ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 14, No. 2, 2019, *pp.*149-155

Wavelet Based Full Approximation Scheme for the Numerical Solution of Burgers' equation arising in Fluid Dynamics using Biorthogonal wavelet

S. C. Shiralashetti [1], L. M. Angadi [2] and A.B. Deshi[3]
[1] Department of Mathematics, Karnatak University Dharwad-580003, India
[2] Department of Mathematics, Govt. First Grade College, Chikodi – 591201, India
[3]Department of Mathematics, KLECET, Chikodi – 591201, India
(Received April 26, 2019, accepted May 29, 2019)

Abstract: Wavelet based methods are the new development in the area of applied mathematics. Wavelets are mathematical tools that cut functions or operators into different frequency components, and then study each component with a resolution matching to its scale. In this paper, we proposed Biorthogonal wavelet based full-approximation scheme for the numerical solution of Burgers' equation arising in fluid dynamics using biorthogonal wavelet filter coefficients as prolongation and restriction operators. The proposed method gives higher accuracy in terms of better convergence with low computational time, which has been demonstrated through the illustrative problem.

Keywords: Biorthogonal wavelet; Multi-resolution analysis; Full approximation scheme; Burgers' equation; Fluid dynamics

1. Introduction

Partial differential equations (PDEs) arise frequently in Mathematical Physics and all branches of Engineering and other allied areas. For this reason the study of such equations is of great important. In this paper, we consider the solution of Burgers' equation, which is nonlinear Parabolic PDE. These equations are time dependent i.e. the solution depends not only on the spatial variable, but also the time dependent. The one dimensional Burgers' equation [1] is,

$$u_t + uu_x = \upsilon u_{xx}, \tag{1.1}$$

where $'\upsilon'$ is the coefficient of kinematic viscosity. This is the suitable model for considering problems associated with nonlinear convective terms, possesses the same form of convective linearity as the inco--mpressible Navier–Stoke's equation and has readily evaluated to get exact solutions for many combinations of initial and boundary conditions.

In last two decades, many authors are applied various numerical methods to solve Burgers' equations, some of them are finite element method [2], Least-squares quadratic B-spline finite element method [3] etc. For large systems, these methods are inefficient in terms of both computer storage and computational cost.

In numerical analysis and computational science the development of effective iterative solvers for nonlinear systems of algebraic equations has been a significant research topic. Classical multigrid begins with a two-grid process. First, iterative relaxation is applied, whose effect is to smooth the error. In this paper, we describe how to apply multigrid to nonlinear problems. Applying multigrid method directly to the nonlinear problems by employing the method so-called Full Approximation Scheme (FAS). Nowadays it is recognized that FAS iterative solvers are highly efficient for nonlinear differential equations introduced by Brandt [4]. Full approximation scheme is suitable for nonlinear problems, which treats directly the nonlinear equations on finer and coarser grids. In FAS, a nonlinear iteration, such as the nonlinear Gauss-Seidel method is applied to smooth the error and the residual is passed from the fine grids to the coarser grids. Vectors from fine grids are transferred to coarser grids with Restriction operator (R), while vectors are transferred from coarse grids to the finer grids with a Prolongation operator (P) respectively. For a detailed treatment of FAS is given in Briggs et al. [5]. An introduction of FAS is found in Hackbusch et al. [6] and Wesseling [7]. Many authors applied the FAS for some class of differential equations. The full-approximation scheme (FAS) is largely applicable in increasing the efficiency of the iterative methods used to solve nonlinear system of algebraic equations. FAS are well-founded numerical method for solving nonlinear system of equations for approximating given differential equation. In this paper, we developed the full-approximation scheme (FAS) for the numerical solution of Burgers' equation. However, when problems with discontinuous or highly oscillatory coefficients, multigrid procedure converge slowly with larger computational time or may break down. To overcome this difficulty, wavelet plays a very important role.

"Wavelets" have been very popular topic of conversations in many scientific and engineering gatherings these days. Some of the researchers have decided that, wavelets as a new basis for representing functions, as a technique for time-frequency analysis, and as a new mathematical subject. Of course, "wavelets" is a versatile tool with very rich mathematical content and great potential for applications. However wavelet analysis is a numerical concept which allows one to represent a function in terms of a set of basis functions, called wavelets, which are localized both in location and scale. The development of multiresolution analysis and the fast wavelet transforms by Avudainayagam and Vani [8] and Bujurke et al. [9-10] led to extensive research in wavelet multigrid schemes to solve certain differential equations arising in fluid dynamics. Shiralashetti et al. [11] had proposed Wavelet based multigrid method for the numerical solution of poisson equations.

This paper outspreads the same approach for the numerical solution of Burgers' equation arising in fluid dynamics i.e. Biorthogonal wavelet based full-approximation scheme (BWFAS) for the numerical solution of Burgers' equation. BWFAS is formulated rather than having one scaling and wavelet function, there are two scaling functions that may generate different multiresolution analysis, and accordingly two different wavelet functions.

The dual scaling and wavelet functions have the following properties:

- They are zero outside of a segment.
- The calculation algorithms are maintained, and thus very simple.
- The associated filters are symmetrical.
- The functions used in the calculations are easier to build numerically than those used in the Daubechies wavelets.

The organization of the paper is as follows. In section 2, Properties of Biorthogonal wavelets are given. Section 3 deals with Biorthogonal spline wavelet operators. Section 4 describes the method of solution. Numerical findings and error analysis are presented in section 5. Finally, conclusions of the proposed work are given in section 6.

2. Properties Biorthogonal Wavelets

The framework of the theory of orthonormal wavelets to the case of biorthogonal wavelets by a modification of the approximation space structure is extended by Cohen et al. [12]. In [13], Ruch and Fleet build a biorthogonal structure called dual multiresolution analysis that allows for the construction of symmetric scaling filters and that can incorporate spline functions. They used instead of scaling $\{a_n\}$ and wavelet $\{b_n\}$ filters, the new construct yields scaling $\{\tilde{a}_n\}$ and wavelet $\{\tilde{b}_n\}$ filters as decomposition and reconstruction. Instead of a single scaling function $\phi(x)$ and wavelet function $\psi(x)$, the dual multiresolution analysis requires a pair of scaling functions $\phi(x)$ and $\tilde{\phi}(x)$ related by a duality condition similarly, a pair of wavelet functions $\psi(x)$ and $\tilde{\Psi}(x)$. To construct the BDWT matrix, the same thing is used in to build the orthogonal discrete wavelet transform matrix. Due to excellent properties of biorthogonality and minimum compact support, CDF wavelets can be useful and convenient, providing guaranty of convergence and accuracy of the approximation in a wide variety of situations.

Let's consider the (5, 3) biorthogonal spline wavelet filter pair,

We have $\tilde{a} = (\tilde{a}_{-1}, \tilde{a}_0, \tilde{a}_1) = \left(\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}}\right)$

and $a = (a_{-2}, a_{-1}, a_0, a_1, a_2) = \left(\frac{-1}{4\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{3}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{-1}{4\sqrt{2}}\right)$

To form the highpass filters. We have $b_k = (-1)^k \tilde{a}_{1-k}$ and $\tilde{b}_k = (-1)^k a_{1-k}$.

The highpass filter pair b_k and $\tilde{b}_k$ for the (5, 3) biorthogonal spline filter pair.

$$b_0 = \frac{1}{2\sqrt{2}}, b_1 = \frac{-1}{\sqrt{2}}, b_2 = \frac{1}{2\sqrt{2}} \text{and } \tilde{b}_{-1} = \frac{1}{4\sqrt{2}}, \tilde{b}_0 = \frac{1}{2\sqrt{2}}, \tilde{b}_1 = \frac{-3}{2\sqrt{2}}, \tilde{b}_2 = \frac{1}{2\sqrt{2}}, \tilde{b}_3 = \frac{1}{4\sqrt{2}}$$

In this paper, we use the filter coefficients which are, Low pass filter coefficients: $a_{-2}, a_{-1}, a_0, a_1, a_2$ and High pass filter coefficients: b_0, b_1, b_2 for decomposition matrix. Low pass filter coefficients: $\tilde{a}_{-1} = b_2, \tilde{a}_0 = -b_1, \tilde{a}_1 = b_0$ and High pass filter coefficients: $\tilde{b}_{-1} = -a_2, \tilde{b}_0 = a_1, \tilde{b}_1 = -a_0, \tilde{b}_2 = a_{-1}, \tilde{b}_3 = -a_{-2}$ for reconstruction matrix.

Discrete wavelet transforms (DWT):

A DWT is a linear transformation that transforms vectors from the standard basis to a wavelet basis. Certain classes of linear operators that correspond to dense matrices in the standard basis can be approximated by sparse matrices in a suitably chosen wavelet basis. The matrix formulation of the discrete biorthogonal spline wavelet transforms (DBSWT) plays an important role in the biorthogonal Spline wavelet full-approximation transform method for the numerical computations. As we already know about the DBSWT matrix and its applications in the wavelet method and is given in [14] as,

Decomposition matrix: Reconstruction matrix:

$$DM_{\mathbf{B}} = \begin{pmatrix} a_{-1} & a_0 & a_1 & a_2 & 0 & 0 & \dots & 0 & 0 & a_{-2} \\ b_1 & b_2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & b_0 \\ 0 & a_{-2} & a_{-1} & a_0 & a_1 & a_2 & \dots & 0 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \ddots & & \dots & & \dots & & & 0 & 0 \\ a_1 & a_2 & 0 & 0 & \dots & \dots & 0 & a_{-2} & a_{-1} & a_0 \\ 0 & 0 & 0 & 0 & \dots & \dots & 0 & b_0 & b_1 & b_2 \end{pmatrix}_{N\times N} \quad \text{and} \quad RW_B = \begin{pmatrix} \tilde{a}_0 & \tilde{a}_1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \tilde{a}_{-1} \\ \tilde{b}_0 & \tilde{b}_1 & \tilde{b}_2 & \tilde{b}_3 & 0 & 0 & \dots & 0 & 0 & \widehat{b}_{-1} \\ 0 & \tilde{a}_{-1} & \tilde{a}_0 & \tilde{a}_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \tilde{b}_{-1} & \tilde{b}_0 & \tilde{b}_1 & \tilde{b}_2 & \tilde{b}_3 & \dots & 0 & 0 & 0 \\ \vdots & \ddots & & & \dots & \dots & & & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & \dots & 0 & \tilde{a}_{-1} & \tilde{a}_0 & \tilde{a}_1 \\ \tilde{b}_2 & \tilde{b}_3 & 0 & 0 & \dots & \dots & 0 & \tilde{b}_{-1} & \tilde{b}_0 & \tilde{b}_1 \end{pmatrix}_{N\times N}$$

Using these matrices, we introduced restriction and prolongation operators respectively similar to multigrid restriction and prolongation operators and the detailed procedure is explained in section 4.

3. Biorthogonal Spline Wavelet Operators

Using these matrices, we introduced biorthogonal spline wavelet restriction and biorthogonal spline wavelet prolongation operators respectively. i.e.,

$$BSW_R = \begin{pmatrix} a_{-1} & a_0 & a_1 & a_2 & 0 & 0 & 0 & \dots & 0 & a_{-2} \\ b_0 & b_1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & b_{-1} \\ 0 & a_{-2} & a_{-1} & a_0 & a_1 & a_2 & 0 & \dots & 0 & 0 \\ 0 & b_{-1} & b_0 & b_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \ddots & & & \dots & & \dots & & 0 & 0 \\ 0 & 0 & \dots & 0 & b_{-1} & b_0 & b_1 & 0 & \dots & 0 & 0 \end{pmatrix}_{\frac{N}{2}\times N} \quad \text{and} \quad BSW_P = \begin{pmatrix} \tilde{a}_0 & \tilde{a}_1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \tilde{a}_{-1} \\ \tilde{b}_0 & \tilde{b}_1 & \tilde{b}_2 & \tilde{b}_3 & 0 & 0 & 0 & \dots & 0 & \tilde{b}_{-1} \\ 0 & \tilde{a}_{-1} & \tilde{a}_0 & \tilde{a}_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \tilde{b}_{-1} & \tilde{b}_0 & \tilde{b}_1 & \tilde{b}_2 & \tilde{b}_3 & 0 & \dots & 0 & 0 \\ \vdots & & \ddots & & & \ddots & & \ddots & & \vdots \\ 0 & \dots & 0 & 0 & \tilde{b}_{-1} & \tilde{b}_0 & \tilde{b}_1 & \tilde{b}_2 & \tilde{b}_3 & 0 & \dots & 0 \end{pmatrix}^{T}_{\frac{N}{2}\times N}$$

4. Method of solution

Consider the Burgers' equation,

$$u_t + uu_x = \upsilon u_{xx}, \quad 0 \le x \le 1, \quad t > 0 \tag{4.1}$$

subject to initial condition (IC) and boundary conditions (BCs).

Where u the real valued function and that is is assumed to be in L2(R) in the interval $0 \le x, t \le 1$. We assume that Eq. (4.1) has a unique solution i.e. u to be determined.

Now discretizing the Eq. (4.1) by using finite difference scheme, we get the system of nonlinear equations of the form,

$$F\left(u_{i,j}\right) = b_{i,j} \tag{4.2}$$

where $i, j = 1,2, \dots\dots N$, which have $N \times N$ equations with $N \times N$ unknowns.

Solving Eq. (4.2) through the iterative method, we get approximate solution v for u.

Approximate solution contains some errors, and therefore required solution equals to sum of approximate solution and error. There are many methods to minimize such error to get the accurate solution.

Some of them are FAS, WFAS etc. Now we are discussing the method of solution of the above mentioned methods as below.

4.1. Full-Approximation Scheme (FAS)

From the system Eq. (4.2), we get the approximate solution v for u. Now we find the residual as

$$[r]_{N\times N} = [b]_{N\times N} - A([v]_{N\times N}). \tag{4.3}$$

Reduce the matrices in the finer level to coarsest level using Restriction operator and then construct the matrices back to finer level from the coarsest level using Prolongation operator.

$$R_O = \frac{1}{4}\begin{pmatrix} 1 & 2 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & \dots & 0 & 0 \\ \vdots & & \ddots & & & \dots & 0 & 0 \\ 0 & 0 & & \dots & & \dots & 1 & 2 \end{pmatrix}_{N/2\times N}$$

and then construct the matrices back to finer level from the coarsest level using Prolongation operator as.

$$P_O = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 2 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & \vdots \\ 0 & 2 & \vdots & \dots & \vdots \\ 0 & 1 & & & \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 2 \end{pmatrix}_{N\times N/2}$$

From Eq. (4.3), $$[r]_{\frac{N}{2}\times\frac{N}{2}} = [R_O]_{\frac{N}{2}\times N}[\mathrm{r}]_{N\times N}[P_O]_{N\times\frac{N}{2}}.$$

Similarly, $$[v]_{\frac{N}{2}\times\frac{N}{2}} = [R_O]_{\frac{N}{2}\times N}[v]_{N\times N}[P_O]_{N\times\frac{N}{2}}$$

and $$A([v]_{\frac{N}{2}\times\frac{N}{2}} + [e]_{\frac{N}{2}\times\frac{N}{2}}) - A([v]_{\frac{N}{2}\times\frac{N}{2}}) = [r]_{\frac{N}{2}\times\frac{N}{2}}. \tag{4.4}$$

Solve Eq. (4.4) with initial guess '0', we get $[e]_{\frac{N}{2}\times\frac{N}{2}}$.

From Eq. (4.4), $$[r]_{\frac{N}{4}\times\frac{N}{4}} = [R_O]_{\frac{N}{4}\times\frac{N}{2}}[r]_{\frac{N}{2}\times\frac{N}{2}}[P_O]_{\frac{N}{2}\times\frac{N}{4}}$$

Similarly, $$[v]_{\frac{N}{4}\times\frac{N}{4}} = [R_O]_{\frac{N}{4}\times\frac{N}{2}}[v]_{\frac{N}{2}\times\frac{N}{2}}[P_O]_{\frac{N}{2}\times\frac{N}{4}}$$

and $$A([v]_{\frac{N}{4}\times\frac{N}{4}} + [e]_{\frac{N}{4}\times\frac{N}{4}}) - A([v]_{\frac{N}{4}\times\frac{N}{4}}) = [r]_{\frac{N}{4}\times\frac{N}{4}}. \tag{4.5}$$

Solve Eq. (4.5) with initial guess '0', we get $[e]_{\frac{N}{4}\times\frac{N}{4}}$.

Then the procedure is continue up to the coarsest level, we have,

$$[r]_{1\times 1} = [R_O]_{1\times 2}[r]_{2\times 2}[P_O]_{2\times 1}.$$

Similarly, $$[v]_{1\times 1} = [R_O]_{1\times 2}[v]_{2\times 2}[P_O]_{2\times 1}$$

and $$A([v]_{1\times 1} + [e]_{1\times 1}) - A([v]_{1\times 1}) = [r]_{1\times 1}. \tag{4.6}$$

Solve Eq. (4.6) we get, $[e]_{1\times 1}$.

Now correct the solution to the finer level, i.e.

$$[e]_{2\times 2} = [P_O]_{2\times 1}[e]_{1\times 1}[R_O]_{1\times 2},$$

$$[e]_{4\times 4} = [P_O]_{4\times 2}[e]_{2\times 2}[R_O]_{2\times 4}$$

and so on we have, $$[e]_{N\times N} = [P_O]_{N\times\frac{N}{2}}[e]_{\frac{N}{2}\times\frac{N}{2}}[R_O]_{\frac{N}{2}\times N}.$$

Correct the solution with error. $[u]_{N\times N} = [v]_{N\times N} + [e]_{N\times N}$.

This is the required solution of the given Eq. (4.1).

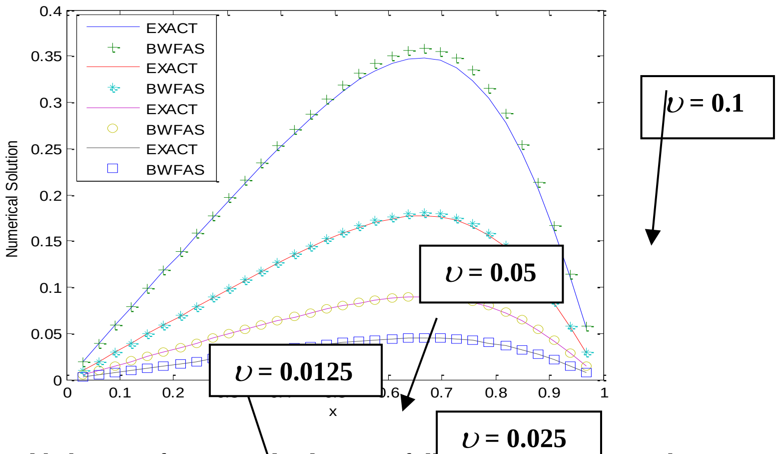


Fig. 1. Physical behavior of numerical solutions of il[illegible]m with $t = 0.01$.

Table 1: Comparison of numerical solutions with exact solution of illustrative problem.

x	t	Numerical solution			Exact solution
		FDM	**FAS**	**BWFAS**	
0.2	0.2	0.012954	0.012954	0.012954	0.012964
0.4	0.2	0.025390	0.025390	0.025390	0.025440
0.6	0.2	0.034358	0.034358	0.034358	0.034524
0.8	0.2	0.029874	0.029874	0.029874	0.030005
0.2	0.4	0.012760	0.012760	0.012760	0.012781
0.4	0.4	0.024906	0.024906	0.024906	0.025007
0.6	0.4	0.033417	0.033417	0.033417	0.033732
0.8	0.4	0.028798	0.028798	0.028798	0.029045
0.2	0.6	0.012567	0.012567	0.012567	0.012600
0.4	0.6	0.024428	0.024428	0.024428	0.024581
0.6	0.6	0.032513	0.032513	0.032513	0.032960
0.8	0.6	0.027777	0.027777	0.027777	0.028128
0.2	0.8	0.012374	0.012374	0.012374	0.012421
0.4	0.8	0.023956	0.023956	0.023956	0.024160
0.6	0.8	0.031644	0.031644	0.031644	0.032209
0.8	0.8	0.026808	0.026808	0.026808	0.027250

4.2. Wavelet Full Approximation Scheme (WFAS)

The same procedure is applied as explained the FAS (Section 4.1) in which replacing operators $BSW_R$and BSW_P in place of R_O and P_O respectively.

5. Numerical Implimentation

In this section, we applied FAS and WFAS for the numerical solution of Burgers' equation and subsequently presented the efficiency of the methods in the form of tables and figures. The error analysis is considered as $E_{max} = max|u_e - u_a|$, where u_e and u_a are exact and approximate solutions respectively.

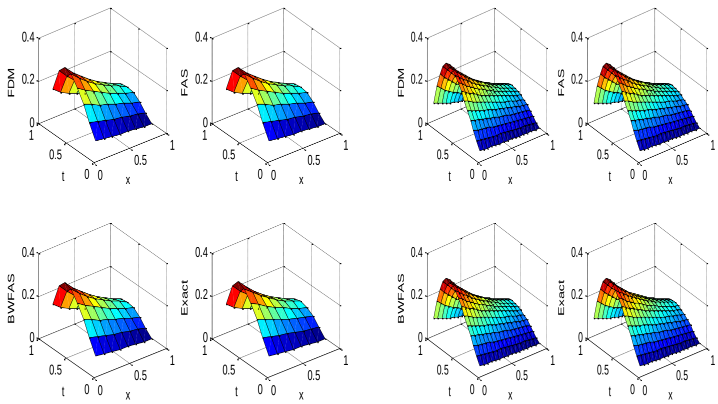
Illustrative Problem: Consider the Burgers equation (4.1) with initial and boundary conditions

subject to the I.C.: $$u(x,0) = \frac{2\upsilon\pi\sin(\pi x)}{2+\cos(\pi x)},\ 0 \le x \le 1 \quad (5.1)$$

and B.C.s: $$u(0,t)=0=u(1,t)\ , \quad 0\le t\le 1 \tag{5.2}$$

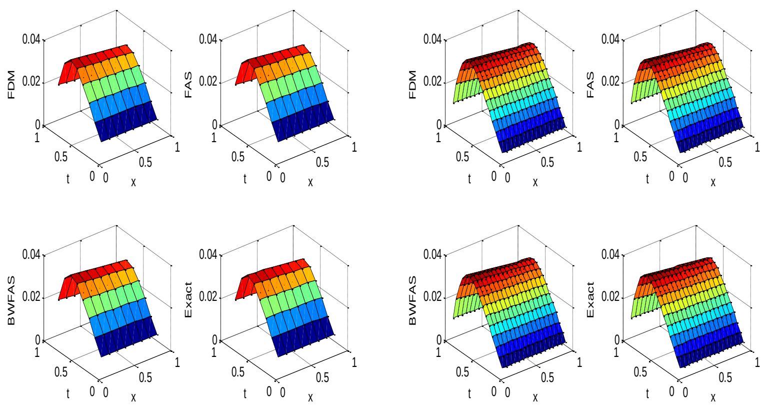
Which has the exact solution $u(x,t) = \dfrac{2\nu\pi e^{-\pi^2 \upsilon t}\sin(\pi x)}{2+e^{-\pi^2\upsilon t}\cos(\pi x)}$ [15].

By applying the method explained in the section 4.1. and 4.2. for different values of υ i.e. $\upsilon =$ 0.1,0.05,0.025,0.0125(with $t = 0.01$), we obtain the numerical solutions of the problem are compared with exact solution is presented in figure 1 and in table 1(for $\upsilon = 0.01$). Also, Physical behavior of numerical solutions of problem in 3D are presented in figures 2 and 3 for $N \times N = 8 \times 8$ & 16×16. The maximum absolute errors with CPU time of the method for $\upsilon = 0.01$ and $\upsilon = 0.0001$are presented in table 2.



$N\times N = 8\times 8$ $N\times N = 16\times 16$

Fig. 2 . Comparison of numerical solutions with exact solution of illustrative problem for $N \times N = 8 \times 8$ & 16×16 when $\upsilon = 0.1$.



$N\times N = 8\times 8$ $N\times N = 16\times 16$

Fig. 3. Comparison of numerical solutions with exact solution of illustrative problem for $N \times N = 8 \times 8$ & $16 \times 16$when $\upsilon = 0.01$.

6. Conclusions

In this paper, we proposed Wavelet based full approximation scheme (BWFAS) for the numerical solution of Burgers' equation arising in fluid dynamics using Biorthogonal wavelets. From the above figures and table shows numerical solutions obtained by BWFAS agree with the exact solution and also gives higher accuracy in terms of better convergence with low computational time than the existing methods for different values of the coefficient of kinematic viscosity ($i.e.\ '\upsilon'$) . Also from the table the error analysis, the convergence of the presented method is observed i.e. the error decreases when the level of resolution N increases and also for smaller values of υ. Hence BWFAS is capable of reducing the volume of the computational work as compared to the classical methods and is very effective for solving non-linear partial differential equations.

Table 2: Maximum error and CPU time (in seconds) of the method of illustrative problem for values $\upsilon = 0.01$ and $\upsilon = 0.0001$.

$N \times N$	Method	**For** $\upsilon = 0.01$				**For** $\upsilon = 0.0001$			
		E_{max}	Setup time	Running time	Total time	E_{max}	Setup time	Running time	Total time
8×8	FDM	6.0965e-04	15.4870	0.0008	15.4878	6.3383e-08	14.2900	0.0015	14.2915
	FAS	6.0965e-04	0.0084	0.0006	0.0090	6.3383e-08	0.0129	0.0005	0.0134
	BWFAS	6.0965e-04	0.0074	0.0002	0.0076	6.3383e-08	0.0072	0.0002	0.0074
16×16	FDM	1.1573e-04	7.6945	0.0008	7.6953	1.1696e-08	3.4785	0.0015	3.4800
	FAS	1.1573e-04	0.0085	0.0005	0.0090	1.1696e-08	0.0120	0.0005	0.0125
	BWFAS	1.1573e-04	0.0073	0.0001	0.0074	1.1696e-08	0.0070	0.0002	0.0072
32×32	FDM	3.0804e-05	7.0325	0.0009	7.0334	3.1054e-09	6.9178	0.0013	6.9181
	FAS	3.0804e-05	0.0091	0.0005	0.0096	3.1054e-09	0.0126	0.0005	0.0131
	BWFAS	3.0804e-05	0.0071	0.0002	0.0073	3.1054e-09	0.0076	0.0002	0.0078
64×64	FDM	1.1117e-05	6.9844	0.0009	6.9852	1.1101e-09	7.9157	0.0009	7.9166
	FAS	1.1117e-05	0.0160	0.0005	0.0165	1.1101e-09	0.0098	0.0005	0.0103
	BWFAS	1.1117e-05	0.0071	0.0002	0.0073	1.1101e-09	0.0075	0.0003	0.0078

References

[1] Sachdev P. L. , Non linear Diffusive Waves, Cambridge University Press (1987).

[2] Ozis T., Aksan E.N., Ozdes A., A finite element approach for solution of Burgers' equation, Appl. Math. Comput. 139 (2003), 417–428.

[3] Kutulay S., Esen A., Dag I., Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method, J. Comput. Appl. Math. 167 (2004), 21–33.

[4] Brandt A., Multi-level adaptive solutions to boundary-value problems. Math. Comp. 31(1977), 333-390.

[5] Briggs W. L., Henson V.E., McCormick S.F., A Multigrid Tutorial, SIAM Philadelphia, 2000.

[6] Hackbusch W., Trottenberg U., Multigrid Methods. Springer-Verlag Berlin, 1982.

[7] Wesseling P., An Introduction to Multigrid Methods. John Wiley Chichester ,1992.

[8] Avudainayagam A., Vani C., Wavelet based multigrid methods for linear and nonlinear elliptic partial differential equations. Appl. Math. Comp. 148 (2004), 307–320.

[9] Bujurke N. M., Salimath C. S., Kudenatti R. B. , Shiralashetti S. C., Analysis of modified Reynolds equation using the wavelet-multigrid scheme. Num. Methods for Partial Diff. Eqn. 23(3) (2006), 692-705.

[10] Bujurke N. M., Salimath C. S., Kudenatti R. B. , Shiralashetti S. C., A fast wavelet- multigrid method to solve elliptic partial differential equations. Appl. Math. Comp. 185(1) (2007), 667-680.

[11] Shiralashetti S. C., Kantli M. H., Deshi A. B. , New wavelet based full-approximation scheme for the numerical solution of nonlinear elliptic partial differential equations, Alexandria Engineering Journal, 55(2016), 2797-2804.

[12] Choen A., Daubechies I., Feauvean J., Biorthogonal based of compactly supported wavelets. Comm. Pure Appl. Math. 45(1992), 485-560.

[13] Ruch D. K., Fleet P. J. V., Wavelet theory an elementary approach with applications. John Wiley and Sons, 2009.

[14] Patrick J, Fleet V., Discrete wavelet transformations an elementary approach with applications. John Wiley and Sons, 2008.

[15] Jain M. K, Iyengar S. R. K, Jain R. K., Computational Methods for Partial Differential Equations, Wiley Eastern Ltd., New Delhi, 1984.