Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 492072, 8 pages
http://dx.doi.org/10.1155/2014/492072

Research Article

A Meshfree Quasi-Interpolation Method for Solving Burgers' Equation

Mingzhu Li,[1,2] Lijuan Chen,[1] and Qiang Ma[2]

[1] *School of Science, Qingdao Technological University, Qingdao 266033, China*
[2] *Department of Mathematics, Harbin Institute of Technology at Weihai, Weihai 264209, China*

Correspondence should be addressed to Qiang Ma; hitmaqiang@hotmail.com

Received 26 April 2014; Revised 29 June 2014; Accepted 1 July 2014; Published 16 July 2014

Academic Editor: Kim M. Liew

The main aim of this work is to consider a meshfree algorithm for solving Burgers' equation with the quartic B-spline quasi-interpolation. Quasi-interpolation is very useful in the study of approximation theory and its applications, since it can yield solutions directly without the need to solve any linear system of equations and overcome the ill-conditioning problem resulting from using the B-spline as a global interpolant. The numerical scheme is presented, by using the derivative of the quasi-interpolation to approximate the spatial derivative of the dependent variable and a low order forward difference to approximate the time derivative of the dependent variable. Compared to other numerical methods, the main advantages of our scheme are higher accuracy and lower computational complexity. Meanwhile, the algorithm is very simple and easy to implement and the numerical experiments show that it is feasible and valid.

1. Introduction

Burgers' equation plays a significant role in various fields, such as turbulence problems, heat conduction, shock waves, continuous stochastic processes, number theory, gas dynamics, and propagation of elastic waves [1–5]. The one-dimensional Burgers' equation first suggested by Bateman [6] and later treated by Burgers [1] has the form

$$U_t + UU_x - \lambda U_{xx} = 0, \tag{1}$$

where $\lambda > 0$ is the coefficient of kinematic viscosity and the subscripts x and t denote space and time derivatives. Initial and boundary conditions are

$$\begin{gathered} U(x,0) = f(x), \quad a \le x \le b, \\ U(a,t) = \beta_1, \quad u(b,t) = \beta_2, \quad t \ge 0, \end{gathered} \tag{2}$$

where β_1, β_2, and $f(x)$ will be chosen in a later section.

Burgers' equation is a quasi-linear parabolic partial differential equation, whose analytic solutions can be constructed from a linear partial differential equation by using Hopf-Cole transformation [1, 2, 7]. But some analytic solutions consist of infinite series, converging very slowly for small viscosity coefficient λ. Thus, many researchers have spent a great deal of effort to compute the solution of Burgers' equation using various numerical methods. Finite difference methods were presented to solve the numerical solution of Burgers' equation in [8–11]. Finite element methods for the solution of Burgers' equation were introduced in [12–15]. Recently, various powerful mathematical methods such as Galerkin finite element method [16, 17], spectral collocation method [18, 19], sinc differential quadrature method [20], factorized diagonal padé approximation [21], B-spline collocation method [22], and reproducing kernel function method [23] have also been used in attempting to solve the equation.

In 1968 Hardy proposed the multiquadric (MQ) which is a kind of radial basis function (RBF). In Franke's review paper, the MQ was rated as one of the best methods among 29 scattered data interpolation and ease of implementation. Since Kansa successfully applied MQ for solving partial differential equation, more and more reasearchers have been attracted

by this meshfree, scattered data approximation scheme [24]. The meshfree method uses a set of scattered nodes, instead of meshing the domain of the problem. It has been successfully applied to solve many physical and engineering problems with only a minimum of meshing or no meshing at all [25–30]. In recent years, many meshfree metheods have been developed, such as the element-free Galerkin method [31], the smooth particle hydrodynamics method [32], the element-free kp-Ritz method [33–36], the meshless local Petrov-Galerkin method [37], and the reproducing kernel particle method [38].

With the use of univariate multiquadric (MQ) quasi-interpolation, solution of Burgers' equations was obtained by Chen and Wu [39]. Moreover, Hon and Mao [40] developed an efficient numerical scheme for Burgers' equation applying the MQ as a spatial approximate scheme and a low order explicit finite difference approximation to the time derivation. Zhu and Wang [41] presented the numerical scheme for solving the Burgers' equation, by using the derivative of the cubic B-spline quasi-interpolation to approximate the time derivative of the dependent variable and a low order forward difference to approximate the time derivative of the dependent variable. In this paper, we provide a numerical scheme to solve Bugers' equation using the quartic B-spline quasi-interpolation. Then we do not require to solve any linear system of equation so that we do not meet the question of the ill-condition of the matrix.Therefore, we can solve the computational time and decrease the numerical error.

This paper is arranged as follows. In Section 2, the definition of quartic B-spline has been described and univariate quartic B-spline quasi-interpolants have been presented. In Section 3, we mainly propose the numerical techniques using quartic B-spline interpolation to solve Burgers' equation. In Section 4, numerical examples of Burgers' equation are presented and compared with those obtained with some previous results. At last, we conclude the paper in Section 5.

2. Univariate Quartic B-Spline Quasi-Interpolant

For an interval $I = [a, b]$, we introduce a set of equally-spaced knots of partition $\Omega = \{x_0, x_1, \ldots, x_n\}$. We assume that $n \geq 5$, $x_i = a + ih$ $(i = 0, 1, \ldots, n)$, $x_0 = a$, and $x_n = b$. Let $S_4[\pi]$ be the space of continuously-differentiable, piecewise, quartic-degree polynomials on π. A detailed description of B-spline functions generated by subdivision regarding the B-splines basis in $S_4[\pi]$ can be found in [45].

The zero degree B-spline is defined as

$$N_{i,0}(x) = \begin{cases} 1, & x \in [x_i, x_{i+1}], \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

and, for positive constant p, it is defined in the following recursive form:

$$N_{i,p} = \frac{x - x_i}{x_{i+p} - x_i} N_{i,p-1}(x) + \frac{x_{i+p+1} - x}{x_{i+P+1} - x_{i+1}} N_{i+1,p-1}, \quad p \geq 1. \tag{4}$$

We apply this recursion to get the quartic B-spline $N_{i,4}(x)$, which is defined in $S_4(\pi)$ as follows:

$$N_{i,4}(x) = \frac{1}{24h^4} \begin{cases} (x - x_{i-2})^4, & x \in [x_{i-2}, x_{i-1}], \\ (x - x_{i-2})^4 - 5(x - x_{i-1})^4, & x \in [x_{i-1}, x_i], \\ (x - x_{i-2})^4 - 5(x - x_{i-1})^4 + 10(x - x_i)^4, & x \in [x_i, x_{i+1}], \\ (x - x_{i+3})^4 - 5(x - x_{i+2})^4, & x \in [x_{i+1}, x_{i+2}], \\ (x - x_{i+3})^4, & x \in [x_{i+2}, x_{i+3}], \\ 0, & \text{otherwise}. \end{cases} \tag{5}$$

As usual, we add multiple knots at the endpoints: $a = x_{-4} = x_{-3} = \cdots = x_0$ and $b = x_n = x_{n+1} = \cdots = x_{n+4}$.

In [24], univariate quartic B-spline quasi-interpolants (*abbr.QIs*) can be defined as operators of the form

$$Q_4(f) = \sum_{j=1}^{n+4} \mu_j N_{j,4}. \tag{6}$$

The coefficients are listed as follows:

$$\mu_1(f) = f_1,$$
$$\mu_2(f) = \frac{17}{105} f_1 + \frac{35}{32} f_2 - \frac{35}{96} f_3 + \frac{21}{160} f_4 - \frac{5}{244} f_5,$$
$$\mu_3(f) = -\frac{19}{45} f_1 + \frac{377}{288} f_2 + \frac{61}{288} f_3 - \frac{59}{480} f_4 + \frac{7}{288} f_5,$$
$$\mu_4(f) = \frac{47}{315} f_1 - \frac{77}{144} f_2 + \frac{251}{144} f_3 - \frac{97}{240} f_4 + \frac{47}{1008} f_5,$$
$$\mu_j(f) = \frac{47}{1152}\left(f_{j-4} + f_{j+1}\right) - \frac{107}{288}\left(f_{j-3} + f_{j-1}\right) + \frac{319}{192} f_{j-2}, \quad j = 5, \ldots, n,$$
$$\mu_{n+1}(f) = \frac{47}{315} f_{n+2} - \frac{77}{144} f_{n+1} + \frac{251}{144} f_n - \frac{97}{240} f_{n-1} + \frac{47}{1008} f_{n-2},$$

$$\begin{aligned}
\mu_{n+2}(f) &= -\frac{19}{45}f_{n+2} + \frac{377}{288}f_{n+1} + \frac{61}{288}f_n - \frac{59}{480}f_{n-1} \\
&\quad + \frac{7}{288}f_{n-2}, \\
\mu_{n+3}(f) &= \frac{17}{105}f_{n+2} + \frac{35}{32}f_{n+1} - \frac{35}{96}f_n + \frac{21}{160}f_{n-1} \\
&\quad - \frac{5}{244}f_{n-2}, \\
\mu_{n+4}(f) &= f_{n+2},
\end{aligned} \tag{7}$$

and $f_i = f(t_i)$, $t_i = (1/2)(x_{i-2} + x_{i-1})$, $i = 1, \ldots, n+2$. For $f \in C^5(I)$, we have the error estimate

$$\|f - Q_4(f)\|_\infty = O\left(h^5\right). \tag{8}$$

We use $\prod_4$ to denote the space of polynomials of the total degree at most 4. In general, we impose that Q_4 is exact on the space $\prod_4$; that is, $Q_4(p) = p$ for all $p \in \prod_4$. As a consequence of this property, the approximation order of Q_4 is $O(h^5)$ on smooth functions. In this paper, the coefficient μ_j is a linear combination of discrete values of f at some points. The main advantage of QIs is that they have a direct construction without solving any system of linear equations. Moreover, they are local in the sense that values of $Q_4 f(x)$ depend only on values of f in a neighborhood of x. Finally, they have a rather small infinity norm and, therefore, are nearly optimal approximant.

Differentiating interpolation polynomials leads to the classic finite difference for the approximate computation of derivatives. Therefore, we can draw a conclusion of approximating derivatives of f by derivatives of $Q_4 f$. The general theory will be developed elsewhere. We can evaluate the value of f at x_i by $(Q_4 f)' = \sum_{j=1}^{n+4} \mu_j(f) N'_{j,4}$ and $(Q_4 f)'' = \sum_{j=1}^{n+4} \mu_j(f) N''_{j,4}$. $N'_{j,4}$ and $N''_{j,4}$ can be computed by the formula of B-spline's derivatives as follows:

$$N_{i,4}^{(k)} = \frac{4!}{(4-k)!}\sum_{j=1}^{n}\alpha_{k,j}N_{i+j,4-k}, \tag{9}$$

where

$$\begin{aligned}
\alpha_{0,0} &= 1, \\
\alpha_{k,0} &= \frac{\alpha_{k-1,0}}{x_{i+3-k} - x_i}, \\
\alpha_{k,k} &= \frac{-\alpha_{k-1,k-1}}{x_{i+5} - x_{i+k}}, \\
\alpha_{k,j} &= \frac{\alpha_{k-j,j} - \alpha_{k-1,j-1}}{x_{i+j+5-k} - x_{i+j}}.
\end{aligned} \tag{10}$$

By some trivial computations, we can obtain the value of $N_{i,4}^{(k)}$ $(k = 0, 1, 2, 3)$ at the knots, which are illustrated in Table 1. Then, we get the differential formulas for quartic B-spline QIs as

Table 1: The values of $N_{i,4}^{(k)}(x)$ at the knots.

	x_{i-1}	x_i	x_{i+1}	x_{i+2}	Otherwise
$N_{i,4}(x)$	$\frac{1}{24}$	$\frac{11}{24}$	$\frac{11}{24}$	$\frac{1}{24}$	0
$N'_{i,4}(x)$	$\frac{1}{6h}$	$\frac{3}{6h}$	$-\frac{3}{6h}$	$-\frac{1}{6h}$	0
$N''_{i,4}(x)$	$\frac{1}{2h^2}$	$-\frac{1}{2h^2}$	$-\frac{1}{2h^2}$	$-\frac{1}{2h^2}$	0
$N'''_{i,4}(x)$	$\frac{1}{h^3}$	$-\frac{3}{h^3}$	$\frac{3}{h^3}$	$-\frac{1}{h^3}$	0

$$\begin{aligned}
f' &= \sum_{j=1}^{n+4}\mu_j(f)N'_{j,4}, \\
f'' &= \sum_{j=1}^{n+4}\mu_j(f)N''_{j,4}.
\end{aligned} \tag{11}$$

3. Numerical Scheme Using the Meshfree Quasi-Interpolation

In this section, we present the numerical scheme for solving Burgers' equation based on the quartic B-spline quasi-interpolation.

Discretizing the Burgers' equation

$$U_t + UU_x - \lambda U_{xx} = 0, \tag{12}$$

in time with meshlength τ, we get

$$\frac{U_j^{k+1} - U_j^k}{\tau} + U_j^k (U_x)_j^k - \lambda (U_{xx})_j^k = 0. \tag{13}$$

We can get

$$U_j^{k+1} = U_j^k + \tau U_j^k (U_x)_j^k - \tau\lambda (U_{xx})_j^k, \tag{14}$$

where U_j^k is the approximation of the value of $U(x,t)$ at the point (x_j, t_k). Then, we can use the derivatives of the quartic B-spline quasi-interpolant $Q_4 U(x_j, t_k)$ to approximate $(U_x)_j^k$ and $(U_{xx})_j^k$. To dump the dispersion of the scheme, we define a switch function $g(x,t)$, whose values are 0 and 1 at the discrete points (x_j, t_k), as follows:

$$g\left(x_j, t_k\right) = \max\left\{0, 1 + \min\left\{0, \operatorname{sign}\left((U_x)_j^k \cdot (U_x)_l^k\right)\right\}\right\}, \tag{15}$$

where $l = j - \operatorname{sign}(U_j^k)$. Thus, the resulting numerical scheme is

$$U_j^{k+1} = U_j^k + \tau U_j^k (U_x)_j^k g\left(x_j, t_k\right) - \tau\lambda (U_{xx})_j^k. \tag{16}$$

Starting from the initial condition, we can compute the numerical solution of Burgers' equation step by step using the B-spline quasi-interpolation scheme (16) and formulas (11).

Table 2: Comparison of exact and numerical solution at $t = 0.001$.

x	$\lambda = 1$			$\lambda = 0.5$		
	Our method	Asai [42]	Exact	Our method	Asai [42]	Exact
0.1	0.653563	0.653589	0.653544	0.327870	0.327874	0.327870
0.2	1.305519	1.305611	1.305534	0.655028	0.655078	0.655069
0.3	1.949321	1.949485	1.949364	0.978449	0.978427	0.978413
0.4	2.565977	2.566103	2.565925	1.288417	1.288485	1.288463
0.5	3.110769	3.110992	3.110739	1.563014	1.563096	1.563064
0.6	3.492902	3.493222	3.492866	1.756653	1.756691	1.756642
0.7	3.549538	3.550079	3.549595	1.787184	1.787281	1.787206
0.8	3.050089	3.050702	3.050134	1.537658	1.537794	1.537694
0.9	1.816492	1.817077	1.816660	0.916795	0.916941	0.916860

Table 3: The computational results at $t = 0.5$ for $\lambda = 0.01$ with $h = 1/36$ and $\tau = 0.01$.

	BSQI [41]	MQQI [39]	QBCM I [17]	QBGM I [43]	Our method
$L_2 \times 10^3$	3.43253	5.77786	0.77033	1.92558	0.85269
$L_\infty \times 10^3$	9.26698	20.8467	3.03817	6.35489	3.79716

4. Numerical Results

To investigate the applicability of the quasi-interpolation method to Burgers' equation, four selected example problems are studied. To show the efficiency of the present method for our problem in comparison with the exact solution, we use the following norms to assess the performance of our scheme:

$$L_\infty = \max_j \left| U_j^{\text{exact}} - U_j^{\text{num}} \right|,$$
$$L_2 = \sqrt{h \sum_{j=1}^{n} \left(U_j^{\text{exact}} - U_j^{\text{num}} \right)^2}. \tag{17}$$

Example 1. Burgers' equation is solved over the region $[0, 1]$ and the initial and boundary conditions are given in Asaithambi [42]:

$$U(x, 0) = \frac{2\lambda\pi \sin \pi x}{\alpha + \cos \pi x} \quad (\alpha > 1),$$
$$U(0, t) = 0, \quad U(1, t) = 0, \quad t > 0, \tag{18}$$

and the exact solution of this problem has the following nice compact closed-form, as given by Wood [46]:

$$U(x, t) = \frac{2\lambda\pi e^{-\pi^2 \lambda t} \sin \pi x}{\alpha + e^{-\pi^2 \lambda t} \cos \pi x} \quad (\alpha > 1). \tag{19}$$

In this computational study, we set $\alpha = 2$, $h = 0.025$, $\Delta\tau = 0.0001$. The comparison of the numerical solutions obtained by the present method, at the different coefficient of kinematic viscosity λ, are presented with the solutions obtained by Asaithambi [42] and the exact solution in Table 2.

Example 2. In this example, we consider the exact solution of Burgers' equation [47]:

$$U(x, t) = \frac{\alpha + \mu + (\mu - \alpha) \exp \eta}{1 + \exp \eta}, \quad 0 \le x \le 1,\ t \ge 0, \tag{20}$$

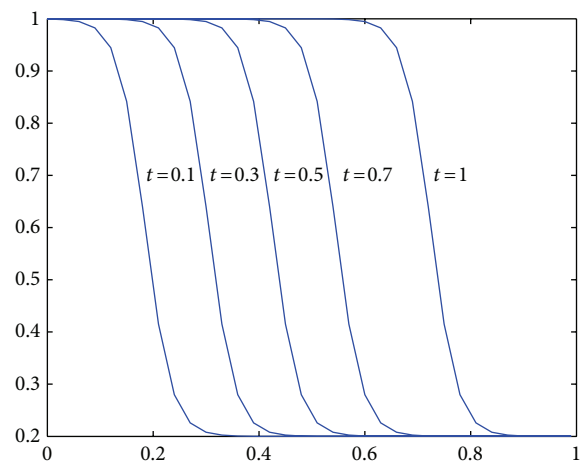

Figure 1: The numerical solutions with $h = 1/36$, $\tau = 0.01$ for $\lambda = 0.01$.

where $\eta = (\alpha(x - \mu t - \gamma))/\lambda$, α, μ, and γ are constants. The boundary conditions are

$$U(0, t) = 1, \quad U(1, t) = 0.2, \quad t \ge 0, \tag{21}$$

and initial condition is used for the exact solution at $t = 0$.

We solve the problem with $\alpha = 0.4$, $\mu = 0.6$, and $\gamma = 0.125$ by our method. In Table 3, L_2 and L_∞ errors at the time level $t = 0.5$ are compared with the error obtained by Chen and Wu [39], Zhu and Wang [41], Dağ et al. [17], and Saka and Dağ [43]. For comparison, the parameters are adopted as time step $\tau = 0.01$, space step $h = 1/36$, and viscosity coefficient $\lambda = 0.01$. From Table 3, we can find that our method provides better accuracy than most methods through the L_2 and L_∞ error norms. The profiles of initial wave and its propagation are depicted at some times in Figure 1.

Table 4: Comparison of results at different time for $\lambda = 0.1$ with $h = 0.025$ and $\tau = 0.0001$.

x	t	Hassanien [44]	Kutluay [11]	Ozis [15]	Our method	Exact
0.25	0.4	0.3089	0.3083	0.3143	0.3089	0.3089
	0.6	0.2407	0.2404	0.2437	0.2407	0.2407
	0.8	0.1957	0.1954	0.1976	0.1957	0.1957
	1.0	0.1626	0.1624	0.1639	0.1626	0.1626
	3.0	0.0272	0.0272	0.0274	0.0272	0.0272
0.50	0.4	0.5696	0.5691	0.5764	0.5696	0.5696
	0.6	0.4472	0.4468	0.4517	0.4472	0.4472
	0.8	0.3592	0.3589	0.3625	0.3592	0.3592
	1.0	0.2919	0.2916	0.2944	0.2919	0.2919
	3.0	0.0402	0.0402	0.0406	0.0402	0.0402
0.75	0.4	0.6254	0.6256	0.6259	0.6257	0.6254
	0.6	0.4872	0.4570	0.4903	0.4872	0.4872
	0.8	0.3739	0.3737	0.3771	0.3739	0.3739
	1.0	0.2875	0.2872	0.2902	0.2875	0.2875
	3.0	0.0298	0.0297	0.0133	0.0298	0.0298

Example 3. Consider Burgers' equation with the initial condition

$$U(x,0) = \sin(\pi x), \quad 0 \le x \le 1 \tag{22}$$

and the boundary conditions

$$U(0,t) = U(1,t) = 0. \tag{23}$$

The analytical solution of this problem was given by Cole [2] in the term of an infinite series as

$$U(x,t) = \frac{2\pi\lambda \sum_{k=1}^{\infty} kA_k \sin(k\pi x) \exp\left(-k^2\pi^2\lambda t\right)}{A_0 + \sum_{k=1}^{\infty} A_k \cos(k\pi x) \exp\left(-k^2\pi^2\lambda t\right)} \tag{24}$$

with the Fourier coefficients

$$A_0 = \int_0^1 \exp\left\{-(2\pi\lambda)^{-1}(1-\cos(\pi x))\right\} dx,$$

$$A_k = 2\int_0^1 \exp\left\{-(2\pi\lambda)^{-1}(1-\cos(\pi x))\right\} \cos(k\pi x)\, dx, \quad k \ge 1. \tag{25}$$

In Table 4, we have computed the numerical solutions of this example at differential time levels with parameter values $\lambda = 0.1$, $h = 0.025$, and $\tau = 0.0001$. The comparison of our results with the exact solutions as well as the solutions obtained in [11, 15, 44] is reported in Table 4. From Table 4, we can find that the presented scheme provides better accuracy. Moreover, in Tables 5, 6 and 7, we compare our method with Hon and Mao's scheme, Chen and Wu's MQQI method, and Zhu's BSQI method at $t = 1$ with $\tau = 0.001$, $h = 0.01$ for $\lambda = 0.1, 0.01, 0.0001$, respectively. For the MQQI method, the shape parameter $c = 7.2\times10^{-3}, 2.9\times10^{-3}, 1.43\times10^{-4}$ for Table 5, respectively, as [39]. Solutions found with the present method are in good agreement with the result and better than other methods. These show that the method works well.

Example 4. We consider particular solution of Burgers' equation:

$$U(x,t) = \frac{x/t}{1+\sqrt{t/t_0}\exp\left(x^2/4\lambda t\right)}, \quad t \ge 1,\ 0 \le x \le 1, \tag{26}$$

where $t_0 = \exp(1/8\lambda)$. Initial condition is obtained from when $t = 1$ is used. Boundary conditions are $U(0,t) = U(1.2,t) = 0$. Analytical solution represents shock-like solution of the one-dimensional Burgers' equation. Parameters $h = 0.02, 0.005$ and $\lambda = 0.005, 0.01$ are selected for comparison over the domain $[0,1]$. Accuracy of our method is shown by calculating the error norms. These together with some previous results are given in Table 8. Table 8 shows that our method provides better accuracy than MQQI method and BSQI method. Although the accuracy is not higher than that of QBCM method, we know that, at each time step, the complexity of our method is lower than theirs. The numerical solutions are depicted with $h = 0.02$, $\tau = 0.001$, and $\lambda = 0.005$ for $t \le 4$ in Figure 2.

5. Conclusion

Following the recent development of the quasi-interpolation method for scattered data interpolation and the meshfree method for solving partial differential equations, this paper combines these ideas and proposes a new meshfree quasi-interpolation method for Burgers' equation. The method does not require solving a large size matrix equation and, hence, the ill-conditioning problem from using B-spline functions as global interpolants can be avoided. We have made comparison studies between the present results and the exact solutions. The agreement of our numerical results with those exact solutions is excellent. For the high-dimensional

Table 5: Comparison of results at $t = 1$ for $\lambda = 0.1$.

x	Hon and Mao [40]	MQQI [39]	BSQI [41]	Our method	Exact
0.1	0.0664	0.07124	0.06628	0.06630	0.06632
0.2	0.1313	0.13431	0.13115	0.13119	0.13121
0.3	0.1928	0.19339	0.19269	0.19271	0.19279
0.4	0.2481	0.24538	0.24792	0.24797	0.24803
0.5	0.2919	0.28517	0.29175	0.29185	0.29191
0.6	0.3159	0.30473	0.31580	0.31598	0.31607
0.7	0.3079	0.29288	0.30791	0.30800	0.30810
0.8	0.2534	0.23784	0.25337	0.25344	0.25372
0.9	0.1459	0.13542	0.14583	0.14587	0.14606

Table 6: Comparison of results at $t = 1$ for $\lambda = 0.01$.

x	Hon and Mao [40]	MQQI [39]	BSQI [41]	Our method	Exact
0.1	0.0755	0.07868	0.07530	0.07538	0.0754
0.2	0.1507	0.15202	0.15049	0.15066	0.1506
0.3	0.2257	0.22554	0.22544	0.22573	0.2257
0.4	0.3003	0.29904	0.30002	0.30028	0.3003
0.5	0.3744	0.37226	0.37407	0.37437	0.3744
0.6	0.4478	0.44484	0.44742	0.44778	0.4478
0.7	0.5202	0.51643	0.51985	0.52038	0.5203
0.8	0.5913	0.58622	0.59106	0.59151	0.5915
0.9	0.6607	0.62956	0.65964	0.66007	0.6600

Table 7: Comparison of results at $t = 1$ for $\lambda = 0.0001$.

x	Hon and Mao [40]	MQQI [39]	BSQI [41]	Our method
0.05	0.0422	0.0422	0.0422	0.0422
0.16	0.1263	0.1263	0.1262	0.1262
0.27	0.2103	0.2103	0.2096	0.2103
0.38	0.2939	0.2939	0.2928	0.2939
0.50	0.3769	0.3769	0.3754	0.3769
0.61	0.4592	0.4592	0.4573	0.4592
0.72	0.5404	0.5404	0.5381	0.5404
0.83	0.6203	0.6201	0.6174	0.6203
0.94	0.6983	0.6957	0.6947	0.6983

Table 8: Comparison of results at different times for $\tau = 0.01$.

	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
$h = 0.02, \lambda = 0.005$	$t = 1.8$	$t = 1.8$	$t = 2.4$	$t = 2.4$	$t = 3.2$	$t = 3.2$
BSQI [41]	1.66464	5.12020	2.06695	6.31491	2.36889	6.85425
QBCM I [17]	0.19127	0.54058	0.14246	0.39241	0.93617	5.54899
QBCM II [17]	0.49130	1.16930	0.41864	0.93664	1.28863	7.49147
MQQI [39]	6.88480	25.6767	7.89738	27.2424	8.56856	2.68122
Our method	0.6642	0.91725	0.7573	1.1465	0.8592	1.2103
$h = 0.02, \lambda = 0.01$	$t = 1.8$	$t = 1.8$	$t = 2.4$	$t = 2.4$	$t = 3.2$	$t = 3.2$
BSQI [41]	0.82751	2.59444	0.98595	2.35031	1.58264	5.73827
QBCM I [17]	0.17014	0.40431	0.20476	0.86363	1.29951	6.69425
QBCM II [17]	0.24003	0.48800	0.30849	1.14760	1.57548	8.06798
MQQI [39]	5.89555	14.7550	6.64358	15.9892	6.90385	16.3403
Our method	0.82751	0.50367	0.46281	1.05625	0.88261	4.73827

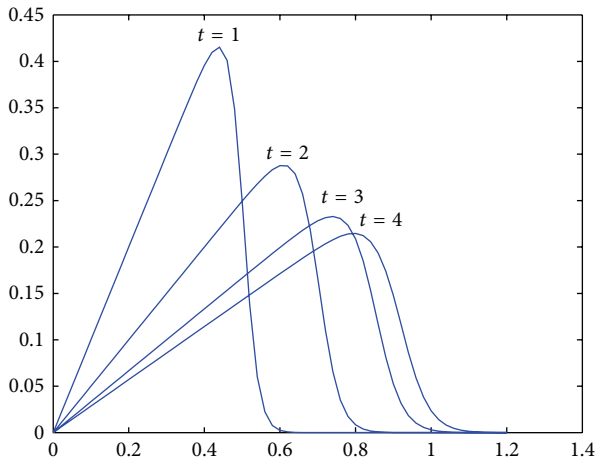

Figure 2: The numerical solutions for $t \leq 4$.

Burgers' equations, we believe our scheme can also be applicable. In this case, we would use multivariate spline quasi-interpolation instead of univariate spline quasi-interpolation. We will consider these problems in our future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Disciplinary Construction Guide Foundation of Harbin Institute of Technology at Weihai (no. WH20140206) and the Scientific Research Foundation of Harbin Institute of Technology at Weihai (no. HIT(WH)201319).

References

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

[2] J. D. Cole, "On a quasi-linear parabolic equation occurring in aerodynamics," *Quarterly of Applied Mathematics*, vol. 9, no. 3, pp. 225–236, 1951.

[3] M. J. Lighthill, "Viscosity effects in sound waves of finite amplitude," in *Surveys in Mechanics*, pp. 250–351, 1956.

[4] L. A. Pospelov, "Propagation of finite amplitude elastic waves(Longitudinal elastic wave of finite amplitude propagation in isotropic solid)," *Soviet Physics-Acoustics*, vol. 11, pp. 302–304, 1966.

[5] B. van der Pol, "On a non-linear partial differential equation satisfied by the logarithm of the Jacobian theta-functions, with arithmetical applications. I, II," *Indagationes Mathematicae*, vol. 13, pp. 261–284, 1951.

[6] H. Bateman, "Some recent researches on the motion of fluids," *Monthly Weather Review*, vol. 43, no. 4, pp. 163–170, 1915.

[7] E. Hopf, "The partial differential equation $U_x + UU_t = U_{xx}$," *Communications on Pure and Applied Mathematics*, vol. 3, no. 3, pp. 201–230, 1950.

[8] I. A. Hassanien, A. A. Salama, and H. A. Hosham, "Fourth-order finite difference method for solving Burgers' equation," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 781–800, 2005.

[9] M. Ciment, S. H. Leventhal, and B. Weinberg, "The operator compact implicit method for parabolic equations," *Journal of Computational Physics*, vol. 28, no. 2, pp. 135–166, 1978.

[10] R. S. Hirsh, "Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique," *Journal of Computational Physics*, vol. 19, no. 1, pp. 90–109, 1975.

[11] S. Kutluay, A. R. Bahadir, and A. Özdeş, "Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods," *Journal of Computational and Applied Mathematics*, vol. 103, no. 2, pp. 251–261, 1999.

[12] P. Arminjon and C. Beauchamp, "A finite element method for Burgers' equation in hydrodynamics," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 3, pp. 415–428, 1978.

[13] L. Iskandar and A. Mohsen, "Some numerical experiments on the splitting of Burgers' equation," *Numerical Methods for Partial Differential Equations*, vol. 8, no. 3, pp. 267–276, 1992.

[14] P. C. Jain and M. Raja, "Splitting-up technique for Burgers' equations," *Indian Journal of Pure and Applied Mathematics*, vol. 10, pp. 1543–1551, 1979.

[15] T. Öziş, E. N. Aksan, and A. Özdeş, "A finite element approach for solution of Burgers' equation," *Applied Mathematics and Computation*, vol. 139, no. 2-3, pp. 417–428, 2003.

[16] A. Dogan, "A Galerkin finite element approach to Burgers' equation," *Applied Mathematics and Computation*, vol. 157, no. 2, pp. 331–346, 2004.

[17] İ. Dağ, B. Saka, and A. Boz, "B-spline Galerkin methods for numerical solutions of the Burgers' equation," *Applied Mathematics and Computation*, vol. 166, no. 3, pp. 506–522, 2005.

[18] A. H. Khater, R. S. Temsah, and M. M. Hassan, "A Chebyshev spectral collocation method for solving Burgers'-type equations," *Journal of Computational and Applied Mathematics*, vol. 222, no. 2, pp. 333–350, 2008.

[19] A. K. Khalifa, K. I. Noor, and M. A. Noor, "Some numerical methods for solving Burgers equation," *International Journal of Physical Sciences*, vol. 6, no. 7, pp. 1702–1710, 2011.

[20] A. Korkmaz and I. Dağ, "Shock wave simulations using sinc differential quadrature method," *Engineering Computations*, vol. 28, no. 6, pp. 654–674, 2011.

[21] K. Altiparmak and T. Özis, "Numerical solution of Burgers' equation with factorized diagonal Padé approximation," *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 21, no. 3-4, pp. 310–319, 2011.

[22] İ. Dağ, D. Irk, and A. Şahin, "B-spline collocation methods for numerical solutions of the Burgers' equation," *Mathematical Problems in Engineering*, vol. 2005, no. 5, pp. 521–538, 2005.

[23] S. S. Xie, S. Heo, S. Kim, G. Woo, and S. Yi, "Numerical solution of one-dimensional Burgers' equation using reproducing kernel function," *Journal of Computational and Applied Mathematics*, vol. 214, no. 2, pp. 417–434, 2008.

[24] P. Sablonniere, "Univariate spline quasi-interpolants and applications to numerical analysis," *Rendiconti del Seminario Matematico*, vol. 63, no. 3, pp. 211–222, 2005.

[25] Y. Ren and X. Li, "A meshfree method for Signorini problems using boundary integral equations," *Mathematical Problems in Engineering*, vol. 2014, Article ID 490127, 12 pages, 2014.

[26] R. J. Cheng, L. W. Zhang, and K. M. Liew, "Modeling of biological population problems using the element-free kp-Ritz method," *Applied Mathematics and Computation*, vol. 227, pp. 274–290, 2014.

[27] L. W. Zhang, Y. J. Deng, and K. M. Liew, "An improved element-free Galerkin method for numerical modeling of the biological population problems," *Engineering Analysis with Boundary Elements*, vol. 40, pp. 181–188, 2014.

[28] L. W. Zhang, P. Zhu, and K. M. Liew, "Thermal buckling of functionally graded plates using a local Kriging meshless method," *Composite Structures*, vol. 108, pp. 472–492, 2014.

[29] Z. X. Lei, L. W. Zhang, K. M. Liew, and J. L. Yu, "Dynamic stability analysis of carbon n anotube-reinforced functionally graded cylindrical panels using the element-free Kp-Ritz method," *Composite Structures*, vol. 113, pp. 328–338, 2014.

[30] K. M. Liew, Z. X. Lei, J. L. Yu, and L. W. Zhang, "Postbuckling of carbon nanotube-reinforced functionally graded cylindrical panels under axial compression using a meshless approach," *Computer Methods in Applied Mechanics and Engineering*, vol. 268, pp. 1–17, 2014.

[31] T. Belytschko, Y. Y. Lu, and L. Gu, "Element-free Galerkin methods," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 2, pp. 229–256, 1994.

[32] J. J. Monaghan, "An introduction to SPH," *Computer Physics Communications*, vol. 48, no. 1, pp. 89–96, 1988.

[33] K. M. Liew, X. Zhao, and T. Y. Ng, "The element-free Kp-Ritz method for vibration of laminated rotating cylindrical panels," *International Journal of Structural Stability and Dynamics*, vol. 2, no. 4, pp. 523–558, 2002.

[34] K. M. Liew, H. Y. Wu, and T. Y. Ng, "Meshless method for modeling of human proximal femur: treatment of nonconvex boundaries and stress analysis," *Computational Mechanics*, vol. 28, no. 5, pp. 390–400, 2002.

[35] K. M. Liew, Y. C. Wu, G. P. Zou, and T. Y. Ng, "Elasto-plasticity revisited: Numerical analysis via reproducing kernel particle method and parametric quadratic programming," *International Journal for Numerical Methods in Engineering*, vol. 55, no. 6, pp. 669–683, 2002.

[36] K. M. Liew, X. L. Chen, and J. N. Reddy, "Mesh-free radial basis function method for buckling analysis of non-uniformly loaded arbitrarily shaped shear deformable plates," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 3–5, pp. 205–224, 2004.

[37] S. N. Atluri and T. Zhu, "A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics," *Computational Mechanics*, vol. 22, no. 2, pp. 117–127, 1998.

[38] W. K. Liu, S. Jun, and Y. F. Zhang, "Reproducing kernel particle methods," *International Journal for Numerical Methods in Fluids*, vol. 20, no. 8-9, pp. 1081–1106, 1995.

[39] R. Chen and Z. Wu, "Applying multiquadratic quasi-interpolation to solve Burgers' equation," *Applied Mathematics and Computation*, vol. 172, no. 1, pp. 472–484, 2006.

[40] Y. C. Hon and X. Z. Mao, "An efficient numerical scheme for Burgers' equation," *Applied Mathematics and Computation*, vol. 95, no. 1, pp. 37–50, 1998.

[41] C. Zhu and R. Wang, "Numerical solution of Burgers' equation by cubic B-Spline quasi-interpolation," *Applied Mathematics and Computation*, vol. 208, no. 1, pp. 260–272, 2009.

[42] A. Asaithambi, "Numerical solution of the Burgers' equation by automatic differentiation," *Applied Mathematics and Computation*, vol. 216, no. 9, pp. 2700–2708, 2010.

[43] B. Saka and I. Dağ, "Quartic B-spline collocation method to the numerical solutions of the Burgers' equation," *Chaos, Solitons and Fractals*, vol. 32, no. 3, pp. 1125–1137, 2007.

[44] I. A. Hassanien, A. A. Salama, and H. A. Hosham, "Fourth-order finite difference method for solving Burgers' equation," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 781–800, 2005.

[45] C. de Boor, *A Practical Guide to Splines*, Springer, New York, NY, USA, 1978.

[46] W. L. Wood, "An exact solution for Burgers' equation," *Communications in Numerical Methods in Engineering*, vol. 22, no. 7, pp. 797–798, 2006.

[47] I. Christie, D. F. Griffiths, and A. R. Mitchell, "Product approximation for nonlinear problems in the finite element method," *IMA Journal of Numerical Analysis*, vol. 1, no. 3, pp. 253–266, 1981.