Cubic B-spline differential quadrature methods and stability for Burgers' equation

Article *in* Engineering Computations · March 2013

DOI: 10.1108/02644401311314312

Idris Dag
Eskisehir Osmangazi University

Received 23 August 2011
Revised 17 March 2012
6 May 2012
Accepted 10 May 2012

Cubic B-spline differential quadrature methods and stability for Burgers' equation

Alper Korkmaz
Department of Mathematics, Çankırı Karatekin University, Çankırı, Turkey, and
İdris Dağ
Department of Mathematics and Computer Science, Eskişehir Osmangazi University, Eskişehir, Turkey

Abstract

Purpose – The purpose of this paper is to simulate numerical solutions of nonlinear Burgers' equation with two well-known problems in order to verify the accuracy of the cubic B-spline differential quadrature methods.

Design/methodology/approach – Cubic B-spline differential quadrature methods have been used to discretize the Burgers' equation in space and the resultant ordinary equation system is integrated via Runge-Kutta method of order four in time. Numerical results are compared with each other and some former results by calculating discrete root mean square and maximum error norms in each case. A matrix stability analysis is also performed by determining eigenvalues of the coefficient matrices numerically.

Findings – Numerical results show that differential quadrature methods based on cubic B-splines generate acceptable solutions of nonlinear Burgers' equation. Constructing hybrid algorithms containing various basis to determine the weighting coefficients for higher order derivative approximations is also possible.

Originality/value – Nonlinear Burgers' equation is solved by cubic B-spline differential quadrature methods.

Keywords Cubic B-spline, Differential quadrature, Burgers' equation, Shock waves, Sinusoidal disturbance, Stability, Differential equations, Matrix algebra

Paper type Research paper

1. Introduction

In the last decades, to solve partial differential equations numerically has become very popular in parallel to developments in computers. So far many numerical methods finite difference, finite elements, spectral methods, etc. developed in order to obtain numerical solutions of partial differential equations (Dehghan and Salehi, 2010; Dehghan and Shokri, 2008; Dehghan, 2006; Saadatmandi and Dehghan, 2011).

Different from classical methods such as finite difference and finite element methods, differential quadrature method (DQM) was developed by Bellman *et al.* (1972). So far, Lagrange and Legendre polynomials, spline functions used to generate various DQMs (Bellman *et al.*, 1972, 1976; Quan and Chang, 1989a, b; Shu and Richards, 1992).

This paper was presented in the Second International Symposium on Computing in Science and Engineering (ISCSE 2011) – Kusadasi/Turkey.

Engineering Computations: International Journal for Computer-Aided Engineering and Software
Vol. 30 No. 3, 2013
pp. 320-344

0264-4401
DOI 10.1108/02644401311314312

Hermite DQM has been proposed by Cheng *et al.* (2005). Recently, DQMs based on spline functions of various degrees have been applied to various problems (Zhong, 2004; Guo and Zhong, 2004; Zhong and Lan, 2006). Guo and Zhong (2004) have determined the weighting coefficients via cardinal sextic B-splines in order to analyze the non-linear free vibrations of beams with various boundary conditions. Zhong and Lan (2006) developed cardinal cubic B-spline DQM by normalizing the problem interval to [0, 1]. Sinc functions has also been used as basis to obtain the weighting coefficients by Bonzani (1997). Wu and Shu have studied radial basis functions to develop DQM to obtain numerical solutions of Burgers' equation (Shu and Wu, 2002).

Because of generating accurate numerical solutions and easy application in the solution process of many problems in physics and engineering, various DQMs have been implemented to solve many engineering and physics problems (Malekzadeh and Karami, 2005; Mansell *et al.*, 1993; Lee *et al.*, 2004; Civalek, 2006; Korkmaz and Dağ, 2009a, b, c, 2008, 2011a, b, c; Korkmaz *et al.*, 2010; Korkmaz, 2010; Saka *et al.*; Mittal and Jiwari, 2009a, b, 2011a, b; Zhou and Cheng, 2011).

Since it has analytical solutions, the Burgers' equation has been used as test problem to verify the numerical algorithms. The form of one-dimensional non-linear Burgers' equation is:

$$U_t + UU_x - \nu U_{xx} = 0, \tag{1}$$

where $\nu > 0$ is the kinematic viscosity coefficient. Initial and boundary conditions are chosen as:

$$U(x, 0) = f(x), \quad a < x < b \tag{2}$$

$$U(a, t) = \alpha_1, U(b, t) = \alpha_2, \quad t \in [0, T]. \tag{3}$$

Burgers' equation is a quasi-linear parabolic partial differential equation, whose analytic solutions can be constructed from a linear partial differential equation by using Hopf-Cole transformation (Burgers, 1948; Hopf, 1950; Cole, 1951). Besides, this equation is very important fluid dynamical model both for the conceptual understanding of a class of physical flows and for testing various numerical algorithms. Since it is a simple form of the Navier-Stokes equations, many researchers focus on this equation due to the form of the non-linear terms and the occurrence of high order derivatives with small coefficients in both. The Burgers' equation is a model in many fields such as gas dynamics (Lighthill, 1956), propagation of elastic waves (Pospelov, 1966), turbulence problems (Burgers, 1948), heat conduction (Cole, 1951), number theory (van der Pol, 1951), shock waves (Cole, 1951) and continuous stochastic processes (Cole, 1951), etc. When arbitrary initial conditions are considered Burgers' equation can be solved analytically (Hopf, 1950). But some analytic solutions consists of infinite series, converging very slowly for small viscosity coefficient ν (Miller, 1966). Thus, many numerical methods including finite difference and finite element were implemented to solve the Burgers' equation numerically (Christie *et al.*, 1981; Nguyen and Reynen, 1982; Herbst *et al.*, 1982; Caldwell and Smith, 1982; Dehghan *et al.*, 2007).

Polynomials of higher degrees make big oscillations. These oscillations may cause large numerical errors. Instead of construction of numerical methods based on high degree polynomials to solve partial differential equations, using piecewise low order polynomials may reduce numerical errors. So far, many numerical methods based on B-splines have been developed (Lakestani and Dehghan, 2012, 2009; Dehghan and Lakestani, 2008; Saka *et al.*, 2009). In this paper, we present the formulation to solve Burgers' equation by differential quadrature based on cubic B-spline functions to verify the validity of the present methods. After spatial discretization, the resulting ordinary differential equation is integrated using classical four stage Runge-Kutta method in time.

2. Differential quadrature method

We consider a uniform partition $a \leq x \leq b$ where $[a, b]$ is the problem domain. Let the knots be $a = x_1 < x_2 < \cdots < x_N = b$ and mesh size be $h = x_m - x_{m-1}$, $m = 2, 3, \ldots, N$. Then, the cubic B-splines τ_k, $k = 0, 1, \ldots, N+1$ span the interval $[a, b]$. The cubic B-splines are defined as:

$$\tau_k(x) = \frac{1}{h^3} \begin{cases} (x - x_{i-2})^3, & [x_{i-2}, x_{i-1}] \\ h^3 + 3h^2(x - x_{i-1}) + 3h(x - x_{i-1})^2 - 3(x - x_{i-1})^3, & [x_{i-1}, x_i] \\ h^3 + 3h^2(x_{i+1} - x) + 3h(x_{i+1} - x)^2 - 3(x_{i+1} - x)^3, & [x_i, x_{i+1}] \\ (x_{i+2} - x)^3, & [x_{i+1}, x_{i+2}] \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

over the domain $[a, b]$ (Prenter, 1975). Let $U(x)$ be a sufficiently smooth function $U(x)$ over the problem domain. Then, the pth order derivative of $U(x)$ with respect to x at a grid point x_i can be approximated by weighted sum of all the functional values in the same domain, that is:

$$U_x^{(p)}(x_i) = \left.\frac{d^{(p)}U(x)}{dx^{(p)}}\right|_{x_i} = \sum_{j=1}^{N} w_{ij}^{(p)} U(x_j), \quad for \quad i = 1, 2, \ldots, N \tag{5}$$

where $w_{ij}^{(p)}$ represents the weighting coefficients, and N is the number of grid points in the whole domain (Bellman *et al.*, 1972).

Forming a basis for the functions defined over $[a, b]$ leads cubic B-splines to be used as test functions to determine the weighting coefficients $w_{ij}^{(p)}$ necessary for the DQM derivative approximations. Using definition of the DQM and cubic B-spline functions as test functions, the main derivative approximation equality:

$$\frac{\partial \tau_k^{(p)}}{\partial x^{(p)}}(x_i) = \sum_{j=-1}^{N+2} w_{ij}^{(p)} \tau_k(x_j), \quad i = 1, 2, \ldots, N; \quad k = 0, 1, \ldots, N+1 \tag{6}$$

is obtained for the approximations of the derivatives of order p. In the equation (6), the grid points x_{-1}, x_0, x_{N+1}, x_{N+2} are the ghost knots and located at out of the problem domain while $w_{i,-1}^{(1)}, w_{i,0}^{(1)}, w_{i,N+1}^{(1)}, w_{i,N+2}^{(1)}$ are ghost coefficients that they will not be used while solving the test problems.

3. First-order derivative approximation

The weighting coefficients $w_{i,j}^{(1)}, j = -1, 0, \ldots, N+2$ related to the grid point x_i are determined by using the test functions τ_k, $k = 0, 1, \ldots, N+1$ at the grid point x_i. Substituting the cubic B-splines τ_k into equation (6) when p is 1 gives the algebraic equation system:

$$\begin{bmatrix} 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} w_{i,-1}^{(1)} \\ w_{i,0}^{(1)} \\ \vdots \\ w_{i,N+2}^{(1)} \end{bmatrix} = \begin{bmatrix} \tau_0'(x_i) \\ \tau_1'(x_i) \\ \vdots \\ \tau_{N+1}'(x_i) \end{bmatrix} \tag{7}$$

where $w_{i,-1}^{(1)}, w_{i,0}^{(1)}, w_{i,N+1}^{(1)}, w_{i,N+2}^{(1)}$ are ghost coefficients. There exist $N+4$ unknowns and $N+2$ equations in the system (7). Deriving the first and the last equations of the system (7), we get:

$$\tau_0''(x_i) = \sum_{j=-1}^{N+2} w_{ij}^{(1)} \tau_0'(x_j)$$

and:

$$\tau_{N+1}''(x_i) = \sum_{j=-1}^{N+2} w_{ij}^{(1)} \tau_{N+1}'(x_j).$$

Adding these equations to the system, the algebraic system becomes of the form:

$$\begin{bmatrix} 3 & 0 & -3 & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 1 \\ & & & & 3 & 0 & -3 \end{bmatrix} \begin{bmatrix} w_{i,-1}^{(1)} \\ w_{i,0}^{(1)} \\ w_{i,1}^{(1)} \\ \vdots \\ w_{i,N+1}^{(1)} \\ w_{i,N+2}^{(1)} \end{bmatrix} = \begin{bmatrix} \tau_0''(x_i) \\ \tau_0'(x_i) \\ \tau_1'(x_i) \\ \vdots \\ \tau_{N+1}'(x_i) \\ \tau_{N+1}''(x_i) \end{bmatrix}$$

containing $N+4$ equations and $N+4$ unknowns. This tri-diagonal system can be solved using efficient Thomas algorithm of the computational complexity of $o(n)$.

3.1 Second-order derivative approximations

Owing to the existence of various basis spanning the problem domain, various basis functions can be used in order to determine the weighting coefficients of the second-order derivative approximations.

3.1.1 Method I. Using cubic B-spline functions and their second derivatives in the fundamental equation (6), all the weighting coefficients can be determined directly. Using the equation:

$$\tau_k''(x_i) = \sum_{j=-1}^{N+2} w_{ij}^{(2)} \tau_k(x_j), \quad i = 1, 2, \ldots, N; \quad k = 0, 1, \ldots, N+1 \tag{8}$$

where $w_{ij}^{(2)}$ is the weighting coefficients of the second-order derivative approximations as $p = 2$ generates the following algebraic equation system:

$$\begin{bmatrix} 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} w_{i,-1}^{(2)} \\ w_{i,0}^{(2)} \\ \vdots \\ \\ w_{i,N+2}^{(2)} \end{bmatrix} = \begin{bmatrix} \tau_0''(x_i) \\ \tau_1''(x_i) \\ \vdots \\ \\ \tau_{N+1}''(x_i) \end{bmatrix}$$

which contains $N+4$ unknowns and $N+2$ equations. Since cubic B-splines do not have third-order derivatives, it is not possible to generate two more equations. But eliminating the first and the last columns of the coefficient matrix and $w_{i,-1}^{(2)}$ and $w_{i,N+2}^{(2)}$ from the unknowns vector converts the system to a solvable system. Then, the system can be solved via Thomas algorithm.

3.1.2 Method II. Existence of more than one basis functions spanning an N-dimensional vector space gives opportunity to determine the weighting coefficients in the same space by various basis. The second method to determine the weighting coefficients of the second-order derivative approximations is based on using the advantages of polynomial based differential quadrature. Once the weighting coefficients of the first-order derivative approximations are determined using cubic B-splines, the weighting coefficients of the second-order derivative approximations can be determined using Shu's (2000) explicit formula:

$$w_{ij}^{(2)} = 2w_{ij}^{(1)} \left(w_{ii}^{(1)} - \frac{1}{(x_i - x_j)} \right), \quad for \quad i \neq j$$

$$w_{ii}^{(2)} = -\sum_{j=1, j\neq i}^{N} w_{ij}^{(2)}$$

3.1.3 Method III. According to Shu (2000), there exist various ways to determine the higher order derivative approximations in the DQM. One of these ways is matrix multiplication approach (MMA). Let:

$$\frac{\partial^2 U}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial U}{\partial x}\right) \tag{9}$$

be the derivative operator of the second-order derivative. When DQM approximation is applied to the left side of the derivative (9), we obtain the equation:

$$U_x^{(2)}(x_i) = \sum_{j=1}^{N} w_{ij}^{(2)} U(x_j), \quad i = 1, 2, \ldots, N \tag{10}$$

Similarly, when DQM is applied to the right side of the equation (9) twice, it becomes of the form:

$$\begin{aligned} U_x^{(2)}(x_i) &= \sum_{k=1}^{N} w_{ik}^{(1)} U_x^{(1)}(x_k) = \sum_{k=1}^{N} w_{ik}^{(1)} \sum_{j=1}^{N} w_{kj}^{(1)} U(x_j) \\ &= \sum_{j=1}^{N} \left[\sum_{k=1}^{N} w_{ik}^{(1)} w_{kj}^{(1)}\right] U(x_j), i = 1, 2, \ldots, N \end{aligned} \tag{11}$$

where $U_x^{(1)}$ is the first-order derivative of $U(x)$. Comparing equations (10) and (11) gives the equation:

$$w_{ij}^{(2)} = \sum_{k=1}^{N} w_{ik}^{(1)} w_{kj}^{(1)} \tag{12}$$

which is the definition of matrix multiplication (Shu, 2000). Assuming $M^{(1)} = w_{ij}^{(1)}, M^{(2)} = w_{ij}^{(2)}$ be matrix forms of the weighting coefficients of the first- and the second-order derivative approximations, respectively, then the matrix multiplication given in equation (12) can be written as:

$$M^{(2)} = M^{(1)} M^{(1)} \tag{13}$$

Equation (13) shows that the weighting coefficients of the second-order derivative approximation can be computed by the matrix multiplication of the weighting coefficients of the first-order derivative approximation. In general case, the weighting coefficient matrix $M^{(p)}$ of the pth order derivative can be determined via multiplication of two weighting coefficient matrices of $M^{(m)}$ and $M^{(n)}$, that is:

$$M^{(p)} = M^{(m)} M^{(n)}, m, n = 1, 2, 3, \ldots, N-1$$

when $m + n = p$ (Shu, 2000).

In the study, MMA is used as the third method to determine the weighting coefficients $M^{(2)}$ when $M^{(1)}$ is determined by cubic B-splines.

4. Discretization and stability analysis

Usage of DQM approximations for spatial discretization of the Burgers' equation leads to the ordinary differential equation system:

$$\frac{\partial U(x_i)}{\partial t} = -U(x_i)\sum_{j=1}^{N} w_{ij}^{(1)} U(x_j) + \nu \sum_{j=1}^{N} w_{ij}^{(2)} U(x_j), \quad i = 1, 2, \ldots, N \tag{14}$$

Due to memory allocation, computational cost and accuracy, we have integrated equation (14) in time with Runge-Kutta method of order four.

Different from classical finite difference methods, Von Neuman stability analysis cannot be performed DQM discretized systems. Instead, matrix stability or energy stability methods have been studied for DQMs in various papers (Tomasiello, 2010, 2003, 2011). In the present study, we perform a matrix stability analysis for the system (14).

A time-dependent problem is given as:

$$\frac{\partial f}{\partial t} = l(f) \tag{15}$$

with proper initial and boundary conditions, where l is a spatial non-linear differential operator. After discretization via DQM and linearization of the non-linear term $U(x)U_x(x)$ by assuming $U(x)$ locally constant (Saka *et al.*, 2009), equation (15) is reduced into a set of ordinary differential equations in time:

$$\frac{d\{U\}}{dt} = [A]\{U\} + \{g\} \tag{16}$$

where $\{U\}$ is an unknown vector of the functional values at the points except both boundary points, $\{g\}$ is a vector containing the non-homogenous part and the boundary conditions and A is the coefficient matrix (Jain, 1983).

The stability of a numerical scheme for numerical integration of equation (16) depends on the stability of the ordinary differential equation (16). If the ordinary differential equation system (16) is not stable, numerical methods may not generate converged solutions. The stability of equation (16) is related to the eigenvalues of the matrix A, since its exact solution is directly determined by the eigenvalues of A. When all Re $(\lambda_i) \leq 0$ for all i is enough to show the stability of the exact solution of $\{U\}$ as $t \rightarrow \infty$ where $\mathrm{Re}(\lambda_i)$ denotes the real part of the eigenvalues λ_i of the matrix A (Jain, 1983). When the eigenvalues are complex, there exist some tolerance that the real parts of eigenvalues may be small positive numbers. For details, the readers are recommended to see Jain (1983).

Even though the theoretical side of the numerical stability seems to be determined easily, when the dimension of A is large and A is full, to determine the eigenvalues of A is not easy. In the study, we have determined the eigenvalues of A using the algorithm given in Press *et al.* (1992).

After implementation of boundary conditions, the equation (14) can be rewritten as:

$$\frac{\partial U_i}{\partial t} = -U_i\sum_{j=2}^{N-1} w_{ij}^{(1)} U_j + \nu \sum_{j=2}^{N-1} w_{ij}^{(2)} U_j + G_i, \quad i = 2, 3, \ldots, N \tag{17}$$

where $G_i = -U_i\left(w_{i,0}^{(1)}U_0 + w_{i,N}^{(1)}U_N\right) + v\left(w_{i,0}^{(2)}U_0 + w_{i,N}^{(2)}U_N\right)$. In the matrix notation, equation (17) can be written as:

$$\begin{bmatrix} \frac{\partial U_2}{\partial t} \\ \frac{\partial U_3}{\partial t} \\ \vdots \\ \\ \frac{\partial U_{N-1}}{\partial t} \end{bmatrix} = A \begin{bmatrix} U_2 \\ U_3 \\ \vdots \\ \\ U_{N-1} \end{bmatrix} + G$$

where $A_{i,j} = -\alpha_i w_{ij}^{(1)} + v w_{ij}^{(2)}$ and $\alpha_i = U_i$.

5. Test problems

The numerical solutions of Burgers' equation are presented for two widely used problems. Accuracy of the method for both test problems are measured with discrete L_2 and L_∞ error norms given by:

$$L_2 = \sqrt{h\sum_{j=0}^{N}\left|U_j^{\text{exact}} - (U_N)_j\right|^2}, \quad L_\infty = |U^{\text{exact}} - U_N|_\infty = \max_j \left|U_j^{\text{exact}} - (U_N)_j\right|$$

Test problem A

Shock-like solution of the Burgers' equation has analytic solution of the form:

$$U(x,t) = \frac{x/t}{1 + \sqrt{(t/\Delta)}\exp(x^2/4\nu t)}, \quad t \geq 1, 0 \leq x \leq 1.2 \tag{18}$$

where $\Delta = \exp(1/8\nu)$ (Nguyen and Reynen, 1982). This solution of Burgers' equation is deflation of an initial pulse as time goes.

The simulations of the deflation is studied by using the functional values of the exact solution (18) at time $t = 1$ as initial condition and the boundary conditions $U(0, t) = U(1.2, t) = 0$. For the sake of comparison of all three approximations, we have chosen the initial dataset $N = 241, \Delta t = 0.001, \nu = 0.005$. The initial shock and the analytical and numerical solutions at time $t = 3.6$ are shown in Figures 1 and 2.

Just only process time of the calculation of weighting coefficients and accuracy comparison of three methods are given in Table I. The computational cost of MMA seems more than the costs of first two approaches.

We also depicted the errors at the terminating time $t = 3.6$. The errors are accumulated at both boundaries for first two methods, in MMA the maximum error is seen just nearby of the left boundary (Figures 3-5).

We have determined the rate of convergence for all three methods by:

$$\text{rate of convergence} \approx \frac{ln(E(N_2)/E(N_1))}{ln(N_1/N_2)}$$

where $E(N_j)$, $j = 1, 2$, the error norm as N_j are number of grid points. The algorithm is run for various numbers of grids as the time step is fixed as $\Delta t = 0.001$. The rates of convergence for each method at the time $t = 3.6$ are tabulated in Table II.

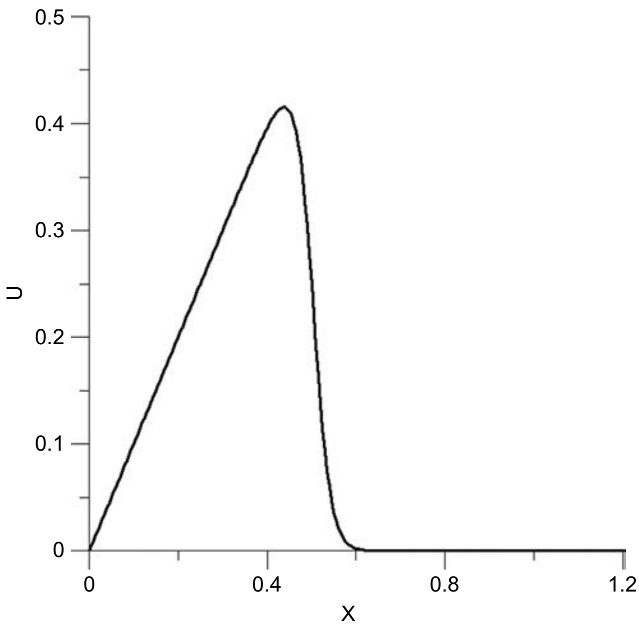


Figure 1.
Initial state

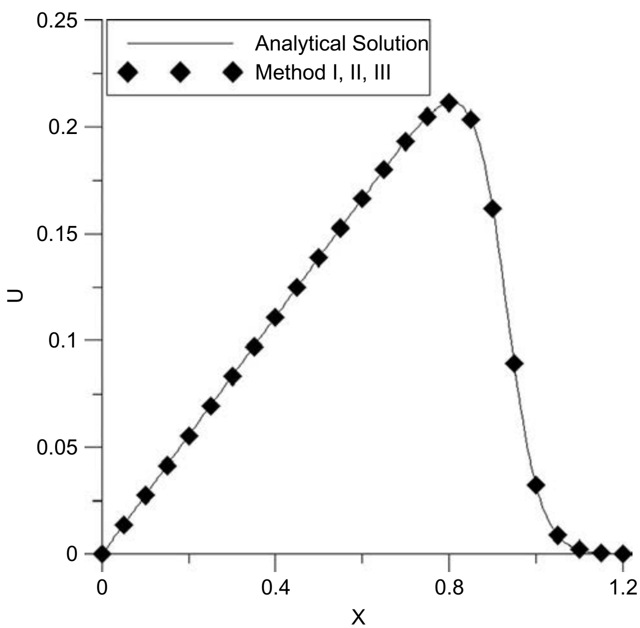


Figure 2.
Solutions at the terminating time $t = 3.6$

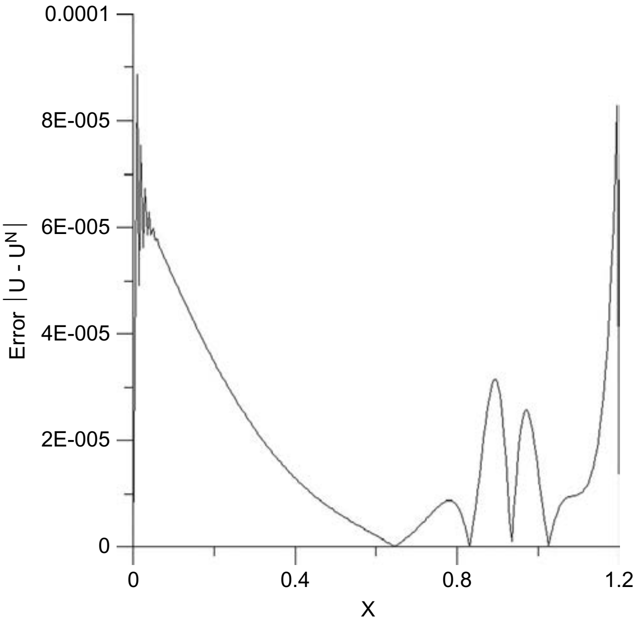
We have also plotted the graphs of N-rate of convergence (both for L_2 and L_∞) in order to be able to see more clearly for all methods (Figures 6 and 7). While the rates of convergence of methods I and II increase as the number of grids increases, the rate of convergence of method III makes big oscillations as the number of grids increases from 21 to 81. When more than 81 grids are used to discretisize the problem domain, the rate of convergence decreases slowly.

A matrix stability analysis is also performed for all approaches. First, the coefficient matrix A is transformed into its similar Hessenberg matrix H via QR method. Owing to the computational capacity of the computer, the procedure is ended when $|A_{3,1}|$ becomes lower than 1×10^{-320}. One should note that the cost of the transformation of a full matrix to its similar Hessenberg form via QR method is very high when N is large. So, we skip to determine the eigenvalues for larger Ns.

Eigenvalues of the proposed methods for various number of grids are shown in Figures 8-11 for method I. As the eigenvalues for $N = 21$ and $N = 31$ have imaginary parts, for $N = 41$ and $N = 61$ the eigenvalues are pure real. Since the real components of all eigenvalues for various number of grids are non-positive, they are in a good agreement with stability conditions.

Table I. Comparison of results at different times over [0, 1.2]

Method	L_2	L_∞	CPU time (s)
Method I	2.69×10^{-5}	8.30×10^{-5}	0.64
Method II	4.91×10^{-5}	1.59×10^{-4}	0.37
Method III	7.85×10^{-5}	3.18×10^{-4}	2.87



Note: Error at terminating time t = 3.6

Figure 3. Method I

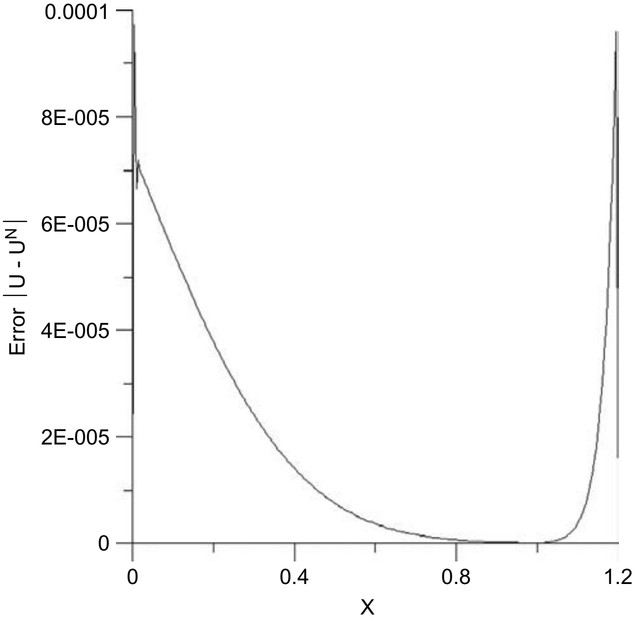


Figure 4.
Method II

Note: Error at terminating time $t = 3.6$

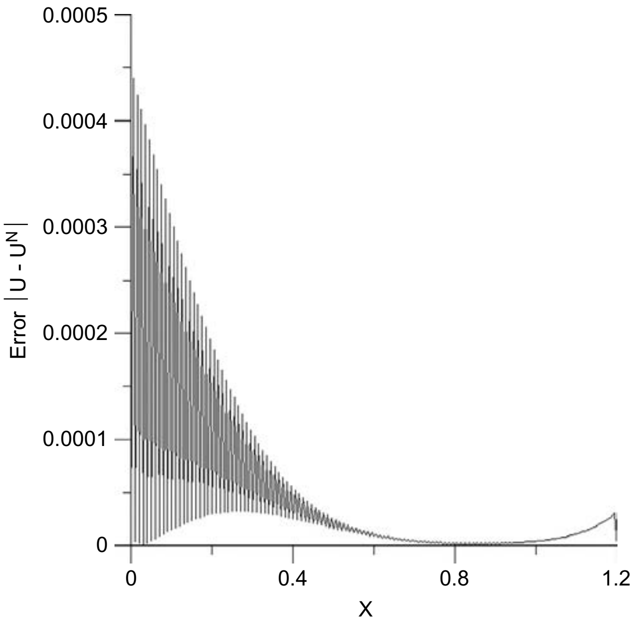


Figure 5.
Method III

Note: Error at terminating time $t = 3.6$

N	Method I		Method II		Method III	
	$L_2 \times 10^3$	*Order*	$L_2 \times 10^3$	*Order*	$L_2 \times 10^3$	*Order*
21	1.64	–	1.41	–	7.05	–
31	1.00	1.27	0.79	1.48	0.94	5.16
41	0.70	1.24	0.57	1.12	0.92	0.06
61	0.44	1.20	0.37	1.09	0.26	3.12
81	0.31	1.22	0.27	1.12	0.20	1.01
101	0.23	1.29	0.21	1.16	0.16	0.89
121	0.18	1.41	0.16	1.25	0.14	0.90
151	0.12	1.64	0.12	1.40	0.11	0.92
161	0.11	1.91	0.11	1.45	0.10	0.84
201	0.06	2.47	0.07	1.76	0.09	0.85
241	0.02	4.83	0.04	2.44	0.07	0.77
	$L_\infty \times 10^3$	*Order*	$L_\infty \times 10^3$	*Order*	$L_\infty \times 10^3$	*Order*
21	3.10	–	3.29	–	11.6	–
31	2.13	0.96	2.22	1.01	1.73	4.89
41	1.61	0.99	1.68	1.00	1.48	0.56
61	1.07	1.04	1.12	1.02	0.95	1.09
81	0.77	1.12	0.83	1.06	0.76	0.81
101	0.59	1.23	0.64	1.12	0.63	0.82
121	0.46	1.38	0.52	1.19	0.54	0.83
151	0.32	1.66	0.39	1.32	0.45	0.82
161	0.28	1.94	0.35	1.47	0.43	0.84
201	0.16	2.38	0.24	1.69	0.36	0.78
241	0.08	3.85	0.15	2.36	0.31	0.73

Table II. Error norms and rates of convergence in space at time $t = 3.6$

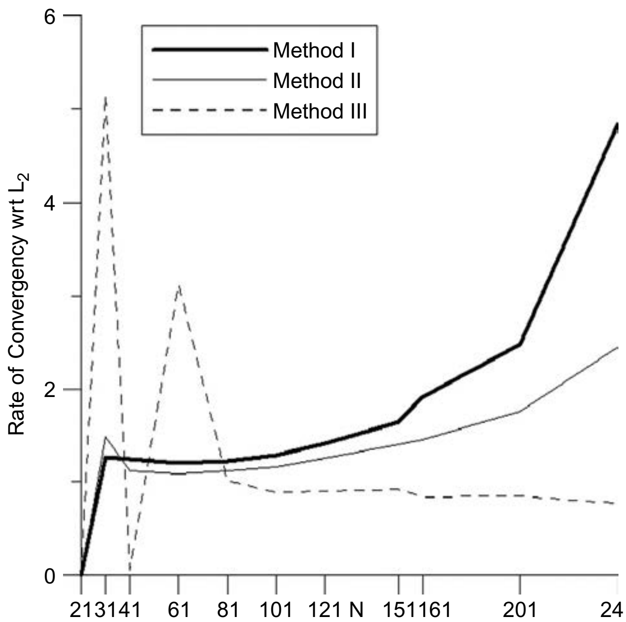


Figure 6. Rate of convergence with respect to L_2

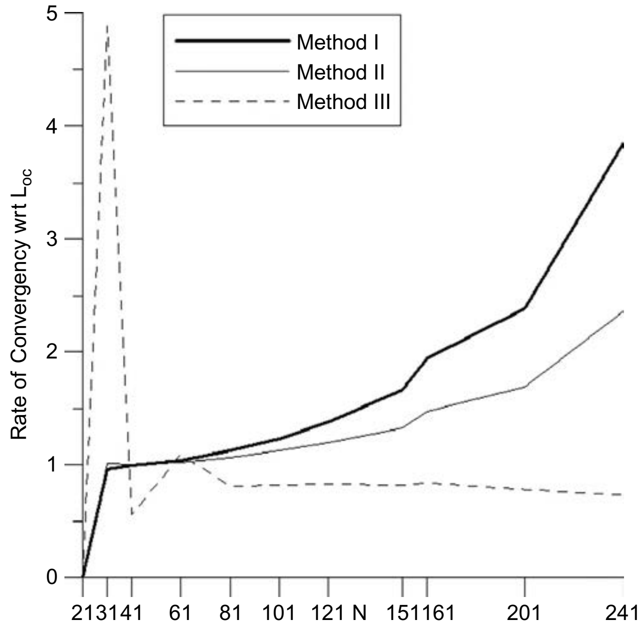
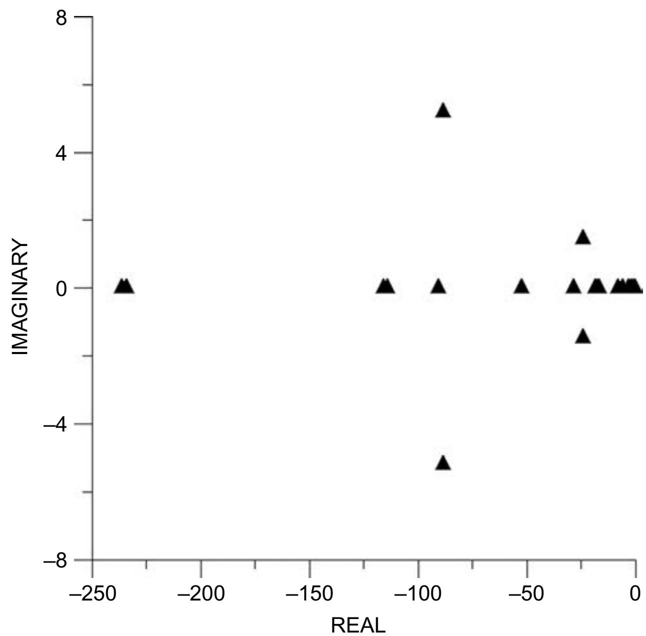
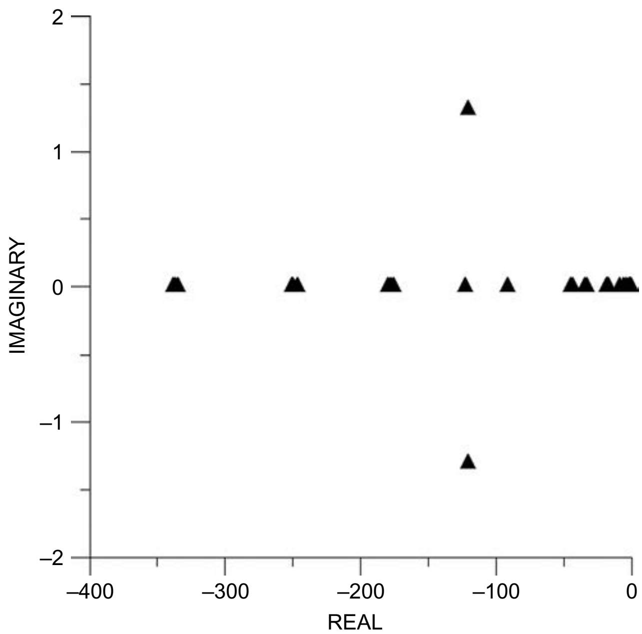


Figure 7.
Rate of convergence with respect to L_∞



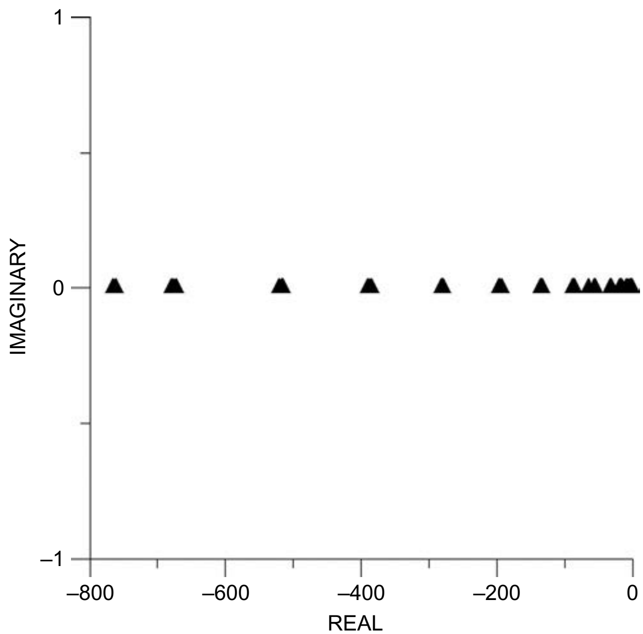
Note: $N = 21$

Figure 8.
Method I



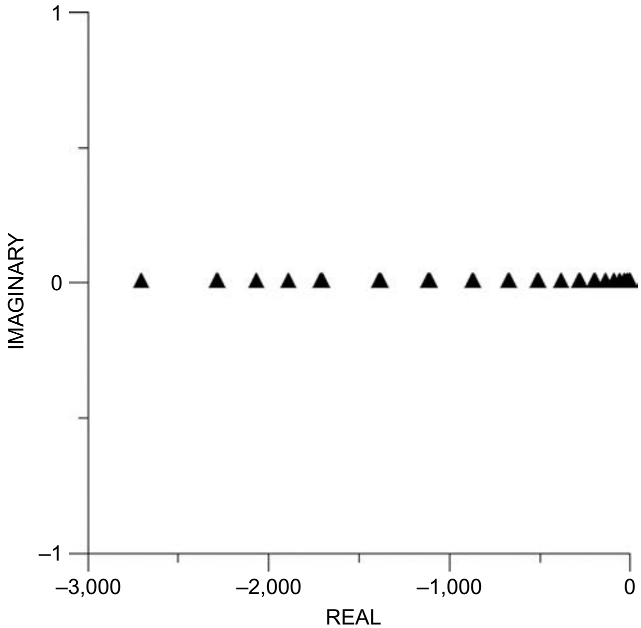
Note: $N = 31$

Figure 9.
Method I



Note: $N = 41$

Figure 10.
Method I



Note: $N = 61$

Figure 11.
Method I

The eigenvalues of the method II for various number of grids are shown in Figures 12-15. For $N = 21$, only one of the eigenvalues is computed positive as 3.50×10^{-2}. Moreover, for $N = 31$, $N = 41$ and $N = 61$, the maximum and the only nonnegative real parts of eigenvalues are determined as 2.06×10^{-2}, 1.08×10^{-2}, 2.55×10^{-2}, respectively. In all cases, the real parts of all the other eigenvalues are negative, which are the indicators of stability.

In Figures 16-19, the eigenvalues for method III are demonstrated. During the stability study of method III, for $N = 41$, the algorithm that determines the eigenvalues of the system has failed to compute all the eigenvalues. Only some of the eigenvalues can be computed. In all cases, the real parts of the eigenvalues are negative.

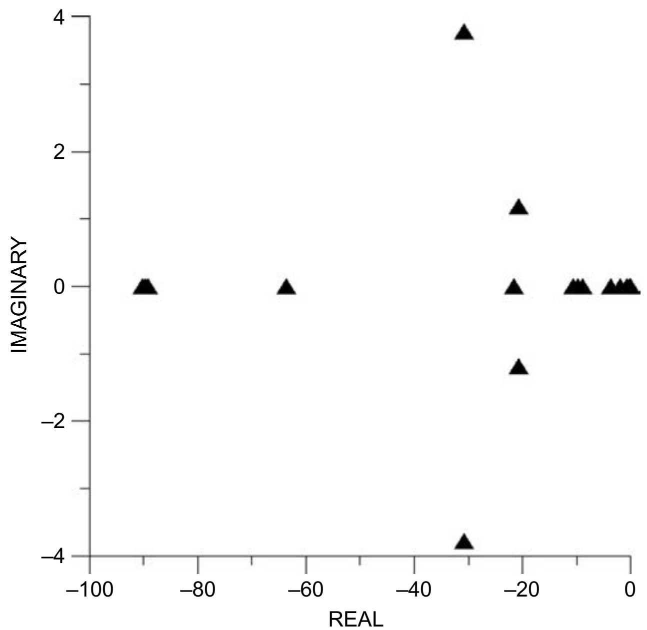
Test problem B

The exact form simulating sinusoidal disturbance solution of Burgers' equation is given by:

$$U(x,t) = \frac{\int_{-\infty}^{\infty}((x-\varsigma)/t)\exp[(-(x-\varsigma)^2/4vt)]\exp\left[(-1/2v)\int_0^{\varsigma}u_0(\eta)d\eta\right]d\varsigma}{\int_{-\infty}^{\infty}\exp[(-(x-\varsigma)^2/4vt)]\exp\left[(-1/2v)\int_0^{\varsigma}u_0(\eta)d\eta\right]d\varsigma}, \tag{19}$$

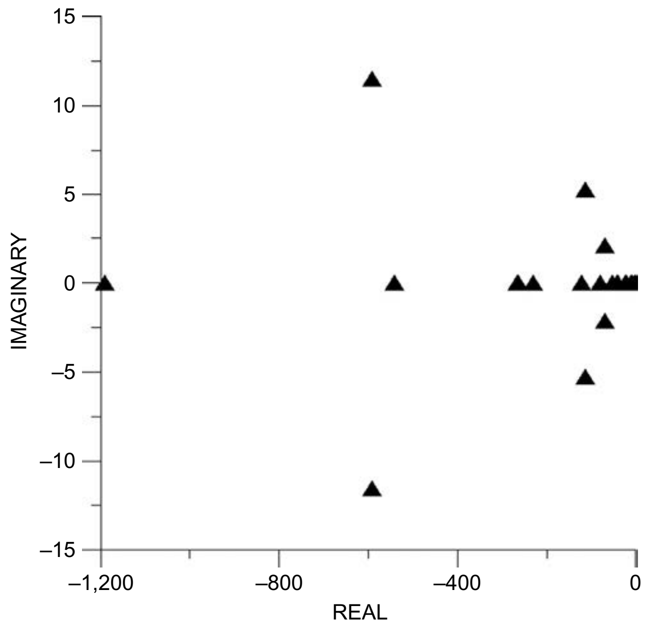
$$0 \leq x \leq 1, t \geq 0$$

where u_0 denotes the initial condition (Cole, 1951). We choose the initial condition $u_0(x) = sin(2\pi x)$ over the interval $[0, 1]$ and the boundary conditions $U(0, t) = U(1, t) = 0$.



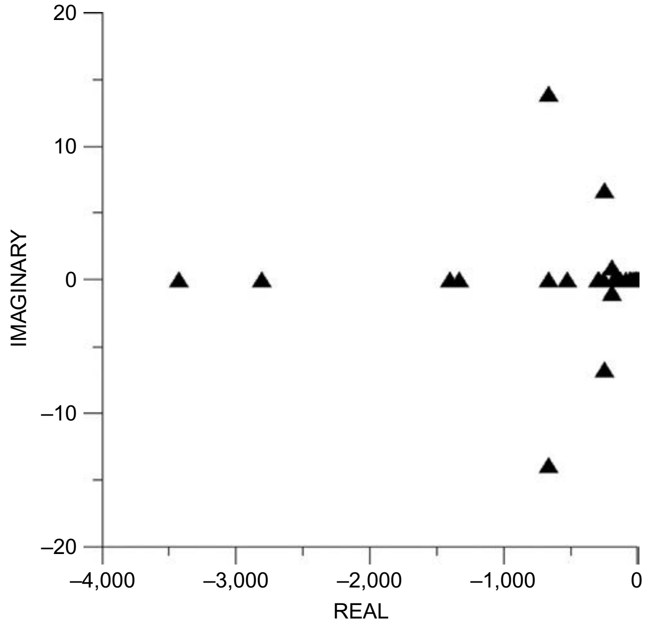
Note: $N = 21$

Figure 12.
Method II



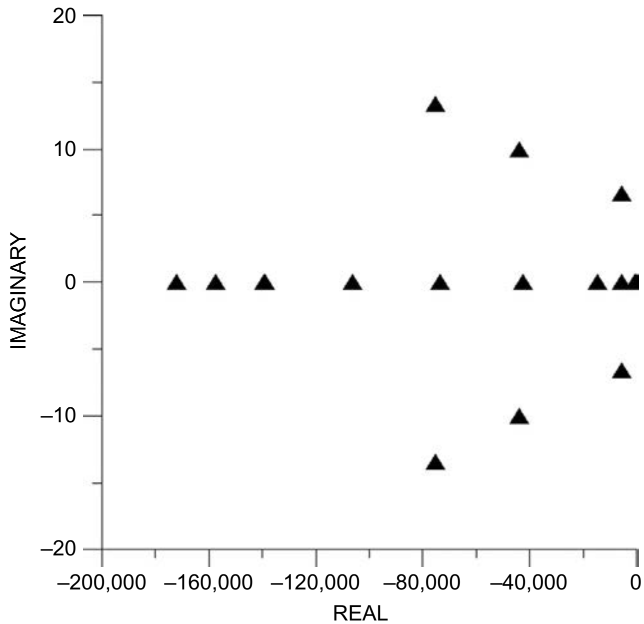
Note: $N = 31$

Figure 13.
Method II



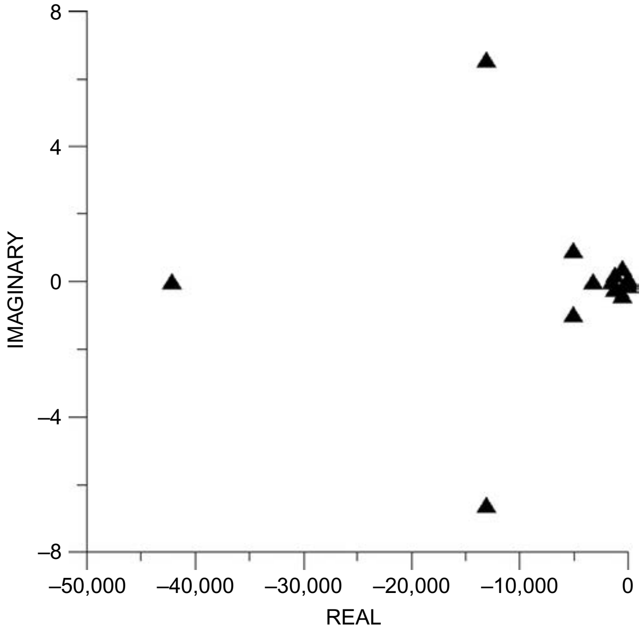
Note: $N = 41$

Figure 14.
Method II



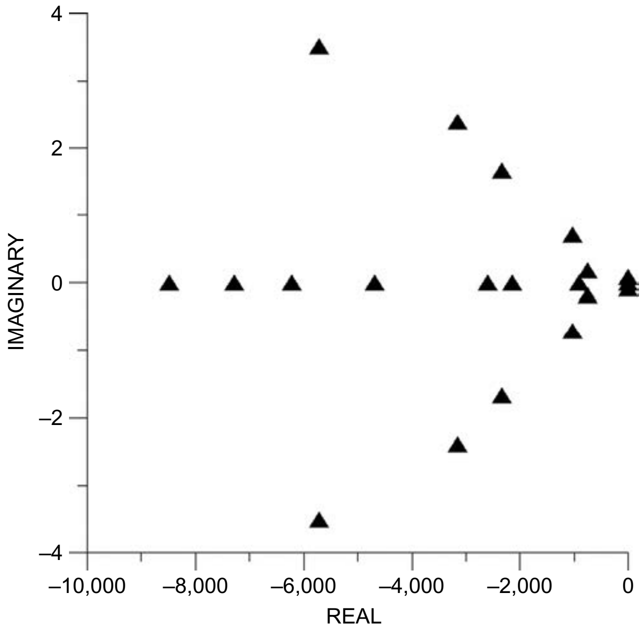
Note: $N = 61$

Figure 15.
Method II



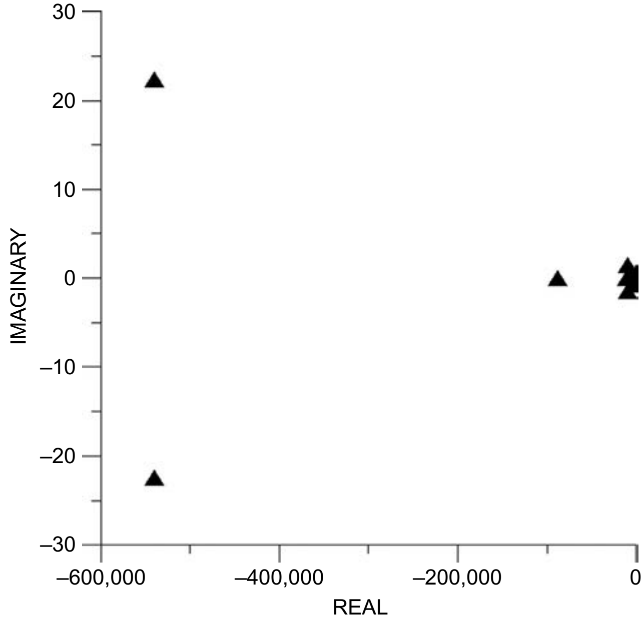
Note: $N = 21$

Figure 16.
Method III



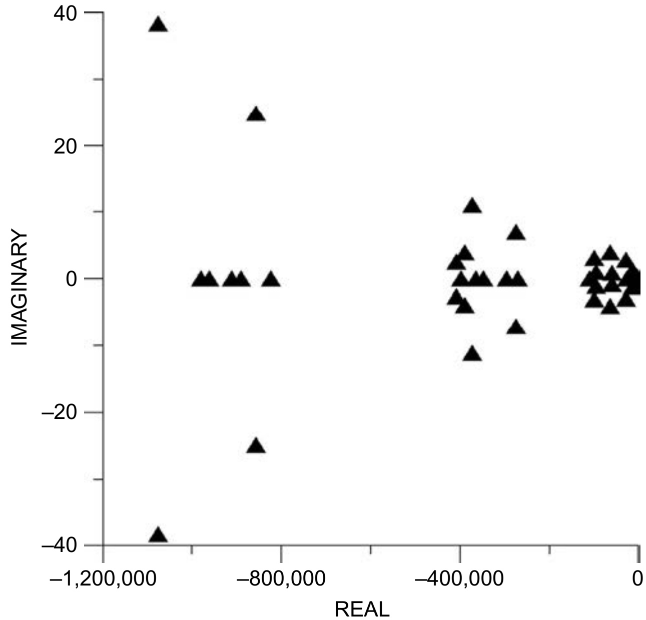
Note: $N = 31$

Figure 17.
Method III



Note: $N = 41$

Figure 18.
Method III



Note: $N = 61$

Figure 19.
Method III

The procedure is run with the parameters $v = \pi/100$, $\Delta t = 0.0025$ for various numbers of grid points. The sinusoidal disturbance simulation is plotted in Figure 20.

The error at terminating time $t = 0.50$ is shown in Figure 21. The maximum error norms at various times for various number of grids and a comparison with an earlier study are shown in Table III. During simulation, DQM generates more accurate solutions than Wavelet-Galerkin method despite the fact that Wavelet-Galerkin method uses smaller time steps.

6. Conclusion

Cubic B-spline based DQMs have been developed to obtain numerical solutions of the Burgers' equation. In method I, from the definition of the DQM, cubic B-splines have been used as basis to determine the weighting coefficients of both the first and the second-order derivative approximations. As method II, the weighting coefficients of the first-order derivative approximation have been determined via cubic B-spline functions while polynomial based DQM has been used to determine the weighting coefficients of the second-order derivative approximations. The usage of the definition of the derivative during the determination of the weighting coefficients of the second-order has been called method III.

In the first case, the shock wave solution has been studied as test problem. The accuracy of the methods has been measured using discrete root mean square error norm and maximum error norm. All the three developed methods generated numerical results successfully. A numerical rate of convergence analysis has been given to show the efficiency of the present methods. Moreover, matrix stability analysis has also been studied for all presented methods.

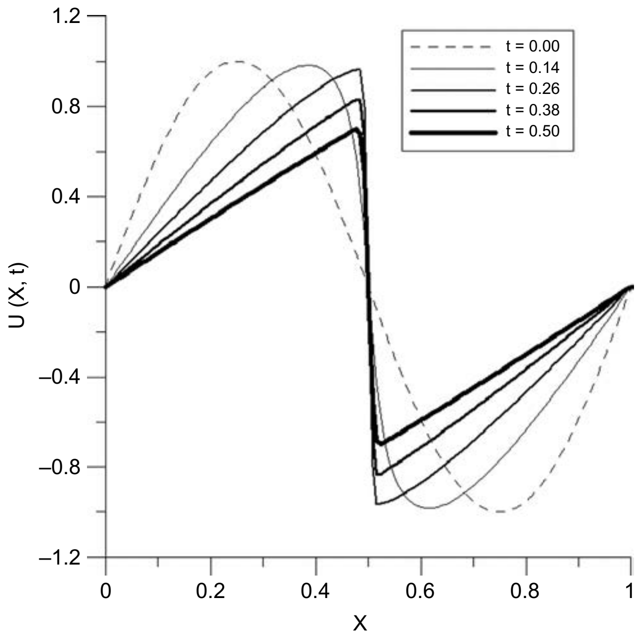


Figure 20. Sinusoidal disturbance simulation for $N = 128$

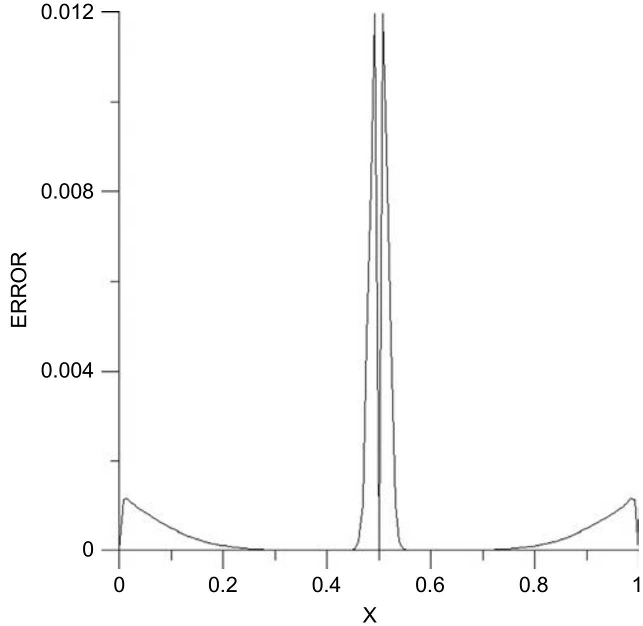


Figure 21.
The error at the time $t = 0.50$ for $N = 128$

Table III.
L_∞ error norm at various times

Method	N	Δt	$t = 0.14$	$t = 0.26$	$t = 0.38$	$t = 0.50$
Method I	16	0.0025	0.0659	0.3559	0.3507	0.3234
Method I	32	0.0025	0.0133	0.2865	0.2853	0.2242
Method I	64	0.0025	0.0074	0.1334	0.1024	0.0544
Method I	128	0.0025	0.0022	0.0155	0.0135	0.0119
Kumar and Mehra (2005)	16	10^{-4}	0.1588	0.6856	0.6955	
Kumar and Mehra (2005)	32	10^{-4}	0.0523	0.4737	0.3676	
Kumar and Mehra (2005)	64	10^{-4}	0.0117	0.2127	0.1550	
Kumar and Mehra (2005)	128	10^{-4}	0.0030	0.0503	0.0422	

The sinusoidal disturbance solution of the Burgers' equation has been studied as the second test problem. All the results for this test problem have been obtained using method I. A comparison with an earlier study has also been given. Even larger time steps has been used during the solution process, DQM has produced more accurate results than wavelet-Taylor Galerkin method.

Consequently, it can be said that DQM based on cubic B-splines generates very accurate numerical solutions for Burgers' equation. At the same time, the easiness of the implementation of DQMs based on cubic B-splines and low memory storage can be counted as advantages of this method. Even though there exists no explicit formulation for the proposed method, its validity and accuracy has been verified without grid refinement or using various grid distributions to increase the convergence. The only disadvantage proposed methods is that the cost of computation of time integration increases due to the structure of the coefficient matrix occurred after spatial discretization is completed via DQM. This disadvantage is general property of DQMs.

References

Bellman, R., Kashef, B.G. and Casti, J. (1972), "Differential quadrature: a technique for the rapid solution of nonlinear differential equations", *Journal of Computational Physics*, Vol. 10, pp. 40-52.

Bellman, R., Kashef, B.G., Lee, E.S. and Vasudevan, R. (1976), *Differential Quadrature and Splines, Computers and Mathematics with Applications*, Pergamon, Oxford, pp. 371-6.

Bonzani, I. (1997), "Solution of non-linear evolution problems by parallelized collocation-interpolation methods", *Computers & Mathematics and Applications*, Vol. 34 No. 12, pp. 71-9.

Burgers, J.M. (1948), "A mathematical model illustrating the theory of turbulence", *Adv. in Applied Mechanics*, Vol. 1, pp. 171-9.

Caldwell, J. and Smith, P. (1982), "Solution of Burgers' equation with a large Reynolds number", *Applied Mathematical Modelling*, Vol. 6, pp. 381-5.

Cheng, J., Wang, B. and Du, S. (2005), "A theoretical analysis of piezoelectric/composite laminate with larger-amplitude deflection effect, part II: Hermite differential quadrature method and application", *International Journal of Solids and Structures*, Vol. 42, pp. 6181-201.

Christie, I., Griffiths, D.F., Mitchell, A.R. and Sanz-Serna, J.M. (1981), "Product approximation for nonlinear problems in the finite element method", *IMA J. Num. Anal.*, Vol. 1, pp. 253-66.

Civalek, Ö. (2006), "Harmonic differential quadrature-finite differences coupled approaches for geometrically nonlinear static and dynamic analysis of rectangular plates on elastic foundation", *Journal of Sound and Vibration*, Vol. 294, pp. 966-80.

Cole, J.D. (1951), "On a quasi-linear parabolic equation in aerodynamics", *Quarterly of Applied Math.*, Vol. 9, pp. 225-36.

Dehghan, M. (2006), "Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices", *Mathematics and Computers in Simulation*, Vol. 71, pp. 16-30.

Dehghan, M. and Lakestani, M. (2008), "Numerical solution of nonlinear system of second-order boundary value problems using cubic B-spline scaling functions", *International Journal of Computer Mathematics*, Vol. 85 No. 9, pp. 1455-61.

Dehghan, M. and Salehi, R. (2010), "The Chebyshev spectral viscosity method for the time dependent Eikonal equation", *Mathematical and Computer Modelling*, Vol. 52, pp. 70-86.

Dehghan, M. and Shokri, A. (2008), "A numerical method for solution of the two-dimensional sine-Gordon equation using the radial basis functions", *Mathematics and Computers in Simulation*, Vol. 79, pp. 700-15.

Dehghan, M., Hamidi, A. and Shakourifar, M. (2007), "The solution of coupled Burgers equations using Adomian-Pade technique", *Applied Mathematics and Computation*, Vol. 189, pp. 1034-47.

Guo, Q. and Zhong, H. (2004), "Non-linear vibration analysis of beams by a spline-based differential quadrature method", *Journal of Sound and Vibration*, Vol. 269, pp. 413-20.

Herbst, B.M., Schoombie, S.W. and Mitchell, A.R. (1982), "A moving Petrov-Galerkin method for transport equations", *International Journal for Numerical Methods in Engineering*, Vol. 18, pp. 1321-36.

Hopf, E. (1950), "The partial differential equation $U_t+UU_x=\mu U_{xx}$", *Comm. Pure App. Math.*, Vol. 3, pp. 201-30.

Jain, M.K. (1983), *Numerical Solution of Differential Equations*, 2nd ed., Wiley, New York, NY.

Korkmaz, A. (2010), "Numerical algorithms for solutions of Korteweg-de Vries equation", *Numerical Methods for Partial Differential Equations*, Vol. 26 No. 6, pp. 1504-21.

Korkmaz, A. and Dağ, İ. (2008), "A differential quadrature algorithm for simulations of nonlinear Schrödinger equation", *Computers & Mathematics with Applications*, Vol. 56 No. 9, pp. 2222-34.

Korkmaz, A. and Dağ, İ. (2009a), "A differential quadrature algorithm for nonlinear Schrödinger equation", *Nonlinear Dynamics*, Vol. 56 Nos 1/2, pp. 69-83.

Korkmaz, A. and Dağ, İ. (2009b), "Crank-Nicolson – differential quadrature algorithms for the Kawahara equation", *Chaos, Solitons & Fractals*, Vol. 42 No. 1, pp. 65-73.

Korkmaz, A. and Dağ, İ. (2009c), "Solitary wave simulations of complex modified Korteweg-de Vries equation using differential quadrature method", *Computers Physics Communications*, Vol. 180 No. 9, pp. 1516-23.

Korkmaz, A. and Dağ, İ. (2011a), "Numerical simulations of complex modified KdV equation using polynomial differential quadrature method", *J. of Mathematics and Statistics*, Vol. 10, W11, pp. 1-13.

Korkmaz, A. and Dağ, İ. (2011b), "Polynomial based differential quadrature method for numerical solution of nonlinear Burgers' equation", *Journal of the Franklin Institute*, Vol. 348 No. 10, pp. 2863-75.

Korkmaz, A. and Dağ, İ. (2011c), "Shock wave simulations using Sinc differential quadrature method", *Engineering Computations: International Journal for Computer-Aided Engineering and Software*, Vol. 28 No. 6, pp. 654-74.

Korkmaz, A., Dağ, İ. and Saka, B. (2010), "Cosine expansion based differential quadrature (CDQ) algorithms for numerical solution of the RLW equation", *Numerical Methods for Partial Differential Equations*, Vol. 26 No. 3, pp. 544-60.

Kumar, B.V.R. and Mehra, M. (2005), "Wavelet-Taylor Galerkin method for the Burgers' equations", *BIT Numerical Mathematics*, Vol. 45, pp. 543-60.

Lakestani, M. and Dehghan, M. (2009), "Numerical solution of Fokker-Planck equation using the cubic B-spline scaling functions", *Numerical Methods for Partial Differential Equations*, Vol. 25, pp. 418-29.

Lakestani, M. and Dehghan, M. (2012), "Numerical solutions of the generalized Kuramoto-Sivashinsky equation using B-spline functions", *Applied Mathematical Modelling*, Vol. 36, pp. 605-17.

Lee, T.S., Hu, G.S. and Shu, C. (2004), "Application of GDQ method for study of mixed convection in horizontal eccentric annuli", *International Journal of Computational Fluid Dynamics*, Vol. 18 No. 1, pp. 71-9.

Lighthill, M.J. (1956), "Viscosity effects in sound waves of finite amplitude", in Batchlor, G.K. and Davies, R.M. (Eds), *Surveys in Mechanics*, Cambridge University Press, Cambridge, pp. 250-351, 2 plates.

Malekzadeh, P. and Karami, G. (2005), "Polynomial and harmonic differential quadrature methods for free vibration of variable thickness thick skew plates", *Engineering Structures*, Vol. 27, pp. 1563-74.

Mansell, G., Merryfield, W., Shizgal, B. and Weinert, U. (1993), "A comparison of differential quadrature methods for the solution of partial differential equations", *Computer Methods in Applied Mechanics and Engineering*, Vol. 104 No. 3, pp. 295-316.

Miller, E.L. (1966), "Predictor-corrector studies of Burgers' model of turbulent flow", MS thesis, University of Delaware, Newark, DE.

Mittal, R.C. and Jiwari, R. (2009a), "Differential quadrature method for two-dimensional Burgers' equations", *International Journal for Computational Methods in Engineering Science and Mechanics*, Vol. 10 No. 6, pp. 450-9.

Mittal, R.C. and Jiwari, R. (2009b), "Numerical study of Burger-Huxley equation by differential quadrature method", *Int. J. of Appl. Math. and Mech.*, Vol. 5 No. 8, pp. 1-9.

Mittal, R.C. and Jiwari, R. (2011a), "A higher order numerical scheme for some nonlinear differential equations: models in biology", *International Journal of Computational Methods in Engineering Science and Mechanics*, Vol. 12 No. 3, pp. 134-40.

Mittal, R.C. and Jiwari, R. (2011b), "Numerical study of two-dimensional reaction-diffusion Brusselator system by differential quadrature method", *International Journal of Computational Methods in Engineering Science and Mechanics*, Vol. 12 No. 1, pp. 14-25.

Nguyen, H. and Reynen, J. (1982), "A space-time finite element approach to Burgers' equation", in Taylor, C., Hinton, E., Owen, D.R.J. and Onate, E. (Eds), *Numerical Methods for Non-linear Problems*, Vol. 2, Pineridge Publisher, Swansea, pp. 718-28.

Pospelov, L.A. (1966), "Propagation of finite-amplitude elastic waves", *Soviet Phys. Acoust.*, Vol. 11, pp. 302-4.

Prenter, P.M. (1975), *Splines and Variational Methods*, Wiley, New York, NY.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992), *Numerical Recipes in C*, Cambridge University Press, New York, NY, 1988.

Quan, J.R. and Chang, C.T. (1989a), "New sightings in involving distributed system equations by the quadrature methods-I", *Analysis, Comput. Chem. Engrg*, Vol. 13, pp. 779-88.

Quan, J.R. and Chang, C.T. (1989b), "New sightings in involving distributed system equations by the quadrature methods-II. Numerical experiments", *Comput. Chem. Engrg*, Vol. 13, pp. 1017-24.

Saadatmandi, A. and Dehghan, M. (2011), "A tau approach for solution of the space fractional diffusion equation", *Computers & Mathematics with Applications*, Vol. 62 No. 3, pp. 1135-42.

Saka, B., Şahin, A. and Dağ, İ. (2009), "B-spline collocation algorithms for numerical solutions of the RLW equation", *Numerical Methods for Partial Differential Equations*, Vol. 27 No. 3, pp. 581-607.

Saka, B., Dağ, İ., Dereli, Y. and Korkmaz, A. (2008), "Three different methods for numerical solutions of the EW equation", *Engineering Analysis with Boundary Elements*, Vol. 32 No. 7, pp. 556-66.

Shu, C. (2000), *Differential Quadrature and Its Application in Engineering*, Springer, London.

Shu, C. and Richards, B.E. (1992), "Application of generalized differential quadrature to solve two dimensional incompressible Navier-Stokes equations", *Inter. Jour. for Num. Meth. in Fluids*, Vol. 15, pp. 791-8.

Shu, C. and Wu, Y.L. (2002), "Development of RBF-DQ method for derivative approximation and its application to simulate natural convection in concentric annuli", *Computational Mechanics*, Vol. 29, pp. 477-85.

Tomasiello, S. (2003), "Stability and accuracy of the iterative differential quadrature method", *International Journal for Numerical Methods in Engineering*, Vol. 58 No. 6, pp. 1277-96.

Tomasiello, S. (2010), "Numerical solutions of the Burgers-Huxley equation by the IDQ method", *International Journal of Computer Mathematics*, Vol. 87 No. 1, pp. 129-40.

Tomasiello, S. (2011), "Numerical stability of DQ solutions of wave problems", *Numerical Algorithms*, Vol. 57 No. 3, pp. 289-312.

van der Pol, B. (1951), "On a non-linear partial differential equation satisfied by the logarithm of the Jacobian theta-functions, with arithmetical applications", *Proc. Acad. Sci. Amsterdam*, Vol. 13, pp. 261-71.

Zhong, H. (2004), "Spline-based differential quadrature for fourth order equations and its application to Kirchhoff plates", *Applied Mathematical Modelling*, Vol. 28, pp. 353-66.

Zhong, H. and Lan, M. (2006), "Solution of nonlinear initial-value problems by the spline-based differential quadrature method", *Journal of Sound and Vibration*, Vol. 296, pp. 908-18.

Zhou, S. and Cheng, X. (2011), "A linearly semi-implicit compact scheme for the Burgers-Huxley equation", *International Journal of Computer Mathematics*, Vol. 88 No. 4.

Corresponding author
Alper Korkmaz can be contacted at: alperkorkmaz7@gmail.com