INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **49**:1347–1358
Published online 7 September 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1028

The alternating segment difference scheme for Burgers' equation

Wenqia Wang*,† and Tongchao Lu‡

School of Mathematics and System Sciences, Shandong University, Jinan, 250100, People's Republic of China

SUMMARY

We give a class of alternating segment Crank–Nicolson (ASC-N) method for solving the Burgers' Equation. However, the ASC-N method was discussed only for solving the diffusion equation by Zhang B. The basic idea of the method is that the grid points on same time level is divided into a number of the groups, the difference equations of each group can be solved independently. The method is unconditionally stable by analysis of linearization procedure. The numerical examples show that the accuracy of the method is better than that of the method discussed by the other authors.

KEY WORDS: Burgers' equation; alternating segment scheme; Saul'yev-type asymmetric schemes; Crank–Nicolson scheme

1. INTRODUCTION

In this paper, we consider the Burgers' equation [1]

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=\varepsilon\frac{\partial^2 u}{\partial x^2},\quad 0<x<L,\quad 0<t<T \tag{1}$$

with the following initial condition (2) and boundary conditions (3):

$$u(x,0) = f(x),\quad 0<x<L \tag{2}$$

$$u(0,t) = g_1(t),\quad u(L,t)=g_2(t),\quad 0<t<T \tag{3}$$

where $\varepsilon>0$ is a constant.

*Correspondence to: Wenqia Wang, School of Mathematics and System Sciences, Shandong University, Jinan, 250100, People's Republic of China.
†E-mail: wangwq@sdu.edu.cn
‡E-mail: lutc@sdu.edu.cn

Contract/grant sponsor: Natural Science Foundation, China; contract/grant number: Y2003A04

Received 14 November 2004
Revised 14 May 2005
Accepted 22 May 2005

Burgers' equation is a quasi-linear parabolic partial differential equation. It is one of the very few nonlinear partial differential equation which can be solved exactly for an arbitrary initial and boundary conditions. Burgers' equation and Navier–Stokes equation are similar due to the form of their nonlinear term and the occurrence of higher order derivatives with small coefficients ε in both. Therefore, as an important simple model for understanding physical flows and testing efficiency of the numerical methods of fluid flow, many peoples are interested in the numerical methods for solving this equation, e.g. References [2–12].

In the research of parallel finite-difference methods for a parabolic partial differential equations, the alternating group explicit (AGE) methods and the alternating segment (block) explicit–implicit (AS(B)E-I) methods can be found in References [13, 14]. Using the Crank–Nicolson and Saul'yev-type schemes, Baolin Zhang also constructed the alternating segment Crank–Nicolson (ASC-N) scheme in References [15, 16] for the diffusion equation.

In this paper, we construct the ASC-N scheme for Burgers' equation. The basic idea of the scheme is that the difference schemes at a certain same time level is designed as a number of small size independent linear systems with the Crank–Nicolson scheme and four Saul'yev-type asymmetric schemes, and these small size systems can be computed independently. The key to these schemes is how Saul'yev-type difference schemes are constructed to assure stability of the ASC-N scheme. The method is unconditionally stable by analysis of linearization procedure, and simple and convenient in a practical sense. The numerical examples show that the accuracy of the method is better than that of the existing method in References [8, 9, 11].

2. THE ASC-N SCHEME

Let $h=\Delta x$ and $\tau=\Delta t$ be the mesh sizes in the x and t directions, respectively, where $h=L/m$ and m is a positive integer. Let u_i^n be the approximate solution for problem (1)–(3) computed at the grid points (x_i,t_n), where $x_i=ih(i=0,1,\ldots,m), t_n=n\tau(n=1,2,\ldots), t_{n+1/2}=t_n+\tau/2$. For simplicity, we denote points (x_i,t_n) by (i,n), $(x_i,t_{n+1/2})=(x_i,t_n+\tau/2)$ by $(i,n+\frac{1}{2})$.

In order to construct the ASC-N scheme, we introduce the Crank–Nicolson scheme and four Saul'yev-type asymmetric schemes for Equation (1). For simplicity, define the following notations:

$$\partial_x u_i^n=(u_{i+1}^n-u_i^n)/h,\quad \partial_{\bar{x}} u_i^n=(u_i^n-u_{i-1}^n)/h,\quad \partial_{\hat{x}} u_i^n=(u_{i+1}^n-u_{i-1}^n)/(2h)$$

$$\partial_t u_i^n=(u_i^{n+1}-u_i^n)/\tau,\quad \partial_x^2 u_i^n=(u_{i+1}^n-2u_i^n+u_{i-1}^n)/h^2$$

2.1. The Crank–Nicolson scheme

The Crank–Nicolson scheme of Equation (1) at point $(x_i,t_{n+1/2})$ is

$$\partial_t u_i^n+\frac{\bar{u}_i^n}{2}\left(\partial_{\hat{x}} u_i^{n+1}+\partial_{\hat{x}} u_i^n\right)=\frac{\varepsilon}{2}(\partial_x^2 u_i^{n+1}+\partial_x^2 u_i^n) \tag{4}$$

where $\bar{u}_i^n=u_i^{n+1/2}$ is given in the following way.

By the way in Reference [10], see Figure 1, the path of fluid particle A is $\overline{CA}$, and the velocity of fluid particle at point $A(x_i, t_{n+1/2})$ is given by $u_i^{n+1/2} = u_c^n$. From definition of the velocity, we get

$$u_c^n = (x_i - x_c) \Big/ \left(\frac{\tau}{2}\right) \tag{5}$$

In addition, by linear interpolation formula, we have

$$u_c^n \approx u_i^n + \frac{u_i^n - u_{i-1}^n}{x_i - x_{i-1}}(x_c - x_i) \tag{6}$$

From (5) and (6), we obtain

$$u_c^n \approx \bar{u}_i^n = \frac{u_i^n}{1 + \frac{\tau}{2h}(u_i^n - u_{i-1}^n)} \tag{7}$$

2.2. *Saul'yev-type asymmetric schemes*

The four asymmetric schemes approximating Equation (1) at the point $(i, n+\frac{1}{2})$ are given by

$$\partial_t u_i^n + \frac{\bar{u}_i^n}{2}\left(\partial_{\hat{x}} u_i^n + \frac{u_{i+1}^{n+1} - u_{i-1}^n}{2h}\right) = \frac{\varepsilon}{2}(\partial_x^2 u_i^n + h^{-1}(\partial_x u_i^{n+1} - \partial_{\bar{x}} u_i^n)) \tag{8}$$

$$\partial_t u_i^n + \frac{\bar{u}_i^n}{2}\left(\partial_{\hat{x}} u_i^n + \frac{u_{i+1}^{n} - u_{i-1}^{n+1}}{2h}\right) = \frac{\varepsilon}{2}(\partial_x^2 u_i^n + h^{-1}(\partial_x u_i^{n} - \partial_{\bar{x}} u_i^{n+1})) \tag{9}$$

$$\partial_t u_i^n + \frac{\bar{u}_i^n}{2}\left(\partial_{\hat{x}} u_i^{n+1} + \frac{u_{i+1}^{n+1} - u_{i-1}^n}{2h}\right) = \frac{\varepsilon}{2}(\partial_x^2 u_i^{n+1} + h^{-1}(\partial_x u_i^{n+1} - \partial_{\bar{x}} u_i^n)) \tag{10}$$

$$\partial_t u_i^n + \frac{\bar{u}_i^n}{2}\left(\partial_{\hat{x}} u_i^{n+1} + \frac{u_{i+1}^{n} - u_{i-1}^{n+1}}{2h}\right) = \frac{\varepsilon}{2}(\partial_x^2 u_i^{n+1} + h^{-1}(\partial_x u_i^{n} - \partial_{\bar{x}} u_i^{n+1})) \tag{11}$$

We will now consider the ASC-N method for either the number of inner points $M = 2Jl + l$ or $M = 2Jl$ $(J \geqslant 1, l \geqslant 3)$, where J and l are positive integers, and $M = m - 1$.

1. *Case of* $M = 2Jl + l$. Suppose that n is an even number, and the values u_i^n of the nth time level are given. To compute the approximate values u_i^{n+1} of the solution $u(x,t)$ at the

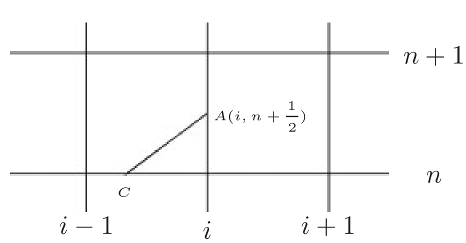

Figure 1. The path of fluid particle.

nodes (x_i, t_{n+1}) of the $(n+1)$st time level and the values u_i^{n+2} of the $(n+2)$nd time level, we construct the ASC-N schemes with formula (4) and (8)–(11). The flow chart of this schemes is displayed in Figure 2. We use • to denote the Crank–Nicolson scheme (4), ⊠ to denote the two asymmetric schemes (8) and (9), and ■ to denote the two asymmetric schemes (10) and (11).

Let $r = \tau/2h^2$, The matrix form of the ASC-N schemes is

$$(\mathbf{I} + r\mathbf{G}_1^{(n)})\mathbf{U}^{n+1} = (\mathbf{I} - r\mathbf{G}_2^{(n)})\mathbf{U}^n + \mathbf{F}_1 \tag{12a}$$

$$(\mathbf{I} + r\mathbf{G}_2^{(n+1)})\mathbf{U}^{n+2} = (\mathbf{I} - r\mathbf{G}_1^{(n+1)})\mathbf{U}^{n+1} + \mathbf{F}_2 \tag{12b}$$

$$n = 0, 2, 4, 6, \ldots$$

where $\mathbf{U}^n = (u_1^n, u_2^n, \ldots, u_{m-1}^n)^{\mathrm{T}}$, $\mathbf{F}_1 = (2rb_1u_0^n, 0, \ldots, 0,\ 2rc_{m-1}u_m^{n+1})^{\mathrm{T}}$, $\mathbf{F}_2 = (2rb_1u_0^{n+2}, 0, \ldots, 0, 2rc_{m-1}u_m^{n+1})^{\mathrm{T}}$, $b_i = \varepsilon + \bar{u}_i^n h/2$, $c_i = \varepsilon - \bar{u}_i^n h/2, \mathbf{U}^n, \mathbf{F}_1, \mathbf{F}_2$ are $(m-1)$ dimensional column vector. The matrices $\mathbf{G}_1^{(n)}$ and $\mathbf{G}_2^{(n)}$ are the block diagonal matrices as follows:

$$\mathbf{G}_1^{(n)} = \mathrm{diag}\{\mathbf{Q}_{2l}^{(1)}, \mathbf{Q}_{2l}^{(2)}, \ldots, \mathbf{Q}_{2l}^{(J)}, \overline{\mathbf{Q}}_l^{(1)}\} \tag{13a}$$

$$\mathbf{G}_2^{(n)} = \mathrm{diag}\{\overline{\mathbf{Q}}_l^{(2)}, \mathbf{Q}_{2l}^{(2)}, \ldots, \mathbf{Q}_{2l}^{(J)}, \mathbf{Q}_{2l}^{(J+1)}\} \tag{13b}$$

where

$$\mathbf{Q}_{2l}^{(j)} = \begin{bmatrix} \varepsilon & -c_1 & & & & & & & & & \\ -b_2 & 2\varepsilon & -c_2 & & & & & & & & \\ & \ddots & \ddots & \ddots & & & & & & & \\ & & -b_{l-1} & 2\varepsilon & -c_{l-1} & & & & & & \\ & & & -b_l & 3\varepsilon & -2c_l & & & & & \\ & & & & -2b_{l+1} & 3\varepsilon & -c_{l+1} & & & & \\ & & & & & -b_{l+2} & 2\varepsilon & -c_{l+2} & & & \\ & & & & & & \ddots & \ddots & \ddots & & \\ & & & & & & & -b_{2l-1} & 2\varepsilon & -c_{2l-1} \\ & & & & & & & & -b_{2l} & \varepsilon \end{bmatrix}_{2l\times 2l,} \quad j = 1, 2, \ldots, J$$

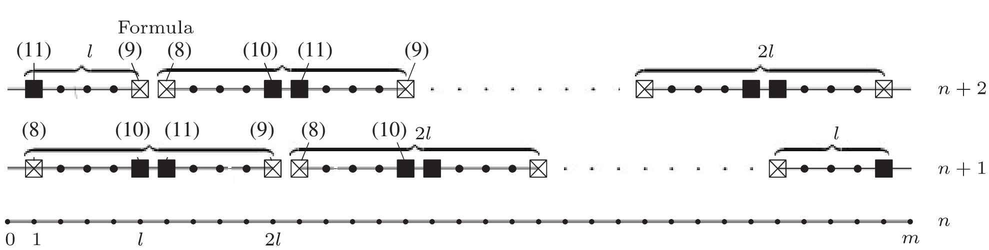

Figure 2. The diagram of the ASC-N scheme for $M = 2Jl + l$.

$$\overline{\mathbf{Q}}_l^{(1)} = \begin{bmatrix} \varepsilon & -c_1 & & & \\ -b_2 & 2\varepsilon & -c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -b_{l-1} & 2\varepsilon & -c_{l-1} \\ & & & -b_l & 3\varepsilon \end{bmatrix}_{l\times l}, \quad \overline{\mathbf{Q}}_l^{(2)} = \begin{bmatrix} 3\varepsilon & -c_1 & & & \\ -b_2 & 2\varepsilon & -c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -b_{l-1} & 2\varepsilon & -c_{l-1} \\ & & & -b_l & \varepsilon \end{bmatrix}_{l\times l}$$

In above formula, b_i and c_i depend the ordinal number j of the segment, e.g. b_i and c_i are $b_{n_{j-1}+i}$ and $c_{n_{j-1}+i}$ on the $(n+1)$st time level or $(n+2)$nd time level, respectively, where n_{j-1} is the ordinal number of the right-endpoint of the $(j-1)$st segment.

From formula (13) we see that the matrices $\mathbf{G}_1^{(n)}$ and $\mathbf{G}_2^{(n+1)}$ are block diagonal matrices, and the ASC-N method decomposes the problem of size $(m-1)$ into a number of small problems of size $2l$ or size l (see Figure 2 and formula (13)), hence this method is intrinsic parallelism.

We may use alternatively the difference scheme (12a) for the $(n+1)$st level and the difference scheme (12b) for the $(n+2)$nd level to find the approximate values of the solution of the problem (1)–(3) starting from the initial time level and using the boundary conditions, and formula (12) is the matrix form for the ASC-N scheme.

2. *Case of* $M=2Jl$. To compute the approximate values u_i^{n+1} of the solution $u(x,t)$ at the grid point (x_i, t_{n+1}) of the $(n+1)$st time level and the values u_i^{n+2} of the $(n+2)$nd time level, we group the grid points at the $(n+1)$st time level into J segments, each segment consisting of $2l$ grid points, then group the grid points at the $(n+2)$nd time level into $(J+1)$ segments. The first segment and $(J+1)$st segment consist of l grid points, the rest of the segments consist of $2l$ grid points. The difference schemes of each segment are chosen as displayed in Figure 3, resulting in the following difference schemes:

$$(\mathbf{I}+r\hat{\mathbf{G}}_1^{(n)})\mathbf{U}^{n+1} = (\mathbf{I}-r\hat{\mathbf{G}}_2^{(n)})\mathbf{U}^n + \mathbf{F}_1 \tag{14a}$$

$$(\mathbf{I}+r\hat{\mathbf{G}}_2^{(n+1)})\mathbf{U}^{n+2} = (\mathbf{I}-r\hat{\mathbf{G}}_1^{(n+1)})\mathbf{U}^{n+1} + \mathbf{F}_2 \tag{14b}$$

$$n = 0, 2, 4, 6, \ldots$$

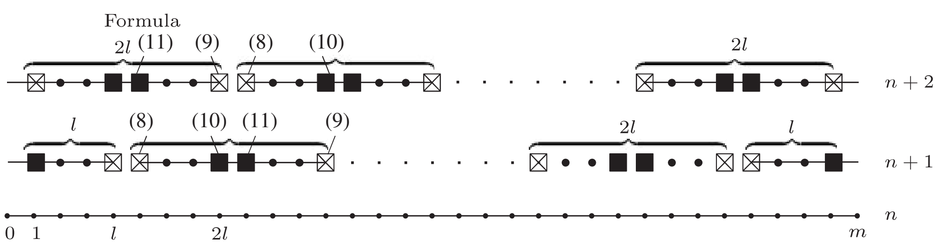

Figure 3. The diagram of the ASC-N scheme for $M=2Jl$.

where $\hat{\mathbf{G}}_1^{(n)}$ and $\hat{\mathbf{G}}_2^{(n)}$ are the block diagonal matrices.

$$\hat{\mathbf{G}}_1^{(n)} = \text{diag}\{\overline{\mathbf{Q}}_l^{(1)}, \mathbf{Q}_{2l}^{(2)}, \ldots, \mathbf{Q}_{2l}^{(J)}, \overline{\mathbf{Q}}_l^{(2)}\} \tag{15a}$$

$$\hat{\mathbf{G}}_2^{(n)} = \text{diag}\{\overline{\mathbf{Q}}_{2l}^{(2)}, \mathbf{Q}_{2l}^{(2)}, \ldots, \mathbf{Q}_{2l}^{(J-1)}, \overline{\mathbf{Q}}_{2l}^{(1)}\} \tag{15b}$$

3. LINEAR STABILITY ANALYSIS AND NUMERICAL EXAMPLES

To analyse the linear stability of the ASC-N method, assume that $\bar{u}_i^n = a$ is a constant, and the boundary condition is in full precision. Thus $G_1^{(n)} = G_1^{(n+1)} = G_1, G_2^{(n)} = G_2^{(n+1)} = G_2$, and we may suppose $g_1(t) = g_2(t) = 0$. Formula (12) can be rewritten as

$$\mathbf{U}^n = \mathbf{G}\mathbf{U}^{n-2}$$

where $\mathbf{G}$ is the growth matrix, $\mathbf{G} = (\mathbf{I} + r\mathbf{G}_2)^{-1}(\mathbf{I} - r\mathbf{G}_1)(\mathbf{I} + r\mathbf{G}_1)^{-1}(\mathbf{I} - r\mathbf{G}_2)$, let

$$\tilde{\mathbf{G}} = (\mathbf{I} + r\mathbf{G}_2)\mathbf{G}(\mathbf{I} + r\mathbf{G}_2)^{-1} = (\mathbf{I} - r\mathbf{G}_1)(\mathbf{I} + r\mathbf{G}_1)^{-1}(\mathbf{I} - r\mathbf{G}_2)(\mathbf{I} + r\mathbf{G}_2)^{-1}$$

It is easy to prove easily that the matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ are nonnegative real matrices. By Kellogg lemmas [17]

$$\|(\mathbf{I} - r\mathbf{G}_i)(\mathbf{I} + r\mathbf{G}_i)^{-1}\|_2 \leqslant 1, \quad i = 1, 2$$

Hence,

$$\rho(\mathbf{G}) = \rho(\tilde{\mathbf{G}}) \leqslant \|(\mathbf{I} - r\mathbf{G}_1)(\mathbf{I} + r\mathbf{G}_1)^{-1}\|_2 \|(\mathbf{I} - r\mathbf{G}_2)(\mathbf{I} + r\mathbf{G}_2)^{-1}\|_2 \leqslant 1$$

where $\rho(G)$ and $\rho(\tilde{G})$ are the spectral radii of the matrices G and $\tilde{G}$, respectively.

Therefore, the method given by (12) is unconditionally stable by the analysis of liberalization procedure.

Similarly, we can prove the linear stability of the scheme (14).

To explain the accuracy and stability of the ASC-N scheme for the Burgers' equation, we now perform the following numerical experiments.

Example 1

In this example, Equation (1) has the exact solution as given in Reference [9]

$$u(x, t) = \frac{0.1\mathrm{e}^{-A} + 0.5\mathrm{e}^{-B} + \mathrm{e}^{-C}}{\mathrm{e}^{-A} + \mathrm{e}^{-B} + \mathrm{e}^{-C}}, \quad 0 \leqslant x \leqslant 1, \ \ t \geqslant 0 \tag{16}$$

where $A = (0.05/\varepsilon)(x - 0.5 + 4.95t)$, $B = (0.25/\varepsilon)(x - 0.5 + 0.75t)$, $C = (0.5/\varepsilon)(x - 0.375)$.

We compare the ASC-N scheme given in this paper with the scheme in Reference [9] in terms of their absolute errors (A.E.) and numerical solution, where the absolute error (A.E.)

Table I. The absolute errors of numerical solutions to Burgers' equation for Example 1, $m=10(h=0.1)$, $l=3, t=1.0, \varepsilon=0.1, \lambda=\tau/h^2$.

x_j	ASC-N Equation (12) $\lambda=2.5$	PR (D)AGE [9] $\lambda=1.0$	DR AGE-IMP [9] $\lambda=1.0$	Exact AGE-CN [9] $\lambda=1.0$	Solution
0.1	1.88×10^{-4}	1.83×10^{-4}	2.9×10^{-2}	1.60×10^{-4}	0.932745
0.2	3.52×10^{-4}	3.26×10^{-4}	6.1×10^{-2}	2.66×10^{-4}	0.911271
0.3	5.31×10^{-4}	3.88×10^{-4}	9.4×10^{-2}	2.59×10^{-4}	0.883314
0.4	1.02×10^{-4}	2.93×10^{-4}	1.236×10^{-1}	7.17×10^{-5}	0.847514
0.5	1.88×10^{-4}	9.6×10^{-5}	1.4498×10^{-1}	3.3×10^{-4}	0.802758
0.6	8.56×10^{-4}	6.94×10^{-4}	1.542×10^{-1}	8.94×10^{-4}	0.748601
0.7	1.42×10^{-3}	1.19×10^{-3}	1.4769×10^{-1}	1.45×10^{-3}	0.685736
0.8	1.85×10^{-3}	1.51×10^{-3}	1.2229×10^{-1}	1.71×10^{-3}	0.616304
0.9	1.46×10^{-3}	1.22×10^{-3}	7.485×10^{-2}	1.34×10^{-3}	0.543775
Number of iterations	No iteration	3	5	13	

Table II. The numerical solutions to Burgers' equation for Example 1, $m=100(h=0.01)$, $l=11, t=0.5, \varepsilon=0.003, \lambda=\tau/h^2$.

x_j	ASC-N Equation (12) $\lambda=50.0$	(D)AGE [9] $\lambda=1.0$	PR AGE-IMP [9] $\lambda=1.0$	DR AGE-CN [9] $\lambda=1.0$	Exact solution
0.1	1.00000	1.0000	0.999999	1.000000	1.000000
0.2	1.00000	1.0000	0.999999	0.999999	1.000000
0.3	1.00000	1.0000	0.999995	0.999999	1.000000
0.4	1.00000	1.0000	0.992646	0.999999	1.000000
0.5	1.00000	1.0000	0.620463	1.000001	0.999985
0.6	0.95298	0.9552	0.360375	0.953063	0.941313
0.7	0.11430	0.1145	0.109650	0.114373	0.113837
0.8	0.10003	0.1000	0.100049	0.100026	0.100018
0.9	0.10000	0.1000	0.100000	0.100000	0.100000
Number of iterations	No iteration		2	10	

is defined by

$$e_j^n = |u_j^n - u(x_j, t_n)|$$

To compare the results for Example 1, we take the same data as that in Reference [18] for parameter ε and space step h. The numerical results are given in Tables I and II. All data in Tables I and II, except that the ones computed with the ASC-N scheme and the exact solution, are taken from Reference [9].

We can see from Table II that the accuracy of the ASC-N scheme is good for large grid ratio $\lambda(\lambda=\tau/h^2)$. This show that the stability of the ASC-N scheme is good. It agrees with the previous stability analysis. These results imply that the ASC-N scheme may use large time step.

Note that the method is simple and convenient in a practical sense, and it avoids the problem of choosing iterative parameters as in Reference [9].

Example 2

We consider Equation (1) with the following initial-boundary conditions:

$$u(x,0) = \sin(\pi x)$$

$$u(0,t) = u(1,t) = 0$$

This problem has the (exact) Fourier series solution

$$u(x,t) = 2\pi\varepsilon \frac{\sum_{k=1}^{\infty} a_k \exp(-k^2\pi^2\varepsilon t)k\, \sin(k\pi x)}{a_0 + \sum_{k=1}^{\infty} a_k \exp(-k^2\pi^2\varepsilon t)\cos(k\pi x)} \tag{17}$$

with Fourier coefficients

$$a_0 = \int_0^1 \exp(\{-(3\pi\varepsilon)^{-1}[1-\cos(\pi x)]\}\, \mathrm{d}x$$

$$a_k = \int_0^1 \exp(\{-(3\pi\varepsilon)^{-1}[1-\cos(\pi x)]\})\cos(k\pi x)\, \mathrm{d}x \quad (k = 1,2,3,\ldots)$$

For this problem, we first compare the ASC-N solutions, the EFD solution and the exact-EFD solution [8]. The numerical results are displayed in Tables III and IV, where the EFD and the exact-EFD represent the explicit finite-difference scheme and the exact-explicit finite-difference scheme in Reference [8], respectively. In Reference [8], Kutluay *et al.*, transformed the Burgers' equation into the linear diffusion equation by using the Cole–Hoft transformation [18], then the linear equation has been solved by the EFD and the exact-EFD schemes. The ASC-N solution in Tables III and IV are obtained by the ASC-N scheme in this paper; the rest of the results are taken from Reference [8].

Table III shows the numerical results for $\varepsilon = 1.0$ with $h = 0.1$ or 0.025 at the time $t = 0.1$. In calculation, the time-step length of the ASC-N scheme is 500 times (or 200 times) as the time-step length used in Reference [8] for $h = 0.1$ (or $h = 0.025$), and the accuracy of the ASC-N solution is better than the EFD and exact-EFD solution. This shown that the stable of the ASC-N scheme for the Burgers' equation is very good. Table IV list the numerical results for small value $\varepsilon = 0.01$ with $h = 0.0125$ at different times. The numerical results show as same conclusion as in Table III.

Next, in order to show that the numerical solutions exhibit the correct physical behaviour, we show the graphs of the numerical solutions in Figure 4 for diffusion coefficient $\varepsilon = 0.005$.

Finally, we examine the convergence rate of the ASC-N scheme. The errors $e_h = \|\mathbf{u}^n - \mathbf{u}\|_{L^2} = \left(\sum_{i=1}^{m} |u(x_i,t_n) - u_i^n|^2 h\right)^{1/2}$ at $t = 0.4$ are displayed in Table V for different mesh refinements, where $\mathbf{u}^n$ is the numerical solution, $\mathbf{u}$ is the values of exact solution at the grid points. The errors appear to be order $O(h^2)$.

Table III. Comparison of the numerical solutions for Example 2 for $\varepsilon = 1.0$ at $t = 0.1$.

	$m = 10(h = 0.1)$			$m = 40(h = 0.025)$			
x_j	ASC-N($l = 3$) $\tau = 0.005$ (20 steps)	EFD [8] $\tau = 0.00001$ (10 000 steps)	Exact-EFD [8] $\tau = 0.00001$ (10 000 steps)	ASC-N($l = 13$) $\tau = 0.002$ (50 steps)	EFD [8] $\tau = 0.00001$ (10 000 steps)	Exact-EFD [8] $\tau = 0.00001$ (10 000 steps)	Exact solution
0.1	0.11017	0.10863	0.11048	0.10955	0.10948	0.10959	0.10954
0.2	0.21106	0.20805	0.21159	0.20982	0.20967	0.20989	0.20979
0.3	0.29414	0.28946	0.29435	0.29195	0.29173	0.29204	0.29190
0.4	0.34943	0.34501	0.35080	0.34757	0.34773	0.34809	0.34792
0.5	0.37390	0.36845	0.37458	0.37129	0.37137	0.37175	0.37158
0.6	0.36194	0.35601	0.36189	0.35883	0.35884	0.35921	0.35905
0.7	0.31269	0.30728	0.31231	0.31020	0.30973	0.31004	0.30991
0.8	0.23030	0.22588	0.22955	0.22809	0.22769	0.22792	0.22782
0.9	0.12207	0.11966	0.12160	0.12085	0.12062	0.12074	0.12069

Table IV. Comparison the numerical solutions with exact solution at different times for Example 2, $m = 80(h = 0.0125), \varepsilon = 0.01$.

		Numerical solution			
x_j	t_n	ASC-N($l = 6$) $\tau = 0.01$	EFD [8] $\tau = 0.0001$	Exact-EFD [8] $\tau = 0.0001$	Exact solution
0.25	0.4	0.3420816	0.34244	0.34164	0.34191
	0.6	0.2690232	0.26905	0.26890	0.26896
	0.8	0.2214990	0.22145	0.22150	0.22148
	1.0	0.1881942	0.18813	0.18825	0.18819
	3.0	0.0751071	0.07509	0.07515	0.07511
0.5	0.4	0.6624275	0.67152	0.65606	0.66071
	0.6	0.5304316	0.53406	0.52658	0.52942
	0.8	0.4397358	0.44143	0.43743	0.43914
	1.0	0.3747873	0.37568	0.37336	0.37442
	3.0	0.1501911	0.15020	0.15015	0.15018
0.75	0.4	0.9140587	0.94675	0.90111	0.91026
	0.6	0.7705525	0.78474	0.75862	0.76724
	0.8	0.6496562	0.65659	0.64129	0.64740
	1.0	0.5575667	0.56135	0.55187	0.55605
	3.0	0.2248605	0.22502	0.22454	0.22481

In addition, when the ASC-N scheme is constructed, the point number using the C-N scheme in each segment maybe inhomogeneous providing Equations (9) and (10), (8) and (11) are used alternating at two adjacent points of the time level of even number and the level of odd number, respectively (see Figure 2 or 3). Therefore, the point number of each segment on same time level maybe inhomogeneous, for example, in Example 2, the point number of each segment on same time level is inhomogeneous for $m = 80$ or 200.

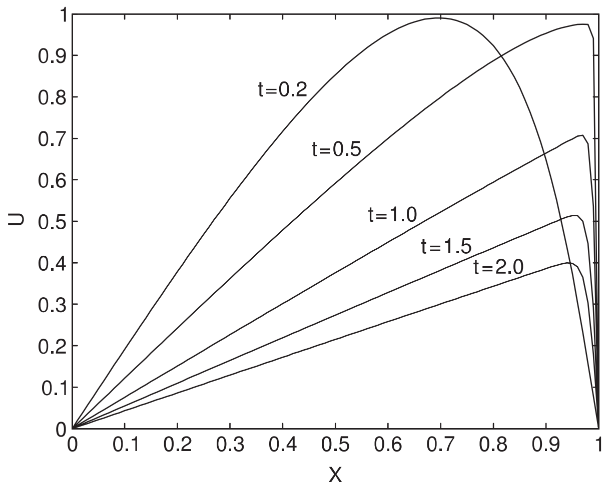

Figure 4. The ASC-N solutions of Example 2 for $\varepsilon = 0.005$, $m = 100$, $l = 11$, $\tau = 0.001$.

Table V. Convergence rate of the ASC-N solutions for Example 2 at $t = 0.4, h = 1/m, \varepsilon = 1.0$.

m	l	λ	e_h	$\frac{e_h}{h^2}$
50	7	2.5	4.7314E-05	1.1828E-01
100	11	2.5	6.9884E-06	6.9884E-02
256	17	2.62144	7.2224E-07	4.7333E-02
400	21	2.5	1.8329E-07	2.9326E-02

Example 3

We consider Equation (1) with the following boundary conditions:

$$u(0, t) = u(L, t) = 0$$

and the initial conditions at time $t = 1$ given by

$$u(x, 1) = \frac{x}{1 + \exp\left[\frac{1}{4\varepsilon}\left(x^2 - \frac{1}{4}\right)\right]}$$

This problem has the exact solution of the form [11]

$$u(x, t) = \frac{x/t}{1 + (t/t_0)^{1/2}\exp(x^2/4\varepsilon t)} \tag{18}$$

where $t_0 = \exp(1/8\varepsilon)$.

The numerical solutions of Example 3 are given in Table VI for $\varepsilon = 0.005$, $L = 8$ with $h = 0.05$ and $\tau = 0.0001$ at $t_n = 1.5$, 3.0 and 4.5. The agreement between our numerical

Table VI. Comparison of results of Example 3 at different times for $\varepsilon = 0.005$ and $[0,L] = [0,8]$ with $h = 0.05$ and $\Delta t = 0.0001$, $m = 160$, $l = 53$.

	$t = 1.5$			$t = 3.0$			$t = 4.5$		
x_j	ASC-N	B-SFEM [11]	Exact	ASC-N	B-SFEM [11]	Exact	ASC–N	B-SFEM [11]	Exact
0.5	0.15329	0.15398	0.15327	0.06427	0.06468	0.06426	0.03799	0.03825	0.03799
1.0	0.26581	0.26634	0.26577	0.11882	0.11942	0.11880	0.07188	0.07231	0.07187
1.5	0.30415	0.30451	0.30412	0.15511	0.15576	0.15509	0.09794	0.09847	0.09793
2.0	0.26141	0.26190	0.26142	0.16763	0.16832	0.16762	0.11339	0.11399	0.11339
2.5	0.17213	0.17268	0.17217	0.15629	0.15699	0.15630	0.11698	0.11761	0.11698
3.0	0.08804	0.08839	0.08807	0.12736	0.12803	0.12738	0.10948	0.11011	0.10949
3.5	0.03581	0.03594	0.03582	0.09130	0.09185	0.09134	0.09367	0.09425	0.09369
4.0	0.01186	0.01189	0.01186	0.05795	0.05834	0.05798	0.07359	0.07409	0.07361
4.5	0.00325	0.00325	0.00325	0.03283	0.03305	0.03284	0.05328	0.05367	0.05330
5.0	0.00074	0.00074	0.00074	0.01673	0.01684	0.01674	0.03570	0.03597	0.03572

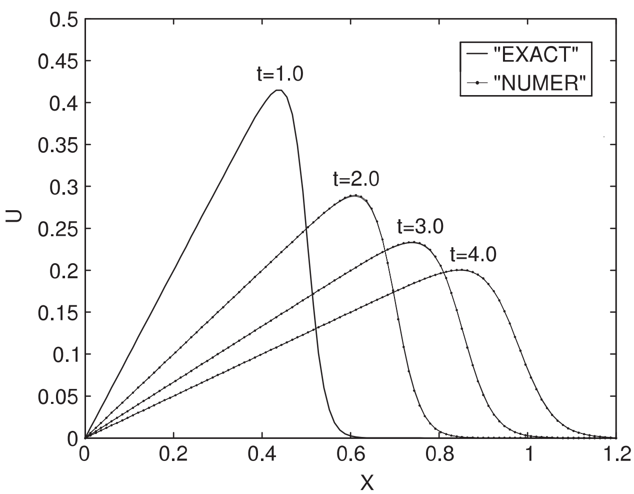

Figure 5. The ASC-N solutions of Example 3 for $\varepsilon = 0.005$, $m = 100$, $l = 11$, $\tau = 0.001$.

solutions and exact solution is satisfactorily good. Since both solutions hit each other after $x = 5.0$, they are not given in Table VI.

Figure 5 illustrates the numerical and exact solutions of Example 3 at different values of t for $\varepsilon = 0.005$ with $h = 0.012$ and $\tau = 0.05$, and both solutions are drawn on the same diagram. The expression 'NUMER' represent the numerical solution given by schemes (12), 'EXACT' represent the values of the exact solution. The graphic solutions in Figure 5 shows that the numerical solutions and the exact solution agree nicely.

ACKNOWLEDGEMENTS

This research is supported by the Natural Science Foundation of Shandong Province of China (Y2003A04).

REFERENCES

1. Burgers J. A mathematical model illustrating the theory of turbulence. *Advances in Applied Mechanics*. Academic Press: New York, 1948.
2. Arina R, Canuto C. A self-adaptive domain decomposition for the viscous inviscied coupling 1. Burgers' equation. *Journal of Computational Physics* 1993; **105**:290–300.
3. Kakuda K, Tosaka N. The generalized boundary element approach to Burgers' equation. *International Journal for Numerical Methods in Engineering* 1990; **29**:245–261.
4. Caldwell J, Wanless P, Cook AE. A finite element approach to Burgers' equation. *Applied Mathematical Modelling* 1981; **5**:189–193.
5. Hon YC, Mao XZ. An efficient numerical scheme for Burgers' equation. *Applied Mathematics and Computation* 1998; **95**:37–50.
6. Mittal RC, Sighhal P. Numerical solution of Burgers' equation. *Communications in Numerical Methods in Engineering* 1993; **9**:297–406.
7. Öziş T, Özdeş A. A direct variational methods applied to Burgers' equation. *Journal of Computational and Applied Mathematics* 1996; **71**:163–175.
8. Kutluay S, Bahadir AR, Özdeş A. Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods. *Jouranl of Computational and Applied Mathematics* 1999; **103**:251–261.
9. Evans DJ, Sahimi MS. The numerical solution of Burgers' equations by the alternating group explicit (AGE) method. *International Journal of Computer Mathematics* 1989; **29**:39–64.
10. Jin C, Liu J. An alternating group explicit iteration method for Burgers' equation. *Chinese Journal of Computational Physics* 1998; **5**:607–613 (in Chinese).
11. Kutluay S, Esen A, Dag I. Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method. *Journal of Computational and Applied Mathematics* 2004; **167**:21–33.
12. Wu Y, Wu X. Linearized and rational approximation method for solving non-linear Burgers' equation. *International Journal for Numerical Methods in Fluids* 2004; **45**:509–525.
13. Evans DJ, Abdullah ARB. Group explicit methods for parabolic equations. *International Journal of Computer Mathematics* 1983; **14**:73–105.
14. Zhang B, Su X. Alternating block explicit–implicit method for two-dimensional diffusion equation. *International Journal of Computer Mathematics* 1991; **38**:241–255.
15. Zhang B, Li W. On alternating segment Crank–Nicolson scheme. *Parallel Computing* 1994; **20**:897–902.
16. Chen J, Zhang B. A class of alternating block Crank–Nicolson method. *International Journal of Computer Mathematics* 1992; **45**:89–112.
17. Kellogg RB. An alternating direction method for operator equations. *Journal of the Society for Industrial and Applied Mathematics* 1964; **12**:848–854.
18. Cole JD. On quasi-linear parabolic equation occurring in aerodynamics. *Quarterly of Applied Mathematics* 1951; **9**:225–236.