International Journal of Applied Mathematics
Volume 25 No. 5 2012, 653-661

TWO LINEARIZATION TECHNIQUES FOR NUMERICAL SOLUTION OF ONE-DIMENSIONAL NONLINEAR CONVECTION-DIFFUSION EQUATION

Miriane M. Cruz[1], Marco Donisete de Campos[2], Estaner Claro Romão[3 §]

[1]Federal University of Itajubá
Campus of Itabira
Itabira, MG, 35900-395, BRAZIL

[2]Federal University of Mato Grosso
Institute of Exact and Earth Sciences
Barra do Garças, MT 78600-000, BRAZIL

[1]e-mail: miriane.cruz@hotmail.com
[2]e-mail: mcampos@ufmt.br
[3]e-mail: estaner23@yahoo.com.br

Abstract: Numerical simulation methods based on partial differential equations form an important part of contemporary science and are widely used in engineering and scientific applications. In this paper, two linearization techniques are used for the numerical solution of the one-dimensional Burgers equation, which represents a simplified version of nonlinear Navier-Stokes equation, using the Finite Difference Method. For the spatial discretization, a second-order implicit scheme was used. The numerical results were compared with the exact solution that can be found in the Reference.

AMS Subject Classification: 35L60
Key Words: finite difference method, numerical solution, fluid flows, Burgers equation

1. Introduction

During the last decades, the Finite Difference Method has become an important tool to solve problems governed by second order differential equations, especially in fluid flow and heat transfer and mass problems. The approximation of the

Received: August 3, 2012

§Correspondence author

derivative is made by using the Taylor series giving a linear system by which it is possible to obtain the numerical results. In this method, the discretization is performed, at first, in the domain and then, in the derivatives that appear in the differential equation and additional conditions. This allows a continuous problem to be transformed into a finite dimensional and thus to provide conditions to solve the problem from a computational code [5]. Thus, we propose a solution, by the finite differences method, for the one-dimensional Burgers equation (nonlinear one-dimensional convection-diffusion equation), which is given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2}, \tag{1}$$

where $u(x,t)$ is the velocity field in the x-direction, Re is the Reynolds number and $x, t \in R$. In this work, two techniques are proposed to linearize the term called as *Formulation 1* and *Formulation 2*, respectively, and presented below.

2. Formulation 1

To solve Eq. (1), we performed the following linearization

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} \Rightarrow \frac{\partial u}{\partial t} + u\frac{\partial E}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2}, \tag{2}$$

where $E = \dfrac{u^2}{2}$ and, in this expression, the value of u will be used in the previous step of time, i.e., u^n, and thus the term becomes a known value for each space coordinate in Eq. (2). Consider the following equation

$$\frac{\partial u^{n+1}}{\partial t} = \frac{1}{Re}\frac{\partial^2 u^{n+1}}{\partial x^2} - \frac{\partial E^n}{\partial x}, \tag{3}$$

using the Crank-Nicolson method [8] to carry out the time discretization as follows

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{1}{2}\left(\frac{1}{Re}\frac{\partial^2 u}{\partial x^2}\right)^{n+1} + \frac{1}{2}\left(\frac{1}{Re}\frac{\partial^2 u}{\partial x^2}\right)^{n} - \frac{\partial E^n}{\partial x}. \tag{4}$$

Finally, the discretization of Eq. (4) is carried out using the Central Difference Method $O(\Delta x^2)$ [2], as follows

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{1}{2}\left(\frac{1}{Re}\left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2}\right)\right) + \frac{1}{2}\left(\frac{1}{Re}\left(\frac{u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n}}{\Delta x^2}\right)\right) - \frac{\partial E_i^n}{\partial x} \Rightarrow$$

$$\left(\frac{-1}{2Re\Delta x^2}\right)u_{i-1}^{n+1}+\left(\frac{1}{\Delta t}+\frac{1}{Re\Delta x^2}\right)u_i^{n+1}+\left(\frac{-1}{2Re\Delta x^2}\right)u_{i+1}^{n+1}=F_i, \qquad (5)$$

where

$$F_i=\left(\frac{1}{2Re\Delta x^2}\right)u_{i-1}^{n}+\left(\frac{1}{\Delta t}-\frac{1}{Re\Delta x^2}\right)u_i^{n}\left(\frac{1}{2Re\Delta x^2}\right)u_{i+1}^{n}-\frac{\partial E_i^n}{\partial x}. \qquad (6)$$

Eqs. (5) and (6) define the linear system which solve Eq. (1) in the $\Omega \in R^2$ domain, where $\Omega = [0,x]\times[0,t]$.

3. Formulation 2

Rewriting Eq. (1) in the current step, we have

$$\frac{\partial u^{n+1}}{\partial t}+u^{n+1}\frac{\partial u^{n+1}}{\partial x}=\frac{1}{Re}\frac{\partial^2 u^{n+1}}{\partial x^2}. \qquad (7)$$

As in the previous formulation, using the Crank Nicolson time discretization method, Eq. (7) results in

$$\frac{u^{n+1}-u^n}{\Delta t}=\frac{1}{2}\left(\frac{1}{Re}\frac{\partial^2 u^{n+1}}{\partial x^2}-u^{n+1}\frac{\partial u^{n+1}}{\partial x}\right)+\frac{1}{2}\left(\frac{1}{Re}\frac{\partial^2 u^{n}}{\partial x^2}-u^{n+1}\frac{\partial u^{n}}{\partial x}\right). \qquad (8)$$

Now, to linearize the term $u^{n+1}\frac{\partial u^{n+1}}{\partial x}$, we propose the following technique:

$$u^{n+1}\frac{\partial u^{n+1}}{\partial x}\cong u^{n+1}\frac{\partial u^{n}}{\partial x}+u^{n}\frac{\partial u^{n+1}}{\partial x}-u^{n}\frac{\partial u^{n}}{\partial x}. \qquad (9)$$

And, substituting it in Eq. (8), it yields

$$\frac{u^{n+1}-u^n}{\Delta t}=\frac{-1}{2Re}\frac{\partial^2 u^{n+1}}{\partial x^2}+\frac{1}{2}u^n u^n\frac{\partial u^{n+1}}{\partial x}+\left(\frac{1}{\Delta t}\frac{\partial u^n}{\partial x}\right)=F_i, \qquad (10)$$

where $F_i=\dfrac{u^n}{\Delta t}+\dfrac{1}{2Re}\dfrac{\partial^2 u^n}{\partial x^2}$.

Now, discretizing in the space by Central Difference Method of $O(\Delta x^2)$, we have

$$\frac{1}{2Re}\left(\frac{u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}}{\Delta x^2}\right)+\frac{u_i^n}{2}\left(\frac{u_{i+1}^{n+1}-u_{i-1}^{n+1}}{2\Delta x}\right)$$
$$+\left(\frac{1}{\Delta t}+\frac{1}{2}\frac{\partial u_i^n}{\partial x}\right)u_i^{n+1}=F_i, \qquad (11)$$

which can be written as

$$A = \frac{1}{\Delta x} + \frac{1}{2}\frac{u_{i+1}^n + u_{i-1}^n}{2\Delta x}, \tag{12a}$$

$$B = \frac{u_i^n}{2}. \tag{12b}$$

Substituting in Eq. (10), it gives

$$\frac{-1}{2Re}\left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2}\right) + B\left(\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x}\right) + Au_i^{n+1} = F_i, \tag{13}$$

where

$$F_i = \frac{u_i^n}{\Delta t} + \frac{1}{2Re}\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}.$$

Rewriting the Eq. (13), we have

$$\left(\frac{-1}{2Re\Delta x^2} - \frac{B}{2\Delta x}\right)u_{i-1}^{n+1} + \left(\frac{1}{Re\Delta x^2} + A\right)u_i^{n+1} + \left(\frac{-1}{2Re\Delta x^2} + \frac{B}{2\Delta x}u_{i-1}^{n+1}\right)u_{i+1}^{n+1} = F_i. \tag{14}$$

4. Numerical Applications

Application 1. Considering the Burgers equation defined by Eq. (1) with $Re = 100$, the specific solution is given by ([1], [3], [4], [9]):

$$u(x,t) = \frac{\gamma + \mu + (\mu - \gamma)e^n}{1 + e^n}, \quad 0 \leq x \leq 1,\ t \geq 0,$$

where $\eta = \gamma Re(x - \mu t - \beta)$ with the arbitrary constants μ, μ and β have $0.6; 0.4$ and 0.125 as their respective values. The initial and boundary conditions are according to the analytical solution proposed. To analyze the results, presented in Table 1, which were varied values of Δx and Δt for the numerical error from the standard L_∞, which is the maximum error in the entire domain. In this table, note that the spatial and/or temporal refinement carry a better numerical accuracy. Note that for fixed Δt, the refinement of Δx becomes unnecessary for a specified value, for example, for $\Delta t = 1/100$, in which the accuracy begins to stagnate for $\Delta x = 1/320$. Comparing the two formulations, the second

Δx	$\Delta t = 1/100$		$\Delta t = 1/1000$		$\Delta t = 1/10000$	
	Form. 1	**Form. 2**	**Form. 1**	**Form. 2**	**Form. 1**	**Form. 2**
1/20	3.48E-01	6.52E-02	3.92E-01	6.54E-02	3.97E-01	6.55E-02
1/40	1.74E-01	1.81E-02	1.63E-01	1.76E-02	1.60E-01	1.75E-02
1/80	7.69E-02	5.73E-03	4.91E-02	4.99E-03	4.64E-02	4.98E-03
1/160	4.27E-02	2.23E-03	2.15E-02	2.19E-03	1.96E-02	2.21E-03
1/320	2.96E-02	1.89E-03	1.12E-02	1.91E-03	9.61E-03	1.93E-03
1/640	2.42E-02	1.83E-03	6.68E-03	1.86E-03	5.24E-03	1.87E-03
1/1280	2.16E-02	1.81E-03	4.56E-03	1.84E-03	3.21E-03	1.85E-03

Table 1: L_∞ norm of numerical error committed in Application 1.

Δx	**Δt - Formulation 1**			
	1/10	**1/50**	**1/100**	**1/200**
1/10	4.40E-02	1.70E-02	1.43E-02	1.30E-02
1/20	3.67E-02	1.07E-02	8.26E-03	7.09E-03
1/40	3.31E-02	7.64E-03	5.24E-03	4.08E-03
1/80	3.12E-02	6.11E-03	3.74E-03	2.59E-03
1/160	3.03E-02	5.36E-03	3.00E-03	1.85E-03
1/320	2.99E-02	4.98E-03	2.63E-03	1.48E-03

Table 2: L_∞ norm of numerical error committed in Application 2.

stands out for refinements in which $\Delta t = 1/100$ and an order is achieved more accurately.

Application 2. Assuming that $Re = 1$ in Eq. (1), the following analytical solution, will be adopted in this application $u(x,t) = \dfrac{2x}{1+2t}$. As before, the initial and boundary conditions are taken according to the analytical solution. In Table 2, the results of Application 2 are presented. As in Application 1, the Δx variation presents an insignificant accuracy change for a fixed Δt, notwithstanding this situation, good results were found. A precision of the 10^{-3} order was not achieved with a significant refinement in the case of *Formulation 1*.

The numerical results for *Formulation 2* showed extremely small errors. Thus, we conclude that in this example, alternative linearization is viable. The variation of Δx showed accuracy with small change for a set Δt. Nevertheless, excellent results were found, where precision of the order of 10^{-9} was achieved in a refinement which is not significant. Due to this, we find it unnecessary to build a table with the results.

Application 3. We adopted $Re = 1$ in Eq. (1) with the following initial and boundary conditions, respectively, $u(0,t) = u(1,t) = 0$ and $u(x,0) =$

	$\Delta t = 0.00001, \Delta x = 0.025$		$\Delta t = 0.0001, \Delta x = 0.00625$		
x	**Form. 1**	**Form. 2**	**Form. 1**	**Form. 2**	**Exact**
0.1	0.10975	0.10958	0.10957	0.10954	0.10953
0.2	0.21036	0.20985	0.20990	0.20979	0.20979
0.3	0.29290	0.29203	0.29210	0.29190	0.29189
0.4	0.34931	0.34809	0.34822	0.34793	0.34792
0.5	0.37317	0.37177	0.37193	0.37158	0.37157
0.6	0.36058	0.35924	0.35939	0.35905	0.35904
0.7	0.31113	0.31008	0.31019	0.30991	0.30990
0.8	0.22858	0.22795	0.22799	0.22782	0.22781
0.9	0.12099	0.12076	0.12075	0.12069	0.12068

Table 3: Numerical results for some points (Re=1 and t=0.1).

		Δx=1/80; Δt=0.00001		**Δx=1/160; Δt=0.0001**		
x	t	**Form. 1**	**Form. 2**	**Form. 1**	**Form. 2**	**Exact**
0.25	0.1	0.26185	0.26150	0.26164	0.26148	0.26148
0.25	0.15	0.16180	0.16150	0.16163	0.16148	0.16148
0.25	0.20	0.09971	0.09493	0.09958	0.09475	0.09947
0.25	0.25	0.06172	0.06110	0.06116	0.06109	0.06108
0.5	0.1	0.38421	0.38347	0.38380	0.38343	0.38342
0.5	0.15	0.23462	0.23410	0.23432	0.23406	0.23406
0.5	0.20	0.14327	0.14292	0.14303	0.14289	0.14289
0.5	0.25	0.08753	0.08726	0.08735	0.08723	0.08723
0.75	0.1	0.28207	0.28161	0.28182	0.28158	0.28157
0.75	0.15	0.17012	0.16977	0.16912	0.16974	0.16974
0.75	0.20	0.10292	0.10268	0.10278	0.10266	0.10266
0.75	0.25	0.06271	0.06231	0.06237	0.06229	0.06229

Table 4: Numerical comparison of results for Re = 1.

$sen(\pi x)$. The exact results for this application can be found in [9]. In general, for two refinements proposed, both formulations showed good results, according to Table 3. We noted that, for smaller refinement on average, *Formulation 1* showed an accuracy order of about 10^{-3}, while that for *Formulation 2* was about 10^{-4}.

Application 4. In this application, $Re = 1$ (Table 4) and $Re = 10$ (Table 5) values were used,with respective initial and boundary conditions, $u(0,t) = u(1,t) = 0$ and $u(x,0) = 4x(1-x)$. The exact results for this application can be found in [6]. Tables 4 and 5 show good results when compared to the exact solution for two values of Re. We made two combinations of refinement for Δx and Δt, and for greater refinement in Δx, we obtained more significant results.

x	t	Δx=1/80; Δt=0.00001		Δx=1/160; Δt=0.0001		Exact
		Form. 1	**Form. 2**	**Form. 1**	**Form. 2**	
0.25	0.4	0.31594	0.31754	0.31671	0.31752	0.31752
0.25	0.6	0.24494	0.24615	0.24553	0.24614	0.24614
0.25	0.8	0.19884	0.19957	0.19919	0.19956	0.19956
0.25	1.0	0.16461	0.16562	0.16640	0.16560	0.16560
0.5	0.4	0.58714	0.58460	0.58581	0.58455	0.58454
0.5	0.6	0.46018	0.45805	0.45906	0.45799	0.45798
0.5	0.8	0.36967	0.36748	0.36851	0.36741	0.36740
0.5	1.0	0.30722	0.29842	0.29942	0.29836	0.29834
0.75	0.4	0.65600	0.64585	0.65076	0.64567	0.64562
0.75	0.6	0.51190	0.50293	0.50722	0.50274	0.50568
0.75	0.8	0.39250	0.38557	0.38863	0.38539	0.38534
0.75	1.0	0.29908	0.29604	0.29780	0.29590	0.29586

Table 5: Numerical comparison of results for Re = 10.

Δt	**1/20**	**1/40**	**1/80**	**1/160**	**1/320**	**1/640**
[10]	2.00E-5	4.14E-6	1.06E-5	3.16E-5	6.91E-5	1.42E-4
Formulation 1	2.78E-2	2.77E-2	2.77E-2	2.77E-2	2.77E-2	2.77E-2
Formulation 2	1.60E-5	3.86E-6	8.24E-7	6.59E-8	1.25E-7	1.72E-7

Table 6: L_∞ norm for $\Delta x = 1/100$.

Application 5. Taking $Re = 1$ in this application, the initial and boundary conditions are in agreement with the analytical solution $u(x,t) = \dfrac{2\sin x}{\cos x + e^t}$, whose computational domain is and $0 \leq x \leq 1$ and $0 \leq t \leq 1$. In this application, a comparison with the numerical results in [10] is performed. In Table 6, Δx is fixed and Δt values vary, showing that in the present work, the refinement step time is constantly improving the results for *Formulation 2*, whereas for *Formulation 1*, the results are established. However, in [10] this is not true. Now in Table 7, $\Delta t = 1/100$ is fixed and in the range of the Δx values. The numerical results of the two studies show improved results for each Δx, except for *Formulation 1.* It is important to note that in Table 7, *Formulation 2* presents around three more orders of accuracy than the work of [10].

Δx	**1/5**	**1/10**	**1/20**	**1/40**	**1/80**
[10]	6.12E-2	1.10E-2	1.63E-3	2.24E-4	2.93E-5
Formulation 1	2.77E-2	2.77E-2	2.77E-2	2.77E-2	2.77E-2
Formulation 2	7.56E-5	1.88E-5	4.69E-6	1.17E-6	2.88E-7

Table 7: L_∞ norm for $\Delta x = 1/1000$.

5. Conclusions

In this study, we obtain better results in the numerical solution of one-dimensional Burgers equation and those obtained by *Formulation 2* were better than *Formulation 1*, which represents a simplified version of nonlinear Navier-Stokes equation. It can be concluded that for a significant numerical accuracy improvement, a refinement, both in space and in time, is necessary. When one of those variables was fixed, in most cases studied, accuracy was stagnant.

References

[1] A.H.A. Ali, L.R.T. Gardner and G.A. Gardner, A collocation method for Burgers equation using cubic splines, *Comput. Methods Appl. Mech. Engrg.*, **100** (1992), 325-337.

[2] T.J. Chung, *Computational Fluid Dynamics*. Cambridge University Press (2002).

[3] I. Dag, D. Irk and B. Saka, A numerical solution of the Burgers equation using cubic B-splines, *Appl. Math. Comp.*, **163** (2005), 199-211.

[4] A. Dogan, A Galerkin finite element approach to Burgers equation, *Appl. Math. Comp.*, **157** (2004), 331-346.

[5] V.G. Ferreira and G.A.B. Lima, Solução Numérica de Equações Diferenciais Parciais. In: V Bienal da Sociedade Brasileira de Matemática, UFPB (2010), In Portuguese.

[6] I.A. Hassanien, A.A. Salama and H.A. Hosham, Fourth-order finite difference method for solving Burges equation, *Appl. Math. Comp.*, **170** (2005), 781-800.

[7] W. Liao, An implicit fourth-order compact finite difference scheme for one-dimensional Burgers equation, *Appl. Math. Comp.*, **206** (2008), 755-764.

[8] J.N. Reddy, *An Introduction to the Finite Element Method*. McGraw-Hill, Singapore (1984).

[9] M. Xu, R.H. Wang and J.H.Z. Qin Fang, A novel numerical scheme for solving Burgers equation, *Comp. Math. Appl.*, **217**, 4473-4482 (2011), 4473-4482.

[10] J. Wang and A. Layton, New numerical methods for Burgers equation based on semi-Lagrangian and modified equation approaches, *Appl. Num. Math.*, **60** (2010), 645-657.

[11] C.C. Zhu and R.H. Wang, Numerical solution of Burgers equation by cubic B-spline quasi-interpolation, *Appl. Math. Comp.*, **208** (2009), 260-272.