Kuwait J. Sci. 46 (1) pp 1-14, 2019

Higher order splitting approaches in analysis of the Burgers equation

Murat Sari[1, *], Huseyin Tunc[1], Muaz Seydaoglu[2]
[1]*Dept. of Mathematics, Faculty of Arts and Science, Yildiz Technical University*
Istanbul 34220, Turkey
[2]*Dept. of Mathematics, Faculty of Arts and Science, Mus Alparslan University*
Mus 49100, Turkey
[*]*Corresponding author: sarim@yildiz.edu.tr*

Abstract

This article proposes some higher order splitting-up techniques based on the cubic B-spline Galerkin finite element method in analyzing the Burgers equation model. The strong form of both conservation and diffusion parts of the time-split Burgers equation have been considered in building the Galerkin approach. To integrate the corresponding ODE system, the Crank-Nicolson time discretization scheme is used. The proposed schemes are shown to be unconditionally stable. Three challenging examples have been considered that have changing values of the kinematic viscosity constant of the medium. Moreover, cases of shock waves of severe gradient are solved and compared with the exact solution and the literature. The qualitative and quantitative results demonstrate that our numerical approach has far higher accuracy than rival methods.

Keywords: Burgers equation; cubic B-spline; extrapolation; Galerkin method; Strang splitting.

1. Introduction

Nonlinear partial differential equations arise in many fields of science, particularly in applied mathematics, physics, engineering, mathematical biology, chemistry, and finance. One of the most important model equation is the Burgers equation. It represents various problems in a broad range of scientific fields, such as heat conduction (Cole, 1951), turbulence and shock waves (Burgers, 1948), longitudinal elastic waves in an isotropic solid (Pospelov, 1966), number theory (Pol, 1951), continuous stochastic processes (Cole, 1951), and so on.

Under certain conditions, and by considering the uniqueness and existence of solutions, the mathematical analysis of Burgers equation was discussed in the literature (Wang & Warnecke, 2003). The Burgers equation was exactly solved by using the Hopf-Cole transformation (Hopf, 1950; Cole, 1951) which converts the equation to a heat diffusion equation. In most of those cases, the solutions involve infinite series which may diverge or converge very slowly for relatively small values of the kinematic viscosity constant ε, which corresponds to steep wave fronts in the propagation of the dynamic wave forms.

Much effort has been spent in solving the Burgers equation over the last couple of decades. Since some exact solutions fail for small kinematic viscosity values (Miller, 1966), $\varepsilon < 0.01$, many researchers have suggested various numerical methods based on different approaches. These include, but are not limited to, the least-squares quadratic B-spline finite element method (Kutluay *et al.,* 2004), hybrid numerical scheme involving wavelets and finite differences (Jiwari, 2015), quadratic B-spline collocation method (Raslan, 2003), spline in tension approximation (Talwar *et al*., 2016), boundary element method (Bahadir & Saglam, 2005), various difference schemes (Liao & Zhu, 2011), lumped Galerkin method (Kutluay & Esen, 2004), high-order time integration formulae (Verma & Verma, 2015), local discontinuous Galerkin method (Shao *et al.,* 2011), a sixth-order CFD scheme (Sari & Gurarslan, 2009), higher order splitting methods (Seydaoglu *et al.*, 2016), differential quadrature method based on B-spline functions (Bashan *et al*., 2015).

This study proposes a Galerkin type finite element method (FEM) in which a strong form of both the conservation and diffusion parts of the equation is preferred rather than the weak form. The use of the strong form of the FEM in analyzing the advection-diffusion processes represented by the Burgers equation has some advantages in comparison to the latter. Note that the weak form and strong form are mathematically equivalent to each other, but computationally this is not the case. The weak form of the equation needs more complicated computers codes. Since the weak form of the model equation requires additional matrices for the residual term of the integration, this gives rise to excessive computational time and may therefore lead to loss of accuracy.

The splitting-up technique for the Burgers equation presented by Jain & Raja (1979) splits the Burgers equation into subproblems and solves each of them with the finite difference method. Similar strategies were considered in references (Jain & Holla, 1978; Jain

et al., 1992) using the cubic spline method for approximate solutions of the Burgers equation. Time and space splitting ideas were considered in reference Saka & Dag (2008). Here each submodel was solved numerically by a quintic B-spline collocation method. High order splitting methods were presented for non-autonomous perturbed parabolic equations in a work of Seydaoglu & Blanes (2014). Seydaoglu *et al.* (2016) presented a numerical solution of the Burgers equation through higher order splitting methods, and they observed order reductions for the Dirichlet, Neumann and Robin boundary conditions.

The outline of this paper is as follows. The governing model equation is explained in Section 2. The considered splitting methods and their implementation to the Burgers equation will be investigated in Section 3. Implementation of the Galerkin approach to the split equations and time integration procedure of the corresponding ODE system are given in Section 4. Some numerical illustrations are presented in Section 5. Section 6 consists of some concluding remarks.

2. Governing equation

Consider the one-dimensional Burgers equation representing the aforementioned problems into the following form:

$$u_t + uu_x = \varepsilon u_{xx}\,, \quad a \le x \le b \tag{1}$$

with the boundary conditions

$$u(a,t) = f_1(t), \quad t > 0$$
$$u(b,t) = f_2(t), \quad t > 0 \tag{2}$$

and initial condition

$$u(x,0) = g(x), \; a < x < b\,, \tag{3}$$

where ε is viscosity constant for ε > 0 and f_1, f_2 and g are known functions. The subscripts x and t indicate differentiations with respect to space and time, respectively.

3. Splitting the model

Let us split the Burgers Equation (1) into subproblems as follows

$$u_t = \varepsilon u_{xx}, \tag{4}$$
$$u_t = -uu_x. \tag{5}$$

The splitting strategy for the Burgers Equation (1) alternately involves solving the subproblems (4) and (5). Let the exact solutions (or a sufficiently accurate numerical approximation) for subproblems (4) and (5), respectively, be the maps ϕ_k^A and ϕ_k^B with time step k. Then one can approximate the solution of (1) for a sufficiently small k as

$$u(x,k) \approx \psi_k g(x), \tag{6}$$

where

$$\psi_k = \phi^A_{ka_1} \circ \phi^B_{kb_1} \circ \dots \circ \phi^A_{ka_s} \circ \phi^B_{kb_p} \circ \phi^A_{ka_{p+1}}, \tag{7}$$

or

$$\psi_k = \phi^B_{kb_1} \circ \phi^A_{ka_1} \circ \dots \phi^B_{kb_s} \circ \phi^A_{ka_p} \circ \phi^B_{kb_{p+1}}. \tag{8}$$

The coefficients a_i and b_i can be defined in the desired order by using the Baker-Campell- Hausdorff formula (Hairer *et al.,* 2006; Creutz & Gocksch, 1989; Suziki, 1990; Yoshida, 1990). The Lie-Trotter splitting method is given as follows:

$$T_k = \phi_k^{L_A} \circ \phi_k^{L_B} \text{ or } T_k = \phi_k^{L_B} \circ \phi_k^{L_A}, \tag{9}$$

and it is first order, i.e. $T_k = \phi_k^{(A+B)} + O(k^2)$. The well-known second order time symmetric method reads

$$S_k = \phi^A_{k/2} \circ \phi^B_k \circ \phi^A_{k/2}, \tag{10}$$

or

$$S_k = \phi^B_{k/2} \circ \phi^A_k \circ \phi^B_{k/2}, \tag{11}$$

which is referred to as the Strang splitting method. Notice that the scheme given with ABA pattern (7) firstly solves subproblem (4) and then subproblem (5) alternately, while the BAB pattern does the same in reverse order. To decide which pattern of the splitting schemes gives more efficient results, we apply the Strang splitting method with a different pattern to Example 1 in the numerical experiments.

However, the splitting schemes with real coefficients with a higher order than two necessarily have at least one negative coefficient (Blanes & Casas, 2005). Thus, one cannot use such schemes for the Burgers equation due to the Laplacian operator. To use the high order splitting method with real positive coefficients, we consider the extrapolation methods

$$\psi_k = \tfrac{4}{3}\phi_{k/2} \circ \phi_{k/2} - \tfrac{1}{3}\phi_k, \tag{12}$$

and

$$\psi_k = \tfrac{81}{40}\phi_{k/3} \circ \phi_{k/3} \circ \phi_{k/3} - \tfrac{16}{15}\phi_{k/2} \circ \phi_{k/2} + \tfrac{1}{24}\phi_k. \tag{13}$$

If one uses the Strang splitting method (10) instead of the main method ϕ_k considered in the extrapolation process, then a fourth-order method reads

$$\psi_k = \tfrac{4}{3}\phi^A_{k/4} \circ \phi^B_{k/2} \circ \phi^A_{k/2} \circ \phi^B_{k/2} \circ \phi^A_{k/4} - \tfrac{1}{3}\phi^A_{k/2} \circ \phi^B_k \circ \phi^A_{k/2}, \tag{14}$$

and a sixth-order method reads

$$\psi_k = \tfrac{81}{40}\phi^A_{k/6} \circ \phi^B_{k/3} \circ \phi^A_{k/3} \circ \phi^B_{k/3} \circ \phi^A_{k/3} \circ \phi^B_{k/3} \circ \phi^A_{k/6} - \tfrac{16}{15}\phi^A_{k/4} \circ \phi^B_{k/2} \circ \phi^A_{k/2} \circ \phi^B_{k/2} \circ \phi^A_{k/4} + \tfrac{1}{24}\phi^A_{k/2} \circ \phi^B_k \circ \phi^A_{k/2}. \tag{15}$$

Hansen & Ostermann (2009) presented a convergence analysis of the exponential splitting methods for the linear evolution equations, and they collected some extensions of their results. Note that similar results were derived independently in reference (Castella *et al*., 2009). Formally, one expects the same convergence results by considering extensions in the literature (Hansen & Ostermann, 2009; Castella *et al*., 2009) for the nonlinear equation by replacing all exponential terms to their corresponding nonlinear flows.

4. Numerical methods

To investigate solutions of problems (1) - (3) through (4) and (5), we have proposed the Galerkin finite element method in strong form with cubic B-spline basis functions for spatial approximation and the Crank-Nicolson method for the time integration of the resulted ordinary differential equation system.

4.1. Cubic B-spline basis functions

The interval [a,b] is partitioned into N finite elements. Each element has equal length h, and element nodes are defined as

$$a = x_0 < x_1 < \cdots < x_N = b,$$

where

$$x_{m+1} = x_m + h \quad (m = 0,1. \ldots, N-1).$$

$$\varphi_m(x) = \frac{1}{h^3} \begin{cases} (x - x_{m-2})^3 \\ h^3 + 3h^2(x - x_{m-1}) + 3h(x - x_{m-1})^2 - 3(x - x_{m-1})^3 \\ h^3 + 3h^2(x_{m+1} - x) + 3h(x_{m+1} - x)^2 - 3(x_{m+1} - x)^3 \\ (x_{m+2} - x)^3 \\ 0 \end{cases},$$

$$x \in \begin{cases} [x_{m-2}, x_{m-1}] \\ [x_{m-1}, x_m] \\ [x_m, x_{m+1}] \\ [x_{m+1}, x_{m+2}] \\ \text{otherwise.} \end{cases} \tag{16}$$

The corresponding cubic —spline basis functions include the set of splines $\{\varphi_{-1}, \varphi_0, \ldots, \varphi_{N+1}\}$, and the global approximation function $\tilde{u}_N(x,t)$ can be expressed as

$$\tilde{u}_N(x,t) = \sum_{m=-1}^{N+1} \beta_m(t)\varphi_m(x), \tag{17}$$

where $\beta_m(t)$ are the time part of approximation function $\tilde{u}_N(x,t)$. It will be determined from the time approximation.

To compute element matrices easily, the local coordinate system in (16) is required. As is the case in the literature (Soliman, 2012; Karakoc *et al*., 2015), letting

$$\sigma = x - x_m,$$

where x in

$[x_m, x_{m+1}]$ $(m = 0,1. \ldots, N-1)$, $0 \le \sigma \le h$, and the basis functions will be in the form

$$\varphi_m(\sigma) = \frac{1}{h^3} \begin{cases} \sigma^3 \\ h^3 + 3h^2\sigma + 3h\sigma^2 - 3\sigma^3 \\ h^3 + 3h^2(h - \sigma) + 3h(h - \sigma)^2 - 3(h - \sigma)^3 \\ (h - \sigma)^3 \\ 0. \end{cases} \tag{18}$$

Each finite element $[x_m, x_{m+1}]$ is covered by the set of four cubic B-splines $\{\varphi_{m-1}, \varphi_m, \varphi_{m+1}, \varphi_{m+2}\}$. Table 1 shows the value of φ_m, φ_m' and φ_m'' at the end pointsof elements $[x_m, x_{m+1}]$.

The local approximation function on the element $[x_m, x_{m+1}]$ is defined as follows:

$$\tilde{u}_N(x,t) = \sum_{i=m-1}^{m+2} \beta_i(t)\varphi_i(x) \tag{19}$$

Values of the local approximation function $\tilde{u}_N(x,t)$ and its first two derivatives at the end points of the interval $[x_m, x_{m+1}]$ are defined in terms of the time dependent quantities β_m (t) using both (19) and Table 1. The corresponding values then become:

$$\begin{aligned} &\tilde{u}_N(x_m,t) = \beta_{m-1} + 4\beta_m + \beta_{m+1}, \\ &\tilde{u}_N(x_{m+1},t) = \beta_m + 4\beta_{m+1} + \beta_{m+2}, \\ &\tilde{u}_N'(x_m,t) = \tfrac{3}{h}(\beta_{m+1} - \beta_{m-1}), \\ &\tilde{u}_N'(x_{m+1},t) = \tfrac{3}{h}(\beta_{m+2} - \beta_m), \\ &\tilde{u}_N''(x_m,t) = \tfrac{6}{h^2}(\beta_{m+1} - 2\beta_m + \beta_{m-1}), \\ &\tilde{u}_N''(x_{m+1},t) = \tfrac{6}{h^2}(\beta_{m+2} - 2\beta_{m+1} + \beta_m). \end{aligned} \tag{20}$$

Now it is time to apply the Galerkin method to both the diffusion part (4) and conservation part (5). By considering element $[x_m, x_{m+1}]$, let us multiply Equations (4) and (5) by a test function v and integrate over the interval $[x_m, x_{m+1}]$. One can then write:

$$\int_{x_m}^{x_{m+1}} v(u_t - \varepsilon u_{xx})dx = 0, \tag{21}$$

$$\int_{x_m}^{x_{m+1}} v(u_t + uu_x)dx = 0. \tag{22}$$

The test function v is selected to be equal to the cubic B-spline basis functions. This type of procedure is known as the Galerkin approach in the finite element method. Using (19) and (18), Equations (21) and (22) yield the following relations:

$$\sum_{j=m-1}^{m+2} \left[\int_0^h \varphi_i\varphi_j d\sigma\right] \frac{d\beta_j^e}{dt} - \varepsilon \sum_{j=m-1}^{m+2} \left[\int_0^h \varphi_i\varphi_j'' d\sigma\right] \beta_j^e = 0 \tag{23}$$

and

$$\sum_{j=m-1}^{m+2} \left[\int_0^h \varphi_i\varphi_j d\sigma\right] \frac{d\beta_j^e}{dt} + \sum_{j=m-1}^{m+2} \sum_{k=m-1}^{m+2} \left[\int_0^h \varphi_i\varphi_j' \varphi_k d\sigma\right] \beta_k^e\beta_j^e = 0 \tag{24}$$

or in a matrix notation

$$M^e \frac{d\beta^e}{dt} - \varepsilon \mathrm{K}^e \beta^e = 0, \quad (25)$$

and

$$M^e \frac{d\beta^e}{dt} + \beta_{\mathrm{k}}^{\mathrm{e}\,\mathrm{T}} L^e \beta^e = 0, \quad (26)$$

where

$$\mathrm{M}_{\mathrm{ij}}^{\mathrm{e}} = \int_0^h \varphi_i \varphi_j d\sigma,$$
$$\mathrm{K}_{\mathrm{ij}}^{\mathrm{e}} = \int_0^h \varphi_i \varphi_j'' \, d\sigma,$$
$$\mathrm{L}_{\mathrm{ijk}}^{\mathrm{e}} = \int_0^h \varphi_i \varphi_j'' \, \varphi_k d\sigma, \quad (27)$$
$$\beta^e = (\beta_{\mathrm{m}-1}, \beta_{\mathrm{m}}, \beta_{\mathrm{m}+1}, \beta_{\mathrm{m}+2})^{\mathrm{T}},$$

and $i,j,k=m-1,m,m+1,m+2$ for the element $[x_m,x_{m+1}]$. In (25) and (26); M^e and K^e are (4×4) matrices and are independent of time. L^e is the (4×4×4) matrix, and L^e can be transformed to a time dependent matrix R by using

$$R_{ij}^e = \sum_{k=i-1}^{i+2} L_{ijk}^e \beta_k^e. \quad (28)$$

After the assembling process of each element, the matrix form will finally be

$$M^* \frac{d\beta}{dt} - \varepsilon K^* \beta = 0, \quad (29)$$

and

$$M^* \frac{d\beta}{dt} + R^* \beta = 0, \quad (30)$$

for the diffusion and conservation parts, respectively. Here M^*, R^* and K^* are (N+3)×(N+3) matrices and $\beta=(\beta_{(-1)},\beta_0,\ldots,\beta_{(N+1)})^T$ is the unknown time approximation vector. Consideration of (29) and (30) gives a system of ordinary differential equations, which are solved using the Crank-Nicolson time integration scheme, as discussed in the following section.

4.2 The Crank-Nicolson scheme

The time discretization procedure of the ODE system (29) and (30) can be explained as follows (Tunc, 2017):

$$\beta = \frac{1}{2}(\{\beta\}_s + \{\beta\}_{s+1}),$$
$$\frac{d\beta}{dt} = \frac{1}{dt}(\{\beta\}_{s+1} - \{\beta\}_s), \quad (31)$$

where t_{s+1}-t_s=dt. Use of (31) makes Equations (29) and (30) as in following forms:

$$[M^* - \frac{dt}{2} \varepsilon K^*)]\{\beta\}_{s+1} = [M^* + \frac{dt}{2} \varepsilon K^*)]\{\beta\}_s \quad (32)$$

and

$$[M^* + \frac{dt}{2} R_{s+1}^*]\{\beta\}_{s+1} = [M^* - \frac{dt}{2} R_s^*]\{\beta\}_s, \quad (33)$$

where matrices M^* and K^* are independent of time while R^* depends on time. Moreover, to cope with difficulties in the non-linearity in the time dependent matrix $R^*_{(s+1)}$, we used the following correction relation:

$$\{\beta\}_{s+1} = \frac{1}{2}(\{\beta\}_s + \{\beta\}_s^*), \quad (34)$$

where $\{\beta\}_s^*$ is the corrected from of $\{\beta\}_s$. Note that to find first approximation $\{\beta\}_{\mathrm{l}}^0$, we need to solve the following $(N+3)\times(N+3)$ linear system:

$$\tilde{u}(x_m,0) = g(x_m) = \{\beta\}_{\mathrm{m}+1}^0 + 4\{\beta\}_{\mathrm{m}}^0 + \{\beta\}_{\mathrm{m}-1}^0,$$
$$\tilde{u}_N'(x_0,0) = g'(x_0) = \frac{3}{h}(\{\beta\}_1^0 - \{\beta\}_{-1}^0)$$
$$\tilde{u}_N'(x_N,0) = g'(x_N) = \frac{3}{h}(\{\beta\}_{\mathrm{N}+1}^0 - \{\beta\}_{\mathrm{N}-1}^0).$$

By using the recursive relation in (32) - (33) and the corrector relation in (34), the diffusion and conservation parts of the Burgers equation is solved under the consideration of the splitting formulae given in Equations (10), (14) and (15). The algorithm of the proposed methods has been presented in Algorithm 1. Computer codes of the algorithm have been produced in MATLAB R2016a.

Table 1. Values of approximate function and its derivatives at the end points of the element.

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

Algorithm 1. Algorithm to find numerical solutions of the Burgers equation by using splitting methods (7) using (6) for a time step

Begin
Initialize spatial interval, time interval, element numbers
Initialize initial and boundary conditions
Procedure
1. Calculate local matrices given in (25)-(26).
2. Decide the type of the splitting approach (say Strang and ABA).
3. Produce a time integration loop for the discrete relations (32)-(33).
4. Evaluate the solution of the diffusion part (32) by considering the given initial condition.
5. Evaluate the solution of the conservation part (33) by considering the initial guess which is produced in the last step.
6. Evaluate the solution of the diffusion part (32) by considering the initial guess which is produced in the last step.
7. The produced solution in step 6 is the final solution.
8. End of the time integration loop.

End

5. Stability analysis

The von Neumann stability analysis is one of the most widely used methods for analyzing the stability of numerical methods that are meant to approximately solve partial differential equations (Kutluay *et al.*, 2004; Soliman, 2012, Sari & Tunc, 2017). As pointed out in the corresponding literature, this stability method is more suitable for the algebraic equation system studied here.

The stability of the current numerical approach directly depends on the individual stability of the diffusion and conservation parts of the Burgers equation. Thus, to understand stability condition of each subproblem, we have analyzed Equations (32)-(33)using the von Neumann theory with the Fourier growth factor defined by

$$\beta_s^n = \tilde{\beta}^n e^{is\tau h}, \quad (35)$$

where τ and h stand for the mode number and the element size, respectively, which are selected for recursive approximations (32)-(33). To evaluate a typical row of (33), $\beta_{(s+1)}$ and β_s values in the time dependent matrices $R^*_{(s+1)}$and R_s^* are considered to be locally constant and equal to p, as is the case in the literature (Sari & Tunc, 2017; Tunc, 2017). The stability analysis of the diffusion and the conservation parts is performed, respectively, as in the following subsections.

5.1 Stability of the Diffusion Part

By considering the entries of the included matrices in Equation (32), a typical row of Equation (32) can be stated as

$$\begin{aligned} &c_1\beta_{s-3}^{n+1} + c_2\beta_{s-2}^{n+1} + c_3\beta_{s-1}^{n+1} + c_4\beta_s^{n+1} + \\ &c_5\beta_{s+1}^{n+1} + c_6\beta_{s+2}^{n+1} + c_7\beta_{s+3}^{n+1} = c_8\beta_{s-3}^{n} + \\ &c_9\beta_{s-2}^{n} + c_{10}\beta_{s-1}^{n} + c_{11}\beta_s^{n} + c_{12}\beta_{s+1}^{n} + \\ &c_{13}\beta_{s+2}^{n} + c_{14}\beta_{s+3}^{n}, \end{aligned} \quad (36)$$

where

$$\begin{aligned} &c_1 = r_1 - 3r_2, c_2 = 120r_1 - 72r_2, \\ &c_3 = 1191r_1 - 45r_2, \quad c_4 = 2416r_1 + 240r_2, \\ &c_5 = 1191r_1 - 45r_2, c_6 = 120r_1 - 72r_2, \\ &c_7 = r_1 - 3r_2, c_8 = r_1 + 3r_2, \\ &c_9 = 120r_1 + 72r_2, \quad c_{10} = 1191r_1 + 45r_2, \\ &c_{11} = 2416r_1 - 240r_2, c_{12} = 1191r_1 + 45r_2 \\ &c_{13} = 120r_1 + 72r_2, c_{14} = r_1 + 3r_2, \\ &r_1 = \frac{h}{140}, \; r_2 = \frac{\varepsilon}{80h}\,\mathrm{dt}. \end{aligned} \quad (37)$$

Substituting (35) into (36) and using the Euler expansion for exponential terms leads to the following relation:

$$(w + 120r_2)\tilde{\beta}^{n+1} = (w^* - 120r_2)\tilde{\beta}^n, \quad (38)$$

where

$$\begin{aligned} w = &(r_1 - 3r_2)\cos(3\tau h) + \\ &(120r_1 - 72r_2)\cos(2\tau h) + \\ &(1191r_1 - 45r_2)\cos(\tau h) + 1208r_1, \end{aligned}$$

$$\begin{aligned} w^* = &(r_1 + 3r_2)\cos(3\tau h) + (120r_1 + \\ &72r_2)\cos(2\tau h) + (1191r_1 + 45r_2)\cos(\tau h) + 1208r_1. \end{aligned}$$

Equation (38) can be rewritten as follows:

$$\tilde{\beta}^{n+1} = \frac{(w^* - 120r_2)}{(w + 120r_2)}\tilde{\beta}^n = z\tilde{\beta}^n, \quad (39)$$

where z is the amplification factor of the iteration (32). Iteration (32) is stable if the modulus of the amplification factor is less than or equal to one, i.e. $|z| \le 1$. By considering $r_1 \ll 1$, the following inequalities always hold $|w^* - 120r_2| \le |w + 120r_2|$ and $|z| \le 1$.

Thus, iteration (32) for the diffusion system is unconditionally stable.

5.2 Stability of the conservation part

A typical row of Equation (33) can be expressed as given in Equation (36) with the following coefficients:

$$\begin{aligned} &c_1 = r_1 - 3r_2, c_2 = 120r_1 - 168r_2, \\ &c_3 = 1191r_1 - 735r_2, c_4 = 2416r_1, \\ &c_5 = 1191r_1 + 735r_2, c_6 = 120r_1 + 168r_2 \\ &c_7 = r_1 + 3r_2 - 3r_3, c_8 = r_1 + 3r_2, \\ &c_9 = 120r_1 + 168r_2, \quad c_{10} = 1191r_1 + 735r_2, \\ &c_{11} = 2416r_1 - 240r_3, \\ &c_{12} = 1191r_1 - 735r_2, \\ &c_{13} = 120r_1 - 168r_2, c_{14} = r_1 - 3r_2, \\ &r_1 = \frac{h}{140}, \; r_2 = \frac{pdt}{80}. \end{aligned} \quad (40)$$

Writing (35) into (36) with the considered coefficients (40) and with the use of the Euler expansion leads to

$$(w + iw^*)\tilde{\beta}^{n+1} = (w - iw^*)\tilde{\beta}^n, \quad (41)$$

where

$$w = r_1\cos(3\tau h) + 120r_1\cos(2\tau h) + 1191\, r_1\cos(\tau h) + 1208r_1,$$

$$w^* = 3r_2\sin(3\tau h) + 168r_2\sin(2\tau h) + 735r_2\sin(\tau h).$$

It is obvious that the amplification factor z=1 and satisfies the stability condition. Hence, iteration (33) for the conservation part of Equation (1) is unconditionally stable. In conclusion, the splitting iteration system (32)-(33) is thus unconditionally stable.

6. Numerical experiments

We present the results for the following schemes with real coefficients:

Strang: The second-order symmetric Strang splitting method (10);

EX4: The fourth-order extrapolation method (14); and,

EX6: The sixth-order extrapolation method (15).

Example 1 (Sari & Gurarslan, 2009) Let us consider Burgers Equation (1) with the initial condition

$$u(x,0) = g(x) = \sin \pi x, \; 0 < x < 1 \quad (42)$$

and homogeneous Dirichlet boundary conditions

$$u(0,t) = 0, \; t > 0, \quad (43)$$

$$u(1,t) = 0, \; t > 0. \quad (44)$$

The exact solution of (1) under the consideration of cases (42)-(44) given by Cole (1951) is

$$u(x,t) = 2\pi\varepsilon \frac{\sum_{n=1}^{\infty} a_n \exp(-n^2\pi^2\varepsilon t) nsin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n \exp(-n^2\pi^2\varepsilon t) cos(n\pi x)} \quad (45)$$

with the Fourier coefficients

$$a_0 = \int_0^1 exp\{-(2\pi\varepsilon)^{-1}[1-\cos(\pi x)]\}dx,$$
$$a_n = 2\int_0^1 exp\{-(2\pi\varepsilon)^{-1}[1-\cos(\pi x)]\}\cos(n\pi x)dx.$$

In all numerical experiments, we prefer to use pattern (10) instead of pattern (11). If one solves nonlinear advection part two times, then the computational cost is higher and the accuracy is less than pattern (10) because of the necessity of the correction relation in the nonlinear advection part. Table 2 shows the comparison of the produced results using the present approach with the literature (Mukundan & Awasthi, 2015) and exact solutions for various spatial points at t=0.5. As seen in the table, the present study is more accurate and more economical, with a far smaller number of elements, in comparison to their results. In addition, it is shown that the pattern ABA produces more accurate results than the pattern BAB in Table 2.

Table 2. Comparison of the produced results at $t = 0.5$ for $\varepsilon = 1,\ dt = 0.001$.

x	**EX6** Splitting Present N=40	**EX4** Splitting Present N=40	Strang Splitting Present (ABA)-N=40	Strang Splitting Present (BAB)-N=40	Mukundan & Awasthi (2015) N=100	Exact
$x = 0.1$	0.00221300	0.00221299	0.00221298	0.00221292	0.002213	0.00221301
$x = 0.2$	0.00421007	0.00421005	0.00421003	0.00420990	0.004209	0.00421007
$x = 0.3$	0.00579612	0.00579610	0.00579606	0.00579589	0.005795	0.00579612
$x = 0.4$	0.00681592	0.00681588	0.00681585	0.00681565	0.006815	0.00681592
$x = 0.5$	0.00716920	0.00716917	0.00716913	0.00716892	0.007168	0.00716921
$x = 0.6$	0.00682072	0.00682069	0.00682066	0.00682045	0.006820	0.00682073
$x = 0.7$	0.00580390	0.00580387	0.00580384	0.00580367	0.005803	0.00580390
$x = 0.8$	0.00421785	0.00421783	0.00421780	0.00421768	0.004217	0.00421785
$x = 0.9$	0.00221781	0.00221780	0.00221779	0.00221772	0.002218	0.00221781

Table 3. Comparison of the produced results at $t = 2.3$ for $\varepsilon = 0.1, dt = 0.01$.

x	**EX6** Splitting Present N=50	**EX4** Splitting Present N=50	Strang Splitting Present N=50	Mukundan & Awasthi (2015) N=100	Exact
$x = 0.1$	0.0221396	0.0221397	0.0221395	0.02253	0.0221396
$x = 0.2$	0.0427956	0.0427957	0.0427954	0.04357	0.0427956
$x = 0.3$	0.0604313	0.0604314	0.0604310	0.06155	0.0604313
$x = 0.4$	0.0734431	0.0734432	0.0734426	0.07485	0.0734431
$x = 0.5$	0.0802310	0.0802311	0.0802302	0.08182	0.0802310
$x = 0.6$	0.0793988	0.0793988	0.0793977	0.08104	0.0793988
$x = 0.7$	0.0701068	0.0701067	0.0701055	0.07161	0.0701068
$x = 0.8$	0.0525198	0.0525196	0.0525186	0.05368	0.0525198
$x = 0.9$	0.0281740	0.0281739	0.0281733	0.02881	0.0281740

Table 3 shows the comparison of the present numerical solution with the exact solution and the literature (Mukundan & Awasthi, 2015). The kinematic viscosity constant is chosen to be ε=0.1. The results are produced for the parameters taken to be dt=0.01 and h=0.02. Even as we consider a smaller number of spatial elements, accuracy of the present method is far higher than the literature (Mukundan & Awasthi, 2015) at every spatial point. The presented results in Table 4 are compared with the literature (Bahadir & Saglam, 2005; Sari & Gurarslan, 2009) and the exact solution. Even with the use of fewer time elements, the comparison revealed that the suggested technique is able to produce more accurate results than the corresponding literature (Bahadir & Saglam, 2005; Sari & Gurarslan, 2009). In the comparison, responses of the physical system have been observed for the elapsed times of t=0.5, t=2.0, and t=4.0 at various positions for h=0.01 (Table 4).

Now it is time to deal with far smaller kinematic viscosity constants. A comparison of the currently produced solutions has been carried out with the literature, and the exact solution for two different viscosity values, ε=0.004 and ε=0.003, respectively (Tables 5-6). The present study reveals that even by using fewer time elements in Table 5, one can find more accurate results than the literature (Jiwari, 2015).

In the work of Aksan (2006), the model equation with conditions (23)-(25) is solved using the quadratic B-spline FEM in the weak form with the Newton iteration for nonlinear systems. In another work (Dag *et al.*, 2005), the researchers used the weak form of the governing equation, the cubic B-spline basis approach

and the first-order splitting approach. The computed results in Table 7 show that the present method is more accurate for a far smaller number of time elements than the results of references (Aksan, 2006; Dag *et al.*, 2005).

Table 8 is another comparison which was completed with various schemes given in the literature (Kutluay *et al.*, 1999; Tsai *et al.*, 2017) with the currently proposed schemes in terms of maximum error norms. The compared results are taken from Table 4 of the reference Tsai *et al*. (2017). As realized from Table 8, the current results are far more accurate than the literature (Tsai *et al*., 2017; Kutluay *et al*., 1999). The numerical method is seen to have a high capacity in capturing gradual nonlinear steep behaviour, $\varepsilon \ll 1$. Figures 1 (a) and (b) show the splitting-up solutions of Example 1 with small parameter values, ε=0.001, ε=0.0005, respectively. In Figure 2, the present numerical methods are also compared in terms of computational costs, i.e. CPU times of the iterations are demonstrated for various number of spatial elements N. Among those methods, the Strang approach is the most economical. If one prescribes boundary conditions, error terms are generally not uniformly bounded on the interval [0 ,T] in the infinite dimensional space, so it is no longer possible to establish a guarantee of convergence order (Hansen & Ostermann, 2009; Seydaoglu *et al*., 2016). Thus, order reductions occured for higher order splitting methods when the Dirichlet boundary conditions were impossed.

In Figure 3(a), we compare the efficiency of the present methods given in Table 2 at the final time t=3. We demonstrate the infinity error norm versus the number of evaluations of ϕ_k^A, which usually requires the more costly computation for several step sizes. As seen in Figure 3(a), in spite of the correct convergence orders not being obtained, the high order extrapolation methods produce more reliable results in terms of both accuracy and computational cost.

Table 4. Comparison of the produced results for $\varepsilon = 0.01$.

x	t	**EX6** Splitting Present dt=0.004	**EX4** Splitting Present dt=0.004	Strang Splitting Present dt=0.004	Bahadir & Saglam, (2005) $dt = 0.001$	Sari & Gurarslan, (2009) $dt = 0.001$	Exact
$x = 0.1$	$t = 0.50$	0.1211435314	0.1211435314	0.1211416234	0.12079	0.12114	0.1211435315
	$t = 2.00$	0.0429637769	0.0429637769	0.0429634430	0.04300	0.04295	0.0429637769
	$t = 4.00$	0.0231042327	0.0231042327	0.0231041297	0.02324	0.02310	0.0231042327
$x = 0.3$	$t = 0.50$	0.3602710556	0.3602710556	0.3602669996	0.36113	0.36027	0.3602710559
	$t = 2.00$	0.1288398903	0.1288398903	0.1288389190	0.12887	0.12882	0.1288398903
	$t = 4.00$	0.0693082904	0.0693082904	0.0693079840	0.06935	0.06930	0.0693082904
$x = 0.5$	$t = 0.50$	0.5886957730	0.5886957729	0.5886945639	0.59559	0.58870	0.5886957735
	$t = 2.00$	0.2145580542	0.2145580542	0.2145565380	0.21468	0.21455	0.2145580543
	$t = 4.00$	0.1154947563	0.1154947560	0.1154942553	0.11550	0.11549	0.1154947563
$x = 0.7$	$t = 0.50$	0.7934934046	0.7934934039	0.7935031783	0.81257	0.79354	0.7934934058
	$t = 2.00$	0.2999977673	0.2999977659	0.2999958750	0.30075	0.29999	0.2999977677
	$t = 4.00$	0.1612146519	0.1612146463	0.1612140083	0.16125	0.16121	0.1612146543
$x = 0.9$	$t = 0.50$	0.9381067387	0.9381059346	0.9381462298	0.97184	0.93822	0.9381083431
	$t = 2.00$	0.3732772096	0.3732774294	0.3732786012	0.37452	0.37328	0.3732776288
	$t = 4.00$	0.1660587273	0.1660588088	0.1660571831	0.16515	0.16605	0.1660587216

Table 5. Comparison of the produced results for the parameter values $\varepsilon = 0.004, h = 0.01$.

x	t	**EX6** Splitting Present $N = 100, dt = 0.0125$	**EX4** Splitting Present $N = 100, dt = 0.0125$	Strang Splitting Present $N = 100, dt = 0.0125$	Jiwari (2015) $dt = 0.001$	Exact
$x = 0.25$	$t = 1$	0.18890403	0.18890403	0.18888074	0.18891	0.18890403
	$t = 5$	0.04697225	0.04697225	0.04697036	0.04697	0.04697225
	$t = 10$	0.02421935	0.02421935	0.02421883	0.02422	0.02421935
	$t = 15$	0.01631540	0.01631540	0.01631517	0.01632	0.01631540
$x = 0.5$	$t = 1$	0.37597616	0.37597617	0.37594050	0.37598	0.37597616
	$t = 5$	0.09393781	0.09393781	0.09393407	0.09394	0.09393781
	$t = 10$	0.04843716	0.04843716	0.04843613	0.04843	0.04843716
	$t = 15$	0.03259459	0.03259459	0.03259412	0.03259	0.03259459
$x = 0.75$	$t = 1$	0.55883376	0.55882869	0.55882287	0.55883	0.55883764
	$t = 5$	0.14088686	0.14088685	0.14088137	0.14089	0.14088686
	$t = 10$	0.07220247	0.07220246	0.07220095	0.07221	0.07220247
	$t = 15$	0.04677529	0.04677529	0.04677452	0.04678	0.04677529

Table 6. Comparison of the produced results for $\varepsilon = 0.003$ and $h = 0.01$.

x	t	**EX6** Splitting Present $N = 100$, $dt = 0.0125$	**EX4** Splitting Present $N = 100$, $dt = 0.0125$	Strang Splitting Present $N = 100$, $dt = 0.0125$	Jiwari (2015) $dt = 0.001$	Exact
$x = 0.25$	$t = 1$	0.18901910	0.18901910	0.18899538	0.18902	0.18901910
	$t = 5$	0.04698094	0.04698094	0.04697901	0.04698	0.04698094
	$t = 10$	0.02422174	0.02422174	0.02422121	0.02422	0.02422174
	$t = 15$	0.01631712	0.01631712	0.01631688	0.01631	0.01631712
$x = 0.5$	$t = 1$	0.37622719	0.37622719	0.37619067	0.37623	0.37622719
	$t = 5$	0.09395531	0.09395531	0.09395150	0.09396	0.09395531
	$t = 10$	0.04844299	0.04844299	0.04844194	0.04844	0.04844299
	$t = 15$	0.03263170	0.03263170	0.03263122	0.03263	0.03263170
$x = 0.75$	$t = 1$	0.55927734	0.55927597	0.55925619	0.55928	0.55927734
	$t = 5$	0.14091634	0.14091634	0.14091072	0.14092	0.14091634
	$t = 10$	0.07260297	0.07260297	0.07260142	0.07261	0.07260298
	$t = 15$	0.04838641	0.04838641	0.04838568	0.04839	0.04838642

Table 7. Comparison of the produced results for $\varepsilon = 0.1$ and $h = 0.0125$.

x	t	**EX6** Splitting Present $dt = 0.001$	**EX4** Splitting Present $dt = 0.001$	Strang Splitting Present $dt = 0.001$	Dag *et al.* (2005) $dt = 0.0001$	Aksan (2006) $dt = 0.0001$	Exact
	t=0.4	0.3088942	0.3088942	0.3088942	0.30890	0.30891	0.3088942
	t=0.6	0.2407390	0.2407390	0.2407390	0.24074	0.24075	0.2407390
x=0.25	t=0.8	0.1956756	0.1956756	0.1956756	0.19568	0.19568	0.1956756
	t=1.0	0.1625648	0.1625648	0.1625648	0.16257	0.16257	0.1625649
	t=3.0	0.0272023	0.0272023	0.0272023	0.02720	0.02721	0.0272023
	t=0.4	0.5696324	0.5696324	0.5696325	0.56964	0.56969	0.5696325
	t=0.6	0.4472055	0.4472055	0.4472055	0.44721	0.44723	0.4472055
x=0.50	t=0.8	0.3592360	0.3592360	0.3592360	0.35924	0.35926	0.3592361
	t=1.0	0.2919159	0.2919159	0.2919159	0.29191	0.29193	0.2919160
	t=3.0	0.0402049	0.0402049	0.0402049	0.04020	0.04021	0.0402049
	t=0.4	0.6254379	0.6254379	0.6254376	0.62541	0.62543	0.6254379
	t=0.6	0.4872150	0.4872150	0.4872148	0.48719	0.48723	0.4872150
x=0.75	t=0.8	0.3739218	0.3739218	0.3739217	0.37390	0.37394	0.3739218
	t=1.0	0.2874745	0.2874745	0.2874744	0.28746	0.28750	0.2874744
	t=3.0	0.0297721	0.0297721	0.0297721	0.02977	0.02978	0.0297721

Table 8. Comparison of maximum error norms of various schemes for $\varepsilon = 0.01, h = 0.0125$.

x	t	EFDM Kutluay *et al.* (1999) $dt = 0.0001$	EEFDM Kutluay *et al.* (1999) $dt = 0.0001$	TFPM Tsai *et al.* (2017) $dt = 0.0001$	Strang Splitting Present $dt = 0.001$	**EX4** Splitting Present $dt = 0.001$	**EX6** Splitting Present $dt = 0.001$
$x = 0.25$	$t = 0.4$	5.54E-4	1.60E-5	9.28E-6	3.40E-08	2.22E-09	1.24E-09
	$t = 0.6$	3.49E-4	1.10E-5	1.05E-5	1.41E-08	2.21E-09	1.05E-09
	$t = 0.8$	2.46E-4	4.40E-6	8.46E-6	4.27E-09	1.06E-09	8.65E-10
	$t = 1.0$	1.85E-4	5.10E-6	5.12E-6	1.11E-10	5.67E-10	6.27E-10
	$t = 3.0$	2.23E-4	2.30E-6	6.35E-6	2.33E-09	3.53E-10	2.16E-11
$x = 0.50$	$t = 0.4$	5.22E-4	7.50E-6	1.44E-5	1.40E-07	4.55E-10	4.54E-09
	$t = 0.6$	4.46E-4	4.50E-6	7.61E-6	7.06E-08	4.53E-09	3.28E-09
	$t = 0.8$	3.56E-4	3.90E-6	2.57E-6	3.58E-08	8.39E-09	1.96E-09
	$t = 1.0$	2.96E-4	4.00E-6	1.14E-5	1.55E-08	9.08E-09	9.75E-10
	$t = 3.0$	3.49E-5	5.10E-6	1.09E-5	4.74E-09	1.49E-10	5.66E-11
$x = 0.75$	$t = 0.4$	1.12E-4	1.80E-5	4.41E-5	2.92E-07	3.83E-08	2.59E-09
	$t = 0.6$	2.05E-4	5.00E-6	7.80E-5	2.02E-07	2.52E-08	4.60E-09
	$t = 0.8$	2.62E-4	1.80E-6	8.28E-5	1.55E-07	8.42E-09	4.69E-09
	$t = 1.0$	2.44E-4	5.60E-6	7.31E-5	1.13E-07	1.95E-10	3.65E-09
	$t = 3.0$	3.21E-5	2.10E-6	9.26E-6	4.87E-09	2.87E-10	7.14E-11

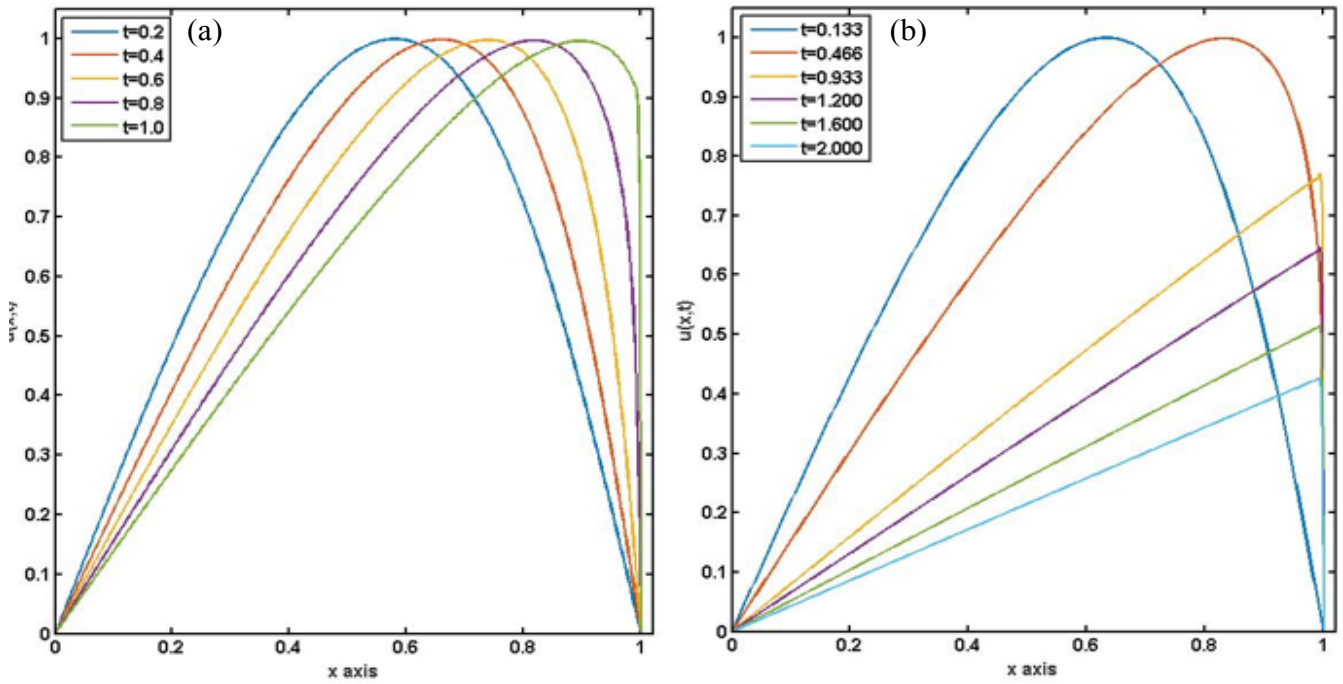

Fig. 1. Numerical solution of Example 1with the Strang splitting approach at different times for the parameters (a) ε =0.001, h=0.0025 and dt=0.002 and (b) ε =0.0005, h=0.002 and dt=0.0013

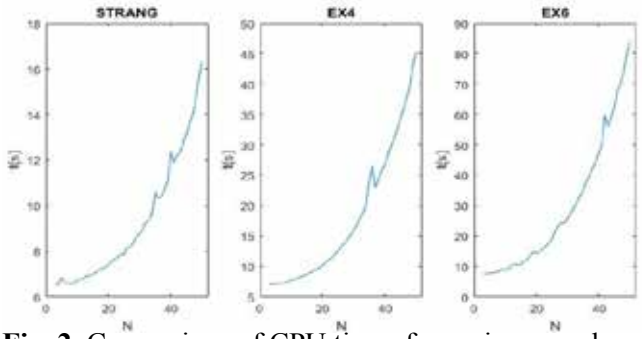

Fig. 2. Comparison of CPU times for various number of spatial elements and dt=0.001

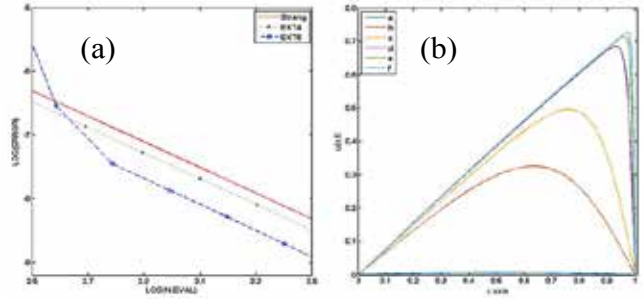

Fig. 3. a) Error versus number of evaluations of ϕ_k^A for the numerical solution of Example 1 at t=3,ε=0.1 and h=0.0125. b) Numerical solution of Example 2 at t=1 with different kinematic viscosity constants a) ε=0.5 b) ε=0.1 c) ε=0.05 d) ε=0.01 e) ε=0.005 f) ε=0.003

Example 2 (Sari & Tunc, 2017) Let us now consider the Burgers equation (1) with initial condition

$$u(x,0) = g(x) = 4x(1-x), \quad 0 < x < 1 \quad (46)$$

and homogeneous boundary conditions

$$u(0,t) = 0, \ t > 0 \quad (47)$$

$$u(1,t) = 0, \ t > 0. \quad (48)$$

The exact solution of (1) under the consideration of cases (46) - (48) is given by Cole (1951) as in (45) but with the Fourier coefficients

$$a_0 = \int_0^1 exp\{-x^2(3\varepsilon)^{-1}(3-2x)\}dx$$

$$a_n = 2\int_0^1 exp\{-x^2(3\varepsilon)^{-1}(3-2x)\}\cos(n\pi x)dx.$$

Table 9 includes the comparison of numerical solutions and with the exact solution with kinematic viscosity ε=1. The calculated results in Table 9 are more accurate than the literature (Kutluay *et al.*, 1999; Shao *et al.*, 2015). To produce the results, far fewer elements in time in comparison to the corresponding references have been used.

Table 10 gives a comparison of the present results with the literature (Kutluay *et al.*, 2004; Kutluay & Esen, 2004) and the exact solution. The current numerical solutions are more accurate than the corresponding literature when the advection is more dominant to the diffusion, ε=0.1 and h=0.0125. The presently calculated solutions are seen to require less effort in time in comparison to those references.

Table 11 is organized to present the numerical results for ε=0.01, dt =0.002 and h=0.01 with various numbers of spatial and time nodes. The present solutions have been compared with the exact and other numerical solutions based on various numerical methods such as finite difference (Sari & Gurarslan, 2009) and the boundary element methods (Bahadir & Saglam, 2005).

Table 9. Comparison of the results produced for $\varepsilon = 1$, $h = 0.0125$, and $dt = 0.0002$.

x	t	**EX6** Splitting $h = 0.0125$, $dt = 0.0002$	**EX6** Splitting $h = 0.0125$, $dt = 0.0002$	Strang Splitting $h = 0.0125$, dt =0.0002	Kutluay *et al.* (1999) $h = 0.0125$, dt=0.0001	Shao *et al.* (2015) $h = 0.25$, dt=0.0001	Exact
x=0.25	t=0.05	0.426285622	0.426285617	0.426285616	0.42629	0.4262864	0.426285623
	t=0.10	0.261479814	0.261479812	0.261479811	0.26149	0.2614801	0.261479814
	t=0.15	0.161477615	0.161477610	0.161477605	0.16148	0.1614777	0.161477615
	t=0.25	0.061087582	0.061087577	0.061087571	0.06109	0.0610875	0.061087582
x=0.50	t=0.05	0.628083727	0.628083724	0.628083717	0.62809	0.6280846	0.628083727
	t=0.10	0.383422416	0.383422404	0.383422386	0.38343	0.3834228	0.383422416
	t=0.15	0.234055329	0.234055317	0.234055300	0.23406	0.2340554	0.234055329
	t=0.25	0.087232703	0.087232695	0.087232685	0.08724	0.0872327	0.087232703
x=0.75	t=0.05	0.465252624	0.465252602	0.465252556	0.46526	0.4652528	0.465252625
	t=0.10	0.281572640	0.281572622	0.281572589	0.28158	0.2815727	0.281572641
	t=0.15	0.169738279	0.169738265	0.169738245	0.16974	0.1697383	0.169738280
	t=0.25	0.062289848	0.062289842	0.062289834	0.06229	0.0622898	0.062289849

Table 10. Comparison of the produced results for $\varepsilon = 0.1$, $h = 0.0125$ and $dt = 0.001$.

x	t	**EX6** Splitting dt =0.001	**EX4** Splitting dt =0.001	Strang Splitting dt =0.001	Kutluay & Esen (2004) dt =0.0001	Kutluay *et al.* (2004) dt =0.0001	Exact
x=0.25	t=0.4	0.3175229	0.3175229	0.3175228	0.32091	0.31749	0.3175229
	t=0.6	0.2461385	0.2461385	0.2461384	0.24910	0.24612	0.2461385
	t=0.8	0.1995553	0.1995553	0.1995553	0.20211	0.19954	0.1995553
	t=1.0	0.1655986	0.1655986	0.1655986	0.16782	0.16559	0.1655986
	t=3.0	0.0277587	0.0277587	0.0277587	0.02828	0.02776	0.0277587
x=0.50	t=0.4	0.5845373	0.5845373	0.5845374	0.58788	0.58448	0.5845373
	t=0.6	0.4579764	0.4579764	0.4579765	0.46174	0.45793	0.4579764
	t=0.8	0.3673982	0.3673982	0.3673982	0.37111	0.36736	0.3673982
	t=1.0	0.2983431	0.2983431	0.2983431	0.30183	0.29831	0.2983431
	t=3.0	0.0410650	0.0410650	0.0410650	0.04185	0.04106	0.0410650
x=0.75	t=0.4	0.6456155	0.6456156	0.6456152	0.65054	0.64547	0.6456155
	t=0.6	0.5026758	0.5026758	0.5026755	0.50825	0.50255	0.5026758
	t=0.8	0.3853355	0.3853355	0.3853353	0.39068	0.38523	0.3853355
	t=1.0	0.2958567	0.2958567	0.2958566	0.30057	0.29578	0.2958567
	t=3.0	0.0304396	0.0304396	0.0304396	0.03106	0.03044	0.0304396

Table 11. Comparison of the produced results for $\varepsilon = 0.01, h = 0.01$ and $dt = 0.002$.

x	t	**EX6** Splitting $dt = 0.002$	**EX4** Splitting $dt = 0.002$	Strang Splitting $dt = 0.002$	Bahadir & Saglam (2005) $dt = 0.001$	Sari & Gurarslan (2009) $dt = 0.001$	Exact
$x = 0.10$	t=0.50	0.12846216	0.12846216	0.12846158	0.12808	0.12846	0.12846216
	t=2.00	0.04381385	0.04381385	0.04381376	0.04388	0.04379	0.04381385
	t=4.00	0.02334500	0.02334500	0.02334497	0.02351	0.02334	0.02334500
$x = 0.30$	t=0.50	0.37848913	0.37848913	0.37848813	0.37956	0.37849	0.37848913
	t=2.00	0.13134519	0.13134519	0.13134493	0.13129	0.13131	0.13134519
	t=4.00	0.07002718	0.07002718	0.07002710	0.07009	0.07002	0.07002718
$x = 0.50$	t=0.50	0.60988613	0.60988613	0.60988613	0.61768	0.60991	0.60988613
	t=2.00	0.21858801	0.21858801	0.21858762	0.21873	0.21858	0.21858801
	t=4.00	0.11668202	0.11668202	0.11668189	0.11671	0.11667	0.11668202
$x = 0.70$	t=0.50	0.80978166	0.80978166	0.80978409	0.83022	0.80986	0.80978166
	t=2.00	0.30534815	0.30534815	0.30534768	0.30614	0.30534	0.30534815
	t=4.00	0.16287830	0.16287830	0.16287813	0.16293	0.16287	0.16287830
$x = 0.90$	t=0.50	0.94601337	0.94601311	0.94602125	0.98068	0.94615	0.94601416
	t=2.00	0.38027320	0.38027324	0.38027364	0.38163	0.38027	0.38027365
	t=4.00	0.16857741	0.16857743	0.16857701	0.16766	0.16857	0.16857741

The present solutions revealed that less computational time is needed to achieve high accuracy, as compared to the previously mentioned effective methods.

The comparison of the present numerical results with the exact solution and the literature Jiwari (2015) with small viscosity values ε=0.004 and ε=0.003 is demonstrated in Tables 12 and 13, respectively. As underlined a couple of times, even a much smaller number of time elements suffice to achieve highly accurate solutions.

As seen in Figure 3(b), the numerical solution of Example 2 varies with gradually decreasing values of the viscosity constant ε, and the solutions tend to have a steep gredient. In Figure 4(a), the computed results in terms of the present methods are presented in Table 10 at time t=3. To assess the accuracy of the current methods, infinity error norm is used. Figure 4 includes the direct relation between the error norm versus the number of evaluations of ϕ_k^A . As seen in Figure 4(a), the extrapolation methods again produce

more acceptable numerical results for both accuracy and computational cost points of view, even if order reductions have occured.

Table 12. Comparison of the results produced with $\varepsilon = 0.004$ and $h = 0.01$ and $dt = 0.01$.

x	t	**EX6** Splitting $dt = 0.01$	**EX4** Splitting $dt = 0.01$	Strang Splitting $dt = 0.01$	Jiwari (2015) $dt = 0.001$	Exact
$x = 0.25$	$t = 1$	0.19639300	0.19639300	0.19637590	0.19636	0.19639300
	$t = 5$	0.04743858	0.04743858	0.04743723	0.04744	0.04743858
	$t = 10$	0.02434263	0.02434263	0.02434227	0.02434	0.02434263
	$t = 15$	0.01637125	0.01637125	0.01637108	0.01637	0.01637125
$x = 0.5$	$t = 1$	0.38849076	0.38849076	0.38846870	0.38842	0.38849076
	$t = 5$	0.09486089	0.09486089	0.09485824	0.09491	0.09486089
	$t = 10$	0.04868313	0.04868313	0.04868240	0.04868	0.04868313
	$t = 15$	0.03270700	0.03270700	0.03270667	0.03270	0.03270700
$x = 0.75$	$t = 1$	0.57319765	0.57319726	0.57318887	0.57312	0.57322509
	$t = 5$	0.14224850	0.14224849	0.14224467	0.14224	0.14224850
	$t = 10$	0.07258104	0.07258104	0.07257997	0.07258	0.07258104
	$t = 15$	0.04696437	0.04696437	0.04696383	0.04696	0.04696437

Table 13. Comparison of the produced results for $\varepsilon = 0.003$, $h = 0.01$ and $\mathrm{dt} = 0.005$.

x	t	**EX6** Splitting $dt = 0.005$	**EX4** Splitting $dt = 0.005$	Strang Splitting $dt = 0.005$	Jiwari (2015) $dt = 0.001$	Exact
$x = 0.25$	$t = 1$	0.19672202	0.19672202	0.19671762	0.19668	0.19672202
	$t = 5$	0.04746474	0.04746474	0.04746439	0.04746	0.04746474
	$t = 10$	0.02434970	0.02434970	0.02434960	0.02434	0.02434970
	$t = 15$	0.01637507	0.01637507	0.01637502	0.01637	0.01637507
$x = 0.5$	$t = 1$	0.38896706	0.38896706	0.38896141	0.38890	0.38896706
	$t = 5$	0.09491170	0.09491170	0.09491101	0.09491	0.09491170
	$t = 10$	0.04869814	0.04869814	0.04869795	0.04870	0.04869814
	$t = 15$	0.03274752	0.03274752	0.03274743	0.03274	0.03274752
$x = 0.75$	$t = 1$	0.57382849	0.57382848	0.57382592	0.57375	0.57382849
	$t = 5$	0.14232395	0.14232395	0.14232295	0.14232	0.14232395
	$t = 10$	0.07298597	0.07298597	0.07298569	0.07298	0.07298597
	$t = 15$	0.04856835	0.04856835	0.04856822	0.04857	0.04696437

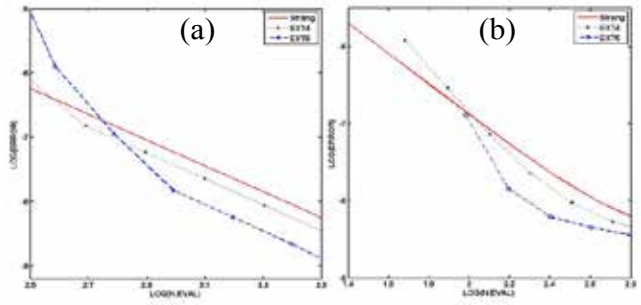

Fig. 4. a) Error versus number of evaluations of ϕ_k^A for the numerical solution of Example 2 at a) t=3,ε=0.1 and h=0.0125 and Example 3 at
b) t=2,ε=0.02 and h=0.025

Example 3 (Iskandar & Mohsen, 1992) Let us consider the Burgers equation (1) with the initial condition,

$$u(x,0) = \frac{1}{Re}\left[x + \tan\left(\frac{x}{2}\right)\right] \tag{49}$$

and the nonhomogeneous and time dependent boundary conditions

$$u(0.5,t) = \frac{1}{Re+t}\left[0.5 + \tan\left(\frac{Re}{4(R+t)}\right)\right] \tag{50}$$

$$u(1.5,t) = \frac{1}{Re+t}\left[1.5 + \tan\left(\frac{3Re}{4(R+t)}\right)\right], \tag{51}$$

where ε=1/Re. The exact solution of the Burgers equation is

$$u(x,t) = \frac{1}{Re+t}\left[x + \tan\left(\frac{xRe}{2(R+t)}\right)\right]. \tag{52}$$

In Figure 4(b), the capability of the current algorithms are shown for the Burgers equation with time-dependent nonhomogeneus boundary conditions. In the figure, the error norm versus the number of evaluations of ϕ_k^A is presented for the parameters h=0.025,ε=0.02 and t=2. It has been observed that order reductions occurred for extrapolation methods as well as for Strang splitting methods (see Figure 4(b)). For further information on order reduction phenemona in diffusion-reaction equations employing a splitting method with nonhomogenous and time-dependent Dirichlet boundary condition, readers are referred to literature by Einkemmer and Ostermann (2015). As seen in Figure 4(b), the extrapolation methods again produce more accurate and more economical numerical results.

6. Conclusions and Recommendation

This article has proposed a higher order splitting-up method based on cubic B-spline Galerkin finite element method in numerically analyzing the advection-diffusion processes. The splitting method is generated by following three approaches: the second-order Strang approach, the fourth-order, and the sixth-order extrapolation approaches. The stability analysis of the suggested method has been studied and shown to be unconditionally stable for both parts of the physical processes. To illustrate the accuracy of the present method, three challenging problems have been considered. Qualitative and quantitative analysis reveal that the current method is capable of producing highly

accurate results even with a smaller number of temporal and spatial elements. The computed solutions agree with the literature and the exact solution. Notice that the present method can capture the steep behavior of the Burgers equation when the advection is dominant. The current numerical study has been carried out for the deterministic advection-diffusion processes. Any further research should involve stochastic advection-diffusion processes.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous referees of the Kuwait Journal of Science for their valuable comments and suggestions to improve the paper.

References

Aksan, E.N. (2006). Quadratic B-spline finite element method for numerical solution of the Burgers equation. Applied Mathematics and Computation, 174: 884-896.

Bahadir, A.R. & Saglam, M. (2005). A mixed finite difference and boundary element approach to one-dimensional Burgers' equation. Applied Mathematics and Computation, 160: 663-673 .

Bashan, A., Karakoc, S.B.G. & Geyikli, T. (2015). Approximation of the KdVB equation by the quintic B-spline differential quadrature method. Kuwait Journal of Science, 42: 67-92.

Blanes, S. & Casas, F. (2005). On the necessity of negative coefficients for operator splitting schemes of order higher than two. Applied Numerical Mathematics, 54: 23-37.

Burgers, J. M. (1948). A mathematical model illustrating the theory of turbulence,. turbulence, Advances in Applied Mechanics, 1: 171-199.

Castella, F., Chartier, P., Descombes, S. & Vilmart, G. (2009). Splitting methods with complex times for parabolic equations. BIT Numerical Mathematics, 49: 487-508.

Cole, J. D. (1951). On a quasi-linear parabolic equation in aerodynamics. Quarterly of Applied Mathematics, 9: 225-236.

Creutz, M. & Gocksch, A. (1989). Higher-order hybrids Monte Carlo algorithms. Physics Letters A, 63: 9-12.

Dag, I., Saka, B. & Boz, A. (2005). B-spline Galerkin methods for numerical solutions of Burgers' equation. Applied Mathematics and Computation, 166: 506-522.

Einkemmer, L. & Ostermann, A. (2015). Overcoming order reduction in diffusion-reaction splitting. Part 1: Dirichlet boundary conditions. SIAM Journal of Scientific Computing, 37: 1-15.

Hairer, E., Lubich, C. & Wanner, G. (2006). Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations. Second Edition. Springer Series in Computational Mathematics 31, Springer: Berlin.

Hansen, E. & Ostermann, A. (2009). High order splitting methods for analytic semigroups exist. BIT Numerical Mathematics, 49: 527-542.

Hansen, E. & Ostermann, A. (2009). Exponential splitting for unbounded operators. Mathematics of Computation, 78: 1485-1496.

Hopf, E. (1950). The partial differential equation $u_t+uu_x=\varepsilon u_{xx}$. Communications on Pure and Applied Mathematics, 9: 201-230.

Iskandar, L. & Mohsen, A. (1992). Some numerical experiments on the splitting of burgers' equation. Numerical Methods for Partial Differential Equations, 8: 267-276.

Jain, P.C. & Raja, M. (1979). Splitting-up technique for Burgers equation. Indian Journal of Pure and Applied Mathematics, 10: 1543-1551.

Jain, P.C. & Holla, D.N. (1978). Numerical solution of coupled Burgers' equations. International Journal of Non-linear Mechanics, 13: 213-222.

Jain, P.C., Shankar, R. & Singh, V. (1992). Cubic spline technique for solution of Burgers' equation with a semi-linear boundary conditions, International Journal for Numerical Methods in Biomedical Engineering, 8: 235-242.

Jiwari, R. (2015). A hybrid numerical scheme for the numerical solution of the Burgers' equation, Computer Physics Communications, 188: 50-67.

Karakoc, S.B.G., Ucar, Y. & Yagmurlu, N. (2015). Numerical solutions of the MRLW equation by cubic B-spline Galerkin finite element method. Kuwait Journal of Science, 42(2): 141-159.

Kutluay, S. & Esen, A. (2004). A lumped Galerkin method for solving the Burgers equation. International Journal of Computer Mathematics, 81: 1433-1444.

Kutluay, S., Esen, A. & Dag, I. (2004). Numerical solutions of the Burgers' equation by the least-squares

quadratic B-spline finite element method. Journal of Computational and Applied Mathematics, 167: 21-33.

Kutluay, S., Bahadir, A.R. & Ozdes, A. (1999). Numerical solution of one-dimensional Burgers' equation: Explicit and exact-explicit finite difference methods. Journal of Computational and Applied Mathematics, 103: 251–261.

Liao, W. & Zhu, J. (2011). Efficient and accurate finite difference schemes for solving one-dimensional Burgers' equation. International Journal of Computer Mathematics, 88: 2575-2590.

Miller, E.L. (1966). Predictor–corrector studies of Burger's model of turbulent flow. M.S. thesis. University of Delaware, Newark-Delaware, USA. Mukundan, V. & Awasthi, A. (2015). Efficient numerical techniques for Burgers' equation. Applied Mathematics and Computation, 262: 282-297.

Prenter, P.M. (1975). Splines and Variational Methods. John Wiley & Sons: New York.

Pospelov, L.A. (1966). Propagation of finite amplitude elastic waves. Soviet Physics Acoustics, 11: 302-304.

Raslan, K.R. (2003). A collocation solution for Burgers equation using quadratic B-spline finite elements. International Journal of Computer Mathematics, 80: 931-938.

Saka, B. & Dag, I. (2008). A numerical study of the Burgers' equation. Journal of the Franklin Institute, 345: 328-348.

Sari, M. & Gurarslan, G. (2009). A sixth-order compact finite difference scheme to the numerical solutions of Burgers' equation. Applied Mathematics and Computation, 208: 475-483.

Sari, M. & Tunc, H. (2017). An optimization technique in analysing the Burgers equation. Sigma Journal of Engineering and Natural Sciences, 35: 369-386.

Seydaoglu, M. & Blanes, S. (2014). High order splitting methods for separable non-autonomous parabolic equations. Applied Numerical Mathematics, 84: 22-32.

Seydaoglu, M., Erdogan, U. & Ozis, T. (2016). Numerical solution of Burgers' equation with higher order splitting methods. Journal of Computational and Applied Mathematics, 291: 410-421.

Shao, L., Feng, X. & He, Y. (2011). The local discontinuous Galerkin finite element method for Burgers equation. Mathematical and Computer Modelling, 54: 2943-2954.

Soliman, A.A. (2012). A Galerkin solution for Burgers equation using cubic B-spline finite elements. Abstract and Applied Analysis, 2012: 1-15.

Suziki, M. (1990). Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, Physics Letters A, 146: 319-323.

Talwar, J., Mohanty, R.K. & Singh, S. (2016). A new algorithm based on spline in tension approximation for 1D parabolic quasi-linear equations on a variable mesh. International Journal of Computer Mathematics, 93: 1771-1786.

Tsai, C., Shih, Y., Lin, Y. & Wang, H. (2017). Tailored finite point method for solving one-dimensional Burgers' equation. International Journal of Computational and Applied Mathematics, 94: 8000-8012.

Tunc, H. (2017). Various finite element techniques for advection-diffusion-reaction processes. MSc thesis. Yildiz Technical University, Istanbul, Turkey.

Van der Pol, B. (1951). On a non-linear partial differential equation satisfied by the logarithm of the Jacobian theta-functions with arithmetical applications. Proceedings of the National Academy of Sciences of the Amsterdam, 13: 261-271.

Verma, A.K. & Verma, L. (2015). Higher order time integration formula with application on Burgers' equation. International Journal of Computer Mathematics, 92:756-771.

Wang, J. & Warnecke, G. (2003). Existence and uniqueness of solutions for a non-uniformly parabolic equation. Journal of Differential Equations, 189: 1–16.

Yoshida, H. (1990). Construction of higher order symplectic integrators. Physics Letters A, 150:262-268.

Submitted: 13-07-2017
Revised: 18-10-2017
Accepted: 02-01-2018

طرق التقسيم ذات الترتيب الأعلى عند التحليل باستخدام معادلة بيرجرز

[1,*] مراد سري ، [1] حسين تونك ، [2] معاذ سيدأوغلو

[1] قسم الرياضيات، كلية الآداب والعلوم، جامعة يلدز التقنية، اسطنبول 34220، تركيا
[2] قسم الرياضيات، كلية الآداب والعلوم، جامعة موس ارباسلان، موس 49100، تركيا

[*]sarim@yildiz.edu.tr

الملخص

تقترح هذه المقالة بعض تقنيات التقسيم ذات الترتيب الأعلى على أساس طريقة جالركن بالشريحة التكعيبية B للعنصر المنتهي عند التحليل باستخدام نموذج معادلة بيرجرز. تمت دراسة الصيغة القوية لكل من أجزاء البقاء والانتشار من معادلة بيرجرز لتقسيم الوقت عند إنشاء طريقة جالركن. ولدمج نظام ODE المماثل، تم استخدام مخططات Crank-Nic-olson لتقسيم الوقت. وكانت المخططات المقترحة مستقرة بدون شروط أو قيود. وتمت دراسة ثلاثة أمثلة صعبة مع تغيير قيم ثابت اللزوجة الحركية للوسط. وتم حل حالات موجات التصادم لمعامل الانحدار ومقارنتها مع الحل الصحيح والأبحاث المنشورة. وأظهرت النتائج النوعية والكمية أن طريقتنا العددية لديها دقة أعلى بكثير من الطرق المُنافسة.