Information Matrix Algorithm of Block Based Bivariate Newton-Like Blending Rational Interpolation

Le Zou, Xiaofeng Wang*

Key Lab of Network and Intelligent Information Processing, Hefei University, Hefei 230601, Anhui, China
*Corresponding author: Xiaofeng Wang
E-mail: xfwang@iim.ac.cn

Abstract
Interpolation has wide application in signal processing, numerical integration, Computer Aided Geometric Design (CAGD), engineering technology and electrochemistry. Block based bivariate Newton-like blending rational interpolation can also be calculated based on information matrix algorithm in addition to block divided differences. The paper studied interpolation theorem, dual interpolation of block-based rational Newton interpolation and information matrix algorithm of them. Numerical example is given to illustrate the effectiveness of the method.
Key words: DUAL INTERPOLATIONS; INFORMATION MATRIX ALGORITHM; BLOCK PARTIAL DIVIED DIFFERENCES

1. Introduction

Interpolation is a hot topic in recent years. Many scholars have devoted themselves to the study of interpolation theory and its applications[1-6].The problem of constructing a continuously defined function from given discrete data is unavoidable whenever one wishes to manipulate the data in a way that requires information not included explicitly in the data. In this age of ever-increasing digitization in the storage, processing, analysis, and communication of information, it is not difficult to find examples of applications where this problem occurs. So interpolation method plays an important pole in numerical analysis. The polynomial interpolation and rational interpolation have been widely applied to the numerical approximation[1], and in the construction of circular [1], graphics image processing, image processing[1,7-10]. Some scholars has been studying their application in image processing, numerical integration, Computer Aided Geometric Design(CAGD),engineering technology, curves/surface[1,7-8]. In recent years, Dyn and Floater[11] et al studied the new multivariate polynomial interpolation based on Lower subsets. Zhao divided the original set of support points into some subsets(blocks), then construct each block by using linear or rational and finally assemble these blocks by Newton method to shape the whole interpolation scheme[1,12,13], which include many classical interpolation as its special cases. The general form of block based bivariate blending rational interpolation with the error estimation is established by introducing two parameters[14], four different block based interpolations are included. Then an efficient algorithm for computing bivariate lacunary rational interpolation is constructed based on block based bivariate blending rational interpolation. Li also studied modified Thiele-Werner blending rational interpolation[15], Werner[16] Combined The Newton interpolation with Thiele continued fractions combined to construct a flexible and stable Thiele-Werner Rational interpolation formula. Wang and Gu generalized Thiele-Werner-type blending rational interpolation to vector-valued osculatory Thiele-Werner type rational interpolation. One of the authors of this paper has constructed several interpolation general formate [18-21], which include Newton interpolation scheme, block-based rational interpolation based and Thiele-Werner-type rational interpolation and generalized the results of Kahng [22,23] to bivariate interpolation case.

Our contribution in this paper is to obtain information matrix algorithm of block based bivariate Newton like blending rational interpolation. The organization of the paper is as follows. In Section2, we give the algorithm of block based bivariate Newton-like blending rational interpolation, and we discuss information matrix algorithm of block-based bivariate Newton-like blending rational interpolation. In Section 3, we present dual interpolation of block-based rational Newton interpolation, recursive algorithm, the information matrix algorithm of block , based Newton like blending rational interpolation. Numerical example is given to show the effectiveness of the results in Section 4.

2. Information Matrix Algorithm of Block Based Bivariate Newton Like Blending Rational Interpolation

Given a set of two dimensional points $\prod_{m,n} = \{(x_i, y_j) | i = 0,1,\ldots,m; j = 0,1,\ldots,n\}$. Suppose $\prod_{m,n} \subset D \subset R^2$,and $f(x,y)$ is a function defined in D , and

$$f(x_i, y_j) = f_{i,j},\ i = 0,1,\ldots,m;\ j = 0,1,\ldots,n\,, \tag{1}$$

We devided $\prod_{m,n}$ into $(u+1)\times(v+1)$ subsets:

$$\prod_{m,n}^{s,t} = \{(x_i, y_j) | c_s \le i \le d_s, h_t \le j \le r_t\}$$
$$(s = 0,1,\ldots,u; t = 0,1,\ldots,v)\,. \tag{2}$$

Tan and Zhao constructed block based bivariate Newton like blending rational interpolation[12]

$$P_{m,n}(x,y) = Z_0(x,y) + Z_1(x,y)\omega_0(x) + \cdots + Z_u(x,y)\omega_0(x)\cdots\omega_{u-1}(x)\,, \tag{3}$$

For $s = 0,1,\ldots,u$,

$$Z_s(x,y) = I_{s,0}(x,y) + I_{s,1}(x,y)\omega_0^*(y) + \cdots + I_{s,v}(x,y)\omega_0^*(y)\cdots\omega_{v-1}^*(y)\,, \tag{4}$$

where

$$\omega_s(x) = \prod_{i=c_s}^{d_s}(x - x_i),\ s = 0,1,\ldots,u-1,\quad \omega_t^*(y) = \prod_{j=h_t}^{r_t}(y - y_j),\ t = 0,1,\ldots,v-1\,, \tag{5}$$

and $I_{s,t}(x,y)$ $(s = 0,1,\ldots,u; t = 0,1,\ldots,v)$ are bivariate polynomials or rational interpolants on the subsets $\prod_{m,n}^{s,t}$.

Block based bivariate Newton like blending rational interpolation can also be calculated based on information matrix algorithm in addition to block divided differences.

Algorithm 1

Step1: Initialization. Let

$$f_{i,j}^{(0,0)} = f(x_i, y_j) = f_{i,j}\,,\ i = 0,1,\cdots,m;\ j = 0,1,\cdots,n. \tag{6}$$

and define the following initial information matrix

$$M_1 = \begin{pmatrix}
f_{c_0,h_0}^{(0,0)} & f_{c_0,h_0+1}^{(0,0)} & \cdots & f_{c_0,r_0}^{(0,0)} & f_{c_1,h_0}^{(0,0)} & f_{c_1,h_0+1}^{(0,0)} & \cdots & f_{c_1,r_0}^{(0,0)} & \cdots & f_{c_u,h_0}^{(0,0)} & f_{c_u,h_0+1}^{(0,0)} & \cdots & f_{c_u,r_0}^{(0,0)} \\
f_{c_0+1,h_0}^{(0,0)} & f_{c_0+1,h_0+1}^{(0,0)} & \cdots & f_{c_0+1,r_0}^{(0,0)} & f_{c_1+1,h_0}^{(0,0)} & f_{c_1+1,h_0+1}^{(0,0)} & \cdots & f_{c_1+1,r_0}^{(0,0)} & \cdots & f_{c_u+1,h_0}^{(0,0)} & f_{c_u+1,h_0+1}^{(0,0)} & \cdots & f_{c_u+1,r_0}^{(0,0)} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots \\
f_{d_0,h_0}^{(0,0)} & f_{d_0,h_0+1}^{(0,0)} & \cdots & f_{d_0,r_0}^{(0,0)} & f_{d_1,h_0}^{(0,0)} & f_{d_1,h_0+1}^{(0,0)} & \cdots & f_{d_1,r_0}^{(0,0)} & \cdots & f_{d_u,h_0}^{(0,0)} & f_{d_u,h_0+1}^{(0,0)} & \cdots & f_{d_u,r_0}^{(0,0)} \\
 & & & & & & & & & & & & \\
f_{c_0,h_1}^{(0,0)} & f_{c_0,h_1+1}^{(0,0)} & \cdots & f_{c_0,r_1}^{(0,0)} & f_{c_1,h_1}^{(0,0)} & f_{c_1,h_1+1}^{(0,0)} & \cdots & f_{c_1,r_1}^{(0,0)} & \cdots & f_{c_u,h_1}^{(0,0)} & f_{c_u,h_1+1}^{(0,0)} & \cdots & f_{c_u,r_1}^{(0,0)} \\
f_{c_0+1,h_1}^{(0,0)} & f_{c_0+1,h_1+1}^{(0,0)} & \cdots & f_{c_0+1,r_1}^{(0,0)} & f_{c_1+1,h_1}^{(0,0)} & f_{c_1+1,h_1+1}^{(0,0)} & \cdots & f_{c_1+1,r_1}^{(0,0)} & \cdots & f_{c_u+1,h_1}^{(0,0)} & f_{c_u+1,h_1+1}^{(0,0)} & \cdots & f_{c_u+1,r_1}^{(0,0)} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots \\
f_{d_0,h_1}^{(0,0)} & f_{d_0,h_1+1}^{(0,0)} & \cdots & f_{d_0,r_1}^{(0,0)} & f_{d_1,h_1}^{(0,0)} & f_{d_1,h_1+1}^{(0,0)} & \cdots & f_{d_1,r_1}^{(0,0)} & & f_{d_u,h_1}^{(0,0)} & f_{d_u,h_1+1}^{(0,0)} & \cdots & f_{d_u,r_1}^{(0,0)} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \cdots & \vdots & \vdots & & \vdots \\
f_{c_0,h_v}^{(0,0)} & f_{c_0,h_v+1}^{(0,0)} & \cdots & f_{c_0,r_v}^{(0,0)} & f_{c_1,h_v}^{(0,0)} & f_{c_1,h_v+1}^{(0,0)} & \cdots & f_{c_1,r_v}^{(0,0)} & \cdots & f_{c_u,h_v}^{(0,0)} & f_{c_u,h_v+1}^{(0,0)} & \cdots & f_{c_u,r_v}^{(0,0)} \\
f_{c_0+1,h_v}^{(0,0)} & f_{c_0+1,h_v+1}^{(0,0)} & \cdots & f_{c_0+1,r_v}^{(0,0)} & f_{c_1+1,h_v}^{(0,0)} & f_{c_1+1,h_v+1}^{(0,0)} & \cdots & f_{c_1+1,r_v}^{(0,0)} & \cdots & f_{c_u+1,h_v}^{(0,0)} & f_{c_u+1,h_v+1}^{(0,0)} & \cdots & f_{c_u+1,r_v}^{(0,0)} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots \\
f_{d_0,h_v}^{(0,0)} & f_{d_0,h_v+1}^{(0,0)} & \cdots & f_{d_0,r_v}^{(0,0)} & f_{d_1,h_v}^{(0,0)} & f_{d_1,h_v+1}^{(0,0)} & \cdots & f_{d_1,r_v}^{(0,0)} & \cdots & f_{d_u,h_v}^{(0,0)} & f_{d_u,h_v+1}^{(0,0)} & \cdots & f_{d_u,r_v}^{(0,0)}
\end{pmatrix} \tag{7}$$

The above matrix can be simplified into partitioned information matrix representation

$$M_2=\begin{bmatrix} A^{(0,0)}_{c_0,d_0,h_0,r_0} & A^{(0,0)}_{c_1,d_1,h_0,r_0} & \cdots & A^{(0,0)}_{c_u,d_u,h_0,r_0} \\ A^{(0,0)}_{c_0,d_0,h_1,r_1} & A^{(0,0)}_{c_1,d_1,h_1,r_1} & \cdots & A^{(0,0)}_{c_u,d_u,h_1,r_1} \\ \vdots & \vdots & & \vdots \\ A^{(0,0)}_{c_0,d_0,h_v,r_v} & A^{(0,0)}_{c_1,d_1,h_v,r_v} & \cdots & A^{(0,0)}_{c_u,d_u,h_v,r_v} \end{bmatrix} \quad (8)$$

where

$$A^{(0,0)}_{c_s,d_s,h_t,r_t}=\begin{pmatrix} f^{(0,0)}_{c_s,h_t} & f^{(0,0)}_{c_s,h_t+1} & \cdots & f^{(0,0)}_{c_s,r_t} \\ f^{(0,0)}_{c_s+1,h_t} & f^{(0,0)}_{c_s+1,h_t+1} & \cdots & f^{(0,0)}_{c_s+1,r_t} \\ \vdots & \vdots & & \vdots \\ f^{(0,0)}_{d_s,h_t} & f^{(0,0)}_{d_s,h_t+1} & \cdots & f^{(0,0)}_{d_s,r_t} \end{pmatrix}, \quad s=0,1,\cdots,u,t=0,1,\cdots,v. \quad (9)$$

Constructing bivariate polynomial interpolation or rational interpolation function $I_{0,0}(x,y)$ on $\prod^{0,0}_{m,n}$,

that is $I_{0,0}(x_i,y_j)=f^{0,0}_{i,j}$,

Step 2: Newton type recursive along with y

For $t=0,1,\ldots,v,\ i=0,1,\ldots,m;\ j=h_t,h_t+1,\ldots,n,$

$$f^{0,t}_{i,j}=\frac{f^{0,t-1}_{i,j}-I_{0,t-1}(x_i,y_j)}{\omega^*_{t-1}(y_j)}, \quad (10)$$

where

$$\omega^*_t(y)=\prod_{j=h_t}^{r_t}(y-y_j),\ t=0,1,\ldots,v-1 \quad (11)$$

The recursive procedure is intended to transform M_2 into the following partitioned matrix by means of (row transformation).

$$M_3=\begin{bmatrix} A^{(0,0)}_{c_0,d_0,h_0,r_0} & A^{(0,0)}_{c_1,d_1,h_0,r_0} & \cdots & A^{(0,0)}_{c_u,d_u,h_0,r_0} \\ A^{(0,1)}_{c_0,d_0,h_1,r_1} & A^{(0,1)}_{c_1,d_1,h_1,r_1} & \cdots & A^{(0,1)}_{c_u,d_u,h_1,r_1} \\ \vdots & \vdots & & \vdots \\ A^{(0,v)}_{c_0,d_0,h_v,r_v} & A^{(0,v)}_{c_1,d_1,h_v,r_v} & \cdots & A^{(0,v)}_{c_u,d_u,h_v,r_v} \end{bmatrix} \quad (12)$$

where

$$A^{(0,t)}_{c_s,d_s,h_t,r_t}=\begin{pmatrix} f^{(0,t)}_{c_s,h_t} & f^{(0,t)}_{c_s,h_t+1} & \cdots & f^{(0,t)}_{c_s,r_t} \\ f^{(0,t)}_{c_s+1,h_t} & f^{(0,t)}_{c_s+1,h_t+1} & \cdots & f^{(0,t)}_{c_s+1,r_t} \\ \vdots & \vdots & & \vdots \\ f^{(0,t)}_{d_s,h_t} & f^{(0,t)}_{d_s,h_t+1} & \cdots & f^{(0,t)}_{d_s,r_t} \end{pmatrix}, s=0,1,\cdots,u,t=0,1,\cdots,v. \quad (13)$$

Constructing bivariate polynomials or rational interpolants $I_{0,t}(x,y)$ $(t=1,\ldots,v)$, on the subsets

$$\prod^{0,t}_{m,n},$$

using function value transformation results

$$A^{(0,t)}_{c_0,d_0,h_t,r_t},$$

namely

$$I_{0,t}(x_i,y_j)=f^{0,t}_{i,j},\ c_0\le i\le d_0,\ h_t\le j\le r_t,\ t=1,\ldots,v. \quad (14)$$

Step 3: Newton type recursive along with x

For $s=1,2,\ldots,u$,

$$f^{s,0}_{i,j}=\frac{f^{s-1,0}_{i,j}-Z_{s-1}(x_i,y_j)}{\omega_{s-1}(x_i)}, \quad (15)$$

$$\omega_s(x)=\prod_{i=c_s}^{d_s}(x-x_i),\ s=0,1,\ldots,u-1 \tag{16}$$

$$Z_s(x,y)=I_{s,0}(x,y)+I_{s,1}(x,y)\omega_0^*(y)+\cdots+I_{s,v}(x,y)\omega_0^*(y)\cdots\omega_{v-1}^*(y). \tag{17}$$

The recursive procedure is intended to transform M_2 into the following partitioned matrix by means of (column transformation).

$$M_4=\begin{bmatrix} A^{(0,0)}_{c_0,d_0,h_0,r_0} & A^{(1,0)}_{c_1,d_1,h_0,r_0} & \cdots & A^{(u,0)}_{c_u,d_u,h_0,r_0} \\ A^{(0,0)}_{c_0,d_0,h_1,r_1} & A^{(1,0)}_{c_1,d_1,h_1,r_1} & \cdots & A^{(u,0)}_{c_u,d_u,h_1,r_1} \\ \vdots & \vdots & & \vdots \\ A^{(0,0)}_{c_0,d_0,h_v,r_v} & A^{(1,0)}_{c_1,d_1,h_v,r_v} & \cdots & A^{(u,0)}_{c_u,d_u,h_v,r_v} \end{bmatrix} \tag{18}$$

where

$$A^{(s,0)}_{c_s,d_s,h_t,r_t}=\begin{pmatrix} f^{(s,0)}_{c_s,h_t} & f^{(s,0)}_{c_s,h_t+1} & \cdots & f^{(s,0)}_{c_s,r_t} \\ f^{(s,0)}_{c_s+1,h_t} & f^{(s,0)}_{c_s+1,h_t+1} & \cdots & f^{(s,0)}_{c_s+1,r_t} \\ \vdots & \vdots & & \vdots \\ f^{(s,0)}_{d_s,h_t} & f^{(s,0)}_{d_s,h_t+1} & \cdots & f^{(s,0)}_{d_s,r_t} \end{pmatrix},\ s=0,1,\cdots,u,t=0,1,\cdots,v. \tag{19}$$

Constructing bivariate polynomials or rational interpolant $I_{s,0}(x,y)$ $(s=1,\ldots,u)$ on the subsets

$$\prod_{m\ n}^{,0},$$

using function value transformation results

$$A^{(s,0)}_{c_s,d_s,h_0,r_0},$$

namely

$$I_{s,0}(x_i,y_j)=f^{s,0}_{i,j},$$

$$c_s\le i\le d_s,\ h_0\le j\le r_0,\ s=1,\ldots,u. \tag{20}$$

Step 4: Newton type recursive along with y

For $s=1,2,\ldots,u$, when $t=1,2,\ldots,v$,

$$f^{s,t}_{i,j}=\frac{f^{s,t-1}_{i,j}-I_{s,t-1}(x_i,y_j)}{\omega^*_{t-1}(y_j)},\ i=c_s,c_s+1,\ldots,m;\ j=h_t,h_t+1,\ldots,n \tag{21}$$

The recursive procedure is intended to transform M_4 into the following partitioned matrix by means of (row transformation).

$$M_5=\begin{bmatrix} A^{(0,0)}_{c_0,d_0,h_0,r_0} & A^{(1,0)}_{c_1,d_1,h_0,r_0} & \cdots & A^{(u,0)}_{c_u,d_u,h_0,r_0} \\ A^{(0,1)}_{c_0,d_0,h_1,r_1} & A^{(1,1)}_{c_1,d_1,h_1,r_1} & \cdots & A^{(u,1)}_{c_u,d_u,h_1,r_1} \\ \vdots & \vdots & & \vdots \\ A^{(0,v)}_{c_0,d_0,h_v,r_v} & A^{(1,v)}_{c_1,d_1,h_v,r_v} & \cdots & A^{(u,v)}_{c_u,d_u,h_v,r_v} \end{bmatrix} \tag{22}$$

where

$$A_{c_s,d_s,h_t,r_t}^{(s,t)} = \begin{pmatrix} f_{c_s,h_t}^{(s,t)} & f_{c_s,h_t+1}^{(s,t)} & \cdots & f_{c_s,r_t}^{(s,t)} \\ f_{c_s+1,h_t}^{(s,t)} & f_{c_s+1,h_t+1}^{(s,t)} & \cdots & f_{c_s+1,r_t}^{(s,t)} \\ \vdots & \vdots & & \vdots \\ f_{d_s,h_t}^{(s,t)} & f_{d_s,h_t+1}^{(s,t)} & \cdots & f_{d_s,r_t}^{(s,t)} \end{pmatrix}, s=0,1,\cdots,u, t=0,1,\cdots,v, \quad (23)$$

Step 5: Constructing bivariate polynomials or rational interpolants $I_{s,t}(x,y)$, on the subsets $\prod_{m,n}^{s,t}$ using function value transformation results $A_{c_s,d_s,h_t,r_t}^{(s,t)}$ $(s=1,2,\ldots,u; t=1,2,\ldots,v)$, namely

$$I_{s,t}(x_i,y_j) = f_{i,j}^{s,t}, \; c_s \le i \le d_s, h_t \le j \le r_t, s=1,2,\ldots,u; t=1,2,\ldots,v. \quad (24)$$

Constructing bivariate Newton-like polynomials $Z_s(x,y)$ about y using $I_{s,t}(x,y)$, namely

$$Z_s(x,y) = I_{s,0}(x,y) + I_{s,1}(x,y)\omega_0^*(y) + \cdots + I_{s,t}(x,y)\omega_0^*(y)\cdots\omega_{t-1}^*(y), \quad s=0,1,\cdots,u, \quad (25)$$

Step 6: Let

$$P_{m,n}(x,y) = Z_0(x,y) + Z_1(x,y)\omega_0(x) + \cdots + Z_s(x,y)\omega_0(x)\cdots\omega_{s-1}(x), \quad (26)$$

Then $P_{m,n}(x,y)$ is the block based bivariate Newton-like blending interpolant on the whole set.

Theorem 1. If all the above interpolants $I_{s,t}(x,y)$ $(s=1,2,\ldots,u; t=1,2,\ldots,v)$ satisfy (21) and (24) exist. Then

$$P_{m,n}(x_i,y_j) = f_{i,j}, \; i=0,1,\ldots,m; j=0,1,\ldots,n. \quad (27)$$

Proof: Suppose $c_s \le i \le d_s, h_t \le j \le r_t$. From (25)-(26) we can get

$$P_{m,n}(x_i,y_j) = Z_0(x_i,y_j) + Z_1(x_i,y_j)\omega_0(x_i) + \cdots + Z_s(x_i,y_j)\omega_0(x_i)\cdots\omega_{s-1}(x_i)$$

and

$$Z_s(x_i,y_j) = I_{s,0}(x_i,y_j) + I_{s,1}(x_i,y_j)\omega_0^*(y_j) + \cdots + I_{s,t}(x_i,y_j)\omega_0^*(y_j)\cdots\omega_{t-1}^*(y_j),$$

From (21) and (24) we can get

$$\begin{aligned} I_{s,t}(x_i,y_j)\omega_0^*(y_j)\cdots\omega_{t-1}^*(y_j) &= f_{i,j}^{s,t}\omega_0^*(y_j)\cdots\omega_{t-1}^*(y_j) \\ &= (f_{i,j}^{s,t-1} - I_{s,t-1}(x_i,y_j))\omega_0^*(y_j)\cdots\omega_{t-2}^*(y_j), \end{aligned}$$

Then we recursively get

$$\begin{aligned} Z_s(x_i,y_j) &= I_{s,0}(x_i,y_j) + I_{s,1}(x_i,y_j)\omega_0^*(y_j) + \cdots + I_{s,t}(x_i,y_j)\omega_0^*(y_j)\cdots\omega_{t-1}^*(y_j) \\ &= I_{s,0}(x_i,y_j) + I_{s,1}(x_i,y_j)\omega_0^*(y_j) + \cdots + (f_{i,j}^{s,t-1} - I_{s,t-1}(x_i,y_j))\omega_0^*(y_j)\cdots\omega_{t-2}^*(y_j) \\ &= I_{s,0}(x_i,y_j) + I_{s,1}(x_i,y_j)\omega_0^*(y_j) + \cdots + f_{i,j}^{s,t-1}\omega_0^*(y_j)\cdots\omega_{t-2}^*(y_j) \\ &= \cdots = f_{i,j}^{s,0}, \end{aligned}$$

From (21) it follows

$$f_{i,j}^{s,0}\omega_0(x_i)\cdots\omega_{s-1}(x_i) = (f_{i,j}^{s-1,0} - Z_{s-1}(x_i,y_j))\omega_0(x_i)\cdots\omega_{s-2}(x_i),$$

It is easy to drive recursively

$$\begin{aligned} P_{m,n}(x_i,y_j) &= Z_0(x_i,y_j)+Z_1(x_i,y_j)\omega_0(x_i)+\cdots+Z_s(x_i,y_j)\omega_0(x_i)\cdots\omega_{s-1}(x_i) \\ &= Z_0(x_i,y_j)+Z_1(x_i,y_j)\omega_0(x_i)+\cdots+f_{i,j}^{s,0}\omega_0(x_i)\cdots\omega_{s-1}(x_i) \\ &= Z_0(x_i,y_j)+Z_1(x_i,y_j)\omega_0(x_i)+\cdots \\ &\quad +(f_{i,j}^{s-1,0}-Z_{s-1}(x_i,y_j))\omega_0(x_i)\cdots\omega_{s-2}(x_i) \\ &= Z_0(x_i,y_j)+Z_1(x_i,y_j)\omega_0(x_i)+\cdots+f_{i,j}^{s-1,0}\omega_0(x_i)\cdots\omega_{s-2}(x_i) \\ &= \cdots = f_{i,j}^{0,0} = f_{i,j}. \end{aligned}$$

3. Information Matrix Algorithm of the Dual Interpolation of Block Based Bivariate Newton-Like Blending Rational Interpolation

With the help of the function value of the block information matrix, we can get information matrix algorithm of the dual interpolation of block based bivariate Newton like blending rational interpolation.

Algorithm 2

Step 1: Initializing. Let

$$f_{i,j}^{(0,0)} = f(x_i,y_j) = f_{i,j},$$

$$i=0,1,\cdots,m;\ j=0,1,\cdots,n. \tag{28}$$

the initial value information matrix (7) is represented by a block matrix (8) - (9).

Constructing bivariate polynomial interpolation or rational interpolation function $I_{0,0}(x,y)$ based on

$$\prod_{m,n}^{0,0},$$

namely $I_{0,0}(x_i,y_j)=f_{i,j}^{0,0}$,

Step 2: Newton type recursive along with x.

For $s=0,1,\ldots,u$, $i=c_s,c_s+1,\ldots,m;\ j=0,1,\ldots,n,$

$$f_{i,j}^{s,0} = \frac{f_{i,j}^{s-1,0}-I_{s-1,0}(x_i,y_j)}{\omega_{s-1}(x_i)}, \tag{29}$$

where

$$\omega_s(x)=\prod_{i=c_s}^{d_s}(x-x_i),\ s=0,1,\ldots,u-1 \tag{30}$$

The recursive procedure is intended to transform M_2 into the following partitioned matrix by means of (column transformation).

$$M_6 = \begin{bmatrix} A^{(0,0)}_{c_0,d_0,h_0,r_0} & A^{(1,0)}_{c_1,d_1,h_0,r_0} & \cdots & A^{(u,0)}_{c_u,d_u,h_0,r_0} \\ A^{(0,0)}_{c_0,d_0,h_1,r_1} & A^{(1,0)}_{c_1,d_1,h_1,r_1} & \cdots & A^{(u,0)}_{c_u,d_u,h_1,r_1} \\ \vdots & \vdots & & \vdots \\ A^{(0,0)}_{c_0,d_0,h_v,r_v} & A^{(1,0)}_{c_1,d_1,h_v,r_v} & \cdots & A^{(u,0)}_{c_u,d_u,h_v,r_v} \end{bmatrix} \tag{31}$$

where

$$A^{(s,0)}_{c_s,d_s,h_t,r_t} = \begin{pmatrix} f^{(s,0)}_{c_s,h_t} & f^{(s,0)}_{c_s,h_t+1} & \cdots & f^{(s,0)}_{c_s,r_t} \\ f^{(s,0)}_{c_s+1,h_t} & f^{(s,0)}_{c_s+1,h_t+1} & \cdots & f^{(s,0)}_{c_s+1,r_t} \\ \vdots & \vdots & & \vdots \\ f^{(s,0)}_{d_s,h_t} & f^{(s,0)}_{d_s,h_t+1} & \cdots & f^{(s,0)}_{d_s,r_t} \end{pmatrix},\ s=0,1,\cdots,u,\ t=0,1,\cdots,v. \tag{32}$$

Constructing bivariate polynomials or rational interpolants $I_{s,0}(x,y)$ $(s=1,\ldots,u)$ on the subsets

$$\prod_{m,n}^{s,0},$$

using function value transformation results

$$A^{(s,0)}_{c_s,d_s,h_0,r_0},$$

namely

$$I_{s,0}(x_i,y_j)=f_{i,j}^{s,0},$$

$$c_s\le i\le d_s,\ h_0\le j\le r_0,\ s=1,\ldots,u. \tag{33}$$

Step 3: Newton type recursive along with y

For $t=1,\ldots,v$,

$$f_{i,j}^{0,t}=\frac{f_{i,j}^{0,t}-S_{t-1}(x_i,y_j)}{\omega^*_{t-1}(y_j)},\ i=0,1,\ldots,m;\ j=h_t,h_t+1,\ldots,n, \quad (34)$$

$$\omega_t^*(y)=\prod_{j=h_t}^{r_t}(y-y_j),\ t=0,1,\ldots,v-1, \quad (35)$$

$$S_t(x,y)=I_{0,t}(x,y)+I_{1,t}(x,y)\omega_0(x)+\cdots+I_{v,t}(x,y)\omega_0(x)\cdots\omega_{v-1}(x). \quad (36)$$

The recursive procedure is intended to transform M_2 into the following partitioned matrix by means of (row transformation).

$$M_7=\begin{bmatrix} A^{(0,0)}_{c_0,d_0,h_0,r_0} & A^{(0,0)}_{c_1,d_1,h_0,r_0} & \cdots & A^{(0,0)}_{c_u,d_u,h_0,r_0} \\ A^{(0,1)}_{c_0,d_0,h_1,r_1} & A^{(0,1)}_{c_1,d_1,h_1,r_1} & \cdots & A^{(0,1)}_{c_u,d_u,h_1,r_1} \\ \vdots & \vdots & & \vdots \\ A^{(0,v)}_{c_0,d_0,h_v,r_v} & A^{(0,v)}_{c_1,d_1,h_v,r_v} & \cdots & A^{(0,v)}_{c_u,d_u,h_v,r_v} \end{bmatrix} \quad (37)$$

where

$$A^{(0,t)}_{c_s,d_s,h_t,r_t}=\begin{pmatrix} f^{(0,t)}_{c_s,h_t} & f^{(0,t)}_{c_s,h_t+1} & \cdots & f^{(0,t)}_{c_s,r_t} \\ f^{(0,t)}_{c_s+1,h_t} & f^{(0,t)}_{c_s+1,h_t+1} & \cdots & f^{(0,t)}_{c_s+1,r_t} \\ \vdots & \vdots & & \vdots \\ f^{(0,t)}_{d_s,h_t} & f^{(0,t)}_{d_s,h_t+1} & \cdots & f^{(0,t)}_{d_s,r_t} \end{pmatrix},\ s=0,1,\cdots,u,t=0,1,\cdots,v. \quad (38)$$

Constructing bivariate polynomials or rational interpolants $I_{0,t}(x,y)$ $(t=1,2,\ldots,v)$, on the subsets

$$\prod^{0,t}_{m,n}$$

using function value transformation results

$$A^{(0,t)}_{c_0,d_0,h_t,r_t},$$

namely

$$I_{0,t}(x_i,y_j)=f^{0,t}_{i,j},$$

$$c_0\le i\le d_0,\ h_t\le j\le r_t,\ t=1,2,\ldots,v. \quad (39)$$

Step 4: Newton type recursive along with x

For $t=1,2,\ldots,v$, when $s=1,2,\ldots,u$,

$$f^{s,t}_{i,j}=\frac{f^{s-1,t}_{i,j}-I_{s-1,t}(x_i,y_j)}{\omega_{s-1}(x_i)},\ i=c_s,c_s+1,\ldots,m;\ j=h_t,h_t+1,\ldots,n \quad (40)$$

The recursive procedure is intended to transforme M_7 into the following partitioned matrix by means of (column transformation).

$$M_8=\begin{bmatrix} A^{(0,0)}_{c_0,d_0,h_0,r_0} & A^{(1,0)}_{c_1,d_1,h_0,r_0} & \cdots & A^{(u,0)}_{c_u,d_u,h_0,r_0} \\ A^{(0,1)}_{c_0,d_0,h_1,r_1} & A^{(1,1)}_{c_1,d_1,h_1,r_1} & \cdots & A^{(u,1)}_{c_u,d_u,h_1,r_1} \\ \vdots & \vdots & & \vdots \\ A^{(0,v)}_{c_0,d_0,h_v,r_v} & A^{(1,v)}_{c_1,d_1,h_v,r_v} & \cdots & A^{(u,v)}_{c_u,d_u,h_v,r_v} \end{bmatrix} \quad (41)$$

where

$$A^{(s,t)}_{c_s,d_s,h_t,r_t}=\begin{pmatrix} f^{(s,t)}_{c_s,h_t} & f^{(s,t)}_{c_s,h_t+1} & \cdots & f^{(s,t)}_{c_s,r_t} \\ f^{(s,t)}_{c_s+1,h_t} & f^{(s,t)}_{c_s+1,h_t+1} & \cdots & f^{(s,t)}_{c_s+1,r_t} \\ \vdots & \vdots & & \vdots \\ f^{(s,t)}_{d_s,h_t} & f^{(s,t)}_{d_s,h_t+1} & \cdots & f^{(s,t)}_{d_s,r_t} \end{pmatrix}, s=0,1,\cdots,u, t=0,1,\cdots,v, \quad (42)$$

Step 5: Constructing bivariate polynomials or rational interpolants $I_{s,t}(x,y)$ on $\prod^{s,t}_{m,n}$, namely

$$I_{s,t}(x_i,y_j)=f^{s,t}_{i,j}, (c_s\le i\le d_s, h_t\le j\le r_t, s=1,2,\ldots,u; t=1,2,\ldots,v). \quad (43)$$

Constructing bivariate Newton-like polynomials $S_t(x,y)$ about x using $I_{s,t}(x,y)$, namely

$$S_t(x,y)=I_{0,t}(x,y)+I_{1,t}(x,y)\omega_0(x)+\cdots+I_{s,t}(x,y)\omega_0(x)\cdots\omega_{s-1}(x). \quad (44)$$

Step 6:let

$$DP_{m,n}(x,y)=S_0(x,y)+S_1(x,y)\omega^*_0(y)+\cdots+S_t(x,y)\omega^*_0(y)\cdots\omega^*_{t-1}(y) \quad (45)$$

Then $DP_{m,n}(x,y)$ is the dual interpolation of block based bivariate Newton-like blending interpolant on the whole set $P_{m,n}(x,y)$.

Theorem 2 If all the above interpolants $I_{s,t}(x,y)$ $(s=1,2,\ldots,u; t=0,1,\ldots,v)$ satisfying (40),(43) exist, then

$$DP_{m,n}(x_i,y_j)=P_{m,n}(x_i,y_j)=f_{i,j}, i=0,1,\ldots,m; j=0,1,\ldots,n. \quad (46)$$

proof: from theorem 1 we can get

$$P_{m,n}(x_i,y_j)=f_{i,j}$$

suppose $c_s\le i\le d_s, h_t\le j\le r_t$.

From (44) and (45), we can get

$$DP_{m,n}(x_i,y_j)=S_0(x_i,y_j)+S_1(x_i,y_j)\omega^*_0(y_j)+\cdots+S_t(x_i,y_j)\omega^*_0(y_j)\cdots\omega^*_{t-1}(y_j)$$

$$S_t(x_i,y_j)=I_{0,t}(x_i,y_j)+I_{1,t}(x_i,y_j)\omega_0(x_i)+\cdots+I_{s,t}(x_i,y_j)\omega_0(x_i)\cdots\omega_{s-1}(x_i).$$

From (40) and (43) we can drive

$$\begin{aligned}I_{s,t}(x_i,y_j)\omega_0(x_i)\cdots\omega_{s-1}(x_i)&=f^{s,t}_{i,j}\omega_0(x_i)\cdots\omega_{s-1}(x_i)\\&=(f^{s-1,t}_{i,j}-I_{s-1,t}(x_i,y_j))\omega_0(x_i)\cdots\omega_{s-2}(x_i),\end{aligned}$$

Then we recursively get

$$\begin{aligned}S_t(x_i,y_j)&=I_{0,t}(x_i,y_j)+I_{1,t}(x_i,y_j)\omega_0(x_i)+\cdots+I_{s,t}(x_i,y_j)\omega_0(x_i)\cdots\omega_{s-1}(x_i)\\&=I_{0,t}(x_i,y_j)+I_{1,t}(x_i,y_j)\omega_0(x_i)+\cdots+(f^{s-1,t}_{i,j}-I_{s-1,t}(x_i,y_j))\omega_0(x_i)\cdots\omega_{s-2}(x_i)\\&=I_{0,t}(x_i,y_j)+I_{1,t}(x_i,y_j)\omega_0(x_i)+\cdots+f^{s-1,t}_{i,j}\omega_0(x_i)\cdots\omega_{s-2}(x_i)\\&=\cdots=f^{0,t}_{i,j},\end{aligned}$$

From (34) it follows

$$f^{0,t}_{i,j}\omega^*_0(y_j)\cdots\omega^*_{t-1}(y_j)=(f^{0,t-1}_{i,j}-S_{t-1}(x_i,y_j))\omega^*_0(y_j)\cdots\omega^*_{t-2}(y_j),$$

It is easy to drive recursively

$$\begin{aligned}DP_{m,n}(x_i,y_j)&=S_0(x_i,y_j)+S_1(x_i,y_j)\omega^*_0(y_j)+\cdots+S_t(x_i,y_j)\omega^*_0(y_j)\cdots\omega^*_{t-1}(y_j)\\&=S_0(x_i,y_j)+S_1(x_i,y_j)\omega^*_0(y_j)+\cdots+f^{0,t}_{i,j}\omega^*_0(y_j)\cdots\omega^*_{t-1}(y_j)\\&=S_0(x_i,y_j)+S_1(x_i,y_j)\omega^*_0(y_j)+\cdots+(f^{0,t-1}_{i,j}-S_{t-1}(x_i,y_j))\omega^*_0(y_j)\cdots\omega^*_{t-2}(y_j)\\&=S_0(x_i,y_j)+S_1(x_i,y_j)\omega^*_0(y_j)+\cdots+f^{0,t-1}_{i,j}\omega^*_0(y_j)\cdots\omega^*_{t-2}(y_j)\\&=\cdots=f^{0,0}_{i,j}=f_{i,j}.\end{aligned}$$

4. Numerical Example

In this section, we take a simple example to show how the algorithms are implemented and how flexible our method is.

Example Assume that the interpolation points and the function values at the support nodes (x_i, y_j) are given as follows

Table 1. Interpolation data

	$y_0 = 0$	$y_1 = 1$	$y_2 = 2$	$y_3 = 3$
$x_0 = 0$	4	5	-1	6
$x_1 = 1$	3	7	2	0
$x_2 = 2$	5	3	1	2
$x_3 = 3$	1	2	-1	4

For convenience, we merely present a few schemes. Let

$$c_0 = 0, d_0 = 2, c_1 = d_1 = 3, h_0 = 0, r_0 = 2, h_1 = r_1 = 3,$$

i.e., $\prod_{33}$ is divided into the following 4 subsets $\prod_{3,3}^{0,0}, \prod_{3,3}^{0,1}, \prod_{3,3}^{1,0}, \prod_{3,3}^{1,1}$

$$\begin{matrix} (0,0) & (0,1) & (0,2) & (0,3) \\ (1,0) & (1,1) & (1,2) & (1,3) \\ (2,0) & (2,1) & (2,2) & (2,3) \\ \\ (3,0) & (3,1) & (3,2) & (3,3) \end{matrix} \quad (47)$$

Scheme 1 Suppose $I_{0,1}(x,y)$, $I_{1,0}(x,y)$ are the Thiele-type continued fractions based on $\prod_{3,3}^{0,1}, \prod_{3,3}^{1,0}$, and $I_{0,0}(x,y)$, $I_{1,1}(x,y)$ are the bivariate Newton interpolation polynomial based on $\prod_{3,3}^{0,0}, \prod_{3,3}^{1,1}$ respectively, then we can get

$$P_{1,1}(x,y) = 4 - x + y + 3xy + \frac{3}{2}x(x-1) - \frac{7}{2}y(y-1) - \frac{9}{2}x(x-1)y - xy(y-1) + \frac{11}{4}x(x-1)y(y-1)$$
$$+\frac{26x-60}{x-18}y(y-1)(y-2) + [\frac{3}{4y-2} - \frac{4}{15}y(y-1)(y-2)]x(x-1)(x-2)$$

Scheme 2 information matrix algorithm of block based bivariate Newton polynomial interpolation

Suppose $I_{0,1}(x,y)$, $I_{1,0}(x,y)$ are the Thiele-type continued fractions based on $\prod_{3,3}^{0,1}, \prod_{3,3}^{1,0}$, and $I_{0,0}(x,y)$, $I_{1,1}(x,y)$ are the bivariate Newton interpolation polynomial based on $\prod_{3,3}^{0,0}, \prod_{3,3}^{1,1}$ respectively, using the algorithm 1, we can get

$$P_2(x,y) = I_{0,0}(x,y) + I_{0,1}(x,y)\omega_0^*(y) + (I_{1,0}(x,y) + I_{1,1}(x,y)\omega_0^*(y))\omega_0(x)$$
$$= 4 - x + y + 3xy + \frac{3}{2}x(x-1) - \frac{7}{2}y(y-1) - \frac{9}{2}x(x-1)y$$
$$- xy(y-1) + \frac{11}{4}x(x-1)y(y-1) + \frac{26x-60}{x-18}y(y-1)(y-2)$$
$$+[\frac{3}{4y-2} - \frac{4}{15}y(y-1)(y-2)]x(x-1)(x-2)$$

Scheme 3 information matrix algorithm of dual interpolation of block based bivariate Newton polynomial interpolation

Suppose $I_{0,1}(x,y)$, $I_{1,0}(x,y)$ are the Thiele-type continued fractions based on $\prod_{3,3}^{0,1}, \prod_{3,3}^{1,0}$, and $I_{0,0}(x,y)$, $I_{1,1}(x,y)$ are the bivariate Newton interpolation polynomial based on $\prod_{3,3}^{0,0}, \prod_{3,3}^{1,1}$ respectively, using the algorithm 2 we can get

$$P_4(x,y) = I_{0,0}(x,y) + I_{1,0}(x,y)\omega_0(x) + (I_{0,1}(x,y) + I_{1,1}(x,y)\omega_0(x))\omega^*{}_0(y)$$
$$= 4 - x + y + 3xy + \frac{3}{2}x(x-1) - \frac{7}{2}y(y-1) - \frac{9}{2}x(x-1)y$$
$$- xy(y-1) + \frac{11}{4}x(x-1)y(y-1) + \frac{3}{4y-2}x(x-1)(x-2)$$
$$+[\frac{26x-60}{x-18} - \frac{4}{15}x(x-1)(x-2)]y(y-1)(y-2)$$

5. Conclusions

In this paper, we disscuss the information matrix algorithm for the block based Newton-like blending rational interpolation and its dual interpolation. Algorithm 1 and 2 have the advantages of using the concept of linear algebra of elementary transformation, but here the row and column transform are block partial divided difference. In addition, the algorithm results 1 and 2 are neat and beautiful. From Theorem 1 and 2, we know block coefficient can be gotten by the s th column element $f(x_i) = f_i, i = 0,1,\ldots,n$ of the matrix M_5, M_8, this is convenient for the practical problems.

One can generalize the results to bivariate composite schemes by the method in the paper[25] and one can drive the vector-valued and matrix-valued interpolation by Samelson[26].

Acknowledgements

The authors would like to express their thanks to the referees for their valuable suggestions. This work was Supported by the grant of Anhui Provincial Natural Science Foundation, Nos.1508085QF116,1308085MF84, the grant of the National Natural Science Foundation of China,Nos.61272024,61005010, the grant of Support Key Project for Excellent Young Talent in College of Anhui Province, No.gxyqZD2016269, the grant of Support Project for Excellent Young Talent in College of Anhui Province (X.F. Wang), the Scientific Research Major Foundation of Education Department of Anhui Province, Nos.KJ2014ZD30, KJ2015A206, Open Project Foundation of Intelligent Information Processing Key Laboratory of Shanxi Province No.201401, Training Object for Academic Leader of Hefei University, No.2014dtr08, Key Constructive Discipline of Hefei University, No.2014xk08.

References

1. Tan J. Q. (2007) *Theory of Continued Fractions and its Applications*, Beijing: Science Publishers.
2. Wang R.H., Zhu G.Q. (2004) *Approximation of Rational Interpolation and its Application*, Science Publishers.
3. Hildebrand F. B.(1974) *Introduction to Numerical Analysis,* New York: McGraw-Hill.
4. Zou L., Wang X.F., Chen Y.P. (2015) An approach of Associated Continued Fractions Blending Rational Interpolation. *Metallurgical and Mining Industry*, 7(6), p.p.361-367.
5. Kuchmins'ka Kh. I., Sus O. M., Vozna S. M. (2003) Approximation Properties of Two-Dimensional Continued Fractions. *Ukrainian Mathematical Journal*, 55(1), p.p.36-54.
6. Varsamis D. N., Karampetakis N. P. (2014) On the Newton Bivariate Polynomial Interpolation with Applications. *Multidim System Sign Process*, 25(1), p.p.179–209.
7. Frank GL.(1999) Thiele Rational Interpolation for the Numerical Computation of the Reversible Randles Sevcik Function in Electrochemistry. *Journal of Scientific Computing*, 14(3), p.p. 259-274
8. Huo X., Tan J.Q., He L, Hu M. (2014) An Automatic Video Scratch Removal Based on Thiele type Continued Fraction. *Multimedia Tools and Applications*, 71(2), p.p.451-467.
9. Bai T., Tan J. Q., Hu M. (2014) A Novel Algorithm for Removal of Salt and Pepper Noise using Continued Fractions Interpolation. *Signal Processing*, 102 ,p.p.247-255.
10. Bai T., Tan J. Q., Hu M. (2014) Compression and Reconstruction of time series data Based on Vector Valued Continued Fraction. *Journal of Information & Computational Science*,11(8), p.p.2459-2466.
11. Dyn N., Floater M. S. (2014) Multivariate Polynomial Interpolation on Lower Sets. *Journal of Approximation Theory*,177, p.p.34-42.
12. Zhao Q. J. , Tan J. Q. (2006) Block-based Newton-like Blending Rational Interpolation. *Journal of Computational Mathematics,* 24(3), p.p.515-526.
13. Zhao Q. J., Tan J. Q. (2006) Block-Based Thiele-Like Blending Rational Interpolation. *Journal of Computational and Applied Mathematics*, 195(1), p.p.312-325.
14. Li C.W., Zhu X.L. (2008) Block Based Bivariate Blending Rational Interpolation *Journal of Hefei Univiversity Technology*, 31(2), p.p.484-488.
15. Li C.W., Zhu X.L., Zou L. (2010) Modified Thiele-Werner Rational Interpolation. *Journal of Mathematical Research & Exposition*, 30(4), p.p.653-663.
16. Werner H. (1979) A Reliable Method for Rational Interpolation, in: L.Wuytack, eds., Padé Approximation and its Applications, Springer, Berlin, p.p.257-277.
17. Wang J.B., Gu C.Q. (2004) Vector Valued Thiele-Werner-type Osculatory Rational Interpolants. *Journal of Computational and Applied Mathematics*, 163(1), p.p.241-252.
18. Zou L., Tang S. (2014) A New Approach to General Interpolation Formulae for Bivariate Interpolation. *Abstract and Applied Analysis*, Article ID 421635,11 pages, Vol.2014. doi:10.1155/2014/421635.
19. Zou L., Tang S. (2012) General Structures of Block Based Interpolational Function. *Communications in Mathematical Research,* 28(3), p.p.193-208.
20. Tang S., Zou L. (2009) A Note on General Frames for Bivariate Interpolation. *Journal of Mathematical Research & Exposition*, 29(4), p.p.700-706,
21. Zou L., Tang S. (2011) Note on General Frames of Symmetry Interpolation. *Communication on Applied Mathematics and Computation,* 25(1), p.p.111-118.

22. Kahng S.W. (1967) Generalized Newton's Interpolation Functions and Their Applications to Chebyshev Approximations, Lockheed Electronics Company Report, 1967.
23. Kahng S.W. (1969) Osculatory Interpolation. *Mathematics of Computation.* 23(107), p.p.621-629.
24. Zhu G. Q., Tan J.Q. (1995) The Duality of Bivariate Vector Valued Ratioanl Interpolants over Rectangular Grids. *Computational Mathematics*,3, p.p. 311- 320.
25. Tan J.Q., Tang S. (2002) Composite schemes for multivariate blending raitonal interpolation. *Journal of Computational and Applied Mathematics*, p.p.144,263–275.
26. Zhu X. L., Zhu G. Q. (1999) A Note on Matrix-valued Rational Interpolations. *Journal of Computational and Applied Mathematics*, 110(1), p.p.129-140.