On the numerical solution of the system of two-dimensional Burgers’ equations by the decomposition method

Salah M. El-Sayed a,*, Doğan Kaya b

a Department of Mathematics, Benha University, Benha 13518, Egypt
b Department of Mathematics, Firat University, Elazig 23119, Turkey

Abstract

Adomian’s decomposition method (ADM) is proposed to approximate the numerical and analytical solutions of system two-dimensional Burgers’ equations (STDBE) with initial conditions. The advantages of this work are the decomposition method reduces the computational work and improvement with regard to its accuracy and rapid convergence. Some examples are given to illustrate the performance of the method described.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Two-dimensional Burgers’ equations; Adomian’s decomposition method; Adomian’s polynomials

1. Introduction

Partial differential equations arise in every field of science, in particular in physics, engineering, chemistry and finance, and are fundamental for the mathematical formulation of continuum models. Systems of partial differential equations have attracted much attention in studying evolution equations describing wave propagation, in investigating the shallow water waves [1,2], and
in examining the chemical reaction–diffusion model of Brusselator [3]. While the Burgers’ equation has been found to describe various kind of phenomena such as a mathematical model of turbulence [4] and the approximate theory of flow through a shock wave traveling in a viscous fluid [5]. Fletcher using the Hopf–Cole transformation [6] gave an analytic solution of STDBE. Several numerical methods of this equation system have been given such as algorithms based on cubic spline function technique [7], applied an explicit–implicit method [8], several multilevel schemes with ADI [9] and implicit finite-difference scheme [10].

Recently a great deal of interest has been focused on the application of Adomian’s decomposition method to solve a wide variety of stochastic and deterministic [11–13]. Although the Adomian’s goal is to find a method to unify linear and nonlinear, ordinary or partial differential equations for solving initial and boundary value problems. It has been shown that the series converges fast, and with only a few terms this series approximates the exact solution with a fairly reasonable error [14–18]. The convergence of the ADM has investigated by several authors [19–23]. In recent work of Ngarhasta et al. [24] have proposed a new approach of convergence of the ADM. In this paper, we shall adapt the algorithm to the solution system of two-dimensional Burgers’ equations. The system of two-dimensional Burgers’ equation [10] will be more easily, quickly, and elegantly by implementing the ADM rather than the traditional methods for the exact solutions as well as numerical solutions, without suffering traditional difficulty.

2. The analysis of the algorithm of ADM

In this section we first describe the algorithm of the ADM as it applies to system of two-dimensional Burgers’ equations:

\[
\begin{align*}
    &u_t + uu_x + vu_y = \frac{1}{R} (u_{xx} + u_{yy}), \\
    &v_t + uu_x + vu_y = \frac{1}{R} (v_{xx} + v_{yy}),
\end{align*}
\]

subject to the initial conditions \(u(x, y, 0) = f_1(x, y)\) and \(v(x, y, 0) = f_2(x, y)\). Where \(u(x, y, t)\) and \(v(x, y, t)\) are the velocity components to be determined and \(R\) is the Reynolds number. Following we define the linear operators \(L_t \equiv \frac{\partial}{\partial t}\), \(L_x \equiv \frac{\partial^2}{\partial x^2}\), \(L_{xx} \equiv \frac{\partial^2}{\partial x^2}\) and \(L_{yy} \equiv \frac{\partial^2}{\partial y^2}\). Therefore we rewrite Eq. (1) as

\[
\begin{align*}
    &L_t u + u L_x u + v L_y u = \frac{1}{R} (L_{xx} u + L_{yy} u), \\
    &L_t v + u L_x v + v L_y v = \frac{1}{R} (L_{xx} v + L_{yy} v).
\end{align*}
\]
By defining the onefold right-inverse operator $L_t^{-1}$ we can formally that

$$
\begin{align*}
    u(x, y, t) &= u(x, y, 0) - L_t^{-1}(uL_t u) - L_t^{-1}(vL_t v) + \frac{1}{R} L_t^{-1}(L_{xx} u + L_{yy} v), \\
    v(x, y, t) &= v(x, y, 0) - L_t^{-1}(uL_t v) - L_t^{-1}(vL_t v) + \frac{1}{R} L_t^{-1}(L_{xx} v + L_{yy} v).
\end{align*}
$$

Therefore we can write

$$
\begin{align*}
    u(x, y, t) &= u(x, y, 0) - L_t^{-1}(M_1(u)) - L_t^{-1}(N_1(u, v)) + \frac{1}{R} L_t^{-1}(L_{xx} u + L_{yy} v), \\
    v(x, y, t) &= v(x, y, 0) - L_t^{-1}(N_2(u, v)) - L_t^{-1}(M_2(v)) + \frac{1}{R} L_t^{-1}(L_{xx} v + L_{yy} v).
\end{align*}
$$

The decomposition method [11–13] suggests that the linear terms $u(x, y, t)$ and $v(x, y, t)$ be decomposed by an infinite series of components

$$
\begin{align*}
    u(x, y, t) &= \sum_{n=0}^{\infty} u_n(x, y, t), \quad v(x, y, t) &= \sum_{n=0}^{\infty} v_n(x, y, t)
\end{align*}
$$

and the nonlinear operators $M_1(u)$, $N_1(u, v)$, $M_2(u, v)$ and $N_2(v)$ by the infinite series:

$$
\begin{align*}
    M_1(u) &= \sum_{n=0}^{\infty} A_n, \quad N_1(u, v) = \sum_{n=0}^{\infty} B_n, \\
    M_2(u, v) &= \sum_{n=0}^{\infty} C_n, \quad N_2(v) = \sum_{n=0}^{\infty} D_n,
\end{align*}
$$

where $u_n(x, y, t)$ and $v_n(x, y, t)$, $n \geq 0$ are the components $u(x, y, t)$ and $v(x, y, t)$ that will be elegantly determined, and $A_n$, $B_n$, $C_n$ and $D_n$ are called Adomian polynomials.

Hence we obtain the components series solution by the following recursive relationship:

$$
\begin{align*}
    u_0 &= f_1(x, y), \quad v_0 = f_2(x, y), \\
    u_{n+1} &= -L_t^{-1}A_n - L_t^{-1}B_n + L_t^{-1}L_{xx} u_n + L_t^{-1}L_{yy} u_n, \\
    v_{n+1} &= -L_t^{-1}C_n - L_t^{-1}D_n + L_t^{-1}L_{xx} v_n + L_t^{-1}L_{yy} v_n,
\end{align*}
$$

where $n \geq 0$.

The Adomian’s polynomials that can be generated for all forms of nonlinearity [11–13]. The Adomian’s polynomials $A_n$, $B_n$, $C_n$ and $D_n$ are generated according to the following algorithms:
Similarly, we can find \( C_n \) and \( D_n \). This formulae is easy to set computer code to get as many polynomial as we need in the calculation. We can give the first few Adomian’s polynomials of the \( A_n \) and \( B_n \) as

\[
A_0 = u_0, u_0,
A_1 = u_0, u_1 + u_0, u_1,
A_2 = u_0, u_2 + u_1, u_1 + u_2, u_0,
A_3 = u_0, u_3 + u_1, u_2 + u_2, u_1 + u_3, u_0,
A_4 = u_0, u_4 + u_1, u_3 + u_2, u_2 + u_3, u_1 + u_4, u_0,
B_0 = u_0, v_0,
B_1 = u_0, v_1 + u_1, v_0,
B_2 = u_0, v_2 + u_1, v_1 + u_2, v_0,
B_3 = u_0, v_3 + u_1, v_2 + u_2, v_1 + u_3, v_0,
B_4 = u_0, v_4 + u_1, v_3 + u_2, v_2 + u_3, v_1 + u_4, v_0
\]

and so on, the rest of the polynomials and of the polynomials \( C_n \) and \( D_n \) can be constructed in a similar manner.

### 3. Numerical implementation of the ADM

We first consider the application of the decomposition method to the STDBE (1) with the initial conditions

\[
u(x, y, 0) = \frac{3}{4} \cdot \frac{1}{4(1 + e^{(x+y)R})}, \quad v(x, y, 0) = \frac{3}{4} + \frac{1}{4(1 + e^{(x+y)R})},
\]

where \( R \) is Reynolds number.

Using (6) with (7) for the functional coupled equation (1) and initial conditions (8) gives
where \( R \) is Reynolds number. These solutions are given in [6].
4. Experimental results

In order to verify numerically whether the proposed methodology lead to higher accuracy, we evaluate the numerical solutions using the \( n \)-term approximation. Tables 1 and 2 show the difference of analytical solution and numerical solution of the absolute error for Reynolds number of \( R = 50 \) and \( R = 100 \). We achieved a very good approximation with the actual solution of

Table 1
The numerical results for \( \phi_n(x, y, t) \) and \( \varphi_n(x, y, t) \) in comparison with the exact solution (8) for \( u(x, y, t) \) and \( v(x, y, t) \) when \( y = 1 \), for the approximate solution of Eq. (1)

\[
R = 50
\]

<table>
<thead>
<tr>
<th>( x )</th>
<th>( t )</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>u - \phi_n</td>
<td>)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>3.92097E-04</td>
<td>7.45129E-04</td>
<td>1.08103E-03</td>
<td>1.41912E-03</td>
<td>1.77878E-03</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>7.28716E-04</td>
<td>1.38450E-03</td>
<td>2.00767E-03</td>
<td>2.63353E-03</td>
<td>3.29701E-03</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>1.34824E-03</td>
<td>2.56042E-03</td>
<td>3.70922E-03</td>
<td>4.85927E-03</td>
<td>6.07018E-03</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>2.47380E-03</td>
<td>4.69430E-03</td>
<td>6.79092E-03</td>
<td>8.87288E-03</td>
<td>1.10407E-02</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>4.47043E-03</td>
<td>8.47191E-03</td>
<td>1.22322E-02</td>
<td>1.59013E-02</td>
<td>1.96540E-02</td>
<td></td>
</tr>
<tr>
<td>(</td>
<td>v - \varphi_n</td>
<td>)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>3.92728E-04</td>
<td>7.45940E-04</td>
<td>1.08194E-03</td>
<td>1.42056E-03</td>
<td>1.78211E-03</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>7.30893E-04</td>
<td>1.38729E-03</td>
<td>2.01080E-03</td>
<td>2.63849E-03</td>
<td>3.30853E-03</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>1.35569E-03</td>
<td>2.56995E-03</td>
<td>3.72037E-03</td>
<td>4.87624E-03</td>
<td>6.10987E-03</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>2.49885E-03</td>
<td>4.72617E-03</td>
<td>6.82661E-03</td>
<td>8.93023E-03</td>
<td>1.11763E-02</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>4.55222E-03</td>
<td>8.57526E-03</td>
<td>1.23912E-02</td>
<td>1.60908E-02</td>
<td>2.01994E-02</td>
<td></td>
</tr>
</tbody>
</table>

Table 2
The numerical results for \( \phi_n(x, y, t) \) and \( \varphi_n(x, y, t) \) in comparison with the exact solution (8) for \( u(x, y, t) \) and \( v(x, y, t) \) when \( y = 1 \), for the approximate solution of Eq. (1)

\[
R = 100
\]

<table>
<thead>
<tr>
<th>( x )</th>
<th>( t )</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>u - \phi_n</td>
<td>)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>2.70433E-06</td>
<td>5.15797E-06</td>
<td>7.95866E-06</td>
<td>1.18869E-05</td>
<td>1.82378E-05</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>9.43849E-06</td>
<td>1.80018E-05</td>
<td>2.77756E-05</td>
<td>4.14827E-05</td>
<td>6.36400E-05</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>3.29370E-05</td>
<td>6.28172E-05</td>
<td>9.69120E-05</td>
<td>1.44706E-04</td>
<td>2.21927E-04</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>1.14882E-04</td>
<td>2.19067E-04</td>
<td>3.37837E-04</td>
<td>5.04072E-04</td>
<td>7.72191E-04</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>4.00011E-04</td>
<td>7.62347E-04</td>
<td>1.17409E-03</td>
<td>1.74726E-03</td>
<td>2.66610E-03</td>
<td></td>
</tr>
<tr>
<td>(</td>
<td>v - \varphi_n</td>
<td>)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>2.70434E-06</td>
<td>5.15799E-06</td>
<td>7.95877E-06</td>
<td>1.18875E-05</td>
<td>1.82396E-05</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>9.43862E-06</td>
<td>1.80021E-05</td>
<td>2.77769E-05</td>
<td>4.14890E-05</td>
<td>6.36613E-05</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>3.29386E-05</td>
<td>6.28200E-05</td>
<td>9.69281E-05</td>
<td>1.44783E-04</td>
<td>2.22187E-04</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>1.14901E-04</td>
<td>2.19101E-04</td>
<td>3.38033E-04</td>
<td>5.05002E-04</td>
<td>7.75359E-04</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>4.00244E-04</td>
<td>7.62761E-04</td>
<td>1.17646E-03</td>
<td>1.75856E-03</td>
<td>2.70462E-03</td>
<td></td>
</tr>
</tbody>
</table>
the equations by using five terms only of the decomposition series derived above. However, many terms can be calculated in order to achieve a high level of accuracy of the decomposition method with help of Mathematica.

Numerical approximations show a high degree of accuracy and in most cases \( \phi_n \) and \( \varphi_n \), the \( n \)-term approximations for \( u \) and \( v \), respectively are accurate for quite low values of \( n \). The numerical results we obtained justify the advantage of this methodology. It is evident that the overall errors can be made smaller by adding new terms of the decomposition series.

Furthermore, as the decomposition method does not require discretization of the variables, i.e. time and space, it is not effected by computation round off errors and one is not faced with necessity of large computer memory and time. The accuracy of the decomposition method for the coupled nonlinear equations controllable and absolute errors are very small with the present choice of \( t, y, \) and \( x \).

5. Conclusions

In conclusion, the Adomian decomposition method was used for finding the exact and approximate solution of the STDBE (1). The method can be also easy to be extended to other nonlinear evaluation equations, with the aid of Mathematica (or Matlab, Maple, Reduce, etc.), the course of solving nonlinear evaluation equations can be carried out in computer. Three coupled nonlinear equations with initial conditions are discussed as demonstrations. It may be concluded that the Adomian methodology is very powerful and efficient technique in finding exact solutions for wide classes of problems. It is also worth noting to point out that the advantage of the decomposition methodology is the fast convergence of the solutions.

Clearly, the series solution methodology can also be applied to many other nonlinear problems. However, as we have seen in the previous sections, the decomposition method does not require linearization or perturbation for obtaining closed form solutions. Additionally, it does not need any discretization to get numerical solutions. For more implementation of the decomposition method, one can look at the Refs. [25–31] and references therein.

References
