Journal of Computational Mathematics, Vol.24, No.4, 2006, 515–526.

BLOCK BASED NEWTON-LIKE BLENDING INTERPOLATION*[1)]

Qian-jin Zhao
(*School of Computer and Information, Hefei University of Technology, Hefei 230009, China Department of Mathematics and Physics, Anhui University of Science and Technology, Huainan 232001, China*)

Jie-qing Tan
(*Institute of Applied Mathematics, Hefei University of Technology, Hefei 230009, China*)

Abstract

Newton's polynomial interpolation may be the favourite linear interpolation in the sense that it is built up by means of the divided differences which can be calculated recursively and produce useful intermediate results. However Newton interpolation is in fact point based interpolation since a new interpolating polynomial with one more degree is obtained by adding a new support point into the current set of support points once at a time. In this paper we extend the point based interpolation to the block based interpolation. Inspired by the idea of the modern architectural design, we first divide the original set of support points into some subsets (blocks), then construct each block by using whatever interpolation means, linear or rational and finally assemble these blocks by Newton's method to shape the whole interpolation scheme. Clearly our method offers many flexible interpolation schemes for choices which include the classical Newton's polynomial interpolation as its special case. A bivariate analogy is also discussed and numerical examples are given to show the effectiveness of our method.

Mathematics subject classification: 41A20, 65D05.
Key words: Interpolation, Block based divided differences, Blending method.

1. Introduction

Denote by Π_n the set of all real or complex polynomials $p(x)$ with degree not exceeding n. Let $S_n = \{(x_i, f_i), i = 0, 1, \ldots, n\}$ be a set of support points, where the support abscissae $x_i, i = 0, 1, \ldots, n$, do not have to be distinct from one another. Then an interpolating polynomial $P_n(x)$ in Π_n can be uniquely determined by S_n. Suppose the support ordinates $f_i, i = 0, 1, \ldots, n$, are the values of a given function $f(x)$ which is defined on the set $I(I \supset X_n)$, here $X_n = \{x_i, i = 0, 1, \ldots, n\}$. Then $P_n(x)$ satisfying $P_n(x_i) = f(x_i), i = 0, 1, \ldots, n$, possesses the following Newton representation ([6])

$$\begin{aligned} P_n(x) &= f[x_0] + f[x_0, x_1](x - x_0) + \cdots \\ &+ f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}), \end{aligned}$$

where $f[x_0, x_1, \ldots, x_i]$ are the divided differences of $f(x)$ at support abscissae $x_0, x_1, \ldots, x_i$, which are defined by the recursions

$$f[x_{i_1}] = f(x_{i_1}),$$

* Received May 20, 2005; Final revised November 9, 2005.
1) Project supported by the National Natural Science Foundation of China under Grant No. 10171026 and No. 60473114, and the Anhui Provincial Natural Science Foundation, China under Grant No. 03046102.

$$f[x_{i_1}, x_{i_2}] = \frac{f(x_{i_1}) - f(x_{i_2})}{x_{i_1} - x_{i_2}},$$
$$f[x_{i_1}, x_{i_2}, \dots, x_{i_k}] = \frac{f[x_{i_1}, \dots, x_{i_{k-2}}, x_{i_k}] - f[x_{i_1}, \dots, x_{i_{k-2}}, x_{i_{k-1}}]}{x_{i_k} - x_{i_{k-1}}}.$$

We want to mention that Newton interpolating polynomials have their nonlinear counterparts, the Thiele's interpolating continued fractions, which are built up on the basis of the inverse differences. The Thiele's continued fraction interpolating the support points S_n is of the following form

$$R_n(x) = f(x_0) + \frac{x - x_0}{\lceil a_1} + \frac{x - x_1}{\lceil a_2} + \cdots + \frac{x - x_{n-1}}{\lceil a_n},$$

where for $i = 1, 2, \dots, n$,

$$a_i = \phi[x_0, x_1, \dots, x_i]$$

is the inverse difference of $f(x)$ at $x_0, x_1, \dots, x_i$, which can be computed recursively as follows

$$\begin{aligned}
\phi[x_i] &= f(x_i),\ i = 0, 1, \dots, n, \\
\phi[x_i, x_j] &= \frac{x_i - x_j}{f(x_i) - f(x_j)}, \\
\phi[x_i, x_j, x_k] &= \frac{x_k - x_j}{\phi[x_i, x_k] - \phi[x_i, x_j]}, \\
\phi[x_i, \dots, x_j, x_k, x_l] &= \frac{x_l - x_k}{\phi[x_i, \dots, x_j, x_l] - \phi[x_i, \dots, x_j, x_k]}.
\end{aligned}$$

It is easy to verify that $R_n(x)$ is a rational function with degrees of numerator and denominator polynomials bounded by $[(n+1)/2]$ and $[n/2]$ respectively, where $[x]$ denotes the greatest integer not exceeding x, and $R_n(x)$ satisfies

$$R_n(x_i) = f(x_i),\ i = 0, 1, \dots, n.$$

One of the authors ([10]) established an extraordinary variety of rational interpolants by applying the Neville's algorithm to continued fractions. One may say that Newton interpolation is point based interpolation since a new interpolating polynomial with one more degree is obtained by adding a new support point into the current set of support points once at a time. In this paper we try to extend the point based interpolation to the block based one. The idea can be summarized into three steps: first we divide the original set of support points into some subsets (blocks), then construct each block by using whatever interpolation means, linear or rational and finally assemble these blocks by Newton's method to shape the whole interpolation scheme.

2. Block Blending Interpolation

As we see, the classical Newton's polynomial interpolation is a point based interpolation. Undoubtedly the Newton's interpolating polynomial established on the whole set X_n largely reduces the flexibility of the interpolation and lacks interactivity in the sense that the interpolant is completely dominated by the original set of supporting points. To obtain a flexible blending rational interpolation, we extend the point based interpolation to the block based one.

2.1 Basic idea

We divide the set X_n into $u + 1$ subsets:

$$\{x_{c_0}, x_{c_0+1,\dots,}x_{d_0}\}, \dots, \{x_{c_u}, x_{c_u+1,\dots,}x_{d_u}\}.$$

These subsets may be obtained by reordering the interpolation points if necessary. Obviously, we have

$$\sum_{s=0}^{u}(d_s - c_s + 1) = n + 1.$$

Let us consider the following function with Newton-like formation:

$$T(x) = I_0(x) + I_1(x)\omega_0(x) + \cdots + I_u(x)\omega_0(x)\cdots\omega_{u-1}(x), \tag{1}$$

where

$$\omega_s(x) = \prod_{i=c_s}^{d_s}(x - x_i), s = 0, 1, \ldots, u - 1, \tag{2}$$

and $I_s(x)(s = 0, 1, \ldots, u)$ are polynomials or rational interpolating functions on the subsets $\{x_{c_s}, x_{c_s+1}, \ldots, x_{d_s}\}(s = 0, 1, \ldots, u)$.
If the above $I_s(x)(s = 0, 1, \ldots, u)$ are chosen so that

$$T(x_i) = f(x_i), x_i \in X_n, \tag{3}$$

then $T(x)$ defined by (1) and (2) is called the block based Newton-like blending interpolant to $f(x)$.

2.2 Block based divided differences

Suppose $X_n \subset I \subset R$, and let $f(x)$ be a real function defined on I such that

$$f(x_i) = f_i, i = 0, 1, \ldots, n. \tag{4}$$

We introduce the following notations:

$$f_i^0 = f_i, i = 0, 1, \ldots, n \tag{5}$$

and for $s = 1, 2, \ldots, u$,

$$f_i^s = \frac{f_i^{s-1} - I_{s-1}(x_i)}{\omega_{s-1}(x_i)}, i = c_s, c_s + 1, \ldots, n, \tag{6}$$

where $I_s(x)(s = 0, 1, \ldots, u)$ are interpolating polynomials or rational interpolating functions on the subsets $\{x_{c_s}, x_{c_s+1}, \ldots, x_{d_s}\}(s = 0, 1, \ldots, u)$, which satisfy

$$I_s(x_i) = f_i^s, i = c_s, c_s + 1, \ldots, d_s; s = 0, 1, \ldots, u. \tag{7}$$

If all these f_i^s exist, they are called the sth block based divided differences for function $f(x)$.

Theorem 1. *If all the above interpolants $I_s(x)$ which satisfy (7) exist, then*

$$T(x_i) = f_i, i = 0, 1, \ldots, n.$$

Proof. For each $i \in \{0, 1, \ldots, n\}$, there is a certain $s \in \{0, 1, \ldots, u\}$ which satisfies $c_s \leq i \leq d_s$. By (1), we have

$$T(x_i) = I_0(x_i) + I_1(x_i)\omega_0(x_i) + \cdots + I_s(x_i)\omega_0(x_i)\cdots\omega_{s-1}(x_i).$$

According to (7) and (6), we obtain

$$\begin{aligned} I_s(x_i)\omega_0(x_i)\cdots\omega_{s-1}(x_i) &= f_i^s\omega_0(x_i)\cdots\omega_{s-1}(x_i) \\ &= \frac{f_i^{s-1}-I_{s-1}(x_i)}{\omega_{s-1}(x_i)}\omega_0(x_i)\cdots\omega_{s-1}(x_i) \\ &= (f_i^{s-1}-I_{s-1}(x_i))\omega_0(x_i)\cdots\omega_{s-2}(x_i). \end{aligned}$$

Then we recursively get

$$\begin{aligned} T(x_i) &= I_0(x_i)+I_1(x_i)\omega_0(x_i)+\cdots+I_s(x_i)\omega_0(x_i)\cdots\omega_{s-1}(x_i) \\ &= I_0(x_i)+I_1(x_i)\omega_0(x_i)+\cdots+(f_i^{s-1}-I_{s-1}(x_i))\omega_0(x_i)\cdots\omega_{s-2}(x_i) \\ &= I_0(x_i)+I_1(x_i)\omega_0(x_i)+\cdots+f_i^{s-1}\omega_0(x_i)\cdots\omega_{s-2}(x_i) \\ &= \cdots = f_i^0 = f_i. \end{aligned}$$

2.3 Special cases

Case 1. If all the $I_s(x)(s=0,1,\ldots,u)$ are the interpolating polynomials $P_s(x)$ on the subsets $\{x_{c_s}, x_{c_s+1}, \ldots, x_{d_s}\}(s=0,1,\ldots,u)$, then the block based Newton-like blending interpolant becomes the classical interpolating polynomial on the whole set X_n:

$$T(x) = P_0(x)+P_1(x)\omega_0(x)+\cdots+P_u(x)\omega_0(x)\cdots\omega_{u-1}(x). \tag{8}$$

It is easy to verify (see [11])

$$\partial P_u(x) = d_u - c_u,$$

$$\partial\omega_s(x) = d_s - c_s + 1, (s=0,1,\ldots,u-1),$$

and

$$\partial T(x) = \sum_{s=0}^{u}(d_s-c_s+1)-1 = n,$$

where $\partial F(x)$ denotes the degree of the polynomial $F(x)$.
Obviously we have

$$T(x) = P_n(x),$$

where the $P_n(x)$ is the Newton interpolating polynomial on the whole set X_n.
Case 2. If all the $I_s(x)(s=0,1,\ldots,u)$ are the Thiele-type interpolating continued fractions $R_s(x)$ on the subsets $\{x_{c_s}, x_{c_s+1}, \ldots, x_{d_s}\}(s=0,1,\ldots,u)$, then we obtain

$$T(x) = R_0(x)+R_1(x)\omega_0(x)+\cdots+R_u(x)\omega_0(x)\cdots\omega_{u-1}(x). \tag{9}$$

In particular case when $u=n$, i.e., each subset contains only one point, all the block based divided differences become the classical divided differences and the block based Newton-like blending interpolant also becomes the classical Newton interpolating polynomial on the whole set X_n.
Especially when $u=0$, the whole set X_n is a unique subset.Then we have

$$T(x) = R_0(x), \tag{10}$$

where $R_0(x)$ is the Thiele-type interpolating continued fraction on the whole set X_n, which means that the classical Thiele-type continued fraction interpolation is also a special case of the block based Newton-like blending interpolation when the whole set X_n is a unique subset.

2.4 Error estimation

We turn now to a discussion of the error in the approximation of a function $f(x)$ by its block based Newton-like blending interpolants.

Theorem 2. *Suppose* $[a,b]$ *is the smallest interval containing* $X_n = \{x_0, x_1, \ldots, x_n\}$ *and* $f(x)$ *is differentiable in* $[a,b]$ *up to* $(n+1)$ *times. Let*

$$T(x) = I_0(x) + I_1(x)\omega_0(x) + \cdots + I_u(x)\omega_0(x)\cdots\omega_{u-1}(x) = \frac{P(x)}{Q(x)}. \tag{11}$$

Then for each $x \in [a,b]$ *there exists a point* $\xi \in (a,b)$ *such that*

$$f(x) - T(x) = \frac{\omega(x)}{Q(x)} \cdot \frac{[f(x)Q(x) - P(x)]^{(n+1)}_{x=\xi}}{(n+1)!}, \tag{12}$$

where $\omega(x) = \prod_{i=0}^{n}(x - x_i)$.

Proof. Let $E(x) = f(x)Q(x) - P(x)$. Then from Theorem 1 and (11) it follows

$$E(x_i) = 0, (i = 0, 1, \ldots, n).$$

Using the Newton interpolation formula (see [11]), we have

$$\begin{aligned} E(x) &= \sum_{i=0}^{n} E[x_0, x_1, \ldots, x_i](x - x_0)\cdots(x - x_{i-1}) \\ &+ (x - x_0)\cdots(x - x_n) \cdot \frac{E^{(n+1)}(\xi)}{(n+1)!} \\ &= \frac{\omega(x)E^{(n+1)}(\xi)}{(n+1)!}, \end{aligned}$$

where $\xi \in (a,b)$.
It is easy to verify

$$\begin{aligned} f(x) - T(x) &= \frac{E(x)}{Q(x)} = \frac{\omega(x)E^{(n+1)}(\xi)}{Q(x)(n+1)!} \\ &= \frac{\omega(x)}{Q(x)} \cdot \frac{[f(x)Q(x) - P(x)]^{(n+1)}_{x=\xi}}{(n+1)!}. \end{aligned}$$

2.5 Numerical examples

In this section, we take a simple example to show the flexibility and the effectiveness of our method.

Example 1. Let $X_5 = \{0, 1, 2, 3, 4, 5\}$ and $\{f_0, f_1, f_2, f_3, f_4, f_5\} = \{1, 2, 2, 0, 1, 2\}$.
According to the block based Newton-like blending interpolation method illustrated in the preceding sections, one can yield the following schemes for interpolants.

Scheme 1: By the case 1 in section 2.3 and considering that the whole set X_5 is a unique subset, we have

$$\begin{aligned} T(x) &= 1 + x - \frac{x(x-1)}{2} - \frac{x(x-1)(x-2)}{6} \\ &+ \frac{x(x-1)(x-2)(x-3)}{4} - \frac{7x(x-1)(x-2)(x-3)(x-4)}{60} \\ &= 1 - \frac{47x}{15} + \frac{103}{12}x^2 - \frac{23}{4}x^3 + \frac{17}{12}x^4 - \frac{7}{60}x^5, \end{aligned}$$

and it is easy to verify

$$T(x_i) = f_i, i = 0, 1, 2, 3, 4, 5.$$

Scheme 2: We divide X_5 into two subsets $X_5^0 = \{0, 1, 2\}$ and $X_5^1 = \{3, 4, 5\}$.
Let $I_0(x)$ be the Newton interpolating polynomial and $I_1(x)$ be the Thiele-type interpolating continued fraction. Then one has

$$\begin{aligned} T(x) &= 1 + x - \frac{x(x-1)}{2} + \frac{5x-16}{42x-120} x(x-1)(x-2) \\ &= \frac{-120 - 170x + 181x^2 - 52x^3 + 5x^4}{42x - 120}. \end{aligned}$$

It is easy to verify

$$T(x_i) = f_i, i = 0, 1, 2, 3, 4, 5.$$

Scheme 3: We divide X_5 into two subsets $X_5^0 = \{0, 1, 2, 3\}$ and $X_5^1 = \{4, 5\}$.
Let $I_0(x)$ be the Thiele-type interpolating continued fraction and $I_1(x)$ be the Newton interpolating polynomial. Then we have

$$T(x) = \frac{180 - 294x + 954x^2 - 730x^3 + 215x^4 - 26x^5 + x^6}{180 - 30x}$$

and it is easy to verify

$$T(x_i) = f_i, i = 0, 1, 2, 3, 4, 5.$$

3. Multivariate Case

3.1 Basic idea

The block based Newton-like blending interpolation method can be generalized to the multivariate case.
Given a set of two dimensional points $\Pi_{mn} = \{(x_i, y_j) \mid i = 0, 1, \ldots, m; j = 0, 1, \ldots, n\}$. Suppose $\Pi_{mn} \subset D \subset R^2$, and let $f(x, y)$ be a real function defined on D such that

$$f(x_i, y_j) = f_{ij}, i = 0, 1, \ldots, m; j = 0, 1, \ldots, n. \tag{13}$$

We divide Π_{mn} into $(u+1) \times (v+1)$ subsets:
$\Pi_{mn}^{st} = \{(x_i, y_j) \mid c_s \le i \le d_s; h_t \le j \le r_t\} (s = 0, 1, \ldots, u; t = 0, 1, \ldots, v)$.
Let us consider the following function with bivariate Newton-like formation:

$$T(x, y) = Z_0(x, y) + Z_1(x, y)\omega_0(x) + \cdots + Z_u(x, y)\omega_0(x) \cdots \omega_{u-1}(x), \tag{14}$$

where for $s = 0, 1, \ldots, u$,

$$Z_s(x, y) = I_{s0}(x, y) + I_{s1}(x, y)\omega_0^*(y) + \cdots + I_{sv}(x, y)\omega_0^*(y) \cdots \omega_{v-1}^*(y), \tag{15}$$

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

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

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

To obtain a block based bivariate Newton-like blending interpolant on the whole set Π_{mn}, one has to compute $I_{st}(x,y)(s=0,1,\ldots,u;t=0,1,\ldots,v)$ so that the function (14) satisfies:

$$T(x_i,y_j)=f_{ij}, i=0,1,\ldots,m; j=0,1,\ldots,n. \tag{16}$$

3.2 Block based bivariate partial divided differences

We introduce the following notations:

$$f_{ij}^{00}=f_{ij}, i=0,1,\ldots,m; j=0,1,\ldots,n. \tag{17}$$

For $t=1,2,\cdots,v$,

$$f_{ij}^{0t}=\frac{f_{ij}^{0,t-1}-I_{0,t-1}(x_i,y_j)}{\omega_{t-1}^*(y_j)}, (i=0,1,\ldots,m; j=h_t,h_t+1,\ldots,n), \tag{18}$$

where $I_{0t}(x,y)(t=0,1,\ldots,v)$ are bivariate polynomials or rational interpolants on the subsets Π_{mn}^{0t}, namely

$$I_{0t}(x_i,y_j)=f_{ij}^{0t}, (c_0\le i\le d_0, h_t\le j\le r_t, t=0,1,\ldots,v). \tag{19}$$

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

$$f_{ij}^{s0}=\frac{f_{ij}^{s-1,0}-Z_{s-1}(x_i,y_j)}{\omega_{s-1}(x_i)}, (i=c_s,c_s+1,\ldots,m; j=0,1,\ldots,n). \tag{20}$$

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

$$f_{ij}^{st}=\frac{f_{ij}^{s,t-1}-I_{s,t-1}(x_i,y_j)}{\omega_{t-1}^*(y_j)}, \quad (i=c_s,c_s+1,\ldots,m; j=h_t,h_t+1,\ldots,n), \tag{21}$$

where $I_{st}(x,y)(s=1,2,\ldots,u;t=0,1,\ldots,v)$ are bivariate polynomials or rational interpolants on the subsets Π_{mn}^{st}, namely

$$\begin{aligned} I_{st}(x_i,y_j)=f_{ij}^{st}, (\quad & c_s\le i\le d_s, h_t\le j\le r_t;\\ & s=1,2,\ldots,u; t=0,1,\ldots,v). \end{aligned} \tag{22}$$

If all f_{ij}^{st} exist, f_{ij}^{st} are called the (s,t)th block based bivariate partial divided differences for function $f(x,y)$.

Theorem 3. *If all the above interpolants $I_{st}(x,y)$ satisfying (19) and (22) exist, then*

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

Proof. Suppose $c_s\le i\le d_s$, and $h_t\le j\le r_t$. By (14) and (22), we have

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

and

$$\begin{aligned} Z_s(x_i,y_j) \quad &= \quad I_{s0}(x_i,y_j)+I_{s1}(x_i,y_j)\omega_0^*(y_j)+\cdots+I_{st}(x_i,y_j)\cdot\\ & \qquad \omega_0^*(y_j)\cdots\omega_{t-1}^*(y_j). \end{aligned}$$

From (22) and (21), it follows

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

Then we recursively obtain

$$\begin{aligned}
Z_s(x_i,y_j) &= I_{s0}(x_i,y_j) + I_{s1}(x_i,y_j)\omega_0^*(y_j) + \cdots + I_{st}(x_i,y_j)\cdot\\
&\quad \omega_0^*(y_j)\cdots\omega_{t-1}^*(y_j)\\
&= I_{s0}(x_i,y_j) + I_{s1}(x_i,y_j)\omega_0^*(y_j) + \cdots + (f_{ij}^{s,t-1} - I_{s,t-1}(x_i,y_j))\cdot\\
&\quad \omega_0^*(y_j)\cdots\omega_{t-2}^*(y_j)\\
&= I_{s0}(x_i,y_j) + I_{s1}(x_i,y_j)\omega_0^*(y_j) + \cdots + f_{ij}^{s,t-1}\omega_0^*(y_j)\cdots\omega_{t-2}^*(y_j)\\
&= \cdots = f_{ij}^{s0}.
\end{aligned}$$

From (20), it follows

$$f_{ij}^{s0}\omega_0(x_i)\cdots\omega_{s-1}(x_i) = (f_{ij}^{s-1,0} - Z_{s-1}(x_i,y_j))\omega_0(x_i)\cdots\omega_{s-2}(x_i).$$

It is easy to obtain recursively

$$\begin{aligned}
T(x_i,y_j) &= Z_0(x_i,y_j) + Z_1(x_i,y_j)\omega_0(x_i) + \cdots + Z_s(x_i,y_j)\cdot\\
&\quad \omega_0(x_i)\cdots\omega_{s-1}(x_i)\\
&= Z_0(x_i,y_j) + Z_1(x_i,y_j)\omega_0(x_i) + \cdots + f_{ij}^{s0}\omega_0(x_i)\cdots\omega_{s-1}(x_i)\\
&= Z_0(x_i,y_j) + Z_1(x_i,y_j)\omega_0(x_i) + \cdots + (f_{ij}^{s-1,0} - Z_{s-1}(x_i,y_j))\cdot\\
&\quad \omega_0(x_i)\cdots\omega_{s-2}(x_i)\\
&= Z_0(x_i,y_j) + Z_1(x_i,y_j)\omega_0(x_i) + \cdots + f_{ij}^{s-1,0}\omega_0(x_i)\cdots\omega_{s-2}(x_i)\\
&= \cdots = f_{ij}^{00} = f_{ij.}
\end{aligned}$$

3.3 Error estimation

We turn now to a discussion of the error in the approximation of a function $f(x,y)$ by its block based bivariate Newton-like blending interpolants.
It is easy to verify the following Theorem 4 in terms of bivariate Newton interpolation formula (see[11]).

Theorem 4. *Suppose $D = [a,b]\times[c,d]$ is a rectangular domain containing Π_{mn} and $f(x,y)\in C^{(m+n+2)}(D)$. Let*
$T(x,y) = Z_0(x,y) + Z_1(x,y)\omega_0(x) + \cdots + Z_u(x,y)\omega_0(x)\cdots\omega_{u-1}(x) = \frac{P(x,y)}{Q(x,y)}$
be the block based bivariate Newton-like blending interpolant on Π_{mn}. Then $\forall (x,y)\in D$ we have

$$\begin{aligned}
f(x,y) - T(x,y) &= \frac{\omega(x)}{Q(x,y)}\cdot\frac{\frac{\partial^{m+1}}{\partial x^{m+1}}[fQ-P]_{x=\xi}}{(m+1)!}\\
&\quad +\frac{\omega^*(y)}{Q(x,y)}\cdot\frac{\frac{\partial^{n+1}}{\partial y^{n+1}}[fQ-P]_{y=\eta}}{(n+1)!}\\
&\quad -\frac{\omega(x)\omega^*(y)}{Q(x,y)}\cdot\frac{\frac{\partial^{n+m+2}}{\partial x^{m+1}\partial y^{n+1}}[fQ-P]_{x=\overline{\xi},y=\overline{\eta}}}{(m+1)!(n+1)!},
\end{aligned}\tag{23}$$

with $\xi, \overline{\xi} \in (a,b)$ *and* $\eta, \overline{\eta} \in (c,d)$,
where

$$\omega(x) = (x-x_0)(x-x_1)\cdots(x-x_m),$$
$$\omega^*(y) = (y-y_0)(y-y_1)\cdots(y-y_n).$$

3.4 Numerical examples

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

Example 2. Suppose the interpolating points and the prescribed values of $f(x,y)$ at the support abscissas (x_i, y_j) are given in the following table

	$y_0 = 0$	$y_1 = 1$	$y_2 = 2$	$y_3 = 3$
$x_0 = 0$	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 Π_{33}^{00}, Π_{33}^{01},Π_{33}^{10} and Π_{33}^{11}

(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)

Scheme 1: Suppose $I_{00}(x,y)$ is the bivariate Thiele-type branched continued fraction interpolant $R_{00}(x,y)$ on the subset Π_{33}^{00}, $I_{01}(x,y)$ is the bivariate Newton interpolating polynomial $P_{01}(x,y)$ on the subset Π_{33}^{01}, $I_{10}(x,y)$ is the bivariate Newton interpolating polynomial $P_{10}(x,y)$ on the subset Π_{33}^{10} and $I_{11}(x,y)$ is also the bivariate Newton interpolating polynomial $P_{11}(x,y)$ on the subset Π_{33}^{11}. Then we have

$$\begin{aligned} T(x,y) &= 4 + \frac{y}{-\frac{7}{5}y + \frac{12}{5}} + \frac{x}{-1 + \frac{y}{\frac{5}{6}y - \frac{1}{6}} + \frac{x-1}{\frac{1}{3} + \frac{7y}{19y-26}}} \\ &+ \left(\frac{11269}{10068}x^2 - \frac{4535}{1678}x + \frac{11}{18}\right)y(y-1)(y-2) \\ &+ x(x-1)(x-2)\left[\left(-\frac{3}{20}y^2 + \frac{9}{20}y - \frac{3}{5}\right) - \frac{2880061}{9000792}y(y-1)(y-2)\right]. \end{aligned}$$

It is easy to see

$$T(x_i, y_j) = f_{ij}, i,j = 0,1,2,3.$$

Scheme 2: Suppose $I_{00}(x,y)$ is the bivariate Newton interpolating polynomial $P_{00}(x,y)$ on the subset Π_{33}^{00}, $I_{01}(x,y)$ is the Thiele-type branched continued fraction interpolant $R_{01}(x,y)$ on the subset Π_{33}^{01}, $I_{10}(x,y)$ is the bivariate Newton interpolating polynomial $P_{10}(x,y)$ on the subset Π_{33}^{10} and $I_{11}(x,y)$ is also the bivariate Newton interpolating polynomial $P_{11}(x,y)$ on the subset Π_{33}^{11}. Then we have

$$T(x,y) = 4 - x + y + 3xy + \frac{3}{2}x(x-1) - \frac{7}{2}y(y-1) - \frac{9}{2}x(x-1)y$$

$$
\begin{aligned}
& -xy(y-1)+\frac{11}{4}x(x-1)y(y-1)+\frac{26x-60}{x-18}y(y-1)(y-2) \\
& +[-\frac{3}{2}+5y-2y^2+\frac{8}{15}y(y-1)(y-2)]x(x-1)(x-2) \\
= & \frac{P(x,y)}{Q(x,y)},
\end{aligned}
$$

where

$$
\begin{aligned}
P(x,y) = & -4320+6180x-12060y+408xy^3+14580y^2-6810x^2 \\
& -21864xy+6936xy^2+28889x^2y-15291x^2y^2-672x^3y^3 \\
& +4701x^3y^2-8079x^3y-3600y^3+1980x^3-90x^4+1792x^2y^3 \\
& +32x^4y^3-216x^4y^2+364x^4y,
\end{aligned}
$$

$$Q(x,y)=60x-1080,$$

and it is easy to see

$$T(x_i,y_j)=f_{ij}, i,j=0,1,2,3.$$

Scheme 3: Suppose $I_{00}(x,y)$ is the bivariate Newton interpolating polynomial $P_{00}(x,y)$ on the subset Π_{33}^{00}, $I_{01}(x,y)$ is the bivariate Newton interpolating polynomial $P_{01}(x,y)$ on the subset Π_{33}^{01}, $I_{10}(x,y)$ is the Thiele-type branched continued fraction interpolant $R_{10}(x,y)$ on the subset Π_{33}^{10} and $I_{11}(x,y)$ is the bivariate Newton interpolating polynomial $P_{11}(x,y)$ on the subset Π_{33}^{11}. Then we have

$$
\begin{aligned}
T(x,y) = & 4-x+y+3xy+\frac{3}{2}x(x-1)-\frac{7}{2}y(y-1)-\frac{9}{2}x(x-1)y \\
& -xy(y-1)+\frac{11}{4}x(x-1)y(y-1) \\
& +(\frac{10}{3}-\frac{5}{4}x-\frac{1}{12}x^2)y(y-1)(y-2) \\
& +[\frac{3}{4y-2}-\frac{49}{180}y(y-1)(y-2)]x(x-1)(x-2) \\
= & \frac{P(x,y)}{Q(x,y)},
\end{aligned}
$$

where

$$
\begin{aligned}
P(x,y) = & 720-990x+570y-911xy^3-6450y^2+1080x^2+2279xy \\
& -2464xy^2-1581x^2y+2181x^2y^2-264x^2y^4+98x^3y^4 \\
& -343x^3y^3+343x^3y^2-98x^3y+646xy^4 \\
& +5460y^3-270x^3-1200y^4-66x^2y^3,
\end{aligned}
$$

$$Q(x,y)=180-360y,$$

and it is easy to verify

$$T(x_i,y_j)=f_{ij}, i,j=0,1,2,3.$$

Scheme 4: Suppose $I_{00}(x,y)$ is the bivariate Newton interpolating polynomial $P_{00}(x,y)$ on the subset Π_{33}^{00}, $I_{01}(x,y)$ is the Thiele-type branched continued fraction interpolant $R_{01}(x,y)$ on the subset Π_{33}^{01}, $I_{10}(x,y)$ is the Thiele-type branched continued fraction interpolant $R_{10}(x,y)$

on the subset Π_{33}^{10} and $I_{11}(x,y)$ is the bivariate Newton interpolating polynomial $P_{11}(x,y)$ on the subset Π_{33}^{11}. Then we have

$$\begin{aligned} T(x,y) &= 4-x+y+3xy+\frac{3}{2}x(x-1)-\frac{7}{2}y(y-1)-\frac{9}{2}x(x-1)y \\ &\quad -xy(y-1)+\frac{11}{4}x(x-1)y(y-1) \\ &\quad +\frac{26x-60}{x-18}y(y-1)(y-2) \\ &\quad +[\frac{3}{4y-2}-\frac{4}{15}y(y-1)(y-2)]x(x-1)(x-2) \\ &= \frac{P(x,y)}{Q(x,y)}, \end{aligned}$$

where

$$\begin{aligned} P(x,y) &= -4320+6180x-3420y+7272xy^3+38700y^2-6810x^2 \\ &\quad -13488xy+12648xy^2+10253x^2y-13933x^2y^2+1792x^2y^4 \\ &\quad -672x^3y^4+2022x^3y^3-1317x^3y^2+57x^3y-4272xy^4 \\ &\quad -32760y^3+1980x^3+7200y^4-90x^4+118x^2y^3+32x^4y^4 \\ &\quad -112x^4y^3+112x^4y^2-32x^4y, \end{aligned}$$

$$Q(x,y)=-120xy+60x+2160y-1080,$$

and it is easy to show

$$T(x_i,y_j)=f_{ij}, i,j=0,1,2,3.$$

4. Conclusion and Future Work

In this paper we present a new kind of block based Newton-like blending interpolants which can be obtained by the Newton's method. There is no doubt that the above method provides us with flexible interpolation schemes for choices which include the classical Newton's polynomial interpolation as its special case. We give a brief discussion of block based Newton-like blending interpolation algorithm, its recursive characteristic properties and error estimation. A bivariate analogy is also discussed. Our future work will be focused on the following aspects:

- How to divide the set X_n and how to choose interpolation method on the every subset to obtain better approximation.
- How to generalize the above block based interpolation to other formation to obtain better approximation.
- Applications of block based interpolation in image processing.

We conclude this paper by pointing out that it is not difficult to generalize the block based Newton-like blending interpolation method to vector-valued case or matrix-valued case ([4, 8, 12]).

Acknowledgement. The authors would like to thank the referees for their helpful suggestions.

References

[1] C. Chaffy, Fractions continues á deux variables et calcul formel, Technical Report, IMAG (1986).

[2] A. Cuyt and B. Verdonk, Multivariate rational interpolation, *Computing*, **34** (1985), 41–61.

[3] A. Cuyt and B. Verdonk, A review of branched continued fraction theory for the construction of multivariate rational approximants, *Appl. Numer. Math.*, **4** (1988), 263–271.

[4] P.R. Graves-Morris, Vector valued rational interpolants I, *Numer. Math.*, **42** (1983), 331–348.

[5] W. Siemaszko, Thiele-type branched continued fractions for two variable functions, *J. Comput. Appl. Math.*, **9** (1983), 137–153.

[6] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, Second Edition, Springer–Verlag, 1992.

[7] J. Tan, Bivariate blending rational interpolants, *Approx. Theory* & *Its Appl.*, **15**:2 (1999), 74–83.

[8] J. Tan and G. Zhu, General framework for vector-valued interpolants, in: Proc. of the 3rd China-Japan Seminar on Numerical Mathematics, Z. Shi ed., Science Press, Beijing/New York, 1998, 273–278.

[9] J. Tan and Y. Fang, Newton-Thiele's rational interpolants, *Numerical Algorithms*, **24** (2000), 141–157.

[10] J. Tan and P. Jiang, A Neville-like method via continued fractions, *J. Comput. Appl. Math.*, **163** (2004), 219–232.

[11] R.H. Wang, Numerical Approximation, Higher Education Press, Beijing, 1999 (in Chinese).

[12] G. Zhu and J. Tan, A note on matrix-valued rational interpolants, *J. Comput. Appl. Math.*, **110** (1999), 129–140.