Operational Matrices for Solving Burgers' Equation by Using Block-Pulse Functions with Error Analysis

K. Maleknejad, E. Babolian, A. Jafari Shaerlar, M. Jahangir

Department of Mathematics, Science and Research Branch, Islamic Azad University (IAU)-Tehran, Iran.

Abstract: This paper is provided to the development of numerical method to deal with Burgers' equation. The technics is based on two dimensional Block Pulse functions (BPFs) under the framework of operational matrices. In this work we apply new ideas to account operational matrix for partial derivative. We show how to convert burgers' equation to nonlinear system of algebra. Error analysis for this method are given by projection operator. The algorithm have been tested for an example. The numerical results obtained by this approach have been compared with the exact solution to show the efficiency of the method.

Key words: Nonlinear parabolic equations, Numerical methods, Direct method, Operational matrices, Burgers' equations.

INTRODUCTION

The one dimensional nonlinear partial equation.

\[
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, 0 \leq x \leq 1, 0 \leq t \leq T
\]  

such that

\[
\begin{align*}
    u(x, 0) &= f(x), \\
    u(0, t) &= g_1(t), \\
    u(1, t) &= g_2(x),
\end{align*}
\]

in known as Burgers' equation. Where \( \varepsilon > 0 \) is the coefficient of the kinematic viscosity, see Polyannin 2000.

Burgers' equation is a fundamental partial differential equation from fluid mechanics. It occurs in various equation areas of applied mathematics, such as modeling of dynamics, heat conduction, acoustic waves, gas dynamics, traffic flow, and many other application involving the study of nonlinear waves.

The mathematical properties of equation (1) have been studied by Cole 1951. He also gave an exact solution of Burgers' equation. Benton and Blatzman have demonstrated about 35 distinct exact solution of Burger-like equations and their classifications. It is well known that the exact solution of Burgers' equation can only be computed for restricted values of \( \varepsilon \). Many of the analytical solution involve Fourier series solutions. The convergence of the analytic solution in very slow due to series solution for the smaller velocity constant \( \varepsilon \), see (Aksan, 2006; Asaithambi, 2010). In these cases, numerical solutions may produces results including non-physical solution. Numerical techniques are tested on the Burgers' equation to make comparison with analytical and exact solutions. Several numerical methods for (1) have been used by many researches, for instance, finite element method (Varoglu, 1980; Caldwell, 1981) finite difference method (Smith, 1989; Ames, 1992) spectral method (Ben, 1997; Babolian, 2009) Adomian decomposition method (Zhu, 2010) direct variational method (Ozas, 1996) and projection method by B-spline (Dag, 2005).

In this paper, we present a simple numerical method for solving Burgers' equation by using two dimensional Block-Pulse functions. Block-pulse functions have been studied extensively as a basic set of functions for signal characterization in system science and control (Maleknejad, 2003; Maleknejad, 2005). This set of functions was first introduced to electrical engineering by (Harmuth, 1969; see Babolian, 2008; Jiang, 1992). We solve (1) by two dimensional Block-Pulse functions and apply their operational matrices. We approximate partial derivatives by \( 2DBPFs \). By using direct method, any partial differential equation is converted to linear or nonlinear algebraic system of equations. The organization of this paper ia as follow: In section 2 we introduce \( 2DBPFs \) and their properties. In section 3 we present operational matrices for partial derivatives. Direct method for solving Burgers' equation are given in section 4. Error analysis for proposed method are investigated. Computational results of the Burgers' equation for a test problem are illustrated.
Two dimensional Block-Pulse functions:

A set of two dimensional Block-Pulse functions \( \Phi_{i_1,i_2}(x,t) \) is defined in the region \( x \in [a, b) \) and \( t \in [0, T) \) as:

\[
\Phi_{i_1,i_2}(x,t) = \begin{cases} 
1 & (i_1)h_1 \leq x < (i_1 + 1)h_1, (i_2)h_2 \leq t < (i_2 + 1)h_2 \\
0 & \text{otherwise}. 
\end{cases} 
\]  

(5)

where \( m_1, m_2 \) are arbitrary positive integers, and \( h_1 = \frac{b-a}{m_1}, h_2 = \frac{T}{m_2} \). There are some properties for 2DBPFs as following:

The 2DBPFs are disjoint, orthogonal and complete set.

We can also expand a two variable function \( u(x,t) \) into BPFs series:

\[
u(x,t) \approx \sum_{i_1=0}^{m_1-1} \sum_{i_2=0}^{m_2-1} u_{i_1,i_2}(x,t),
\]  

(6)

through determining the block pulse coefficients:

\[
u_{i_1,i_2} = \frac{1}{h_1 h_2} \int_{(i_1)h_1}^{(i_1+1)h_1} \int_{(i_2)h_2}^{(i_2+1)h_2} u(x,t) \, dx \, dt,
\]  

(7)

Also, for vector forms, consider the \( m \times m \) elements of 2DBPFs:

\[
\Phi(x,t) = [\Phi_{0,0}, \Phi_{0,1}, \ldots, \Phi_{0,m-1}, \ldots, \Phi_{m-1,0}, \ldots, \Phi_{m-1,m-1}]^T(x,t).
\]  

(8)

The two important properties of 2DBPFs are given as:

(i):

\[
\Phi(x,t) \Phi^T(x,t) V = \bar{V} \Phi(x,t),
\]  

(9)

where \( V \) is an \( m \times m \) vector and \( \bar{V} = \text{diag}(V) \). Moreover, it can be clearly concluded that for every \( m \times m \) matrix \( B \):

(ii):

\[
\Phi^T(x,t) B \Phi(x,t) = \bar{B}^T \Phi(x,t),
\]  

(10)

where \( \bar{B} \) is an \( m \times m \) column vector with elements equal to the diagonal entries of matrix \( B \). For simplicity, we use \( m_1 = m_2 = m \).

Let \( D_T = \{(x,t) : a < x < b, 0 < t < T\} \) where \( -\infty < a < b < \infty \), and \( \partial D_T \) be the parabolic boundary of \( D_T \). If \( a, b \) are finite,

\[
\partial D_T = \{x = a, x = b, 0 \leq t \leq T\} \cup \{a \leq x \leq b, t = 0\}.
\]

If \( a, b \) are infinite,

\[
\partial D_T = \{x \in \mathbb{R}, t = 0\}
\]

And

\[
L^{2,1}(D_T) = \{u(x,t) : u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial t^2} \in L^2(D_T)\}.
\]  

(11)

without loss of generality, set \( a = 0, b = 1 \) and \( T = 1 \). The inner product \( < u, v > \) and norm \( || \) in \( L^{2,1}(D_T) \) are defined as follows:

\[
<u(x,t), v(x,t)> = \int_0^1 \int_0^1 u(x,t)v(x,t) \, dx \, dt,
\]  

(12)

\[
\|u(x,t)\| = \left( \int_0^1 \int_0^1 u^2(x,t) \, dx \, dt \right)^{1/2}.
\]  

(13)

Let \( P_m \) be the projection operator defined on \( L^{2,1}(D_T) \) → \( \mathbb{B} \), where \( \mathbb{B} \) is finite \( m^2 \) dimensional, as:

\[
u_m(x,t) = P_m u(x,t) = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} u_{ij} \Phi_{ij}(x,t).
\]  

(14)
First, we find an estimation of $\|u - P_m u\|$ for arbitrary $u \in L^2(\mathbb{D})$.

Let $u(x, t)$ be defined on $L^2(\mathbb{D})$ and $P_m$ be projection operator defined by (13) then

$$
\|u - P_m u\| \leq \frac{\text{max}|u|}{2\sqrt{3}m},
$$

where $\text{max}|u| = \text{max}_{i,j=0}^{m-1}|u_{ij}|$ for $0 \leq i, j \leq m - 1$.

**Proof:** The integral $\int_0^1 \int_0^1 u_{ij}\Phi(x, y)\,dxdy$ is a ramp $\frac{u_{ij}}{m}(t - \frac{i}{m})$ on the subinterval $[\frac{i}{m}, \frac{i+1}{m}] \times [\frac{j}{m}, \frac{j+1}{m}]$ with average value $\frac{u_{ij}}{2m^2}$.

The error in approximating the ramp by this constant value over the subinterval $[\frac{i}{m}, \frac{i+1}{m}] \times [\frac{j}{m}, \frac{j+1}{m}] = I_{ij}$ is

$$
r_{i,j}(s, t) = \frac{u_{ij}}{2m^2} - \frac{u_{ij}}{m}(t - \frac{i}{m}),
$$

hence, using $E_{ij}$ as least square of the error on $I_{ij}$, we have

$$
E_{ij}^2 = \int_{\frac{i}{m}}^{\frac{i+1}{m}} \int_{\frac{j}{m}}^{\frac{j+1}{m}} (r_{i,j}(s, t))^2dsdt \leq \frac{|u_{ij}|^2}{12m^2}.
$$

$$
E_{ij} \leq \frac{|u_{ij}|}{2\sqrt{3}m^2},
$$

and on the interval $\mathbb{D}$ we have

$$
\|u - P_m u\| = \text{max}E_{ij} \leq \frac{\text{max}|u|}{2\sqrt{3}m}.
$$

**Operational matrix for partial derivatives:**

The expansion of function $u(x, t)$ over $\mathbb{D}$ with respect to $\Phi_{i,j}(x, t), i, j = 0, 1, ..., m - 1$, can be written as.

$$
u(x, t) \approx \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} u_{ij}\Phi_{i,j}(x, t) = U^T\Phi = \Phi U,
$$

where $U = [u_{0,0}, u_{0,1}, ..., u_{0,m-1}, u_{1,0}, ..., u_{1,m-1}, ..., u_{m-1,0}, u_{m-1,1}, ..., u_{m-1,m-1}]^T$, $\Phi = [\Phi_{0,0}, \Phi_{0,1}, ..., \Phi_{0,m-1}, \Phi_{1,0}, ..., \Phi_{1,m-1}, ..., \Phi_{m-1,0}, \Phi_{m-1,1}, ..., \Phi_{m-1,m-1}]^T$ and

$$
\Phi_{i,j}(x, t) = \begin{cases} 1 & \frac{i}{m} \leq x < \frac{i+1}{m}, \frac{j}{m} \leq t < \frac{j+1}{m} \\
0 & \text{otherwise}
\end{cases}
$$

$$
u_{i,j} = \frac{1}{h^2} \int_{\frac{i}{m}}^{\frac{i+1}{m}} \int_{\frac{j}{m}}^{\frac{j+1}{m}} u(x, t)\,dxdt.
$$

Now, expressing $\int_0^1 \int_0^t \Phi_{i,j}(s, y)\,dsdy$, in terms of the 2DBPFs as:

$$
\int_0^1 \int_0^t \Phi_{i,j}(s, y)\,dsdy \approx [0, 0, h^2, ..., h^2],
$$

in which $\frac{h^2}{2}$ is $i$th component. Thus

$$
\int_0^1 \int_0^t \Phi(s, y)\,dsdy \approx P\Phi(x, t),
$$

where $P$ is $m^2 \times m^2$ matrix and is called operational matrix of double integration and can be denoted by $P = \frac{h^2}{2} P_2$, where

\[
P_2 = \begin{pmatrix}
1 & 2 & 2 & \cdots & 2 \\
0 & 1 & 2 & \cdots & 2 \\
0 & 0 & 1 & \cdots & 2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1
\end{pmatrix}
\]
so, the double integral of every function \( u(x,t) \) can be approximated by:

\[
\int_0^1 \int_0^t u(s,y) ds dy \approx \frac{h^2}{2} U^TP_2 \Phi(x,t),
\]  

(26)

by similar method \( \int_0^1 \Phi_{ij}(s,t) ds \), in terms of 2DBPFs as:

\[
\int_0^1 \Phi_{ij}(s,t) ds \approx [0,0,\ldots,h,0,0,\ldots,0]^T \Phi(\ldots),
\]

(27)

And

\[
\int_0^1 \Phi(s,t) ds \approx hI\Phi(\ldots).
\]

(28)

Now, we compute operational matrix for \( \Phi \).

Lemma 2: Suppose \( u \in L^2(D_T) \) and \( u \) is defined on parabolic boundary \( \partial_pD_T \) then operational matrix for \( \frac{\partial u}{\partial t} \) by 2DBPFs is approximated as:

\[
\frac{\partial u(x,t)}{\partial t} \approx (U_t^d)^T \Phi(x,t)
\]

(29)

that:

\[
U_t^d = \frac{2}{h} (U^T - U_f \Delta_1) P_{-1}^{-1},
\]

(30)

where \( \Delta_1 \) is the following \( m^2 \times m^2 \) matrix as:

\[
\Delta_1 = \begin{pmatrix}
1 & 0 & 0 & \ldots & 0 \\
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 1 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & 0
\end{pmatrix}
\]

(31)

With

\[
H_{m\times m} = \begin{pmatrix}
1 & 0 & 0 & \ldots & 0 \\
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 1 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & 0
\end{pmatrix}
\]

(32)

that \( U_f \) is initial boundary vector of \( \partial_pD_T \).

Proof: By applying approximation \( \frac{\partial u}{\partial t} \approx (U_t^d)^T \Phi \) in (24) instead of \( u \) we have:

\[
\int_0^1 \int_0^t \frac{\partial u(x,y)}{\partial y} ds dy \approx \frac{h^2}{2} (U_t^d)^T P_2 \Phi(x,t),
\]

(33)

And

\[
\int_0^1 \int_0^t \frac{\partial u(x,y)}{\partial y} ds dy = \int_0^1 (u(s,t) - u(s,0)) ds \\
= \int_0^1 (U^T \Phi(s,t) - U_f^T \Phi(s,0)) ds \\
= hU^T \Phi(x,t) - hU_f^T \Delta_1 \Phi(x,t),
\]

(34)

from (29) and (30) we can conclude:

\[
U_t^d = \frac{2}{h} (U^T - U_f \Delta_1) P_{-1}^{-1}.
\]

(35)
by the same method, operational matrix for $\frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}$ are given as follows. 

Lemma 3: If $u \in L^2(D_T)$ and defined in parabolic boundary $\partial_P D_T$ then operational matrix for $\frac{\partial u}{\partial x}$ and $\frac{\partial^2 u}{\partial x^2}$ by 2DBPFs are approximated as:

$$
\frac{\partial u}{\partial x} \approx (U^d_x)^T \Phi(x, t), \quad (36)
$$

$$
\frac{\partial^2 u}{\partial x^2} \approx (U^d_{xx})^T \Phi(x, t), \quad (37)
$$

Where

$$
U^d_x = \frac{1}{h}(U^T_{g_2} \Delta_3 - U^T_{g_1} \Delta_2)P_2^{-1}, \quad (38)
$$

$$
U^d_{xx} = \frac{1}{k^2}(U^T_{g_2} \Delta_3 - U^T_{g_1} \Delta_2)P_2^{-1}(\Delta_3 - \Delta_2)P_2^{-1}, \quad (39)
$$

and $\Delta_2, \Delta_3$ are the following $m^2 \times m^2$ matrices:

$$
\Delta_2 = \begin{pmatrix}
1_{m \times m} & 0 & \ldots & 0 \\
0 & 1 & \ldots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & 1
\end{pmatrix}, \quad (40)
$$

$$
\Delta_3 = \begin{pmatrix}
0_{m \times m} & 0 & \ldots & 0 \\
0 & 1 & \ldots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & 1
\end{pmatrix}, \quad (41)
$$

and $U_{g_1}, U_{g_2}$ are boundary vectors of $\partial_P D_T$.

Direct method for solving nonlinear PDEs:

The results obtained in previous section are used to introduce a direct efficient and simple method to solve equations (1) – (4). We consider equations (1) – (4) of the form:

$$
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, (x, t) \in D_T
$$

$$
u(x, 0) = f(x),
$$

$$
u(0, t) = g_1(x),
$$

$$
u(1, t) = g_2(x). \quad (42)
$$

Approximating function $u \frac{\partial u}{\partial x}$ with respect to 2DBPFs:

$$
u(x, t) \frac{\partial u}{\partial x} \approx \Phi^T(x, t)U(U^d_x)^T \Phi(x, t)
$$

$$
= (U(U^d_x)^T) \Phi(x, t). \quad (44)
$$

By substituting the above equations into (36) and using boundary and initial conditions, we obtain a nonlinear system with $u_{i,j}(i, j = 0, 1, \ldots, m - 1)$ as unknowns:

$$
(U^d_x)^T - \varepsilon (U^d_{xx})^T + U(U^d_x)^T = 0. \quad (45)
$$

Error analysis:

Let the problem be of the form

$$
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, (x, t) \in D_T
$$

$$
u(x, 0) = f(x),
$$

$$
u(0, t) = g_1(t),
$$

$$
u(1, t) = g_2(t), \quad (46)
$$

where $f(x), g_1(t), g_2(t)$ belong to $L^2[0, 1]$, and $L$ is linear.
By using (13), the discrete approximation of (39) is:

$$\frac{\partial u_m}{\partial t} + u_m \frac{\partial u_m}{\partial x} = \varepsilon \frac{\partial^2 u_m}{\partial t^2} + e,$$

where, for each $(x,t), P_m u(x,t)$ belongs to an $m^2$-dimensional subspace $\mathbb{B}$.

**Theorem 1:** Let $u(x,t)$ and $f(x,t)$ be in $L^2,1(D_T)$ and $u_m(x,t)$ be approximate solution by 2DBPFs of (13)

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial t^2} + e,$$

then

$$\|e\| \leq \frac{1}{2\sqrt{3}m} (\max|\frac{\partial u}{\partial t}| + \varepsilon \max|\frac{\partial^2 u}{\partial x^2}| + \max|u \frac{\partial u}{\partial x}| + Pu_m Pm \alpha x |\frac{\partial u}{\partial x}|)$$

Proof: By using properties of projection operators,

$$e = \frac{\partial u_m(x,t)}{\partial t} - \frac{\partial u(x,t)}{\partial t} - \varepsilon \left(\frac{\partial^2 u_m(x,t)}{\partial x^2} - \frac{\partial^2 u(x,t)}{\partial x^2}\right) + u_m \frac{\partial u_m(x,t)}{\partial x} - u \frac{\partial u(x,t)}{\partial x}$$

$$e = \varepsilon \frac{\partial (u_m - u)}{\partial t} - \varepsilon \frac{\partial^2 (u_m - u)}{\partial x^2} + u_m \frac{\partial (u_m - u)}{\partial x} - (u - u_m) \frac{\partial u}{\partial x}$$

$$e = \varepsilon \frac{\partial ((P_m - I)u)}{\partial t} - \varepsilon \frac{\partial^2 ((P_m - I)u)}{\partial x^2} + u_m \frac{\partial ((P_m - I)u)}{\partial x} - (I - P_m)u \frac{\partial u}{\partial x}$$

$$\|e\| \leq \frac{1}{2\sqrt{3}m} (\max|\frac{\partial u}{\partial t}| + \varepsilon \max|\frac{\partial^2 u}{\partial x^2}| + \max|u \frac{\partial u}{\partial x}| + Pu_m Pm \alpha x |\frac{\partial u}{\partial x}|)$$

$$\|e\| \leq \frac{A}{2\sqrt{3}m}$$

where $A = \max|\frac{\partial u}{\partial t}| + \varepsilon \max|\frac{\partial^2 u}{\partial x^2}| + \max|u \frac{\partial u}{\partial x}| + Pu_m Pm \alpha x |\frac{\partial u}{\partial x}|$ for $(x,t) \in D_T$, so by hypothesis of the theorem, $A$ is a finite number and $\|e\| = O(\frac{1}{m})$. So, if $m \to \infty$ then $\|e\|$ tends to zero.

**Numerical example:**

We present numerical results to illustrate the effectiveness of the proposed method. Consider Burgers’ equation (1) with the following initial and boundary functions.

$$u(x,0) = 2\pi a \sin(\pi x) a + a a, 0 \leq x \leq 1, a > 1,$$

$$u(0,t) = u(1,t) = 0.$$  

The exact solution of (1) with above conditions was given in Asaithambi 2010 as.

$$u(x,0) = \frac{2\pi a e^{-\pi^2 \varepsilon^2 \sin(\pi x)}}{a + \pi e^{-\pi^2 \varepsilon^2 \cos(\pi x)}}$$

We solve Burgers' equation (1) by direct method and numerical results obtained can be compared with exact solution. The numerical results show that with increasing $m$, the approximate solution gets better. To show the accuracy of method we report norm of the error which is defined by:

$$e_{i,j} = \|u(x_i, t_j) - u_m(x_i, t_j)\|.$$  

Computational results for some $\varepsilon, m$ and $a$ are given in Tables 1 and 2, Also error surface for some $\varepsilon, m$ and $a$ are shown in Figures 1, 2, 3.

**Conclusion:**

In this paper, we introduced a new numerical scheme for Burgers’ equation by two dimension block pulse functions and their operational matrices for partial derivatives. This method can be used for any linear and nonlinear partial differential equations. We can say that this method is feasible and the error is acceptable. the
implementation of the present method is a very easy acceptable and valid. We can use other piecewise constant functions for example Haar, Walsh and wavelets.

Fig. 1: Error surface for \( m = 16, a = 2, \varepsilon = 0.001 \).

Fig. 2: Error surface for \( m = 16, a = 2, \varepsilon = 0.0000 \).

Fig. 3: Error surface for \( m = 20, a = 100, \varepsilon = 0.00001 \).

Table 1: Numerical results for \( u(x, 0), a = 2, m = 20, \Delta = 0.05, \varepsilon = 0.01 \).

<table>
<thead>
<tr>
<th>( x )</th>
<th>Numerical</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>6.4917e-3</td>
<td>1.311e-3</td>
</tr>
<tr>
<td>0.4</td>
<td>6.4046e-3</td>
<td>2.5769e-2</td>
</tr>
<tr>
<td>0.6</td>
<td>6.3182e-3</td>
<td>3.5132e-2</td>
</tr>
<tr>
<td>0.8</td>
<td>6.2324e-3</td>
<td>3.0754e-2</td>
</tr>
</tbody>
</table>

Table 2: Numerical results for \( u(x, 0), a = 2, m = 20, \Delta = 0.05, \varepsilon = 0.01 \).

<table>
<thead>
<tr>
<th>( x )</th>
<th>Numerical</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>6.5793e-6</td>
<td>1.3147e-5</td>
</tr>
<tr>
<td>0.4</td>
<td>6.5792e-6</td>
<td>2.5879e-5</td>
</tr>
<tr>
<td>0.6</td>
<td>6.5791e-6</td>
<td>3.5338e-5</td>
</tr>
<tr>
<td>0.8</td>
<td>6.5790e-6</td>
<td>3.1009e-5</td>
</tr>
</tbody>
</table>

REFERENCES


