

Numerical solution of Burgers' equation by B-spline collocation

M. Yousefi¹ · J. Rashidinia¹ · M. Yousefi² · M. Moudi³

Received: 25 December 2014 / Accepted: 26 February 2016 / Published online: 21 March 2016
© African Mathematical Union and Springer-Verlag Berlin Heidelberg 2016

Abstract In this paper, the Burgers' equation which is two-dimensional in space, time dependent parabolic differential equation was solved by b-spline collocation algorithms for solving two-dimensional parabolic partial differential equation. At first b-spline interpolation is introduced moreover, the numerical solution is represented as a bi-variate piecewise polynomial with unknown time-dependent coefficients are determined by requiring the numerical solution to satisfy the PDE at a number of points within the spatial domain i.e. we collocate simultaneously in both spatial dimensions. The accuracy of the proposed method is demonstrated by some test problems. The numerical results are found good agreement with exact solution.

Keywords Burgers' equation · B-spline Collocation · Tensor product · Gaussian points

Mathematics Subject Classification 35R20 · 35N20 · 35K51 · 35N10 · 58J35

1 Introduction

Consider the two-dimensional Burgers equation

$$\frac{\partial u}{\partial t} + \mu \frac{\partial u}{\partial x} + \mu \frac{\partial u}{\partial y} = \xi \frac{\partial^2 u}{\partial x^2} + \xi \frac{\partial^2 u}{\partial y^2} \quad (x, y, t) \in \Omega \times (t_0, t_{out}] \quad (1)$$

where $\Omega = \{(x, y) | a < x < b, c < y < d\}$

✉ M. Yousefi
mah_yousefi@yahoo.com

¹ School of Mathematics, Iran University of Science and Technology, Narmak Tehran 16844-13114, Iran

² Energy and Mechanical Engineering Department, Abbaspour-Power and Water, College of Engineering, Shahid Beheshti University, P.O. Box.16765-1719, Tehran, Iran

³ Institute of Applied Materials-Reliability of Components and Systems (IAM-ZBS), Engelbert-Arnold-Straße, 476131 Karlsruhe, Germany

With exact solution

$$u(x, y, t) = \frac{1}{1 + e^{\frac{x+y-t}{2\xi}}}, \quad (2)$$

Subject to the initial condition

$$\begin{aligned} u(x, y, t_0) &= u_0(x, y), \\ (x, y, t) &\in \Omega \times (t_0, t_{out}] \end{aligned} \quad (3)$$

The Burgers equation is a fundamental pde from fluid mechanics. It occurs in various areas of applied mathematics such as modeling of gas dynamics and traffic flow.

In 1915, Bateman [1] introduced the one-dimensional Burgers equation and mentioned. The steady solution is worth of the study. It was later treated by Burger [2] as a mathematical model for turbulence and after whom such an equation is widely referred to as Burgers' equation many researchers have used various numerical methods to solve the Burgers equation [3–6]. In these methods fall into the following classes: finite difference method [7, 8], finite volume method [9], finite element method [10], boundary element method [11], and etc.

In this paper, we discuss the development of two numerical algorithms based on B-spline collocation for solving the Burgers' equation. We consider B-spline collocation in a tensor product framework to discretize both spatial domains. We presented an algorithm that uses a fast block LU scheme with modified alternate row and column elimination.

R. D. Russell and his colleagues [13] suggested a fast algorithm upon a matrix block eigenvalue decomposition to solve spline collocation matrices. In this research, the matrices that arise with this method have two important advantages:

A similar fast algorithm based upon LU decomposition with amendment.

Alternate row and column elimination with partial pivoting to take advantage of the structure of matrices that arise and also it was used just one collocation point per subinterval, the linear system that arise are the smallest among all type of piecewise polynomial collocation method for this problem.

2 Collocation method

We consider a 2D rectangular grid based on a mesh of $N + 1$ points ($N > 1$) in $[a, b]$ and a mesh of $M + 1$ points ($M > 1$) in $[c, d]$ such that

$$a = x_0 < x_1 < \dots < x_N = b, \quad c = y_0 < y_1 < \dots < y_M = d \quad (4)$$

We associate with the mesh on the x domain, C^1 -continuous piecewise polynomials of degree p , i.e., we have a polynomial of degree p for the i th subinterval, $[x_{i-1}, x_i]$, $i = 1, \dots, N$, with C^1 -continuity imposed at the internal mesh points. Consequently the dimension of this piecewise polynomial subspace is $KF = K(p - 1) + 2$. Similarly, in the y domain, we have a polynomial of degree q for the i th subinterval, $[y_{i-1}, y_i]$, $i = 1, \dots, M$, with C^1 -continuity imposed at the internal mesh points. Consequently the dimension of this piecewise polynomial subspace is $LF = L(q - 1) + 2$.

To represent the piecewise polynomials, we employ B-spline bases. Let $\{\rho_i\}_{i=1}^N$, $\{\eta_j\}_{j=1}^M$ be the B-splines bases associated with the above meshes. Thus an approximation $U_N(x, y, t)$ to the exact solution $U(x, y, t)$ can be expressed as a linear combination of the tensor product

of the B-spline bases functions in x and y with time-dependent coefficients, $\delta_{ij}(t)$ as follows:

$$U_N(x, y, t) = \sum_{i=1}^M \sum_{j=1}^N \delta_{ij}(t) \rho_i(x) \eta_j(y) \tag{5}$$

We define $\{\alpha_i\}_{i=1}^{p-1}$ and $\{\beta_j\}_{j=1}^{q-1}$ to be the canonical Gaussian points on $[0,1]$ with $0 < \alpha_1 < \dots < \alpha_{p-1} < 1$, and $0 < \beta_1 < \dots < \beta_{q-1} < 1$. The collocation points in the x domain are then defined by

$$\begin{aligned} \lambda_1 &= a \\ \lambda_t &= x_{i-1} + h_i \alpha_j, t = 1 + (i - 1) \cdot (p - 1) + j, \quad i = 1, \dots, N, j = 1, \dots, p - 1 \\ \lambda_{NC} &= b \end{aligned} \tag{6}$$

where $h_i = x_i - x_{i-1}$, $i = 1, \dots, N$
 The collocation points in the y domain are defined to be

$$\begin{aligned} \omega_1 &= c \\ \omega_t &= x_{i-1} + k_i \beta_j, t = 1 + (i - 1) \cdot (q - 1) + j, \quad i = 1, \dots, M, j = 1, \dots, q - 1 \\ \omega_{MC} &= d \end{aligned} \tag{7}$$

where $k_i = y_i - y_{i-1}$, $i = 1, \dots, M$

The Burgers' equation is discretized over the spatial domain by simultaneously collocating at the points $\{\lambda_i\}_{i=1}^{KF-1}$ in x and the points $\{\omega_j\}_{j=1}^{LF-1}$ in y . The collocation conditions yield the following ODEs in time:

$$u_t(\lambda_i, \omega_j, t) + \mu u_x(\lambda_i, \omega_j, t) + \mu \frac{\partial u}{\partial y}(\lambda_i, \omega_j, t) = \xi u_{xx}(\lambda_i, \omega_j, t) + \xi u_{yy}(\lambda_i, \omega_j, t) \tag{8}$$

where $i = 2, \dots, KF - 1$, and $j = 2, \dots, LF - 1$.

The boundary conditions are

$$u(a, y, t) - \frac{1}{1 + e^{\frac{a+y-t}{2\xi}}} = 0, \quad t \in (t_0, t_{out}), y \in (c, d), \quad (\lambda_1 = a) \tag{9}$$

$$u(b, y, t) - \frac{1}{1 + e^{\frac{b+y-t}{2\xi}}} = 0, \quad t \in (t_0, t_{out}), y \in (c, d), \quad (\lambda_{KF} = b) \tag{10}$$

$$u(x, c, t) - \frac{1}{1 + e^{\frac{x+c-t}{2\xi}}} = 0, \quad t \in (t_0, t_{out}), x \in (a, b), \quad (\omega_1 = c) \tag{11}$$

$$u(x, d, t) - \frac{1}{1 + e^{\frac{d+x-t}{2\xi}}} = 0, \quad t \in (t_0, t_{out}), x \in (a, b). \quad (\omega_{LF} = c) \tag{12}$$

By substituting (5) into (8)–(11), we get equations in the terms of the unknowns $p_{ij}(t)$. We can then rewrite these equations in matrix Form:

$$G(\bar{P}(t)) = \bar{F}(t, \bar{p}(t)) \tag{13}$$

In (13), $\bar{p}(t)$ is the B-spline coefficient vector, it has the form,

$$P(t) = \begin{bmatrix} \bar{p}_1(t) \\ \bar{p}_2(t) \\ \vdots \\ \bar{p}_{KF}(t) \end{bmatrix}, \quad \text{where } \bar{P}_i(t) = \begin{bmatrix} p_{i1}(t) \\ p_{i2}(t) \\ \vdots \\ p_{iKF}(t) \end{bmatrix} \tag{14}$$

The right hand side vector

$$\bar{F}(t, \bar{P}(t)) = \begin{bmatrix} F_1(t, \bar{P}(t)) \\ F_2(t, \bar{P}(t)) \\ \vdots \\ F_{KF}(t, \bar{P}(t)) \end{bmatrix} \tag{15}$$

where each $F_i(t, \bar{P}(t))$ has LF components. The expressions for $F_i(t, \bar{P}(t)), i = 1, \dots, KF$, The matrix G in, can be written as

$$G = G_x \otimes G_y \tag{16}$$

3 2D B-spline

The B-spline coefficients characterize the projection of the approximate solution onto the B-spline tensor product basis. We can rewrite this 2D B-spline projection in matrix form as:

$$\bar{U}(\bar{\lambda}, \bar{\omega}, t) = (N_x \otimes N_y) \bar{P}(t) = Q \bar{P}(t) \tag{17}$$

where $\bar{U}(\bar{\lambda}, \bar{\omega}, t)$ is the evaluation of $U(x, y, t)$ and

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{KF} \end{bmatrix} \otimes e_{LF}, \quad \omega = e_{KF} \otimes \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{LF} \end{bmatrix} \tag{18}$$

where e_{LF} is the vector of 1s of length LF , e_{KF} is the vector of 1s of length KF .

A fast block matrix system solution algorithm

The algorithm we use is as follows. We assume that N_x and N_y are :

$$\begin{aligned} N_x &= L_x U_x \\ N_y &= L_y U_y \end{aligned} \tag{19}$$

where L_x, L_y are lower triangular matrices, and U_x, U_y are upper triangular matrices. Let us simplify the notation by writing (4.12) as

$$(N_x \otimes N_y) p = c \tag{20}$$

where $p = P(t)$ and $c = \bar{U}(\lambda, \omega, t)$. The above system can then be rewritten as

$$((L_x U_x) \otimes (L_y U_y)) p = c \tag{21}$$

Based on a property of the Kronecker product, we can rewrite the above system as

$$((L_x \otimes L_y)(U_x \otimes U_y)) p = c \tag{22}$$

This system can then be solved in 4 steps:

- Step 1: Solve $(L_x \otimes I_{LF}) \tilde{v} = c$ for \tilde{v}
- Step 2: Then solve $(I_{KF} \otimes L_y) v = \tilde{v}$ for v
- Step 3: Then solve $(U_x \otimes I_{LF}) \tilde{w} = v$ for \tilde{w}

Step 4: Then solve $(I_{KF} \otimes U_y)w = \tilde{w}$ for w

where I_{LF} is the $LF \times LF$ identity matrix and I_{KF} is the $KF \times KF$ identity matrix. Based on the analysis given in [12]. The cost for step 1 is $O(N^2 p^3)$, the cost of step 2 is $O(N^2 p^3)$, the cost of step 3 $O(N^2 p^3)$ is and the cost of step 4 is $O(N^2 p^3)$. Thus the total cost of for the algorithm is $O(N^2 p^3)$.

If we were to solve (16) simply by the coefficient matrix as a blocks of size $O(p(LF \times LF)) = O(NP^2 \times NP^2)$, the cost of this almost block diagonal with block of this size is $O(N(NP^2)^3) = O(N^4 P^6)$ thus by using this algorithm, we have substantial saving. steps (1) and (2) solve the system

$$(L_x \otimes L_y)v = c \tag{23}$$

for v .

substitute from step (2) into step (1); we have

$$(L_x \otimes I_{LF})(I_{KF} \otimes L_y)v = c \tag{24}$$

Then we can rewrite this equation by using the property of the Kronker product as

$$(L_x \otimes L_y)v = c \tag{25}$$

A similar argument can be used to show that steps (3) and (4) solve the linear System

$$(U_x \otimes U_y)w = v \tag{26}$$

At certain points in the algorithm, we know the B-spline coefficients, $\bar{P}(t)$ and We should evaluate the approximate solution at the collocation points $\bar{U}(\lambda, \omega, t)$. That is, we compute $(N_x \otimes N_y)p = c$, where we know N_x, N_y and p . We compute c with using a matrix multiplication.

4 Numerical implementation of algorithm

We consider the 2D Burgers' equation (1) the problem domain is $(x, y) \in (0, 1) \times (0, 1), t > 0$
The exact solution is

$$u(x, y, t) = \frac{1}{1 + e^{\frac{x+y-t}{2\xi}}}$$

We set $\xi = 0.01$, and the boundary conditions is

$$\begin{aligned} u(0, y, t) - \frac{1}{1 + e^{\frac{y-t}{2\xi}}} &= 0, \quad t \in (0, 1), y \in (0, 1) \\ u(1, y, t) - \frac{1}{1 + e^{\frac{1+y-t}{2\xi}}} &= 0, \quad t \in (0, 1), y \in (0, 1) \\ u(x, 0, t) - \frac{1}{1 + e^{\frac{x-t}{2\xi}}} &= 0, \quad t \in (0, 1), x \in (0, 1) \\ u(x, 1, t) - \frac{1}{1 + e^{\frac{x+1-t}{2\xi}}} &= 0, \quad t \in (0, 1), x \in (0, 1) \end{aligned}$$

and initial condition is

$$u(x, 0) = \frac{1}{1 + e^{\frac{x+y}{2\xi}}}, (x, y) \in [0, 1] \times [0, 1]$$

Table 1 The numerical result for (1) equation and corresponding approximate convergence rate

NCOLL = 3				
NUM	TERR	RERR	RATE	
4	10.01×10^{-2}	–	–	
8	3.66×10^{-3}	27.02	4.81	
16	1.06×10^{-4}	31.32	4.89	
32	4.29×10^{-6}	26.78	4.65	
64	6.69×10^{-7}	32.41	4.96	

Table 2 The numerical result for (1) equation and corresponding approximate convergence rate

NCOLL = 4				
NUM	TERR	RERR	RATE	
4	4.06×10^{-3}	–	–	
8	5.74×10^{-4}	50.43	5.36	
16	1.09×10^{-5}	52.56	5.81	
32	1.89×10^{-7}	54.98	5.78	
64	2.05×10^{-8}	55.53	5.86	

The following notation will be used in numerical result.

NCOLL : The number of collocation points per subinterval

NUM : The number of subintervals

AERR : The absolute error

RERR : The relative error

TERR : The true error

RATE: The rate of convergence

Conclusion

A new numerical method, which is based on 2D B-spline collocation algorithm. This leads to an approximation of Burgers' equation by a large system of time-dependent differential algebraic equation, we then solve using a high quality differential algebraic equation solver. The amount of computation and memory is substantially reduced. In Tables 1, 2, for the collocation solutions we compute at $t = 1$, the observed AERR, RERR, TERR, and corresponding approximate convergence rates. From the two tables we observe that the expected rates of convergence are indeed observed in the 2D case.

References

1. Bateman, H.: Some recent researches on the motion of fluids. *Monthly Weather Rev.* **43**, 163–170 (1915)
2. J.M. Burger, A mathematical model illustrating the theory of turbulence, in: *Adv. In App. Mech.* I, Academic Press, New York, 1948, pp. 171–199
3. Varoglu, E., Finn, W.D.: Space-time finite element incorporating characteristics for the Burgers' equation. *Int. J. Numer. Meth. Eng.* **16**, 171–184 (1980)
4. Evans, D.J., Abdullah, A.R.: The group explicit method for the solution of Burgers' equation equation. *Computing* **32**, 239–253 (1984)

5. E.L. Miller, Predictor–corrector studies of Burgers' model of turbulent Flow, M.S. Thesis, University of Delaware, Newark, DE, 1966
6. Abd-el-Malek, M.B., El-Mansi, S.M.A.: Group theoretic methods applied to Burgers' equation. *J. Comput. Appl. Math.* **115**, 1–12 (2000)
7. Gulsu, M.: A finite difference approach for solution of Burgers' equation. *Appl. Math. Comput.* **175**, 1245–1255 (2006)
8. Hassanien, I.A., Salama, A.A., Hosham, H.A.: Fourth-order finite difference method for solving Burgers' equation. *Appl. Math. Comput.* **170**, 781–800 (2005)
9. Bihari, B., Harten, A.: Multiresolution schemes for the numerical solution of 2D conservation laws. *SIAM J. Sci. Comput.* **18**, 315–345 (1997)
10. Aksan, E.N.: A numerical solution of Burgers' equation by finite element method constructed on the method of discretization in time. *Appl. Math. Comput.* **170**, 895–904 (2005)
11. Chino, E., Tosaka, N.: Dual reciprocity boundary element analysis of time-independent Burgers' equation. *Eng. Anal. Boundary Elem.* **21**, 3–47 (1996)
12. Russell, R.D., Sun, W.: Spline collocation differentiation matrices. *SIAM J. Numer. Anal.* **34**(6), 2274–2287 (1997)
13. Diaz, J.C., Fairweather, G., Keast, P.: FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate Row and column elimination. *ACM Trans. Math. Software* **9**(3), 358–375 (1983)