INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **45**:509–525 (DOI: 10.1002/fld.714)

Linearized and rational approximation method for solving non-linear Burgers' equation

Yun Wu and Xionghua Wu*,†

Department of Applied Mathematics, Tongji University, Shanghai 200092, People's Republic of China

SUMMARY

A new numerical method called linearized and rational approximation method is presented to solve non-linear evolution equations. The utility of the method is demonstrated for the case of differentiation of functions involving steep gradients. The solution of Burgers' equation is presented to illustrate the effectiveness of the technique for the solution of non-linear evolution equations exhibiting nearly discontinuous solutions. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: linearized and rational approximation method; Burgers' equation; A-stable; differential quadrature method; Sinc functions; *Padé* approximation

1. INTRODUCTION

The differential quadrature method (DQM) can be considered as the 'direct approach' of the traditional collocation (pseudo-spectral) methods in that the governing equations are analogized in terms of practical physical variables instead of usually fictitious expansion (spectral) coefficients [1, 2]. The advantages of the DQM over the latter lie in the ease of its implementation and more flexibility to choose grid points. The DQM has been extensively employed to approximate spatial partial derivative. The method can yield highly accurate solutions to the boundary value problems with a minimal computing effort, namely, the so-called spectral accuracy [1].

As known [3], the application of DQM requires the interpolation in space. Originally, this interpolation was obtained by using Lagrange polynomials. More recently, the use of Sinc functions [4, 5] was proposed in References [6–9] to solve initial boundary value problems for non-linear evolution equations and it was applied to the solution of non-linear wave equations [6] and to non-linear convection diffusion equations [7]. The advantages of the use of Sinc

*Correspondence to: Xionghua Wu, Department of Applied Mathematics, Tongji University, Shanghai 200092, People's Republic of China.
†E-mail: wuxhlu@sh163.net

Contract/grant sponsor: National Science Foundation of China

Published online 8 April 2004

Received 23 July 2003
Revised 27 January 2004

functions rely on the spectral approximation properties [9]. These properties cannot be stated with the same accuracy in the case of Lagrange polynomials. Sinc functions seem to capture oscillating behaviours in space, hence, are useful to deal with problems characterized by this type of solutions. In this paper, we find that the use of Sinc functions significantly improves computational accuracy and efficiency for the solution of non-linear evolution equations which develop interior gradients, while the use of Lagrange interpolation gives poor accuracy.

When a partial differential equation is discretized in the spatial domain by using the collocation method, the resulting equations in time are usually stiff. Stiff problems arise in many areas. If the methods which have finite regions of absolute stability are used to solve stiff equations, the step size Δt must be constrained so that the stability requirement can be satisfied. Thus the step size Δt must be a sufficiently small. This leads to require much computational effort to solve the stiff equations. Because A-stable methods have no stability constraint on Δt, A-stable methods are suitable to solve stiff equations. However, it has proved that all the classical explicit multistep methods and the explicit Runge–Kutta methods are not A-stable and the implicit linear multistep methods of order greater than two cannot be A-stable [10]. The implicit Runge–Kutta methods are a class of methods which have suitable stability characteristics for use on stiff systems. However, the difficulty of these methods is that there is a large system of non-linear equations to solve at each step, which requires much computational effort. Consequently, these methods are not very efficient. In this paper, an A-stable method called the improved rational approximation method is presented based on the scaling and squaring with diagonal *Padé* approximation method [11] and the precise integration method [12]. The new method is highly accurate and easy to program.

There are many numerical computational techniques for solving the non-linear evolution equation. The implicit Crank–Nichlson scheme is one of the most popular numerical methods. The merit of the implicit Crank–Nichlson scheme is that the method can produce highly accurate solutions yet keeping unconditional numerical stability for non-linear evolution equation. However, the much more computational effort of solving the result implicit non-linear algebraic equations has led most researches to stay with explicit methods and their potential instabilities. In this paper, an increment linearized method is proposed to linearize the non-linear evolution equation. The characteristic of the method is to compute the values of the increment function, $\Delta u = u^{k+1} - u^k$, and then to obtain the values of u^{k+1} by $u^{k+1} = u^k + \Delta u$. We linearize the equation of the increment variable Δu by neglecting the non-linear terms of the order $O(\Delta t^2)$. The numerical experiments demonstrate that the increment linearized method has the advantages of easy formulation, high accuracy and relatively little computational effort.

In this paper, a new numerical method called linearized and rational approximation method (LRAM) is presented to solve non-linear evolution equations. The utility of LRAM is demonstrated through the solution of Burger's equation, a popular benchmark for testing numerical solutions to partial differential equation.

The contents of this paper refer to the above topics and developed in six sections. In particular:

- the first one is the Introduction.
- Section 2 provides technical information on the Sinc functions.
- Section 3 presents the improved rational approximation method for the matrix exponential and its stability.

- Section 4 introduces the main steps of the LRAM through the solution of Burgers' equation.
- Section 5 provides computational experiments and analysis.
- Last section provides a critical discussion of the contents of this paper.

2. SINC FUNCTIONS

Consider $u=u(t,x)$ defined over $[0,1]\times[0,1]$, such that $u=u(x;t)$ is a one one map from $[0,1]$ into $[0,1]$, for every $t\in[0,1]$. Moreover, consider the collocation

$$i=1,\ldots,n,\quad I_x=\{x_1=0,\ldots,x_i,\ldots,x_n=1\} \tag{1}$$

which may be equally spaced, with $x_i=(i-1)h$ and $h=1/(n-1)$. In general, $u=u(t,x)$ can be interpolated and approximated by means of Sinc-type functions

$$u(t,x)=\sum_{j=1}^{n} S_j(x;h)u_j(t) \tag{2}$$

where $u_j(t)=u(t,x_j)$ and

$$S_j(x,h)=\frac{\sin z_j}{z_j},\quad z_j=\frac{\pi}{h}(x-jh) \tag{3}$$

Taking the derivative of Equation (2) to the first and second order, we have

$$\frac{\partial u}{\partial x}(t;x_i)=\sum_{j=1}^{n} a_{ij}u_j(t),\quad \frac{\partial^2 u}{\partial x^2}(t;x_i)=\sum_{j=1}^{n} b_{ij}u_j(t) \tag{4}$$

where a_{ij} and b_{ij} are weighting coefficients related to first- and second-order derivatives, namely

$$a_{ij}=\frac{\partial S_j}{\partial x}(x_i;h),\quad b_{ij}=\frac{\partial^2 S_j}{\partial x^2}(x_i;h) \tag{5}$$

According to Reference [8], we have

$$a_{ij}=\frac{(-1)^{i-j}}{h(i-j)},\quad a_{ii}=0,\quad b_{ij}=\frac{2(-1)^{i-j+1}}{h^2(i-j)^2},\quad b_{ii}=-\frac{1}{3}\left(\frac{\pi}{h}\right)^2,\quad i\neq j \tag{6}$$

Let matrix $A=(a_{ij})$, $B=(b_{ij})$.

The paper by Bonzani [7] organizes how the approximation errors induced by the application of Sinc interpolation techniques propagate in the solution of initial boundary value problems.

Considering that Sinc interpolation is defined on the whole real axis, we consider, following Reference [7], a function $f(t,x)$ such that the variable x is defined on the whole real axis R with f decreasing, for all times, to zero rapidly as $x\to\pm\infty$

$$|f(t;x)|\leqslant A\mathrm{e}^{-\alpha|x|}\quad \forall t\in[0,1] \tag{7}$$

where A and α are positive constants. Let us set

$$f_N(t;x)=\sum_{i=-N}^{N} f_i(t)S_i(x;h) \tag{8}$$

with $N>0$, and consider at fixed time, the error definition

$$e_N=\|E_N(f,h)\|_\infty=\sup_{x\in R}|E_N(f,h)|=\sup_{x\in R}|f-f_N| \tag{9}$$

the step h is chosen as follows:

$$h=\frac{k}{\sqrt{N}},\quad k>0 \tag{10}$$

As known [5], error (9) has the functional form $O(\mathrm{e}^{-c\sqrt{N}})$, i.e. is bounded as follows:

$$\|E_N(f,h)\|_\infty\leqslant C\mathrm{e}^{-c\sqrt{N}} \tag{11}$$

with C and c positive constants depending on A, α, and k. According again to Reference [5], for the r-derivatives of $f(t,x)$ the following estimate holds:

$$e_N^{(r)}=\|E_N^{(r)}(f,h)\|_\infty=\sup_{x\in R}|f^{(r)}-f_N^{(r)}|\leqslant C_r N^{r+1/2}\mathrm{e}^{-c_r\sqrt{N}} \tag{12}$$

where C_r and c_r are positive constants depending on A,α,k.

Introduce now an auxiliary function $v=v(t,x)$, defined over $[0,1]\times R$, which satisfies inequality (7) for every $t\in[0,1]$ and such that

$$v(t,x)=u(t,x)\quad \forall x\in[0,1] \tag{13}$$

If in Equation (10) the constants N and k (and consequently, the step h) in agreement with collocation (1) defined on [0,1] are chosen as follows:

$$h=\frac{1}{n-1}=\frac{k}{\sqrt{N}} \tag{14}$$

then for every $t\in[0,1]$, Equations (11) and (12) give the upper bounds for both errors $e_N(v,h)$ and $\mathrm{e}_N^{(r)}(v,h)$, related to function v on the whole real axis R, as well as for errors $e_N(v,h)$ and $\mathrm{e}_N^{(r)}(v,h)$ referred to the restriction on [0,1] of the same function. In fact, as a consequence of Equation (13) and of the definition of L_∞-norm, one has

$$\|E_N(v,h)\|_\infty\geqslant\|E_N(u,h)\|_\infty,\quad \|E_N^{(r)}(v,h)\|_\infty\geqslant\|E_N^{(r)}(u,h)\|_\infty \tag{15}$$

Then the above approximations (2) and (4) are proved [5] to have the functional form $O(\mathrm{e}^{-c\sqrt{N}})$ for the rate of convergence of the error of an N-points approximation in the space interval [0,1].

3. THE IMPROVED RATIONAL APPROXIMATION METHOD TO THE MATRIX EXPONENTIAL AND ITS STABILITY

3.1. The improved rational approximation method to the matrix exponential

One of the most frequently computed matrix functions is the exponential

$$e^{At} = \sum_{k=0}^{\infty} \frac{(At)^k}{k!} \tag{16}$$

Numerous algorithms for computing e^{At} have been proposed, but most of them are of dubious numerical quality, as is pointed out in the survey article by Moler and Van Loan [13].

The scaling and squaring with diagonal *Padé* approximation method was presented in Reference [11]. A very useful class of approximations, the *Padé* functions, are used. The *Padé* functions defined by

$$R_{pq}(z) = D_{pq}(z)^{-1} N_{pq}(z) \tag{17}$$

where

$$N_{pq}(z) = \sum_{k=0}^{p} \frac{(p+q-k)!\,p!}{(p+q)!k!(p-k)!} z^k \tag{18}$$

$$D_{pq}(z) = \sum_{k=0}^{q} \frac{(p+q-k)!q!}{(p+q)!k!(q-k)!} (-z)^k \tag{19}$$

Notice that, $R_{p0}(z) = 1 + z + \cdots + z^p/p!$ is the pth order Taylor polynomial. Unfortunately, the *Padé* approximations are good only near the origin. However, this problem can be overcome by exploiting the fact that $e^{At} = (e^{At/m})^m$. In particular, we scale At by m such that $R_{pq}(At/m)$ is a suitable accurate approximation to $e^{At/m}$ and then raise the resulting matrix to the mth power. Typically, m is chosen to be a power of 2. It amounts to repeated squaring.

According to Reference [11], the following estimate holds:

$$\frac{\|e^A - F_{pq}\|_\infty}{\|e^A\|_\infty} \leqslant \varepsilon(p,q)\|A\|_\infty e^{\varepsilon(p,q)\|A\|_\infty} \tag{20}$$

where

$$F_{pq} = \left(R_{pq}\left(\frac{A}{2^N}\right)\right)^{2^N}$$

$$\varepsilon(p,q) = 2^{3-(p+q)} \frac{p!q!}{(p+q)!(p+q+1)!}$$

Based on the scaling and squaring with diagonal *Padé* approximation method and the precise integration method, we present a new method called the improved rational approximation

method to compute e^{At}. In the following we give a brief introduction to the improved rational approximation method.

T is given by

$$T = e^{A\Delta t}$$

It can be rewritten as

$$T = (e^{A\tau})^m = (e^B)^m \tag{21}$$

where $\tau = \Delta t/2^N$, $B = A\tau$ and $m = 2^N$. The choice of N depends on the specific problems.

By using the *Padé* functions with $p = q = 4$, we have

$$e^B \cong \frac{I + \frac{1}{2}B + \frac{3}{28}B^2 + \frac{1}{84}B^3 + \frac{1}{1680}B^4}{I - \frac{1}{2}B + \frac{3}{28}B^2 - \frac{1}{84}B^3 + \frac{1}{1680}B^4} \tag{22}$$

Where I is the unit matrix Equation (22) can be rewritten as

$$e^B \cong \frac{I + \frac{3}{28}B^2 + \frac{1}{1680}B^4 + (\frac{1}{2}I + \frac{1}{84}B^2)B}{I + \frac{3}{28}B^2 + \frac{1}{1680}B^4 - (\frac{1}{2}I + \frac{1}{84}B^2)B} \tag{23}$$

Let $D = B^2$, $U_0 = (\frac{3}{28}I + \frac{1}{1680}D)D$, $V_0 = (\frac{1}{2}I + \frac{1}{84}D)B$, (23) can be expressed as

$$e^B = \frac{I + U_0 + V_0}{I + U_0 - V_0}$$

$$(I + U_0 - V_0)(e^B - I) = 2V_0$$

$$e^B = I + (I + F)^{-1}(2V_0)$$

where $F = U_0 - V_0$. Let $T_0 = (I + F)^{-1}(2V_0)$, we have

$$e^B = I + T_0 \tag{24}$$

Substitution of Equation (24) into Equation (21) gives

$$T = (I + T_0)^{2^N}$$

A recurrence procedure of computing T is proposed

$$T_{i+1} = T_i(2I + T_i), \quad i = 0, 1, \ldots, N - 1 \tag{25}$$

Finally, we have

$$T = I + T_N \tag{26}$$

Since τ is a extremely small time interval, F is 'close' to the zero matrix. The condition number of $I+F$ is small, so that T_0 can be computed accurately by Gauss method. In order to assure the accuracy of T, we only compute the T_i and do not directly compute $I + T_i$, which is the main idea of the precise integration method to obtain the highly accurate solution.

3.2. Stability

Obviously stability of the ordinary differential equation of order one is not only a property of the numerical method but also the differential equation it is used to solve. In order that we can talk about the stability of the numerical method, we always examine the stability when it is used to solve the test problem:

$$y' = \lambda y \tag{27}$$

where λ is a complex constant. (To assure the differential equation inherently stable, we assume that $Re\,\lambda < 0$). The solutions of Equation (27) can be expressed as

$$y = e^{\lambda t} y_0$$

Let the time step be Δt. We can obtain equally space points in time interval:

$$t^0 = 0, \qquad t^1 = \Delta t, \qquad t^2 = 2\Delta t, \ldots, \qquad t^k = k\Delta t, \ldots$$

then

$$y^1 = e^{\lambda \Delta t} y_0, \qquad y^2 = e^{\lambda \Delta t} y^1, \ldots, y^{k+1} = e^{\lambda \Delta t} y^k, \ldots$$

By using the improved rational approximation method, we have

$$y^{k+1} = T y^k \tag{28}$$

where T is obtained by (26). Equation (28) also can be written as

$$y^{k+1} = \left(\frac{1 + \frac{3}{28}(\lambda\tau)^2 + \frac{1}{1680}(\lambda\tau)^4 + \lambda\tau(\frac{1}{2} + \frac{1}{84}(\lambda\tau)^2)}{1 + \frac{3}{28}(\lambda\tau)^2 + \frac{1}{1680}(\lambda\tau)^4 - \lambda\tau(\frac{1}{2} + \frac{1}{84}(\lambda\tau)^2)} \right)^{2^N} y^k$$

If $Re\,\lambda < 0$,

$$\left| \frac{1 + \frac{3}{28}(\lambda\tau)^2 + \frac{1}{1680}(\lambda\tau)^4 + \lambda\tau(\frac{1}{2} + \frac{1}{84}(\lambda\tau)^2)}{1 + \frac{3}{28}(\lambda\tau)^2 + \frac{1}{1680}(\lambda\tau)^4 - \lambda\tau(\frac{1}{2} + \frac{1}{84}(\lambda\tau)^2)} \right| < 1 \quad \forall \tau$$

can be proved easily [14]. Thus the improved rational approximation method is A-stable. There is no restriction on Δt. The method is suitable to solve the stiff equations.

4. LRAM FOR THE NON-LINEAR BURGERS' EQUATION

In this paper, the main steps of the LRAM are introduced through the solution of Burgers' equation.

Burger's equation is the non-linear partial differential equation defined by

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \frac{1}{Re} \frac{\partial^2 u}{\partial x^2} \tag{29}$$

and has become a popular benchmark for testing numerical solutions to partial differential equations. Equation (29) serves as a useful model since it possesses features in common with

the Navier–Stokes equation in that the term $u(\partial u/\partial x)$ represents a non-linear convective term while $(1/Re)(\partial^2 u/\partial x^2)$ represents a dissipative term. The Reynolds number, Re, determines the importance of convection versus that of dissipation. In addition, for certain boundary and initial conditions, the solution can approach a discontinuous or shock behaviour. Finally, since analytical solutions to Burger's equation have been investigated extensively [15–18] and are available for a wide range of initial and boundary conditions [15–20], the convergence properties of the various discretization schemes are readily assessed.

We consider the following initial boundary value problem:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2}, \quad -1 \leqslant x \leqslant 1, \ \ 0 \leqslant t \leqslant T \tag{30}$$

$$u(-1,t) = u(1,t) = 0, \quad 0 \leqslant t \leqslant T \tag{31}$$

$$u(x,0) = -\sin \pi x, \quad -1 \leqslant x \leqslant 1 \tag{32}$$

Cole [16] obtained an analytical solution to (30)–(32) which was later tabulated by Berton and Platzman [20] and is given by

$$u(x,t) = \frac{4\pi}{Re}\left[\sum_{n=1}^{\infty} n a_n \mathrm{e}^{-n^2\pi^2 t/Re} \sin n\pi x \Big/ \left(a_0 + 2\sum_{n=1}^{\infty} a_n \mathrm{e}^{-n^2\pi^2 t/Re} \cos n\pi x\right)\right] \tag{33}$$

where $a_n = (-1)^n I_n(Re/2\pi)$ and $I_n(z)$ denotes the modified Bessel function of the first kind.

In order to be in agreement with collocation (1) defined on $x \in [0,1]$, we make the following co-ordinate transform:

$$\xi = \frac{x+1}{2}, \quad v(\xi,t) = u(x,t) \tag{34}$$

Then Equations (30)–(32) can be expressed as

$$\frac{\partial v}{\partial t} + \frac{1}{2} v \frac{\partial v}{\partial \xi} = \frac{1}{4Re}\frac{\partial^2 v}{\partial \xi^2}, \quad 0 \leqslant \xi \leqslant 1, \ \ 0 \leqslant t \leqslant T \tag{35}$$

$$v(0,t) = v(1,t) = 0, \quad 0 \leqslant t \leqslant T \tag{36}$$

$$v(\xi,0) = -\sin \pi(2\xi - 1), \quad 0 \leqslant \xi \leqslant 1 \tag{37}$$

In the following, we give the main steps of the LRAM for solving Equations (35)–(37).

Step 1: Linearize Equation (35) by the increment linearized method. Let

$$v(\xi,t) = v^k(\xi) + w(\xi,t), \quad t^k \leqslant t \leqslant t^{k+1} \tag{38}$$

where $v^k(\xi) = v(\xi,t^k)$. t^k is the time of the kth layer. Obviously, $w(\xi,t^k) = 0$. Define $\Delta t = t^{k+1} - t^k$. Equation (35) can be expressed as

$$\frac{\partial w}{\partial t} = -\frac{1}{2} w \frac{\partial v^k}{\partial \xi} - \frac{1}{2} v^k \frac{\partial w}{\partial \xi} - \frac{1}{2} w \frac{\partial w}{\partial \xi} + \frac{1}{4Re}\frac{\partial^2 w}{\partial \xi^2} - \frac{1}{2} v^k \frac{\partial v^k}{\partial \xi} + \frac{1}{4Re}\frac{\partial^2 v^k}{\partial \xi^2} \tag{39}$$

Drop the non-linear term, $\frac{1}{2} w\partial w/\partial\xi$, of the order $O(\Delta t^2)$, we can get the linear equation

$$\frac{\partial w}{\partial t} = -\frac{1}{2} w \frac{\partial v^k}{\partial \xi} - \frac{1}{2} v^k \frac{\partial w}{\partial \xi} + \frac{1}{4Re} \frac{\partial^2 w}{\partial \xi^2} - \frac{1}{2} v^k \frac{\partial v^k}{\partial \xi} + \frac{1}{4Re} \frac{\partial^2 v^k}{\partial \xi^2} \tag{40}$$

The increment linearized method has the advantages of easy formulation, high accuracy and relatively little computational effort.

Step 2: Discretize the equation in the spatial domain by applying DQM using the Sinc functions.

The interval $[0,1]$ in ξ direction is discretized by considering the set $\xi_1 = 0, \ldots, \xi_i = (i-1)h, \ldots, \xi_n = 1$, where $h = 1/(n-1)$. By Applying DQM, Equation (40) can be rewritten as,

$$\frac{\partial W}{\partial t} = -\frac{1}{2} W \circ (AV^k) - \frac{1}{2} V^k \circ (AW) + \frac{1}{4Re} BW - \frac{1}{2} V^k \circ (AV^k) + \frac{1}{4Re} BV^k \tag{41}$$

where '$\circ$' denotes the Hadamard product of matrices. $A = (a_{ij})$ and $B = (b_{ij})$ are, respectively, DQ weighting coefficient matrices of the 1st and 2nd order derivative in ξ direction according to (6). $W = (W_i)$ and $V^k = (V_i^k)$ are vectors and $W_i = w(t; \xi_i), V_i^k = v^k(\xi_i)$.

Step 3: Obtain the equation for the internal nodes by using the block matrix technique.

W, V^k, A and B are, respectively, split into the following block matrices:

$$W = \begin{bmatrix} W_1 \\ \bar{W} \\ W_n \end{bmatrix}, \quad V^k = \begin{bmatrix} V_1^k \\ \bar{V}^k \\ V_n^k \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{12}^{\mathrm{T}} & A_{13} \\ A_{21} & \bar{A} & A_{23} \\ A_{31} & A_{32}^{\mathrm{T}} & A_{33} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12}^{\mathrm{T}} & B_{13} \\ B_{21} & \bar{B} & B_{23} \\ B_{31} & B_{32}^{\mathrm{T}} & B_{33} \end{bmatrix} \tag{42}$$

where

$$\bar{W} = [W_2, W_3, \ldots, W_{n-1}]^{\mathrm{T}}, \quad \bar{V}_k = [V_2^k, V_3^k, \ldots, V_{n-1}^k]^{\mathrm{T}}$$

$$A_{11} = a_{11}, \quad A_{12} = [a_{12}, a_{13}, \ldots, a_{1(n-1)}]^{\mathrm{T}}, \quad A_{13} = a_{1n}, \quad A_{21} = [a_{21}, a_{31}, \ldots, a_{(n-1)1}]^{\mathrm{T}}$$

$$A_{23} = [a_{2n}, a_{3n}, \ldots, a_{(n-1)n}]^{\mathrm{T}}, \quad A_{31} = a_{n1}, \quad A_{32} = [a_{n2}, a_{n3}, \ldots, a_{n(n-1)}]^{\mathrm{T}}, \quad A_{33} = a_{nn}$$

$$B_{11} = b_{11}, \quad B_{12} = [b_{12}, b_{13}, \ldots, b_{1(n-1)}]^{\mathrm{T}}, \quad B_{13} = b_{1n}, \quad B_{21} = [b_{21}, b_{31}, \ldots, b_{(n-1)1}]^{\mathrm{T}}$$

$$B_{23} = [b_{2n}, b_{3n}, \ldots, b_{(n-1)n}]^{\mathrm{T}}, \quad B_{31} = b_{n1}, \quad B_{32} = [b_{n2}, b_{n3}, \ldots, b_{n(n-1)}]^{\mathrm{T}}, \quad B_{33} = b_{nn}$$

$$\bar{A} = \begin{bmatrix} a_{22} & \cdots & a_{2(n-1)} \\ \vdots & \ddots & \vdots \\ a_{(n-1)2} & \cdots & a_{(n-1)(n-1)} \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} b_{22} & \cdots & b_{2(n-1)} \\ \vdots & \ddots & \vdots \\ b_{(n-1)2} & \cdots & b_{(n-1)(n-1)} \end{bmatrix}$$

After the block matrix being used, W_1, W_n represent the values at the boundary points. Substituting Equation (42) into Equation (41), we can get the equation satisfied by the internal

nodes

$$\frac{\partial \bar{W}}{\partial t} = -\frac{1}{2}\bar{W} \circ (A_{21}V_1^k + \bar{A}\bar{V}^k + A_{23}V_n^k) - \frac{1}{2}\bar{V}^k \circ (A_{21}W_1 + \bar{A}\bar{W} + A_{23}W_n) + \frac{1}{4Re}(B_{21}W_1 + \bar{B}\bar{W}$$
$$+B_{23}W_n) - \frac{1}{2}\bar{V}^k \circ (A_{21}V_1^k + \bar{A}\bar{V}^k + A_{23}V_n^k) + \frac{1}{4Re}(B_{21}V_1^k + \bar{B}\bar{V}^k + B_{23}V_n^k) \tag{43}$$

According to the boundary condition (36), $V_1^k = V_n^k = 0$, $V_1^{k+1} = V_n^{k+1} = 0$. Thus $W_1 = W_n = 0$. Equation (43) can be expressed as

$$\frac{\partial \bar{W}}{\partial t} = -\frac{1}{2}\bar{W} \circ \bar{A}\bar{V}^k - \frac{1}{2}\bar{V}^k \circ \bar{A}\bar{W} + \frac{1}{4Re}\bar{B}\bar{W} - \frac{1}{2}\bar{V}^k \circ \bar{A}\bar{V}^k + \frac{1}{4Re}\bar{B}\bar{V}^k \tag{44}$$

Equation (44) can be rewritten as

$$\frac{\mathrm{d}\bar{W}}{\mathrm{d}t} = G\bar{W} + D \tag{45}$$

Here

$$G = -diag\left(\frac{1}{2}\bar{A}\bar{V}^k\right) - diag\left(\frac{1}{2}\bar{V}^k\right)\bar{A} + \frac{1}{4Re}\bar{B}$$

$$D = -diag\left(\frac{1}{2}\bar{V}^k\right)\bar{A}\bar{V}^k + \frac{1}{4Re}\bar{B}\bar{V}^k$$

where the matrix G and vector D are known. The $diag$(U) denotes diagonalization of vector U.

Step 4: Solve the ordinary differential equations (45) by the improved rational approximation method.

Equation (45) can be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathrm{e}^{-Gt}\bar{W}) = \mathrm{e}^{-Gt}D \tag{46}$$

Integrate Equation (46), we have

$$\bar{W}^{k+1} = \mathrm{e}^{G\Delta t}\bar{W}^k + \int_{t^k}^{t^{k+1}} \mathrm{e}^{G(t^{k+1}-t)}D\,\mathrm{d}t$$
$$= \mathrm{e}^{G\Delta t}\bar{W}^k + \int_0^{\Delta t} \mathrm{e}^{G(\Delta t-\gamma)}D\,\mathrm{d}\gamma \tag{47}$$

Since $\bar{W}^k = 0$,

$$\bar{W}^{k+1} = \int_0^{\Delta t} \mathrm{e}^{G(\Delta t-\gamma)}D\,\mathrm{d}\gamma \tag{48}$$

Applying the trapezoid integration formula, we have

$$\bar{W}^{k+1} = \frac{\Delta t}{2}(\mathrm{e}^{G\Delta t}D + D) \tag{49}$$

Apply the improved rational approximation method to compute $\mathrm{e}^{G\Delta t}$. Then the value of $\bar{W}^{k+1}$ is obtained. According to (38), we can get the value of V^{k+1}.

The trapezoid integration formula reaches the second-order accuracy. Equation (39) is linearized by neglecting the non-linear terms of the order $O(\Delta t^2)$. According to (20), the order of the improved approximation method computing $\mathrm{e}^{G\Delta t}$ is greater than two. Therefore, scheme (49) can reach the second-order accuracy.

In the following, we introduce how to solve the Burger's equation by the Newton's method in order to be compared with the LRAM. We give the formula of the Jacobian matrix, and computation effort of the Newton's method is reduced greatly. If the Newton's method, not computing the Jacobian matrix, requires much more computation effort to solve the non-linear system of equations.

Equation (35) can be rewritten as

$$\frac{\partial v}{\partial t} - Lv = 0 \tag{50}$$

where

$$Lv = \frac{1}{4Re}\frac{\partial^2 v}{\partial \xi^2} - \frac{1}{2}v\frac{\partial v}{\partial \xi} \tag{51}$$

Now using the central difference approximation for $\partial v/\partial t$, we have

$$\frac{v^{k+1} - v^k}{\Delta t} - \frac{1}{2}L(v^{k+1} + v^k) = 0 \tag{52}$$

where $\Delta t = t^{k+1} - t^k$. Equation (52) also can be expressed as

$$\frac{1}{4Re}\frac{\partial^2 v^{k+1}}{\partial \xi^2} - \frac{1}{2}v^{k+1}\frac{\partial v^{k+1}}{\partial \xi} - \frac{2}{\Delta t}v^{k+1} = -\left(\frac{1}{4Re}\frac{\partial^2 v^k}{\partial \xi^2} - \frac{1}{2}v^k\frac{\partial v^k}{\partial \xi} + \frac{2}{\Delta t}v^k\right) \tag{53}$$

Applying DQM for Equation (53), we have

$$\frac{1}{4Re}BV^{k+1} - \frac{1}{2}V^{k+1} \circ AV^{k+1} - \frac{2}{\Delta t}V^{k+1} = -\left(\frac{1}{4Re}BV^k - \frac{1}{2}V^k \circ AV^k + \frac{2}{\Delta t}V^k\right) \tag{54}$$

where V^k, A and B are defined as (42). So we can get the equation satisfied by the internal nodes

$$\frac{1}{4Re}\bar{B}\bar{V}^{k+1} - \frac{1}{2}\bar{V}^{k+1} \circ \bar{A}\bar{V}^{k+1} - \frac{2}{\Delta t}\bar{V}^{k+1} = F \tag{55}$$

where

$$F = -\left(\frac{1}{4Re}\bar{B}\bar{V}^k - \frac{1}{2}\bar{V}^k \circ \bar{A}\bar{V}^k + \frac{2}{\Delta t}\bar{V}^k\right)$$

Let $Y = \bar{V}^{k+1}$, we have

$$\varphi(Y) = \frac{1}{4Re}\bar{B}Y - \frac{1}{2}Y \circ \bar{A}Y - \frac{2}{\Delta t}Y - F = 0 \tag{56}$$

By using SJT product [2], we have

$$\frac{\partial \varphi(Y)}{\partial Y} = \frac{1}{4Re}\bar{B} - \frac{1}{2}I \diamond \bar{A}Y - \bar{A} \diamond \left(\frac{1}{2}Y\right) - \frac{2}{\Delta t}I$$

where '$\diamond$' denotes the SJT product, and I denotes the identity matrix. According to Reference [2], it also can be expressed as

$$\frac{\partial \varphi(Y)}{\partial Y} = \frac{1}{4Re}\bar{B} - \frac{1}{2}diag(\bar{A}Y) - diag\left(\frac{1}{2}Y\right)\bar{A} - \frac{2}{\Delta t}I \tag{57}$$

where *diag*(V) denotes diagonalization of vector V. Equation (57) is the formula of the Jacobian matrix. Therefore, the iteration formula of the Newton–Raphson method in solution of the Equation (56) is

$$Y^{(n+1)} = Y^{(n)} - \left[\frac{\partial \varphi(Y^{(n)})}{\partial Y}\right]^{-1} \varphi(Y^{(n)}) \tag{58}$$

5. NUMERICAL RESULTS AND ANALYSIS

Table I shows the stiffness of the Equation (45) with different time. In the Table I, s is called the stiffness ratio, $s = \max_{1 \leqslant j \leqslant m} |Re(\lambda_j)| / \min_{1 \leqslant j \leqslant m} |Re(\lambda_j)|$. It can be seen that Equation (45) is a stiff equation and the stiffness is very serious. If the time integration for Equation (35) is performed using a explicit fourth-order Runge–Kutta scheme, it was found that the explicit fourth-order Runge–Kutta scheme had a maximum time-step limitation imposed by stability. In the paper [21], the time step is chosen sufficiently small ($\Delta t = 10^{-4}$) to solve the Burger's equation by the Runge–Kutta scheme. Consequently for Equation (45), the explicit Runge–Kutta schemes are not competitive.

For moderately large values of the Reynolds number, the initial sine wave for Burger's equation develops nearly into a sawtooth wave at a non-dimensional time of about $t = 0.3$. Before this time, numerical evaluation of the solution is relatively easy due to the smooth form of the initial function. At later times, however, computing an accurate numerical solution becomes increasingly difficult as the solution steepens at the origin.

Let the number of grid points in ξ direction is n and the time step is Δt. Define maxerror as being the maximum of the absolute errors. Testing CPU time is executed on a Pentium 733 computer with 256 M memory. Matlab6.5 is used for programming.

Table I. Stiffness of Equation (45).

t	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12
s	1337	1308	1146	944	762	613	495	400	322	257	203	157

Table II. Maximal error for Burgers' equation by the LRAM: $t = 0.3$, $\Delta t = 0.03$, $Re = 100$.

n	29	33	37	41	45	49	53	57
Maxerror	0.0093	0.0075	0.0063	0.0054	0.0048	0.0043	0.0039	0.0036
CPU time (s)	0.34	0.36	0.36	0.40	0.40	0.41	0.42	0.45

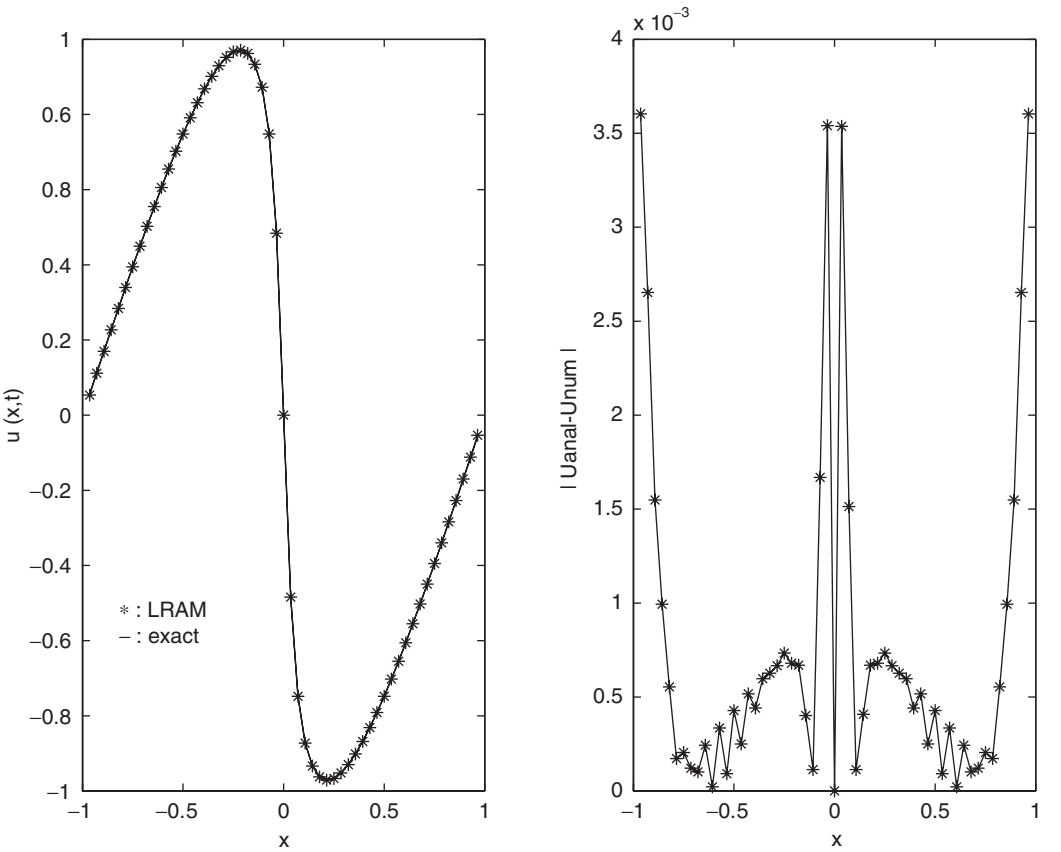

Figure 1. Numerical results and absolute errors for Burgers' equation by the LRAM ($Re = 100$, $t = 0.3$, $\Delta t = 0.03$, $n = 55$).

The maximal error at $t = 0.3$ by the LRAM with the time step $\Delta t = 0.03$ are shown in the Table II using different n. With the increase in n, we observe improved accuracy, as expected. In contrast, as reported in Reference [21], a time step of 10^{-4} was used to achieve comparable accuracy. The computational effort by the LRAM is significantly little. The LRAM proves to be efficient for the stiff problems.

Figures 1–3 show the numerical solutions and absolute errors by the LRAM for $Re = 100$ at $t = 0.3$, 0.4, 0.5, respectively. Also shown is the analytical solution, (33). $|U_{\text{anal}} - U_{\text{num}}|$

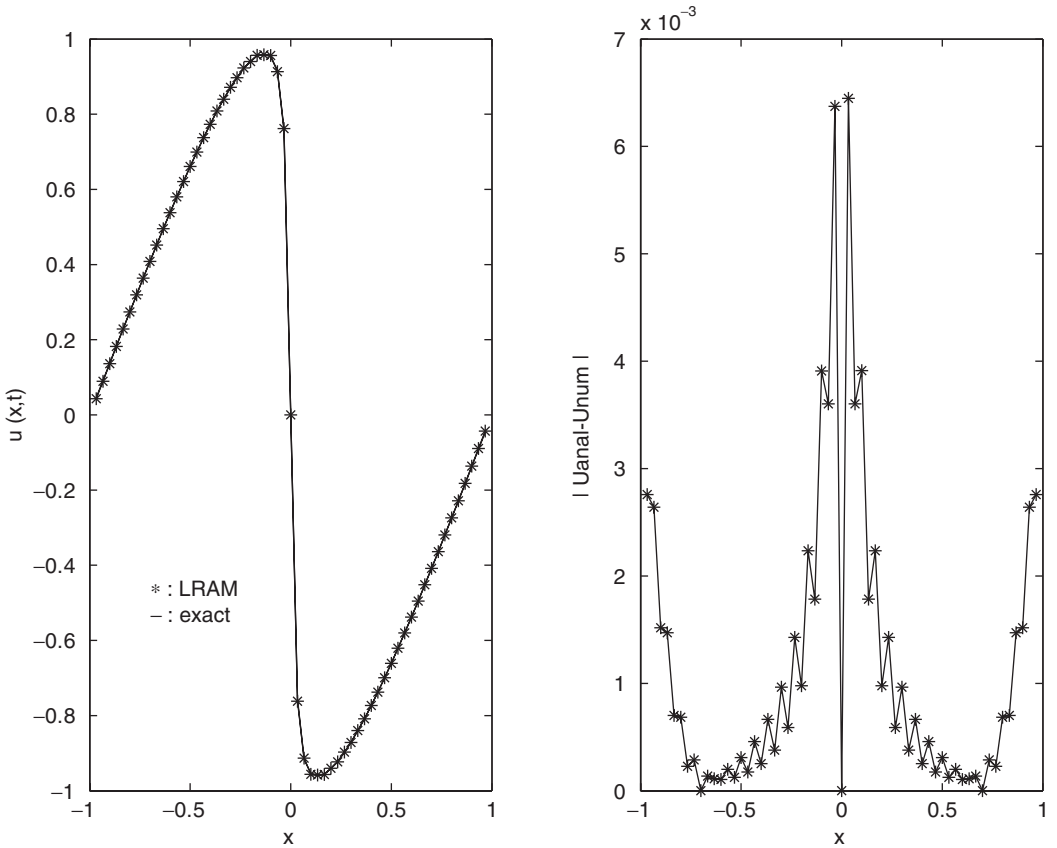

Figure 2. Numerical results and absolute errors for Burgers' equation by the LRAM ($Re = 100$, $t = 0.4$, $\Delta t = 0.02$, $n = 59$).

denotes the absolute errors. As the time is increased, the 'shock' profile becomes steeper. For all cases, the numerical solutions by using the LRAM give good agreement with the analytical ones. The use of the LRAM significantly improves computational accuracy for the problems involving steep gradients. The errors obtained by using the LRAM are substantially smaller than those obtained using the methods in Reference [21]. Figures 1–3 show that the boundary layer is sharp and has no overshoot. The numerical solutions by using the LRAM are essentially oscillation-free. In contrast to the split range quadrature discretization method (SRQDM) in Reference [21], the LRAM requires much less formulation and programming effort and can yield highly accurate solutions with relatively little computational effort. The numerical results by using the LRAM still remain highly accurate even though the time t reaches 0.5. In fact, we can get more accurate numerical solutions by decreasing the time step size and meanwhile increasing the number of grid points. For example, the maximal error obtained by using the LRAM is 4.43×10^{-4} at $t = 0.3$ for $Re = 100$, $\Delta t = 0.0003$ and $n = 500$. The Sinc functions are known to yield spectral accuracy for smooth periodic functions [22].

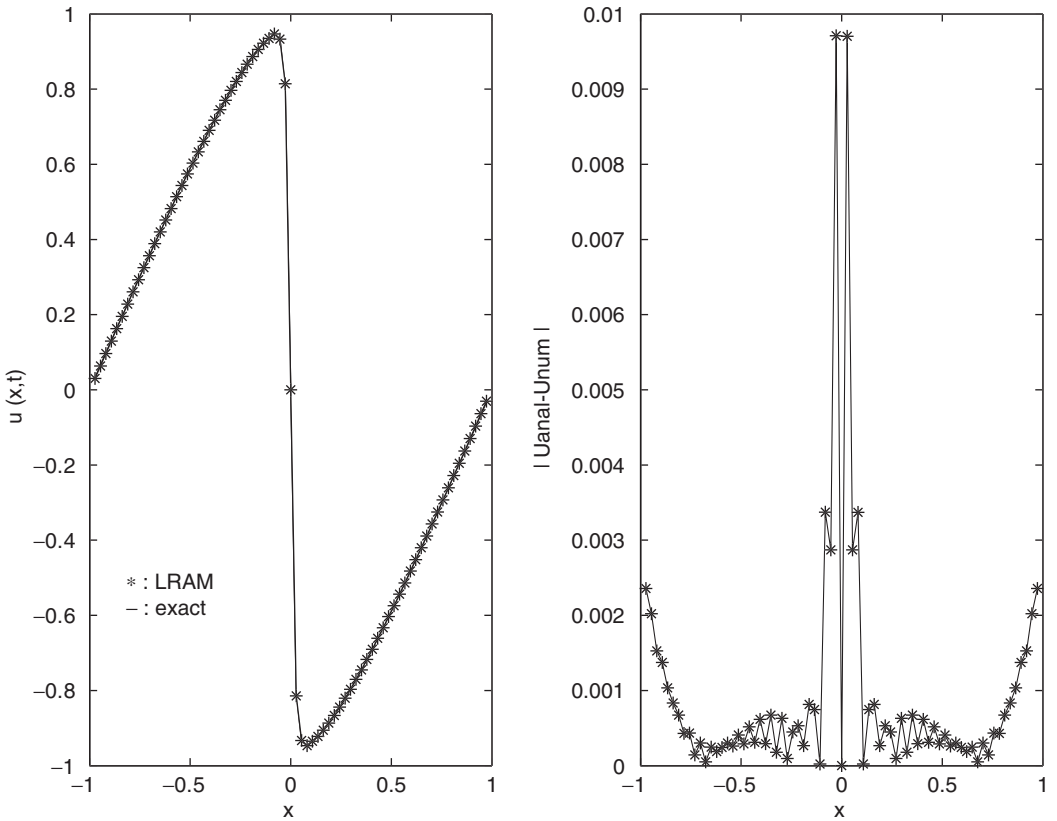

Figure 3. Numerical results and absolute errors for Burgers' equation by the LRAM ($Re = 100$, $t = 0.5$, $\Delta t = 0.025$, $n = 73$).

Table III. Maximal error for Burgers' equation by Newton's method: $t = 0.3$, $\Delta t = 0.03$, $Re = 100$.

n	29	33	37	41	45	49	53	57
Maxerror	0.0094	0.0075	0.0063	0.0055	0.0049	0.0044	0.0040	0.0038
CPU time	0.44	0.44	0.44	0.45	0.47	0.50	0.51	0.55

The maximal errors at $t = 0.3$ by the Newton's method (58) using different n are shown in Table III. The difficulty of using the Newton's method is to obtain the Jacobian matrix. It is well known that the Jacobian matrix is not easy to compute. If the Newton's method, not computing the Jacobian matrix, requires much more computation effort to solve the non-linear system of equations. The LRAM overcomes the above drawbacks of the Newton's method. The LRAM linearizes the non-linear evolution equation by neglecting the non-linear terms of the order $O(\Delta t^2)$, and it can yield highly accurate solutions with little computation effort. Here, we obtained the formula of the Jacobian matrix, and computation effort of using the

Table IV. Computational-efficiency comparison, Burgers' equation, $t = 0.3$, $\Delta t = 0.03$, $Re = 100$.

n	21	23	25	27	29	31	33
Maxerror(Sinc)	0.0261	0.0207	0.0157	0.0121	0.0093	0.0087	0.0075
Maxerror(Lagrange)	0.0574	0.0499	0.0432	0.0373	0.0321	0.0275	0.0234

Newton's method is reduced greatly. The computational efficiency of the LRAM and the Newton's method (58) can be compared by considering the results shown in the Tables II and III. It can be seen that the CPU time by the LRAM is less than the one by the Newton's method to achieve comparable accuracy.

The computational-efficiency comparison of the Sinc functions and Lagrange interpolations for Burgers' equation at $t = 0.3$ is shown in Table IV. The expansive Chebyshev–Gauss–Loabtto points are used in the case of Lagrange interpolations. It is apparent that the Sinc functions are more efficient than the Lagrange interpolations for the Burgers' equation involving steep gradients. It also can be seen that the accuracy of the solution increases rapidly with n by using the LRAM.

6. CONCLUSIONS

We have shown that the use of the LRAM significantly improves computational accuracy for the problems involving steep gradients. The LRAM requires much less formulation and programming effort and can yield highly accurate solutions for non-linear Burgers' equations with relatively little computational effort. We believe that the LRAM can be applied more widely in scientific and engineering computation. In addition, we find that the Sinc functions are more efficient than the Lagrange interpolations for the Burgers' equation involving steep gradients.

ACKNOWLEDGEMENT

Research supported by the National Science Foundation of China.

REFERENCES

1. Bert CW, Malik M. Differential quadrature method in computional mechanics: a review. *Applied Mechanics Reviews* 1996; **49**:1–28.
2. Chen W. Differential quadrature method and its applications in engineering-applying special matrix product to non-linear computations. *Ph.D. Thesis*, Shanghai Jiao Tong University, 1996.
3. Bellman R, Casti J. Differential quadrature and long term integration. *Journal of Mathematical Analysis and Applications* 1971; **34**:235–238.
4. Lund J, Bowers K. *Sinc Methods for Quadratures and Differential Equations*. SIAM: Philadelphia, PA, 1992.
5. Stenger F. *Numerical Methods Based on Sinc and Analytic Functions*. Springer: Berlin, 1993.
6. Bellomo N, Ridolfi L. Solution of non-linear initial-boundary value problem by Sinc collocation–interpolation methods. *Computers & Mathematics and Applications* 1995; **29**(4):15–28.
7. Bonzani I. Solution of non-linear evolution problems by parallelized collocation–interpolation methods. *Computers & Mathematics and Applications* 1997; **34**(12):71–79.
8. Bonzani I. Domain decomposition and discretization of continuous mathematical models. *Computers & Mathematics and Applications* 1999; **38**:79–88.

9. Bellomo N, De Angelis E, Graziano L, Romano A. Solution of non-linear problems in applied sciences by generalized collocation methods and mathematica. *Computers & Mathematics and Applications* 2001; **41**: 1343–1363.
10. Noye J. *Computional Techniques for Differential Equations*. Elsevier Science Publishers B.V.: Netherlands, 1984.
11. Golub GH, Van Loan CF. *Martix Computation* (3rd edn). Johns Hopkins University Press: Baltimore, MD, 1996.
12. Zhong WX. *Applied Mechanics Dual System*. Science Publishers: China, 2002 (in Chinese).
13. Moler CB, Van Loan CF. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review* 1978; **20**:801–836.
14. Hairer E, Wannex G. *Solving Ordinary Differential Equations (2)—Stiff and Differential-Algebraic Problems* (2nd revised edn). Springer: Heidelberg, 1996.
15. Fletcher CAJ. Burgers' equation: a model for all reasons. In *Numerical Solutions of Partial Differential Equations*, Noye J (ed). North-Holland: Amsterdam, 1982.
16. Cole JD. On a quasi-linear parabolic equation occurring in aerodynamics. *Quarterly of Applied Mathematics* 1951; **9**:225–236.
17. Hopf E. The partial differential equation $u_t + uu_x = \mu u_{xx}$. *Communications on Pure and Applied Mathematics* 1950; **3**:201–230.
18. Rodin EY. On some approximate and exact solutions of boundary value problems for Burgers' equation. *Journal of Mathematical Analysis and Applications* 1970; **30**:401–414.
19. Rodin EY. A riccati solution for Burgers' equation. *Quarterly of Applied Mathematics* 1970; **27**:541–544.
20. Benton ER, Platzman GN. A table of solutions of the one-dimensional Burgers equation. *Quarterly of Applied Mathematics* 1972; **30**:195–212.
21. Mansell G, Merryfield W, Shizgal B, Weinert U. A comparison of differential quadrature methods for the solution of partial differential equations. *Computer Methods in Applied Mechanics and Engineering* 1993; **104**: 295–316.
22. Trefethen LN. *Spectral Methods in Matlab*. SIAM: Philadelphia, PA, 2000.

