



DIFFERENTIAL QUADRATURE SOLUTIONS OF THE GENERALIZED BURGERS–FISHER EQUATION WITH A STRONG STABILITY PRESERVING HIGH-ORDER TIME INTEGRATION

Murat Sari

Department of Mathematics, Faculty of Art and Science, Pamukkale University, Denizli
20070, Turkey, msari@pau.edu.tr

Abstract- Numerical solutions of the generalized Burgers-Fisher equation are presented based on a polynomial-based differential quadrature method with minimal computational effort. To achieve this, a combination of a polynomial-based differential quadrature method in space and a third-order strong stability preserving Runge-Kutta scheme in time have been used. The proposed technique successfully worked to give reliable results in the form of numerical approximation converging very rapidly. The computed results have been compared with the exact solution to show the required accuracy of the method. The approximate solutions to the nonlinear equations were obtained. The approach is seen to be a very reliable alternative to the rival techniques for realistic problems.

Key Words- Generalized Burgers-Fisher Equation, Differential Quadrature Method, Nonlinear PDE, Strong Stability Preserving Runge-Kutta.

1. INTRODUCTION

Mathematical modelling of many physical systems leads to nonlinear ordinary/partial differential equations in various fields of science. Generalized Burgers–Fisher equation being a nonlinear partial differential equation is of great importance for describing different mechanisms. Fisher [1] first proposed the well-known equation, encountered in various disciplines, as a model for the propagation of a mutant gene with $u(x, t)$ displaying the density of advantage. Later, the equation has been used as a basis for a wide variety of models for different problems. The most general form of the Fisher equation is called the generalized Burgers–Fisher equation.

Many researchers have spent a great deal of effort to compute the solution of the Burgers-Fisher equation using various numerical methods. A numerical simulation and explicit solutions of the generalized Burgers-Fisher equation were presented by Kaya and El-Sayed [2]. A restrictive Padé approximation for the solution of the generalized Burgers-Fisher equation was introduced by Ismail and Rabboh [3]. With the use of Adomian decomposition method, solution of Burgers-Huxley and Burgers-Fisher equations were obtained by Ismail et al. [4]. Recently some various powerful mathematical methods such as tanh function methods [5-7], tanh-coth method [8], variational iteration method [9], factorization method [10] and spectral collocation method [11] have also been used in attempting to solve the equation.

To the best of the author’s knowledge, the differential quadrature method (DQM), where approximations of the spatial derivatives have been based on a polynomial of high degree, has not been implemented for the problems in physical phenomena represented by the generalized Burgers–Fisher equation so far. The DQM is an efficient discretization technique in solving initial and/or boundary value problems accurately.

After the DQM was introduced by Bellman et al. [12], it has been successfully employed in finding the solutions of many problems in applied sciences [13-24]. Recent comparative studies show that the DQM provides highly accurate and efficient solutions of the differential equations taking a noticeably small number of grid points. Due to the aforementioned advantages, the DQM has been predicted by its proponents as a potential alternative to the conventional solution techniques such as the finite difference and finite element methods. However, the DQM are of some problem-dependent difficulties, for example usage in mostly small scale problems, interfacial problems, etc. To overcome these difficulties, some researchers have extended the DQM by combining it with some other techniques or modifying it [25-29].

In the DQM, the weighting coefficients in that weighting sum are determined using test functions. Among the many kinds of test functions, the Lagrange interpolation polynomial is widely used since it has no limitation on the choice of the grid points. This leads to polynomial-based differential quadrature (PDQ) method which is suitable in most problems. For problems with periodic behaviours, Fourier series expansion can be the best approximation giving the Fourier expansion-based differential quadrature (FDQ) method first appeared in [30,31]. The ease for computation of weighting coefficients in explicit formulations [32] for both cases is based on the analysis of function approximation and linear vector space.

Although there has been some research into the numerical approximation of the generalized Burgers–Fisher equation, there has not been any work for finding the numerical solution of the generalized Burgers–Fisher equation with the PDQ method. In this paper, therefore, the use of the PDQ method has been expanded to have the numerical solutions of the generalized Burgers–Fisher equation.

Unlike some previous techniques using various transformations to reduce the equation into more simple equation then solve it, the nonlinear equations are solved without transforming the equation by using the current methodology. This method has also additional advantages over some rival techniques, mainly, ease in use and computationally cost-effectiveness in order to find solutions of the given nonlinear equations. The combination of the PDQ method in space with a third-order strong stability preserving Runge-Kutta (SSP-RK3) scheme in time [33] provides an efficient explicit solution with high accuracy and minimal computational effort for the problems represented by the generalized Burgers-Fisher equation.

The methodology is useful for obtaining numerical approximations of linear/nonlinear differential equations. Furthermore, use of considerably small number of grids and less need of storage make the method advantageous. The computed results obtained by this way have been compared with the exact solution to show the required accuracy of it. Moreover, the present method can be used for solving nonlinear partial differential equations arising in various areas.

2. THE MODEL EQUATION

Many problems encountered in various mechanisms lead to the generalized Burgers-Fisher equation. The generalized Burgers-Fisher equation [4], arising in various fields of science, of the form

$$u_t + \alpha u^\delta u_x - u_{xx} = \beta u(1 - u^\delta), \quad a \leq x \leq b, \quad t \geq 0 \quad (1)$$

with the initial condition

$$u(x, 0) = \left[\frac{1}{2} + \frac{1}{2} \tanh(a_1 x) \right]^{1/\delta}. \tag{2}$$

The exact solution of equation (1) is

$$u(x, t) = \left(\frac{1}{2} + \frac{1}{2} \tanh[a_1(x - a_2 t)] \right)^{1/\delta}, \quad t \geq 0 \tag{3}$$

where

$$a_1 = \frac{-\alpha\delta}{2(\delta+1)}, \quad a_2 = \frac{\alpha}{\delta+1} + \frac{\beta(\delta+1)}{\alpha}, \tag{4}$$

where α, β and δ are parameters. The boundary conditions are taken from the exact solution (3). This study shows that the proposed algorithm is capable of achieving high accuracy for the problems represented by the generalized Burgers-Fisher equation.

3. POLYNOMIAL-BASED DIFFERENTIAL QUADRATURE METHOD

To drive the derivatives of a function in this method, the basis of it is used. It follows that the partial derivative of a function with respect to a space variable can be approximated by a weighted linear combination of function values at some intermediate points in the corresponding variable.

The locations of the sampling points play an important role in the accuracy of the solution of the differential equations. Use of uniform grids can be considered to be the most convenient method. Mostly, the DQM gives more accurate solutions using the so-called Chebyshev-Gauss-Lobatto points. For a domain specified by $a \leq x \leq b$ and discretized by a set of non-uniform grid points, then the coordinate of any point i can be evaluated by [13,19]

$$x_i = a + \frac{1}{2} (1 - \cos(\frac{i-1}{N-1} \pi)) (b - a). \tag{5}$$

The values of function $u(x, t)$ at any time on the above grid points are given as $u(x_i, t)$, $i = 1, 2, \dots, N$. Here N indicates the number of grid points. The differential quadrature discretizations of the first and second order spatial derivatives are given by, respectively:

$$\begin{aligned} u_x(x_i, t) &= \sum_{j=1}^N a_{ij} u(x_j, t), \quad i = 1, 2, \dots, N \\ u_{xx}(x_i, t) &= \sum_{j=1}^N b_{ij} u(x_j, t), \quad i = 1, 2, \dots, N \end{aligned} \tag{6}$$

where a_{ij} and b_{ij} are the weighting coefficients of the first and second order derivatives, respectively [32]. When the weighting coefficients are determined, the nodal values of the governing differential equation at the mesh points are established. The coefficients of the first order derivatives are as follows [32]:

$$\begin{aligned} a_{ij} &= \frac{1}{x_j - x_i} \prod_{k=1, k \neq i, j}^N \frac{x_i - x_k}{x_j - x_k}, \quad i \neq j, \\ a_{ii} &= - \sum_{j=1, j \neq i}^N a_{ij}, \end{aligned} \tag{7}$$

For the weighting coefficients of the second order derivative, the formulae are [32]:

$$b_{ij} = 2a_{ij}(a_{ii} - \frac{1}{x_i - x_j}), \quad i \neq j, \quad (8)$$

$$b_{ii} = - \sum_{j=1, j \neq i}^N b_{ij}.$$

To gain the accurate numerical solution of differential equations, proper implementation of the boundary conditions is also of importance. For prescribing the boundary conditions, equation (1) should only be applied at the interior nodes since the solution at the boundary nodes is known. Thus, equation (1) can be written as follows:

$$\frac{du_i}{dt} = \beta u_i(1 - u_i^\delta) - \alpha u_i^\delta \sum_{k=2}^{N-1} a_{ik} u_k + \sum_{k=2}^{N-1} b_{ik} u_k + s_i, \quad i = 2, 3, \dots, N-1, \quad (9)$$

where

$$s_i = -\alpha u_i^\delta a_{i1} u_1 + b_{i1} u_1 - \alpha u_i^\delta a_{iN} u_N + b_{iN} u_N.$$

After the discretization, equation (9) can be reduced into a set of ordinary differential equations in time. So

$$\frac{du_i}{dt} = Lu_i \quad (10)$$

where L denotes a spatial nonlinear differential operator. Using the SSP-RK3 scheme, equation (10) is integrated from time t_0 to $t_0 + \Delta t$ through the following operations [33]

$$u_i^{(1)} = u_i^m + \Delta t Lu_i^m,$$

$$u_i^{(2)} = \frac{3}{4} u_i^m + \frac{1}{4} u_i^{(1)} + \frac{1}{4} \Delta t Lu_i^{(1)},$$

$$u_i^{m+1} = \frac{1}{3} u_i^m + \frac{2}{3} u_i^{(2)} + \frac{2}{3} \Delta t Lu_i^{(2)}.$$

In this methodology, each spatial derivative on the right hand side of equation (10) was computed with the use of the PDQ method and then the semi-discrete equation (10) was integrated using the low storage scheme.

4. NUMERICAL EXPERIMENTS

To find out whether the proposed algorithm gives rise to accurate solutions, the PDQ solutions are evaluated for some examples of the generalized Burgers–Fisher equation. To verify the efficiency and the versatility of the method for the current problem, in comparison with the exact solution, absolute and relative errors for various values of α , β and δ is reported in the following examples.

Let us take the generalized Burgers-Fisher equation in the form (1) with the initial condition (2) and boundary conditions taken from the exact solution. The computed results are compared with the analytical solutions. The numerical computations were performed using non-uniform grids. All computations were carried out using some codes produced in Visual Basic 6.0. The parameters N and Δt are taken to be 16 and 0.0001 (except in Table 4), respectively in the following examples. The absolute errors and the computed solutions for some values of the parameters δ , α and β are shown in Tables 1-7. In the same tables, relative errors are also presented. As

various problems of science were modelled by nonlinear partial differential equations and since therefore the generalized Burgers–Fisher equation is of high importance, various values of δ have been considered throughout the examples. For the computational work, examples 1-2 given by Ismail et al.[4] and example 3 given by [2] are selected. The parameters a, b in the boundary conditions are taken to be 0, 100 and -1, 1 for example 4 and example 6, respectively. The parameter values in the boundary conditions are chosen as 0, 1 for the rest of examples.

Table 1. The results, absolute and relative errors for $\delta = 1$, $\alpha = 0.001$ and $\beta = 0.001$

x_i	t	Exact	PDQ-SSP-RK3	Absolute Error	Relative Error
x_3	0.001	0.4999948	0.4999948	8.72E-08	1.74E-07
	0.005	0.4999958	0.4999956	2.73E-07	5.46E-07
	0.010	0.4999971	0.4999967	4.19E-07	8.37E-07
x_8	0.001	0.4999443	0.4999442	1.04E-07	2.08E-07
	0.005	0.4999453	0.4999448	5.21E-07	1.04E-06
	0.010	0.4999465	0.4999455	1.04E-06	2.08E-06
x_{13}	0.001	0.4998872	0.4998871	1.03E-07	2.07E-07
	0.005	0.4998882	0.4998878	4.34E-07	8.69E-07
	0.010	0.4998894	0.4998887	7.31E-07	1.46E-06

Table 2. The results, absolute and relative errors for $\delta = 2$, $\alpha = 1$ and $\beta = 1$

x_i	t	Exact	PDQ-SSP-RK3	Absolute Error	Relative Error
x_3	0.001	0.702390	0.702264	1.26E-04	1.80E-04
	0.005	0.703970	0.703573	3.96E-04	5.63E-04
	0.010	0.705940	0.705332	6.08E-04	8.62E-04
x_8	0.001	0.653047	0.652868	1.79E-04	2.74E-04
	0.005	0.654710	0.653817	8.94E-04	1.36E-03
	0.010	0.656787	0.655005	1.78E-03	2.71E-03
x_{13}	0.001	0.595117	0.594912	2.05E-04	3.45E-04
	0.005	0.596825	0.595965	8.60E-04	1.44E-03
	0.010	0.598959	0.597517	1.44E-03	2.41E-03

Table 3. The results, absolute and relative errors for $\alpha = 0.1$, $\beta = -0.0025$ and $\delta = 8$

x_i	t	Exact	PDQ-SSP-RK3	Absolute Error	Relative Error
x_3	0.10	0.9166744	0.9166813	6.89E-06	7.52E-06
	0.20	0.9165652	0.9165741	8.92E-06	9.73E-06
	0.30	0.9164558	0.9164655	9.68E-06	1.06E-05
x_8	0.10	0.9145924	0.9146302	3.79E-05	4.14E-05
	0.20	0.9144814	0.9145341	5.27E-05	5.77E-05
	0.30	0.9143703	0.9144286	5.83E-05	6.38E-05
x_{13}	0.10	0.9122027	0.9122174	1.47E-05	1.62E-05
	0.20	0.9120898	0.9121090	1.93E-05	2.11E-05
	0.30	0.9119768	0.9119978	2.10E-05	2.30E-05

Table 4. The average absolute and average relative errors for various Δt values with $\alpha = -1$, $\beta = 1$, $\delta = 1$, $t = 0.01$, $a = 0$, $b = 100$.

N	$\Delta t = 0.01$		$\Delta t = 0.0001$		$\Delta t = 0.000001$	
	Average Absolute Error	Average Relative Error	Average Absolute Error	Average Relative Error	Average Absolute Error	Average Relative Error
6	3.63E-04	3.57E-04	5.6125E-05	1.91E-05	5.31E-05	2.34E-05
7	3.25E-04	3.46E-04	5.1905E-05	1.69E-05	4.92E-05	2.05E-05
8	2.76E-04	2.95E-04	3.4797E-05	1.85E-06	3.24E-05	4.80E-06
9	2.37E-04	2.92E-04	2.5554E-05	1.45E-05	2.34E-05	1.17E-05
10	2.20E-04	2.80E-04	3.4103E-05	3.26E-05	3.22E-05	3.01E-05

Example 1. In Table 1, absolute and relative errors were shown for various values of x and t with $\delta = 1$, $\alpha = 0.001$ and $\beta = 0.001$. Considering the values of the parameters, a comparison has been made between the computed results and the exact solution.

Example 2. The PDQ-SSP-RK3 results of the generalized Burger’s-Fisher equation for various values of x and t with $\delta = 2$, $\alpha = 1$ and $\beta = 1$ are listed in Table 2. In the table, absolute and relative errors were shown for the parameters. The computed solutions are compared with the exact solution in Table 2.

Example 3. Absolute and relative errors were shown in Table 3 for various values of x and t with $\alpha = 0.1$, $\beta = -0.0025$. The results of the PDQ-SSP-RK3 have been presented in Table 3. Comparison of the current results with the exact solution showed that the presented results are very accurate in this example.

Example 4. The average absolute and average relative errors were presented in Table 4 for various Δt values with $\alpha = -1$, $\beta = 1$, $\delta = 1$, $t = 0.01$, $a = 0$, $b = 100$. Note that nearly the same error is obtained for all discretizations; the error is close to the level of machine accuracy, and this is believed to be a saturation effect (see Table 4).

Table 5. The absolute errors for $\delta = 1$ and different values of α , β , x and t

x	t	$\alpha = 0.01, \beta = 0.01$		$\alpha = 0.0001, \beta = 0.0001$	
		Absolute Error	Relative Error	Absolute Error	Relative Error
x_3	1	2.14E-05	4.27E-05	2.15E-07	4.31E-07
	10	2.04E-05	3.89E-05	2.15E-07	4.30E-07
	50	1.53E-05	2.46E-05	2.15E-07	4.29E-07
x_8	1	1.28E-04	2.56E-04	1.29E-06	2.58E-06
	10	1.22E-04	2.33E-04	1.29E-06	2.57E-06
	50	9.16E-05	1.47E-04	1.28E-06	2.56E-06
x_{13}	1	4.49E-05	8.95E-05	4.50E-07	9.00E-07
	10	4.28E-05	8.16E-05	4.50E-07	9.00E-07
	50	3.20E-05	5.15E-05	4.49E-07	8.95E-07

Table 6. The absolute and relative errors for different values of α , β , δ , x , t with $a = -1, b = 1$

x	t	$\alpha = -0.1, \beta = 0.1,$		$\alpha = -0.01, \beta = 0.01,$		$\alpha = -0.001, \beta = 0.001,$	
		$\delta = 1$		$\delta = 100$		$\delta = 10$	
		Absolute Error	Relative Error	Absolute Error	Relative Error	Absolute Error	Relative Error
x_3	0.01	4.18E-05	8.35E-05	8.25E-06	8.31E-06	7.8E-07	8.37E-07
	0.10	1.47E-04	2.92E-04	2.73E-05	2.75E-05	2.8E-06	2.95E-06
	1.00	2.05E-04	3.89E-04	1.36E-05	1.37E-05	4.0E-06	4.27E-06
x_8	0.01	1.03E-04	2.04E-04	2.05E-05	2.06E-05	1.9E-06	2.08E-06
	0.10	7.83E-04	1.54E-03	1.48E-04	1.49E-04	1.5E-05	1.59E-05
	1.00	1.21E-03	2.28E-03	8.28E-05	8.31E-05	2.4E-05	2.55E-05
x_{13}	0.01	7.14E-05	1.40E-04	1.43E-05	1.44E-05	1.4E-06	1.46E-06
	0.10	2.94E-04	5.72E-04	5.56E-05	5.59E-05	5.6E-06	6.03E-06
	1.00	4.18E-04	7.78E-04	2.84E-05	2.85E-05	8.3E-06	8.91E-06

Example 5. As seen in Table 5, for large time values, behaviour of the solution has also been considered and seen to be very accurate.

Example 6. This example considers examples different boundary conditions ($a = -1, b = 1$). In the example, effects of selection of the parameters δ, α, β have been observed in Tables 6,7 for various values of x and t . The absolute and relative errors have been computed for various values of α, β, δ, x and t . As is the previous examples, the results of the PDQ-SSP-RK3 have been presented in Tables 6,7 and the agreement between the results of the current algorithm and exact solution is seen to be very good.

In the examples; although very few number of grids are used and even when δ is taken to be very high, the PDQ-SSP-RK3 results are seen to be very accurate. The tables showed that a very good approximation to the actual solution of the equations was achieved by using the present method. The current method is a very reliable alternative to the rival methods which face the well-known difficulties.

Table 7. The absolute and relative errors for different values of α , β , δ , x , t with $a = -1, b = 1$

x	t	$\alpha = -0.1, \beta = 0.1,$		$\alpha = -0.01, \beta = 0.01,$		$\alpha = -0.001, \beta = 0.001,$	
		$\delta = 100$		$\delta = 1$		$\delta = 10$	
		Absolute Error	Relative Error	Absolute Error	Relative Error	Absolute Error	Relative Error
x_3	0.01	1.27E-04	1.28E-04	6.89E-06	1.38E-05	1.29E-06	1.38E-06
	0.10	2.45E-04	2.46E-04	2.84E-05	5.68E-05	5.30E-06	5.68E-06
	1.00	7.90E-09	7.90E-09	7.97E-05	1.59E-04	1.48E-05	1.59E-05
x_8	0.01	1.78E-04	1.79E-04	1.04E-05	2.07E-05	1.94E-06	2.08E-06
	0.10	9.26E-04	9.29E-04	1.02E-04	2.04E-04	1.91E-05	2.05E-05
	1.00	5.52E-08	5.52E-08	4.68E-04	9.29E-04	8.71E-05	9.33E-05
x_{13}	0.01	1.51E-04	1.51E-04	9.68E-06	1.93E-05	1.81E-06	1.94E-06
	0.10	3.97E-04	3.98E-04	5.39E-05	1.07E-04	1.01E-05	1.08E-05
	1.00	1.58E-08	1.58E-08	1.65E-04	3.27E-04	3.08E-05	3.30E-05

5. CONCLUSIONS

This paper proposed a combination of polynomial based differential quadrature method in space and a third-order strong stability preserving Runge-Kutta scheme in time for solving the generalized Burgers-Fisher equation. Comparisons of the computed results with exact solutions showed that the method has the capability of solving the generalized Burgers-Fisher equation and is also capable of producing accurate solutions with minimal computational effort. The performance of the technique for the considered problems was measured by comparing with the exact solutions. It was seen that the combined technique approximates the exact solution very well. Ease in use and computationally cost-effectiveness have made the present method an efficient alternative to some rival methods in solving the problems modelled by the nonlinear partial differential equations.

6. ACKNOWLEDGEMENTS

The author would like to thank anonymous referees of *Mathematical and Computational Applications* for their valuable comments and suggestions to improve most of this paper.

7. REFERENCES

1. R.A. Fisher, The wave of advance of advantageous genes, *Annals of Eugenics*, 7, 355–369, 1937.
2. D. Kaya and S.M. El-Sayed, A numerical simulation and explicit solutions of the generalized Burger- Fisher equation, *Applied Mathematics and Computation*, 152, 403-413, 2004.
3. H.N.A. Ismail and A.A.A. Rabboh, A restrictive Padé approximation for the solution of the generalized Fisher and Burger-Fisher equation, *Applied Mathematics and Computation*, 154, 203-210, 2004.
4. H.N.A. Ismail, K. Raslan and A.A.A. Rabboh, Adomian decomposition method for Burger's Huxley and Burger's-Fisher equations, *Applied Mathematics and Computation*, 159, 291-301, 2004.
5. H. Chen and H. Zhang, New multiple soliton solutions to the general Burgers–Fisher equation and the Kuramoto–Sivashinsky equation, *Chaos, Solitons and Fractals* 19, 71–76, 2004.
6. S.A. El-Wakil and M.A. Abdou, Modified extended tanh-function method for solving nonlinear partial differential equation, *Chaos, Solitons and Fractals*, 31,1256–1264, 2007.
7. A.M. Wazwaz, The tanh method for generalized forms of nonlinear heat conduction and Burgers-Fisher equations, *Applied Mathematics and Computation*, 169, 321–338, 2005.
8. A.M. Wazwaz, The tanh–coth method for solitons and kink solutions for nonlinear parabolic equations, *Applied Mathematics and Computation*, 188, 1467–1475, 2007.
9. M. Moghimi and F.S.A. Hejazi, Variational iteration method for solving generalized Burger–Fisher and Burger equations, *Chaos, Solitons and Fractals*, 33, 1756–1761, 2007.

10. H. Fahmy, Travelling wave solutions for some time-delayed equations through factorizations, *Chaos, Solitons and Fractals*, 38,1209-1216, 2008.
11. A. Golbabai and M. Javidi, A spectral domain decomposition approach for the generalized Burger's–Fisher equation, *Chaos, Solitons and Fractals*, 39, 385-392, 2009.
12. R. Bellman, B.G. Kashef and J. Casti, Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations, *Journal of Computational Physics*, 10, 40-52, 1972.
13. C.W. Bert and M. Malik, Differential quadrature method in computational mechanics: A review, *Applied Mechanics Review*, 49, 1-27, 1996.
14. O. Civalek, Large Deflection Static and Dynamic Analysis of Thin Circular Plates Resting on Two-Parameter Elastic Foundation: HDQ/FD Coupled Methodology Approaches, *International Journal of Computational Mechanics*, 2(2), 271-291, 2005.
15. B. Kaya, Solution of the advection-diffusion equation using the differential quadrature method, *KSCE Journal of Civil Engineering*, 14(1), 69-75, 2010.
16. B. Kaya, Y. Arısoy, Differential Quadrature Method for Linear Long Wave Propagation in Open Channels, 'Wave Propagation in Materials for Modern Applications', Ed.; Andrey Petrin, ISBN 978-953-7619-65-7, Published by Intech, Vukovar, Croatia, 253-266, 2010.
17. M. Sari, Differential quadrature method for singularly perturbed two-point boundary value problems, *Journal of Applied Sciences* 8, 1091–1096, 2008.
18. C. Shu, Generalized differential-integral quadrature and application to the simulation of incompressible viscous flows including parallel computation, Ph.D. Thesis, University of Glasgow, UK, 1991.
19. C. Shu and B.E. Richards, Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations, *International Journal for Numerical Methods in Fluids* 15(7), 791-798, 1992.
20. U. Yücel, Approximations of Sturm-Liouville eigenvalues using differential quadrature (DQ) method, *Journal of Computational and Applied Mathematics*, 192, 310-319, 2006.
21. U. Yücel and M. Sari, Differential quadrature method (DQM) for a class of singular two-point boundary value problems, *International Journal of Computer Mathematics*, 86, 465–475, 2009.
22. Z. Zong and KY. Lam, A localized differential quadrature (LDQ) method and its applications to the 2D wave equation, *Computational Mechanics*, 29, 382-391, 2002.
23. M. Sari and G. Gürarlan, Numerical solutions of the generalized Burgers-Huxley equation by a differential quadrature method, *Mathematical Problems in Engineering*, 2009, 2009, Article ID 370765.
24. G. Gürarlan and M. Sari, Numerical solutions of linear and nonlinear diffusion equations by a differential quadrature method (DQM), *Communications in Numerical Methods in Engineering* (2009) DOI: 10.1002/cnm.1292
25. S.T. Choi and Y.T. Chou, *Structural analysis by differential quadrature method using the modified weighting matrices*, Proceedings of American Society of Mechanical Engineers International Computer in Engineering Conference, Atlanta, Georgia, U.S.A., 1998.

26. K.M. Liew, Y.Q. Huang and J.N. Reddy, Analysis of general shaped thin plates by the moving least-squares differential quadrature method, *Finite Elements in Analysis and Design*, 40, 1453–1474, 2004.
27. F.L. Liu and K.M. Liew, Differential quadrature element method: a new approach for free vibration analysis of polar Mindlin plates having discontinuities, *Computer Methods in Applied Mechanics and Engineering*, 179, 407–423, 1999.
28. A.G. Striz, W.L. Chen and C.W. Bert, Static analysis of structures by quadratic element method (QEM), *International Journal of Solids and Structures*, 31, 2807–2818, 1994.
29. H. Zhong, Triangular differential quadrature and its application to elastostatic analysis of Reissner plates, *International Journal of Solids and Structures*, 38, 2821–2832, 2001.
30. C. Shu and Y.T. Chew, Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems, *Communications in Numerical Methods in Engineering*, 13, 643–653, 1997.
31. C. Shu and H. Xue, Explicit computation of weighting coefficients in the harmonic differential quadrature, *Journal of Sound and Vibration*, 204, 549–555, 1997.
32. C. Shu, *Differential Quadrature and Its Application in Engineering*, Springer-Verlag, London, 2000.
33. S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, *SIAM Review* 43, 89–112, 2001.