A novel approach for numeric study of 2D biological population model

Brajesh Kumar Singh

Abstract: In the present paper, the modified cubic B-spline differential quadrature method (MCB-DQM) has been implemented for the numerical computation of two-dimensional biological population model (BPM). The method is based on differential quadrature in which the weighting coefficients are computed by using MCB as a set of basis functions. We present three test problems to confirm the efficiency and accuracy of the method for BPM, which shows that the MCB-DQM solutions are in good agreement with the results obtained by the recent schemes: improved element-free Galerkin method by Zhang et al. and element-free kp-Ritz method by Cheng et al. The order of convergence of MCB-DQM for the solutions of BPM is shown to be quadratic.

1. Introduction
The dispersal or emigration plays a significant role in the regulation of population of some species. In the recent years, the biological problems have been received a much attention among the researchers due to its key role in the regulation of population of some species. The diffusion of a biological species in a region $\mathbb{R}^2$ is described by three functions: (i) population density $\rho(x, t)$, (ii) diffusion...
velocity $v(x, t)$ and (iii) population supply $\sigma(x, t)$ (Gurtin & Maccamy, 1977), where $x = (x, y) \in \mathbb{R}^2$ and $t$ is time.

The population density $\rho$ represents the number of individuals per unit volume at $(x, t)$, and it's integral over any sub-region $D$ of region $\mathbb{R}^2$ gives the total population of $D$ at time $t$ whereas $\sigma(x, t)$ denotes the rate at which individuals are supplied in per unit volume at position $x$ at time $t$ by births and deaths. The diffusion velocity $v(x, t)$ represents the average velocity of the individuals occupying the position $x$ at the time $t$, and it represents the flow of population from one point to another point. The entities $\rho$, $v$ and $\sigma$ must be consistent with law of population balance (Gurney & Nisbet, 1975; Lu, 2000), given below in Equation (1): for every regular sub-region $D$ of $\mathbb{R}^2$ at time $t$

$$\frac{d}{dt} \int_D \rho \, dV + \int_D \rho v \cdot n \, dV = \int_D \sigma \, dV$$

where $n$ is the outward unit normal to the boundary $\partial D$ of the region $D$. The law (1) shows that “the sum of the rate of change of population of $D$ and the rate at which the individuals leave the region $D$ across its boundary $\partial D$ is equal to the rate at which the individuals are supplied directly to the region $D$”.

For $\sigma = \sigma(\rho)$ and $v = -\lambda(\rho)\nabla\rho$, where $\lambda(\rho) > 0$ for $\rho > 0$ and $\nabla$ is the Laplace operator, the following nonlinear degenerate parabolic partial differential equation (DE) for $\rho$ is reduced to

$$\frac{\partial \rho}{\partial t} = \nabla^2(\phi(\rho)) + \sigma(\rho),$$

where $\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ and $\phi(\rho)$ be function of $\rho$ (Gurtin & Maccamy, 1977). In Gurney and Nisbet (1975), $\sigma(\rho)$ is used as a special case in the modeling of the population of animals. The movements were made generally either by mature animals driven by mature invaders or by the young animals just reaching maturity moving out of their parental territory to establish breeding territory of their own. In each case, it is much more probable to consider that they will be directed toward nearby vacant territory. Therefore, in this model, the movement will take place almost exclusively down the population density gradient and will be more rapid at high population densities than at low ones. To model this situation, the authors further assumed a walk through a rectangular grid, in which at each step, an animal may either stay at its present location or may move toward the lowest population density (Gurney & Nisbet, 1975). The probability distribution between these two possibilities being determined by the magnitude of the population density gradient at the grid side is concerned. This model is correspond to Equation (2) with $\phi(\rho) = \rho^2$ as

$$\frac{d \rho}{d t} = \nabla^2(\rho^2) + \sigma(\rho).$$

Some properties of Equation (3) such as Holder-estimates and its solutions were studied in Lu (2000). Equation (3) is the generalization of the following lows:

(a) For $\sigma(\rho) = c\rho$, $c$ is a constant (Malthusian Law Gurtin and Maccamy, 1977).

(b) For $\sigma(\rho) = c_1\rho - c_2\rho^2$, where $c_1, c_2 > 0$ are constant (Verhulst Law Gurtin & Maccamy, 1977).

(c) For $\sigma(\rho) = c_3\rho^a$ with $c \geq 0$ and $0 < a < 1$ (Porous media Bear, 1972; Okubo, 1980).

Consider two-dimensional (2D) biological population model (BPM) (3) with $\sigma(\rho) = h\rho^2(1 - \rho^2)$ (Zhang, Deng, & Liew, 2014) as follows:

$$\frac{\partial \rho}{\partial t} = \nabla^2(\rho^2) + h\rho^2(1 - \rho^2), \forall (x, t) \in \partial D \times (0, T),$$

$$\rho(x, 0) = \rho^0(x), \quad \forall x \in D,$$

where $D = [a, b] \times [c, d]$, and $h, \alpha, \beta$ and $r$ are real numbers. The boundary conditions are taken as
\{ \rho(x, t) = \psi(x, t), \quad \forall (x, t) \in \partial D \times (0, T) \}. 

(5)

In the recent years, a lot of numerical techniques have been developed for the numerical computation of time-dependent partial DEs (Arora, Mittal, & Singh, 2014; Arora & Singh, 2013; Cheng, Zhang, & Liew, 2014; Shivanian, 2013; Singh & Kumar, 2016c; Srivastava & Singh, 2014; Zhang et al., 2014). The existence, regularity and uniqueness of the weak solutions for degenerate parabolic equations (see, Aronson, 1986; Dibenedetto, 1993; Giuggioli & Kenkre, 2003; Gurney & Nisbet, 1975; Gurtin & MacCamy, 1977; Jager & Lu, 1997; Lu, 2000). The analytical solutions for time-fractional order population problems were obtained using fractional reduced transform method (Singh & Kumar, 2016a, 2016b; Singh & Srivastava, 2015; Srivastava, Mishra, Kumar, Singh, & Awasthi, 2014) by Srivastava, Kumar, Awasthi, and Singh (2014). The computation of various type of population problems is done by using homotopy perturbation method (HPM) (Liu, Li, & Zhang, 2011; Roul, 2010), homotopy analysis method (HAM) (Arafa, Rida, & Mohamed, 2011), and variational iteration method (Shakeri & Dehghan, 2006). A meshless local radial point interpolation numerical method to simulate a nonlinear partial integro-DE arising in population dynamics is given by Shivanian (2013). Recently, the numerical computation of BPM model (4) is done by Cheng et al. (2014) using element-free kp-Ritz method, and by Zhang et al. (2014) using improved element-free Galerkin method (IEFGM).

Table 1. Comparison of numerical solution of Example 1 at \( x = 0.5, y = 0.5 \)

| \( t = 0.1 \) | \( t = 0.2 \) | \( t = 0.3 \) | \( t = 0.4 \) | \( t = 0.5 \) |
|----------|----------|----------|----------|----------|
| MCB-DQM  | 1.26282  | 1.14254  | 1.03381  | 0.93543  | 0.84641  |
| IEFGM (Zhang et al., 2014) | 1.2574   | 1.1382   | 1.0302   | 0.9325   | 0.8440   |
| Exact    | 1.2628   | 1.1426   | 1.0339   | 0.9355   | 0.8465   |
| CPU(s)   | 0.31     | 0.59     | 0.90     | 1.22     | 1.51     |

Table 2. The errors in Example 1 at different time levels \( t \leq 0.5 \)

| \( M = N \) | \( \mathcal{L}_2 \) | \( \mathcal{L}_\infty \) | \( \mathcal{L}_2 \) | \( \mathcal{L}_\infty \) | \( \mathcal{L}_2 \) | \( \mathcal{L}_\infty \) | \( \mathcal{L}_2 \) | \( \mathcal{L}_\infty \) |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 13       | 2.689E-05 | 1.052E-04 | 1.24E-05 | 0.33    | 0.33    | 0.33    | 0.33    | 0.33    |
| 26       | 1.479E-05 | 1.069E-04 | 1.183E-05| 2.31    | 2.31    | 2.31    | 2.31    | 2.31    |

Table 3. The errors in Example 1 for large time levels \( 5 \leq t \leq 20 \) with \( h_x = h_y = 1/12 \)

| \( t = 5 \) | \( t = 10 \) | \( t = 15 \) | \( t = 20 \) |
|----------|----------|----------|----------|
| \( \mathcal{L}_2 \) | 2.33E-07 | 1.57E-09 | 1.06E-11 | 7.14E-14 |
| \( \mathcal{L}_\infty \) | 3.40E-07 | 2.30E-09 | 1.55E-11 | 1.04E-13 |
| CPU(s)   | 15.17    | 30.31    | 45.34    | 61.43    |

Table 4. Rate of convergence (ROC) of Example 1

| \( M = N \) | \( t = 0.4 \) | \( t = 0.1 \) |
|----------|----------|----------|
| \( \mathcal{L}_2 \) | ROC | CPU(s) | \( \mathcal{L}_2 \) | ROC | CPU(s) |
| 5        | 1.17E-04 | 0.16    | 1.56E-04 | 0.04 |
| 10       | 2.85E-05 | 2.03    | 3.85E-05 | 2.02 |
| 15       | 1.22E-05 | 2.09    | 1.88E-05 | 1.76 |
| 20       | 8.23E-06 | 1.38    | 1.40E-05 | 1.02 |
The main aim of this paper is to implement “modified cubic B-spline differential quadrature method (MCB-DQM)” (Arora & Singh, 2013; Singh & Arora, 2014; Singh & Bianca, 2016) for the numerical computation of 2D BPM. The scheme is based on differential quadrature where the modified cubic B-spline functions are used as set of basis functions to determine the weighting coefficients. The MCB-DQM is used to convert the given system of PDEs into a system of first order ODEs, in time. The resulting system of ODEs is solved using the SSP-RK54 scheme. Three test problems are considered to demonstrate the accuracy and utility of the method. The maximum absolute errors $L_\infty$ and $L_2$ errors in the MCB-DQM solutions have been compared with the errors due to some recent exiting schemes.

The article is organized as follows. In Section 2, the description of the MCB-DQM is given. In Sections 3, procedure for implementation of method is described. Three numerical examples are given to establish the applicability and accuracy of the proposed method in Section . Section 5 concludes our study.

Figure 1. The absolute errors BPM in Example 1 at different time levels $t \leq 1$ with parameters $h_x = h_y = 0.04, \Delta t = 0.0001$. 

![Graph](image-url)
2. Description of MCB-DQM

Bellman, Kashef, and Casti (1972) have introduced the DQM. DQM is an approximation to the derivatives of a function that is obtained by means of the weighted sum of the functional values at certain discrete points. In DQM, the weighting coefficients are evaluated using various test functions such as spline functions, sinc function, Lagrange interpolation polynomials, modified (extended) cubic B-spline, etc. (Arora & Singh, 2013; Korkmaz & Dağ, 2011, 2013; Quan & Chang, 1989a, 1989b; Shu, Chen, Xue, & Du, 2001; Shu & Richards, 1992; Singh, Arora, & Singh, 2016; Singh & Arora, 2014; Singh & Kumar, 2016c; Zhong, 2004). The weighting coefficients are dependent on the spatial grid spacing, so one can assume NM grid points on the rectangle \( D = [a, b] \times [c, d] \) distributed uniformly, that is, \( a = x_1 < x_2, \ldots, x_{N-1} < x_N = b \) with \( h_x = x_{j+1} - x_j \), and \( c = y_1 < y_2, \ldots, y_{M-1} < y_M = d \) with \( h_y = y_{i+1} - y_i \). The solution \( \rho(x, y, t) \) at any time on the knot \((x_j, y_j)\) is \( \rho(x_j, y_j, t) \) for \( i = 1, \ldots, N \) and \( j = 1, \ldots, M \).

The approximate value of \( n \)-th order partial derivative of the function \( \rho(x, y, t) \) with respect to \( x \) at \((x_j, y_j)\) is given by

\[
\frac{\partial^n \rho_{ij}}{\partial x^n} = \sum_{k=1}^{N} a_{ik}^{(n)} \rho_{kj}, \quad i = 1, 2, \ldots, N, \tag{6}
\]

and the approximate value of \( m \)-th order partial derivative of the function \( \rho(x, y, t) \) with respect to \( y \) at \((x_j, y_j)\) is given by

\[
\frac{\partial^m \rho_{ij}}{\partial y^m} = \sum_{k=1}^{M} b_{jk}^{(m)} \rho_{ki}, \quad j = 1, 2, \ldots, M, \tag{7}
\]

where \( a_{ij} \) and \( b_{ij} \) denote the weighting coefficients of the partial derivatives \( \frac{\partial^n \rho_i}{\partial x^n} \) and \( \frac{\partial^m \rho_i}{\partial y^m} \) at \((x_j, y_j)\), respectively.

The cubic B-spline basis functions at the knots are defined as follows:

\[
\varphi_j(x) = \frac{1}{h_x^3} \begin{cases} 
(x - x_{j-2})^3, & x \in [x_{j-2}, x_{j-1}) \\
(x - x_{j-2})^3 - 4(x - x_{j-1})^3, & x \in [x_{j-1}, x_j) \\
(x_{j+2} - x)^3 - 4(x_{j+1} - x)^3, & x \in [x_j, x_{j+1}) \\
(x_{j+2} - x)^3, & x \in [x_{j+1}, x_{j+2}) \\
0, & \text{otherwise,}
\end{cases} \quad (8)
\]

where \( \{\varphi_0, \varphi_1, \ldots, \varphi_N, \varphi_{N+1}\} \) forms a basis over the interval \([a, b]\).

**Lemma 1** The numerical values of \( \varphi_j \) and its derivatives \( \varphi_j', \varphi_j'' \) at \( j \)-th nodal point are evaluated as

\[
\varphi_j(x_j) = \begin{cases} 
4, & i-j = 0 \\
1, & i-j = \pm 1 \\
0, & \text{else}
\end{cases}, \quad \varphi_j'(x_j) = \begin{cases} 
\pm 3/h_x, & i-j = \pm 1 \\
0, & \text{else}
\end{cases}, \quad \varphi_j''(x_j) = \begin{cases} 
-12/h^2_x, & i-j = 0 \\
6/h^2_x, & i-j = \pm 1 \\
0, & \text{else}
\end{cases}.
\]

The modified cubic B-spline basis functions at the knots are defined from Equation (8) in such a way that the resulting matrix system of equations remain diagonally dominant (Arora & Singh, 2013):

\[
\begin{aligned}
\phi_1(x) &= \varphi_1(x) + 2 \varphi_0(x) \\
\phi_2(x) &= \varphi_2(x) - \varphi_0(x) \\
\phi_j(x) &= \varphi_j(x) \quad \text{for } j = 3, \ldots, N - 2 \\
\phi_{N-1}(x) &= \varphi_{N-1}(x) - \varphi_N(x) \\
\phi_N(x) &= \varphi_N(x) + 2 \varphi_{N+1}(x)
\end{aligned} \tag{9}
\]

where \( \{\phi_1, \phi_2, \ldots, \phi_N\} \) forms a basis over the interval \([a, b]\).
2.1. Computation of the weighting coefficients $a_y^{(r)}$ and $b_y^{(r)}$ ($r = 1, 2$)

To find the weighting coefficients $a_y^{(1)}$, the modified cubic B-spline $\phi_y(x)$, $k = 1, 2, \ldots, N$ is used in Equation (6) due to fixed $y$ axis. The first-order derivative approximation at the grid point $(x_i, y_j)$ is

$$
\phi'_y(x_i) = \sum_{j=1}^{N} a_y^{(1)} \phi_y(x_j), \quad k = 1, \ldots, N.
$$

(10)

By Lemma 1 and Equations (9) and (10) is reduced into a tridiagonal system of linear equations

$$
\Phi \bar{a}[i] = \bar{H}[i], \quad \text{for } i = 1, \ldots, N,
$$

(11)

where $\Phi = [\phi_y]$ is the coefficient matrix of order $N$ whose $i$th row is given by $\Phi[i] = [\phi_{y1}, \phi_{y2}, \phi_{y3}, \ldots, \phi_{yN}]$, $\bar{a}[i]$ denotes the weighting coefficient vector corresponding to grid point $x_i$, that is, $\bar{a}[i] = [a_y^{(1)}, a_y^{(2)}, \ldots, a_y^{(N)}]$, and the coefficient vector $\bar{H}[i] = [h_{y1}, h_{y2}, \ldots, h_{yN}]^T$ corresponding to $x_i$, $i = 1, 2, \ldots, N$ are evaluated as

$$
\bar{H}[1] = \begin{bmatrix}
-6/h_x \\
6/h_x \\
0 \\
\vdots \\
0
\end{bmatrix}, \quad \bar{H}[2] = \begin{bmatrix}
-3/h_x \\
0 \\
3/h_x \\
0 \\
\vdots \\
0
\end{bmatrix}, \quad \ldots, \bar{H}[N - 1] = \begin{bmatrix}
0 \\
\vdots \\
0 \\
3/h_x \\
0 \\
-6/h_x
\end{bmatrix}, \quad \text{and } \bar{H}[N] = \begin{bmatrix}
0 \\
\vdots \\
0 \\
3/h_x \\
0 \\
6/h_x
\end{bmatrix}.
$$

Now, we apply the well-known “Thomas algorithm” to solve the resulting tridiagonal system of equations which provides the vector $\bar{a}[i]$, that is, the weighting coefficients $a_y^{(1)}, a_y^{(2)}, \ldots, a_y^{(N)}$, for $i = 1, \ldots, N$.

Similarly, the weighting coefficients $b_y^{(1)}$ can be computed by means of the modified cubic B-spline $\phi_y(y)$, $k = 1, 2, \ldots, M$ in Equation (7).

Using the coefficients $a_y^{(1)}$ and $b_y^{(1)}$, the weighting coefficients of higher order spatial derivatives can be obtained using the Shu’s (2000) recursive formulae

$$
a_y^{(r)} = r \left[ a_y^{(r-1)} - \frac{a_y^{(r-1)} - a_y^{(r-1)}}{x_j - y_j} \right], \quad \text{if } i \neq j, \quad \text{and } a_y^{(r)} = -\sum_{i=1}^{N} a_y^{(r-1)}, \quad \text{for } i, j = 1, 2, \ldots, N.
$$

$$
b_y^{(r)} = r \left[ b_y^{(r-1)} - \frac{b_y^{(r-1)} - b_y^{(r-1)}}{x_j - y_j} \right], \quad \text{if } i \neq j, \quad \text{and } b_y^{(r)} = -\sum_{i=1}^{N} b_y^{(r-1)}, \quad \text{for } i, j = 1, 2, \ldots, M.
$$

(12)

where $r$ represents the order of the spatial derivative, $a_y^{(r)}$ and $b_y^{(r)}$ are the weighting coefficients of rth-order partial derivatives of $u(x, y, t)$ at the point $(x_i, y_j)$ with respect to $x$ and $y$, respectively. In particular, the weighting coefficients $a_y^{(2)}, b_y^{(2)}$ of order 2 can be obtained by taking $r = 2$ in (12).

3. Implementation of MCB-DQM for 2D BPM

After putting the values of the spatial derivatives approximated using MCB-DQM, Equation (4) can be rewritten as

$$
\begin{align*}
\frac{a_\phi}{a_t} &= 2 \left( \rho_y \sum_{k=1}^{N} a^{(2)}_{yk} \rho_{yk} + \left( \sum_{k=1}^{N} a^{(1)}_{yk} \rho_{yk} \right)^2 + \rho_y \sum_{k=1}^{M} b^{(2)}_{yk} \rho_{yk} + \left( \sum_{k=1}^{M} b^{(1)}_{yk} \rho_{yk} \right)^2 \right) + \sigma(\rho_y) \\
\rho_y(t = 0) &= \rho_y^0, \quad 1 \leq i \leq N, 1 \leq j \leq M,
\end{align*}
$$

(13)
where \( \rho_{ij} = \rho(x, y, t) \) and \( \sigma(\rho) = h \rho_{ij}^0(1 - r \rho_{ij}^0) \). On implementing the conditions (5), Equation (13) reduced to

\[
\begin{cases}
\frac{d\rho_{ij}}{dt} = L(\rho_{ij}) \\
\rho_{ij}(t = 0) = \rho_{ij}^0, \quad 2 \leq i \leq N - 1, 2 \leq j \leq M - 1,
\end{cases}
\]

where \( L \) is nonlinear differential operator defined by

\[
L(\rho_{ij}) = 2 \left( \rho_{ij} \left( \sum_{k=2}^{N-1} a_{ik}^{(2)} \rho_{kj} + \sum_{k=2}^{M-1} b_{ik}^{(2)} \rho_{ik} \right) + \left( \delta_{ij} + \sum_{k=2}^{M-1} b_{ik}^{(1)} \rho_{ik} \right) \right) + S_i,
\]

\[
\delta_{ij} = b_{ij}^1 \rho_{ij} + b_{ij}^M \rho_{ij}; \quad \epsilon_{ij} = a_{ij}^1 \rho_{ij} + a_{ij}^M \rho_{ij};
\]

\[
S_i = 2 \rho_{ij} \left( a_{ij}^2 \rho_{ij} + a_{ij}^M \rho_{ij} + b_{ij}^1 \rho_{ij} + b_{ij}^M \rho_{ij} \right) + h \rho_{ij}^0(1 - r \rho_{ij}^0).
\]

The resulting system of initial values problem (14) can be solved by various schemes available in literature. Strong stability preserving time discretization methods were introduced to address the need for nonlinear stability properties in the time discretization, as well as the spatial discretization, of hyperbolic Partial DEs. Strong stability preserving Runge-Kutta (SSP-RK) schemes have some excellent properties such as large regions of absolute stability and low storage, SSP-RK54 is strongly stable scheme with Euler time discretizations for nonlinear hyperbolic PDEs (see Gottlieb, 2005; Spiteri & Ruuth, 2002, 2004 and the papers therein). This is why, we prefer SSP-RK54 scheme as defined below, to solve the system of first-order ODEs (14):

\[
\rho^{(1)} = \rho^m + 0.391752226571890 \bigtriangleup t L(\rho^m)
\]

\[
\rho^{(2)} = 0.444370493651235 \rho^m + 0.555629506348765 \rho^{(1)} + 0.368410593050371 \bigtriangleup t L(\rho^{(1)})
\]

\[
\rho^{(3)} = 0.620101851488403 \rho^m + 0.37989148511597 \rho^{(2)} + 0.251891774271694 \bigtriangleup t L(\rho^{(2)})
\]

\[
\rho^{(4)} = 0.178079954393132 \rho^m + 0.821920045606868 \rho^{(3)} + 0.544974750228521 \bigtriangleup t L(\rho^{(3)})
\]

\[
\rho^{(m+1)} = 0.517231671970585 \rho^{(4)} + 0.096059710526147 \rho^{(3)} + 0.0636924868666290 \bigtriangleup t L(\rho^{(3)})
\]

\[
+ 0.386708617530269 \rho^{(4)} + 0.226007483236906 \bigtriangleup t L(\rho^{(4)})
\]

**Lemma 2** (Prenter, 1975) If the function \( u \in C^4[a, b] \) such that

\[
u(x) = \sum_{j=1}^{N} \phi_j(x)u(x_j) + E(x),
\]

where \( \phi_j(x) \) is the cubic B-spline function. Then for any partition of \([a, b]\) with the grids distributed uniformly

(i) \(|E(x)|_\infty \leq \frac{1}{32} h^4 \|u^4(x)\|_\infty \);

(ii) \(|E^{(1)}(x)|_\infty \leq \frac{\sqrt{15}h^4}{4!} \|u^4(x)\|_\infty \);

(iii) \(|E^{(2)}(x)|_\infty \leq \frac{3}{16} h^4 \|u^4(x)\|_\infty \).

On setting \( a_{ij}^{(1)} = \frac{\partial \phi_j(x)}{\partial x} \bigg|_{x=x_i} \) and \( a_{ij}^{(2)} = \frac{\partial^2 \phi_j(x)}{\partial x^2} \bigg|_{x=x_i} \), and if \( \rho \in C^0(\mathbb{D}, \mathbb{D}) \), then from Lemma 2, we get

\[
\left| \frac{\partial \rho_{ij}}{\partial x} - \sum_{k=1}^{N} a_{ik}^{(n)} \rho_{kj} \right| \leq \frac{\sqrt{3} + 9}{216} h^3 M_a \equiv O(h^3),
\]

where \( h = \max\{h_x, h_y\} \) and \( M_a = \max \left\{ |\rho_{ij}^{(n)}| : 1 \leq i \leq N, 1 \leq j \leq M \right\} \).
(b) \[
\frac{\partial^2 \rho}{\partial x^2} - \sum_{k=1}^{N} a_{jk}^2 \rho_{jk} \equiv O(h^2)
\]

Therefore, we get
\[
\frac{\partial \rho}{\partial t} - L(\rho) \equiv O(h^2).
\]

This shows that the order of convergence of the MCB-DQM for BPM is two, which is confirmed numerically from Tables 4 and 8.

| Table 5. Errors in BPM given in Example 2 at \(0 \leq t \leq 1.0\) for different time steps and \(h_x = h_y = 0.1\) |
|-----------------|------------------|------------------|------------------|
| \(t\)           | \(\Delta t = 0.001\) | \(\Delta t = 0.0005\) | \(\Delta t = 0.0001\) |
|                 | \(L_2\) | \(L_\infty\) | CPU(s) | \(L_2\) | \(L_\infty\) | CPU(s) | \(L_2\) | \(L_\infty\) | CPU(s) |
| 0.1             | 1.172E-04 | 3.689E-04 | 0.03  | 5.872E-05 | 1.827E-04 | 0.06  | 4.156E-05 | 6.096E-05 | 0.19  |
| 0.5             | 1.179E-04 | 3.691E-04 | 0.09  | 5.897E-05 | 1.828E-04 | 0.20  | 4.380E-05 | 6.265E-05 | 0.94  |
| 1.0             | 1.184E-04 | 3.694E-04 | 0.20  | 5.901E-05 | 1.829E-04 | 0.40  | 4.334E-05 | 6.164E-05 | 1.92  |

| Table 6. Errors in BPM given in Example 2 at \(0 \leq t \leq 1.0\) with \(\Delta t = 0.0001\) |
|-----------------|------------------|------------------|
| \(h_x = h_y\)  | Errors           | \(t = 0.1\)     | \(t = 0.2\)     | \(t = 0.5\)     | \(t = 1.0\)     |
| 0.03            | \(L_2\)         | 6.43E-06         | 6.39E-06         | 6.36E-06         | 6.31E-06         |
|                 | \(L_\infty\)    | 3.70E-05         | 3.69E-05         | 3.69E-05         | 3.69E-05         |
|                 | CPU(s)          | 4.84             | 9.76             | 24.4             | 49.11            |
| 0.04            | \(L_2\)         | 9.46E-06         | 9.53E-06         | 9.56E-06         | 9.39E-06         |
|                 | \(L_\infty\)    | 3.65E-05         | 3.65E-05         | 3.65E-05         | 3.65E-05         |
|                 | CPU(s)          | 2.23             | 4.48             | 11.24            | 22.52            |

| Table 7. The errors in Example 2 for large time levels \(5 \leq t \leq 20\) with \(h_x = h_y = 1/12\) |
|-----------------|------------------|------------------|
| Errors          | \(t = 5\)       | \(t = 10\)      | \(t = 15\)      | \(t = 20\)      |
| \(L_2\)        | 2.71E-05         | 2.50E-05         | 2.33E-05         | 2.19E-05         |
| \(L_\infty\)   | 3.64E-05         | 3.65E-05         | 3.65E-05         | 3.66E-05         |
| CPU(s)          | 12.14            | 30.22            | 45.67            | 60.94            |

| Table 8. Rate of convergence of Example 2 |
|-----------------|------------------|------------------|
| \(M = N\)       | \(t = 1.0\)     | \(t = 0.5\)     |
|                 | \(L_2\) | ROC | CPU(s) | \(L_2\) | ROC | CPU(s) |
| 6               | 1.66E-04 | 0.36 | 0.16E-04 | 0.20 |
| 11              | 4.33E-05 | 2.21 | 1.90   | 4.38E-05 | 2.21 | 0.94 |
| 16              | 1.78E-05 | 2.38 | 5.45   | 1.80E-05 | 2.37 | 2.70 |
| 21              | 9.46E-06 | 2.32 | 12.00  | 9.53E-06 | 2.34 | 5.93 |
| 26              | 6.84E-06 | 1.52 | 22.33  | 6.34E-06 | 1.91 | 11.4 |

| Table 9. Comparison of numerical solution of Example 3 at \(x = 0.5, y = 0.5\) |
|-----------------|------------------|
| \(t\)           | MCB-DQM         | IEFGM (Zhang et al., 2014) | Exact |
|                 | CPU(s)          | CPU(s)         | CPU(s) |
| 0.1             | 0.5078          | 0.5069         | 0.5101 |
| 0.2             | 0.5170          | 0.5165         | 0.5204 |
| 0.3             | 0.5273          | 0.5268         | 0.5309 |
| 0.4             | 0.5379          | 0.5373         | 0.5416 |
| 0.5             | 0.5488          | 0.5481         | 0.5526 |
|                 | 0.31             | 0.61          | 0.94    |
|                 | 1.20             | 1.48          |        |
4. Numerical studies

This section deals with the main goal, the numerical study of three test problems of BPM approximated by MCB-DQM with SSP-RK54 scheme. The accuracy and the efficiency of the method is measured by evaluating the computational order, the $L_2$-error norm and the maximum error norm ($L_\infty$), and their computational time.

The errors in Example 3 for large time levels $2 \leq t \leq 10$ with $h_x = h_y = 1/12$

| Errors | $t = 2$ | $t = 5$ | $t = 8$ | $t = 10$ |
|--------|---------|---------|--------|---------|
| $L_2$  | 7.94E-03 | 1.50E-02 | 3.06E-02 | 2.07E-02 |
| $L_\infty$ | 2.31E-03 | 4.10E-03 | 1.13E-02 | 9.34E-02 |
| CPU(s) | 5.87 | 14.84 | 23.79 | 29.60 |

Figure 2. The approximate solution of BPM in Example 1 at different time levels $t \leq 1$ with parameters $h_x = h_y = 0.04$, $\Delta t = 0.0001$. 
Example 1 We consider BPM (4) with the parameters $h = -1$, $\alpha = 1 = \beta$ and $r = -\frac{8}{9}$ in the region $D = [0, 1]^2$, subject to the initial condition:
\[ \rho^0(x) = e^{\frac{-x}{u_1D6FC}}, \forall x \in \Omega, \]
and the boundary conditions are extracted from the exact solution
\[ \rho(x, t) = e^{\frac{-x}{u1D6FD}}, \forall x \in \Omega, t \geq 0. \]

The solutions are obtained with $\triangle t = 0.0001$. In Table 1, MCB-DQM solutions at time $0.1 \leq t \leq 0.5$ are compared with the solutions in Zhang et al. (2014) and the exact solutions. The $L_2$, $L_{\infty}$ errors and CPU time at $t = 0.1, 0.4$ is reported in Table 2 while the $L_2$ and $L_{\infty}$ error norms for large time $5 \leq t \leq 20$ is reported in Table 3. Table 4 shows that the convergence order for the BPM is quadratic. The computational time in seconds, CPU(s), for different time intervals $t \leq 20$ and various grid sizes with $\triangle t = 0.0001$ is also reported in the above Tables 1–4. The absolute errors in the BPM at different time levels $t \leq 1$ is depicted in Figure 1, and the surface solution is depicted in Figure 2. Tables and Figure 1 confirms that our results are in good agreement with the results obtained in Cheng et al. (2014), Zhang et al. (2014) and exact solutions. The behavior of the solutions show the similar characteristics as depicted in Cheng et al. (2014), Zhang et al. (2014).

Example 2 We consider BPM (4) with the parameters $h = 1/96$, $\alpha = -1$, $\beta = 1$ and $r = -48$ in the region $D = [0, 1]^2$, subject to the initial condition:
\[ \rho^0(x) = \frac{1}{4 \sqrt{2}} \sqrt{(x^2 + y^2) + y + 5}, \forall x \in \Omega \]
while the boundary conditions are extracted from the exact solution
\[ \rho(x, t) = \frac{1}{4 \sqrt{2}} \sqrt{(x^2 + y^2) + y + \frac{t}{3} + 5}, (x, t) \in \Omega \times [0, T]. \]
The solutions are obtained for different values of $\triangle t$ and $h_x = h_y$ with $t \leq 1$ and different time steps, are reported in Table 5, and $L_2$ and $L_{\infty}$ error norms at different time levels $t \leq 1$ with $h_x = h_y = 0.1$ and different time steps, are reported in Table 6, while the $L_2$ and $L_{\infty}$ error norms for large time levels $5 \leq t \leq 20$ are reported in Table 7. The computational time in seconds, CPU(s), for different time intervals $t \leq 20$ and various grid sizes with $\triangle t = 0.0001$ is also reported in the above Tables 5–8. The contour and surface plots of absolute errors in the BPM at $t = 0.1, 0.5, 1.0$ are depicted in Figure 3, and the surface solutions at these time intervals is depicted in Figure 4. It is evident that our results are in good agreement with the results obtained in Zhang et al. (2014) and exact solutions. Table 8 shows that the convergence quadratic. The behavior of $\rho(x, t)$ shows the similar characteristics as depicted in Zhang et al. (2014).

Example 3 We consider BPM (4) with the parameters $h = 1/5, \alpha = 1$ and $r = 0$ in the region $D = [0, 1]^2$, subject to the initial condition:
Figure 5. Contour plot of absolute errors of $\rho(x, t)$ in Example 3 for $t = 0.1, 0.2, 0.3, 0.5$ with parameters $h_x = h_y = 0.05, \Delta t = 0.0001$. 
while the boundary conditions are extracted from the exact solution

\[ \rho(x, t) = e^{t \sqrt{xy}}, \quad (x, t) \in D \times [0, T]. \]

The solutions are computed with \( \Delta t = 0.0001 \). In Table 9, the computed results, for \( h_x = h_y = 1/13 \), are compared with the recent results in Zhang et al. (2014) and the exact solutions. The \( L_2 \) and \( L_{\infty} \) error norms for large time levels \( 2 \leq t \leq 10 \) are reported in Table 10. The computational time in
seconds, CPU(s), for different time intervals $t \leq 10$, and $h_x = h_y = 1/12$ with $\Delta t = 0.0001$ is also reported in the Tables 9 and 10. The contour plots of absolute errors in the BPM at $t = 0.1, 0.2, 0.3, 0.5$ are depicted in Figure 5, and the surface solutions at these time intervals is depicted in Figure 6. It is evident that our results are in good agreement with the results obtained in Zhang et al. (2014) and exact solutions. The behavior of $p(x, t)$ shows the similar characteristics as depicted in Cheng et al. (2014), Zhang et al. (2014).

5. Conclusions
In the present paper, the MCB-DQM has been implemented for the numerical computation of 2D BPM. We present three test problems to confirm the efficiency and accuracy of the method for BPM, which shows that the the MCB-DQM solutions are in good agreement with that of obtained by the recent results using IEFGM by Zhang et al. (2014) and element free kp-Ritz method by Cheng et al. (2014).

Further, the scheme is tested for large time for the given problems which shows that the results obtained for large time levels are in good agreement with the exact solutions. It is found that the order of convergence of MCB-DQM for the solutions of BPM is quadratic.

Funding
The author received no direct funding for this research.

Author details
Brajesh Kumar Singh 1
E-mail: bksingh053@gmail.com
1 Department of Applied Mathematics, School of Physical Sciences, Babasaheb Bhimrao Ambedkar University, Lucknow 226 025, India.

Citation information
Cite this article as: A novel approach for numeric study of 2D biological population model, Brajesh Kumar Singh, Cogent Mathematics (2016), 3: 1261527.

References
Arafa, A. A., Rida, S. Z., & Mohamed, H. (2011). Homotopy analysis method for solving biological population model. Communications in Theoretical Physics, 56, 797–800.
Aronson, D. G. (1986). The porous medium equation. Lecture Notes in Mathematics, 10, 12–24.
Arora, G., Mittal, R. C., & Singh, B. K. (2014). Numerical solution of BBM-burger equation with quartic B-spline collocation method. Journal of Engineering Science and Technology, 8(Special Issue 1), 104–116.
Arora, G., & Singh, B. K. (2013). Numerical solution of Burgers’ equation with modified cubic B-spline differential quadrature method. Applied Mathematics and Computation, 224, 166–177.
Bear, J. (1972). Dynamics of fluids in porous media. New York, NY: American Elsevier.
Bellman, R, Kashef, B. G., & Casti, J. (1972). Differential quadrature: A technique for the rapid solution of nonlinear differential equations. Journal of Computational Physics, 10, 40–52.
Cheng, R. J., Zhang, L. W., & Liew, K. M. (2014). Modelling of biological population problems using the element-free kp-Ritz method. Applied Mathematics and Computation, 227, 274–290.
Dibenedetto, E. (1993). Degenerate parabolic equations. New York, NY: Springer-Verlag.
Giuggioli, L., & Kenkre, V. M. (2003). Analytic solutions of a nonlinear convective equation in population dynamics. Physica D, 183, 245–259.
Gottlieb, S. (2005). On high order strong stability preserving Runge-Kutta and multi step time discretizations. Journal of Scientific Computing, 25, 105–128.
Gurtin, M. E., & MacCamy, R. C. (1977). On the diffusion of biological population. Mathematical Biosciences, 33, 35–49.
Gurney, W. S. C., & Nisbet, R. M. (1975). The regulation of inhomogeneous populations. Journal of Theoretical Biology, 52, 441–457.
Jager, W., & Lu, Y. G. (1997). Global regularity of solutions for general degenerate parabolic equations in 1D. Journal of Differential Equations, 140, 365–377.
Korkmaz, A., & Dag, I. (2011). Shock wave simulations using sinc differential quadrature method. Engineering with Computers, 28, 654–674.
Korkmaz, A., & Dag, I. (2013). Cubic B-spline differential quadrature methods and stability for Burgers’ equation, 30, 320–344.
Liu, Y., Li, Z., & Zhang, Y. (2011). Homotopy perturbation method to fractional biological population equation. Fractional Differential Calculus, 2, 117–124.
Lu, Y. G. (2000). H-order estimates of solutions of biological populations. Applied Mathematics Letters, 13, 123–126.
Okubo, A. (1980). Diffusion and ecological problem, mathematical models, Biomathematics 10. Berlin: Springer Prentier. P. M. (1975). Splines and Variational Methods. New York, NY: Wiley.
Quan, J. R., & Cheng, C. T. (1989a). New insights in solving distributed system equations by the quadrature methods-I. Computers & Chemical Engineering, 13, 779–788.
Quan, J. R., & Cheng, C. T. (1989b). New insights in solving distributed system equations by the quadrature methods-II. Computers & Chemical Engineering, 13, 1017–1024.
Roul, P. (2010). Application of homotopy perturbation method to biological population model. Applied Mathematics and Computation, 10, 1369–1378.
Shakeri, F., & Dehghan, M. (2006). Numerical solution of a biological population model using He’s variational iteration method. Computers & Mathematics with Applications, 54, 1197–1209.
Shivanian, E. (2013). Analysis of meshless local radial point interpolation (MLRPI) on a nonlinear partial integro-differential equation arising in population dynamics. Engineering Analysis with Boundary Elements, 37, 1693–1702.
Shu, C. (2000). Differential quadrature and its application in engineering. Great Britain: Athenaeum Press.
Shu, C., Chen, W., Xue, H., & Du, H. (2001). Numerical study of grid distribution effect on accuracy of DQ analysis of beams and plates by error estimation of derivative approximation. International Journal for Numerical Methods in Engineering, 51, 159–179.
Shu, C., & Richards, B. E. (1992). Application of generalized differential quadrature to solve two dimensional incompressible Navier-Stokes equations. International
Singh, B. K., & Arora, G. (2014). A numerical scheme to solve Fisher-type reaction-diffusion equations. Nonlinear Studies/Mesa- Mathematics in Engineering, Science and Aerospace, 5, 153–164.

Singh, B. K., Arora, G., & Singh, M. K. (2016). A numerical scheme for the generalized Burgers-Huxley equation. Journal of the Egyptian Mathematical Society. doi:10.1016/j.joems.2015.11.003

Singh, B. K., & Bianco, C. (2016). A new numerical approach for the solutions of partial differential equations in three-dimensional space. Applied Mathematics & Information Sciences, 10(5), 1–10.

Singh, B. K., & Kumar, P. (2016a). FRDTM for numerical simulation of multi-dimensional, time-fractional model of Navier-Stokes equation. Ain Shams Engineering Journal. doi:10.1016/j.jaesej.2016.04.009

Singh, B. K., & Kumar, P. (2016b). Numerical computation for time-fractional gas dynamics equations by fractional reduced differential transforms method. Journal of Mathematics and System Science, 6, 248–259.

Singh, B. K., & Kumar, P. (2016c). A novel approach for numerical computation of Burgers’ equation (1 + 1) and (2 + 1) dimension. Alexandria Engineering Journal. doi:10.1016/j.aej.2016.08.023

Singh, B. K., & Srivastava, V. K. (2015). Approximate series solution of multi-dimensional, time fractional-order (heat-like) diffusion equations using FRDTM. Royal Society Open Sciences, 2, 140511. doi:10.1098/rsos.140511

Spiteri, J. R., & Ruuth, S. J. (2002). A new class of optimal high-order strongstability-preserving time-stepping schemes. SIAM Journal on Numerical Analysis, 40, 469–491.

Spiteri, J. R., & Ruuth, S. J. (2004). Downwinding in high-order strong-stability-preserving Runge-Kutta methods. SIAM Journal on Numerical Analysis, 42, 974–996.

Srivastava, V. K., Kumar, S., Awasthi, M. K., & Singh, B. K. (2014). Two-dimensional time fractional-order biological population model and its analytical solution. Egyptian Journal of Basic and Applied Sciences, 1, 71–76.

Srivastava, V. K., Mishra, N., Kumar, S., Singh, B. K., & Awasthi, M. K. (2014). Reduced differential transform method for solving (1+n)-dimensional Burgers’ equation. Egyptian Journal of Basic and Applied Sciences, 1, 115–119.

Srivastava, V. K., & Singh, B. K. (2014). A Robust finite difference scheme for the numerical solutions of two dimensional time-dependent coupled nonlinear Burgers’ equations. International Journal of Applied Mathematics and Mechanics, 10, 28–39.

Zhong, H. (2004). Spline-based differential quadrature for fourth order equations and its application to Kirchhoff plates. Applied Mathematical Modelling, 28, 353–366.