A Fourth-order Singly Diagonally Implicit Runge-Kutta Method for Solving One-dimensional Burgers' Equation

Dingwen Deng*, and Tingting Pan

Abstract—**In this paper, a fourth-order methods of lines (MOL) based on Hopf-Cole transformation is derived for solving one-dimensional (1D) Burgers' equation by the combinations of singly diagonally implicit Runge-Kutta method (S-DIRKM), compact finite difference method and Simpson's rule. By matrix analysis method, it is proved that it is unconditionally stable. Numerical results show its efficiency.**

Index Terms—**Burgers' equation; Hopf-Cole transformation; Compact finite difference scheme; Runge-Kutta methods**

I. Introduction

THIS study mainly focuses on high order numerical approximation to 1D Burgers' equation with nonhomogeneous boundary conditions as follows

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = \gamma \dfrac{\partial^2 u}{\partial x^2}, \quad (x,t) \in [a,b] \times [0,T], \\ u(a,t) = \alpha(t), \quad u(b,t) = \beta(t), \quad t \in [0,T], \\ u(x,0) = \phi(x), \quad a \le x \le b, \end{cases} \tag{1}$$

where a, b and γ are all constants and γ is the coefficient of kinematic viscosity. Burger's equation (1) (cf. [1]) was extensively applied in many fields, such as turbulent fluid motion, gas dynamics, traffic flow and shock waves in a viscous medium.

Exact solution to Eq. (1) with a restricted of initial and boundary conditions (IBCs) is obtained by using some analysical method such as Hopf-Cole transformation, homotopy perturbation method, Adomian's decomposition method and differential transformation method. However, as for Eq. (1) with arbitrary IBCs, there exist no generally analytical methods. As a result, over the years, a great deal of attention has been paid on the developments of numerical algorithms for the solution of Burger's equation. With regards to these details, please refer to references [7]–[17].

Recently, compact finite difference methods (CFDMs) (cf. [8]–[12], [18]–[21]) have been developed to solve Eq. (1) in [8]–[12]. However, although these CFDMs have many merits in terms of spatial accuracy and implementation, initial value methods (IVMs) used in these algorithms have either low-order accuracy or conditional stability.

Manuscript received March 22, 2015; revised June 7, 2015. This work is partially supported by the National Natural Science Foundation of China (Grant Nos. 11401294, 11326046, 11261040), Youth Natural Science Foundation of Jiangxi Provincial Education Department (Grant No. GJJ14545), Youth Natural Science Foundation of Jiangxi province (Grant No. 20142BAB211003).

$*$ Dingwen Deng, Corresponding Author, College of Mathematics and Information Science, Nanchang Hangkong University, Nanchang 330063, China (Email: dengdingwen2010@163.com).

Tingting Pan, Public Department, Jiangxi Nursing Vocational and Technical College, Nanchang 330052, China

In this paper, we continue to study the CFDM of Eq. (1). Firstly, a fourth-order compact finite difference scheme at inner and boundary nodes is derived, thus forming a stiff IVPs. Secondly, a fourth-order A-stable singly diagonally implicit Runge-Kutta method (SDIRKM) (c.f. [21], [22]) is used to solve this IVPs. Finally, corresponding numerical solution is obtained by solving tri-diagonal equations, which are established by using Simpson's rule to Hopf-Cole transformation. Numerical results show our method is reasonable.

II. Fourth-order numerical methods

This section is devoted to the derivation of the new numerical method.

A. Notations and auxiliary Lemma

For convenience, denote temporal increment by Δt, and suppose that there exist two positive integers K, k such that $\Delta t = T/K$, $t_k = k\Delta t$, $0 \le k \le K$. Moreover, $h = (b-a)/n$ represents grid spacing. The spatial grid nodes $x_j = a + jh$, $j = 0, 1, \ldots, n$ form the following sets $\bar{\Omega}_h = \{x_j | 0 \le j \le n\}$. On $\bar{\Omega}_h$, we define grid function space $\mathbf{S}_h = \{\mathbf{w} | (w_0, w_1, \ldots, w_{n-1}, w_n)^T\}$ and introduce centered difference operator $\delta_x^2 w_j = (w_{j+1} - 2w_j + w_{j-1})/h^2$. In what follows, a lemma used later is firstly given.

Lemma 1 (cf. [18]) Letting $w(x) \in \mathcal{C}^5[a,b]$, then we have that

$$\begin{aligned} w^{'}(x_0) &= \frac{w(x_1) - w(x_0)}{h} - \frac{5h}{12} w^{''}(x_0) \\ &\quad - \frac{h}{12} w^{''}(x_1) - \frac{h^2}{12} w^{(3)}(x_0) + \mathcal{O}(h^4), \\ w^{'}(x_n) &= \frac{w(x_n) - w(x_{n-1})}{h} + \frac{5h}{12} w^{''}(x_n) \\ &\quad + \frac{h}{12} w^{''}(x_{n-1}) - \frac{h^2}{12} w^{(3)}(x_n) + \mathcal{O}(h^4). \end{aligned}$$

B. Establishment of numerical algorithms

The application of the Hopf-Cole transformation

$$u(x,t) = -2\gamma \frac{v_x(x,t)}{v(x,t)}, \tag{2}$$

to Eq. (1) yields an equivalent initial boundary value problem (IBVP) as follow

$$\begin{cases} \dfrac{\partial v}{\partial t} = \gamma \dfrac{\partial^2 v}{\partial x^2}, \qquad (\mathbf{x}, t) \in (a,b) \times [0,T]; \\ 2\gamma v_x(a,t) + \alpha(t) v(a,t) = 0, \\ 2\gamma v_x(b,t) + \beta(t) v(b,t) = 0, \qquad t \in [0,T]; \\ v(\mathbf{x}, 0) = \exp\Big\{ -\displaystyle\int_a^x \frac{\phi(s)}{2\gamma} ds \Big\}, \quad a \le x \le b. \end{cases} \tag{3}$$

For convenience, denote the approximations to $v(x_j,t)$ and $v(x_j,t_k)$ by $V_j(t)$ and V_j^k, respectively. Whereas the approximation to $u(x_j,t_k)$ is represented by U_j^k. Clearly, corresponding vectors $\mathbf{V}^k(t)$, $\mathbf{V}^k$ and $\mathbf{U}^k$ are all belong to $\mathbf{S}_h$.

For deriving a fourth-order semi-discrete scheme of Eq. (3), we first develop fourth-order boundary schemes. Clearly, it follows from Lemma 1 that

$$\begin{aligned}\frac{\partial v(x_0,t)}{\partial x} &= \frac{v(x_1,t)-v(x_0,t)}{h} - \frac{5h}{12}\frac{\partial^2 v(x_0,t)}{\partial x^2} \\ &\quad - \frac{h}{12}\frac{\partial^2 v(x_1,t)}{\partial x^2} - \frac{h^2}{12}\frac{\partial^3 v(x_0,t)}{\partial x^3} \\ &\quad + \mathcal{O}(h^4),\end{aligned} \tag{4}$$

$$\begin{aligned}\frac{\partial v(x_n,t)}{\partial x} &= \frac{v(x_n,t)-v(x_{n-1},t)}{h} \\ &\quad + \frac{5h}{12}\frac{\partial^2 v(x_n,t)}{\partial x^2} + \frac{h}{12}\frac{\partial^2 v(x_{n-1},t)}{\partial x^2} \\ &\quad - \frac{h^2}{12}\frac{\partial^3 v(x_n,t)}{\partial x^3} + \mathcal{O}(h^4),\end{aligned} \tag{5}$$

at boundary nodes (x_0,t) and (x_n,t), respectively. By Eq. (3), it is easy to find that

$$\begin{aligned}&v_{xx}(x_0,t) = \frac{1}{\gamma}v_t(x_0,t); \qquad v_{xx}(x_1,t) = \frac{1}{\gamma}v_t(x_1,t); \\ &v_{xxx}(x_0,t) = \frac{1}{\gamma}v_{tx}(x_0,t) \\ &\qquad = -\frac{1}{2\gamma^2}\Big[\alpha_t(t)v(x_0,t) + \alpha(t)v_t(x_0,t)\Big].\end{aligned} \tag{6}$$

The substitution of above Eqs. (6) into Eq. (4) yields

$$\begin{aligned}&\Big[\frac{5}{12} - \frac{\alpha(t)h}{24\gamma}\Big]v_t(x_0,t) + \frac{1}{12}v_t(x_1,t) \\ &= \Big[\frac{\alpha_t(t)h}{24\gamma} + \frac{\alpha(t)}{2h} - \frac{\gamma}{h^2}\Big]v(x_0,t) + \frac{\gamma}{h^2}v(x_1,t).\end{aligned} \tag{7}$$

Also, applying the same techniques as those used in the derivation of (7), we have that

$$\begin{aligned}&\Big[\frac{5}{12} + \frac{\beta(t)h}{24\gamma}\Big]v_t(x_n,t) + \frac{1}{12}v_t(x_{n-1},t) \\ &= \frac{\gamma}{h^2}v(x_{n-1},t) + \Big[-\frac{\beta_t(t)h}{24\gamma} \\ &\quad - \frac{\beta(t)}{2h} - \frac{\gamma}{h^2}\Big]v(x_n,t).\end{aligned} \tag{8}$$

Then, using fourth-order compact finite difference scheme to approximate second-order spatial derivative at interior nodes, we obtain that

$$\begin{aligned}&v_t(x_j,t) = \gamma\frac{\delta_x^2}{1+(h^2\delta_x^2)/12}v(x_j,t) + \mathcal{O}(h^4), \\ &j = 1,2,\ldots,n-1;\end{aligned} \tag{9}$$

which can be equivalently written as

$$\begin{aligned}&(1+\frac{h^2\delta_x^2}{12})v_t(x_j,t) = \gamma\delta_x^2 v(x_j,t) \\ &+\mathcal{O}(h^4), \qquad j=1,2,\ldots,n-1.\end{aligned} \tag{10}$$

Therefore, the combination of (7), (8) with (10) yields that a fourth-order semi-discretization scheme of IBVP (3) as follow

$$\begin{cases}\dfrac{d\mathbf{V}(t)}{dt} = L(\mathbf{V}(t),t), \quad 0 \le t \le T, \\ \mathbf{V}(0) = \mathbf{V}^0,\end{cases} \tag{11}$$

in which $L(\mathbf{V}(t),t) = (A^{-1}B)\mathbf{V}(t)$, and A and B are both tri-diagonal matrices of order $n+1$ as follows:

$$A = \begin{pmatrix} \frac{5}{12} - \frac{\alpha(t)h}{24\gamma} & \frac{1}{12} & & & \\ \frac{1}{12} & \frac{5}{6} & \frac{1}{12} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{12} & \frac{5}{6} & \frac{1}{12} \\ & & & \frac{1}{12} & \frac{5}{12} + \frac{\beta(t)h}{24\gamma} \end{pmatrix}_{(n+1)\times(n+1)},$$

$$B = \begin{pmatrix} \xi_1 & \frac{\gamma}{h^2} & & & \\ \frac{\gamma}{h^2} & -\frac{2\gamma}{h^2} & \frac{\gamma}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{\gamma}{h^2} & -\frac{2\gamma}{h^2} & \frac{\gamma}{h^2} \\ & & & \frac{\gamma}{h^2} & \xi_2 \end{pmatrix}_{(n+1)\times(n+1)},$$

where $\xi_1 = \frac{\alpha_t(t)h}{24\gamma} + \frac{\alpha(t)}{2h} - \frac{\gamma}{h^2}$ and $\xi_2 = -\frac{\gamma}{h^2} - \frac{\beta(t)}{2h} - \frac{\beta_t(t)h}{24\gamma}$. Here, meshsize h should be satisfied the following condition: $\max\limits_{t\in[0,T]}\{\alpha(t), -\beta(t)\}h < 8\gamma$, which makes matrix A strictly diagonally dominant, and thus ensuring its invertibility.

The subsequent work is to solve the initial value problem (IVP) (11) by using a A-stable initial value methods (IVM), which has no restriction for temporal grids. From the point of view of accuracy and time cost, a fourth-order A-stable SDIRKM is introduced as follows

$$\begin{cases}\mathbf{Y}^{(1)} = L\Big(\mathbf{V}^n + \rho\Delta t\mathbf{Y}^{(1)}, t_n + \rho\Delta t\Big), \\ \mathbf{Y}^{(2)} = L\Big(\mathbf{V}^n + (\frac{1}{2}-\rho)\Delta t\mathbf{Y}^{(1)} + \rho\Delta t\mathbf{Y}^{(2)}, \\ \qquad t_n + \dfrac{\Delta t}{2}\Big), \\ \mathbf{Y}^{(3)} = L\Big(\mathbf{V}^n + 2\rho\Delta t\mathbf{Y}^{(1)} + (1-4\rho)\Delta t\mathbf{Y}^{(2)} \\ \qquad +\rho\Delta t\mathbf{Y}^{(3)}, t_n + (1-\rho)\Delta t\Big), \\ \mathbf{V}^{k+1} = \mathbf{V}^k + \varrho\Delta t\mathbf{Y}^{(1)} + (1-2\varrho)\Delta t\mathbf{Y}^{(2)} \\ \qquad +\varrho\Delta t\mathbf{Y}^{(3)},\end{cases} \tag{12}$$

where $\rho = 1/2 + (\sqrt{3}/3)\cos(\pi/18)$ and $\varrho = [24(1/2-\rho)^2]^{-1}$. Numerical results confirm that this SDIRKM is effective.

In the end, we only need calculate $\mathbf{U}^{(k+1)}$ from $\mathbf{V}^{(k+1)}$. By integrating (2) with respect to variable x on the interval $[x_{j-1}, x_{j+1}]$ for $j = 1,2,\ldots,n-1$, we firstly find that

$$\begin{aligned}\int_{x_{j-1}}^{x_{j+1}} u(x,t_{k+1})dx &= -2\gamma\int_{x_{j-1}}^{x_{j+1}} \frac{v_x(x,t_{k+1})}{v(x,t_{k+1})}dx \\ &= -2\gamma\ln\Big|\frac{v(x_{j+1},t_{k+1})}{v(x_{j-1},t_{k+1})}\Big|.\end{aligned}$$

Secondly, the use of the Simpson's rule for the integration of above equation infers that

$$\begin{aligned}&u(x_{j-1},t_{k+1}) + 4u(x_j,t_{k+1}) + u(x_{j+1},t_{k+1}) \\ &= -\frac{6\gamma}{h}\ln\Big|\frac{v(x_{j+1},t_{k+1})}{v(x_{j-1},t_{k+1})}\Big| + \mathcal{O}(h^4).\end{aligned}$$

Finally, Omitting $\mathcal{O}(h^4)$ and replacing $v(x_j,t_{k+1})$ with V_j^{k+1} in above equation yields that

$$\begin{cases} 4U_1^{k+1}+U_2^{k+1}=F_1^{k+1}-u(x_0,t_{k+1}),\\ U_{j-1}^{k+1}+4U_j^{k+1}+U_{j+1}^{k+1}=F_j^{k+1},\\ (j=2,\dots,n-2),\\ U_{n-2}^{k+1}+4U_{n-1}^{k+1}=F_{n-1}^{k+1}-u(x_n,t_{k+1}), \end{cases} \tag{13}$$

in which $u(x_0,t_{k+1})=\alpha(t_{k+1})$, $u(x_n,t_{k+1})=\beta(t_{k+1})$ and

$$F_j^{k+1}=-\frac{6\gamma}{h}\ln\Big|\frac{V_{j+1}^{k+1}}{V_{j-1}^{k+1}}\Big|, \qquad j=1,2,\dots,n-1.$$

As coefficient matrix associated with algebraic equations (13) is strictly diagonally dominant, this algebraic equations (13) is uniquely solvable and solved using Thomas algorithm.

III. Stability analysis

This section is suggested to the stability analysis of the new solver. firstly, By simple inference, the following Corollary holds.

Corollary As $\alpha(t)=\beta(t)=0$, matrices A and B are both tridiagonal and symmetric matrices. Also, they are both nonsingular.

Let $\alpha(t)=\beta(t)=0$ and $C=A^{-1}B$. Then, by some computations, it is easy to find that the stable function of the SDIRKM (12), which is applied to solve IVP (11), is defined by $V^{k+1}=R(\Delta tC)V^k$, where $R(z)=Q(x)/P(z)$, $Q(z)=1+(1-3\rho)z+(1/2-3\rho+3\rho^2)z^2+[1/6-(3/2)\rho+3\rho^2-\rho^3]z^3$ and $P(z)=(1-\rho z)^3$.

Lemma 1 [21] As $z\le 0$, it holds that $|R(z)|\le 1$.

Lemma 2 Let $\alpha(t)=\beta(t)=0$. Then the eigenvalues of matrix $A^{-1}B$ are all real and non-positive.

Proof. Let λ be an arbitrary eigenvalue of $A^{-1}B$, and $\mathbf{x}\in\mathbf{R^{n+1}}$ be corresponding eigenvector. Writing $\mathbf{x}=(x_1,x_2,\dots,x_{n+1})^T$, then we have that $(A^{-1}B)\mathbf{x}=\lambda\mathbf{x}$, which further implies that $\lambda\mathbf{x}^TA\mathbf{x}=\mathbf{x}^TB\mathbf{x}$. By simple computation, we easily find that $\mathbf{x}^TB\mathbf{x}=-\frac{\gamma}{h^2}\sum_{j=1}^{n}(x_{j-1}-x_j)^2\le 0$ and $\mathbf{x}^TA\mathbf{x}=\frac{1}{3}x_1^2+\frac{2}{3}\sum_{j=2}^{n-1}x_j^2+\frac{1}{3}x_n^2+\frac{1}{12}\sum_{j=1}^{n-1}(x_j+x_{j+1})^2\ge 0$, which imply that λ must be less than zero to make $\lambda\mathbf{x}^TA\mathbf{x}=\mathbf{x}^TB\mathbf{x}$ hold.

Therefore, it follows from Lemma 1 and Lemma 2 that the following Theorem 1 holds.

Theorem 1 New algorithm is unconditionally stable.

IV. Numerical examples

Here, for comparison, another two Total Variation Diminishing Runge-Kutta methods (TVDRKMs) (see [23]) is introduced to replace fourth-order SDIRKM (2.11). The first one is the following third-order TVDRKM :

$$\begin{cases} \mathbf{Y}^{(1)}=\mathbf{V}^k+\Delta tL\Big(\mathbf{V}^n,t_n\Big),\\ \mathbf{Y}^{(2)}=\frac{3}{4}\mathbf{V}^k+\frac{1}{4}\mathbf{Y}^{(1)}+\frac{\Delta t}{4}L\Big(\mathbf{Y}^{(1)},t_n+\Delta t\Big),\\ \mathbf{V}^{k+1}=\frac{1}{3}\mathbf{V}^k+\frac{2}{3}\mathbf{Y}^{(2)}\\ \qquad +\frac{2\Delta t}{3}L\Big(\mathbf{Y}^{(2)},t_n+\frac{\Delta t}{2}\Big). \end{cases} \tag{14}$$

The second one is fourth-order TVD RKM as follow

$$\begin{cases} \mathbf{Y}^{(1)}=\mathbf{V}^k+A_1\Delta tL\Big(\mathbf{V}^n,t_n\Big),\\ \mathbf{Y}^{(2)}=B_1\mathbf{V}^k+B_2\mathbf{Y}^{(1)}+B_3\Delta tL\Big(\mathbf{Y}^{(1)},t_n+B_4\Delta t\Big),\\ \mathbf{Y}^{(3)}=C_1\mathbf{V}^k+C_2\mathbf{Y}^{(2)}+C_3\Delta tL\Big(\mathbf{Y}^{(2)},t_n+C_4\Delta t\Big),\\ \mathbf{Y}^{(4)}=D_1\mathbf{V}^k+D_2\mathbf{Y}^{(3)}+D_3\Delta tL\Big(\mathbf{Y}^{(3)},t_n+D_4\Delta t\Big),\\ \mathbf{V}^{k+1}=E_1\mathbf{V}^k+E_2\mathbf{Y}^{(2)}+E_3\mathbf{Y}^{(3)}+E_4\Delta tL\Big(\mathbf{Y}^{(3)},\\ t_n+E_5\Delta t\Big)+E_6\mathbf{Y}^{(4)}+E_7\Delta tL(\mathbf{Y}^{(4)},t_n+E_8\Delta t), \end{cases} \tag{15}$$

where $A_1=0.39175222700392, B_1=0.44437049406734, B_2=0.55562950593266, B_3=0.36841059262959, B_4=0.39175222700392, C_1=0.62010185138540, C_2=0.37989814861460, C_3=0.25189177424738, C_4=0.58607968896780, D_1=0.17807995410773, D_2=0.82192004589227, D_3=0.54497475021237, D_4=0.47454236302687, E_1=0.00683325884039, E_2=0.51723167208978, E_3=0.12759831133288, E_4=0.08460416338212, E_5=0.47454236302687, E_6=0.34833675773694, E_7=0.22600748319395, E_8=0.93501063100924$.

In what follows, for the sake of clearness and convenience, it is necessary to conclude our algorithms as follows. Assume that $\mathbf{V}^k$ is known.

Algorithm 1: First solving $\mathbf{V}^{k+1}$ using third-order TVDRKM (14), then we obtain $\mathbf{U}^{k+1}$ by the use of the Thomas algorithm to (13).

Algorithm 2: $\mathbf{V}^{k+1}$ is solved using fourth-order TVDRKM (15). Then, we obtain $\mathbf{U}^{k+1}$ by applying the Thomas algorithm to (13).

Algorithm 3: Firstly compute $\mathbf{V}^{k+1}$ by applying SDIRKM (12), then calculate $\mathbf{U}^{k+1}$ by the application of the Thomas algorithm to (13).

In this section, three IBVPs are carried out to testify the utility and adaptability of our algorithms. L^2- and L^∞-norm errors at $t_N=N\Delta t$ between exact and calculated solutions are defined as follows $\|e^N\|=\Big[\sum_{j=1}^{n-1}(U_j^N-u_j^N)^2h\Big]^{\frac{1}{2}}$, $\|e^N\|_\infty=\max_{1\le j\le n-1}(U_j^N-u_j^N)$, respectively. They and the total elapsed time (CPU) in seconds delivered in each case are applied to measure the performance of the numerical algorithms. Moreover, convergence rates in L^∞- and L^2-norms are also defined as follows: rate=$[\log_2\|e^N(2h)\|_\infty/\|e^N(h)\|_\infty]$ and rate2=$[\log_2\|e^N(2h)\|/\|e^N(h)\|]$, respectively, as $\Delta t=h$, (see Table I).

Example 1 To test the accuracy and efficiency of our algorithms, we consider Burgers' equation (1) with initial and boundary conditions [12]

$$\begin{aligned} u(x,0)&=2\gamma\frac{\pi\sin(\pi x)}{\sigma+\cos(\pi x)}, && x\in(0,1);\\ u(0,t)&=u(1,t)=0, && t\in(0,T]; \end{aligned}$$

where $\sigma>1$ is a parameter. By Hopf-Cole transformation, it is not difficult to find that

$$v(x,0)=\frac{\sigma+\cos(\pi x)}{\sigma+1}.$$

And the exact solution to this problem is given as follow

$$u(x,t) = \frac{2\gamma\pi e^{-\pi^2\gamma t}\sin(\pi x)}{\sigma + e^{-\pi^2\gamma t}\cos(\pi x)}, \qquad 0 < x < 1.$$

For this problem, we take $\sigma = 3$ and $\gamma = 0.1$, and display the numerical results in Table I-Table II. To accurately assess the performance of Algorithm 1, for a fixed grid, we run 200 times, then take the average computational time as time cost (CPU time) in Tables I. From Tables I, we can see that Algorithm 3 has a convergence rate of $\mathcal{O}(h^4)$ in L^∞ and L^2-norms as $\Delta t = h$, and costs reasonable computational time.

Table II shows superiority of Algorithm 3 over Algorithm 1 and Algorithm 2 in terms of stability. For example, all of solutions, which are obtained using Algorithm 1 with $\Delta t = 0.1h$ and $h \leq 1/80$ and using Algorithm 2 with $\Delta t = 0.1h$ and $h = 1/160$, respectively, fail to converge, while for arbitrary given h, solution provided by Algorithm 3 with $\Delta t = h$ is convergent.

Example 2 For comparison with other existing numerical methods, numerical approximation to the following Burgers' equation (1) with initial and boundary conditions [12]

$$\begin{aligned} &u(x,0) = \sin(\pi x), \qquad x \in (0,1);\\ &u(0,t) = u(1,t) = 0, \quad t \in (0,T]. \end{aligned}$$

is further considered. The exact solution to this problem is

$$u(x,t) = 2\pi\gamma \frac{\sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\gamma t) n \sin(n\pi x)}{c_0 + \sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\gamma t)\cos(n\pi x)}, \qquad (16)$$

where the coefficients are defined by

$$\begin{aligned} c_0 &= \int_0^1 \exp(-\frac{1-\cos(\pi x)}{2\pi\gamma})dx,\\ c_n &= 2\int_0^1 \exp(-\frac{1-\cos(\pi x)}{2\pi\gamma})\cos(n\pi x)dx, \ (n = 1,2,3,\ldots). \end{aligned}$$

According to Hopf-Cole transformation, we can also find that

$$v(x,0) = \exp(-\frac{1-\cos(\pi x)}{2\pi\gamma}), \qquad x \in [0,1].$$

For calculating errors of numerical solution, exact Fourier series solution (16) should be accurately evaluated. Here, a number N is taken such that $c_N \leq 1.0e-15$.

In Table III, as compact difference method proposed in [11] is second-order accurate in time and fourth-order accurate in space, grid relation $\Delta t = h^2$ is taken to make Xie's method convergent with an order of $\mathcal{O}(h^4)$. Also, grid relation $\Delta t = 0.05h$ for Algorithm 2 is chosen to make sure stability of the Algorithm. From this table, we can see that with the same meshsize h, solutions obtained using Xie's method, Algorithm 2 and Algorithm 3, respectively, have the almost same accuracy, however, the cost of Algorithm 3 is the lowest.

Table IV shows that Algorithm 3 is the most robust in comparison with other methods developed in [7], [11], [13]. Evolution graphs of numerical solution for several parameter γ are depicted in Figure 1, from which we can observe the asymptotic behavior of solution to this problem.

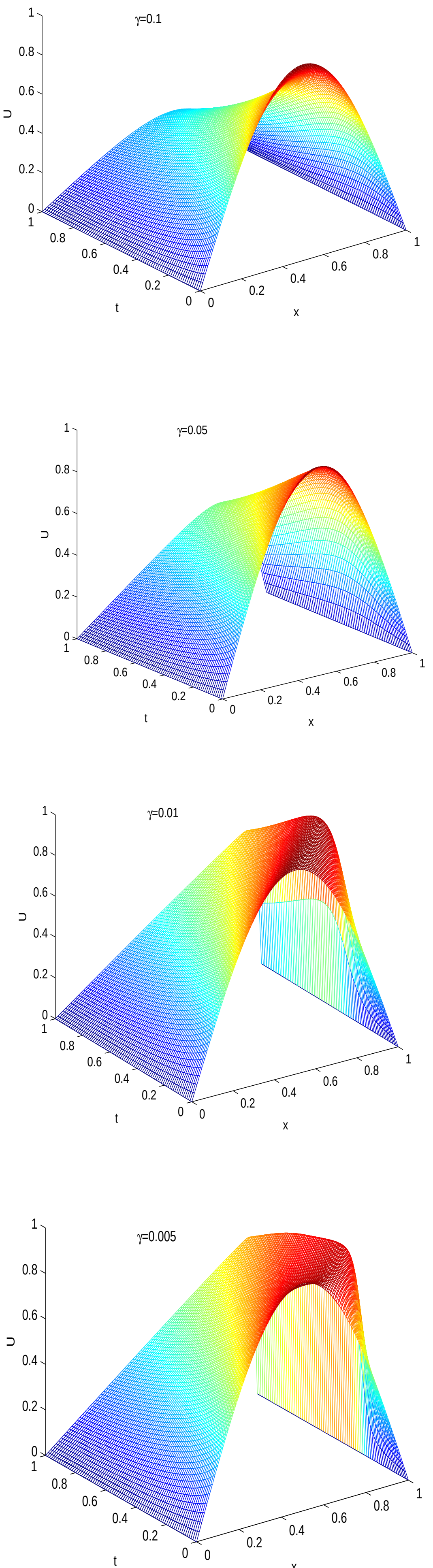


Fig. 1. Example 2 with different parameter γ (solved by Algorithm 3 with $\Delta t = h = 0.0125$): Time evolution graphs of numerical solution at $t = 1$.

TABLE I
COMPUTATIONAL RESULTS AT $t = 1$ FOR EXAMPLE 1, OBTAINED USING ALGORITHM 3 WITH $\Delta t = h$.

h	$1/4$	$1/8$	$1/16$	$1/32$	$1/64$	$1/128$
$\|e^N\|_\infty$	$3.2721e-04$	$1.6457e-05$	$9.88014e-07$	$6.2644e-08$	$3.9211e-09$	$2.4559e-10$
$rate$	$*$	4.3134	4.0581	3.9792	3.9978	3.9970
$\|e^N\|$	$1.6968e-04$	$8.6036e-06$	$5.1540e-07$	$3.2084e-08$	$2.0096e-09$	$1.2591e-10$
$rate2$	$*$	4.3017	4.0612	4.0058	3.9969	3.9964
CPU	$7.50e-05$	$1.60e-04$	$2.35e-04$	$5.50e-04$	$2.19e-03$	$1.31e-02$

TABLE II
COMPARISON OF NUMERICAL RESULTS FOR EXAMPLE 1 OBTAINED USING THREE ALGORITHMS

	Algorithm 1		Algorithm 2		Algorithm 3	
	$\Delta t = 0.1h$		$\Delta t = 0.1h$		$\Delta t = h$	
h	$\|e^N\|_\infty$	$\|e^N\|$	$\|e^N\|_\infty$	$\|e^N\|$	$\|e^N\|_\infty$	$\|e^N\|$
$1/5$	$1.1759e-04$	$5.9583e-05$	$1.1757e-04$	$5.9576e-05$	$1.2615e-04$	$6.3046e-05$
$1/10$	$6.0544e-06$	$3.1434e-06$	$6.0521e-06$	$3.1426e-06$	$6.7294e-06$	$3.4454e-06$
$1/20$	$3.6030e-07$	$1.8834e-07$	$3.6001e-07$	$1.8823e-07$	$4.0790e-07$	$2.1049e-07$
$1/40$	$2.2261e-08$	$1.1654e-08$	$2.2221e-08$	$1.1639e-08$	$2.5660e-08$	$1.3149e-08$
$1/80$	NaN	NaN	$1.3836e-09$	$7.2383e-10$	$1.6070e-09$	$8.2375e-10$
$1/160$	NaN	NaN	NaN	NaN	$1.0062e-10$	$5.1584e-11$

TABLE III
COMPARISON OF NUMERICAL RESULTS AT $t = 1$ FOR EXAMPLE 2 WITH $\gamma = 0.05$.

	h	$1/20$	$1/40$	$1/80$	$1/160$	$1/320$
HOCM [11]	$\|e^N\|_\infty$	$2.5171e-04$	$1.3765e-05$	$8.6955e-07$	$5.3973e-08$	$3.3676e-09$
$\Delta t = h^2$	$\|e^N\|$	$6.8583e-05$	$3.9127e-06$	$2.3927e-07$	$1.4874e-08$	$9.2840e-10$
	CPU	$9.35e-04$	$5.08e-03$	$3.10e-02$	0.50	16.89
Algorithm 2	$\|e^N\|_\infty$	$2.5219e-04$	$1.3794e-05$	$8.7175e-07$	$5.4086e-08$	$3.3504e-09$
$\Delta t = 0.05h$	$\|e^N\|$	$6.8773e-05$	$3.9254e-06$	$2.4007e-07$	$1.4916e-08$	$9.2336e-10$
	CPU	$5.31e-03$	$1.60e-02$	$9.30e-02$	0.75	18.07
Algorithm 3	$\|e^N\|_\infty$	$2.6218e-04$	$1.4836e-05$	$9.4213e-07$	$5.8831e-08$	$3.6858e-09$
$\Delta t = h$	$\|e^N\|$	$7.2655e-05$	$4.2683e-06$	$2.6529e-07$	$1.6629e-08$	$1.0423e-09$
	CPU	$3.15e-04$	$8.60e-04$	$3.75e-03$	$2.40e-02$	0.187

TABLE IV
COMPARISON OF NUMERICAL RESULTS AT $t = 1$ FOR EXAMPLE 2 WITH $\gamma = 0.05$, $\Delta t = h = 0.01$.

	t	1	1.5	2	2.5	3
HOCM [11]	$\|e^N\|_\infty$	$1.4097e-05$	$6.5103e-06$	$2.1697e-06$	$7.5075e-07$	$4.7140e-07$
	$\|e^N\|$	$7.2033e-06$	$3.1703e-06$	$1.2009e-06$	$4.9196e-07$	$2.8844e-07$
	CPU	$2.66e-03$	$3.05e-03$	$3.52e-03$	$3.91e-03$	$8.75e-03$
FDM [7]	$\|e^N\|_\infty$	$7.2819e-05$	$3.6426e-05$	$1.7624e-05$	$9.0853e-06$	$4.8601e-06$
	$\|e^N\|$	$3.1252e-05$	$2.0219e-05$	$1.1393e-05$	$6.2535e-06$	$3.2578e-06$
	CPU	$9.40e-04$	$1.41e-03$	$1.72e-03$	$2.26e-03$	$2.65e-03$
FEM [13]	$\|e^N\|_\infty$	$2.5646e-04$	$7.4400e-05$	$5.3496e-05$	$4.3241e-05$	$3.6328e-05$
	$\|e^N\|$	$1.2426e-04$	$4.7442e-05$	$3.6311e-05$	$3.0815e-05$	$2.5678e-05$
	CPU	$9.35e-04$	$1.41e-03$	$1.80e-03$	$2.27e-03$	$2.58e-03$
Algorithm 3	$\|e^N\|_\infty$	$3.8401e-07$	$9.6245e-08$	$3.0356e-08$	$1.1462e-08$	$4.8477e-09$
	$\|e^N\|$	$1.0874e-07$	$3.0993e-08$	$1.0945e-08$	$4.5609e-09$	$2.1156e-09$
	CPU	$6.64e-03$	$8.99e-03$	$1.13e-02$	$1.38e-02$	$1.62e-02$

Example 3 In this example, we solve IBVP (1) with initial-boundary conditions $u(x,0) = \{[\alpha + \mu + (\mu - \alpha)\exp[\alpha(x-\beta)/\gamma]\}/\{1+\exp[\alpha(x-\beta)/\gamma]\}$, $u(0,t) = 1$ and $u(1,t) = 0.2$, whose exact solution, i.e. traveling wave is

$$u(x,t) = \frac{\alpha + \mu + (\mu - \alpha)\exp(\eta)}{1+\exp(\eta)}$$

where $\eta = \alpha(x - \mu t - \beta)/\gamma$, α, β and μ are all constants. Similar to above problems, it is easy to find that

$$v(x,0) = \exp[\frac{(\alpha-\mu)(x-a)}{2\gamma}]\frac{1+\exp\{[\alpha(\beta-x)]/\gamma\}}{1+\exp\{[\alpha(\beta-a)]/\gamma\}}.$$

As literature [11], we take parameters $\alpha = 0.4$, $\mu = 0.6$, and $\beta = 0.125$. Numerical results are listed in Tables V–VII and Figure 2. From these data, we can deduce that Algorithm 3 outperforms Xie' [11] and Liao' [21] methods and Algorithm 2 in aspects of the accuracy and time cost. Also, Figure 2 shows that Algorithm 3 has a good capacity of simulation for a smaller γ.

V. CONCLUSIONS

In this article, a fourth-order numerical algorithm has been developed for solving 1D Burgers' equation. From numerical and theoretical results show that there is no grid restriction for this new method. Also numerical results show the superiority of our algorithm over Algorithm 1 (i.e. third-order TVDRKM), Algorithm 2 (i.e. fourth-order TVDRKM) and previous numerical methods devised in [7], [11]–[13] in terms of accuracy and computational cost.

TABLE V
COMPARISON BETWEEN EXACT AND NUMERICAL SOLUTIONS OF EXAMPLE 3 WITH $\gamma = 0.005$ AT $t = 1$, ($h = 0.002$).

	$\Delta t = 0.05h$	$\Delta t = 0.5h$			
x	Algorithm 2	Algorithm 3	Xie [11]	Liao [12]	exact solution
0.7	0.90464646058042	0.90465142905401	0.88207964747440	0.90551049986242	0.90463766238230
0.72	0.67895413930985	0.67896550910158	0.64123294440869	0.68094989423482	0.67895012808995
0.74	0.38517923470717	0.38518765326492	0.36184587622641	0.38666396133177	0.38518017320078
0.8	0.20197796098899	0.20197807770597	0.20173594132308	0.20199860991807	0.20197809852531
$\Vert e^N\Vert_\infty$	$1.1218e-05$	$1.5699e-05$	$3.7834e-02$	$2.0790e-03$	$*$
$\Vert e^N\Vert$	$7.4966e-06$	$8.0346e-06$	$9.0417e-03$	$3.8013e-04$	$*$
CPU	37.125	3.828	0.328	0.329	$*$

TABLE VI
COMPARISON OF NUMERICAL RESULTS AT $t = 1$ FOR EXAMPLE 3 WITH $\gamma = 0.0025$, $h = 0.002$.

	t	0.2	0.4	0.6	0.8	1.0
Xie [11]	$\Vert e^N\Vert_\infty$	$3.1587e-02$	$4.6043e-02$	$1.1548e-01$	$1.8655e-01$	$2.5604e-01$
$\Delta t = 0.5h$	$\Vert e^N\Vert$	$1.0622e-02$	$1.5992e-02$	$2.3254e-02$	$3.1745e-02$	$4.0611e-02$
	CPU	0.266	0.344	0.453	0.531	0.625
Liao [12]	$\Vert e^N\Vert_\infty$	$2.8439e-03$	$6.0177e-03$	$9.3536e-03$	$1.2706e-02$	$1.6062e-02$
$\Delta t = 0.5h$	$\Vert e^N\Vert$	$3.6584e-04$	$7.8105e-04$	$1.2138e-03$	$1.6477e-03$	$2.0817e-03$
	CPU	0.266	0.359	0.455	0.546	0.641
Algorithm 2	$\Vert e^N\Vert_\infty$	$4.0883e-04$	$5.5939e-04$	$5.5167e-04$	$5.2987e-04$	$5.0739e-04$
$\Delta t = 0.1h$	$\Vert e^N\Vert$	$7.9345e-05$	$1.0674e-04$	$1.1683e-04$	$1.2501e-04$	$1.3269e-04$
	CPU	19.172	38.125	57.016	76.047	94.891
Algorithm 3	$\Vert e^N\Vert_\infty$	$3.2135e-04$	$3.8435e-04$	$2.8963e-04$	$1.8592e-04$	$1.7705e-04$
$\Delta t = 0.5h$	$\Vert e^N\Vert$	$7.2418e-05$	$9.3995e-05$	$1.0152e-04$	$1.0930e-04$	$1.1826e-04$
	CPU	0.937	1.688	2.469	3.035	3.422

TABLE VII
COMPARISON OF NUMERICAL RESULTS AT $t = 1$ FOR EXAMPLE 3 WITH DIFFERENT γ, $h = 0.001$.

	γ	0.009	0.007	0.005	0.003	0.001
Xie [11]	$\Vert e^N\Vert_\infty$	$6.6602e-04$	$6.6602e-04$	$9.6423e-03$	$4.2083e-02$	$6.6619e-01$
$\Delta t = 0.5h$	$\Vert e^N\Vert$	$4.5370e-04$	$1.0534e-03$	$2.2967e-03$	$7.2300e-03$	$7.5986e-02$
	CPU	6.625	7.093	8.640	9.750	9.797
Liao [12]	$\Vert e^N\Vert_\infty$	$1.3640e-03$	$4.5181e-04$	$5.3489e-04$	$2.3978e-03$	$6.1255e-02$
$\Delta t = 0.5h$	$\Vert e^N\Vert$	$3.3473e-04$	$3.3473e-04$	$9.7659e-05$	$9.7659e-05$	$5.0355e-03$
	CPU	6.547	7.078	8.703	9.859	9.953
Algorithm 2	$\Vert e^N\Vert_\infty$	$1.2747e-03$	$2.6207e-04$	$1.4322e-05$	$1.2007e-05$	$3.4259e-03$
$\Delta t = 0.04h$	$\Vert e^N\Vert$	$3.1287e-04$	$5.6679e-05$	$2.6823e-06$	$3.9175e-06$	$3.9576e-04$
	CPU	1914.844	1919.734	1936.625	1953.657	1943.781
Algorithm 3	$\Vert e^N\Vert_\infty$	$1.2748e-03$	$2.6220e-04$	$1.5011e-05$	$5.4175e-06$	$5.3895e-04$
$\Delta t = 0.5h$	$\Vert e^N\Vert$	$3.1288e-04$	$5.6706e-05$	$2.8062e-06$	$3.6184e-06$	$2.8890e-04$
	CPU	34.375	34.375	34.844	38.390	38.390

REFERENCES

[1] J.M. Burgers, "A mathematical model illustrating the theory of turbulence," Advances in Applied Mechanics, vol. I, Academic Press, New York, pp. 171–199, 1948.

[2] J.D. Cole, "On a quasi-linear parabolic equations occuring in aerodynamics," Quarterly of Applied Mathematics, vol. 9, pp. 225–236, 1951.

[3] E. Hopf, "The partial differential equqation $u_t + uu_x = \mu u_{xx}$," Communications on Pure and Applied Mathematics, vol. 3, pp. 201–230, 1950.

[4] E. Benton, G.W. Platzman, "A table of solutions of the one-dimensional Burgers equations," Quarterly of Applied Mathematics, vol. 30, pp. 195–212, 1972.

[5] A. Kelleci and A. Yıldırım, "An efficient numerical method for solving coupled Burgers' equation by combining homotopy perturbation and Pade techniques," Numerical Methods for Partial Differential Equations, vol. 27, pp. 980–995, 2001.

[6] N. Taghizadeh, M. Akbari, and A. Ghelichzadeh, "Exact solution of Burgers equations by homotopy perturbation method and reduced differential transformation method," Australian Journal of Basic and Applied Sciences, vol. 5, pp. 580–589, 2011.

[7] Mohan. K. Kadalbajoo and A. Awasthi, "A numerical method based on Crank-Nicolson scheme for Burgers equation," Applied Mathematics and Computation, vol. 182, 1430–1442, 2006.

[8] I.A. Hassanien, A.A. Salama, and H.A. Hosham, "Fourth-order finite difference method for solving Burgers' equation," Applied Mathematics and Computation, vol. 170, pp. 781–800, 2005.

[9] M. Sari and G. G$\ddot{u}$rarslan, "A sixth-order compact finite difference scheme to the numerical solutions of Burgers equation," Applied Mathematics and Computation, vol. 208, pp. 475–483, 2009.

[10] P. Zhang and J. Wang, "A predictor-corrector compact finite difference scheme for Burger' equation," Applied Mathematics and Computation, vol. 219, pp. 892–898, 2012.

[11] S. Xie, G. Li, and S. Heo, "A compact finite difference method for solving Burgers' equation," International Journal for Numerical Methods in Fluids, vol. 62, pp. 747–764, 2010.

[12] W. Liao and J. Zhu, "An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation," Applied Mathematics and Computation, vol. 206 pp. 755-764, 2008.

[13] T. $\ddot{O}$zis, E.N. Aksan, and A. $\ddot{O}$zdes, "A finite element approach for solution of Burgers' equation," Applied Mathematics and Computation, vol. 139, pp. 417–428, 2003.

[14] S. Kutluay, A. Esen, and I. Dagb, "Numerical solutions of the Burgers equation by the least-squares quadratic B-spline finite element method," Journal of Computational and Applied Mathematics, vol. 167, pp. 21–33, 2004.

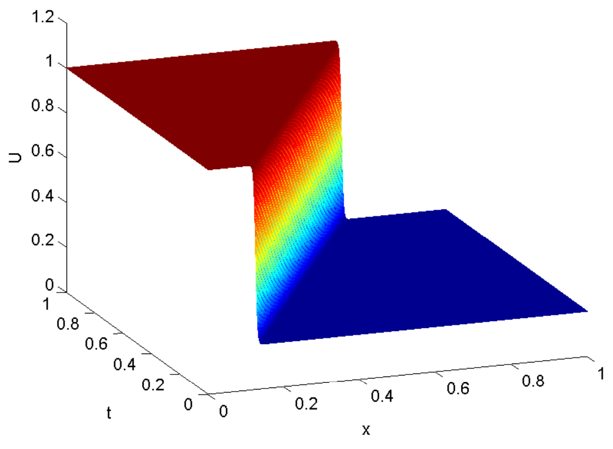
[15] R. Zhang, X. Yu, and G. Zhao, "Local discontinuous Galerkin method for solving Burgers and coupled Burgers equations," Chinese Physics B, vol. 20, Arthcle ID 110205, 6 pages, 2011.

[16] A. Hashemian, H.M. Shodja, "A meshless approach for solution of Burgers equation," Journal of Computational and Applied Mathematics, 220, pp. 226–239, 2008.

[17] R. Mokhtari, A.S. Toodar, and N.G. Chegini, "Application of the generalized differential quadrature method in solving Burgers'equations," Communications in Theoretical Physics, vol. 56, pp. 1009–1015, 2011.

[18] H. Cao, L. Liu, Y. Zhang, and S. Fu, "A fourth-order method of the convection-diffusion equations with Neumann boundary conditions," Applied Mathematics and Computation, vol. 217, pp. 9133–9141, 2011.

[19] B. Wongsaijai, K. Poochinapan, and T. Disyadej, "A compact finite



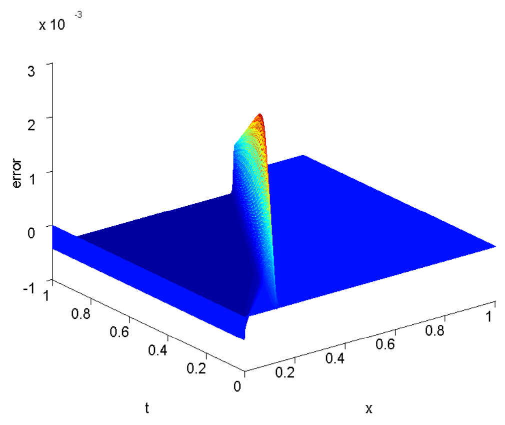


Fig. 2. Example 3 with $\gamma = 0.001$ (solved by Algorithm 3 with $\Delta t = 0.0005, h = 0.001$): Numerical solution and corresponding errors

difference method for solving the general Rosenau-RLW Equation," IAENG International Journal of Applied Mathematics, vol. 44, no. 4, pp. 192-199, 2014.

[20] J.C. Chen and W. Chen, "Two-dimensional nonlinear wave dynamics in blasius boundary layer flow using combined compact difference methods," IAENG International Journal of Applied Mathematics, vol. 41, no. 2, pp. 162-171, 2011.

[21] W. Liao and Y. Yan, "Singly diagonally implicit Runge-Kutta method for time-dependent reaction-diffusion equation," Numerical Methods for Partial Differential Equations," vol. 27, pp. 1423–1441, 2011.

[22] N. Senu, M. Suleiman, F. Ismail, and M. Othman, "A singly diagonally implicit Runge-Kutta-Nyström method for solving oscillatory problem-s," IAENG International Journal of Applied Mathematics, vol. 41, no. 2, pp. 155-161, 2008.

[23] C.W. Shu, "A survey of strong stability preserving high order time discretizations methods, In: Collected Lectures on the preservation of Stability under Discretization," SIAM, Philadelphia, 2002.