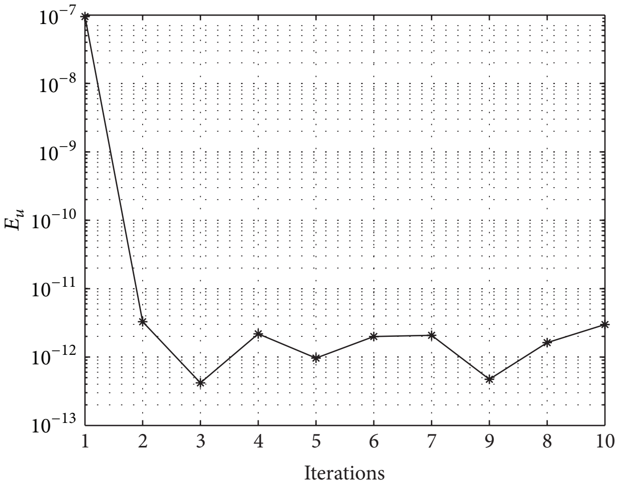


Figure 17: Modified KdV-Burger equation convergence graph.

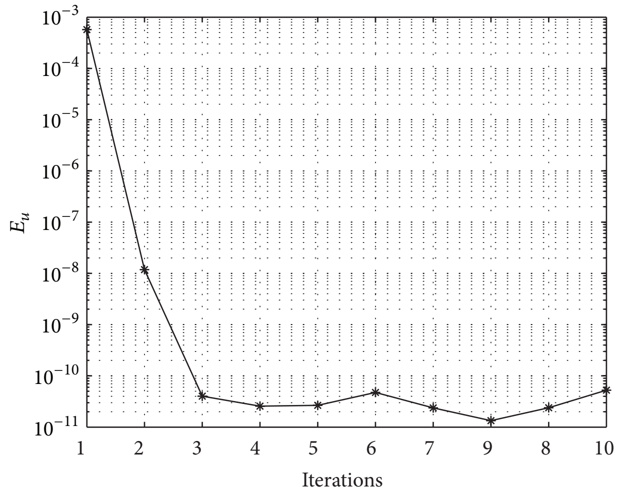


Figure 15: Fitzhugh-Nagumo equation convergence graph.

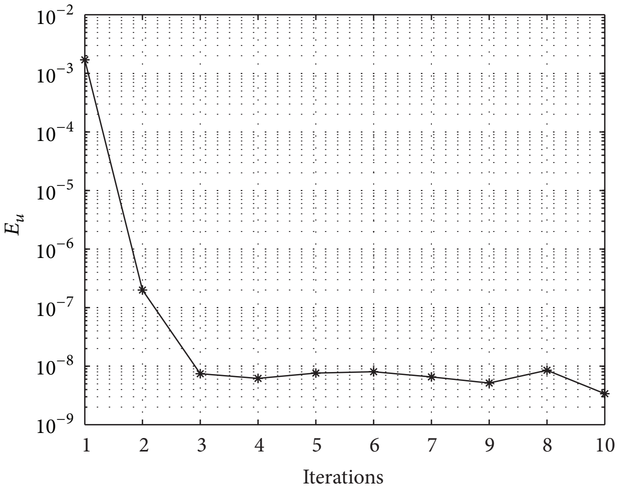


Figure 18: Modified KdV equation convergence graph.

not improve with an increase in the number of iterations. This plateau level gives an estimate of the maximum error that can be achieved when using the proposed method with a certain number of collocation points. It is worth remarking that the accuracy of the method depends on the number of collocation points in both the x and t directions. The results from Figures 13–18 clearly demonstrate that the BI-SQLM is accurate.

5. Conclusion

This paper has presented a new Chebyshev collocation spectral method for solving general nonlinear evolution partial differential equations. The bivariate interpolated spectral quasilinearisation method (BI-SQLM) was developed by combining elements of the quasilinearisation method and Chebyshev spectral collocation with bivariate Lagrange interpolation. The main goal of the current study was to assess the accuracy, robustness, and effectiveness of the method in solving nonlinear partial differential equations.

Numerical simulations were conducted on the modified KdV-Burger equation, highly nonlinear modified KdV equation, the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, and Burgers-Huxley equation. It is evident from the study that the BI-SQLM gives accurate results in a computationally efficient manner. Further evidence from this study is that the BI-SQLM gives solutions that are uniformly accurate and valid in large intervals of space and time domains. The apparent success of the method can be attributed to the use of the Chebyshev spectral collocation method with bivariate Lagrange interpolation in space and time for differentiating. This work contributes to the existing body of literature on quasilinearisation tools for solving complex nonlinear partial differential equations. Further work needs to be done to establish whether the BI-SQLM can be equally successful in solving coupled systems of equations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported in part by the National Research Foundation of South Africa (Grant no. 85596).

References

[1] G. Adomian, *Stochastic Systems*, vol. 169 of *Mathematics in Science and Engineering*, Academic Press, Orlando, Fla, USA, 1983.

[2] G. Adomian, “A review of the decomposition method in applied mathematics,” *Journal of Mathematical Analysis and Applications*, vol. 135, no. 2, pp. 501–544, 1988.

[3] L. Bougoffa and R. C. Rach, “Solving nonlocal initial-boundary value problems for linear and nonlinear parabolic and hyperbolic partial differential equations by the Adomian decomposition method,” *Applied Mathematics and Computation*, vol. 225, pp. 50–61, 2013.

[4] S. J. Liao, *Advances in Homotopy Analysis Method*, World Scientific Publishing, Singapore, 2014.

[5] J. He, “Application of homotopy perturbation method to nonlinear wave equations,” *Chaos, Solitons and Fractals*, vol. 26, no. 3, pp. 695–700, 2005.

[6] S. Abbasbandy, “The application of homotopy analysis method to solve a generalized Hirota-Satsuma coupled KdV equation,” *Physics Letters A: General, Atomic and Solid State Physics*, vol. 361, no. 6, pp. 478–483, 2007.

[7] L. Song and H. Zhang, “Application of homotopy analysis method to fractional KdV-Burgers-KURamoto equation,” *Physics Letters A*, vol. 367, no. 1-2, pp. 88–94, 2007.

[8] E. J. Parkes and B. R. Duffy, “An automated tanh-function method for finding solitary wave solutions to non-linear evolution equations,” *Computer Physics Communications*, vol. 98, no. 3, pp. 288–300, 1996.

[9] B. R. Duffy and E. J. Parkes, “Travelling solitary wave solutions to a seventh-order generalized KdV equation,” *Physics Letters A*, vol. 214, no. 5-6, pp. 271–272, 1996.

[10] Z. B. Li, “Exact solitary wave solutions of nonlinear evolution equations,” in *Mathematics Mechanization and Application*, X. S. Gao and D. M. Wang, Eds., Academic Press, San Diego, Calif, USA, 2000.

[11] U. Lepik, “Numerical solution of evolution equations by the Haar wavelet method,” *Applied Mathematics and Computation*, vol. 185, no. 1, pp. 695–704, 2007.

[12] I. Celik, “Haar wavelet method for solving generalized Burgers-Huxley equation,” *Arab Journal of Mathematical Sciences*, vol. 18, no. 1, pp. 25–37, 2012.

[13] G. Hariharan, K. Kannan, and K. R. Sharma, “Haar wavelet method for solving Fisher's equation,” *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 284–292, 2009.

[14] J. He and X. Wu, “Exp-function method for nonlinear wave equations,” *Chaos, Solitons & Fractals*, vol. 30, no. 3, pp. 700–708, 2006.

[15] C. Chun, “Solitons and periodic solutions for the fifth-order KdV equation with the Exp-function method,” *Physics Letters A*, vol. 372, no. 16, pp. 2760–2766, 2008.

[16] X. H. Wu and J. H. He, “EXP-function method and its application to nonlinear equations,” *Chaos, Solitons & Fractals*, vol. 38, no. 3, pp. 903–910, 2008.

[17] F. W. Wubs and E. D. de Goede, “An explicit-implicit method for a class of time-dependent partial differential equations,” *Applied Numerical Mathematics*, vol. 9, no. 2, pp. 157–181, 1992.

[18] E. M. E. Elbarbary and M. El-Kady, “Chebyshev finite difference approximation for the boundary value problems,” *Applied Mathematics and Computation*, vol. 139, no. 2-3, pp. 513–523, 2003.

[19] A. C. Vliegenthart, “On finite-difference methods for the Korteweg-de Vries equation,” *Journal of Engineering Mathematics*, vol. 5, pp. 137–155, 1971.

[20] J. Argyris and M. Haase, “An engineer’s guide to soliton phenomena: application of the finite element method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 61, no. 1, pp. 71–122, 1987.

[21] G. F. Carey and Y. Shen, “Approximations of the KdV equation by least squares finite elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 93, no. 1, pp. 1–11, 1991.

[22] K. Djidjeli, W. G. Price, P. Temarel, and E. H. Twizell, “A linearized implicit pseudo-spectral method for certain nonlinear water wave equations,” *Communications in Numerical*

Methods in Engineering with Biomedical Applications, vol. 14, no. 10, pp. 977–993, 1998.

[23] A. H. Khater, R. S. Temsah, and M. M. Hassan, "A Chebyshev spectral collocation method for solving Burger's-type equations," *Journal of Computational and Applied Mathematics*, vol. 222, no. 2, pp. 333–350, 2008.

[24] D. Olmos and B. D. Shizgal, "A pseudospectral method of solution of Fisher's equation," *Journal of Computational and Applied Mathematics*, vol. 193, no. 1, pp. 219–242, 2006.

[25] M. Javidi, "Spectral collocation method for the solution of the generalized Burger-Fisher equation," *Applied Mathematics and Computation*, vol. 174, no. 1, pp. 345–352, 2006.

[26] M. Javidi, "A numerical solution of the generalized Burgers-Huxley equation by spectral collocation method," *Applied Mathematics and Computation*, vol. 178, no. 2, pp. 338–344, 2006.

[27] M. Dehghan and F. Fakhar-Izadi, "Pseudospectral methods for Nagumo equation," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 27, no. 4, pp. 553–561, 2011.

[28] T. A. Driscoll, "A composite Runge-Kutta method for the spectral solution of semilinear PDEs," *Journal of Computational Physics*, vol. 182, no. 2, pp. 357–367, 2002.

[29] M. T. Darvishi, S. Kheybari, and F. Khani, "A numerical solution of Korteweg-de Vries equation by pseudospectral method using Darvishi's preconditionings," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 98–105, 2006.

[30] M. T. Darvishi, S. Kheybari, and F. Khani, "Spectral collocation method and Darvishi's preconditionings to solve the generalized Burgers-Huxley equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 10, pp. 2091–2103, 2008.

[31] B. A. Jacobs and C. Harley, "Two hybrid methods for solving two-dimensional linear time-fractional partial differential equations," *Abstract and Applied Analysis*, vol. 2014, Article ID 757204, 10 pages, 2014.

[32] E. Tohidi and A. Kilicman, "An efficient spectral approximation for solving several types of parabolic pdes with nonlocal boundary conditions," *Mathematical Problems in Engineering*, vol. 2014, Article ID 369029, 6 pages, 2014.

[33] R. E. Bellman and R. E. Kalaba, *Quasilinearization and Nonlinear Boundary-Value Problems*, vol. 3 of *Modern Analytic and Computional Methods in Science and Mathematics*, American Elsevier, New York, NY, USA, 1965.

[34] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, Philadelphia, Pa, USA, 2000.

[35] A. Golbabai and M. Javidi, "A spectral domain decomposition approach for the generalized Burger's-Fisher equation," *Chaos, Solitons & Fractals*, vol. 39, no. 1, pp. 385–392, 2009.

[36] A. Wazwaz and A. Gorguis, "An analytic study of Fisher's equation by using Adomian decomposition method," *Applied Mathematics and Computation*, vol. 154, no. 3, pp. 609–620, 2004.

[37] H. Li and Y. Guo, "New exact solutions to the FitzHugh-Nagumo equation," *Applied Mathematics and Computation*, vol. 180, no. 2, pp. 524–528, 2006.

[38] E. Fan, "Traveling wave solutions for nonlinear equations using symbolic computation," *Computers & Mathematics with Applications*, vol. 43, no. 6-7, pp. 671–680, 2002.

[39] Y. N. Kyrychko, M. V. Bartuccelli, and K. B. Blyuss, "Persistence of travelling wave solutions of a fourth order diffusion system," *Journal of Computational and Applied Mathematics*, vol. 176, no. 2, pp. 433–443, 2005.

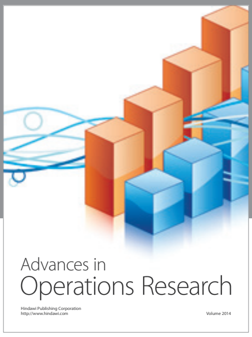
[40] I. Hashim, M. S. M. Noorani, and M. R. Said Al-Hadidi, "Solving the generalized Burgers-Huxley equation using the Adomian decomposition method," *Mathematical and Computer Modelling*, vol. 43, no. 11-12, pp. 1404–1411, 2006.

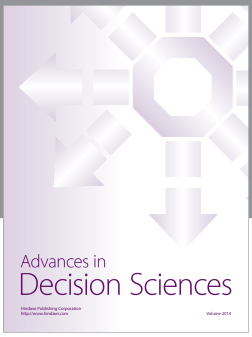
[41] X. Y. Wang, Z. S. Zhu, and Y. K. Lu, "Solitary wave solutions of the generalised Burgers-Huxley equation," *Journal of Physics A: Mathematical and General*, vol. 23, no. 3, pp. 271–274, 1990.

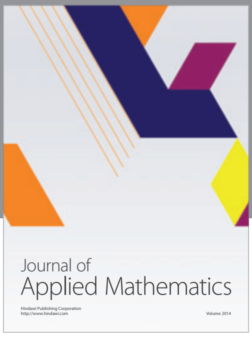
[42] M. A. Helal and M. S. Mehanna, "A comparison between two different methods for solving KdV-Burgers equation," *Chaos, Solitons and Fractals*, vol. 28, no. 2, pp. 320–326, 2006.

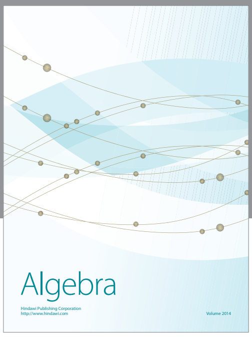
[43] J. M. Burgers, "A mathematical model illustrating the theory of turbulence," in *Advances in Applied Mechanics*, vol. 1, pp. 171–199, 1948.

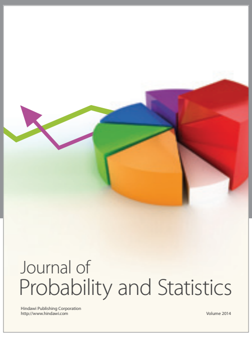
[44] J. D. Cole, "On a quasi-linear parabolic equation occurring in aerodynamics," *Quarterly of Applied Mathematics*, vol. 9, pp. 225–236, 1951.


Advances in
Operations Research

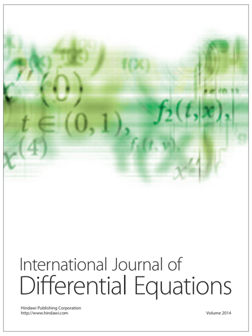

Advances in
Decision Sciences


Journal of
Applied Mathematics

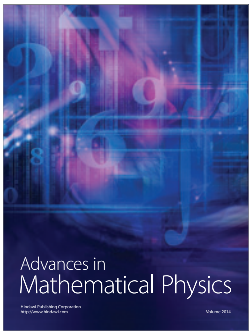

Algebra

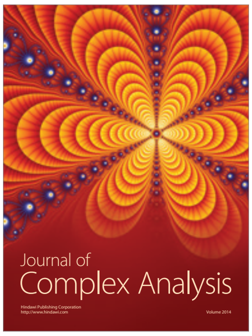

Journal of
Probability and Statistics

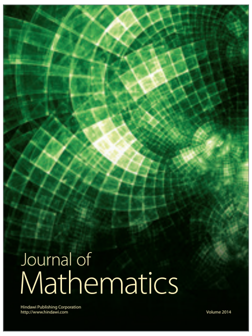
The Scientific
World Journal

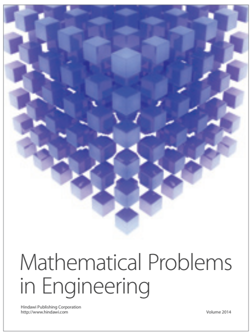

International Journal of
Differential Equations

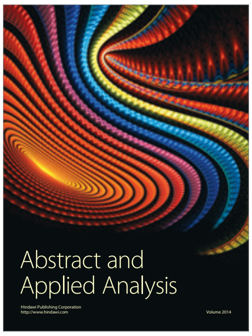
International Journal of
Combinatorics

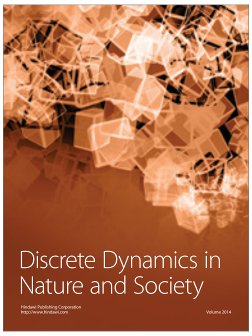

Advances in
Mathematical Physics

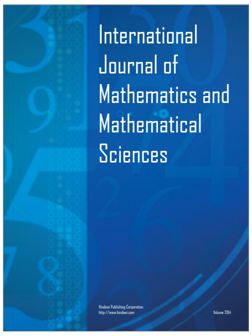

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

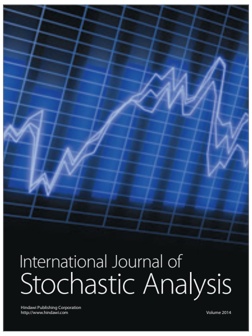

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization