

IMPROVED FINITE INTEGRATION METHOD FOR MULTI-DIMENSIONAL NONLINEAR BURGERS' EQUATION WITH SHOCK WAVE

Y. LI, M. LI, AND Y. C. HON

College of Mathematics, Taiyuan University of Technology
Taiyuan 030024, China

College of Mathematics, Taiyuan University of Technology, Taiyuan 030024, China

Department of Mathematics, City University of Hong Kong
Hong Kong SAR, China

ABSTRACT. Based on the recently developed finite integration method (FIM) for solving one-dimensional linear partial differential equations, we extend in this paper the method to tackle multi-dimensional nonlinear problems with shock wave. One of the main advantages of FIM is the use of numerical integration instead of finite quotient formula in approximating the solution and its derivatives. This completely avoids the well known roundoff-discretization error optimization problem in Finite Difference Method (FDM) and hence can solve stiff problems with certain kinds of singularity. In this paper, we further extend the FIM for multi-dimensional partial differential equations and demonstrate this advantage by solving both 1D and 2D nonlinear Burgers' equations with shock waves. Numerical results indicate that the FIM gives a convergence order of $O(h^2)$ in using the simplest trapezoidal rule for numerical integration. It is expected to achieve better convergence rate if higher order numerical quadrature formula such as Simpson's rule is adopted. The second main advantage of the FIM is its nearly lower triangular resultant matrix system which can be inverted easily by using standard matrix solver such as Matlab. In order to better capture the stiff shock wave when the Reynold number is large in solving the nonlinear Burger's equation, we adapt a high order Lagrange interpolating scheme for fitting the boundary condition and transform the original resultant coefficient matrix to get an improved FIM solution. Numerical 1D and 2D examples are given and compared with the existing numerical methods of FDM, Finite Element Method (FEM) and meshless Radial Basis Functions (RBFs) method. Sensitivity analyses on temporal and spatial step lengths are also performed to indicate the stability of the FIM.

AMS (MOS) Subject Classification. 65M70.

1. Background

Partial differential equations arose from the studies of continuous physical phenomena. Among these the modeling of advection and dispersion effects in ocean hydrology to simulate the movement of shock wave resulted into a nonlinear Burger's equation [1]. In its simplest form, the Burger's equation contains a nonlinear advection term and a dissipation term for the simulation of wave motion. It serves as a

simplification of a more complex and sophisticated model for physical phenomena such as acoustic transmission; aerofoil flow; turbulence; and supersonic flow. In general, if the initial concentration distribution and boundary condition are given, an analytical solution is attainable from solving a well-posed problem defined on regular domain. In real application, however, the domain is usually irregular and some kinds of numerical methods are inevitable for approximation of the solution. The most well developed numerical methods for solving partial differential equations are FDM and FEM. In simulating stiff shock wave from approximating the solution of nonlinear Burger's equation, the FDM encounters serious difficulty due to the use of finite quotient formula which fails to tackle the discontinuity of the solution's derivatives at the peak of the shock. The FEM with moving nodes technique by Herbst et al. [2] and Caldwell et al. [3] and least squares approach by Kutluay et al. [4] can better capture the moving shock wave but becomes a tedious task for problems in higher dimension. In the last decades, some meshless methods using RBFs by Hon & Mao [5] and GRKPM by Hashemian and Shodja [6] have been developed as an alternative numerical scheme for solving one-dimensional nonlinear Burger's equation.

For two-dimensional problems, the existing works include the use of fully implicit finite difference scheme by Bahadir [7]; extended FEM by [8, 9]; fourth order compact finite element scheme with a fourth order Du Fort Frankel algorithm by Radwan [10]; generalized Boundary Element Method (BEM) by Kakuda and Tosaka [11]; dual reciprocity BEM by Tosaka [12]; modified RBFs by Li et al. [13]; and Method of Fundamental Solutions (MFS) by Young et al. [14].

Recently, Wen et al. [15, 16] developed a new FIM to solve one-dimensional partial differential equations. Unlike FDM which uses finite quotient formula, the FIM uses numerical quadratic integration rule and hence avoids the well-known roundoff-discretization error optimization problem in using FDM. Numerical results given in this paper show that, even with the simplest trapezoidal rule for numerical integration, the FIM provides a very stable, efficient, and highly accurate solution for solving the multi-dimensional nonlinear Burger's equation with shock wave. Similar to the finite difference method, the FIM approximates the solution at the grid points, equally or unequally distributed, by using numerical integration and hence can provide a stable, accurate and efficient approximated solution to stiff problems with shock wave. In order to better capture the stiff shock wave when the Reynold number is large in solving the nonlinear Burger's equation, we adapt a high order Lagrange interpolating scheme for fitting the boundary condition and transform the original resultant coefficient matrix to get an improved FIM solution. Numerical 1D and 2D examples are given and compared with the existing numerical methods of FDM, Finite Element Method (FEM) and meshless Radial Basis Functions (RBFs)

method. Sensitivity analyses on temporal and spatial step lengths are also performed to indicate the stability of the FIM.

The paper is organised as follow: In Section 2, a brief introduction on the finite integration method for one-dimensional partial differential equation and its extension for multi-dimensional partial differential equation are given. The application of this improved FIM for solving multi-dimensional nonlinear Burgers' equation is demonstrated in Section 3 while numerical examples given in Section 4 for both 1D and 2D problems with comparison to existing works and sensitivity analyses on temporary and spatial step lengths. Conclusion is then given in the final Section 5.

2. Finite integration method

Assume that $f(x)$ is defined on $[a, b]$, which contains equally spaced nodes $x_k = a + (k - 1)h$ where $h = (b - a)/N$, $k = 1, 2, \dots, N - 1$. In each subinterval, we use the numerical quadrature formula by trapezoidal rule:

$$(2.1) \quad \int_{x_k}^{x_{k+1}} f(x)dx \approx \frac{f(x_k) + f(x_{k+1})}{2}h.$$

The integration of $f(x)$ from x_1 to x_k :

$$(2.2) \quad F^{(1)}(x_k) = \int_{x_1}^{x_k} f(\xi)d\xi,$$

can be expressed in terms of function values $f(x_i)$ as:

$$(2.3) \quad F^{(1)}(x_k) = \sum_{i=1}^{k-1} \int_{x_i}^{x_{i+1}} f(\xi)d\xi \approx \sum_{i=1}^{k-1} \frac{f(x_{i+1}) + f(x_i)}{2}h \triangleq \sum_{i=1}^k a_{ki}^{(1)} f(x_i),$$

in which

$$(2.4) \quad a_{ki}^{(1)} = \begin{cases} 0, & \\ \left. \begin{array}{l} h/2 \quad i = 1, \\ h \quad i = 2, 3, \dots, k - 1, \quad (k \geq 2), \\ h/2 \quad i = k, \\ 0 \quad \text{otherwise.} \end{array} \right\} \end{cases}$$

In matrix form:

$$(2.5) \quad \mathbf{F}^{(1)} = \mathbf{A}^{(1)}\mathbf{f},$$

where $\mathbf{F}^{(1)} = [F^{(1)}(x_1), \dots, F^{(1)}(x_N)]^T$, $\mathbf{f} = [f(x_1), \dots, f(x_N)]^T$, and $\mathbf{A}^{(1)}$ is the single layer integration matrix:

$$(2.6) \quad \mathbf{A}^{(1)} = h \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 1 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1 & 1 & 1/2 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1/2 & 1 & 1 & 1 & \dots & 1/2 \end{bmatrix}_{N \times N}.$$

The double layer integration of $f(x)$ from x_1 to x_k is:

$$(2.7) \quad F^{(2)}(x_k) = \int_{x_1}^{x_k} \int_{x_1}^{\eta} f(\xi) d\xi d\eta.$$

Using trapezoidal rule we have:

$$(2.8) \quad F^{(2)}(x_k) \approx \sum_{i=1}^k a_{ki}^{(1)} \int_{x_1}^{x_i} f(\xi) d\xi \approx \sum_{i=1}^k \sum_{j=1}^i a_{ki}^{(1)} a_{ij}^{(1)} f(x_j) \triangleq \sum_{i=1}^k a_{ki}^{(2)} f(x_i),$$

which can be written in matrix form:

$$(2.9) \quad \mathbf{F}^{(2)} = \mathbf{A}^{(2)} \mathbf{f} = (\mathbf{A}^{(1)})^2 \mathbf{f},$$

where $a_{ki}^{(2)} = (a_{ki}^{(1)})^2$, $\mathbf{F}^{(2)} = [F^{(2)}(x_1), \dots, F^{(2)}(x_N)]^T$, $\mathbf{A}^{(2)}$ is the double layer integration matrix which can be derived from $\mathbf{A}^{(1)}$.

Similarly, for multiple layer integration of $f(x)$ from x_1 to x_k :

$$(2.10) \quad \begin{aligned} F^{(m)}(x_k) &= \int_{x_1}^{x_k} \dots \int_{x_1}^{\xi_2} f(\xi_1) d\xi_1 \dots d\xi_m \\ &\approx \sum_{i_m=1}^k \dots \sum_{j=1}^{i_1} a_{ki_m}^{(1)} \dots a_{i_1 j}^{(1)} f(x_j) \triangleq \sum_{i=1}^k a_{ki}^{(m)} f(x_i), \end{aligned}$$

whose matrix form can be expressed as:

$$(2.11) \quad \mathbf{F}^{(m)} = \mathbf{A}^{(m)} \mathbf{f} = (\mathbf{A}^{(1)})^m \mathbf{f},$$

where $a_{ki}^{(m)} = (a_{ki}^{(1)})^m$, $\mathbf{F}^{(m)} = [F^{(m)}(x_1), \dots, F^{(m)}(x_N)]^T$, $\mathbf{A}^{(m)}$ is the m^{th} layer integration matrix which can be obtained by $\mathbf{A}^{(1)}$.

For better illustration of the proposed method in two-dimensional case, we assume that the computation domain $\Omega = [a, b] \times [c, d]$ is meshed by a uniform grid with points of $M = N_1 \times N_2$ (N_1 and N_2 are the total number of columns and rows, respectively) with grid sizes h_x and h_y ($h_x = (b-a)/(N_1-1)$, $h_y = (d-c)/(N_2-1)$) in x -direction and y -direction, respectively.

For the convenience of computation, we index the numbering of grid points along x -direction by the global numbering system (Fig. 1(G)) and the grid points along y -direction by the local numbering system (Fig. 1(L)).

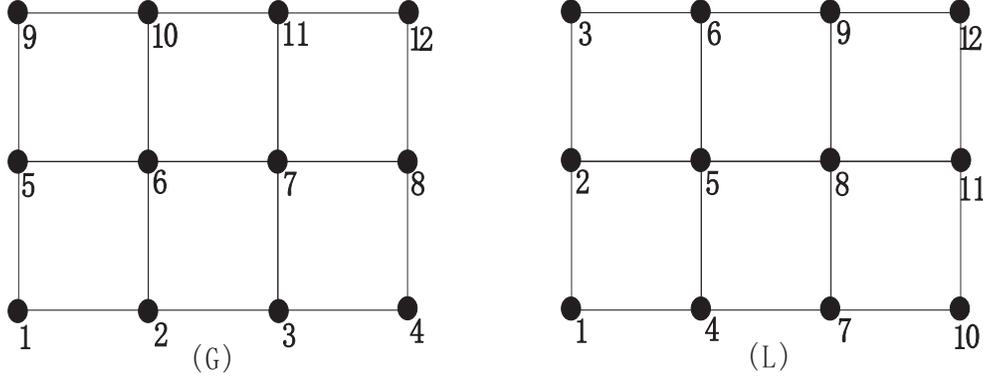


FIGURE 1. Numbering of the grid points globally (G) and locally (L)

Denote $F_x(x, y)$ and $F_y(x, y)$ to be the integration with respect to x and y , respectively. The single and double layer integrations along x -direction in the global numbering system are:

$$(2.12) \quad F_x^{(1)}(x_k, y) = \int_{x_1}^{x_k} f(\xi, y) d\xi \triangleq \sum_{i=1}^k a_{ki,x}^{(1)} f(x_i, y),$$

$$F_x^{(2)}(x_k, y_k) = \int_{x_1}^{x_k} \int_{x_1}^{\xi_2} f(\xi_1, y) d\xi_1 d\xi_2 \triangleq \sum_{i=1}^k a_{ki,x}^{(2)} f(x_i, y),$$

where $k = N_1 \times (j-1) + i$, i and j represent numbers of column and row, respectively. Here, y is considered to be constant. When y is fixed, $a_{ki,x}^{(1)}$ equals $a_{ki}^{(1)}$. Here subscript x is used to denote computation along x -direction. In matrix form:

$$(2.13) \quad \mathbf{F}_x^{(1)} = \mathbf{A}_x^{(1)} \mathbf{f},$$

$$(2.14) \quad \mathbf{F}_x^{(2)} = (\mathbf{A}_x^{(1)})^2 \mathbf{f},$$

where $\mathbf{f} = [f(x_1, y_1), \dots, f(x_k, y_k), \dots]^T$, $\mathbf{F}_x^{(1)} = [F_x^{(1)}(x_1, y_1), \dots, F_x^{(1)}(x_k, y_k), \dots]^T$, $\mathbf{F}_x^{(2)} = [F_x^{(2)}(x_1, y_1), \dots, F_x^{(2)}(x_k, y_k), \dots]^T$ and

$$(2.15) \quad \mathbf{A}_x^{(1)} = \frac{h_x}{h} \underbrace{\begin{bmatrix} \mathbf{A}^{(1)} & 0 & \dots & 0 \\ 0 & \mathbf{A}^{(1)} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathbf{A}^{(1)} \end{bmatrix}}_{N_2}.$$

Here, $\mathbf{A}^{(1)}$ is the integration matrix given in Eq. (2.6) with size $N_1 \times N_1$. Note that these grid points are indexed by the global numerical system.

Similarly, the single and double layer integrations with respect to y can be expressed in the local numbering system using symbols with subscript y as:

$$(2.16) \quad F_y^{(1)}(x, y_l) = \int_{y_1}^{y_l} f(x, \eta) d\eta \triangleq \sum_{i=1}^l \tilde{a}_{li,y}^{(1)} f(x, y_i),$$

$$F_y^{(2)}(x, y_l) = \int_{y_1}^{y_l} \int_{y_1}^{\eta_2} f(x, \eta_1) d\eta_1 d\eta_2 \triangleq \sum_{i=1}^l \tilde{a}_{li,y}^{(2)} f(x, y_i),$$

whose matrix forms are:

$$(2.17) \quad \tilde{\mathbf{F}}_y^{(1)} = \tilde{\mathbf{A}}_y^{(1)} \tilde{\mathbf{f}},$$

$$(2.18) \quad \tilde{\mathbf{F}}_y^{(2)} = (\tilde{\mathbf{A}}_y^{(1)})^2 \tilde{\mathbf{f}},$$

where $\tilde{\mathbf{f}} = [f(x_1, y_1), \dots, f(x_l, y_l), \dots]^T$, $\tilde{\mathbf{F}}_y^{(1)} = [F_y^{(1)}(x_1, y_1), \dots, F_y^{(1)}(x_l, y_l), \dots]^T$, $\tilde{\mathbf{F}}_y^{(2)} = [F_y^{(2)}(x_1, y_1), \dots, F_y^{(2)}(x_l, y_l), \dots]^T$,

$$(2.19) \quad \tilde{\mathbf{A}}_y^{(1)} = \frac{h_y}{h} \underbrace{\begin{bmatrix} \mathbf{A}^{(1)} & 0 & \cdots & 0 \\ 0 & \mathbf{A}^{(1)} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \mathbf{A}^{(1)} \end{bmatrix}}_{N_1},$$

in which $\mathbf{A}^{(1)}$ is the integration matrix given in Eq. (2.6) with size $N_2 \times N_2$. Here, x is considered to be constant and $l = N_2 \times (i-1) + j$, i and j are numbers of column and row, respectively. Note that these grid points are indexed by the local numbering system.

The integration matrix and integrand in the local system can be transformed to the global one by using transformation matrix \mathbf{T} :

$$(2.20) \quad \mathbf{F}_y^{(1)} = \mathbf{T} \tilde{\mathbf{F}}_y^{(1)},$$

$$(2.21) \quad \mathbf{f} = \mathbf{T} \tilde{\mathbf{f}}.$$

For instance, if $N_1 = 4$, $N_2 = 3$, then the transformation matrix \mathbf{T} is given by:

$$(2.22) \quad \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

in which the point $k = N_1 \times (j-1) + i$ in the global numbering system becomes the point $l = N_2 \times (i-1) + j$ in the local numbering system. It is trivial that

\mathbf{T} is nonsingular and all its elements vanish except $\mathbf{T}_{N_1 \times (j-1)+i, N_2 \times (i-1)+j} = 1$ ($i = 1, 2, \dots, N_1, j = 1, 2, \dots, N_2$).

Therefore, the integration matrix with respect to y in the global numbering system can be obtained by a simple re-arrangement on the index number of the grid points:

$$(2.23) \quad \mathbf{A}_y^{(1)} = \mathbf{T} \tilde{\mathbf{A}}_y^{(1)} \mathbf{T}^{-1},$$

where the inverse of matrix \mathbf{T} is simply obtained by

$$(2.24) \quad \mathbf{T}^{-1} = \mathbf{T}^T.$$

For some careful rearrangement of grid points, the matrix \mathbf{T} becomes circulant and related work can then be found from Kuo et al. [17].

Similarly, the higher order integration with respect to x and y can be expressed in the following matrix form:

$$(2.25) \quad \begin{aligned} \mathbf{F}_x^{(m)} &= (\mathbf{A}_x^{(1)})^m \mathbf{f}, \quad (m > 2), \\ \mathbf{F}_y^{(m)} &= (\mathbf{A}_y^{(1)})^m \mathbf{f}. \end{aligned}$$

Remark: The two-dimensional integration can be easily extended to multi-dimensional case:

$$(2.26) \quad \mathbf{F}^{(m_1, m_2, \dots, m_n)} = (\mathbf{A}_{x_1}^{(1)})^{m_1} (\mathbf{A}_{x_2}^{(1)})^{m_2} \dots (\mathbf{A}_{x_i}^{(1)})^{m_i}, \dots (\mathbf{A}_{x_n}^{(1)})^{m_n} \mathbf{f},$$

where $\mathbf{A}_{x_1}^{(1)}, \mathbf{A}_{x_2}^{(1)}, \dots, \mathbf{A}_{x_i}^{(1)}, \dots, \mathbf{A}_{x_n}^{(1)}$ are single layer integration matrices in x_i direction.

It is worth to note that all the integration matrices are lower-triangular whose inversion can be obtained at very low computational cost. This gives the FIM the distinct advantage in providing an efficient numerical scheme for solving partial differential equations. Furthermore, the use of numerical integration instead of finite quotient formula for the approximation of solutions and their derivatives at the grid points avoids the well-known roundoff-discretization error optimization problem in FDM.

To demonstrate these distinct advantages, we apply the FIM to solve the following two-dimensional partial differential equation:

$$(2.27) \quad \alpha_1(x, y) \frac{\partial^2 u(x, y)}{\partial x^2} + \alpha_2(x, y) \frac{\partial^2 u(x, y)}{\partial y^2} + \alpha_3(x, y) u(x, y) = \beta(x, y), \quad (x, y) \in \Omega,$$

under boundary condition:

$$(2.28) \quad \Lambda u(x, y) = \omega(x, y), \quad (x, y) \in \partial\Omega,$$

where $\mathbf{\Lambda}$ is a boundary operator, $\alpha_1(x, y)$, $\alpha_2(x, y)$, $\alpha_3(x, y)$, $\beta(x, y)$ and $\omega(x, y)$ are given functions. Applying the FIM to Eq. (2.27) with four-layer integration, we obtain:

$$\begin{aligned}
(2.29) \quad & \int_{y_1}^y \int_{y_1}^{\eta_2} \int_{x_1}^x \int_{x_1}^{\xi_2} \alpha_1(\xi_1, \eta_1) \frac{\partial^2 u(\xi_1, \eta_1)}{\partial \xi_1^2} + \alpha_2(\xi_1, \eta_1) \frac{\partial^2 u(\xi_1, \eta_1)}{\partial \eta_1^2} \\
& + \alpha_3(\xi_1, \eta_1) u(\xi_1, \eta_1) d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
& = \int_{y_1}^y \int_{y_1}^{\eta_2} \int_{x_1}^x \int_{x_1}^{\xi_2} \beta(\xi_1, \eta_1) d\xi_1 d\xi_2 d\eta_1 d\eta_2.
\end{aligned}$$

Using integration by part, we have

$$\begin{aligned}
(2.30) \quad & \int_{y_1}^y \int_{y_1}^{\eta_2} \left[\alpha_1(x, \eta_1) u(x, \eta_1) - 2 \int_{x_1}^x \frac{\partial \alpha_1(\xi_1, \eta_1)}{\partial \xi_1} u(\xi_1, \eta_1) \right. \\
& + \left. \int_{x_1}^x \int_{x_1}^{\xi_2} \frac{\partial^2 \alpha_1(\xi_1, \eta_1)}{\partial \xi_1^2} u(\xi_1, \eta_1) d\xi_1 d\xi_2 \right] d\eta_1 d\eta_2 \\
& + \int_{x_1}^x \int_{x_1}^{\xi_2} \left[\alpha_2(\xi_1, y) u(\xi_1, y) - 2 \int_{y_1}^y \frac{\partial \alpha_2(\xi_1, \eta_1)}{\partial \eta_1} u(\xi_1, \eta_1) \right. \\
& + \left. \int_{y_1}^y \int_{y_1}^{\eta_2} \frac{\partial^2 \alpha_2(\xi_1, \eta_1)}{\partial \eta_1^2} u(\xi_1, \eta_1) d\eta_1 d\eta_2 \right] \\
& d\xi_1 d\xi_2 + \int_{y_1}^y \int_{y_1}^{\eta_2} \int_{x_1}^x \int_{x_1}^{\xi_2} \alpha_3(\xi_1, \eta_1) u(\xi_1, \eta_1) d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
& + x f_0(y) + f_1(y) + y g_0(x) + g_1(x) \\
& = \int_{y_1}^y \int_{y_1}^{\eta_2} \int_{x_1}^x \int_{x_1}^{\xi_2} \beta(\xi_1, \eta_1) d\xi_1 d\xi_2 d\eta_1 d\eta_2
\end{aligned}$$

where $f_0(y)$, $f_1(y)$, $g_0(x)$ and $g_1(x)$ are arbitrary functions assumed to be approximated by Lagrange interpolating polynomial:

$$\begin{aligned}
(2.31) \quad & f_l(y) = \sum_{k=1}^{P_2} \prod_{\substack{j=1 \\ j \neq k}}^{P_2} \frac{(y - \bar{y}_j)}{(\bar{y}_k - \bar{y}_j)} f_l(\bar{y}_k), \\
& g_l(x) = \sum_{k=1}^{P_1} \prod_{\substack{i=1 \\ i \neq k}}^{P_1} \frac{(x - \bar{x}_i)}{(\bar{x}_k - \bar{x}_i)} g_l(\bar{x}_k), \quad (l = 0, 1),
\end{aligned}$$

where (\bar{x}_i, \bar{y}_j) ($i = 1, 2, \dots, P_1$, $j = 1, 2, \dots, P_2$) are interpolated points on the boundary, $f_l(\bar{y}_1), \dots, f_l(\bar{y}_{P_2})$ and $g_l(\bar{x}_1), \dots, g_l(\bar{x}_{P_1})$ are unknown values on these interpolated points which will be determined from the given boundary condition Eq. (2.28).

Eq. (2.30) can be written in matrix form:

$$\begin{aligned}
(2.32) \quad & \{(\mathbf{A}_y^{(1)})^2 [\alpha_1 - 2\mathbf{A}_x^{(1)} \alpha_{1,x} + (\mathbf{A}_x^{(1)})^2 \alpha_{1,xx}] \\
& + (\mathbf{A}_x^{(1)})^2 [\alpha_2 - 2\mathbf{A}_y^{(1)} \alpha_{2,y} + (\mathbf{A}_y^{(1)})^2 \alpha_{2,yy}] + (\mathbf{A}_y^{(1)})^2 (\mathbf{A}_x^{(1)})^2 \alpha_3\} \mathbf{u} \\
& = (\mathbf{A}_y^{(1)})^2 (\mathbf{A}_x^{(1)})^2 \beta - \mathbf{X} \Phi_y \mathbf{f}_0 - \Phi_y \mathbf{f}_1 - \mathbf{Y} \Phi_x \mathbf{g}_0 - \Phi_x \mathbf{g}_1,
\end{aligned}$$

where $\mathbf{X} = \text{diag}\{x_1, x_2, \dots, x_M\}$, $\mathbf{Y} = \text{diag}\{y_1, y_2, \dots, y_M\}$, $\mathbf{f}_l = [f_l^1, \dots, f_l^i, \dots, f_l^{P_2}]^T$, $\mathbf{g}_l = [g_l^1, \dots, g_l^j, \dots, g_l^{P_1}]^T$, $f_l^i = f_l(\bar{y}_i)$, $g_l^j = g_l(\bar{x}_j)$, $\mathbf{u} = [u(x_1, y_1), u(x_2, y_2), \dots, u(x_M, y_M)]^T$ and $\beta = [\beta(x_1, y_1), \beta(x_2, y_2), \dots, \beta(x_M, y_M)]^T$,

$$(2.33) \quad \alpha_i = \begin{bmatrix} \alpha_i(x_1, y_1) & 0 & \cdots & 0 \\ 0 & \alpha_i(x_2, y_2) & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \alpha_i(x_M, y_M) \end{bmatrix}_{M \times M} \quad (i = 1, 2, 3),$$

$$(2.34) \quad \alpha_{i,\cdot} = \begin{bmatrix} \frac{\partial \alpha_i(x, y)}{\partial \cdot} \Big|_{(x_1, y_1)} & 0 & \cdots & 0 \\ 0 & \frac{\partial \alpha_i(x, y)}{\partial \cdot} \Big|_{(x_2, y_2)} & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \frac{\partial \alpha_i(x, y)}{\partial \cdot} \Big|_{(x_M, y_M)} \end{bmatrix}_{M \times M} \quad (i = 1, 2, 3),$$

$$(2.35) \quad \Phi_y = \begin{bmatrix} \prod_{k=2}^{P_2} \frac{y_1 - \bar{y}_k}{\bar{y}_1 - \bar{y}_k} & \cdots & \prod_{\substack{k=1 \\ k \neq j}}^{P_2} \frac{y_1 - \bar{y}_k}{\bar{y}_j - \bar{y}_k} & \cdots & \prod_{k=1}^{P_2-1} \frac{y_1 - \bar{y}_k}{\bar{y}_{P_2} - \bar{y}_k} \\ \vdots & & \vdots & & \vdots \\ \prod_{k=2}^{P_2} \frac{y_i - \bar{y}_k}{\bar{y}_1 - \bar{y}_k} & \cdots & \prod_{\substack{k=1 \\ k \neq j}}^{P_2} \frac{y_i - \bar{y}_k}{\bar{y}_j - \bar{y}_k} & \cdots & \prod_{k=1}^{P_2-1} \frac{y_i - \bar{y}_k}{\bar{y}_{P_2} - \bar{y}_k} \\ \vdots & & \vdots & & \vdots \\ \prod_{k=2}^{P_2} \frac{y_M - \bar{y}_k}{\bar{y}_1 - \bar{y}_k} & \cdots & \prod_{\substack{k=1 \\ k \neq j}}^{P_2} \frac{y_M - \bar{y}_k}{\bar{y}_j - \bar{y}_k} & \cdots & \prod_{k=1}^{P_2-1} \frac{y_M - \bar{y}_k}{\bar{y}_{P_2} - \bar{y}_k} \end{bmatrix}_{M \times P_2},$$

$$(2.36) \quad \Phi_x = \begin{bmatrix} \prod_{k=2}^{P_1} \frac{x_1 - \bar{x}_k}{\bar{x}_1 - \bar{x}_k} & \cdots & \prod_{\substack{k=1 \\ k \neq j}}^{P_1} \frac{x_1 - \bar{x}_k}{\bar{x}_j - \bar{x}_k} & \cdots & \prod_{k=1}^{P_1-1} \frac{x_1 - \bar{x}_k}{\bar{x}_{P_1} - \bar{x}_k} \\ \vdots & & \vdots & & \vdots \\ \prod_{k=2}^{P_1} \frac{x_i - \bar{x}_k}{\bar{x}_1 - \bar{x}_k} & \cdots & \prod_{\substack{k=1 \\ k \neq j}}^{P_1} \frac{x_i - \bar{x}_k}{\bar{x}_j - \bar{x}_k} & \cdots & \prod_{k=1}^{P_1-1} \frac{x_i - \bar{x}_k}{\bar{x}_{P_1} - \bar{x}_k} \\ \vdots & & \vdots & & \vdots \\ \prod_{k=2}^{P_1} \frac{x_M - \bar{x}_k}{\bar{x}_1 - \bar{x}_k} & \cdots & \prod_{\substack{k=1 \\ k \neq j}}^{P_1} \frac{x_M - \bar{x}_k}{\bar{x}_j - \bar{x}_k} & \cdots & \prod_{k=1}^{P_1-1} \frac{x_M - \bar{x}_k}{\bar{x}_{P_1} - \bar{x}_k} \end{bmatrix}_{M \times P_1}.$$

Eq. (2.32) can be simplified as:

$$(2.37) \quad \mathbf{K}\mathbf{u} = \mathbf{B},$$

where

$$(2.38) \quad \mathbf{K} = \begin{bmatrix} \mathbf{L} & \mathbf{F} \\ \mathbf{D} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_i \\ \mathbf{B}_b \end{bmatrix}.$$

Here, $\mathbf{L} = (\mathbf{A}_y^{(1)})^2[\alpha_1 - 2\mathbf{A}_x^{(1)}\alpha_{1,x} + (\mathbf{A}_x^{(1)})^2\alpha_{1,xx}] + (\mathbf{A}_x^{(1)})^2[\alpha_2 - 2(\mathbf{A}_y^{(1)})\alpha_{2,y} + (\mathbf{A}_y^{(1)})^2\alpha_{2,yy}] + (\mathbf{A}_y^{(1)})^2(\mathbf{A}_x^{(1)})^2\alpha_3$ is a lower triangular matrix in which more than half of the elements are zero, $\mathbf{F} = \mathbf{X}\Phi_y + \Phi_y + \mathbf{Y}\Phi_x + \Phi_x$ is a almost full matrix, \mathbf{D} is a nearly diagonal matrix associated with boundary condition, $\mathbf{0}$ is a null matrix, $\mathbf{B}_i = (\mathbf{A}_y^{(1)})^2(\mathbf{A}_x^{(1)})^2\beta$, and $\mathbf{B}_b = [\mathbf{f}_0; \mathbf{f}_1; \mathbf{g}_0; \mathbf{g}_1]$. For illustration, we display in Fig. 2 the sparsity pattern of

matrix \mathbf{K} for the case when $N_1 = N_2 = 5$ and $P_1 = P_2 = 4$. Due to the fact that all elements in the first row of the matrix \mathbf{L} are zeroes and all its first elements are less than unity, the use of the standard Gaussian elimination method with pivoting strategy to solve the linear system Eq. (2.32) will fail to give a stable approximation. In fact, if the size of the matrix \mathbf{K} is large, there appears to have a rank deficiency by using the rank check in Matlab. In our following computation, we overcome this problem by exchanging matrices \mathbf{L} with \mathbf{F} and \mathbf{D} with $\mathbf{0}$ and apply high order Lagrange interpolation scheme instead of simple linear interpolation in the original FIM to obtain a very stable and accurate solution by using the MATLAB matrix solver. After rearrangement,

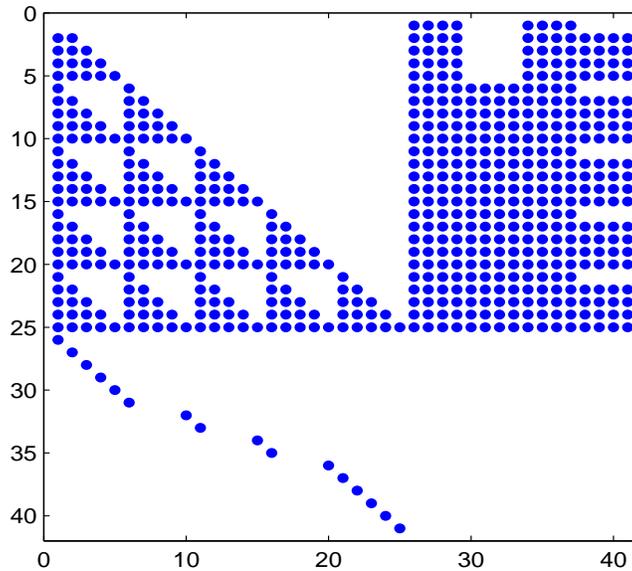


FIGURE 2. The sparsity pattern of matrix \mathbf{K} when $N_1 = N_2 = 5$ and $P_1 = P_2 = 4$

$$(2.39) \quad \mathbf{K} = \begin{bmatrix} \mathbf{F} & \mathbf{L} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_b \\ \mathbf{B}_i \end{bmatrix}.$$

We display in Fig. 3 the new sparsity pattern of matrix \mathbf{K} .

3. Numerical examples for one- and two- dimensional Burgers' equation

To demonstrate the distinct integration advantage of the proposed FIM, we apply in this section the method to solve both one- and two dimensional nonlinear Burger's equations with shock wave. Consider first the following one-dimensional Burgers' equation:

$$(3.1) \quad \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \frac{1}{R} \frac{\partial^2 u}{\partial x^2},$$

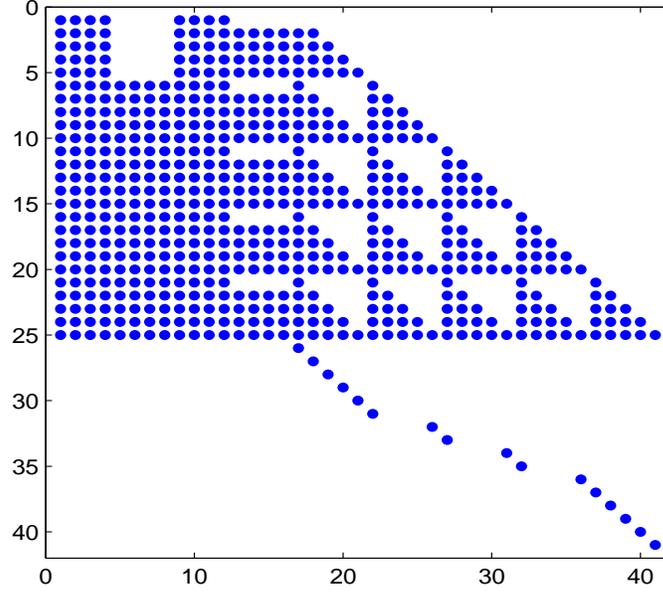


FIGURE 3. The sparsity pattern of matrix \mathbf{K} after rearrangement.

subject to initial condition:

$$(3.2) \quad u(x, t_0) = \phi(x), \quad x \in [a, b],$$

and boundary conditions:

$$(3.3) \quad u(a, t) = \psi_1(t), \quad u(b, t) = \psi_2(t), \quad t > 0.$$

Using the first order forward difference approximation scheme for the time derivative [5], we have:

$$(3.4) \quad u^m + \delta t(u^{m-1} \frac{\partial u^m}{\partial x} - \frac{1}{R} \frac{\partial^2 u^m}{\partial x^2}) = u^{m-1}, \quad (m \geq 1),$$

where δt is time step, u^m and u^{m-1} are numerical values in the m^{th} and $(m - 1)^{th}$ iterations. Applying FIM to Eq. (3.4), we obtain the following equation at point x_k :

$$(3.5) \quad \int_{x_1}^{x_k} \int_{x_1}^{\eta} u^m(\xi) + \delta t \left[u^{m-1}(\xi) \frac{\partial u^m(\xi)}{\partial \xi} - \frac{1}{R} \frac{\partial^2 u^m(\xi)}{\partial \xi^2} \right] d\xi d\eta = \int_{x_1}^{x_k} \int_{x_1}^{\eta} u^{m-1}(\xi) d\xi d\eta.$$

Using integration by part scheme we have:

$$(3.6) \quad \int_{x_1}^{x_k} \int_{x_1}^{\eta} u^m(\xi) d\xi d\eta = \sum_{i=1}^k a_{ki}^{(2)} u^m(x_i),$$

$$(3.7) \quad \int_{x_1}^{x_k} \int_{x_1}^{\eta} u^{m-1}(\xi) \frac{\partial u^m(\xi)}{\partial \xi} d\xi d\eta = \sum_{i=1}^k a_{ki}^{(1)} \int_{x_1}^{x_i} u^{m-1}(\xi) \frac{\partial u^m(\xi)}{\partial \xi} d\xi$$

$$\begin{aligned}
&= \sum_{i=1}^k a_{ki}^{(1)} \sum_{j=1}^{i-1} \frac{u^{m-1}(x_j) + u^{m-1}(x_{j+1})}{2} \int_{x_j}^{x_{j+1}} \frac{\partial u^m(\xi)}{\partial \xi} d\xi \\
&= \frac{1}{2} \sum_{i=1}^k a_{ki}^{(1)} \sum_{j=1}^{i-1} [u^{m-1}(x_j) + u^{m-1}(x_{j+1})][u^m(x_{j+1}) - u^m(x_j) + c_0],
\end{aligned}$$

$$(3.8) \quad \int_{x_1}^{x_k} \int_{x_1}^{\eta} \frac{\partial^2 u^m(\xi)}{\partial \xi^2} d\xi d\eta = u^m(x_k) + c_0 x_k + c_1,$$

where c_0 and c_1 are arbitrary integration constants. By substituting Eqs. (3.7)–(3.6) into Eq. (3.5) we have:

$$(3.9) \quad [(\mathbf{A}^{(1)})^2 + \delta t(\mathbf{A}^{(1)}\mathbf{Q} + \frac{1}{R}\mathbf{I})]\mathbf{u}^m + c_0\mathbf{x} + c_1\mathbf{e} = (\mathbf{A}^{(1)})^2\mathbf{u}^{m-1},$$

where c_0 and c_1 are revised arbitrary constants, \mathbf{I} is the identity matrix, $\mathbf{u}^m = [u^m(x_1), u^m(x_2), \dots, u^m(x_N)]^T$, $\mathbf{u}^{m-1} = [u^{m-1}(x_1), u^{m-1}(x_2), \dots, u^{m-1}(x_N)]^T$, $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$, $\mathbf{e} = [1, 1, \dots, 1]^T$, and $\mathbf{Q} = (q_{ij})_{N \times N}$. Here,

$$\begin{aligned}
q_{1j} &= 0, \\
q_{ij} &= \frac{1}{2} \begin{cases} -u^{m-1}(x_1) - u^{m-1}(x_2), & j = 1, \\ u^{m-1}(x_{j-1}) + u^{m-1}(x_j), & j = i, \\ u^{m-1}(x_{j-1}) - u^{m-1}(x_{j+1}), & j = 2, 3, \dots, i-1, \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}$$

From the given boundary condition (3.3) we then obtain the following system of iterative linear equations for a total of $(N+2)$ unknowns including \mathbf{u}^m , c_0 and c_1 :

$$(3.10) \quad \begin{bmatrix} \mathbf{K} & \mathbf{x} & \mathbf{e} \\ \mathbf{e}_1 & 0 & 0 \\ \mathbf{e}_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^m \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^{(1)})^2\mathbf{u}^{m-1} \\ \psi_1(t) \\ \psi_2(t) \end{bmatrix},$$

where $\mathbf{K} = (\mathbf{A}^{(1)})^2 + \delta t(\mathbf{A}^{(1)}\mathbf{Q} + \frac{1}{R}\mathbf{I})$, $\mathbf{e}_1 = [1, 0, \dots, 0]_{1 \times N}$, $\mathbf{e}_2 = [0, 0, \dots, 1]_{1 \times N}$. The solution u can then be approximated by solving the above system starting from the given initial condition (3.2).

For two-dimensional case, we consider the following coupled Burgers' equation:

$$(3.11) \quad \begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= \frac{1}{R} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= \frac{1}{R} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right), \end{aligned}$$

subject to initial condition:

$$(3.12) \quad u(x, y, t_0) = \phi_1(x, y), v(x, y, t_0) = \phi_2(x, y), (x, y) \in \Omega,$$

and boundary condition:

$$(3.13) \quad u(x, y, t) = \psi_3(x, y, t), v(x, y, t) = \psi_4(x, y, t), (x, y) \in \partial\Omega.$$

As the FIM solution process for u and v are similar, we only give the details in the following for solving u .

Using the forward difference scheme for the time derivative in Eq. (3.11), we have

$$(3.14) \quad u^m + \delta t u^{m-1} \frac{\partial u^m}{\partial x} + \delta t v^{m-1} \frac{\partial u^m}{\partial y} - \frac{\delta t}{R} \left(\frac{\partial^2 u^m}{\partial x^2} + \frac{\partial^2 u^m}{\partial y^2} \right) = u^{m-1}.$$

Applying FIM two-layer integration, we obtain

$$(3.15) \quad \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \int_{x_1}^{x_k} \int_{x_1}^{\xi_2} [u^m(\xi_1, \eta_1) + \delta t u^{m-1}(\xi_1, \eta_1) \frac{\partial u^m(\xi_1, \eta_1)}{\partial \xi_1} + \delta t v^{m-1}(\xi_1, \eta_1) \frac{\partial u^m(\xi_1, \eta_1)}{\partial \eta_1} - \frac{\delta t}{R} \left(\frac{\partial^2 u^m(\xi_1, \eta_1)}{\partial \xi_1^2} + \frac{\partial^2 u^m(\xi_1, \eta_1)}{\partial \eta_1^2} \right)] d\xi_1 d\xi_2 d\eta_1 d\eta_2$$

$$(3.16) \quad = \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \int_{x_1}^{x_k} \int_{x_1}^{\xi_2} u^{m-1}(\xi_1, \eta_1) d\xi_1 d\xi_2 d\eta_1 d\eta_2.$$

Using integration by part technique, we have

$$(3.17) \quad \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \int_{x_1}^{x_k} \int_{x_1}^{\xi_2} u^m(\xi_1, \eta_1) d\xi_1 d\xi_2 d\eta_1 d\eta_2 = \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \left[\sum_{i=1}^k a_{ki,x}^{(2)} u^m(x_i, \eta_1) \right] d\eta_1 d\eta_2 = \sum_{j=1}^k a_{kj,y}^{(2)} \sum_{i=1}^k a_{ki,x}^{(2)} u^m(x_i, y_j),$$

$$(3.18) \quad \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \int_{x_1}^{x_k} \int_{x_1}^{\xi_2} u^{m-1}(\xi_1, \eta_1) \frac{\partial u^m(\xi_1, \eta_1)}{\partial \xi_1} d\xi_1 d\xi_2 d\eta_1 d\eta_2 = \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \left[\sum_{i=1}^k a_{ki,x}^{(1)} \sum_{j=1}^{i-1} \int_{x_j}^{x_{j+1}} u^{m-1}(\xi_1, \eta_1) \frac{\partial u^m(\xi_1, \eta_1)}{\partial \xi_1} d\xi_1 \right] d\eta_1 d\eta_2 = \sum_{i=1}^k a_{ki,x}^{(1)} \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \sum_{j=1}^{i-1} \frac{u^{m-1}(x_j, \eta_1) + u^{m-1}(x_{j+1}, \eta_1)}{2} \times [u^m(x_{j+1}, \eta_1) - u^m(x_j, \eta_1) + f_0(\eta_1)] d\eta_1 d\eta_2 = \frac{1}{2} \sum_{i=1}^k a_{ki,x}^{(1)} \sum_{l=1}^k a_{kl,y}^{(2)} \sum_{j=1}^{i-1} [u^{m-1}(x_j, y_l) + u^{m-1}(x_{j+1}, y_l)] \times [u^m(x_{j+1}, y_l) - u^m(x_j, y_l) + f_0(y_l)],$$

Similarly, we then obtain:

$$(3.19) \quad \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \int_{x_1}^{x_k} \int_{x_1}^{\xi_2} v^{m-1}(\xi_1, \eta_1) \frac{\partial u^m(\xi_1, \eta_1)}{\partial \eta_1} d\xi_1 d\xi_2 d\eta_1 d\eta_2$$

$$\begin{aligned}
&= \frac{1}{2} \sum_{i=1}^k a_{ki,y}^{(1)} \sum_{l=1}^k a_{kl,x}^{(2)} \sum_{j=1}^{i-1} [v^{m-1}(x_l, y_{j+1}) + v^{m-1}(x_l, y_j)] \\
&\quad \times [u^m(x_l, y_{j+1}) - u^m(x_l, y_j) + g_0(x_l)], \\
(3.20) \quad &\int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \int_{x_1}^{x_k} \int_{x_1}^{\xi_2} \frac{\partial^2 u^m(\xi_1, \eta_1)}{\partial \xi_1^2} d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
&= \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} [u^m(x_k, \eta_1) + x_k f_1(\eta_1) + f_2(\eta_1)] d\eta_1 d\eta_2 \\
&= \sum_{i=1}^k a_{ki,y}^{(2)} [u^m(x_k, y_i) + x_k f_1(y_i) + f_2(y_i)],
\end{aligned}$$

and

$$(3.21) \quad \int_{y_1}^{y_k} \int_{y_1}^{\eta_2} \int_{x_1}^{x_k} \int_{x_1}^{\xi_2} \frac{\partial^2 u^m(\xi_1, \eta_1)}{\partial \eta_1^2} d\xi_1 d\xi_2 d\eta_1 d\eta_2 = \sum_{i=1}^k a_{ki,x}^{(2)} [u^m(x_i, y_k) + y_k g_1(x_i) + g_2(x_i)],$$

where $f_0(y)$, $f_1(y)$, $f_2(y)$ and $g_0(x)$, $g_1(x)$, $g_2(x)$ are arbitrary one-dimensional functions whose values to be determined from the given boundary condition as introduced in Eq. (2.31). Finally, we need to solve a system of iterative linear equations:

$$\mathbf{K}\mathbf{u}^m = \mathbf{B}^{(m-1)},$$

where \mathbf{K} and \mathbf{B} are the matrices introduced previously in Eq. (2.38)

Remark: The forward difference technique has the convergence order of $\mathcal{O}(\delta t)$. Therefore, the error estimate for the nonlinear time-dependent Burgers' equation is expected to be $\mathcal{O}(\delta t) + \mathcal{O}(h^2)$. Better convergence rate will be achieved by higher order time integration scheme and numerical quadrature formula such as Simpson's rule and Lagrange formula.

4. Numerical results

For numerical verification of the effectiveness of the proposed FIM, we give in this section both 1D and 2D examples on solving the nonlinear Burger's equations with shock wave. Comparisons with analytical solution and previous works by FEM, FDM and RBFs are made.

The accuracy of the numerical results are measured in terms of absolute error E_a , relative error norm E_r , maximum error E_∞ , and root mean square error E_{rmse} defined as:

$$(4.1) \quad E_a = |u_i^a - u_i^n|, \quad (1 \leq i \leq N),$$

$$(4.2) \quad E_r = \left[\frac{\sum_{i=1}^N (u_i^a - u_i^n)^2}{\sum_{i=1}^N (u_i^a)^2} \right]^{1/2},$$

$$(4.3) \quad E_\infty = \|u^a - u^n\|_\infty = \max |u_i^a - u_i^n|, \quad (1 \leq i \leq N),$$

$$(4.4) \quad E_{rmse} = \|u^a - u^n\|_2 = \left[\frac{\sum_{i=1}^N (u_i^a - u_i^n)^2}{N} \right]^{1/2},$$

where u_i^a and u_i^n are analytical and numerical solutions at the i -th grid point.

Example 1. Consider the one-dimensional Burgers' equation Eqs. (3.1–3.3) with $a = 0, b = 1, \phi(x) = \sin(\pi x)$, and $\psi_1(t) = \psi_2(t) = 0$:

$$(4.5) \quad u(x, 0) = \sin(\pi x), \quad x \in [0, 1],$$

$$(4.6) \quad u(0, t) = u(1, t) = 0, \quad t > 0.$$

The analytical solution provided by Cole [18] is

$$(4.7) \quad u(x, t) = \frac{2\pi}{R} \frac{\sum_{n=1}^{\infty} n a_n \exp(-n^2 \pi^2 t / R) \sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n \exp(-n^2 \pi^2 t / R) \cos(n\pi x)},$$

where

$$(4.8) \quad a_0 = \int_0^1 \exp\{-(2\pi/R)^{-1}[1 - \cos(n\pi x)]\} dx,$$

$$a_n = 2 \int_0^1 \exp\{-(2\pi/R)^{-1}[1 - \cos(\pi x)]\} \cos(n\pi x) dx \quad (n = 1, 2, 3, \dots).$$

The FIM approximation for the solution u is then obtained by solving iteratively the linear system (3.10) in which $\psi_1(t) = \psi_2(t) = 0, u^0(x) = \sin \pi x$. In the computation, we choose $\delta t = 0.0001, h = 0.0125$. We display in Fig. 3 the profiles of u at different times with a wide range of Reynolds number R . It can be observed that when $R > 100$ a sharp wave front developed near $x = 1$ at time $t = 0.5$ and afterwards it starts to decay. The comparison of numerical results with FDM[19], FEM[4] and FIM for $R = 10$ and 100 is shown in Table 1. It is evident that the improved FIM gives a better approximation when both methods using the same value δt and h .

In order to better capture the moving shock wave when Reynolds number R is large, Caldwell et al. [3] adopted the technique of altering the size of elements at each stage to ensure more elements located closer to the peak. Hon and Zhao [5] developed an algorithm based on using meshless MQ method with a technique ‘chasing the peak’ to capture the shock wave. In this paper, we develop an adaptive FIM to locate the peak at each iterative time step and redistribute the grid points evenly on both sides of the peak. Precisely, more points will be located in the region with large gradient where the moving shock wave occurs.

TABLE 1. Comparison of numerical and analytical solutions of Example 1 for $R = 100$.

x	t	u^a	FEM		FDM		FIM	
			[4]	E_a	[18]	E_a	u^n	E_a
0.25	0.4	0.34191	0.34819	6.28e-03	0.34244	5.30e-04	0.34183	8.00e-05
	0.6	0.26896	0.27536	6.40e-03	0.26905	9.00e-05	0.26891	5.00e-05
	0.8	0.22148	0.22752	6.04e-03	0.22145	3.00e-05	0.22145	3.00e-05
	1.0	0.18819	0.19375	5.56e-03	0.18813	6.00e-05	0.18817	2.00e-05
	3.0	0.07511	0.07754	2.43e-03	0.07509	2.00e-05	0.07511	0
0.5	0.4	0.66071	0.66543	4.72e-03	0.67152	1.08e-02	0.66054	1.70e-04
	0.6	0.52942	0.53525	5.83e-03	0.53406	4.64e-03	0.52931	1.10e-04
	0.8	0.43914	0.44526	6.12e-03	0.44143	2.29e-03	0.43906	8.00e-05
	1.0	0.37442	0.38047	6.05e-03	0.37568	1.26e-03	0.37437	5.00e-05
	3.0	0.15018	0.15362	3.44e-03	0.15020	2.00e-05	0.15017	1.00e-05
0.75	0.4	0.91026	0.91201	1.75e-03	0.94675	3.65e-02	0.90998	2.80e-04
	0.6	0.76724	0.77132	4.08e-03	0.78474	1.75e-02	0.76705	1.90e-04
	0.8	0.64740	0.65254	5.14e-03	0.65659	9.19e-03	0.64727	1.30e-04
	1.0	0.55605	0.56157	5.52e-03	0.56135	5.30e-03	0.55596	9.00e-05
	3.0	0.22481	0.22874	3.92e-03	0.22502	2.10e-04	0.22483	2.00e-05

In this adaptive finite integration method, the integration matrix $\mathbf{A}^{(1)}$ is modified to:

$$(4.9) \quad \mathbf{A}^{(1)} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \frac{h_1}{2} & \frac{h_1}{2} & 0 & \cdots & 0 \\ \frac{h_1}{2} & \frac{h_1+h_2}{2} & \frac{h_2}{2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{h_1}{2} & \frac{h_1+h_2}{2} & \frac{h_2+h_3}{2} & \cdots & \frac{h_{N-1}}{2} \end{bmatrix}_{N \times N},$$

where h_i ($i = 1, \dots, N-1$) is the size of the i -th sub-interval. Assume x^* is the point closest to the peak, N is an odd number and $h_l = \frac{X^* - x_1}{(N-1)/2}$, $h_r = \frac{X_N - x^*}{(N-1)/2}$, hence we have $h_1 = h_2 = \cdots = h_{(N-1)/2} = h_l$, $h_{(N-1)/2+1} = h_{N/2} = \cdots = h_{N-1} = h_r$.

Comparisons among FEM with moving node technique [3], multiquadric (MQ) [5], Forth-order FDM [20] and FIM for large Reynolds number $R = 10,000$ are presented in Table 2. It can be seen that FIM not only achieves most accurate but also can give stable approximation to the solution up to $x = 0.9999$. Except FEM with moving node technique all the methods failed to produce approximation of the solution beyond $x = 0.95$. It is clear that the approximation by FEM, however, is not accurate in comparing with the FIM.

TABLE 2. Results of different methods of Example 1 for $R = 10,000$

	Christie accurate solution	FDM forth-order $\delta t = 0.01, N = 180$	MQ $\delta t = 0.001$ $N = 10$	FIM $\delta t = 0.001$ $N = 400$	FEM moving node $\delta t = 0.001^a$	FIM(adaptive) $\delta t = 0.001$ $N = 130$
0.0556	0.0422	0.0379	0.0424	0.0421	0.0422	0.0421
0.1111	0.0843	0.0834	0.0843	0.0843	0.0844	0.0842
0.1667	0.1263	0.1213	0.1263	0.1263	0.1266	0.1263
0.2222	0.1684	0.1667	0.1684	0.1684	0.1687	0.1684
0.2778	0.2103	0.2044	0.2103	0.2103	0.2108	0.2104
0.3333	0.2522	0.2469	0.2522	0.2522	0.2527	0.2522
0.3889	0.2939	0.2872	0.2939	0.2939	0.2946	0.2938
0.4444	0.3355	0.3322	0.3355	0.3355	0.3362	0.3354
0.5000	0.3769	0.3769	0.3769	0.3769	0.3778	0.3769
0.5556	0.4182	0.4140	0.4182	0.4182	0.4191	0.4183
0.6111	0.4592	0.4584	0.4592	0.4592	0.4601	0.4592
0.6667	0.5000	0.4951	0.4999	0.4999	0.5009	0.4998
0.7222	0.5404	0.5388	0.5404	0.5404	0.5414	0.5401
0.7778	0.5806	0.5749	0.5805	0.5805	0.5816	0.5804
0.8333	0.6203	0.6179	0.6201	0.6202	0.6213	0.6203
0.8889	0.6596	0.6533	0.6600	0.6594	0.6605	0.6596
0.9444	0.6983	0.6952	0.6957	0.6967	0.6992	0.6981
0.952820	-	-	-	-	0.7049	0.7043
0.983549	-	-	-	-	0.7260	0.7265
0.993873	-	-	-	-	0.7330	0.7295
0.999155	-	-	-	-	0.7335	0.7297
0.999386	-	-	-	-	0.7208	0.7168
0.999550	-	-	-	-	0.6850	0.6808
0.999708	-	-	-	-	0.5837	0.5789
0.999819	-	-	-	-	0.4299	0.4258
0.999906	-	-	-	-	0.2461	0.2431

a:with 16 intervals.

Fig. 5 shows the enlarged profile of waves in the vicinity of the right boundary at various times and the distribution of grid points near the peak at a specific time. It clearly shows that the improved FIM gives robust results even in the very vicinity of the right margin with extraordinary high gradient. It makes this scheme very useful when it comes to deal with problem shock wave.

Example 2. Consider the two-dimensional coupled Burgers' equation Eq. (3.11) with analytical solution in computational domain $\Omega = [0, 1] \times [0, 1]$ obtained using a Hopf-Cole transformation [21]:

$$(4.10) \quad u(x, y, t) = \frac{3}{4} - \frac{1}{4(1 + e^{R(-t-4x+4y)/32})},$$

$$(4.11) \quad v(x, y, t) = \frac{3}{4} + \frac{1}{4(1 + e^{R(-t-4x+4y)/32})}.$$

The initial and boundary condition can be derived from the analytical solution. For simplicity, only component u is considered in the following computation in which we let $\delta t = 0.01, h = 0.025$.

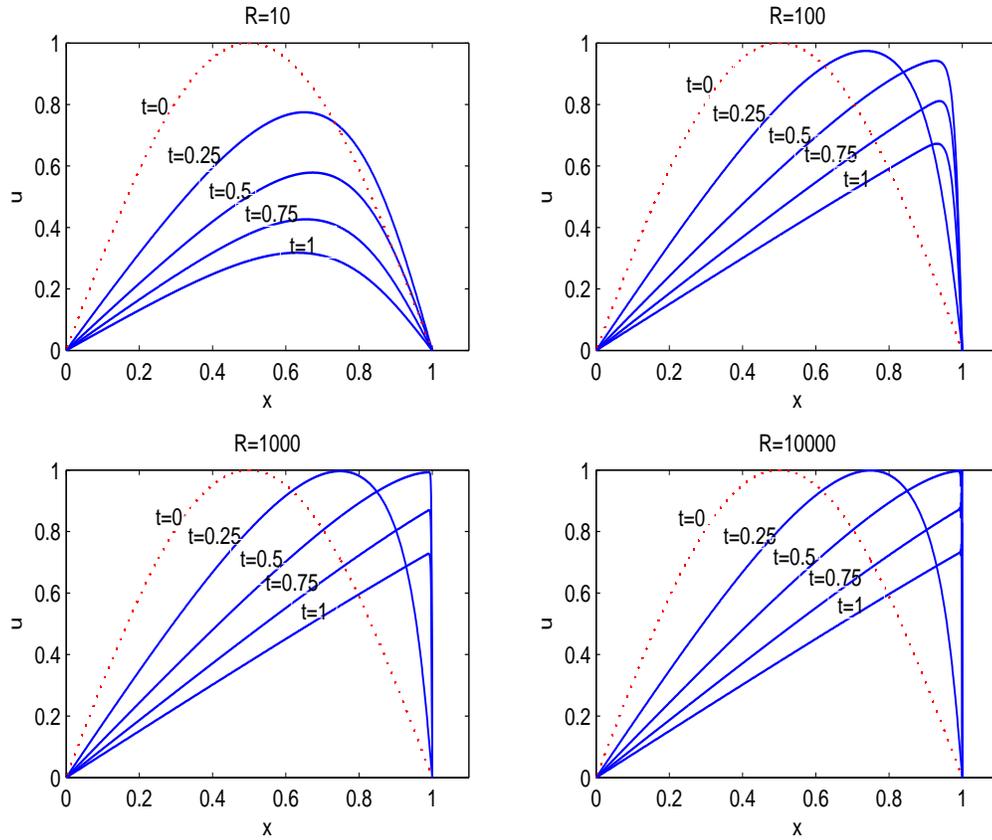


FIGURE 4. Profiles of u at different times when $R=10; 100; 1000; 10000$

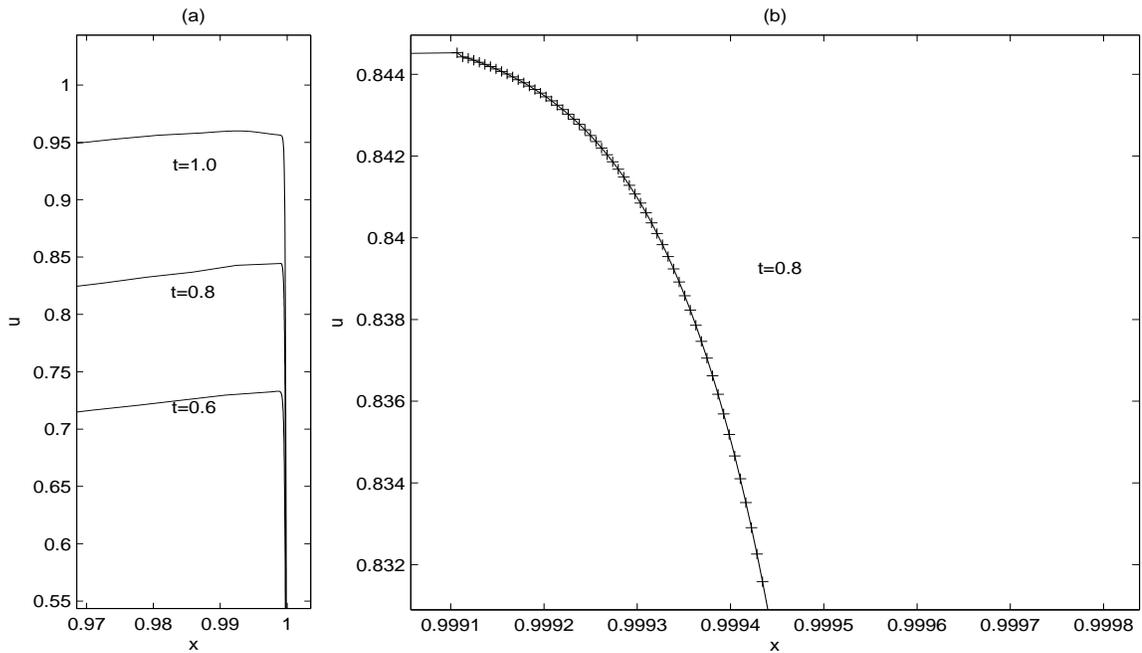


FIGURE 5. (a) Profile of u in the vicinity of $x = 1$ for $R = 10,000$; (b) Distribution of grid points near the peak at time $t = 0.8$.

Fig. 6 shows numerical solutions of u for a wide range of R from 10 to 1000 whereas Fig. 7 presents the absolute error between numerical and analytical solution for different value of R with the same 1601 grid points. It can be observed from these figures that the improved FIM can simulate well the moving shock wave for higher dimensional Burgers' equation with large Reynolds number. To further demonstrate the accuracy of the FIM, we present these errors E_∞ , E_{rmse} and E_r of u in Table 3.

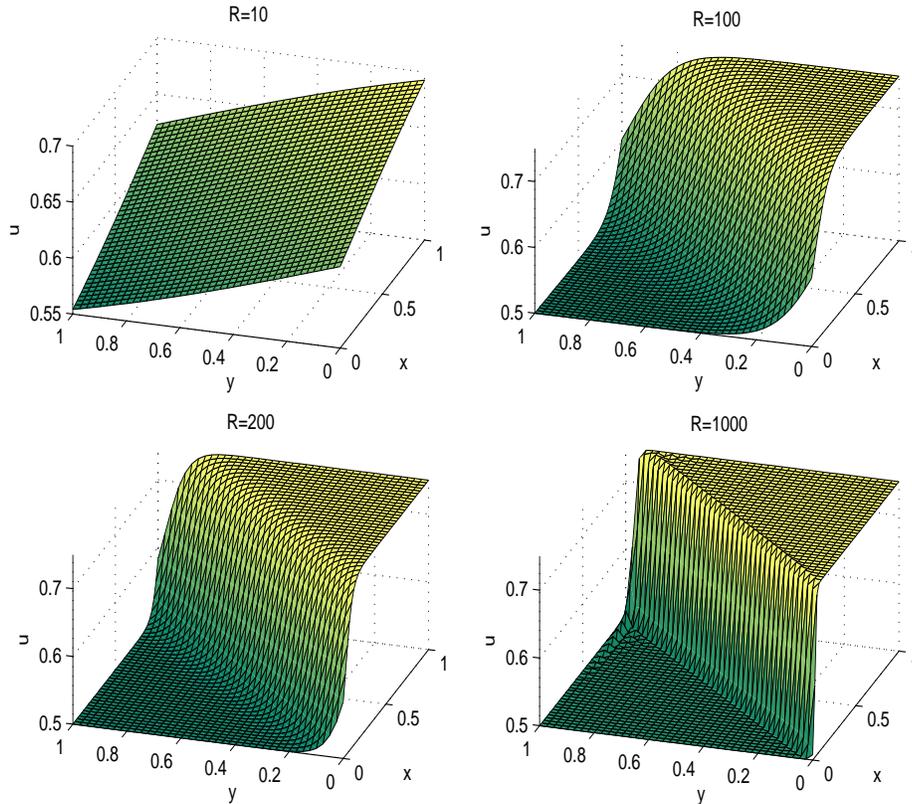
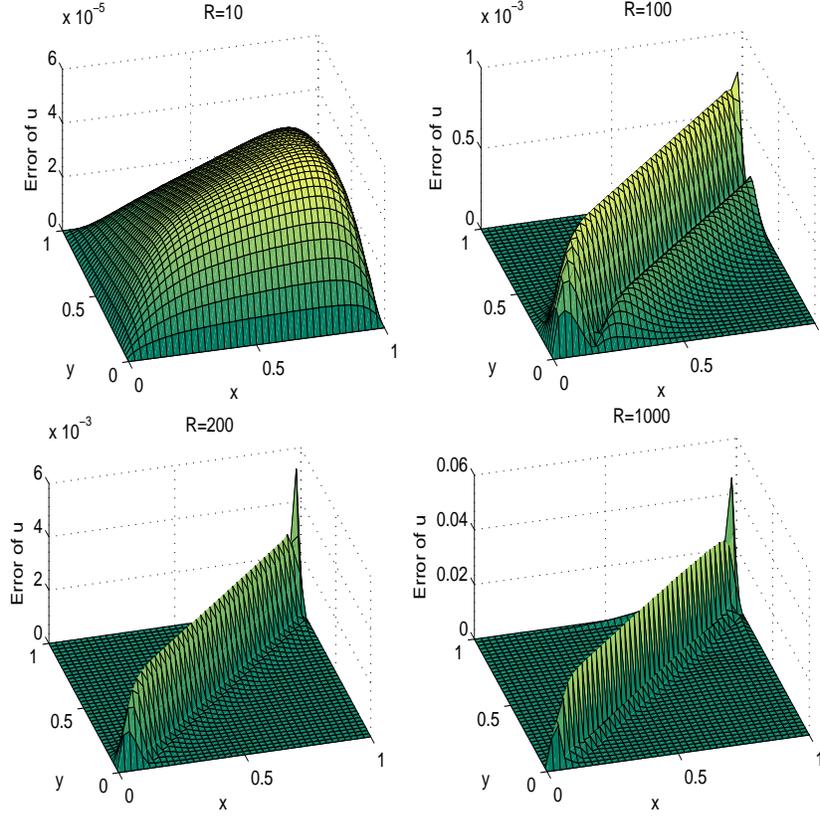


FIGURE 6. Numerical solutions of u for various R at $t = 0.1$.

TABLE 3. Errors of u with $h = 0.025$ at $t = 0.1$.

Reynolds number	E_∞	E_{rmse}	E_r
10	5.07e-05	2.98e-05	2.13e-04
100	7.74e-04	2.41e-04	3.84e-04
200	5.40e-03	7.98e-04	1.30e-03
1000	4.72e-02	5.50e-03	8.70e-03

Detailed numerical solutions and associated absolute errors at some specified points for $R = 100$ at different times are displayed in Table 4. It is evident that the proposed FIM is stable and accurate. For comparison, Table 5 presents numerical results calculated by FDM [7]. The improved FIM can obtain a almost similar accuracy while using spatial step only half of that by FDM.

FIGURE 7. Absolute error of u for various R at $t = 0.1$.TABLE 4. Numerical solutions and errors of u at some specified points for $R = 100$.

(x, y)	$t = 0.1$			$t = 0.75$			$t = 1.25$		
	u^a	u^n	E_a	u^a	u^n	E_a	u^a	u^n	E_a
(0.1, 0.1)	0.50119	0.50119	5.45e-06	0.50016	0.50016	1.09e-06	0.50003	0.50003	2.27e-07
(0.9, 0.1)	0.50001	0.50001	1.58e-09	0.50000	0.50000	1.65e-08	0.50000	0.50000	2.21e-09
(0.3, 0.3)	0.60373	0.60362	1.05e-04	0.52189	0.52217	2.82e-04	0.50493	0.50502	8.62e-05
(0.7, 0.3)	0.50119	0.50120	6.45e-06	0.50016	0.50017	3.29e-06	0.50003	0.50004	7.12e-07
(0.1, 0.5)	0.74765	0.74761	3.71e-05	0.73360	0.73321	3.85e-04	0.68727	0.68659	6.79e-04
(0.5, 0.5)	0.60373	0.60362	1.06e-04	0.52189	0.52224	3.44e-04	0.50493	0.50507	1.37e-04
(0.9, 0.5)	0.50119	0.50120	6.48e-06	0.50016	0.50017	5.15e-06	0.50003	0.50004	1.14e-06
(0.3, 0.7)	0.74765	0.74761	4.12e-05	0.73360	0.73259	1.01e-03	0.68727	0.68546	1.81e-03
(0.7, 0.7)	0.60373	0.60362	1.06e-04	0.52189	0.52226	3.64e-04	0.50493	0.50510	1.66e-04
(0.1, 0.9)	0.74998	0.74998	2.44e-07	0.74988	0.74988	3.71e-06	0.74944	0.74942	1.75e-05
(0.5, 0.9)	0.74765	0.74761	4.12e-05	0.73360	0.73221	1.39e-03	0.68727	0.68447	2.80e-03
(0.9, 0.9)	0.60373	0.60362	1.06e-04	0.52189	0.52226	3.73e-04	0.50493	0.50511	1.77e-04

Sensitivity analysis: Finally, we perform a sensitivity of the FIM on the lengths of time step δt and grid size h . The sensitivity is measured by the following rates of convergence:

$$(4.12) \quad \text{Order}_{\delta t} = \frac{\log_{10}(\|u^a - u^{\delta t_i}\|_{\infty} / \|u^a - u^{\delta t_{i+1}}\|_{\infty})}{\log_{10}(\delta t_i / \delta t_{i+1})},$$

TABLE 5. Numerical solutions and errors of u at $t = 2$ at some specific points for $R = 100$ with $h = 0.05$, $\delta t = 0.01$.

(x, y)	u^a	FDM		FIM			
		$h = 0.05$	E_a	$h = 0.1$	E_a	$h = 0.05$	E_a
(0.1, 0.1)	0.50048	0.49983	6.52e-04	0.50005	4.23e-04	0.50048	3.07e-06
(0.5, 0.1)	0.50000	0.49930	7.03e-04	0.49933	6.76e-04	0.50000	6.17e-07
(0.9, 0.1)	0.50000	0.49930	7.00e-04	0.49982	1.75e-04	0.50000	1.06e-06
(0.3, 0.3)	0.50048	0.49977	7.12e-04	0.49949	9.87e-04	0.50048	9.05e-06
(0.7, 0.3)	0.50000	0.49930	7.03e-04	0.49912	8.80e-04	0.50001	1.02e-05
(0.1, 0.5)	0.55568	0.55461	1.07e-04	0.55423	7.70e-04	0.55505	4.95e-05
(0.5, 0.5)	0.50048	0.49973	7.52e-04	0.49727	3.20e-03	0.50048	1.03e-05
(0.9, 0.5)	0.50000	0.49931	6.93e-04	0.49967	3.31e-04	0.50000	2.46e-07
(0.3, 0.7)	0.55568	0.55429	1.39e-03	0.55179	3.21e-03	0.55447	5.27e-04
(0.7, 0.7)	0.50048	0.49970	7.82e-04	0.49965	8.23e-04	0.50048	8.78e-06
(0.1, 0.9)	0.74426	0.74340	8.56e-04	0.74341	7.60e-04	0.74394	2.26e-04
(0.5, 0.9)	0.55568	0.55413	1.55e-03	0.54482	1.02e-02	0.55407	9.26e-04
(0.9, 0.9)	0.50048	0.50001	4.72e-04	0.50002	4.57e-04	0.50048	6.70e-06

$$(4.13) \quad Order_h = \frac{\log_{10}(\| u^a - u^{h_i} \|_{\infty} / \| u^a - u^{h_{i+1}} \|_{\infty})}{\log_{10}(h_i/h_{i+1})},$$

where u^{h_i} and $u^{\delta t_i}$ are numerical solutions with spatial step size h_i and time step δt_i , respectively. Tables 6 and 7 shows their rates of convergence. It can be seen from these tables that the optimal convergence depends on the ratio of δt and h , which is a well known fact in using FDM.

TABLE 6. Temporal rate of convergence when $t = 1$ and 2 for $R = 100$.

δt	$t = 1$		$t = 2$	
	E_{∞}	$Order_{\delta t}$	E_{∞}	$Order_{\delta t}$
0.10	0.01852	-	0.01845	-
0.08	0.01385	1.30	0.01399	1.24
0.06	0.00923	1.41	0.00943	1.37
0.04	0.00610	1.02	0.00506	1.54

5. Conclusion

In this paper, an improved FIM is presented for solving multi-dimensional Burgers' equation with shock wave. The improved FIM uses quadrature formulas with trapezoidal rule for numerical integration at each discrete point in the computational domain. Comparing with FDM the FIM uses integration operator instead of finite

TABLE 7. Spatial rate of convergence when $\delta t = 0.04$ and 0.03 for $R = 100$.

h	$t = 1, \delta t = 0.04$		$t = 2, \delta t = 0.03$	
	E_∞	$Order_h$	E_∞	$Order_h$
0.10	0.01902	-	0.01219	-
0.08	0.01284	1.76	0.00983	0.96
0.06	0.00814	1.58	0.00643	1.48
0.04	0.00522	1.10	0.00346	1.53

quotient formula. This completely avoids the well known roundoff-discretization error optimization problem SSin FDM. This allows the improved FIM to provide an accurate and stable approximation to these kinds of stiff problems. Numerical results indicate that the convergence order of the improved FIM in using the simplest trapezoidal rule for solving partial differential equations is proportional to $\mathcal{O}(h^2)$, where h is spatial step. A higher convergence order is expected by using higher order numerical quadrature such as Simpson's and Lagrange rule. This will be our future work.

ACKNOWLEDGEMENT

The work described in this paper was partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No. CityU 101211) and the Shanxi Hundred Talent Plan.

REFERENCES

- [1] JM Burgers, A mathematical model illustrating the theory of turbulence, *Advances in applied mechanics*, 1:171–199, 1948.
- [2] BM Herbst, SW Schoombie, DF Griffiths and AR Mitchell, Generalized petrov-galerkin methods for the numerical solution of burger' equation, *International journal for numerical in engineering*, 20(7):1273–1289, 1984.
- [3] J Caldwell, P Wanless and AE Cook, Solution of burgers' equation for large reynolds number using finite elements with moving nodes, *Applied mathematical modelling*, 11(3):211–214, 1987.
- [4] S Kutluay, A Esen and I Dag, Numerical solutions of the Burgers equation by the least-squares quadratic B-spline finite element method, *Journal of Computational and Applied Mathematics*, 167(1):21–33, 2004.
- [5] YC Hon and XZ Mao, An efficient numerical scheme for burgers' equation, *Applied Mathematics and Computation*, 95(1):37-50, 1998.
- [6] Alierza Hashemian and Hossein M Shodja, A meshless approach for solution of burgers equation, *Journal of Computational and Applied Mathematics*, 220(1):226–239, 2008.
- [7] A Refik Bahadir, A fully implicit finite-difference scheme for two-dimensional burgers' equations, *Applied Mathematics and Computation*, 137(1):131–137, 2003.
- [8] PC Jain and M Raja, Splitting-up technique for burgers' equations, *Indian J. Pure Appl. Math.*, 10:1543–1551, 1979.

- [9] CAJ Fletcher, A comparison of finite element and finite difference solutions of the one- and two-dimensional burgers' equations, *Journal of Computational physics*, 51(1):159–188, 1983.
- [10] Samir F Radwan, Comparison of higher-order accurate schemes for solving the two-dimensional unsteady burgers' equation, *Journal of Computational and Applied Mathematics*, 174(2):383–397, 2005.
- [11] Kazuhio Kakuda and Nobuyoshi Tosaka, The generalized boundary element approach to burgers' equation, *International journal for numerical methods in engineering*, 29(2):245–261, 1990.
- [12] Eiichi Chino and Nobuyoshi Tosaka, Dual reciprocity boundary element analysis of time-independent burger's equation, *Engineering analysis with boundary elements*, 21(3):261–270, 1998.
- [13] Jichun Li, YC Hon and CS Chen, Numerical comparison of two meshless methods using radial basis function, *Engineering Analysis with boundary elements*, 26(3):205–225, 2002.
- [14] DL Young, CM Fan, SP Hu and SN Atluri, The eulerian-lagrangian method of fundamental solution for two-dimensional unsteady burgers equations, *Engineering analysis with boundary elements*, 32(5):395–412, 2008.
- [15] PH Wen, YC Hon, M Li and T Korakianitis, Finite integration method for partial differential equations, *Appl. Math. Model.*, 37(24):10092–101106, 2013.
- [16] M Li, YC Hon, T Korakianitis and PH Wen, Finite integration method for nonlocal elastic bar under static and dynamic loads, *Engineering Analysis with Boundary elements*, 37(5):842–849, 2013.
- [17] SR Kuo, JT Chen and CX Huang, Analytical study and numerical experiments for true and spurious eigen-solutions of a circulant cavity using the real-part dual bem, *International Journal for Numerical Methods in Engineering*, 48(9):1401–1422, 2000.
- [18] J.D. Cole, On a quasi-linear parabolic equations occurring in aerodynamics, *Quart. Appl. Math.*, 9:225–236, 1951.
- [19] S Kutluay, AR Bahadir and A Özdes, Numerical solution of one-dimensional burgers equation: explicit and exact-explicit finite difference methods, *Journal of Computational and Applied Mathematics*, 103(2):251–261, 1999.
- [20] IA Hassanien, AA Salama and HA Hosham, Fourth-order finite difference method for solving burgers equatixon, *Applied Mathematical and Computation*, 170(2):781–800, 2005.
- [21] Clive AJ Fletcher, Generating exact solutions of the two-dimensional burgers' equation, *International Journal for Numerical Methods in Fluids*, 3(3):213–216, 1983.