

# A NUMERICAL IMPLEMENTATION OF FOURTH ORDER TIME INTEGRATION FORMULA VIA MOL ON THE UNSTEADY BURGERS' EQUATION

B. Mayur Prakash<sup>1</sup>, Ashish Awasthi<sup>2</sup>, S. Jayraj<sup>3</sup>

<sup>1</sup>Department of Mechanical Engineering, National Institute of Technology Calicut, bonkile.mayur@gmail.com

<sup>2</sup>Department of Mathematics, National Institute of Technology Calicut, aawasthi@nitc.ac.in

<sup>3</sup>Department of Mechanical Engineering, National Institute of Technology Calicut, sjayaraj@nitc.ac.in

## ABSTRACT

In this paper, method-of-lines (MOL) in conjunction with optimal five-stage, order four strong stability-preserving time-stepping Runge-Kutta (SSP-RK54) scheme is proposed to find the approximate solution of the unsteady Burgers' equation. The MOL is used in space, and the SSP-RK54 scheme is used in time for solving the resulting system of ordinary differential equations. To check the efficiency and accuracy of the method, two test examples of Burgers' equation are included with their numerical solutions,  $L_2$  and  $L_\infty$  errors and comparison is done with the exact solution for modest values of Reynold numbers ( $Re$ ). The numerical solutions obtained typically does not exhibit nonlinear instabilities. It is also observed that, the proposed method yields satisfactory results for all the cases considered. The proposed method is found to be robust for low and high  $Re$  therefore the proposed method also be useful for general models involving the solution of advection-diffusion equations.

## 1 Introduction

We consider a very famous one-dimensional non-linear unsteady Burgers' equation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \frac{1}{Re} \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1, \quad 0 < t < T \quad (1)$$

$$u(x, 0) = f(x), \quad 0 < x < 1 \quad (2)$$

$$u(0, t) = g_1(t), \quad u(1, t) = g_2(t), \quad 0 < t < T \quad (3)$$

where  $u$ ,  $x$ ,  $t$  and  $Re$  are the velocity, spatial coordinate, time and Reynold number respectively. In 1915, Harry Bateman (1882-1946) [1], an English mathematician, introduced this equation in his paper along with its initial and boundary conditions. Later in 1948, Johannes Martinus Burgers (1895-1981) [2], a Dutch physicist, explained the mathematical modeling of turbulence with the help of Equation (1). In order to honor the contributions of Burgers, this equation is well known as the "Burgers' equation".

In last decades, the application of this model in various important fields has encouraged researcher to solve the Burgers' equation more efficiently. In 1930, Rothe [7] who

was from former Soviet Union, has introduced method-of-lines (MOL) in his paper. The relative merits and demerits of MOL with an ordinary differential equation solver to classical explicit and implicit finite difference techniques were compared by Kurtz [5]. Shampine [8] has investigated factors influencing the choice of ordinary differential equation solver for the numerical solution of advection-diffusion partial differential equation by MOL. Oymak [6] has proposed a time-accurate Navier-Stokes code based on the MOL approach with a non-iterative algorithm for the pressure. The special class of high-order time discretization methods were used to design the high-order methods that are nonlinearly stable. These methods are well known as strong-stable-preserving (SSP) methods [9] [10]. SSP time discretization methods are popular and effective algorithms for the simulation of partial differential equations having discontinuous or shock-like solutions. The main motivation for this work is the need of robust scheme for the solution of Burgers' equation with high Re which is a common fluid flow situation. In this paper, Equation (1) is semi-discretized with respect to spatial variable by MOL. This result in a system of ordinary differential equations in the variable "t". The resulting system of ordinary differential equations is solved by SSP-RK54 scheme [10]. SSP-RK54 scheme is selected as it required less storage space which result in less accumulation of numerical errors. The accuracy and reliability of the proposed method is verified by performing several numerical experiments.

## 2 Difference Scheme

MOL which is a popular semi-discretization method for the solution of time-dependent partial differential equations is used. The obtained set of ordinary differential equation is integrated using SSP-RK54 scheme. The solution domain of Equation (1) is discretized with uniform mesh. The space interval  $[0, 1]$  is divided into  $N$  equal subinterval. Assuming  $\Delta x = 1/N$  as the mesh width in space and  $x_i$  is set as  $x_i = i\Delta x$  for  $i = 0, 1, \dots, N$ .

### 2.1 Semi-discretization: Method of Lines (MOL)

Equation (1) is first discretized in spatial variable to obtain a semi-discrete MOL scheme. Convective term ( $u \frac{\partial u}{\partial x}$ ) is discretized by using second order upwind method that maintain unidirectional flow of information. Whereas central difference is used for discretization of diffusive term ( $\frac{1}{Re} \frac{\partial^2 u}{\partial x^2}$ ). Equation (1) is written as

$$\frac{\partial u_i}{\partial t} = \frac{1}{Re} \left( \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} \right) - u_i \left( \frac{3u_i - 4u_{i-1} + u_{i-2}}{2\Delta x} \right) \quad (4)$$

Equation (4) is written as system of ordinary differential equations in time, that is, for  $i = 2, \dots, N$ , we have

$$\frac{d}{dt}(u_i) = L(u_i) \quad (5)$$

where  $L$ , denotes a spatial non-linear differential operator

$$L(u_i) = \frac{1}{Re} \left( \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} \right) - u_i \left( \frac{3u_i - 4u_{i-1} + u_{i-2}}{2\Delta x} \right) \quad (6)$$

Table 1: The coefficients of the optimal SSPRK (5,4) scheme [10].

|               |                  |                  |                  |                  |                  |  |
|---------------|------------------|------------------|------------------|------------------|------------------|--|
|               | 1                |                  |                  |                  |                  |  |
|               | 0.44437049406734 | 0.55562950593266 |                  |                  |                  |  |
| $\alpha_{jk}$ | 0.62010185138540 | 0                | 0.37989814861460 |                  |                  |  |
|               | 0.17807995410773 | 0                | 0                | 0.82192004589227 |                  |  |
|               | 0.00683325884039 | 0                | 0.51723167208978 | 0.12759831133288 | 0.34833675773694 |  |
|               | 0.39175222700392 |                  |                  |                  |                  |  |
|               | 0                | 0.36841059262959 |                  |                  |                  |  |
| $\beta_{jk}$  | 0                | 0                | 0.25189177424738 |                  |                  |  |
|               | 0                | 0                | 0                | 0.54497475021237 |                  |  |
|               | 0                | 0                | 0                | 0.08460416338212 | 0.22600748319395 |  |

## 2.2 Time Integration

In this way, we find a system of ordinary differential equation (5) which is solved by five-stage, order four strong stability-preserving time-stepping Runge-Kutta (SSP-RK54) scheme [10]. The class of SSP method used is defined by the property that the number of stages  $s$  ( $s = 5$ ) is greater than the order  $k$  ( $k = 4$ ) of the method.

### 2.2.1 SSP-RK54

The objective of SSP-RK54 is to maintain the strong stability property while achieving higher order accuracy in time. Consider an  $s$ -stage, explicit RungeKutta method written in the form

$$U^{(0)} = U^n, \quad (7)$$

$$U^{(j)} = \sum_{k=0}^{j-1} [\alpha_{jk} U^{(k)} + \Delta t \beta_{jk} L(U^{(k)})], \quad j = 1, 2, \dots, s \quad (8)$$

$$U^{n+1} = U^s, \quad (9)$$

where all  $\alpha_{jk} \geq 0$  and  $\alpha_{jk} = 0$  only if  $\beta_{jk} = 0$  [9]. SSP-RK54 scheme has a Courant-Friedrichs-Levy (CFL) coefficient of 1.50818004975927. For consistency, we must have  $\sum_{k=0}^{j-1} \alpha_{jk} = 1$ ,  $j = 1, 2, \dots, s$ . Table (1) gives the value of  $\alpha_{jk}$  and  $\beta_{jk}$  coefficients. Equation (5) is integrated from time  $t_n$  to  $t_n + \Delta t$  through the following operations and consequently the solution  $u(x, t)$  at a particular time level is completely known. The fully discrete scheme for Equation (1) is as follow

$$u_i^{(0)} = u_i^n \quad (10)$$

where  $u_i^n$  is a initial condition

$$u_i^{(1)} = u_i^{(0)} + \Delta t (0.39175222700392) \left[ \frac{1}{Re} \left( \frac{u_{i-1}^{(0)} - 2u_i^{(0)} + u_{i+1}^{(0)}}{\Delta x^2} \right) - u_i^{(0)} \left( \frac{3u_i^{(0)} - 4u_{i-1}^{(0)} + u_{i-2}^{(0)}}{2\Delta x} \right) \right] \quad (11)$$

$$u_i^{(2)} = (0.44437049406734)u_i^{(0)} + (0.55562950593266)u_i^{(1)} + \Delta t (0.36841059262959)$$

$$\left[ \frac{1}{Re} \left( \frac{u_{i-1}^{(1)} - 2u_i^{(1)} + u_{i+1}^{(1)}}{\Delta x^2} \right) - u_i^{(1)} \left( \frac{3u_i^{(1)} - 4u_{i-1}^{(1)} + u_{i-2}^{(1)}}{2\Delta x} \right) \right] \quad (12)$$

$$u_i^{(3)} = (0.62010185138540)u_i^{(0)} + (0.37989814861460)u_i^{(2)} + \Delta t(0.25189177424738) \left[ \frac{1}{Re} \left( \frac{u_{i-1}^{(2)} - 2u_i^{(2)} + u_{i+1}^{(2)}}{\Delta x^2} \right) - u_i^{(2)} \left( \frac{3u_i^{(2)} - 4u_{i-1}^{(2)} + u_{i-2}^{(2)}}{2\Delta x} \right) \right] \quad (13)$$

$$u_i^{(4)} = (0.17807995410773)u_i^{(0)} + (0.82192004589227)u_i^{(3)} + \Delta t(0.54497475021237) \left[ \frac{1}{Re} \left( \frac{u_{i-1}^{(3)} - 2u_i^{(3)} + u_{i+1}^{(3)}}{\Delta x^2} \right) - u_i^{(3)} \left( \frac{3u_i^{(3)} - 4u_{i-1}^{(3)} + u_{i-2}^{(3)}}{2\Delta x} \right) \right] \quad (14)$$

Final solution is given by

$$u_i^{(n+1)} = (0.00683325884039)u_i^{(0)} + (0.51723167208978)u_i^{(2)} + (0.12759831133288)u_i^{(3)} + \Delta t(0.08460416338212) \left[ \frac{1}{Re} \left( \frac{u_{i-1}^{(3)} - 2u_i^{(3)} + u_{i+1}^{(3)}}{\Delta x^2} \right) - u_i^{(3)} \left( \frac{3u_i^{(3)} - 4u_{i-1}^{(3)} + u_{i-2}^{(3)}}{2\Delta x} \right) \right] + (0.34833675773694)u_i^{(4)} + \Delta t(0.22600748319395) \left[ \frac{1}{Re} \left( \frac{u_{i-1}^{(4)} - 2u_i^{(4)} + u_{i+1}^{(4)}}{\Delta x^2} \right) - u_i^{(4)} \left( \frac{3u_i^{(4)} - 4u_{i-1}^{(4)} + u_{i-2}^{(4)}}{2\Delta x} \right) \right] \quad (15)$$

where  $i = 2, \dots, N$ . For next iteration  $u_i^n = u_i^{n+1}$ .

### 3 Numerical Experiments

In this section, the numerical solutions by the proposed method are evaluated for two test examples of Burgers' equation. Exact solution was elaborated by Kadalbajoo and Awasthi [4] for both examples 1 and 2. The accuracy of numerical method is measured with the help of exact solution. Also,  $L_2$  and  $L_\infty$  errors are evaluated to measure the accuracy and the efficiency of the proposed method.  $L_2$  and  $L_\infty$  errors are defined as

$$L_2 = \left( \sum_{i=1}^N [u_i^{exact} - u_i^+]^2 \right)^{\frac{1}{2}}, \quad L_\infty = \max_{1 \leq i \leq N} |u_i^{exact} - u_i^+| \quad (16)$$

where  $u_i^+$  represent the numerical solution at node  $i$ .

**Example 1.** Burgers' equation (1) with initial condition and homogeneous boundary conditions

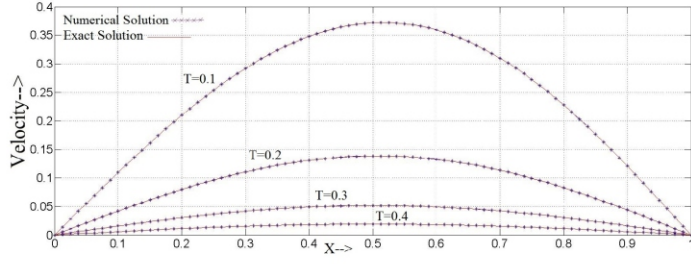
$$u(x, 0) = \sin(\pi x), \quad 0 < x < 1, \quad (17)$$

$$u(0, t) = u(1, t) = 0, \quad 0 \leq t \leq \tau. \quad (18)$$

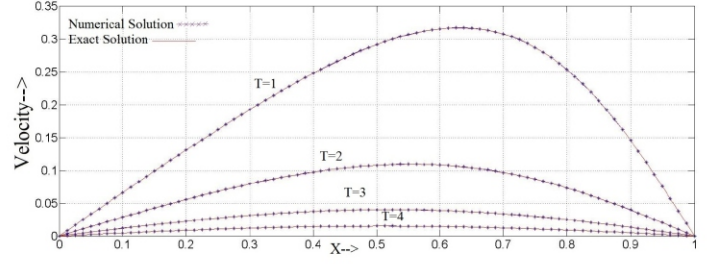
**Example 2.** Burgers' equation (1) with the following initial condition and boundary conditions

$$u(x, 0) = 4x(1 - x), \quad 0 < x < 1, \quad (19)$$

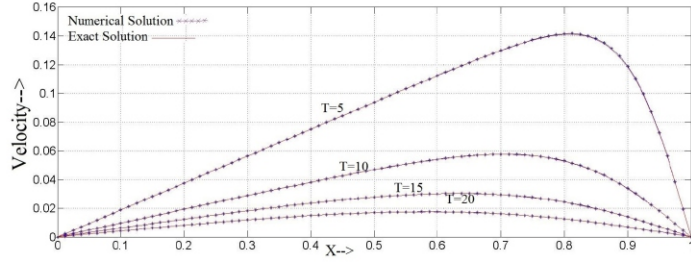
$$u(0, t) = 0 = u(1, t), \quad 0 \leq t \leq \tau. \quad (20)$$



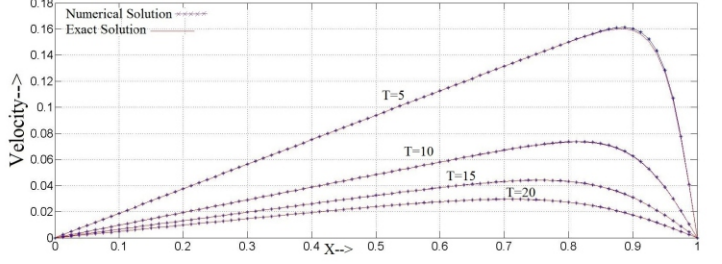
(a)



(b)

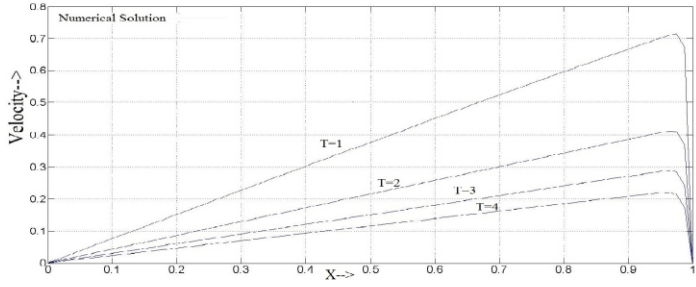


(c)

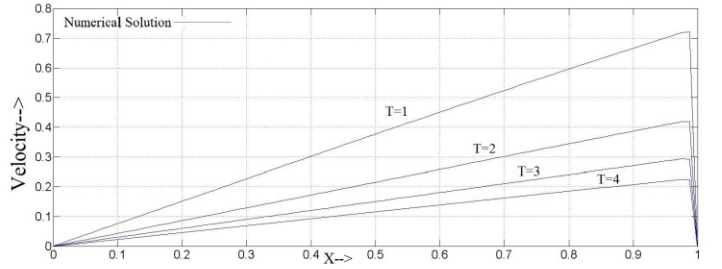


(d)

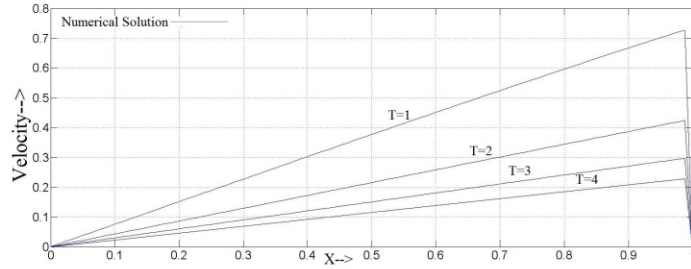
Figure 1: Numerical solutions of example 1 at several times for  $N=80$  with different values of  $Re$  and  $\Delta t$ , (a)  $Re = 1, \Delta t = 0.0001$ ; (b)  $Re = 10, \Delta t = 0.0001$ ; (c)  $Re = 100, \Delta t = 0.01$ ; (d)  $Re = 200, \Delta t = 0.01$ .



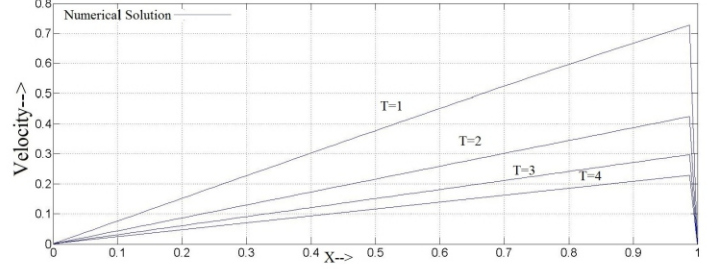
(a)



(b)

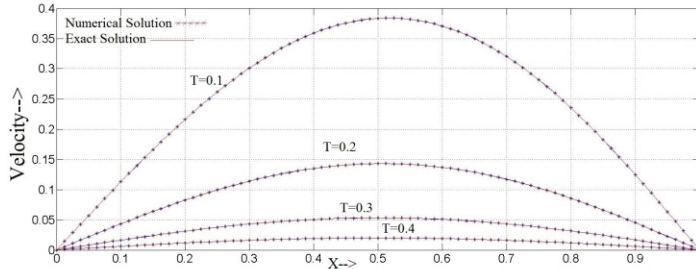


(c)

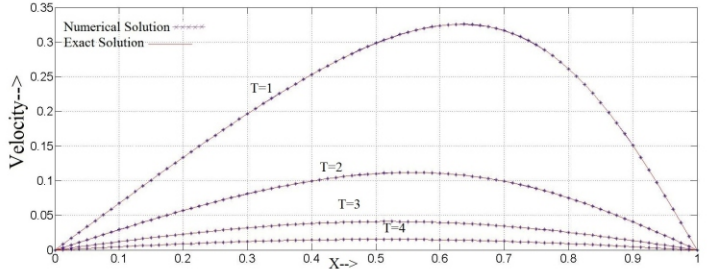


(d)

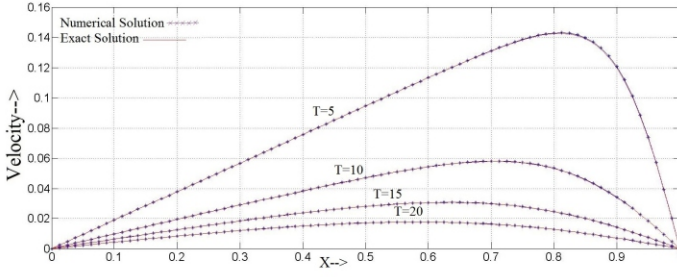
Figure 2: Numerical solution of example 1 at several times for  $N=80$  with  $\Delta t = 0.0001$  and different values of  $Re$ , (a)  $Re = 10^3$ ; (b)  $Re = 10^4$ ; (c)  $Re = 10^5$ ; (d)  $Re = 10^6$ .



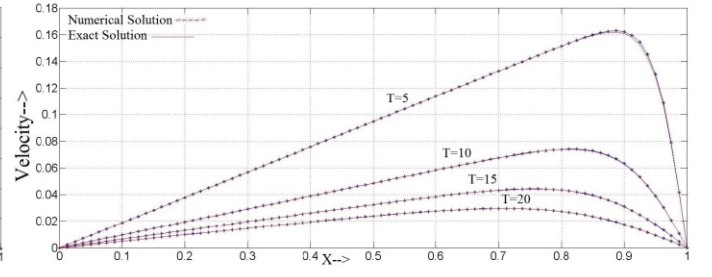
(a)



(b)

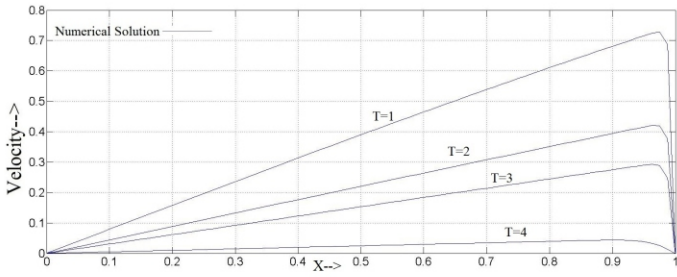


(c)

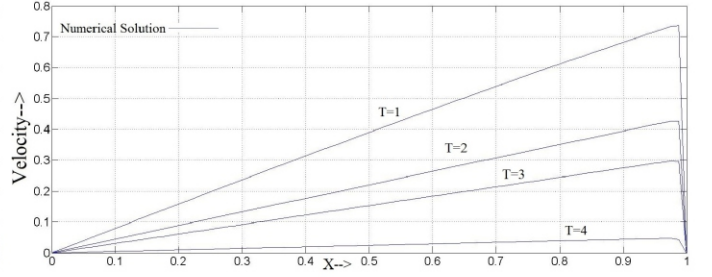


(d)

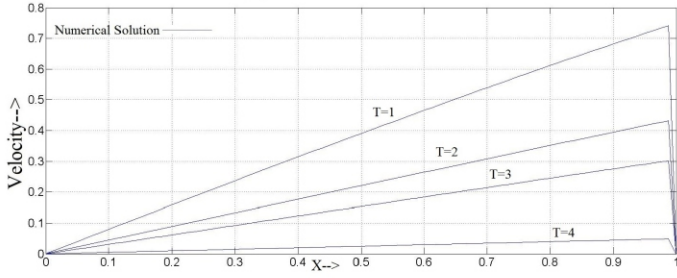
Figure 3: Numerical solution of example 2 at several times for  $N=80$  with different values of  $Re$  and  $\Delta t$ , (a)  $Re = 1, \Delta t = 0.0001$ ; (b)  $Re = 10, \Delta t = 0.0001$ ; (c)  $Re = 100, \Delta t = 0.01$ ; (d)  $Re = 200, \Delta t = 0.01$ .



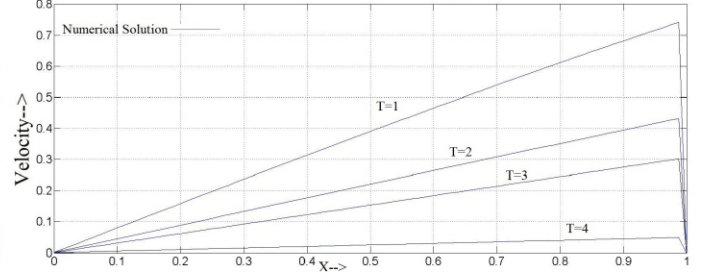
(a)



(b)



(c)



(d)

Figure 4: Numerical solution of example 2 at several times for  $N=80$  with different values of  $Re$  and  $\Delta t$ , (a)  $Re = 10^3, \Delta t = 0.0001$ ; (b)  $Re = 10^4, \Delta t = 0.0001$ ; (c)  $Re = 10^5, \Delta t = 0.0001$ ; (d)  $Re = 10^6, \Delta t = 0.0001$ .

Figure 1 and 3 show the graphs for computed and exact solutions at different time for various values of  $Re$ . Reynold number  $Re = 1, 10, 100$  and  $200$  are considered with  $\Delta x = 0.0125$  to compare computed solutions with exact solutions. From these graphs it is observed that the proposed method gives accurate results for any value of time step  $\Delta t$ . It is found that the computed results show better agreement with the exact solution.  $L_2$  and  $L_\infty$  errors are presented for example 1 (Table 2) and example 2 (Table 3) with  $\Delta x = 0.0125$  for different  $Re$  and time level. Numerical difficulties arise in the solution for the case of large  $Re$  because exact solution fails when  $Re$  is high [11]. Under most practical conditions, the flow in a circular pipe is turbulent for  $Re \geq 4000$  [3]. The proposed method gives accurate results for turbulent flow also. The numerical solutions of example 1 (Figure 2) and example 2 (Figure 4) are plotted for Reynold numbers  $Re = 10^3, 10^4, 10^5$  and  $10^6$  at different time levels. A small time increment ( $\Delta t = 0.0001$ ) is chosen to ensure high accuracy. Significantly contrasting behavior is

Table 2:  $L_2$  and  $L_\infty$  errors of Example 1 for  $\Delta t = 0.0001$  at different  $Re$  and time (T)

| $Re = 10$ |                         |                         | $Re = 1$ |                         |                         |
|-----------|-------------------------|-------------------------|----------|-------------------------|-------------------------|
| T         | $L_2$                   | $L_\infty$              | T        | $L_2$                   | $L_\infty$              |
| 1         | $1.2964 \times 10^{-3}$ | $2.3846 \times 10^{-4}$ | 0.1      | $2.8874 \times 10^{-4}$ | $4.9342 \times 10^{-5}$ |
| 2         | $4.5825 \times 10^{-4}$ | $7.4246 \times 10^{-5}$ | 0.2      | $2.1400 \times 10^{-4}$ | $3.4100 \times 10^{-5}$ |
| 3         | $1.4344 \times 10^{-4}$ | $2.2806 \times 10^{-5}$ | 0.3      | $4.5458 \times 10^{-4}$ | $7.1879 \times 10^{-5}$ |
| 4         | $4.1592 \times 10^{-5}$ | $6.5860 \times 10^{-6}$ | 0.4      | $6.1320 \times 10^{-5}$ | $9.6950 \times 10^{-6}$ |

---

| $Re = 100$ |                         |                         | $Re = 200$ |                         |                         |
|------------|-------------------------|-------------------------|------------|-------------------------|-------------------------|
| T          | $L_2$                   | $L_\infty$              | T          | $L_2$                   | $L_\infty$              |
| 5          | $2.2278 \times 10^{-3}$ | $6.8530 \times 10^{-4}$ | 7          | $2.5599 \times 10^{-3}$ | $9.4898 \times 10^{-4}$ |
| 10         | $5.3430 \times 10^{-4}$ | $1.1698 \times 10^{-4}$ | 10         | $1.1851 \times 10^{-3}$ | $3.7044 \times 10^{-4}$ |
| 15         | $2.4224 \times 10^{-4}$ | $4.3855 \times 10^{-5}$ | 15         | $4.9769 \times 10^{-4}$ | $1.2703 \times 10^{-4}$ |
| 20         | $1.3402 \times 10^{-4}$ | $2.2167 \times 10^{-5}$ | 20         | $2.7435 \times 10^{-4}$ | $6.0575 \times 10^{-5}$ |

Table 3:  $L_2$  and  $L_\infty$  errors of Example 2 for  $\Delta t = 0.0001$  at different  $Re$  and time (T)

| $Re = 10$ |                         |                         | $Re = 1$ |                         |                         |
|-----------|-------------------------|-------------------------|----------|-------------------------|-------------------------|
| T         | $L_2$                   | $L_\infty$              | T        | $L_2$                   | $L_\infty$              |
| 1         | $1.6206 \times 10^{-5}$ | $2.3846 \times 10^{-4}$ | 0.1      | $3.6093 \times 10^{-6}$ | $4.9342 \times 10^{-5}$ |
| 2         | $5.7282 \times 10^{-6}$ | $7.4246 \times 10^{-5}$ | 0.2      | $2.6750 \times 10^{-6}$ | $3.4100 \times 10^{-5}$ |
| 3         | $1.7931 \times 10^{-6}$ | $2.2806 \times 10^{-5}$ | 0.3      | $5.6823 \times 10^{-6}$ | $7.1879 \times 10^{-5}$ |
| 4         | $5.1991 \times 10^{-7}$ | $6.5860 \times 10^{-6}$ | 0.4      | $7.6651 \times 10^{-7}$ | $9.6950 \times 10^{-6}$ |

---

| $Re = 100$ |                         |                         | $Re = 200$ |                         |                         |
|------------|-------------------------|-------------------------|------------|-------------------------|-------------------------|
| T          | $L_2$                   | $L_\infty$              | T          | $L_2$                   | $L_\infty$              |
| 5          | $2.7848 \times 10^{-5}$ | $6.8530 \times 10^{-4}$ | 7          | $3.1999 \times 10^{-5}$ | $9.4898 \times 10^{-4}$ |
| 10         | $6.6788 \times 10^{-6}$ | $1.1698 \times 10^{-4}$ | 10         | $1.4814 \times 10^{-5}$ | $3.7044 \times 10^{-4}$ |
| 15         | $3.0281 \times 10^{-6}$ | $4.3855 \times 10^{-5}$ | 15         | $6.2212 \times 10^{-6}$ | $1.2703 \times 10^{-4}$ |
| 20         | $1.6753 \times 10^{-6}$ | $2.2167 \times 10^{-5}$ | 20         | $3.4294 \times 10^{-6}$ | $6.0575 \times 10^{-5}$ |

observed for high Reynold number. When  $Re$  is high, the diffusive term ( $\frac{1}{Re} \frac{\partial^2 u}{\partial x^2}$ ) tends to zero and it's effect is negligible on numerical solution. In such situation Equation (1) which is basically viscous Burgers' equation behaves like inviscid Burgers' equation.

## 4 Conclusions

A new method has been developed to solve the non-linear partial differential equations. In this new approach semi-discretization of partial differential equation in spatial variable is done by MOL. In this way, we find a system of ordinary differential equations which is solved by SSP-RK54 scheme. To check the efficiency and accuracy of the method, two test examples of Burgers' equation are considered and numerical solutions are compared at various values of  $Re$ . The numerical errors are also evaluated in  $L_2$  and  $L_\infty$  norms. It is evident that proposed method produces better results and approach to the exact solution. The present numerical experiments have confirmed that the proposed method

is unconditionally stable even for high Reynold numbers, where other methods are found to be fail. This method may also be implemented to solve numerically higher dimensional nonlinear partial differential equations in fluid mechanics.

## References

- [1] H. Bateman, Some recent researches on the motion of fluids, *Monthly Weather Rev.* 43 (1915) 163–170.
- [2] J. Burger, A mathematical model illustrating the theory of turbulence, *Adv. in Appl. Mech. I*, Academic Press, New York (1948) 171–199.
- [3] Y. A. Cengel, J. M. Cimbala, *Fluid Mechanics, Fundamentals and Application*, McGraw-Hill, New York.
- [4] M. Kadalbajoo, A. Awasthi, A numerical method based on crank-nicolson scheme for burgers' equation, *Appl. Math. Comput.* 182 (2006) 1430–1442.
- [5] L. A. Kurtz, R. E. Smith, C. L. Parks, L. R. Boney, A comparison of the method of lines to finite difference techniques in solving time-dependent partial differential equations, *Computers and Fluids* 6 (1977) 49–70.
- [6] O. Oymak, N. Selcuk, Method-of-lines solution of time-dependent two-dimensional navier-stokes equations, *International journal for numerical methods in fluids* 23 (1996) 455–466.
- [7] E. Rothe, Zweidimensionale parabolisehe randwertaufgaben als grenzfall eindimensionaler randwertaufgaben, *Mathematische Annalen* 102 (1930) 650–670.
- [8] L. F. Shampine, Ode solvers and the method of lines, *Numerical Methods for Partial Differential Equations* 10 (1994) 739–755.
- [9] C. W. Shu, Total-variation-diminishing time discretizations, *SIAM J. Sci. Statist. Comput.* 9 (1988) 1073–1084.
- [10] R. Spiteri, S. J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, *SIAM J. Numer. Anal.* 40 (2002) 469–491.
- [11] D. S. Zhang, G. W. Wei, D. J. Kouri, D. K. Hoffman, Burgers equation with high reynolds number, *Phys. Fluids* 9(3) (1997) 1853–1855.