A new compact finite difference quasilinearization method for nonlinear evolution partial differential equations

Abstract: This article presents a new method of solving partial differential equations. The method is an improvement of the previously reported compact finite difference quasilinearization method (CFDQLM) which is a combination of compact finite difference schemes and quasilinearization techniques. Previous applications of compact finite difference (FD) schemes when solving parabolic partial differential equations has been solely on discretizing the spatial variables and another numerical technique used to discretize temporal variables. In this work we attempt, for the first time, to use the compact FD schemes in both space and time. This ensures that the rich benefits of the compact FD schemes are carried over to the time variable as well, which improves the overall accuracy of the method. The proposed method is tested on four nonlinear evolution equations. The method produced highly accurate results which are portrayed in tables and graphs.

Keywords: Compact finite differences, Quasilinearization, Nonlinear evolution equations

MSC: 35K61, 65M06

1 Introduction

Many researchers are now using higher order compact finite difference schemes in place of the conventional second order finite difference to solve differential equations [1–5]. This is because significant improvements to the accuracy of numerical solutions have been obtained by using fourth or sixth order compact FD schemes. Another advantage is that the high accuracy is obtained on coarser grids which ensures greater computational efficiency. Lele [6] presented various compact finite difference schemes for applications such as evaluating high order derivatives, interpolation and filtering.

A number of researchers have used compact FD schemes to solve partial differential equations arising in various fields. In all the previous work the compact FD schemes were only applied on space variables and other discretization techniques are used for the time variables. Quite a number of numerical techniques have been paired with the compact FD schemes in order to discretize time. For example [7] used a two-step predictor-corrector algorithm called McCormack method to find the solution of Burgers equation. Li and Vishal [3] used the classical fourth-order four-stage Runge-Kutta method (RK4). A lot of of researchers have combined the compact FD schemes with the third order total variation diminishing Runge-Kutta (TVD-RK3) scheme (see [4, 8–10]). Dlamini et al [11] combined the compact FD schemes with the Crank-Nicolson technique in time to solve unsteady boundary layer flow problems.
However these techniques limit the accuracy of the difference scheme to fourth order or less in time which means small mesh sizes have to be used in order to get desirable accuracy, and thus much more computational work is involved. The aim is to try the compact FD schemes (sixth order) on the time variable as well which will allow us to archive greater accuracy with a rough step size.

The purpose of this investigation is to explore the possibility of applying sixth order compact FD schemes in both space and time variables. We consider this on a few examples of nonlinear evolution PDEs. These are important equations arising in a number of fields of science and engineering. They are used to describe many complex nonlinear settings in applications such as vibration and wave propagation, fluid mechanics, plasma physics, quantum mechanics, nonlinear optics, solid state physics, chemical kinematics, physical chemistry, population dynamics, and many other areas of mathematical modelling. Understanding the behavior of these problems requires finding solutions of the differential equations and analyzing them. Most of the evolution equations are very complex to solve analytically and so we rely on approximate solutions. So the development of numerical solutions to solve such problems continues to be an active area of research.

Numerical experiments show that implementing the compact FD schemes on both space and time yield highly accurate results. This was experimented on the Fisher’s, Burgers-Fisher, Burgers-Huxley and Fitzhugh-Nagumo equations. Before using the compact FD schemes we first linearize the PDEs using a quasilinearization technique developed by Bellman and Kalaba. The method is therefore called the compact finite difference quasilinearization method (CFDQLM). The rest of the paper can be summarized as follows, in section 2 we give a description of how to use the sixth order compact finite differences to discretize in space and in time as well as giving a description of how the CFDQLM is used to solve evolution partial differential equations. In section 3 we outline the examples considered in this work. We then present and discuss results in section 4 and in the last section we conclude.

2 Compact finite difference schemes

In this section we introduce the compact finite difference schemes and illustrate how we use them to solve nonlinear evolution partial differential equations. We consider nonlinear PDEs of the form

\[
\frac{\partial u}{\partial t} = G \left( u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2} \right)
\]

defined on the the region

\[ t \in [0, T], \quad x \in [a, b] \]

with boundary condition

\[ u(a, t) = u_a, \quad u(b, t) = u_b. \]

In the derivation of the compact FD schemes we consider the function \( u(x, t) \) that depends on space variable \( x \) and temporal variable \( t \). We consider one-dimensional uniform meshes on the regions \([a, b]\) and \([0, T]\), with nodes \( x_i \) \((i = 1, 2, \ldots, N_x)\) and \( t_j \((j = 1, 2, \ldots, N_t)\) respectively, where

\[ a = x_1 < x_2 < \cdots < x_{N_x} = b \]

and

\[ 0 = t_1 < t_2 < \cdots < t_{N_t} = T \]

The distance between any two successive nodes is a constant \( \Delta x = x_i - x_{i-1} \) for the space variable and \( \Delta t = t_j - t_{j-1} \) for the temporal variable. We first show how we approximate the derivatives with respect to \( x \) using the compact FD schemes. Sixth order approximations of the first and second derivatives with respect to \( x \), at interior nodes can be obtained using the following schemes (see [6] for details)

\[
\frac{1}{3} u'_{i-1,j} + u'_{i,j} + \frac{1}{3} u'_{i+1,j} = \frac{14}{9} u_{i+1,j} - u_{i-1,j} + \frac{1}{9} u_{i+2,j} - y_{i-2,j} + \frac{2}{4} u_i + y_{i+2,j} - y_{i-2,j} \quad \text{for} \quad i = 1, 2, \ldots, N_x - 1,
\]

\[
\frac{2}{11} u''_{i-1,j} + u''_{i,j} + \frac{2}{11} u''_{i+1,j} = \frac{12}{11} u_{i+1,j} - u_{i-1,j} + \frac{3}{11} u_{i+2,j} - y_{i-2,j} + \frac{3}{8} u_i + y_{i+2,j} - y_{i-2,j} \quad \text{for} \quad i = 1, 2, \ldots, N_x - 1,
\]

where \( u' \) and \( u'' \) denote the first and second derivatives with respect to \( x \), respectively.
where \( u_{i,j} = u(x_i, t_j) \) and the primes denote differentiation with respect to \( x \). We apply the compact FD approximation for the first and second derivatives given by (4) and (5) respectively, at the interior nodes \( (i = 2, \ldots, N_x - 1) \). Since we know boundary conditions at \( i = 1 \) and \( i = N_x \), the compact FD schemes must be adjusted for the nodes near the boundary points. In order to maintain the order \( O(h^6) \) accuracy at the boundary points as in the interior points and to maintain the same tridiagonal format, we use the following one sided scheme at \( i = 2 \),

\[
\frac{u''_{2,j} + \frac{1}{3}u'''_{3,j}}{\Delta x} = \frac{1}{\Delta x} \left( \frac{-7}{45}u_{1,j} + \frac{17}{12}u_{2,j} + \frac{83}{36}u_{3,j} - \frac{11}{9}u_{4,j} + \frac{2}{3}u_{5,j} - \frac{37}{180}u_{6,j} + \frac{1}{36}u_{7,j} \right),
\]

and when \( i = N_x - 1 \) we use

\[
\frac{1}{3}u'_{N_x - 2,j} + u'_{N_x - 1,j} = \frac{1}{\Delta x} \left( \frac{7}{45}u_{N_x,j} + \frac{17}{12}u_{N_x - 1,j} - \frac{83}{36}u_{N_x - 2,j} + \frac{11}{9}u_{N_x - 3,j} - \frac{2}{3}u_{N_x - 4,j} + \frac{37}{180}u_{N_x - 5,j} + \frac{1}{36}u_{N_x - 6,j} \right).
\]

Similarly, for the second derivatives, we use

\[
\frac{u''_{2,j} + \frac{2}{11}u'''_{3,j}}{\Delta x^2} = \frac{1}{(\Delta x)^2} \left( \frac{31}{45}u_{1,j} + \frac{19}{110}u_{2,j} - \frac{339}{110}u_{3,j} + \frac{1933}{396}u_{4,j} - \frac{40}{11}u_{5,j} + \frac{96}{55}u_{6,j} - \frac{479}{990}u_{7,j} + \frac{13}{220}u_{8,j} \right),
\]

at \( i = 2 \) and

\[
\frac{2}{11}u''_{N_x - 2,j} + u''_{N_x - 1,j} = \frac{1}{(\Delta x)^2} \left( \frac{31}{45}u_{N_x,j} + \frac{19}{110}u_{N_x - 1,j} - \frac{339}{110}u_{N_x - 2,j} + \frac{1933}{396}u_{N_x - 3,j} - \frac{40}{11}u_{N_x - 4,j} + \frac{96}{55}u_{N_x - 5,j} + \frac{479}{990}u_{N_x - 6,j} + \frac{13}{220}u_{N_x - 7,j} \right)
\]

at \( i = N_x - 1 \). Using the above equations, the equations for approximating the first and second order derivatives can be expressed as

\[
A_x U'_{i,j} = B_x U_{i,j} + K_x,
\]

\[
A_{xx} U''_{i,j} = B_{xx} U_{i,j} + K_{xx},
\]

where

\[
A_x = \begin{bmatrix}
1 & \frac{1}{3} & & & \\
\frac{1}{3} & 1 & \frac{1}{3} & & \\
& \frac{1}{3} & 1 & \frac{1}{3} & \\
& & \ddots & \ddots & \ddots \\
& & & \frac{1}{3} & 1 & \\
& & & & \frac{1}{3} & 1
\end{bmatrix}_{(N_x-2)\times(N_x-2)},
\]

\[
K_x = \frac{1}{\Delta x} \begin{bmatrix}
\frac{-7}{45}y_1 \\
\frac{-36}{y_1} \\
0 \\
\vdots \\
0 \\
\frac{-9y_N}{36} \\
\frac{75y_N}{36}
\end{bmatrix}_{(N_x-2)\times1}.
\]
A new compact finite difference quasilinearization method

\[ B_x = \frac{1}{\Delta x} \begin{bmatrix} \frac{-17}{12} & \frac{83}{36} & -\frac{11}{9} & \frac{2}{3} & -\frac{37}{180} & \frac{1}{36} \\ \frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} \\ -\frac{1}{36} & \frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{-1}{36} & \frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} \\ -\frac{1}{36} & \frac{37}{180} & \frac{2}{3} & \frac{11}{9} & -\frac{83}{36} \end{bmatrix} \in (N_x-2) \times (N_x-2) \]

\[ A_{xx} = \begin{bmatrix} \frac{1}{2} & \frac{1}{11} \\ \frac{2}{11} & \frac{1}{2} \\ \frac{2}{11} & \frac{1}{2} \\ \frac{2}{11} & \frac{1}{2} \end{bmatrix}, \quad K_{xx} = \frac{1}{(\Delta x)^2} \begin{bmatrix} \frac{31}{45} \gamma_1 \\ \frac{3}{44} \gamma_1 \\ 0 \\ \frac{3}{44} \gamma_N \end{bmatrix} \in (N_x-2) \times 1 \]

\[ B_{xx} = \frac{1}{(\Delta x)^2} \begin{bmatrix} \frac{3}{44} \gamma_N & \frac{31}{45} \gamma_1 \\ \frac{3}{44} \gamma_1 & 0 \end{bmatrix} \in (N_x-2) \times (N_x-2) \]

From equations (10) and (11), \( U' \) and \( U'' \) can be expressed as;

\[ U'_{i,j} = E_x U_{i,j} + H_x \]

\[ U''_{i,j} = E_{xx} U_{i,j} + H_{xx} \]

where

\[ E_x = A_x^{-1} B_x, \quad E_{xx} = A_x^{-1} B_{xx}, \quad H_x = A_x^{-1} K_x, \quad H_{xx} = A_x^{-1} K_{xx} \]

Next we show how we use the compact FD schemes to approximate the time derivatives. Approximation of the first derivative at interior points is given by

\[ \frac{1}{3} U'_{i,j-1} + \dot{U}_{i,j} + \frac{1}{3} U'_{i,j+1} = \frac{14}{9} \frac{U_{i,j+1} - U_{i,j-1}}{2\Delta t} + \frac{1}{9} \frac{U_{i,j+2} - 2U_{i,j} + U_{i,j-2}}{4\Delta t}. \]
where the dots denote differentiation with respect to time and \( \Delta t \) is the distance between successive nodes. We adjust the schemes for \( i = 1, 2, N_t - 1 \) and \( N_t \) with the following one sided schemes respectively.

\[
\frac{1}{3} \dot{u}_{i+1} + \frac{1}{3} \dot{u}_{i+2} = \frac{1}{\Delta t} \left( -\frac{451}{180} u_{i+1} + \frac{1003}{180} u_{i+2} - \frac{20}{3} u_{i+3} + \frac{55}{9} u_{i+4} - \frac{125}{36} u_{i+5} + \frac{67}{60} u_{i+6} - \frac{7}{45} u_{i+7} \right),
\]

(16)

\[
\frac{1}{3} \dot{u}_{i+1} + \frac{1}{3} \dot{u}_{i} = \frac{1}{\Delta t} \left( \frac{35}{36} u_{i+1} + \frac{7}{12} u_{i+2} - \frac{7}{36} u_{i+3} + \frac{17}{12} u_{i+4} - \frac{7}{36} u_{i+5} + \frac{1}{36} u_{i+6} \right),
\]

(17)

\[
\frac{1}{3} \dot{u}_{i+1} + \dot{u}_{i} + \frac{1}{3} \dot{u}_{i-1} = \frac{1}{\Delta t} \left( \frac{35}{36} u_{i+1} - \frac{7}{12} u_{i+2} - \frac{7}{36} u_{i+3} - \frac{7}{12} u_{i+4} + \frac{7}{36} u_{i+5} - \frac{1}{36} u_{i+6} \right),
\]

(18)

The equation for approximating the first time derivative is obtained by combining equations (15) - (19) and is given by

\[
A_t \dot{U}_{i,.} = B_t U_{i,.},
\]

(20)

where \( A_t = A_x \) and

\[
B_t = \frac{1}{\Delta t} \begin{bmatrix}
-\frac{451}{180} & \frac{1003}{180} & -\frac{20}{3} & \frac{55}{9} & -\frac{125}{36} & \frac{67}{60} & -\frac{7}{45} \\
-\frac{35}{36} & \frac{7}{12} & -\frac{7}{36} & 1 & -\frac{7}{12} & \frac{7}{36} & -\frac{1}{36} \\
-\frac{1}{36} & \frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} \\
& & \ddots & \ddots & \ddots & \ddots & \ddots \\
& & & & & & & \\
-\frac{1}{36} & -\frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} \\
\frac{1}{36} & -\frac{7}{12} & \frac{7}{36} & -1 & \frac{7}{12} & \frac{35}{36} \\
\frac{7}{45} & -\frac{67}{60} & \frac{125}{36} & -\frac{55}{9} & \frac{20}{3} & -\frac{1003}{180} & \frac{451}{180}
\end{bmatrix} N_t \times N_t
\]

From equation (20), \( \dot{U} \) can be expressed as

\[
\dot{U}_{i,.} = E_t U_{i,.} = \sum_{k=1}^{N_t} e_{j,k} u(x_j, t_k), \quad i = 2, 3, \ldots, N_x - 1, \quad j = 1, 2, \ldots, N_t
\]

(21)

where \( E_t = A_t^{-1} B_t \) and \( e_{j,k} \) are the elements of \( E_t \) and \( U_{i,.} = [u_{i,1}, u_{i,2}, \ldots, u_{i,N_t}] \).

To solve the PDE (1) we start by linearizing it by using the quasilinearization method which was proposed by Bellman and Kalaba [12]. It is convenient to split the function \( G \) in (1) into its linear and nonlinear components and rewrite the governing equation in the form,

\[
L[u, u', u''] + N[u, u', u''] - \dot{u} = 0,
\]

(22)

where the dot and primes denote the time and space derivatives, respectively. \( L \) is a linear operator and \( N \) is a non-linear operator. If we assume that the difference \( u_{r+1} - u_r \) and all its space derivatives is small, then we can approximate the non-linear operator \( N \) using the linear terms of the Taylor series and hence

\[
N[u, u', u''] \approx N[u_r, u'_r, u''_r] + \sum_{k=0}^{2} \frac{\partial N}{\partial u^{(k)}} (u^{(k)}_{r+1} - u^{(k)}_r)
\]

(23)
where \( r \) and \( r + 1 \) denote previous and current iterations respectively.

Equation (23) can be expressed as

\[
N[u, u', u''] \approx N[u_r, u_r', u_r''] + \sum_{k=0}^{2} \phi_{k,r}[u_r, u_r', u_r'']u_r^{(k)} - \sum_{k=0}^{2} \phi_{k,r}[u_r, u_r', u_r'']u_r^{(k)}
\]  

(24)

where

\[
\phi_{k,r}[u_r, u_r', u_r''] = \frac{\partial N}{\partial u^{(k)}}[u_r, u_r', u_r'].
\]

(25)

Substituting equation (24) into equation (22), we get

\[
L[u_{r+1}, u_{r+1}', u_{r+1}''] + \sum_{k=0}^{2} \phi_{k,r}u_{r+1}^{(k)} - \bar{u}_{r+1} = R_r[u_r, u_r', u_r']
\]  

(26)

where

\[
R_r[u_r, u_r', u_r''] = \sum_{k=0}^{2} \phi_{k,r}u_r^{(k)} - N[u_r, u_r', u_r''].
\]

Substituting (21) into (26) we get

\[
L[U_{r+1,j}, U_{r+1,j}', U_{r+1,j}''] + \sum_{k=0}^{2} \Phi_{k,r}U_{r+1,j}^{(k)} - \sum_{k=0}^{N_r} e_{j,k}U_{r+1,k} = R_r[U_{r,j}, U_{r,j}', U_{r,j}'']
\]  

(27)

for \( j = 1, 2, 3, \ldots, N_r \), where

\[
\Phi_{k,r} = \begin{bmatrix}
\phi_{k,r}(x_2, t_j) \\
\phi_{k,r}(x_1, t_j) \\
\vdots \\
\phi_{k,r}(x_{N_x-1}, t_j)
\end{bmatrix}
\]

(28)

and

\[
U_{r+1,j} = [u_{r+1,j}(x_2), u_{r+1,j}(x_3), \ldots, u_{r+1,j}(x_{N_x-1})]
\]

(29)

Since the initial condition is known, then we express equation (27) as

\[
L[U_{r+1,j}, U_{r+1,j}', U_{r+1,j}''] + \sum_{k=0}^{2} \Phi_{k,r}U_{r+1,j}^{(k)} - \sum_{k=0}^{N_r} e_{j,k}U_{r+1,k} = R_j
\]  

(30)

where

\[
R_j = R_r[U_{r,j}, U_{r,j}', U_{r,j}''] + e_{j,1}U_1 - L[H_x, H_{xx}] - \Phi_{1,r}H_x - \Phi_{2,r}H_{xx}, \quad j = 2, 3, \ldots, N_r.
\]

(31)

Equation (30) can be expressed as the following \((N_r - 1)(N_x - 1) \times (N_r - 1)(N_x - 1)\) matrix system

\[
\begin{bmatrix}
X_{2,2} & X_{2,3} & \cdots & X_{2,N_x} \\
X_{3,2} & X_{3,3} & \cdots & X_{3,N_x} \\
\vdots & \vdots & \ddots & \vdots \\
X_{N_x,2} & X_{N_x,3} & \cdots & X_{N_x,N_x}
\end{bmatrix}
\begin{bmatrix}
U_2 \\
U_3 \\
\vdots \\
U_{N_x}
\end{bmatrix}
= \begin{bmatrix}
R_2 \\
R_3 \\
\vdots \\
R_{N_x}
\end{bmatrix}
\]

(32)

where

\[
X_{i,j} = L[I, E_x, E_{xx}] + \Phi_{0,r} + \Phi_{1,r}E_x + \Phi_{2,r}E_{xx} - e_{i,j}I
\]

(33)

\[
X_{i,j} = -e_{i,j}I, \quad \text{when } i \neq j,
\]

(34)

\[
i, j = 2, 3, \ldots, N_r
\]

and \( I \) is the identity matrix of size \((N_x - 1) \times (N_x - 1)\). Solving equation (28) gives \( u(x_i, t_j) \).
3 Numerical experiments

We apply the proposed algorithm to well-known nonlinear PDEs of the form (1) with exact solutions. In order to determine the level of accuracy of the CFDQLM approximate solution, at a particular time level, in comparison with the exact solution we report maximum error which is defined by

$$E_N = \max \{ |u(x_r, t) - \tilde{u}(x_r, t)| : 0 \leq r \leq N \},$$

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

**Example 3.1** (Fisher’s equation). We consider the Fisher’s equation which represents a reactive-diffusive system and is encountered in chemical kinetics and population dynamics applications. The equation is given by

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

subject to the initial condition

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

and exact solution [13]

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

where $\alpha$ is a constant. For this example, the appropriate linear operator $L$ and nonlinear operator $N$ are chosen as

$$L(u) = u'' + \alpha u, \quad N(u) = -\alpha u^2.$$

We linearize the nonlinear operator $N$ by expanding using the Taylor series expansion. We assume that the difference $u_{r+1} - u_r$ and its derivatives is very small.

$$N \approx N[u_r, u'_r, u''_r] + \sum_{k=0}^2 \phi_{k,r} u_r^{(k)} - \sum_{k=0}^2 \phi_{k,r} u_r^{(k)}$$

$$= \alpha u_r^2 - 2\alpha u_r (u_{r+1} - u_r)$$

The coefficients are obtained by taking only the linear terms of the Taylor expansion and they are given as

$$\phi_{0,r} = \frac{\partial N}{\partial u}[u_r, u'_r, u''_r] = -2\alpha u_r$$

$$\phi_{1,r} = \frac{\partial N}{\partial u'}[u_r, u'_r, u''_r] = 0$$

$$\phi_{2,r} = \frac{\partial N}{\partial u''}[u_r, u'_r, u''_r] = 0$$

and

$$R_r = \sum_{k=0}^2 \phi_{k,r} u_r^{(k)} - N[u_r, u'_r, u''_r] = -\alpha u_r^2$$

which is consistent with the quasilinearization discussed in the previous section.

Therefore, the linearized equation can be expressed as

$$u''_{r+1} + \phi_{0,r} u_{r+1} + \alpha u_{r+1} - \tilde{u} = R_r$$
Applying the compact FD schemes both in \( x \) and \( t \), and initial condition, we get

\[
E_{xx} U_{r+1,j} + \Phi_{0,r} U_{r+1,j} + \alpha U_{r+1,j} - \sum_{k=2}^{N_t} c_{j,k} U_{r+1,k} = R_j
\]  

Equation (47) can be expressed as

\[
\begin{bmatrix}
X_{2,2} & X_{2,3} & \cdots & X_{2,N_t} \\
X_{3,2} & X_{3,3} & \cdots & X_{3,N_t} \\
\vdots & \vdots & \ddots & \vdots \\
X_{N_t,2} & X_{N_t,3} & \cdots & X_{N_t,N_t}
\end{bmatrix}
\begin{bmatrix}
U_2 \\
U_3 \\
\vdots \\
U_{N_t}
\end{bmatrix}
= 
\begin{bmatrix}
R_2 \\
R_3 \\
\vdots \\
R_{N_t}
\end{bmatrix}
\]  

where

\[
X_{i,i} = E_{xx} + \Phi_{0,r}^{(i)} + (\alpha - e_{i,i}) I \\
X_{i,j} = -e_{i,j} I, \quad \text{when } i \neq j, \\
R_j = R_r + e_{i,1} U_1 - H_{xx}
\]

The approximate solution is obtained by solving (48). The rest of the examples are solved in a similar manner.

**Example 3.2** (Burgers-Fisher equation). We consider the generalized Burgers-Fisher equation [14]

\[
\frac{\partial u}{\partial t} + \alpha u \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1 - u^\delta),
\]

with initial condition

\[
u(x, 0) = \left\{ \frac{1}{2} + \frac{1}{2} \tanh \left( \frac{-\alpha \delta}{2(\delta + 1)^2} \right) \right\}^{\frac{1}{\delta}},
\]

and exact solution

\[
u(x, t) = \left\{ \frac{1}{2} + \frac{1}{2} \tanh \left( \frac{-\alpha \delta}{2(\delta + 1)} \left[ x - \left( \frac{\alpha}{\delta + 1} + \frac{\beta(\delta + 1)}{\alpha} \right) t \right] \right) \right\}^{\frac{1}{\delta}},
\]

where \( \alpha, \beta \) and \( \delta \) are parameters. In this work the parameters are chosen to be \( \alpha = \beta = \delta = 1 \).

**Example 3.3** (Burgers-Huxley equation). We consider the Burgers-Huxley equation

\[
\frac{\partial u}{\partial t} + \alpha u^\delta u_x = \frac{\partial^2 u}{\partial x^2} + \beta u(1 - u^\delta)(u^\delta - \gamma),
\]

where \( \alpha, \beta \geq 0 \) are constant parameters, \( \delta \) 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

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

is

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

where

\[
r = \sqrt{\alpha^2 + 8\beta} \quad \text{and} \quad c = \frac{(\alpha - r)(2\gamma - 1) + 2\alpha}{4}
\]

The general solution (57) is reported in [15, 16].
Example 3.4 (Fitzhugh-Nagumo equation). The Fitzhugh-Nagumo equation is given by
\[
\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u(u - \alpha)(1 - u)
\]  
with initial condition
\[
\begin{align*}
    u(x, 0) &= \frac{1}{2} \left[ 1 - \coth \left( \frac{x}{2\sqrt{2}} \right) \right]. \\
    u(x, t) &= \frac{1}{2} \left[ 1 - \coth \left( \frac{x}{2\sqrt{2}} + \frac{2\alpha - 1}{4} t \right) \right].
\end{align*}
\]
This equation has the exact solution [17]
\[
\begin{align*}
    u(x, t) &= \frac{1}{2} \left[ 1 - \coth \left( \frac{x}{2\sqrt{2}} + \frac{2\alpha - 1}{4} t \right) \right],
\end{align*}
\]  
where \(\alpha\) is a parameter.

4 Results and discussion

In this section we present the results of the implementation of the CFDQLM on the nonlinear evolution equations mentioned above. In Tables 1 - 4 we show the maximum errors between the exact solutions and the CFDQLM results for the Fisher’s, Fishers-Burger, Burgers-Huxely and Fitzhugh-Nagumo equations respectively. The results were obtained with grid points \(N_x = 30\) for the space variable \(x\). As shown in the tables, for the time variable \(t\), the number of grid points \(N_t\) varied from 10 to 50. It can be seen that after \(N_t = 40\) there is a small change in the error. This means that \(N_t = 40\) is sufficient to give sufficient accuracy. As shown by the results, the method gives highly accurate results of at least \(10^{-12}\) which is achieved with remarkably few grid points in both \(x\) and \(t\). The same observation is seen in Figure 5 which shows the errors for each of the equations considered. It can be seen that the average error is around \(10^{-12}\) for the values of \(t\) considered. Another remarkable observation is that the method gives accurate results even for \(t > 1\). A lot of methods lose significant accuracy when \(t > 1\).

A graphical comparison between the CFDQLM results and the exact solutions is shown in Figures 1-4 for the Fisher’s, Burgers-Fisher, Burgers-Huxely, Fitzhugh-Nagumo equations respectively. A good agreement is observed between the two sets of results.

Figure 6 shows the convergence plots of the CFDQLM for the four equations. It can be seen that the method reaches full convergence after 4 iterations. The convergence criteria is defined as the infinity norm of the difference between the solution at current iteration and previous iteration, ie,
\[
||U_{r+1} - U_r||_\infty,
\]
where \(r\) and \(r + 1\) denote previous and current iterations. This shows that the method quickly converges to the true solution.

| \(t\) | \(N_t\) | 10   | 20   | 30   | 40   | 50   |
|------|--------|------|------|------|------|------|
| 0.2  | 3.396e-007 | 1.086e-009 | 3.592e-011 | 2.725e-012 | 2.734e-013 |
| 0.4  | 2.605e-008 | 1.951e-010 | 7.812e-012 | 7.934e-013 | 1.291e-013 |
| 0.6  | 8.689e-009 | 7.175e-011 | 3.761e-012 | 7.182e-013 | 1.475e-013 |
| 0.8  | 3.480e-009 | 6.084e-011 | 4.795e-012 | 8.085e-013 | 2.018e-013 |
| 1.0  | 3.342e-009 | 5.979e-011 | 2.542e-012 | 7.814e-013 | 1.212e-013 |
| 1.2  | 3.176e-009 | 5.649e-011 | 4.247e-012 | 6.779e-013 | 1.575e-013 |
| 1.4  | 2.766e-009 | 5.487e-011 | 1.562e-012 | 5.637e-013 | 2.598e-014 |
| 1.6  | 3.614e-009 | 6.219e-011 | 3.941e-012 | 5.197e-013 | 9.348e-014 |
| 1.8  | 6.478e-009 | 8.894e-011 | 7.390e-012 | 6.311e-013 | 2.415e-013 |
| 2.0  | 2.625e-008 | 5.748e-010 | 3.641e-011 | 4.902e-012 | 1.015e-012 |
Table 2. Maximum errors $E_N$ for Burgers-Fisher equation when $\alpha = \gamma = \delta = 1$ using $N_x = 30$

| $t$ | $N_t$ | 10  | 20  | 30  | 40  | 50  |
|-----|-------|-----|-----|-----|-----|-----|
| 0.2 | 8.512e-007 | 1.357e-008 | 9.097e-010 | 1.249e-010 | 2.632e-011 |
| 0.4 | 7.122e-008 | 2.127e-009 | 1.448e-010 | 2.060e-011 | 4.494e-012 |
| 0.6 | 1.655e-008 | 4.160e-010 | 2.949e-011 | 4.601e-012 | 1.080e-012 |
| 0.8 | 1.678e-008 | 6.381e-011 | 3.567e-012 | 5.065e-013 | 1.065e-013 |
| 1.0 | 3.027e-008 | 7.375e-011 | 1.066e-011 | 1.684e-012 | 4.735e-013 |
| 1.2 | 2.395e-008 | 9.651e-011 | 1.218e-011 | 2.303e-012 | 6.368e-013 |
| 1.4 | 5.896e-008 | 4.151e-011 | 1.330e-011 | 2.010e-012 | 5.536e-013 |
| 1.6 | 6.942e-008 | 6.562e-011 | 4.527e-012 | 1.202e-012 | 3.662e-013 |
| 1.8 | 1.183e-007 | 2.337e-010 | 9.879e-012 | 5.163e-013 | 1.923e-013 |
| 2.0 | 4.251e-007 | 1.529e-009 | 5.956e-011 | 6.207e-012 | 1.095e-012 |

Table 3. Maximum errors $E_N$ for Burgers-Huxley equation when $\alpha = \beta = 1$ and $\gamma = 0.5$ using $N_x = 30$

| $t$ | $N_t$ | 10  | 20  | 30  | 40  | 50  |
|-----|-------|-----|-----|-----|-----|-----|
| 0.2 | 3.247e-009 | 1.403e-011 | 6.777e-013 | 1.793e-014 | 1.066e-013 |
| 0.4 | 2.054e-010 | 2.187e-012 | 7.661e-014 | 6.317e-014 | 8.055e-014 |
| 0.6 | 1.191e-010 | 4.759e-013 | 3.342e-014 | 7.988e-014 | 6.156e-014 |
| 0.8 | 2.602e-011 | 1.812e-013 | 6.134e-014 | 5.929e-014 | 5.534e-014 |
| 1.0 | 1.151e-010 | 6.362e-014 | 3.586e-014 | 3.625e-014 | 4.829e-014 |
| 1.2 | 9.804e-011 | 9.576e-014 | 3.125e-014 | 4.452e-014 | 4.186e-014 |
| 1.4 | 2.114e-010 | 3.592e-013 | 5.662e-014 | 2.798e-014 | 5.851e-014 |
| 1.6 | 2.506e-010 | 9.730e-013 | 2.920e-014 | 2.542e-014 | 3.930e-014 |
| 1.8 | 4.486e-010 | 1.959e-012 | 1.222e-013 | 4.019e-014 | 2.298e-014 |
| 2.0 | 1.721e-009 | 1.364e-011 | 8.060e-013 | 1.065e-013 | 3.186e-014 |

Table 4. Maximum errors $E_N$ for Fitzhugh-Nagumo when $\alpha = 1$ using $N_x = 30$

| $t$ | $N_t$ | 10  | 20  | 30  | 40  | 50  |
|-----|-------|-----|-----|-----|-----|-----|
| 0.2 | 3.872e-009 | 2.016e-011 | 1.180e-012 | 1.693e-013 | 3.142e-014 |
| 0.4 | 2.486e-010 | 3.313e-012 | 1.953e-013 | 3.686e-014 | 2.287e-014 |
| 0.6 | 1.230e-010 | 8.290e-013 | 6.362e-014 | 2.059e-014 | 2.010e-014 |
| 0.8 | 6.523e-012 | 4.283e-013 | 5.429e-014 | 2.759e-014 | 3.003e-014 |
| 1.0 | 9.432e-011 | 3.036e-013 | 7.488e-014 | 2.803e-014 | 2.587e-014 |
| 1.2 | 6.756e-011 | 1.702e-013 | 6.334e-014 | 2.942e-014 | 3.675e-014 |
| 1.4 | 1.699e-010 | 4.746e-014 | 6.606e-014 | 3.830e-014 | 2.831e-014 |
| 1.6 | 1.924e-010 | 4.236e-013 | 3.775e-014 | 2.681e-014 | 4.013e-014 |
| 1.8 | 3.477e-010 | 1.222e-012 | 9.048e-014 | 4.147e-014 | 4.041e-014 |
| 2.0 | 1.355e-009 | 9.256e-012 | 5.085e-013 | 6.395e-014 | 2.986e-014 |
Fig. 1. Solution of the Fisher’s equation

Fig. 2. Solution of Burgers–Fisher equation

Fig. 3. Solution of the Burgers-Huxley equation
Fig. 4. Solution of the Fitzhugh-Nagumo equation

![Solution of the Fitzhugh-Nagumo equation](image)

Fig. 5. Error norms at different time levels for the Fishers, Burgers-Fisher, Burgers-Huxley and the Fitzhugh-Nagumo equations

![Error norms at different time levels](image)

Fig. 6. Convergence graphs for the Fishers, Burgers-Fisher, Burgers-Huxley and the Fitzhugh-Nagumo equations

![Convergence graphs](image)
5 Conclusion

In this paper, we have extended the use of compact finite difference schemes to both space and time variable when solving parabolic differential equations. This idea was tested on nonlinear evolution equations, namely Fisher's, Burgers-Fisher, Burgers-Huxley and Fitzhugh-Nagumo equations. The quasilinearization technique was used to first linearize the equations. The results obtained show that the method is highly accurate. A noteworthy result is that the high accuracy is obtained using few grid points on both space and time variables. This makes the method more computationally efficient.

For future work, we plan to extend the application of the method to:
- systems of PDEs
- PDEs with higher spatial domains
- hyperbolic PDEs

References

[1] Dlamini P.G., Motsa S.S., Khumalo M., On the comparison between compact finite difference and pseudospectral approaches for solving similarity boundary layer problems, Mathematical Problems in Engineering, 2013, Article ID 746489, 15 pages.
[2] During B., Fournie M., Jungel A., High order compact finite difference schemes for a nonlinear Black-Scholes equation, International Journal of Theoretical and Applied Finance, 2003, 6, 767-789.
[3] Li J. and Visbal M. R., High-order compact schemes for nonlinear dispersive waves, Journal of Scientific Computing, 2006, 26, 1-23.
[4] Sari M., Solution of the porous media equation by a compact finite difference method, Mathematical Problems in Engineering, 2009, Article ID 912541, 13 pages.
[5] Shah A., Yuan L., and Khan A., Upwind compact finite difference scheme for time-accurate solution of the incompressible Navier-Stokes equations, Applied Mathematics and Computation, 2010, 215, 3201-3213.
[6] Lele S. K., Compact finite difference schemes with spectral-like resolution, Journal of Computational Physics, 1992, 103, 16-42.
[7] Zhang P.G., Wang J.P., A predictor-corrector compact finite difference scheme for Burgers' equation, Applied Mathematics and Computation, 2012, 219, 892-898
[8] Bastani M. and Salkuyeh D. K., A highly accurate method to solve Fisher's equation, Journal of Physics, 2012, 78, 335-346.
[9] Sari M. and Gurarslan G., A sixth-order compact finite difference scheme to the numerical solutions of Burgers' equation, Applied Mathematics and Computation, 2009, 208, 475-483.
[10] Sari M. and Gurarslan G., A sixth-order compact finite difference method for the one-dimensional sine-Gordon equation, International Journal for Numerical Methods in Biomedical Engineering, 2011, 27, 1126-1138.
[11] Dlamini P.G., Motsa S.S., Khumalo M., Higher order compact finite difference schemes for unsteady boundary layer flow problems, Advances in Mathematical Physics, 2013, Article ID 941096, 10 pages.
[12] Bellman R. E. and Kalaba R. E., Quasilinearization and Nonlinear Boundary-Value Problems, Elsevier, New York, NY, USA, 1965.
[13] Wazwaz A.M. and Gorguis A., An analytic study of Fisher's equation by using Adomian decomposition method, Applied Mathematics and Computation, 2004, 154, 609-620.
[14] Golbabai A., Javidi M., A spectral domain decomposition approach for the generalized Burger's-Fisher equation, Chaos, Solitons and Fractals, 2009, 39, 385-392.
[15] Hashim I., Noorani M.S.M., Said Al-Hadidi M.R., Solving the generalized Burgers-Huxley equation using the Adomian decomposition method, Math. Comput. Model., 2006, 43, 1404-1411.
[16] Wang X.Y., Zhu Z.S. and Lu Y.K., Solitary wave solutions of the generalised Burgers-Huxley equation, J. Phys. A: Math., 1990, 23, 271-274.
[17] H.Li, Y.Guo, New exact solutions to the FitzHugh-Nagumo equation, Applied Mathematics and Computation, 2006, 180, 524-558.