Article

Quasi-Interpolation Operators for Bivariate Quintic Spline Spaces and Their Applications

Rengui Yu 1,*, Chungang Zhu 2, Xianmin Hou 1 and Li Yin 1

1 College of Information and Management Science, Henan Agricultural University, Zhengzhou 450000, China; houxianmin83@126.com (X.H.); yinli7655@163.com (L.Y.)
2 School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China; cgzhu@dlut.edu.cn
* Correspondence: yurengui2005@163.com; Tel.: +86-371-5699-0030

Academic Editor: Fazal M. Mahomed
Received: 10 October 2016; Accepted: 27 December 2016; Published: 19 January 2017

Abstract: Splines and quasi-interpolation operators are important both in approximation theory and applications. In this paper, we construct a family of quasi-interpolation operators for the bivariate quintic spline spaces $S^3_5(\Delta_{mn}^2)$. Moreover, the properties of the proposed quasi-interpolation operators are studied, as well as its applications for solving the two-dimensional Burgers’ equation and image reconstruction. Some numerical examples show that these methods, which are easy to implement, provide accurate results.

Keywords: bivariate spline space; quasi-interpolation operator; Type-2 triangulation; Burgers’ equations; image reconstruction

1. Introduction

Spline functions are very important in both approximation theory and applications in science and engineering. Essentially, a spline is a piecewise polynomial function with certain smoothness. The special importance of spline functions is due to the mechanical meaning of the univariate spline, which was discussed in the famous paper written by Schoenberg [1]. Univariate splines were introduced and analyzed in the seminal paper by Schoenberg [1] in 1946. Multivariate splines are the generalizations of univariate splines. In 1976, de Boor [2] generalized univariate B-splines to multivariate splines. However, the generalizations of these kinds of definitions are inconvenient to the basic theoretical research. The study of multivariate B-splines was not active until the generalized functional expressions came up. The generalized functional expressions (including simplex splines, Box splines and conical splines, etc.) were given by de Boor, and Dahmen and Micchelli [3,4]. In 1975, Wang [5] established the so-called “smoothing cofactor–conformality method” to study the general theory on multivariate splines for any partition by using the methods of function theory and algebraic geometry. Splines have been widely applied to fields such as function approximation and numerical analysis, computer geometry, computer aided geometric design, image processing, and so on [6–12]. In fact, spline functions have become a fundamental tool in these fields. Bivariate and trivariate splines are easy to store, evaluate and manipulate on a computer, so they are well-suited to address the resolution of many problems of practical interest.

Burgers’ equation plays a significant role in the study of partial differential equations from fluid mechanics. It occurs in various areas of applied mathematics, such as modeling of dynamics, heat conduction, and acoustic waves [13–15]. Numerical techniques for the Burgers’ equation usually fall into the following classes: finite differences, finite elements, and spectral methods [16]. Recently, the Adomian Decomposition Method (ADM) gained much attention for solving Burgers’ equation [17–19]. A discrete ADM method is also proposed in [16]. In addition, a B-spline quasi-interpolation method is proposed in [12] for solving a 1D Burgers’ equation. There are many methods for image reconstruction,
such as those based on convolution back-projection, the harmony search algorithm, and compressed sensing [20–22]. In this paper, we construct bivariate quasi-interpolation operators to solve the 2D Burgers’ equation and reconstruct images.

The paper is organized as follows. In Section 2, we study the bivariate quintic spline spaces \( S_{\Delta}^{3}(\Lambda_{mn}^{(2)}) \) by using the smoothing cofactor–conformality method. A family of quasi-interpolation operators are also presented in Section 2. In Section 3, we give some examples for solving the 2D Burgers’ equation and reconstruct images. Moreover, comparisons with other techniques are also provided.

2. The Bivariate Spline Space \( S_{\Delta}^{3}(\Lambda_{mn}^{(2)}) \)

2.1. The Spaces \( S_{\Delta}^{3}(\Lambda_{mn}^{(2)}) \)

Let \( D \) be a domain in \( \mathbb{R}^2 \) and \( \mathbb{P}_k \) be the collection of all bivariate polynomials with real coefficients and total degree \( \leq k \),

\[
\mathbb{P}_k := \{ p(x, y) = \sum_{i=0}^{k} \sum_{j=0}^{k-i} c_{ij} x^i y^j | c_{ij} \in \mathbb{R} \}. \tag{1}
\]

Using a finite number of irreducible algebraic curves to carry out the partition \( \Delta \), we divide the domain \( D \) into a finite number of sub-domains \( D_1, D_2, \ldots, D_N \). Each sub-domain is called a cell. The line segments that form the boundary of each cell are called the “edges”, and intersection points of the edges are called the “vertices”. The space of multivariate spline functions is defined by

\[
S_{\Delta}^{\mu}(\Lambda) := \{ s \in C^{\mu}(D) | s|_{D_i} \in \mathbb{P}_k, i = 1, \ldots, N \}. \tag{2}
\]

A spline \( s \) is a piecewise polynomial function of degree \( k \) possessing continuous partial derivatives up to the order \( \mu \) in \( D \).

Suppose \( D = [0, m] \times [0, n] \) for given positive integers \( m \) and \( n \), endowed with the decomposition induced by the four-directional mesh \( \Lambda_{mn}^{(2)} \) with grid lines:

\[
x - i = 0, \quad y - i = 0, \quad x - y - i = 0, \quad x + y - i = 0, \quad i \in \mathbb{Z}.
\]

We have the following result [10,23].

**Theorem 1.** For the bivariate spline space \( S_{\Delta}^{\mu}(\Lambda_{mn}^{(2)}) \), it holds

\[
\dim S_{\Delta}^{\mu}(\Lambda_{mn}^{(2)}) = \left( \frac{k + 2}{2} \right) + (3m + 3n - 4) \left( \frac{k - \mu + 1}{2} \right) + mn \left( \frac{k - 2\mu}{2} \right) + (m - 1)(n - 1) \cdot d_{k}^{\mu}(4), \tag{3}
\]

where

\[
d_{k}^{\mu}(4) := \frac{1}{2} \left( k - \mu - \left[ \frac{\mu + 1}{3} \right] \right) \cdot \left( 3k - 5\mu + 3 \left[ \frac{\mu + 1}{3} \right] + 1 \right).
\]

Since each interior grid point in \( D \) is the intersection of exactly four lines from the grid partition \( \Lambda_{mn}^{(2)} \), the degree \( k \) and the smoothness \( \mu \) must satisfy the relationship [10]

\[
\mu < \frac{3k - 1}{4}.
\]

It is easy to see that the spaces \( S_{\Delta}^{1}(\Lambda_{mn}^{(2)}), S_{\Delta}^{2}(\Lambda_{mn}^{(2)}), S_{\Delta}^{3}(\Lambda_{mn}^{(2)}), \ldots \) have a locally supported basis [10]. For \( S_{\Delta}^{1}(\Lambda_{mn}^{(2)}) \) and \( S_{\Delta}^{2}(\Lambda_{mn}^{(2)}) \), we refer to [10,24–28] for more details. We discuss the spline spaces \( S_{\Delta}^{3}(\Lambda_{mn}^{(2)}) \) in the following sections.
2.2. Basis of $S^3_3(\Delta_{mn}^{(2)})$

By (3), we get the dimension of $S^3_3(\Delta_{mn}^{(2)})$ as follows:

$$\dim S^3_3(\Delta_{mn}^{(2)}) = \binom{5 + 2}{2} + (3m + 3n - 4)\binom{5 - 3 + 1}{2} + 0 + (m - 1)(n - 1) \cdot 4$$

$$= 2mn + 7m + 7n + 11.$$  

(4)

By using the smoothing cofactor–conformality method [5,10], we obtain two bivariate splines $A$ and $B$ whose supports are shown in Figure 1a,b. The centers of the supports are $(-1/2 \cdot h, -1/2 \cdot h)$ and $(0,0)$, respectively. Here, the considered domain $D$ is $[0, mh] \times [0, nh]$. The local supports of $A(x, y)$ and $B(x, y)$ are minimal, and for $A(x, y)$, there are four symmetry axes:

$$x + 1/2 \cdot h = 0, \quad y + 1/2 \cdot h = 0, \quad x - y = 0, \quad x + y + h = 0,$$

while, for $B(x, y)$, there also exist four symmetry axes:

$$x = 0, \quad y = 0, \quad x - y = 0, \quad x + y = 0.$$

The restriction $p_i$ of $A$ to the cell $D_i (i = 1, \ldots, 7)$ in Figure 1a are as follows:

$$p_1(x, y) = \frac{1}{20h^5}(x - \frac{3}{2}h)^4(x - 5y + h),$$

$$p_2(x, y) = p_1(x, y) + \frac{1}{16h^5}(-x - y + 2h)(x - y - h)^4,$$

$$p_3(x, y) = \frac{1}{80h^5}(-x - y + 2h)^5,$$

$$p_5(x, y) = p_2(x, y) + \frac{1}{40h^5}(3x - 2y - \frac{11}{2}h)(x + y - h)^4,$$

$$p_4(x, y) = p_5(x, y) + \frac{1}{16h^5}(x + y - 2h)(x - y - h)^4,$$

$$p_6(x, y) = p_5(x, y) + \frac{1}{20h^5}(-3x - 5y - h)(x - \frac{1}{2}h)^4,$$

$$p_7(x, y) = p_6(x, y) + \frac{1}{80h^5}(x + y + 10h)(x + y)^4.$$

The restriction $q_i$ of $B$ to the cell $D_i (i = 1, \ldots, 10)$, in Figure 1b are as follows:

$$q_1(x, y) = \frac{1}{40h^5}(-x + \frac{5}{2}h)^5,$$

$$q_2(x, y) = q_1(x, y) + \frac{1}{320h^5}(4x + 6y - 13h)(x - y - 2h)^4,$$

$$q_3(x, y) = \frac{1}{320h^5}(-4x + 6y + 7h)(x + y - 3h)^4,$$

$$q_4(x, y) = q_3(x, y) + \frac{1}{40h^5}(x + 5y - 9h)(x - \frac{3}{2}h)^4,$$

$$q_5(x, y) = q_4(x, y) - \frac{1}{160h^5}(x + y + 3h)(x + y - 2h)^4,$$

$$q_6(x, y) = q_5(x, y) + \frac{1}{320h^5}(4x - 6y - 13h)(x + y - 2h)^4.$$
Then, $G$ Let Theorem 2.

The expressions of the restrictions of $A$ and $B$ to the other cells are obtained by symmetry.

$$q_7(x, y) = q_6(x, y) + \frac{1}{80h^3} (2x - 13h)(x - \frac{3}{2}h)^4,$$

$$q_8(x, y) = q_5(x, y) - \frac{1}{40h^3} (5x + y - 13h)(y - \frac{1}{2}h)^4,$$

$$q_9(x, y) = q_8(x, y) + \frac{1}{320h^3} (-6x + 4y + 11h)(x + y - h)^4,$$

$$q_{10}(x, y) = q_9(x, y) + \frac{1}{4h^4} (x - \frac{1}{2}h)^4.$$

The expressions of the restrictions of $A$ and $B$ to the other cells are obtained by symmetry.

![Figure 1](image.png)

Figure 1. The supports of two B-splines in $S^3_2(\Delta_{mn}^{(2)})$. (a) $A(x, y)$; (b) $B(x, y)$.

Denote by $A_{ij}, B_{ij}$ the translates of $A$ and $B$, i.e., for all $ij \in \mathbb{Z}$

$$A_{ij}(x, y) := A(x - ih - \frac{1}{2}h, y - jh - \frac{1}{2}h),$$

$$B_{ij}(x, y) := B(x - ih - \frac{1}{2}h, y - jh - \frac{1}{2}h).$$

(5)

It is clear that the index sets for which the functions $A_{ij}$ and $B_{ij}$ do not vanish identically on $D$ are

$$E = \{(i, j) = (\alpha, \beta) : -1 \leq \alpha \leq m + 1, -1 \leq \beta \leq n + 1\},$$

$$F = \{(i, j) = (\alpha, \beta) : -2 \leq \alpha \leq m + 1, -2 \leq \beta \leq n + 1$$

$$(i, j) \neq (-2, -2), (m + 1, -2), (-2, n + 1), (m + 1, n + 1)\}.$$

Since the number of these splines is $2mn + 7m + 7n + 21$, which is larger than the dimension of $S^3_2(\Delta_{mn}^{(2)})$, these splines are linearly dependent. For constructing a basis of $S^3_2(\Delta_{mn}^{(2)})$, we need to delete ten splines from the ones. We have the following result.

**Theorem 2.** Let

$$G^{1}_{s,t} := \{A_{s,t} : (s, t) \in E \setminus \{(m - 1, n), (m, n), (m + 1, n), (m - 1, n + 1), (m, n + 1)\}\},$$

$$G^{2}_{s,t} := \{B_{s,t} : (s, t) \in F \setminus \{(m - 1, n - 1), (m, n - 1), (m + 1, n - 1), (m, n), (m + 1, n)\}\}.$$

Then, $G^{1}_{s,t} \cup G^{2}_{s,t}$ is a basis of $S^3_2(\Delta_{mn}^{(2)})$. 
Since the cardinality of $G_1 \cup G_2$ is the same as the dimension of $S_3^5(\Delta_{mn})$, it is sufficient to prove that $G_1 \cup G_2$ is a linearly independent set on $D$. This can be done by following the proof of Theorem 3.1 in [29].

By checking the sums of the appropriate Bézier coefficients, we have the following identities:

$$
\sum_{(i,j) \in E} (-1)^{i+j} A_{ij}(x,y) = 0, \quad \sum_{(i,j) \in F} (-1)^{i+j} B_{ij}(x,y) = 0,
$$

$$
\sum_{(i,j) \in E} A_{ij}(x,y) = 1, \quad \sum_{(i,j) \in F} B_{ij}(x,y) = 1.
$$

2.3. Quasi-Interpolation Operators for $S_3^5(\Delta_{mn})$

From the basis functions in the previous section, we can construct various kinds of quasi-interpolation operators.

**Theorem 3.** Let

$$
L(f) := \sum_{i,j \in E} f(ih - \frac{1}{2}h, jh - \frac{1}{2}h) A_{ij},
$$

$$
V(f) := \sum_{i,j \in F} g(ih, jh) B_{ij}.
$$

Then, for $p \in P_1 \cup \{xy\}$, it holds

$$
L(p) = p, V(p) = p, (x, y) \in D.
$$

In applications, a linear combination $C_{ij}$ of splines $A_{ij}$ and $B_{ij}$ is used [30,31]. It is given by

$$
C_{ij} = \frac{1}{12} (A_{ij} + A_{ij+1} + A_{i+1,j} + A_{i+1,j+1}) + \frac{2}{3} B_{ij}.
$$

The support of $C_{ij}$ is the union of the involved splines $A_{k,l}$ and $B_{k,l}$. The center of the support is $(i + 1/2 \cdot h, j + 1/2 \cdot h)$ and the number of $C_{ij}$ does not vanish identically on $D$ is $mn + 4m + 4n + 16$, which is less than the dimension of $S_3^5(\Delta_{mn})$, so all $C_{ij}$ can only span a proper subspace of $S_3^5(\Delta_{mn})$. The shape of one $C_{ij}$ is shown in Figure 2b. The shapes of the splines $A$ and $B$ are displayed in Figure 3. The splines $\{C_{ij}\}$ form a partition of unity.

\[Image of Figure 2\]

**Figure 2.** The support and shape of $C_{ij}(x,y)$ in $S_3^5(\Delta_{mn})$. (a) $C_{ij}(x,y)$; (b) $C_{-1,-1}(x,y)$. 
Figure 3. The shapes of two B-splines in $S_3^2(\Delta_{mn})$. (a) $A(x,y)$; (b) $B(x,y)$.

It is worthwhile to note that, only using $C_{i,j}$, we can construct higher precision quasi-interpolation operators by the following theorems.

Let $\Omega$ denote an open set containing $D$ and $f_{i,j} = f(ih,jh)$. Define the variation diminishing operator $W : C(\Omega) \rightarrow S_3^2(\Delta_{mn})$:

$$W(f) = \sum_{i=-2}^{m+1} \sum_{j=-2}^{n+1} f_{i,j} C_{i,j}. \quad (6)$$

Note that $W$ is a linear operator, and by simple checking, we have the following results.

**Theorem 4.** For all $(x,y) \in D$, $f \in P_1 \bigcup \text{span}\{xy\}$, we have

$$W(f) = f.$$

In order to preserve identities for all polynomials in $P_2$ and $P_3$, we define other kinds of linear operators $W' : C(\Omega) \rightarrow S_3^2(\Delta_{mn})$:

$$W'(f) = \sum_{i=-2}^{m+1} \sum_{j=-2}^{n+1} \lambda_{i,j}(f) C_{i,j}. \quad (7)$$

where

$$\lambda_{i,j}(f) = w_1 \cdot f_{i-1,j+1} + w_2 \cdot f_{i,j+1} + w_3 \cdot f_{i+1,j+1} + w_4 \cdot f_{i-1,j+\frac{1}{2}} + w_5 \cdot f_{i,j+\frac{1}{2}} + w_6 \cdot f_{i+1,j+\frac{1}{2}} + w_7 \cdot f_{i-1,j-\frac{1}{2}} + w_8 \cdot f_{i,j-\frac{1}{2}} + w_9 \cdot f_{i+1,j-\frac{1}{2}}.$$

Note that each linear functional $\lambda_{i,j}$ depends on nine function values of $f$ at the grid points and the corresponding midpoints in the support of $C_{i,j}$. We have the following results:

**Theorem 5.** For all $(x,y) \in D$, $f \in P_3 \bigcup \text{span}\{x^3, xy^3\}$, we have

$$w_1 = w_3 = w_7 = w_9 = -\frac{5}{12}, \quad w_2 = w_4 = w_6 = w_8, \quad w_5 = \frac{8}{3} - 2w_8,$$

$$W'(f) = f.$$
There is a unique value of \(w_8\) for which the corresponding operator is exact on \(P_4\). 

**Theorem 6.** For all \((x, y) \in D, f \in P_4\), \(\text{span}\{x^4, y^4\}\),

\[
\begin{align*}
    w_1 &= w_3 = w_7 = w_9 = \frac{133}{180}, \\
    w_2 &= w_4 = w_6 = w_8 = -\frac{104}{45}, \\
    w_5 &= \frac{328}{45},
\end{align*}
\]

we have

\[W'(f) = f.\]

Note that Theorem 6 has a better result than Theorem 5 but at the cost of using all nine function values. The conclusion of Theorem 5 can be used with flexibility, i.e., having the opportunity of choosing approximate \(w_i\) for given problems. The commonly used coefficients \(w_i\) of Theorem 5 are as follows:

\[
\begin{align*}
    w_1 &= w_3 = w_7 = w_9 = \frac{5}{12}, \\
    w_2 &= w_4 = w_6 = w_8 = 0, \\
    w_5 &= \frac{8}{3}.
\end{align*}
\]

We have the following result:

**Theorem 7.** For all \((x, y) \in D, f \in P_3 \cup \text{span}\{x^3 y, xy^3\}\), we have

\[W'(f) = f,\]

where

\[
\lambda_{i,j}(f) = \frac{8}{3} f_{i,j} - \frac{5}{12} \left[ f_{i-\frac{1}{2}, j-\frac{1}{2}} + f_{i+\frac{1}{2}, j-\frac{1}{2}} + f_{i-\frac{1}{2}, j+\frac{1}{2}} + f_{i+\frac{1}{2}, j+\frac{1}{2}} \right].
\]

To prove Theorems 5–7, we need to testify the conclusions at each sub-region \(D_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]\) (see Figure 4a). For \(I\) in \(D_{ij}\), we have a properly posed set of nodes for multivariate spline interpolation (21 points, see Figure 4b) [10]. Next, by computing the values of \((W'f)(x, y)\) in (7) at these 21 points (noted by \(P\)), we get the fact that \((W'f)(p) = f(p), \forall p \in P\). The same fact can be obtained similarly for \(II, III, IV\) in \(D_{ij}\), respectively. Hence, these theorems hold by the arbitrariness of \(D_{ij}\).

**Figure 4.** Sub-region of \(D\). (a) \(D_{ij}\); (b) properly posed set of nodes for interpolation.

In the next section, we give two applications of the quasi-interpolation operator \(W'\) in Theorem 7.
3. Applications of Quasi-Interpolation Operator $W'$

3.1. Solving 2D Burgers’ Equations

Here, we consider the system of 2D Burgers’ equations [32]. One can refer to [32,33] for more details:

$$u_t + uu_x + vu_y = \frac{1}{R}(u_{xx} + u_{yy}),$$

$$v_t + uu_x + vu_y = \frac{1}{R}(v_{xx} + v_{yy}),$$

with initial conditions

$$u(x, y, 0) = f(x, y), \quad (x, y) \in D,$$

$$v(x, y, 0) = g(x, y), \quad (x, y) \in D,$$

and boundary conditions:

$$u(x, y, t) = f_1(x, y, t), \quad (x, y) \in \partial D, \quad t > 0,$$

$$v(x, y, t) = g_1(x, y, t), \quad (x, y) \in \partial D, \quad t > 0,$$

where $D = \{(x, y) \mid a \leq x \leq b, a \leq y \leq b\}$ and $\partial D$ denotes its boundary. Functions $u$ and $v$ are the velocity components to be determined, $f, g, f_1$ and $g_1$ are known functions, and $R$ is the Reynolds number.

Discretizing Burgers’ Equation (8) in the time domain with step $\tau$ and using the derivatives of $(W'u)(x, y), (W'v)(x, y)$ defined in Theorem 7 to approximate the corresponding derivatives of $u(x, y, t)$ and $v(x, y, t)$ yields

$$u_{i,j}^{n+1} = u_{i,j}^n + \tau \left( \frac{1}{R} \left( ((W'u)_{xx})_{i,j}^n + ((W'u)_{yy})_{i,j}^n \right) - u_{i,j}^n ((W'u)_x)_{i,j}^n - v_{i,j}^n ((W'u)_y)_{i,j}^n \right),$$

$$v_{i,j}^{n+1} = v_{i,j}^n + \tau \left( \frac{1}{R} \left( ((W'v)_{xx})_{i,j}^n + ((W'v)_{yy})_{i,j}^n \right) - u_{i,j}^n ((W'v)_x)_{i,j}^n - v_{i,j}^n ((W'v)_y)_{i,j}^n \right),$$

where $u_{i,j}^n, v_{i,j}^n$ are the approximation of the value of $u(x, y, t), v(x, y, t)$ at the uniform mesh grid $(i\Delta x, j\Delta y, n\tau)$. This scheme provides a numerical solution for Burgers’ equation, which is called multivariate spline quasi-interpolation (MSQI) scheme.

By iterating this scheme, we obtain the numerical solution for Burgers’ equations. We give the following examples.

**Example 1.** A Hopf–Cole transformation [34] allows for determining the exact solution

$$u(x, y, t) = \frac{3}{4} - \frac{1}{4(1 + e^{(R(-t-4x+4y)/80)})/32},$$

$$v(x, y, t) = \frac{3}{4} + \frac{1}{4(1 + e^{(R(-t-4x+4y)/80)})/32},$$

of Burgers’ equation with initial conditions and boundary conditions provided by the exact solution. In this example, we use the same Reynolds number $R = 80$, time step size $\tau = 10^{-4}$ and computational domain $D = \{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}$. For the uniform mesh grid, we use a looser one with $h_x = h_y = 0.1$ for a lower computation.

Figure 5a,b show the approximation solutions of $u(x, y, t)$ and $v(x, y, t)$ at steady state, respectively. In the meantime, the numerical solution for different times and different mesh grid points are given in Tables 1 and 2.
Figure 5. A numerical illustration of approximation solutions \(u(x, y, t)\) (a) and \(v(x, y, t)\) (b) by multivariate spline quasi-interpolation (MSQI) with \(R = 80, \tau = 10^{-4}, h_x = h_y = 0.1\) at \(t = 0.5\).

Table 1. Comparison of the numerical results by MSQI scheme with exact solutions \(u(x, y, t)\) with \(R = 80, \tau = 10^{-4}\) at different times \(t\).

<table>
<thead>
<tr>
<th>Mesh Grid</th>
<th>(t = 0.05)</th>
<th>(t = 0.2)</th>
<th>(t = 0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(u_{\text{num}})</td>
<td>(u_{\text{exact}})</td>
<td>(u_{\text{num}})</td>
</tr>
<tr>
<td>(0.1, 0.1)</td>
<td>0.61749</td>
<td>0.61720</td>
<td>0.59474</td>
</tr>
<tr>
<td>(0.9, 0.2)</td>
<td>0.50021</td>
<td>0.50020</td>
<td>0.50015</td>
</tr>
<tr>
<td>(0.8, 0.3)</td>
<td>0.50150</td>
<td>0.50148</td>
<td>0.50108</td>
</tr>
<tr>
<td>(0.9, 0.5)</td>
<td>0.50403</td>
<td>0.50398</td>
<td>0.50289</td>
</tr>
<tr>
<td>(0.8, 0.6)</td>
<td>0.52640</td>
<td>0.52667</td>
<td>0.51859</td>
</tr>
<tr>
<td>(0.2, 0.8)</td>
<td>0.74931</td>
<td>0.74930</td>
<td>0.74900</td>
</tr>
<tr>
<td>(0.9, 0.9)</td>
<td>0.61720</td>
<td>0.61720</td>
<td>0.59456</td>
</tr>
</tbody>
</table>

Table 2. Comparison of the numerical results by MSQI with exact solutions \(v(x, y, t)\) with \(R = 80, \tau = 10^{-4}\) at different times \(t\).

<table>
<thead>
<tr>
<th>Mesh Grid</th>
<th>(t = 0.05)</th>
<th>(t = 0.2)</th>
<th>(t = 0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(v_{\text{num}})</td>
<td>(v_{\text{exact}})</td>
<td>(v_{\text{num}})</td>
</tr>
<tr>
<td>(0.1, 0.1)</td>
<td>0.88251</td>
<td>0.88280</td>
<td>0.90526</td>
</tr>
<tr>
<td>(0.9, 0.2)</td>
<td>0.99979</td>
<td>0.99980</td>
<td>0.99985</td>
</tr>
<tr>
<td>(0.8, 0.3)</td>
<td>0.99850</td>
<td>0.99852</td>
<td>0.99892</td>
</tr>
<tr>
<td>(0.9, 0.5)</td>
<td>0.99597</td>
<td>0.99602</td>
<td>0.99711</td>
</tr>
<tr>
<td>(0.8, 0.6)</td>
<td>0.97360</td>
<td>0.97333</td>
<td>0.98141</td>
</tr>
<tr>
<td>(0.2, 0.8)</td>
<td>0.75009</td>
<td>0.75070</td>
<td>0.75100</td>
</tr>
<tr>
<td>(0.9, 0.9)</td>
<td>0.88250</td>
<td>0.88280</td>
<td>0.90544</td>
</tr>
</tbody>
</table>

Tables 3 and 4 show the absolute errors of the MSQI method in comparison with Bahadir [32]. The results of the discrete ADM method [16] are also listed in these two tables for comparison. A good approximation with the exact solution of the equations is achieved by using only a looser mesh grid.
Table 3. Comparison of absolute errors for $u(x,y,t)$ with $R = 80$, $\tau = 10^{-4}$ at different times $t$.

<table>
<thead>
<tr>
<th>Mesh Grid</th>
<th>$t = 0.01$</th>
<th>$t = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.1, 0.1)</td>
<td>1.63803 × 10^{-5}</td>
<td>7.24132 × 10^{-5}</td>
</tr>
<tr>
<td>(0.5, 0.1)</td>
<td>1.85815 × 10^{-5}</td>
<td>2.42869 × 10^{-5}</td>
</tr>
<tr>
<td>(0.9, 0.1)</td>
<td>1.64831 × 10^{-7}</td>
<td>8.39751 × 10^{-6}</td>
</tr>
<tr>
<td>(0.3, 0.3)</td>
<td>1.65000 × 10^{-5}</td>
<td>8.25331 × 10^{-5}</td>
</tr>
<tr>
<td>(0.7, 0.3)</td>
<td>1.94033 × 10^{-5}</td>
<td>3.43163 × 10^{-5}</td>
</tr>
<tr>
<td>(0.1, 0.5)</td>
<td>1.61309 × 10^{-7}</td>
<td>5.62014 × 10^{-5}</td>
</tr>
</tbody>
</table>

Table 4. Comparison of absolute errors for $v(x,y,t)$ with $R = 80$, $\tau = 10^{-4}$ at different times $t$.

<table>
<thead>
<tr>
<th>Mesh Grid</th>
<th>$t = 0.01$</th>
<th>$t = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.1, 0.1)</td>
<td>1.63803 × 10^{-5}</td>
<td>8.35601 × 10^{-5}</td>
</tr>
<tr>
<td>(0.5, 0.1)</td>
<td>1.85815 × 10^{-5}</td>
<td>5.13642 × 10^{-5}</td>
</tr>
<tr>
<td>(0.9, 0.1)</td>
<td>1.64832 × 10^{-7}</td>
<td>7.03296 × 10^{-6}</td>
</tr>
<tr>
<td>(0.3, 0.3)</td>
<td>1.65000 × 10^{-5}</td>
<td>6.15201 × 10^{-5}</td>
</tr>
<tr>
<td>(0.7, 0.3)</td>
<td>1.94033 × 10^{-5}</td>
<td>5.41000 × 10^{-5}</td>
</tr>
<tr>
<td>(0.1, 0.5)</td>
<td>1.61310 × 10^{-7}</td>
<td>7.35192 × 10^{-5}</td>
</tr>
</tbody>
</table>

Example 2. In the second problem, we consider the 2D Burgers’ equations with the following initial conditions [33]

\[
\begin{align*}
  u(x, y, 0) &= x + y, \\
  v(x, y, 0) &= x - y,
\end{align*}
\]

and the computational domain has been taken as $D = \{(x, y) : 0 \leq x \leq 0.5, 0 \leq y \leq 0.5\}$. The exact solutions are as follows:

\[
\begin{align*}
  u(x, y, t) &= \frac{x + y - 2xt}{1 - 2t^2}, \\
  v(x, y, t) &= \frac{x - y - 2yt}{1 - 2t^2}.
\end{align*}
\]

Numerical results using the MSQI method are listed in Tables 5 and 6. In these two tables, we also provide numerical errors and comparison with those given in [16]. All the results in Tables 5 and 6 are calculated with uniform mesh grid $h_x = h_y = 0.05$, time step size $\tau = 10^{-4}$ and arbitrary Reynolds number $R$ at different times $t$. In Figure 6a,b, we have plotted the profiles of the approximation solutions by the MSQI method at $t = 0.1$.

Table 5. Comparison of numerical solutions with the exact solutions for $u$ and $v$ at $t = 0.1$, $N = 10$, and errors are absolute errors.

<table>
<thead>
<tr>
<th>Mesh Grid</th>
<th>$u_{num}$</th>
<th>$error_u$</th>
<th>$error_u$ [16] $N = 20$</th>
<th>$v_{num}$</th>
<th>$error_v$</th>
<th>$error_v$ [16] $N = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.1, 0.1)</td>
<td>0.18367</td>
<td>$3.29366 \times 10^{-6}$</td>
<td>3.30750 $\times 10^{-6}$</td>
<td>-0.02041</td>
<td>7.46922 $\times 10^{-7}$</td>
<td>1.05384 $\times 10^{-6}$</td>
</tr>
<tr>
<td>(0.3, 0.1)</td>
<td>0.34693</td>
<td>$5.45525 \times 10^{-6}$</td>
<td>5.56160 $\times 10^{-6}$</td>
<td>0.18367</td>
<td>3.14176 $\times 10^{-6}$</td>
<td>3.10770 $\times 10^{-6}$</td>
</tr>
<tr>
<td>(0.2, 0.2)</td>
<td>0.36734</td>
<td>$6.41938 \times 10^{-6}$</td>
<td>6.61520 $\times 10^{-6}$</td>
<td>-0.04081</td>
<td>1.80110 $\times 10^{-7}$</td>
<td>2.10766 $\times 10^{-6}$</td>
</tr>
<tr>
<td>(0.4, 0.4)</td>
<td>0.53060</td>
<td>$8.79085 \times 10^{-6}$</td>
<td>8.86940 $\times 10^{-6}$</td>
<td>0.16326</td>
<td>2.16219 $\times 10^{-6}$</td>
<td>2.25400 $\times 10^{-6}$</td>
</tr>
<tr>
<td>(0.3, 0.3)</td>
<td>0.55101</td>
<td>$9.85992 \times 10^{-6}$</td>
<td>9.92330 $\times 10^{-6}$</td>
<td>-0.06122</td>
<td>3.08212 $\times 10^{-6}$</td>
<td>3.16150 $\times 10^{-6}$</td>
</tr>
<tr>
<td>(0.2, 0.4)</td>
<td>0.57142</td>
<td>$1.00279 \times 10^{-5}$</td>
<td>1.09769 $\times 10^{-5}$</td>
<td>-0.28571</td>
<td>7.37580 $\times 10^{-6}$</td>
<td>8.57700 $\times 10^{-6}$</td>
</tr>
<tr>
<td>(0.5, 0.5)</td>
<td>0.91835</td>
<td>$1.65391 \times 10^{-5}$</td>
<td>1.65386 $\times 10^{-5}$</td>
<td>-0.10204</td>
<td>5.26189 $\times 10^{-6}$</td>
<td>5.26920 $\times 10^{-6}$</td>
</tr>
</tbody>
</table>
Table 6. Comparison of numerical solutions with the exact solutions for \( u \) and \( v \) at \( t = 0.4, N = 10 \), and errors are absolute errors.

<table>
<thead>
<tr>
<th>Mesh Grid</th>
<th>( u_{\text{num}} )</th>
<th>( \text{error}_u )</th>
<th>( v_{\text{num}} )</th>
<th>( \text{error}_v )</th>
<th>( \text{error}_v ) [16]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (0.1, 0.1) )</td>
<td>0.17645</td>
<td>( 1.56636 \times 10^{-5} )</td>
<td>1.01945 ( \times 10^{-4} )</td>
<td>-0.11762</td>
<td>2.30033 ( \times 10^{-5} )</td>
</tr>
<tr>
<td>( (0.3, 0.1) )</td>
<td>0.23529</td>
<td>( 4.91795 \times 10^{-6} )</td>
<td>5.58724 ( \times 10^{-4} )</td>
<td>0.17646</td>
<td>5.73342 ( \times 10^{-4} )</td>
</tr>
<tr>
<td>( (0.2, 0.2) )</td>
<td>0.35291</td>
<td>( 3.32840 \times 10^{-5} )</td>
<td>2.03891 ( \times 10^{-4} )</td>
<td>0.11762</td>
<td>3.54833 ( \times 10^{-4} )</td>
</tr>
<tr>
<td>( (0.4, 0.2) )</td>
<td>0.41174</td>
<td>( 2.52872 \times 10^{-5} )</td>
<td>1.01945 ( \times 10^{-4} )</td>
<td>0.11762</td>
<td>1.09666 ( \times 10^{-4} )</td>
</tr>
<tr>
<td>( (0.3, 0.3) )</td>
<td>0.52936</td>
<td>( 5.66418 \times 10^{-5} )</td>
<td>3.58337 ( \times 10^{-4} )</td>
<td>-0.35284</td>
<td>1.06450 ( \times 10^{-3} )</td>
</tr>
<tr>
<td>( (0.2, 0.4) )</td>
<td>0.64701</td>
<td>( 5.11863 \times 10^{-5} )</td>
<td>4.89963 ( \times 10^{-4} )</td>
<td>-0.76460</td>
<td>1.67222 ( \times 10^{-3} )</td>
</tr>
<tr>
<td>( (0.5, 0.5) )</td>
<td>0.88225</td>
<td>( 1.03951 \times 10^{-4} )</td>
<td>5.09728 ( \times 10^{-4} )</td>
<td>-0.58804</td>
<td>1.77417 ( \times 10^{-3} )</td>
</tr>
</tbody>
</table>

From these tables, we can see that the MSQI scheme achieves an excellent approximation with the exact solutions of the equations. Though some of the results are not better than the ones in [16], the MSQI method has a simpler construction, easy implementation, smaller calculation and was less time-consuming.

### 3.2. Image Reconstruction

The digital-image-processing technique is applied more and more extensively at present, which can be seen in real-time image transmission, digital image restoration, extracting facial features, image synthesis, image compression, and encryption, and so on [35,36].

In this section, we use the multivariate spline quasi-interpolation operator \( W' \) defined by Theorem 7 to deal with problems of image reconstruction. For the testing image, we use the 2D gray image Lena [37] with pixels of \( 256 \times 256 \), which can be seen in Figure 7a.

Figure 7b is the 32,768 \( (256 \times 128) \) sampling of pixels from Lena that are used in the quasi-interpolation operator \( W' \). The reconstruction image of Lena is shown in Figure 7c. We also give the reconstruction image of Lena by using the quasi-interpolation operator \( W_{mn} \) [10,23] in \( S_2^{-1}(\Delta_{mn}) \) in Figure 8c. For the operator \( W_{mn} \), we can see [10,23] for more details. Drawings of partial enlargement of the reconstruction images with \( W' \) and \( W_{mn} \) are illustrated in Figure 8a,b respectively.
Figure 7. Results of the 2D gray image Lena. (a) original; (b) sampling pixels; (c) by $W' f$ in $S_3^3(\Delta_{mn}^{(2)})$.

Figure 8. Reconstructed image by $W_{mn}$ (a) and partial enlarged image of reconstructions by $W'$ (b) and $W_{mn}$ (c).

We also have used Barbara and Peppers 2D gray images to test the performance of the operator $W'$. The number of sampling pixels is also 32,768. The results are demonstrated in Figures 9 and 10.

Figure 9. Results of 2D gray image Barbara. (a) original; (b) sampling pixels; (c) by $W'$ in $S_3^3(\Delta_{mn}^{(2)})$. 
4. Conclusions

From the reconstruction images, we can see that the operator $W'$ has a better performance with smaller image distortion. If we give a partition $[0, mh] \times [0, nh] (m \neq n)$ for the testing images, we can use less pixels for reconstruction. At this time, image distortion of the reconstruction images may become bigger in contrast with the original ones. Note that the operator $W'$ can also serve as one technique for problems with zooming in and out of images. Other quasi-interpolation operators can also be used for these problems.

Acknowledgments: This work is partly supported by the National Natural Science Foundation of China (Nos. 11671068, 11271060) and the Fundamental Research Funds for the Central Universities (Nos. DUT16LK38).

Author Contributions: Rengui Yu and Chungang Zhu designed the research. Rengui Yu, Xianmin Hou, and Li Yin were responsible for all simulations and analyses of the results. Rengui Yu wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schoenberg, I.J. Contributions to the problem of approximation of equidistant data by analytic functions. 


