Applied Mathematics, 2020, 11, 137-145
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Numerical Discrete-Domain Integral Formulations for Generalized Burger-Fisher Equation

Okey Oseloka Onyejekwe, Beruk Minale, Fikru Habtamu, Tesfaye Amha, Getenet Tamiru, Bethelhem Mengistu, Yohaness Demiss, Nahom Alemseged, Computational Science and Dynamics Systems Group

Computational Science Program, Addis Ababa University, Arat-Kilo Campus, Addis Ababa, Ethiopia
Email: okuzaks@yahoo.com

How to cite this paper: Onyejekwe, O.O., Minale, B., Habtamu, F., Amha, T., Tamiru, G., Mengistu, B., Demiss, Y., Alemseged, N. and Computational Science and Dynamics Systems Group (2020) Numerical Discrete-Domain Integral Formulations for Generalized Burger-Fisher Equation. *Applied Mathematics*, **11**, 137-145.
https://doi.org/10.4236/am.2020.113012

Received: February 5, 2019
Accepted: March 2, 2020
Published: March 5, 2020

Abstract

In this study we use a boundary integral element-based numerical technique to solve the generalized Burger-Fisher equation. The essential feature of this method is the fundamental integral representation of the solution inside the problem domain by means of both the boundary and domain values. The occurrences of domain integrals within the problem arising from nonlinearity as well as the temporal derivative are not avoided or transferred to the boundary. However, unlike the classical boundary element approach, they are resolved within a finite-element-type discrete domain. The utility and correctness of this formulation are proved by comparing the results obtained herein with closed form solutions.

Keywords

Burger-Fisher Equation, Domain Integrals, Boundary Element, Finite-Element-Type Discrete Domain, Boundary Integral, Numerical, Nonlinearity

1. Introduction

Burger-Fisher equation describes the interaction between linear diffusion, nonlinear convection and nonlinear logistic-reaction and has wide applications in several areas of engineering and applied science. Several numerical methods have been proposed for its solution. A few of them include cubic B-spline interpolation quasi-interpolation technique (Zhu and Wang [1]), variational iteration method (Abdou and Soliman [2], Zhao *et al.* [3]) non-standard finite difference scheme (Mickens and Gumel [4]), spectral collocation approach (Javidi [5]), homotopic mapping methods (MO [6]), direct discontinuous Galerkin method (Zhang and Zhang [7]) and other methods (Wazwaz and Gorgius [8], Wazwaz [9]), Wazwaz [10]).

A major thrust of this paper is to further display the utility of one of the emerging families of domain-based integro-differential numerical techniques (Onyejekwe [11] [12], and [13]) for the solution of transient nonlinear partial differential equations. The approach is based on discretizing the governing differential equation into a local or element based system of equations which finally yield a sparse coefficient matrix thereby making it competitive with FEM for problems of similar rigor.

For the problem set up, all the nodal points are inter-connected just like in the finite element method (FEM). In case of a linear element, only two nodal points are required for discretization in each sub-domain. Both the dependent variable and its spatial derivative constitute the two degrees of freedom at each node. Within each element, the problem constants and parameters represent those of the element under consideration. Thus, the method is completely domain-discretized and element based. It should however be realized that the implementation of such an idea, requires a deeper numerical insight into the properties of the corresponding nonlinear integral operators and their application to elements resulting from a discretized problem domain. The paper is aimed at further strengthening this approach by applying these ideas to a highly nonlinear, transient one-dimensional problem which more often than not, poses a considerable numerical challenge to classical boundary element method (BEM) application.

In this paper, we consider the generalized Burger-Fisher equation described as:

$$D\frac{\partial^2\theta}{\partial x^2}=\frac{\partial\theta}{\partial t}+\alpha\theta^d\frac{\partial\theta}{\partial x}+\mu\theta\left(\theta^d-1\right),\ 0\le x\le 1,\ t>0 \tag{1}$$

where, D is the diffusion coefficient, and α,μ and d are the equation parameters. The exact solution is

$$\theta(x,t)=\left[\frac{1}{2}+\frac{1}{2}\tanh\left[\pi_1\left(x-\pi_2 t\right)\right]\right]^{1/d},\ t\ge 0 \tag{2}$$

with initial and boundary conditions:

$$\theta(x,0)=\left[\frac{1}{2}+\frac{1}{2}\tanh\left(\pi_1 x\right)\right]^{1/d} \tag{3}$$

$$\theta(0,t)=\left[\frac{1}{2}+\frac{1}{2}\tanh\left(\pi_1\pi_2 t\right)\right]^{1/d},\ t\ge 0 \tag{4}$$

$$\theta(1,t)=\left[\frac{1}{2}+\frac{1}{2}\tanh\left[\pi_1\left(1-\pi_2 t\right)\right]\right]^{1/d},\ t\ge 0 \tag{5}$$

where, $\pi_1=\frac{-\alpha d}{2D(1+d)},\pi_2=\frac{\alpha}{(1+d)}+\frac{\left(D*\mu(1+d)\right)}{\alpha}$

2. Mathematical Formulation

The numerical approach adopted herein is initiated by implementing the integral replication of Equation (1). This is achieved by recasting the governing differential equation into its Poisson form and applying the Green's second identity to obtain:

$$\int_{x_x}^{x_L}\left[\theta\delta\left(x-x_i\right)-\frac{G}{D}\left(\alpha\theta^d\frac{\partial\theta}{\partial x}+\frac{\partial\theta}{\partial t}+\mu\theta\left(\theta^d-1\right)\right)\right]\mathrm{d}x=\left[\theta\frac{\mathrm{d}G}{\mathrm{d}x}-G\frac{\mathrm{d}\theta}{\mathrm{d}x}\right] \tag{6}$$

where, $G\left(x,x_i\right)=\left[\frac{\left|x-x_i\right|+K}{2}\right]$ represents the free-space Green's function and K is an arbitrary constant. We may mention in passing that when the Poisson equation is solved using the boundary integral technique, a domain integral exists. The way and manner of handling the domain integral constitutes the theme of several boundary element method (BEM) solution techniques. Equation (6) is solved on each element of the problem domain and then assembled in a finite-element sense.

Major aspects of the scheme are as follows: The formulation is motivated by two important characteristics with some inherent properties of BEM formulation. The first is the "elementization" or discretization of the problem domain. This finite-element-like formulation guarantees local compact support. This attributes guarantees that a slim and banded coefficient matrix which can be stored efficiently even for large field problems and is easily amenable to numerical solution. The second characteristic is BEM's straightforward formulation and accuracy

The problem domain is discretized into elements and within each element, the dependent variables and their functions are approximated by interpolating functions in space to guarantee the continuity of the dependent variable and its derivative throughout the discrete elements.

$$\begin{gathered}\frac{\partial\theta\left(x,t\right)}{\partial x}=\varphi\left(x,t\right)\approx\Omega_1\varphi_1+\Omega_2\varphi_2,\ \theta\left(x,t\right)\approx\Omega_1\theta_1+\Omega_2\theta_2\\ \frac{\partial\theta\left(x,t\right)}{\partial t}\approx\Omega_1\left\{\frac{\partial\theta\left(x,t\right)}{\partial t}\right\}_1+\Omega_2\left\{\frac{\partial\theta\left(x,t\right)}{\partial t}\right\}_2\end{gathered} \tag{7}$$

Substitute Equation (7) into Equation (6) to give:

$$\begin{aligned}&-\lambda\theta\left(x_i,t\right)+\left[\theta\frac{\mathrm{d}G}{\mathrm{d}x}-G\frac{\partial\theta}{\partial x}\right]_{x_0}^{x_L}+\int_{x_0}^{x_L}\frac{G}{D}\left(\alpha\left\{\Omega_1\theta_1^d+\Omega_2\theta_2^d\right\}\left\{\Omega_1\varphi_1+\Omega_2\varphi_2\right\}\right.\\&\left.+\left\{\Omega_1\frac{\mathrm{d}\theta_1}{\mathrm{d}t}+\Omega_2\frac{\mathrm{d}\theta_2}{\mathrm{d}t}\right\}+\left\{\Omega_1\left(\mu\theta_1\left[\theta_1^d-1\right]\right)+\Omega_2\left(\mu\theta_2\left[\theta_2^d-1\right]\right)\right\}\right)\mathrm{d}x=0\end{aligned} \tag{8}$$

In order to make Equation (8) amenable to element-by-element computation, we covert the spatial linear interpolating functions into local coordinate that has its origin at node 1 of any generic element. viz: $\Omega_1\left(\xi\right)=1-\xi$, $\Omega_2\left(\xi\right)=\xi$, $\xi=\left(x-x_i\right)/l$, $0\le\xi\le1$. The spatial derivative of the free space Green function is $\mathrm{d}G\left(x,x_i\right)/\mathrm{d}x=0.5\left[H\left(x-x_i\right)-H\left(x_i-x\right)\right]$, where *H*(), is the Heaviside function. A general representation of the numerical solution within the problem domain in discrete form is obtained by representing Equation (8) on each of the elemental nodes.

For node 1 Equation (8) becomes:

$$\begin{aligned}&\frac{1}{2}\left\{\upsilon\left(-\theta_1+\theta_2+\varphi_1-\left(l+l_m\right)\varphi_2\right)\right\}\\&+\frac{1}{2}\left\{l\int_0^1\left(l\xi+1\right)\left[a\left(\Omega_1\theta_1^d+\Omega_2\theta_2^d\right)\left(\Omega_1\varphi_1+\Omega_2\varphi_2\right)\right.\right.\\&\left.\left.+\left(\Omega_1\frac{\mathrm{d}\theta_1}{\mathrm{d}t}+\Omega_2\frac{\mathrm{d}\theta_2}{\mathrm{d}t}\right)+\left(\Omega_1\left(\mu\theta_1\left[\theta_1^d-1\right]\right)+\Omega_2\left(\mu\theta_2\left[\theta_2^d-1\right]\right)\right)\right]\mathrm{d}\xi\right\}=0\end{aligned}\tag{9}$$

Similarly at node 2 we obtain

$$\begin{aligned}&\frac{1}{2}\left\{\left(\theta_1-\theta_2+\left(l+l_m\right)\varphi_1-\varphi_2\right)\right\}\\&+\frac{1}{2}\left\{l\int_0^1\left[l\left(1-\xi\right)+1\right]\left[a\left(\Omega_1\theta_1^d+\Omega_2\theta_2^d\right)\left(\Omega_1\varphi_1+\Omega_2\varphi_2\right)\right.\right.\\&\left.\left.+\left(\Omega_1\frac{\mathrm{d}\theta_1}{\mathrm{d}t}+\Omega_2\frac{\mathrm{d}\theta_2}{\mathrm{d}t}\right)+\left(\Omega_1\left(\mu\theta_1\left[\theta_1^d-1\right]\right)+\Omega_2\left(\mu\theta_2\left[\theta_2^d-1\right]\right)\right)\right]\mathrm{d}\xi\right\}=0\end{aligned}\tag{10}$$

Equation (9) and Equation (10) are put in compact matrix form to yield:

$$\left(R_{ij}\theta_j+L_{ij}\varphi_j\right)+\alpha\Lambda_{ijk}\theta_j\varphi_k+\frac{T_{ij}}{D}\left(\frac{\mathrm{d}\theta_j}{\mathrm{d}t}+\mu\phi_j\left\{\phi_j^d-1\right\}\right)=0\ \text{ where }\ i,j,k=1,2\tag{11}$$

Equation (11) is system of first order nonlinear differential equations in time which is solved for θ and $\mathrm{d}\theta/\mathrm{d}x$ at the nodes. The temporal term in Equation (11) still needs to be discretized. There are several ways of achieving this but for the time being, we use a two-level time scheme. This approximates the time scheme, *i.e.*, $t=t_{m+\alpha}=t_m+z\Delta t$. As a result, Equation (11) becomes:

$$\begin{aligned}&z\left[\left(R_{ij}\theta_j^{(m+1)}+L_{ij}\varphi_j^{(m+1)}\right)+aV_{ijl}\theta_j^{(m+1)}\varphi_j^{(m+1)}\right]\\&+(1-z)\left[\left(R_{ij}\theta_j^{(m)}+L_{ij}\varphi_j^{(m)}\right)+aV_{ijl}\theta_j^{(m)}\varphi_j^{(m)}\right]\\&+T_{ij}\left(\frac{1}{\Delta t}\left\{\theta_j^{(m+1)}-\theta_j^{(m)}\right\}+(z)\mu\theta_j^{(m+1)}\left\{\theta_j^{d(m+1)}-1\right\}+(1-z)\mu\theta_j^{(m)}\left\{\theta_j^{d(m)}-1\right\}\right)\\&\equiv s_i=0\end{aligned}\tag{12}$$

where z is the weighting factor and subscripts m and $m+1$ denote the previous and current time levels and the time step is defined as $\Delta t=t_{m+1}-t_m$. We elect to use the Newton Raphson's technique to resolve the nonlinearity in Equation (12). The computational procedure starts with a known estimate (guessed estimate) of an unknown quantity say:

$\left\{\theta_j^{(m+1)},\varphi_j^{(m+1)}\right\}^{\mathrm{T}}$, and update it to $\left\{\theta_j^{(m+1,k)}+\Delta\theta_j^{(m+1,k+1)},\varphi_j^{(m+1,k)}+\Delta\varphi_j^{(m+1,k+1)}\right\}^{\mathrm{T}}$, eventually, the increment is accounted for by the solution of the matrix equation:

$$J_{ij}^{(m+1,k)}\left\{\begin{matrix}\Delta\theta_j^{(m+1,k+1)}\\\Delta\varphi_j^{(m+1,k+1)}\end{matrix}\right\}=-s_i^{(m+1,k)}\tag{13}$$

where $J_{ij}^{(m+1,k)}$ is the coefficient or the Jacobian matrix. The key item here is the determination of the Jacobian matrix. The solution of Equation (13) yields the increments used in subsequent iterations.

The Jacobian for Equation (12) is:

$$\begin{aligned}\frac{\partial s_i}{\partial \theta_j^{(m+1)}} &= \left[\alpha\left\{R_{ij} + aV_{ijl}\varphi_j^{(m+1)} + (2d)\mu\left(2\theta^{(2d-1)} - 1\right)\right\} + \frac{T_{il}}{\Delta t}\right],\\ \frac{\partial s_i}{\partial \upsilon_j^{(m+1)}} &= \alpha\left[L_{ij} + aV_{ijl}\theta_j^{(m+1)}\right], \quad \text{where } i, j, l = 1, 2\end{aligned} \tag{14}$$

The main advantage of these systems is that they have a direct construction, in addition they are local, in the sense that the dependent variables depend only on the values in the neighborhood of any specific node under consideration.

3. Numerical Experiments and Discussion

In order to test the accuracy and utility of the current formulation, numerical solutions of some test problems are compared with analytical results for different problem parameters. The following parameters were found to provide stable and accurate results for all the problems tested herein. $\Delta t = 0.001$, $d = 1$, $N(\text{total grid points}) = 21$. Relative and absolute errors $abs\left(\left(\theta_{num} - \theta_{exact}\right)/\theta_{num}\right)$, $abs\left(\theta_{num} - \theta_{exact}\right)$ are used to compute the error magnitude.

Table 1 displays the characteristics of the scalar profile for equal weighting of the diffusion and nonlinear inertia terms. The ability of the integral formulation to deal with such cases is demonstrated by the magnitudes of the absolute and relative errors. It can be observed that at each computational point, the error magnitudes display a slight increase as time progresses. This is probably due to the influence of the reaction term.

On the other hand, when the diffusion term is dominant, as shown in Table 2, the damping effect on the scalar profile is shown by the smaller magnitude of the relative and absolute errors.

Table 1. Comparison of numerical and analytic results for $0 \le x \le 1, \alpha = 1.0$ and $\mu = 1.0, D = 1.0$.

x--coordinate	Time	Numerical	Analytical	Abs. Error	Rel. Error
0.1	0.001	0.4780415	0.4878149	9.7734034e−03	2.044676e−02
	0.005	0.4787573	0.4890642	1.0306984e−02	2.1528622e−02
	0.01	0.4858720	0.4906261	4.7540963e−03	4.7846678e−03
0.5	0.001	0.4380772	0.4381312	5.3942204e−05	1.2313406e−04
	0.005	0.4391986	0.4393624	1.6385317e−04	3.7307307e−04
	0.01	0.4405147	0.4409025	3.8778782e−04	8.8030612e−04
0.8	0.001	0.4013811	0.4016127	2.3162365e−04	5.7706668e−04
	0.005	0.3992964	0.4028149	3.5184920e−03	8.8117300e−03
	0.01	0.4008402	0.4043193	3.4790635e−03	8.6794281e−03

Table 3 and Table 4 are for cases where the diffusion term again plays a more prominent role that than the nonlinear inertia term. For both cases there is a damping effect extends to the negative nonlinear reaction term as well. The error terms for both cases are significantly less than the two previous cases.

The ability of the formulation to handle relatively steep gradients as the value of the reaction term is significantly increased is shown in Table 5. Convergent results are noticeable as can be seen by the homoclinic decrease of error magnitudes as time increases.

For Table 6 and Table 7, the diffusion coefficient was set at 0.5; $\alpha = 1.0$ in Table 6. This results in the dominance of the inertia term over diffusion and the equation adopts a more hyperbolic form. The reverse is the case in Table 7 where $\alpha = 0.001$. The previous case results in a more computational challenge, because of the overarching influence of the nonlinear inertia term. This effect is displayed by the magnitudes of the errors displayed in Table 6 and Table 7. As can be seen greater errors are displayed in Table 6 than the Table 7. The influence of the reaction terms remain the same for both cases.

Table 2. Comparison of numerical and analytic results for $0 \le x \le 1, \alpha = 1.0$ and $\mu = 0.0, D = 1.0$.

x--coordinate	Time	Numerical	Analytical	Abs. Error	Rel. Error
0.1	0.001	0.4777927	0.4875651	9.7723305e−03	2.0453075e−02
	0.005	0.4775103	0.4878149	1.0304570e−02	2.1579785e−02
	0.01	0.4833752	0.4881272	4.7520101e−03	9.8308930e−03
0.5	0.001	0.4378311	0.4378850	5.3912401e−05	1.231351e−02
	0.005	0.4379676	0.4381312	1.6358495e−04	2.7350925e−04
	0.01	0.4380523	0.4384389	3.8659573e−04	8.825332e−04
0.8	0.001	0.4011410	0.4013724	2.3144484e−04	5.7696633e−04
	0.005	0.3980983	0.4016127	3.5144389e−04	8.8280691e−04
	0.01	0.3984419	0.4019131	3.4712255e−03	8.7119932e−03

Table 3. Comparison of numerical and analytic results for $0 \le x \le 1, \alpha = 0.1$ and $\mu = -0.0025, D = 1.0$.

x--coordinate	time	Numerical	Analytical	Abs. Error	Rel. Error
0.1	0.001	0.4892023	0.4987500	0.9547681e−03	3.1482705e−03
	0.005	0.4886842	0.4987500	1.0065764e−02	2.0597685e−02
	0.01	0.4942080	0.4987500	4.5419931e−03	9.1904486e−03
0.5	0.001	0.4937497	0.4937503	6.2584877e−07	1.2675425e−06
	0.005	0.4937465	0.4937503	3.7848949e−03	7.6656834e−06
	0.01	0.4936377	0.4937503	1.1262298e−04	2.2814906e−04
0.8	0.001	0.4897544	0.4900013	2.4694204e−04	5.0421612e−04
	0.005	0.4855816	0.4900013	4.4197142e−03	9.1018975e−03
	0.01	0.4857789	0.4900013	4.2223930e−03	8.6920056e−03

Table 4. Comparison of numerical and analytic results for $0 \le x \le 1, \alpha = 0.001$ and $\mu = -0.0025, D = 1.0$.

x--coordinate	time	Numerical	Analytical	Abs. Error	Rel. Error
0.1	0.1	0.4995117	0.4998131	3.0145049e−04	6.0349039e−04
	0.5	0.4995601	0.4995656	5.4836275e−06	1.0976911e−05
	1.0	0.4992565	0.4992563	2.0861626e−07	4.1785391e−07
0.5	0.1	0.4983398	0.4993131	9.7331405e−04	1.9531132e−03
	0.5	0.4990464	0.4990656	1.9192697e−05	3.8458737e−05
	1.0	0.4987555	0.4987563	8.0466270e−07	1.6133412e−06
0.8	0.1	0.4983643	0.4989381	5.738772e−04	1.1514554e−03
	0.5	0.4986793	0.4986906	1.1384487e−05	2.2829277e−05
	1.0	0.4983808	0.4983813	4.7683716e−07	9.5677274e−07

Table 5. Comparison of numerical and analytic results for $0 \le x \le 1, \alpha = 1.0$ and $\mu = 10.0, D = 1.0$.

x--coordinate	time	Numerical	Analytical	Abs. Error	Rel. Error
0.1	0.1	0.7307805	0.7310488	2.6834011e−04	3.6719661e−04
	0.5	0.9933019	0.9933066	4.7087669e−06	6.8407649e−06
	1.0	0.9989547	0.9999546	1.1920929e−07	1.1921469e−07
0.5	0.1	0.7301725	0.7310095	8.3696842e−04	1.1462121e−03
	0.5	0.9932919	0.9933053	1.3470650e−05	1.3561623e−05
	1.0	0.9999547	0.9999546	1.1920929e−07	1.1921469e−07
0.8	0.1	0.7304779	0.7309800	5.0204992e−04	6.8728963e−04
	0.5	0.9932957	0.9933043	8.6426735e−06	8.7010076e−06
	1.0	0.9999546	0.9999546	0.0000000e+00	0.000000e+00

Table 6. Comparison of numerical and analytic results for $-0.5 \le x \le 0.5, \alpha = 1.0$ and $\mu = 0.01, D = 0.5$.

x--coordinate	time	Numerical	Analytical	Abs. Error	Rel. Error
−0.4	0.1	0.6083902	0.6108770	2.4868251e−03	4.0875496e−03
	0.5	0.6545661	0.6581360	3.5698414e−03	5.4537524e−03
	1.0	0.7091637	0.7129992	3.8354993e−03	5.4084826e−03
−0.2	0.1	0.5722150	0.5624226	5.2011013e−03	9.3339924e−03
	0.5	0.6032375	0.6118270	8.5895658e−03	1.4239112e−02
	1.0	0.6609223	0.6704001	9.4777346e−03	1.4340163e−02
0.4	0.1	0.4115621	0.4136249	2.0627677e−03	5.0120442e−03
	0.5	0.4598642	0.4638130	3.9488375e−03	8.5869646e−03
	1.0	0.5228702	0.5274711	4.6008825e−03	8.7992819e−03

Table 7. Comparison of numerical and analytic results $-0.5 \le x \le 0.5, \alpha = 0.001$ and $\mu = 0.01, D = 0.5$.

x--coordinate	time	Numerical	Analytical	Abs. Error	Rel. Error
−0.4	0.1	0.4997850	0.5003500	5.6499243e−04	1.6263075e−03
	0.5	0.5012816	0.5013500	6.8426132e−05	1.3650438e−04
	1.0	0.5025941	0.5026001	5.9604645e−06	1.1859400e−05
−0.2	0.1	0.4989803	0.5003000	1.3196468e−03	2.6446870e−03
	0.5	0.5011207	0.5013126	1.7929077e−04	3.5777959e−04
	1.0	0.5025345	0.5025501	1.5556812e−05	3.0956704e−05
0.4	0.1	0.4995852	0.5001500	5.6484342e−04	1.1306248e−03
	0.5	0.5010815	0.5011501	6.8604994e−05	1.3691376e−04
	1.0	0.5023942	0.5024001	5.9008598e−06	1.1745477e−05

4. Conclusion

In the work reported herein, a boundary-integral domain-discretized formulation with uniform grids and time steps is applied for the solution of the Burger-Fisher equation. The formulation is straightforward and amenable to handling nonlinearity. Encouraging results are obtained for different parameter values with relatively small number of elements. This numerical experience, further justifies the need to amend and adapt BEM to become a more competitive numerical tool.

Acknowledgements

A lot of thanks to the anonymous reviewers who significantly improved the quality of this paper

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Zhu, C.-G. and Wang, R.-H. (2009) Numerical Solution of Burgers' Equation by Cubic B-Spline Quasi-Interpolation. *Applied Mathematics and Computation*, **208**, 260-272. https://doi.org/10.1016/j.amc.2008.11.045

[2] Abdou, M.A. and Soliman, A.A. (2005) Numerical Solutions of Nonlinear Evolution Equations Using Variational Iteration Method. *Journal of Computational and Applied Mathematics*, **207**, 111-120. https://doi.org/10.1016/j.cam.2006.07.016

[3] Zhao, C.-Z, Yu, X.J. and Zhu, J. (2010) Direct Discontinuous Galerkin Method for the Generalized Burger-Fisher Equation. *Chinese Physics B*, **19**, Article ID: 0702203.

[4] Mickens, R.E. and Gumel, A.B. (2002) Construction and Analysis of a Nonstandard Finite Difference Scheme for the Burgers-Fisher Equation. *Journal of Sound and Vibration*, **257**, 791-797. https://doi.org/10.1006/jsvi.2001.4240

[5] Javidi, M. (2006) Spectral Collocation Method for the Solution of Generalized Burger-Fisher Equation. *Applied Mathematics and Computation*, **174**, 45-52. https://doi.org/10.1016/j.amc.2005.04.084

[6] Mo, J.-Q. (2009) A Variational Iteration Solving Method for a Class of Generalized Boussinesq Equations. *Chinese Physics Letters*, **26**, Article ID: 060202. https://doi.org/10.1088/0256-307X/26/6/060202

[7] Zhang, R.-P. and Zhang, L.-W. (2012) Direct Discontinuous Galerkin Method for Generalized Burgers-Fisher Equation. *Chinese Physics B*, **21**, Article ID: 090206. https://doi.org/10.1088/1674-1056/21/9/090206

[8] Wazwaz, A.M. and Gorguis, A. (2004) An Analytic Study of Fishers Equation by Using Adomian Decomposition Method*. Applied Mathematics and Computation*, **154**, 609-620. https://doi.org/10.1016/S0096-3003(03)00738-0

[9] Wazwaz, A.M. (2005) The Tanh Method for Generalized Forms of Nonlinear Heat Conduction and Burgers-Fisher Equation. *Applied Mathematics and Computation*, **169**, 321-338. https://doi.org/10.1016/j.amc.2004.09.054

[10] Wazwaz, A.M. (2008) Analytic Study of Burgers, Fisher, Huxley Equations and Two Combined Forms of These Equations. *Applied Mathematics and Computation*, **195**, 754-761. https://doi.org/10.1016/j.amc.2007.05.020

[11] Onyejekwe, O.O. (2005) An Efficient Green Element Algorithm for Radial Flow. *Applied Mathematics and Computation*, **165**, 635-645. https://doi.org/10.1016/j.amc.2004.04.099

[12] Onyejekwe, O.O. (2006) A Note on Green Element Method Discretization for Poisson Equation in Polar Coordinates. *Applied Mathematics Letters*, **19**, 785-788. https://doi.org/10.1016/j.aml.2005.08.025

[13] Onyejekwe, O.O. (2009) A Modified Boundary Integral Evolution Formulation for the Wave Equation. *Advances in Engineering Software*, **40**, 519-526. https://doi.org/10.1016/j.advengsoft.2008.10.002