Maryam Khazaei
University of Colorado Denver
Master of Science Project
Spring 2018

Topic:

"Numerical solution of two dimensional coupled viscous Burgers equation using modified cubic B-spline differential quadrature method" by H. S. Shukla , Mohammad Tamsir,[1] Vineet K. Srivastava, and Jai Kumar,

Citation: AIP Advances 4, 117134 (2014);
doi: 10.1063/1.4902507

Abstract:

Numerical solution of Burgers equation is a natural first step towards developing methods for the computation of complex flows. It also has become customary to test new approaches in computational fluid dynamics by implementing them for Burgers equation. Burgers equation is also a simple model for understanding various complex physical flows.

In this project, I've studied the modified cubic B-spline differential quadrature method (MCB-DQM) that the above authors used to solve the two-dimensional nonlinear coupled viscous Burgers equation. In MCB-DQM, the modified cubic B-spline is the basis function, and the spatial derivatives of a function are approximated using the weighted sum of the functional values the certain discrete points. In DQM, the weighting coefficients may be determined using several kinds of test functions such as spline function, sinc function, Lagrange interpolation polynomials etc. Then the coupled Burgers equation is reduced into a system of ordinary differential equations. An optimal and fourth-order strongly stable Runge–Kutta scheme is used for solving the ODE. Comparing the result from MCB-DQM and the exact or other numerical solutions in two examples, the efficiency and reliability of the method for solving this kind of two dimensional nonlinear PDE is obtained.

Chapter 1:

Introduction and basic definitions

We consider the well known two dimensional unsteady coupled viscous Burgers equation of the form

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \qquad (1.1)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \qquad (1.2)$$

With initial conditions:

$$u(x, y, 0) = \psi_1(x, y)\ ; \quad (x, y) \in \Omega \quad \text{and} \quad v(x, y, 0) = \psi_2(x, y)\ ; \quad (x, y) \in \Omega \qquad (1.3)$$

and Dirichlet boundary conditions:

$$u(x, y, t) = \xi(x, y, t)\ ; \quad \text{and} \quad v(x, y, t) = \zeta(x, y, t)\ ; \ (x, y) \in \partial\Omega,\ t > 0 \qquad (1.4)$$

where the computational domain is $\Omega = \{(x, y): a \le x \le b, c \le x \le d\}$ and $\partial\Omega$ is its boundary,

$u(x, y, t)$ and $v(x, y, t)$ are the velocity components to be determined, ψ_1, ψ_2, ξ and ζ are known functions. The unsteady term is $\frac{\partial u}{\partial t}$ and etc. The nonlinear convection term is $u\frac{\partial u}{\partial x}$ and etc. The diffusion term is $\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$ and etc. Re is the Reynolds number.

Many problems can be modelled by the Burgers equation [1]. For example, the Burgers equation can be considered as an approach to the Navier-Stokes equations [2, 3] since both contain nonlinear terms of the type: unknown functions multiplied by a first derivative and both contain higher-order terms multiplied by a small parameter. The Burgers equation is one of the very few nonlinear partial differential equation which can be solved exactly. The study of the general properties of the Burgers equation has motivated considerable attention due to its applications in field as diverse as number theory, gas dynamics, heat conduction, elasticity, etc. Analytic solution of two dimensional coupled Burgers equations was first given by Fletcher [4] using the Hopf-Cole transformation. Numerical solution of the coupled Burgers equations is solved by many researchers. In recent years, researchers [5–9] proposed variant of differential quadrature method for the numerical solution of one and two dimensional linear and nonlinear differential equations. Korkmaz and Dag [10, 11] proposed cubic B-spline and sinc differential quadrature methods. Arora and Singh [12] proposed modified cubic B-spline differential quadrature method

(MCB-DQM) and applied on one dimensional Burgers equation to check its efficiency and accuracy.

Definitions

Definition: Spline

The theory of spline functions is a very attractive field of approximation theory. Usually, a spline is a piecewise polynomial function defined in region D, such that there exists a decomposition of D into sub regions in each of which the function is a polynomial of some degree k.

The term "spline" is used to refer to a wide class of functions that are used in applications requiring data interpolation and/or smoothing. A function S(x) is a spline of degree k on [a, b] if

- S ∈ C^{k-1} [a, b].
- a = $t_0 < t_1 < \cdots < t_n$ = b

and

- $$S(x)=\begin{cases} S_0(x), & t_0 < x < t_1 \\ S_1(x), & t_1 < x < t_2 \\ \dots\dots & \\ S_{n-1}(x), & t_{n-1} < x < t_n \end{cases} \qquad \text{where} \quad S_i(x) \in P^k \;, i = 0, 1, \dots, n-1 \qquad (1.5)$$

Definition: B-Spline

The B-spline of degree k is denoted by $\varphi_i^{\,k}$(x), where i ∈ Z, and then we have the following properties:

1. Support ($\varphi_i^{\,k}$) = $[x_i, x_{i+k+1}]$.
2. $\varphi_i^{\,k}$(x)≥ 0, ∀x∈ R (non-negativity).
3. $\sum_{i=-\infty}^{\infty} \varphi_i^{\,k}(x) = 1$, ∀x∈ R (partition of unity). (1.6)

Alternative approach to drive the B-spline relations:

We consider equally-spaced knots of a partition π : a = $x_0 < x_1 < \dots < x_n$ on [a, b]. The alternative approach for deriving the B-splines is more applicable with respect to the recurrence relation for the formulations of B-splines of higher degrees. At first, we recall that the kth forward difference f(x_0) of a given function f(x) at x_0, which is defined recursively by the following [13], [14]:

$\Delta f(x_0) = f(x_1) - f(x_0)$

$\Delta^{k+1} f(x_0) = \Delta^k f(x_1) - \Delta^k f(x_0)$ (1.7)

Definition: The function $(\mathbf{x}-\mathbf{t})^{\boldsymbol{m}}{}_+$

$$(\mathrm{x}-\mathrm{t})^{m}{}_{+} = \begin{cases} (\mathrm{x}-\mathrm{t})^m & \mathrm{x} \geq \mathrm{t} \\ 0 & \mathrm{x} < \mathrm{t} \end{cases} \qquad (1.8)$$

It is clear that $(\mathrm{x}-\mathrm{t})^{m}{}_{+}$ is (m−1) times continuously differentiable with respect to t and x.

The B-spline of order m is defined as follows:

$$\varphi_i^m(t) = \frac{1}{h^m} \sum_{j=0}^{m+1} \binom{m+1}{j} (-1)^{m+1-j} (x_{i-2+j} - t)_+^m = \frac{1}{h^m} \Delta^{m+1} (x_{i-2} - t)_+^m \qquad (1.9)$$

Hence, we can obtain the B-spline of various orders by taking various values of m. let m = 1; thus,

m = 1 → $\varphi_i^1(t) = \frac{1}{h}\Delta \quad (x_{i-2} - t)_+^1 = \frac{1}{h}[\, (x_{i-2} - t)_+ - 2(x_{i-1} - t)_+ + (x_i - t)_+]$

$$\varphi_i^1(t) = \frac{1}{h} \begin{cases} (x_i - t) - 2(x_{i-1} - t) & x_{i-2} < t \leq x_{i-1} \\ (x_i - t) & x_{i-1} < t \leq x_i \\ 0 & otherwise \end{cases}$$

(1.10)

By considering different value for m, we can get different degree of B-Spline.

Definition: Differential Quadrature

Consider the nonlinear first-order partial differential equation

u_t = g(t, x, u, u_x), $\quad -\infty < x < \infty, \quad 0 < t$ (1.11)

with initial condition

u(0, x) = h(x), (1.12)

an equation arising in many mathematical models of physical processes. Let us make the assumption such that the function u satisfying equation (1.11) and (1.12) is sufficiently smooth to allow us to write the approximate relation

$u_x(t, x_i) \cong \sum_{j=1}^{N} a_{ij}\, u(t, x_j)$, i= 1, 2, …., N (1.13)

One method for determining the coefficients a_{ij} will be discussed below. Substitution of equation (3) into equation (1) yields the set of N ordinary differential equations

u_t (t, x_i) ≅ g(t, x_i, $u(t, x_i)$, $\sum_{j=1}^{N} a_{ij}\, u(t, x_j)$), (1.14)

With initial conditions

u(0. x_i) = h(x_i), i= 1, 2, …, N (1.15)

Hence, under the assumption that (1.13) is valid, we have succeeded in reducing the task of solving equation (1.11) to the task of solving a set of ordinary differential equations with prescribed initial values. Considering the differential quadrature technique in practice turns out that relativity low order differential quadrature is all that is needed, the total amount of storage and time required on the machine is thus quite low.

Definition: Diagonally dominant

In mathematics, a square matrix is said to be diagonally dominant if for every row of the matrix, the magnitude of the diagonal entry in a row is larger than or equal to the sum of the magnitudes of all the other (non-diagonal) entries in that row. More precisely, the matrix A is diagonally dominant if for all i,

$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$ (1.16)

where a_{ij} denotes the entry in the ith row and jth column.

Chapter 2

Method:

Modified Cubic Spline Differential Quadrature Method:

In 1972, Bellman et al.[19] introduced differential quadrature method (DQM). This method approximates the spatial derivatives of a function using the weighted sum of the functional values at the certain discrete points. The weighting coefficients in DQM are determined using several kinds of test functions such as spline function, sinc function, etc. This section revisits the MCB-DQM [12, 16] in order to complete our problem in two-space dimension. It is assumed that M and N grid points: a = $x_1 < x_2, \ldots < x_M$ = b and c = $y_1 < y_2, \ldots < y_N$ = d are uniformly distributed with spatial step size Δx = $x_{i+1} - x_i$ and Δy = $y_{i+1} - y_i$ in x and y directions, respectively. The first and second order spatial partial derivatives of u(x, y, t)with respect to x (keeping y_j as fixed) and with respect to y (keeping x_i as fixed), approximated at x_i and y_j, respectively, are defined as:

$$\frac{\partial \mathrm{u}(x_i, y_j, \mathrm{t})}{\partial \mathrm{x}} = \sum_{k=1}^{M} w_{ik}^{(1)} \ \mathrm{u}(x_k, \ y_j, \mathrm{t}), \quad \text{i =1, 2, ..., M, j=1, ..., N} \qquad (2.1)$$

$$\frac{\partial^2 \mathrm{u}(x_i, y_j, \mathrm{t})}{\partial x^2} = \sum_{k=1}^{M} w_{ik}^{(2)} \ \mathrm{u}(x_k, \ y_j, \mathrm{t}), \quad \text{i =1, 2, ..., M, j=1, ..., N} \qquad (2.2)$$

$$\frac{\partial \mathrm{u}(x_i, y_j, \mathrm{t})}{\partial \mathrm{y}} = \sum_{k=1}^{N} \overline{w_{jk}^{(1)}} \ \mathrm{u}(x_i, \ y_k, \mathrm{t}), \quad \text{i =1, 2, ..., M, j=1, ..., N} \qquad (2.3)$$

$$\frac{\partial^2 \mathrm{u}(x_i, y_j, \mathrm{t})}{\partial y^2} = \sum_{k=1}^{N} \overline{w_{jk}^{(2)}} \ \mathrm{u}(x_i, \ y_k, \mathrm{t}), \quad \text{i =1, 2, ..., M, j=1, ..., N} \qquad (2.4)$$

Likely, we approximate the first and second order spatial partial derivatives of v(x, y, t) with respect to x and with respect to y.

Here $w_{ij}^{(r)}$ and $\overline{w_{ij}^{(r)}}$, r = 1,2 are the weighting coefficients of the rth order spatial partial derivatives with respect to x and y. The cubic B-spline basis functions [9] at the knots are defined as:

$$\varphi_m(x) = \frac{1}{(\triangle x)^3}\begin{cases} (x-x_{m-2})^3 & x_{m-2} < t \le x_{m-1} \\ (x-x_{m-2})^3 - 4(x-x_{m-1})^3 & x_{m-1} < t \le x_m \\ (x_{m+2}-x)^3 - 4(x_{m+1}-x)^3 & x_{m-1} < t \le x_m \\ (x_{m+2}-x)^3 & x_{m+1} < t \le x_{m+2} \\ 0 & otherwise \end{cases} \quad (2.5)$$

Note that φ and φ are distinct functions. The cubic B-Spline basis functions are modified in such way that the resulting matrix system of equations is diagonally dominant. The modified cubic B-Spline basis functions at the knots are modified as followed [17]:

$\phi_1(x)= \varphi_1(x) + 2\varphi_0(x)$

$\phi_2(x)= \varphi_2(x) - \varphi_0(x)$

$\phi_m(x)= \varphi_m$ (x) m = 3, 4, ..., N-2

$\phi_{N-1}(x)= \varphi_{N-1}$ (x) - φ_{N+1} (x)

ϕ_N (x)= φ_N + 2φ_{N+1} (x) (2.6)

Where $\{\phi_1, \phi_2, \ldots, \phi_N\}$ forms a basis over the domain Ω = {(x, y): a ≤ x ≤ b; c ≤ y ≤ d}.

In Equation (2.1), substituting the values of ϕ_m(x), m = 1,2, ..., N, we get a system of linear equations:

$$\phi'_m(x_i) = \sum_{k=1}^{M} w_{ik}^{(1)} \phi_m(x_k), \quad i=1, 2, \ldots, M \quad (2.7)$$

TABLE I. Coefficients of the cubic B-spline φ_m and its derivatives at the node x_m.

	x_{m-2}	x_{m-1}	x_m	x_{m+1}	x_{m+2}
$\varphi_m(x)$	0	1	4	1	0
$\varphi_m{}'$	0	3/h	0	-3/h	0
$\varphi_m{}''$	0	$6/h^2$	$-12/h^2$	$6/h^2$	0

With the help of Equation (2.5), (2.6) and table I, Equation (2.7) reduces into a tridiagonal system of equations:

$AW^{(1)}[i] = R[i]$, for i = 1,2,...,M, (2.8)

Where the weighting coefficient vector corresponding to x_i is $W^{(1)}$[i]= [$w_{i1}^{(1)}, w_{i2}^{(1)}, \dots, w_{iN}^{(1)}$] and R[i] = [$\phi'_{1,i}, \phi'_{2,i}, \dots, \phi'_{N-1,i}, \phi'_{N,i}$] where we define $\phi_{s,i} = \phi_s(x_i), s = 1,2, \dots, N$ *and* $i = 1, 2, \dots, M$. Then the coefficient matrix A has the form

$$A = \begin{bmatrix} \phi_{1,1} & \phi_{1,1} & 0 & 0 & & \cdots & 0 \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} & 0 & & \cdots & 0 \\ & \phi_{3,2} & \phi_{3,3} & \phi_{3,4} & 0 & & \\ & \ddots & \ddots & & \ddots & & \\ & & \phi_{N-1,N-2} & \phi_{N-1,N-1} & \phi_{N-1,N} & & \\ & & & \phi_{N,N-1} & & \phi_{N,N} & \end{bmatrix} \tag{2.9}$$

The coefficient matrix A is invertible. Using the Thomas algorithm, the tridiagonal system of linear equations (2.12) is solved for each i, which gives the weighting coefficients $w_{ik}^{(1)}$ of the first order partial derivative. Similarly, the weighting coefficients $w_{ij}^{(2)}$, 1 ≤i, j ≤ N are determined. Weighting coefficients $w_{ij}^{(2)}$, 1 ≤i, j ≤ N, can be computed as: [9]

$$w_{ij}^{(r)} = r\left(w_{ij}^{(1)} w_{ii}^{(r-1)} - \frac{w_{ij}^{(r-1)}}{x_i - x_j}\right), \text{ , for i} \neq \text{j and i = 1,2,3,...,N; r = 2,3,...,N} - 1 \tag{2.10}$$

$$w_{ii}^{(r)} = -\sum_{j=1,\ i \neq j}^{N} w_{ij}^{(r)}, \quad \text{for i=j} \tag{2.11}$$

where $w_{ij}^{(r-1)}$ and $w_{ij}^{(r)}$ are the weighting coefficients of the (r−1)th and rth order partial derivatives with respect to x. In the similar way, the weighting coefficients $\overline{w_{jk}^{(1)}}$ of the first order partial derivatives with respect to y and weighting coefficients $\overline{w_{ij}^{(2)}}$,1 ≤i,j ≤ N for the second derivatives can be computed.

Chapter 3

MCB-DQM for two – dimensional coupled Burger equation

On substituting the approximated values of spatial derivatives computed by MCB-DQM, Equation (1.1) can be written as:

$$\frac{\partial \mathrm{u}(x_i, y_j, \mathrm{t})}{\partial \mathrm{t}} = -\mathrm{u}(x_i, y_j, t) \sum_{k=1}^{M} w_{ik}^{(1)} \ \mathrm{u}(x_k,\ y_j, t) - \mathrm{v}(x_i,\ y_j, t) \sum_{k=1}^{N} \overline{w_{jk}^{(1)}} \ \mathrm{u}(x_i,\ y_k, t)$$
$$+\frac{1}{Re} \left[\sum_{k=1}^{M} w_{ik}^{(2)} \ \mathrm{u}(x_k,\ y_j, t) - \sum_{k=1}^{N} \overline{w_{jk}^{(2)}} \ \mathrm{u}(x_i,\ y_k, t)\right]$$
$$, (x_i,\ y_j, t) \in \mathrm{R},\ \mathrm{t} > 0,\quad \mathrm{i} = 1, 2, \ldots, \mathrm{M},\ \ \mathrm{j} = 1, 2, \ldots, \mathrm{N}. \tag{3.1}$$

In a similar way,

$$\frac{\partial \mathrm{v}(x_i, y_j, \mathrm{t})}{\partial \mathrm{t}} = -\mathrm{u}(x_i, y_j, t) \sum_{k=1}^{M} w_{ik}^{(1)} \ \mathrm{v}(x_k,\ y_j, t) - \mathrm{v}(x_i,\ y_j, t) \sum_{k=1}^{N} \overline{w_{jk}^{(1)}} \ \mathrm{v}(x_i,\ y_k, t)$$
$$+\frac{1}{Re} \left[\sum_{k=1}^{M} w_{ik}^{(2)} \ \mathrm{v}(x_k,\ y_j, t) - \sum_{k=1}^{N} \overline{w_{jk}^{(2)}} \ \mathrm{v}(x_i,\ y_k, t)\right]$$
$$, (x_i,\ y_j, t) \in \mathrm{R},\ \mathrm{t} > 0,\quad \mathrm{i} = 1, 2, \ldots, \mathrm{M},\ \ \mathrm{j} = 1, 2, \ldots, \mathrm{N}. \tag{3.2}$$

Reducing Equation (3.1) and Equation (3.2) into a system of ordinary differential equations:

$$\frac{\mathrm{du}(x_i, y_j, \mathrm{t})}{\mathrm{dt}} = F_{i,j}(\mathrm{u}(x_1,\ y_1, \mathrm{t}), \mathrm{u}(x_2,\ y_2, \mathrm{t}), \ldots., \mathrm{u}(x_M,\ y_N, \mathrm{t})),\ \mathrm{i} = 1, 2, \ldots, \mathrm{M} \text{ and } \mathrm{j} = 1, 2, \ldots, \mathrm{N}. \tag{3.3}$$

$$\frac{\mathrm{dv}(x_i, y_j, \mathrm{t})}{\mathrm{dt}} = F_{i,j}(\mathrm{v}(x_1,\ y_1, \mathrm{t}), \mathrm{v}(x_2,\ y_2, \mathrm{t}), \ldots., \mathrm{v}(x_M,\ y_N, \mathrm{t})),\ ,\ \mathrm{i} = 1, 2, \ldots, \mathrm{M} \text{ and } \mathrm{j} = 1, 2, \ldots, \mathrm{N}. \tag{3.4}$$

Equation (3.3) and (3.4) together with initial conditions (1.3) and Dirichlet boundary conditions (1.4) are solved by a five-stage and fourth order strong stability-preserving Runge-Kutta method (SSP-RK54 schemes). SSP Runge-Kutta schemes tend to have a kind of excellent properties such as large regions of absolute stability and low storage. [20, 21]

Chapter IV.

Computational Result

We provide MCB-DQM numerical solutions for two dimensional coupled Burgers equation given in the introduction part by considering one problem. The accuracy and consistency of the scheme is measured in terms of error norms L_2 and L_∞, restricted to the points on the grid, defined as:

$$L_2 = \| u_{\text{exact}} - u_{\text{computed}} \|_2 = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} |u_{i,j_{exact}} - u_{i,j_{computed}}|^2}$$

$$L_\infty = \| u_{\text{exact}} - u_{\text{computed}} \|_\infty = max_{i,j}|u_{i,j_{exact}} - u_{i,j_{computed}}| \quad (4.1)$$

We represent u_{exact} and u_{computed} as the exact (or other authors" computed) and computed solutions respectively. Numerical solutions to Equation (1.1) and (1.2) are tested for the following problem.

Problem:

The analytical solutions of Equation (1.1) and (1.2) can be generated as:[4]

$$u(x, y, t) = \frac{3}{4} - \frac{1}{4[1+\exp((-4x+4y-t)Re/32)]}$$

$$v(x, y, t) = \frac{3}{4} + \frac{1}{4[1+\exp((-4x+4y-t)Re/32)]} \quad (4.2)$$

The computational domain can be considered by the square domain Ω = {(x, y): 0 ≤ x ≤ 1, 0 ≤ y ≤ 1}, and the initial and boundary conditions for u(x, y, t) and v(x, y, t) are taken from the analytical solutions (4.2). For the test problem I, we have taken a grid size 20×20 with time step Δt = 0.0001 and Re = 100. Computed and exact values of u are shown in Tables II along with the results given by Srivastava et al. [18, 15] and Bahadir [1] at some typical grid point. The results show that the proposed scheme produces more accurate results than Bahadir, but of unknown efficiency. [1] Tables IV and V show the errors L_2 and L_∞ ,and also the rate of convergence of u and v components, respectively, at Re = 100, t = 1.0 for Δt = 0.0001.

It can be observed from tables IV and V, that MCB-DQM performs better than Srivastava et al.[15] and gives more than quadratic rate of convergence. Fig. 1 shows the computed MCB-DQM solutions of u and v for Re = 100 at t =0.5 and Fig. 2 shows analytical solutions of u and v respectively.

TABLE II. Comparison of MCB-DQM and exact solutions of u for Re = 100, 20×20 grid and Δt = 0.0001.

	t = 0.5					t = 2.0				
Grid (x, y)	MCB-DQM	Exact	I-LFDM[14]	Expo-FDM[15]	Bahadir[9]	MCB-DQM	Exact	I-LFDM[14]	Expo-FDM[15]	Bahadir[9]
(0.1,0.1)	0.54412	0.54332	0.54300	0.54300	0.54235	0.50050	0.50048	0.50047	0.50047	0.49983
(0.5,0.1)	0.50037	0.50035	0.50034	0.50034	0.49964	0.50000	0.50000	0.50000	0.50000	0.49930
(0.9,0.1)	0.50000	0.50000	0.50000	0.50000	0.49931	0.50000	0.50000	0.50000	0.50000	0.49930
(0.3, 0.3)	0.54388	0.54332	0.54269	0.54270	0.54207	0.50050	0.50048	0.50044	0.50044	0.49977
(0.7, 0.3)	0.50037	0.50035	0.50032	0.50032	0.49961	0.50000	0.50000	0.50000	0.50000	0.49930
(0.1, 0.5)	0.74196	0.74221	0.74215	0.74215	0.74130	0.55632	0.55568	0.55515	0.55516	0.55461
(0.5, 0.5)	0.54347	0.54332	0.54251	0.54252	0.54222	0.50050	0.50048	0.50041	0.50041	0.49973
(0.9, 0.5)	0.50035	0.50035	0.50030	0.50030	0.49997	0.50001	0.50000	0.50000	0.50000	0.49931
(0.3, 0.7)	0.74211	0.74221	0.74211	0.74212	0.74146	0.55597	0.55568	0.55482	0.55482	0.55429
(0.7, 0.7)	0.54327	0.54332	0.54246	0.54247	0.54243	0.50054	0.50048	0.50038	0.50038	0.49970
(0.1, 0.9)	0.74994	0.74995	0.74994	0.74994	0.74913	0.74406	0.74426	0.74420	0.74420	0.74340
(0.5, 0.9)	0.74219	0.74221	0.74210	0.74210	0.74201	0.55575	0.55568	0.55450	0.55451	0.55413
(0.9, 0.9)	0.54333	0.54332	0.54228	0.54229	0.54232	0.50052	0.50048	0.50053	0.50053	0.50001

TABLE IV. Errors and rate of convergence for u-component for Re = 100, Δt = 0.0001 at t = 1.0.

	L_2			L_∞		
		MCB-DQM			MCB-DQM	
Grid	Expo-FDM[15]		Rate	Expo-FDM[15]		Rate
4×4	8.5708e-002	1.6388e-002	-	9.7046e-002	2.8788e-003	
8×8	4.9429e-002	1.9286e-003	3.0875	4.6886e-002	1.9572e-004	3.8786
16×16	1.9192e-002	3.9474e-004	2.2881	2.0467e-002	2.0486e-005	3.2561
32×32	8.6812e-003	8.1181e-005	2.2817	9.0744e-003	2.2202e-006	3.2059
64×64	-	1.5322e-005	2.4055	-	2.1838e-007	3.3458

TABLE V. Errors and rate of convergence for v-component for Re = 100, Δt = 0.0001 at t = 1.0.

	L_2			L_∞		
		MCB-DQM			MCB-DQM	
Grid	Expo-FDM[15]		Rate	Expo-FDM[15]		Rate
4×4	8.5708e-002	1.6388e-002	-	9.7046e-002	2.8788e-003	-
8×8	4.9431e-002	1.9286e-003	3.0875	4.6887e-002	1.9573e-004	3.8786
16×16	1.9196e-002	3.9474e-004	2.2881	2.0471e-002	2.0486e-005	3.2561
32×32	8.6878e-003	8.1181e-005	2.2817	9.0813e-003	2.2202e-006	3.2059
64×64	-	1.5322e-005	2.4055	-	2.1838e-007	3.3458

FIG. 1. Numerical solution for u (left) and v (right) components at t = 0.5 with Δt = 0.0001, Re = 100 and grid size 20×20 for the test problem 1.

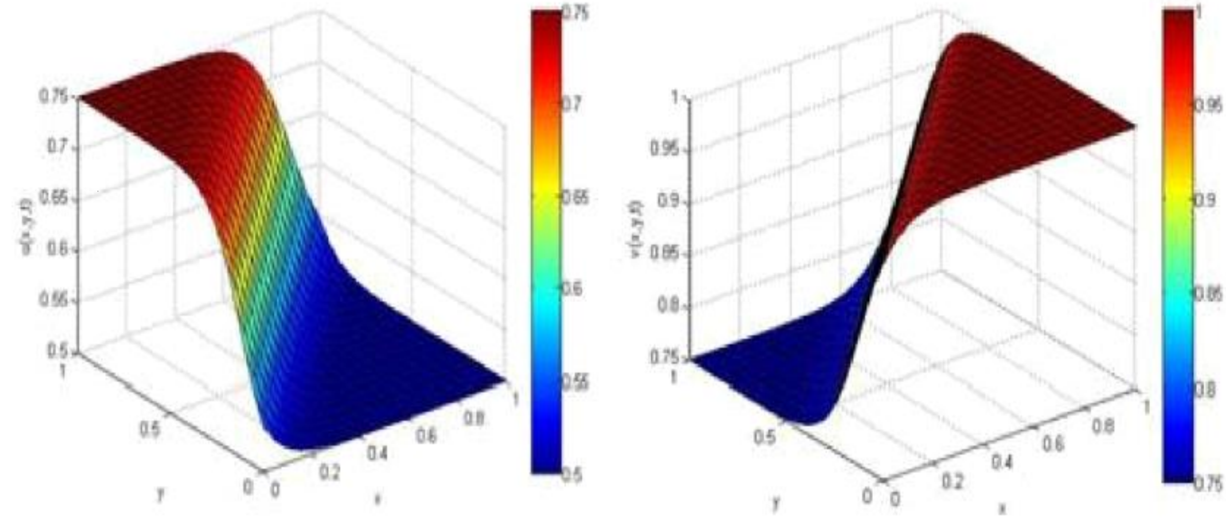
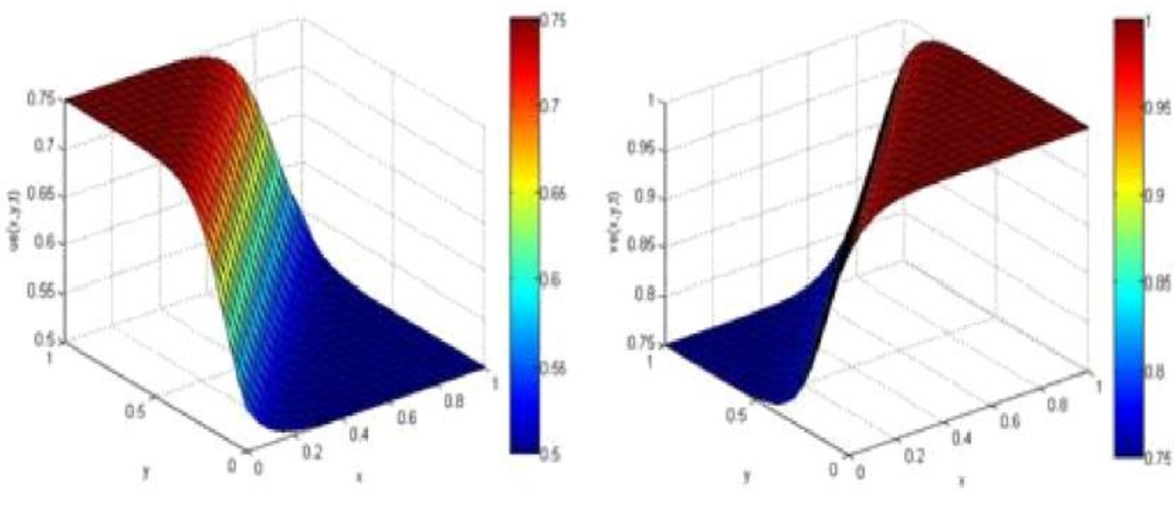


FIG. 2. Exact solution for u (left) and v (right) components at t = 0.5 with Δt = 0.0001, Re = 100 and 20×20 grid for the test problem I.



V. Conclusion:

A modified cubic B-spline differential quadrature method is studied for the numerical solution of two dimensional nonlinear coupled viscous Burgers equations.

In the first chapter, some basic definitions and methods are given. In the second chapter the procedure of Modified Cubic Spline Differential Quadrature Method is discussed. Numerical study shows that MCB-DQM results are in good agreement with the exact solutions. Comparing the error norms obtained by this scheme and exponential finite difference scheme provided by Srivastava [15], this method error is better than those obtained by exponential finite difference scheme. Further, it can also be noticed that the rate of convergence of the described scheme is more than quadratic. Also in order to comparing the stability between MCB – DQM and Numerical solutions of coupled Burgers' equations by an implicit finite-difference scheme [15], Srivastava [15] used implicit first order in time stability that is not that stable than this project stability that is SSP-Runge Kutta scheme. Obtained results show that MCB-DQM is a promising numerical scheme for solving the higher dimensional nonlinear physical problems governed by partial differential equations.

References:

1. A. R. Bahadir,"A fully implicit finite-difference scheme for two dimensional Burgers equation," Appl. Math. Comput. 137, 131 (2003).
2. S. E. Esipov, "Coupled Burgers' equations: a model of poly-dispersive sedimentation," Phys. Rev.52, 3711 (1995).
3. V. K. Srivastava and M. Tamsir, "Crank-Nicolson semi-implicit approach for numerical solutions of two-dimensional coupled nonlinear Burgers' equations," Int. J. Appl. Mech. Eng.17(2), 571 (2012).
4. C. A. J. Fletcher, "Generating exact solutions of the two dimensional Burgers equation,"Int.J.Numer.Meth.Fluids3,213 (1983).
5. R. Bellman, B. G. Kashef, and J. Casti, "Differential quadrature: a technique for the rapid solution of nonlinear differential equations," J. Comput. Phy.10, 40–52 (1972).
6. C. Shu and B. E. Richards, "Application of generalized differential quadrature to solve two dimensional incompressible navier-Stokes equations," Int. J. Numer. Meth. Fluids15, 791–798 (1992).
7. J. R. Quan and C. T. Chang, "New insights in solving distributed system equations by the quadrature methods-I," Comput. Chem. Eng.13, 779–788 (1989).
8. C. Shu and H. Xue, "Explicit computation of weighting coefficients in the harmonic differential quadrature, "J.SoundVib. 204(3), 549–555 (1997).

9. C. Shu, Differential Quadrature and its Application in Engineering (Athenaeum Press Ltd, Great Britain, 2000).
10. A. KorkmazandI. Dag, “Cubic B -spline differential quadrature methods and stability for Burgers equation,” Eng.Comput. Int. J. Comput. Aided Eng. Software 30(3), 320–344 (2013).
11. A. KorkmazandI. Dag, “Shock wave simulations using sinc differential quadrature method, ” Eng. Comput. Int. J. Comput. Aided Eng. Software28(6), 654–674 (2011).
12. G. Arora and B. K. Singh, “Numerical solution of Burgers’ equation with modified cubic B-spline differential quadrature method,” Applied Math. Comput.224(1), 166–177 (2013).
13. Rashidinia, J., and Sharifi, Sh., “Survey of B-spline functions to approximate the solution of mathematical problems”, Iran University of Science and Technology (2011).
14. Printer, P. M, “Splines and variational Methods”, Colorado State University, Wiley Classics Edition published (1975).
15. V. K. Srivastava, S. Singh, and M. K. Awasthi, “Numerical solutions of coupled Burgers’ equations by an implicit finite difference scheme,” AIP Advances 3, 082131 (2013).
16. R. Jiwari and J. Yuan, “A computational modeling of two dimensional reaction–diffusion Brussel at or system arising in chemical processes,” J. Math. Chem.52(6), 1535–1551 (2014).
17. R. C. Mittal, R. K. Jain, Numerical solutions of nonlinear Burgers equation with modified cubic B-Spline collocation method, Appl. Math. Comput. 218 7839-7855 (2012).
18. V.K. Srivastava, M. K. Awasthi, and S. Singh, “An implicit logarithm finite difference technique for two dimensional coupled viscous Burgers’ equation,” AIP Advances 3, 122105 (2013).
19. R. Bellman, B. G. Kashef, and J. Casti, “Differential quadrature: a technique for the rapid solution of nonlinear differential equations,” J. Comput. Phy.10, 40–52 (1972).
20. S. Gottlieb, D. I. Ketcheson, and C. W. Shu, “High Order Strong Stability Preserving Time Discretizations,”J. Sci. Comput. 38, 251–289 (2009).
21. J. F. B. Kraaijevanger, “Contractivity of Runge-Kutta methods,” BIT Numerical Maths.31, 482–528 (1991).