MODIFIED LOCAL CRANK-NICOLSON METHOD FOR GENERALIZED BURGERS-HUXLEY EQUATION

PENGZHAN HUANG and ABDURISHIT ABDUWALI

Communicated by Vasile Brînzănescu

The Modified Local Crank-Nicolson method is applied to solve generalized Burgers-Huxley equation. New difference scheme that is explicit, conditionally stable, and easy to compute is obtained for the considered equation. Numerical experiment is carried out in support of the given method.

AMS 2010 Subject Classification: 65M06, 65M12.

Key words: modified local Crank-Nicolson method, generalized Burgers-Huxley equation, linearization approach, explicit scheme, stability analysis.

1. INTRODUCTION

Nonlinear partial differential equations arising from many scientific phenomena play a major role in science and engineering. One of the most well-known nonlinear partial differential equation is the Burgers-Huxley equation. This equation is of high importance for describing the interaction between reaction mechanisms, convection effects, and diffusion transports.

The generalized Burgers-Huxley equation is of the form

$$u_t + \alpha u^\delta u_x - \lambda u_{xx} = \beta u(1-u^\delta)(\eta u^\delta - \gamma), \qquad a \le x \le b, \quad t \ge 0, \tag{1}$$

where $\alpha, \beta, \gamma, \eta, \lambda$ and δ are parameters such that $\alpha, \beta, \lambda \ge 0$ and $\delta > 0$.

This equation also includes several known evolution equations depending on the given parameters. The case $\delta = 1$ contains several known evolution equations:

(1) if $\alpha = 0$, then (1) becomes the Huxley equation, sometimes known as the Fitzhugh-Nagumo equation;

(2) by setting $\alpha = 0$, $\eta = 0$ and $\gamma = -1$ in (1), this equation changes to the Fisher equation;

(3) if $\beta = 0$, then (1) becomes the Burgers equation;

(4) the values $\eta = 0$ and $\gamma = -1$ reduce (1) to the Burgers-Fisher equation;

(5) if we set $\alpha = 0$ and $\gamma = -1$ in (1), then (1) becomes the Newell-Whitehead equation.

Approximate solutions of nonlinear differential equations are of importance in physical problems. Thus, the approximate method is of great practical significance, and has drawn close attention of by many people. Various methods have been used to solve this equation, among them are adomian decomposition method (ADM) [6, 7, 10], variational iteration method (VIM) [3], exp-function method [4], spectral collocation method [9], homotopy perturbation method (HPM) [5] and homotopy analysis method (HAM) [15]. In addition, a numerical solution of the generalized Burger's-Huxley equation, based on collocation method using Radial basis functions (RBFs), called Kansa's approach was presented by Khattak [14]. Tomasiello [16] introduced a generalized version of the Iterative Differential Quadrature (IDQ) method for solving this equation for the first time.

Recently, Abduwali introduced the Local Crank-Nicolson method [1] and the Modified Local Crank-Nicolson (MLCN) method [2] for the heat conduction equation and Burgers' equation [8]. The MLCN method transforms the partial differential equation into ordinary differential equations, and uses the Trotter Product formula of the exponential function to approximate the coefficient matrix of these ordinary differential equations. The MLCN solver separates this matrix into some simple matrices, and employs the Crank-Nicolson method to obtain the time updated solution. The MLCN is a stable and explicit difference scheme with simple computation.

In this paper, the MLCN is applied to solve the generalized Burgers-Huxley equation. A new difference scheme for the considered equation is formed, that produces a nonlinear system. This system is in turn solved using a linearization approach; *i.e.*, the system is linearized by allowing the nonlinearities to lag one time step behind, and the resulting system of linear equations is solved using an iterative algorithm. Our work in this paper, is not only successfully used to solve the generalized Burgers-Huxley equation, but can also be used to develop MLCN for solving nonlinear partial differential equations. The rest of this paper is organized as follows. In section 2, the new scheme of generalized Burgers-Huxley equation is constructed. Stability analysis is given for the MLCN scheme of the considered equation. Section 3 is dedicated to the numerical example. Follow by conclusion and references.

2. DESCRIPTION OF THE NEW SCHEME FOR GENERALIZED BURGERS-HUXLEY EQUATION

Consider the generalized Burgers-Huxley equation (1) in $D = \{(x,t) \in \Omega \times (0,T] = (0,1) \times (0,1]\}$ with the initial condition

$$u(x,0) = u_0(x)\ , \quad x \in \Omega = (0,1), \tag{2}$$

and boundary conditions

$$u(0,t) = u(1,t) = 0\ , \quad t \in (0,1], \tag{3}$$

where u_0 is a given function.

For (1), using central difference quotient instead of differential term of space, we obtain the following semi-discrete equation:

$$\frac{dV(t)}{dt} = \frac{1}{2h^2} A V(t). \tag{4}$$

Let $h = 1/M$ be the mesh width in space and set $x_i = ih$ for $i = 1, 2, \ldots, M-1$. Moreover, $V(t)$ in (4) is in the form $V(t) = [v(x_1,t), v(x_2,t), \ldots, v(x_{M-1},t)]^T$. $v(x_i,t)$ is the approximate solution of $u(x_i,t)$, and $v_i := v(x_i,t)$. Let $c_i = 2h^2\beta(1-v_i^\delta)(\eta v_i^\delta - \gamma)$. Then A, a $(M-1)\times(M-1)$ tri-diagonal matrix, can be written

(5)
$$A = \begin{pmatrix} c_1 - 4\lambda & 2\lambda - \alpha h v_1^\delta & & & \\ 2\lambda + \alpha h v_2^\delta & c_2 - 4\lambda & 2\lambda - \alpha h v_2^\delta & & \\ & \ddots & \ddots & \ddots & \\ & & 2\lambda + \alpha h v_{M-2}^\delta & c_{M-2} - 4\lambda & 2\lambda - \alpha h v_{M-2}^\delta \\ & & & 2\lambda + \alpha h v_{M-1}^\delta & c_{M-1} - 4\lambda \end{pmatrix}.$$

The solution of (4) with initial vector $V(0) = [v(x_1,0), v(x_2,0), \ldots, v(x_{M-1},0)]^T$ can be expressed as

$$V(t) = \exp\left(\frac{t}{2h^2} A\right) V(0). \tag{6}$$

Let $\tau = T/N$ be the mesh width in time and set $t_n = n\tau$ for $n = 1, 2, \ldots, N$. Moreover, $V(t_n)$ can be written in the form $V(t_n) = [v(x_1,t_n), v(x_2,t_n), \ldots, v(x_{M-1},t_n)]^T$ and $v_i^n := v(x_i,t_n)$. The nonlinear system (6) can be linearized by allowing the nonlinearities to lag one time step behind. Thus we have

$$V(t_{n+1}) = \exp\left(\frac{\tau}{2h^2} A\right) V(t_n), \tag{7}$$

where $A =$

(8)
$$\begin{pmatrix} c_1^n-4\lambda & 2\lambda-\alpha h(v_1^n)^\delta & & & & \mathbf{0} \\ 2\lambda+\alpha h(v_2^n)^\delta & c_2^n-4\lambda & 2\lambda-\alpha h(v_2^n)^\delta & & & \\ & \ddots & \ddots & \ddots & & \\ & & 2\lambda+\alpha h(v_{M-2}^n)^\delta & c_{M-2}^n-4\lambda & 2\lambda-\alpha h(v_{M-2}^n)^\delta \\ \mathbf{0} & & & 2\lambda+\alpha h(v_{M-1}^n)^\delta & c_{M-1}^n-4\lambda \end{pmatrix},$$

and $c_i^n = 2h^2\beta(1-(v_i^n)^\delta)(\eta(v_i^n)^\delta-\gamma)$.

Consider the Crank-Nicolson scheme for generalized Burgers-Huxley equation

$$\begin{aligned} &\frac{v_i^{n+1}-v_i^n}{\tau} + \alpha(v_i^n)^\delta \left(\frac{v_{i+1}^{n+1}-v_{i-1}^{n+1}}{4h} + \frac{v_{i+1}^n-v_{i-1}^n}{4h}\right) \\ &=\lambda\left(\frac{v_{i+1}^{n+1}-2v_i^{n+1}+v_{i-1}^{n+1}}{2h^2} + \frac{v_{i+1}^n-2v_i^n+v_{i-1}^n}{2h^2}\right) + \frac{c_i^n(v_i^{n+1}+v_i^n)}{4h^2}. \end{aligned} \tag{9}$$

Note that (9) can be rewritten as

$$\begin{aligned} &-\mu(2\lambda-\alpha h(v_i^n)^\delta)v_{i+1}^{n+1} + (1+4\mu\lambda-\mu c_i^n)v_i^{n+1} - \mu(2\lambda+\alpha h(v_i^n)^\delta)v_{i-1}^{n+1} \\ =&\mu(2\lambda-\alpha h(v_i^n)^\delta)v_{i+1}^n + (1-4\mu\lambda+\mu c_i^n)v_i^n + \mu(2\lambda+\alpha h(v_i^n)^\delta)v_{i-1}^n, \end{aligned}$$

where the mesh ratio $\mu = \frac{\tau}{4h^2}$. Its matrix form is

$$V(t_{n+1}) = ((I-\mu A)^{-1}(I+\mu A))V(t_n). \tag{10}$$

From (7) and (10), we obtain the approximation as follows:

$$\exp(\frac{\tau}{2h^2}A) \approx (I-\mu A)^{-1}(I+\mu A). \tag{11}$$

Remark 1. One can also get (11) directly by using (1, 1)-Padé approximation of the matrix exponential function $\exp(\frac{\tau}{2h^2}A)$. The (1, 1)-Padé approximation of $\exp(z)$ is $\frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}$. Taking $z = \frac{\tau}{2h^2}A$, we obtain $\exp(\frac{\tau}{2h^2}A) \approx (1-\frac{\tau}{4h^2}A)^{-1}(1+\frac{\tau}{4h^2}A)$. Moreover, choosing $\mu = \frac{\tau}{4h^2}$, we arrive at (11).

From (11), we know that we should consider the approximation of $\exp(\frac{\tau}{2h^2}A)$ in order to obtain a new numerical method. Thus, we introduce a lemma on Trotter's product formula.

LEMMA 2.1. *Let the matrix A can be denoted as $A = \sum_{i=1}^{M-1} A_i$. Then*

$$\exp(\frac{t}{h^2}A) = \lim_{\sigma\to\infty}\left(\prod_{i=1}^{M-1}\exp(\frac{tA_i}{\sigma h^2})\right)^{\sigma}, \quad \sigma = 1, 2, \ldots \tag{12}$$

for any h, t.

It follows from Lemma 2.1

$$\exp(\frac{\tau}{2h^2}A) \approx \prod_{i=1}^{M-1}\exp(\frac{\tau A_i}{2h^2}), \tag{13}$$

so (13) is a new approximation. And in order to use this approximation, we split matrix A in (7) as follows:

$$A_1 = \begin{pmatrix} c_1^n - 4\lambda & 2\lambda - \alpha h(v_1^n)^\delta & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$A_i = \begin{pmatrix} 0 & & & & & & \mathbf{0} \\ \vdots & & \ddots & & & & \\ 0 & \cdots & 2\lambda + \alpha h(v_i^n)^\delta & c_i^n - 4\lambda & 2\lambda - \alpha h(v_i^n)^\delta & \cdots & 0 \\ & & & & \ddots & & \vdots \\ \mathbf{0} & & & & & & 0 \end{pmatrix}, \tag{14}$$

where $i = 2, 3, \ldots, M-2$,

$$A_{M-1} = \begin{pmatrix} 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 2\lambda + \alpha h(v_{M-1}^n)^\delta & c_{M-1}^n - 4\lambda \end{pmatrix}.$$

For any $i, i = 1, 2, \ldots, M-1$, the following relation holds

$$\exp(\frac{\tau}{2h^2}A_i) \approx (I - \mu A_i)^{-1}(I + \mu A_i). \tag{15}$$

Then applying (13) and (15), we see that

$$\exp(\frac{\tau}{2h^2}A) \approx \prod_{i=1}^{M-1}(I - \mu A_i)^{-1}(I + \mu A_i). \tag{16}$$

Consequently, combination of (7) and (16) yields a new scheme, *i.e.*,

$$V_1(t_{n+1}) = \prod_{i=1}^{M-1} ((I-\mu A_i)^{-1}(I+\mu A_i))V_1(t_n). \tag{17}$$

In order to improve the numerical accuracy of (17), we define $B_i = A_{M-i}$. By substituting B_i into (17), we deduce that

$$V_2(t_{n+1}) = \prod_{i=1}^{M-1} ((I-\mu B_i)^{-1}(I+\mu B_i))V_2(t_n). \tag{18}$$

Next, take the arithmetic mean of (17) and (18), *i.e.*, $V(t_{n+1}) = \frac{1}{2}(V_1(t_{n+1}) + V_2(t_{n+1}))$. Denoting the coefficient matrix of $V(t_n)$ by $C(\mu)$, we have

$$V(t_{n+1}) = C(\mu)V(t_n), \tag{19}$$

where $C(\mu) = \frac{1}{2}\Big(\prod_{i=1}^{M-1}((I-\mu A_i)^{-1}(I+\mu A_i)) + \prod_{i=1}^{M-1}((I-\mu B_i)^{-1}(I+\mu B_i))\Big)$. So, (19) is the wanted new scheme. We refer to the above method as Modified Local Crank-Nicolson (MLCN) method.

Remark 2. By the expansion formula, we have

$$\exp\left(\frac{\tau}{2h^2}A\right) = \sum_{i=0}^{\infty} \frac{1}{i!}\left(\frac{\tau}{2h^2}A\right)^i.$$

The equation on the right hand side of (13) can be rewritten as

$$\begin{aligned}\prod_{i=1}^{M-1} \exp\left(\frac{\tau A_i}{2h^2}\right) &= I + \frac{\tau}{2h^2}A + \left(\frac{\tau}{2h^2}\right)^2 \Big(A_1A_2 + A_1A_3 + \cdots + A_1A_{M-1} \\ &+ A_2A_3 + A_2A_4 + \cdots + A_2A_{M-1} + \cdots + A_{M-2}A_{M-1} \\ &+ \frac{1}{2}(A_1^2 + A_2^2 + \cdots + A_{M-1}^2)\Big) + \cdots .\end{aligned}$$

If we replace A_i by $B_i = A_{M-i}$, then we obtain

$$\begin{aligned}&\prod_{i=1}^{M-1} \exp\left(\frac{\tau B_i}{2h^2}\right) \\ =& I + \frac{\tau}{2h^2}A + \left(\frac{\tau}{2h^2}\right)^2 \Big(A_{M-1}A_{M-2} + A_{M-1}A_{M-3} + \cdots + A_{M-1}A_1 \\ &+ A_{M-2}A_{M-3} + A_{M-2}A_{M-4} + \cdots + A_{M-2}A_1 + \cdots + A_2A_1 \\ &+ \frac{1}{2}(A_{M-1}^2 + A_{M-2}^2 + \cdots + A_1^2)\Big) + \cdots .\end{aligned}$$

By taking the arithmetic mean of above two equations, we have

$$\frac{1}{2}\left(\prod_{i=1}^{M-1}\exp\left(\frac{\tau A_i}{2h^2}\right)+\prod_{i=1}^{M-1}\exp\left(\frac{\tau B_i}{2h^2}\right)\right)=I+\frac{\tau}{2h^2}A+\frac{1}{2!}\left(\frac{\tau}{2h^2}A\right)^2+\cdots.$$

Note that

$$\exp\left(\frac{\tau}{2h^2}A\right)=I+\frac{\tau}{2h^2}A+\frac{1}{2!}\left(\frac{\tau}{2h^2}A\right)^2+\cdots.$$

So, we could find that scheme (19) approximates the solution of (4) with the improved accuracy. In fact, based on taking the arithmetic mean and utilizing the non-commutativity of the matrix multiplication, we can improve the approximation accuracy for $\exp(\frac{\tau}{2h^2}A)$.

The matrix $(I+\mu A_i)$ can be denoted by a simple form:

$$(I+\mu A_i)=\begin{pmatrix} I_{i-2} & & \\ & S_i & \\ & & I_{M-i-2}\end{pmatrix}, i=2,3,\ldots,M-2, \tag{20}$$

where I_i is an $i\times i$ identity matrix. Let $a_i=2\lambda-\alpha h(v_i^n)^\delta, b_i=2\lambda+\alpha h(v_i^n)^\delta$. Then

$$S_i=\begin{pmatrix} 1 & 0 & 0\\ \mu b_i & 1+\mu c_i^n-4\mu\lambda & \mu a_i\\ 0 & 0 & 1\end{pmatrix}.$$

Similar to (20), we have

$$(I-\mu A_i)^{-1}=\begin{pmatrix} I_{i-2} & & \\ & R_i^{-1} & \\ & & I_{M-i-2}\end{pmatrix}, i=2,3,\ldots,M-2, \tag{21}$$

$$R_i^{-1}=\begin{pmatrix} 1 & 0 & 0\\ \frac{\mu b_i}{1-\mu c_i^n+4\mu\lambda} & \frac{1}{1-\mu c_i^n+4\mu\lambda} & \frac{\mu a_i}{1-\mu c_i^n+4\mu\lambda}\\ 0 & 0 & 1\end{pmatrix}.$$

Thus, we obtain an explicit expression of $V(t_{n+1})$. Clearly, (19) is an explicit scheme. Because we split A into some simple matrices as (14), we can obtain the inverse of these matrices exactly, although we have to find the inverse of the matrix in (19). Hence, it avoids solving the linear equations with large coefficient matrix, which is very important in numerical computation.

THEOREM 2.1. *Let matrix A be written as $A=\sum_{i=1}^{M-1}A_i$. Suppose $c_i^n\le 0$ depending on β, η and γ. Then, for the split method expressed by* (14), *the difference scheme* (19) *is stable.*

Proof. Let ν_i be any eigenvalue of matrix A_i, and ρ_i be any eigenvalue of matrix $(I-\mu A_i)^{-1}(I+\mu A_i)$. Because of $\lambda \geq 0$ and $c_i^n \leq 0$, clearly, we have $\nu_i \leq 0$, and $|\rho_i| = |(1+\mu\nu_i)/(1-\mu\nu_i)| \leq 1$, so $\prod_{i=1}^{M-1} |\rho_i| \leq 1$.

Therefore, the absolute value of any eigenvalue of the coefficient matrix $C(\mu)$ of difference scheme (19) is not greater than 1. By the definition of stability, the new difference scheme is stable. □

3. NUMERICAL EXAMPLE

In order to demonstrate the adaptability of the present method, we consider a test example. The piecewise uniform mesh [11, 13] is used to tabulate the maximum error. These are defined as

$$E_\lambda^N = \max_{0\leq i,n\leq M,N} |v^M(x_i,t_n) - v^{2M}(x_i,t_n)|, \quad E^M = \max_\lambda E_\lambda^M,$$

where superscript indicates the number of mesh points used in the spatial direction.

Example 1. Taking $\alpha = 1$, $\beta = \frac{2}{3}$, $\eta = 1$, $\delta = 1$ and $\gamma = 1$, we obtain a Burgers-Huxley equation as follows

$$\begin{cases} u_t + uu_x - \lambda u_{xx} = \frac{2}{3}(1-u)(u-1)u\ , & 0 < x < 1, 0 < t \leq T, \\ u(x,0) = \sin(\pi x), & 0 \leq x \leq 1, \\ u(0,t) = u(1,t) = 0, & 0 < t \leq T. \end{cases}$$

From the above given parameters, it is clear that $c_i^n = \frac{4}{3}h^2(1-v_i^n)(v_i^n-1)$ is not greater than 0. Hence, the difference scheme for this given equation is stable by Theorem 2.1. The maximum absolute errors are given in Table 1 by using the proposed method on a fitted piecewise uniform mesh, *i.e.* a Shishkin mesh for different values of λ and M at $T = 0.1$ with $\tau = 0.001$. In Table 2, the maximum absolute errors for different values of λ and M at $T = 0.1$ with $\tau = 0.004$ are shown. We have seen from Tables 1 and 2 that the results of the MLCN scheme (19) are good.

TABLE 1

The maximum absolute errors at $T = 0.1$ for $\tau = 0.001$

M	λ					E^M
	2^{-2}	2^{-4}	2^{-10}	2^{-20}	2^{-30}	
8	0.27563	0.31077	0.37511	0.37853	0.37853	0.37853
16	0.05937	0.09198	0.10122	0.10019	0.10019	0.10122
32	0.01516	0.02225	0.02755	0.02746	0.02746	0.02755

TABLE 2

The maximum absolute errors at $T = 0.1$ for $\tau = 0.004$

M	λ					E^M
	2^{-2}	2^{-4}	2^{-10}	2^{-20}	2^{-30}	
8	0.08518	0.17972	0.31137	0.31587	0.31587	0.31587
16	0.01609	0.03033	0.01706	0.01709	0.01709	0.03033
32	0.01167	0.00606	0.00486	0.00736	0.00736	0.01167

The effect of λ can be seen from Fig. 1, and the profiles of the numerical solutions for the fixed value of T and for different values of λ are given. This figure shows the profiles for $T = 0.1$ and for different values of $\lambda = 2^{-1},\ 2^{-5},\ \ 2^{-10},\ 2^{-20}$. The final time on the layer behavior can be seen from Figs. 2 and 3. In these two figures, the profiles of the numerical solutions for the fixed value of λ and for different values of T are given. The former figure shows the profiles for $\lambda = 2^{-2}$ and for different values of $T = 0.1,\ 0.5,\ 1.0$. The latter figure shows the profiles for $\lambda = 2^{-7}$ and for different values of $T = 0.1,\ 0.5,\ 1.0$.

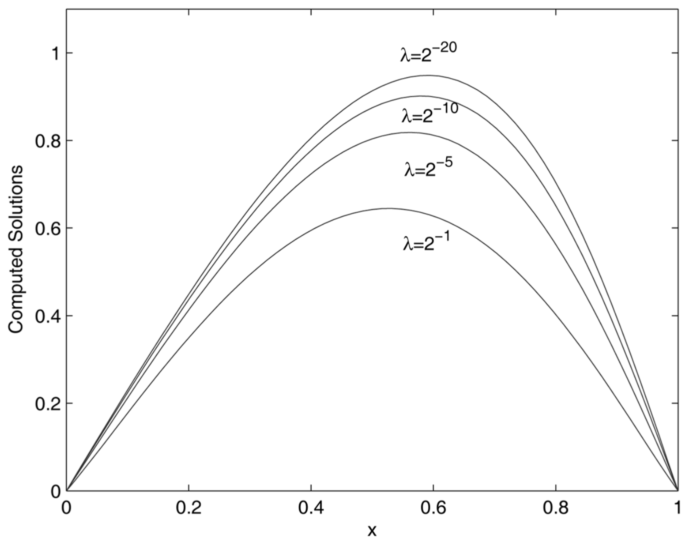

Fig. 1 – Numerical solutions at $T = 0.1$ for $\lambda = 2^{-1},\ 2^{-5},\ \ 2^{-10},\ 2^{-20}$ and $\tau = 0.004$.

It is well-known that one of the severe difficulties in approximating the solution of the given problem is the presence of the parameter λ [12]. A shock of the solution may occur after some time, even if the initial data is smooth.

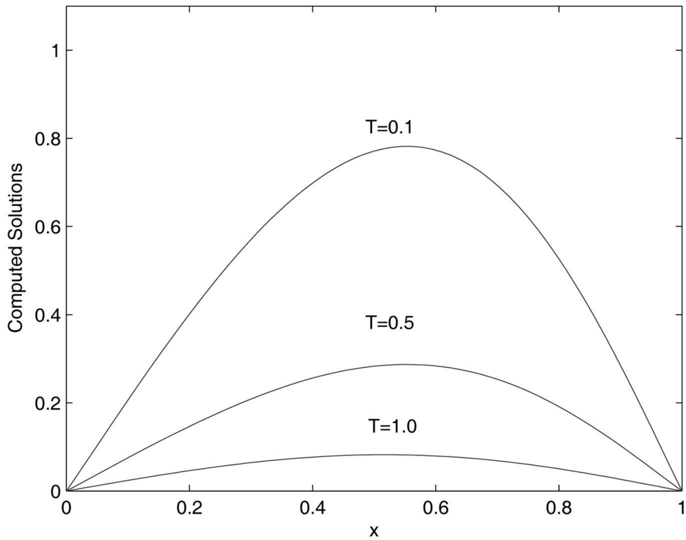

Fig. 2 – Numerical solutions at $T = 0.1,\ 0.5,\ 1.0$ for $\lambda = 2^{-2}$, $M = 64$ and $\tau = 0.004$.

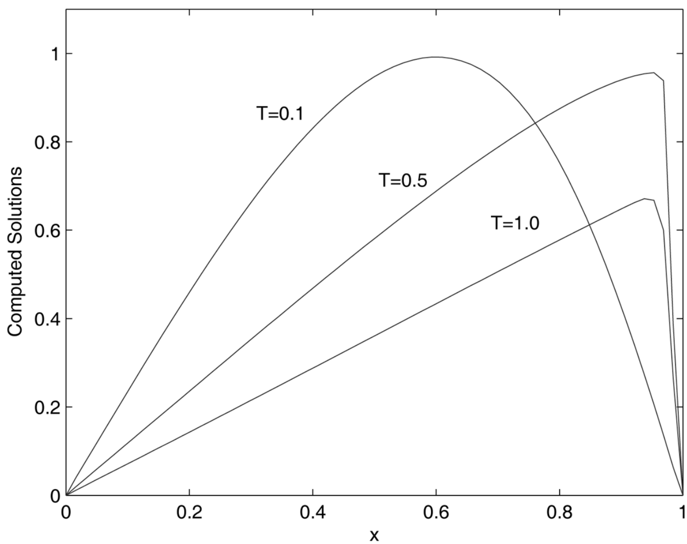

Fig. 3 – Numerical solutions at $T = 0.1,\ 0.5,\ 1.0$ for $\lambda = 2^{-7}$, $M = 64$ and $\tau = 0.004$.

Hence, a robust and accurate numerical algorithm should be able to capture the shock and the numerical solution should exhibit the correct physical behavior. From Figs. 2 and 3, the propagation front is steeper for smaller value of λ, and the maximum point of the solution tilts towards the right end point. It is clear that the method presented in this paper faithfully mimics the dynamics of the corresponding differential equation.

4. CONCLUSION

The MLCN method for generalized Burgers-Huxley equation has been presented. It is shown that the method is a stable explicit difference scheme, if $c_i^n \leq 0$. In support of the given method, the test example has been considered and implemented successfully. The advantage of the proposed method is that it is very easy to use it to solve generalized Burgers-Huxley equation and the numerical results exhibit the correct physical behavior. Therefore, it is suggested using the MLCN to get the numerical solution of the generalized Burgers-Huxley equation effectively.

Acknowledgments. The authors would like to thank the editor and referees for their valuable comments and suggestions which helped us to improve the results of this paper. This research was supported by the NSF of China (Grant No. 11401511) and the Scientific Research Program of the Higher Education Institution of Xinjiang (Grant No. XJEDU2014S002).

REFERENCES

[1] A. Abduwali, M. Sakakihara and H. Niki, *A local Crank-Nicolson method for solving the heat equation.* Hiroshima Math. J. **24** (1994), 1–13.

[2] A. Abduwali, *A corrector local C-N method for the two-dimensional heat equation.* Math. Numer. Sin. **19** (1997), 267–276 (in Chinese).

[3] B. Batiha, M.S.M. Noorani and I. Hashim, *Application of variational iteration method to the generalized Burgers-Huxley equation.* Chaos Soliton Fract. **36** (2008), 660–663.

[4] H. Gao and R.X. Zhao, *New exact solutions to the generalized Burgers-Huxley equation.* Appl. Math. Comput. **217** (2010), 1598–1603.

[5] S.H. Hashemi, H.R.M Daniali and D.D. Ganji, *Numerical simulation of the generalized Huxley equation by He's homotopy perturbation method.* Appl. Math. Comput. **192** (2007), 157–161.

[6] I. Hashim, M.S.M. Noorani and B. Batiha, *A note on the Adomian decomposition method for the generalized Huxley equation.* Appl. Math. Comput. **181** (2006), 1439–1445.

[7] I. Hashim, M.S.M. Noorani and M.R. Said Al-Hadidi, *Solving the generalized Burgers-Huxley equation using the Adomian decomposition method.* Math. Comput. Model. **43** (2006), 1404–1411.

[8] P.Z. Huang and A. Abduwali, *The Modified Local Crank-Nicolson method for one- and two-dimensional Burgers' equations.* Comput. Math. Appl. **59** (2010), 2452–2463.

[9] M. Javidi, *A numerical solution of the generalized Burger's-Huxley equation by spectral collocation method.* Appl. Math. Comput. **178** (2006), 338–344.

[10] M. Javidi and A. Golbabai, *A new domain decomposition algorithm for generalized Burger's-Huxley equation based on Chebyshev polynomials and preconditioning.* Chaos Soliton Fract. **39** (2009), 849–857.

[11] M.K. Kadalbajoo and A. Awasthi, *Crank-Nicolson finite difference method based on a midpoint upwind scheme on a non-uniform mesh for time-dependent singularly perturbed convection-diffusion equations.* Int. J. Comput. Math. **85** (2008), 771–790.

[12] M.K. Kadalbajoo and K.C. Patidar, *Singularly perturbed problems in partial differential equations: a survey.* Appl. Math. Comput. **134** (2003), 371–429.

[13] A. Kaushik and M.D. Sharma, *A uniformly convergent numerical method on non-uniform mesh for singularly perturbed unsteady Burger-Huxley equation.* Appl. Math. Comput. **195** (2008), 688–706.

[14] A.J. Khattak, *A computational meshless method for the generalized Burger's-Huxley equation.* Appl. Math. Model. **33** (2009), 3718–3729.

[15] A. Molabahrami and F. Khani, *The homotopy analysis method to solve the Burgers-Huxley equation.* Nonlinear Anal. Real World Appl. **10** (2009), 589–600.

[16] S. Tomasiello, *Numerical solutions of the Burgers-Huxley equation by the IDQ Method.* Int. J. Comput. Math. **87** (2010), 129–140.

Received 29 March 2014

Xinjiang University,
College of Mathematics and System Sciences,
Urumqi 830046,
P.R. China
hpzh007@yahoo.com

Xinjiang University,
College of Mathematics and System Sciences,
Urumqi 830046,
P.R. China
rashit@xju.edu.cn