A New Interpolation Method by NTP Curves and Surfaces

Gang Liu, Guo-Jin Wang

Department of Mathematics, Zhejiang University, Hangzhou, 310027, China
State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, 310027, China
*wanggg@zju.edu.cn

Abstract: In this paper, we research on constructing an explicit parametric curve to be taken as the limitation curve of Progressive Iteration Approximation (PIA) which can interpolate some scattered data points by using normalized totally positive (NTP) basis. We prove that when the data points and corresponding parameter list are given, the limitation curve has the unique explicit expression. There is a similar conclusion also for surface case. By specially choosing two kinds of NTP bases, Said-Bézier type generalized Ball (SBGB) basis and DP basis, based on the formula for the inverse matrix of Vandermonde matrix, we deduce an explicit matrix solutions of the limitation curve and surface corresponding to these NTP bases. Our results avoid the tedious calculation of the inverse matrix and hence will gain extensive application in reverse engineering.

Keywords: Normalized Totally Positive basis; Said-Bézier type generalized Ball basis; DP basis; Progressive Iteration Approximation; Interpolation

I. INTRODUCTION

Since an effective rational cubic curve was developed by Ball et al. in the CONSURF system in 1974 [1], a mass of treatises on the construction of its high degree generalized forms has published. In 1987 and 1989, Wang-Ball curve and Said-Ball curve [2] [3] emerged as the times require respectively. In 1996, Hu et al. [4] gave some algorithms for degree elevation, degree reduction and recursive evaluation of both Wang-Ball and Said-Ball curves, and pointed out that all recursive evaluations of Wang-Ball and Said-Ball curves are faster than that of Bézier curve. In detail, the evaluation time complexity of Wang-Ball curve declines to linear, however, Said-Ball curve does not. In 2000, Wu [5] defined two new families of generalized Ball curves with position parameters, which are Said-Bézier generalized Ball curve and Wang-Said generalized Ball curve, abbreviated as SBGB curve and WSBG curve respectively. SBGB curve take Said-Ball and Bézier curves as its two special cases and WSBG curve take Wang-Ball and Said-Ball curves as its special cases. They possess many common properties of both generalized Ball curves and Bézier curve. In 2004, Wu [6] gave the dual bases of these two basis families as well as the Marsden identities.

Generally speaking, in addition to pursuing quickness of evaluation, shape designers also require that parametric curve preserves the shape of the control polygon as far as possible. Theoretically speaking, shape-preserving is closely related to normalized totally positive (Abbreviated as NTP) property of the basis. Delgado and Peña [9] in 2003, called as DP basis by some researchers later, gives a positive response. DP basis not only is provided with NTP property, but also has linear time complexity of evaluation algorithm. It makes up for the shortage of both Wang-Ball basis and Said-Ball basis more or less. Later, Jiang and Wang [10] researched some transformation formulae between DP basis and Bernstein basis.

To sum up, SBGB curve and DP curve are all NTP curves with excellent properties. In order to apply these curves in reverse engineering, we often need to seek proper control points to blend SBGB basis or DP basis and then generate corresponding parametric curve to interpolate given data points, so does parametric surface. It is to do this that is the theme of our manuscript. There have been plentiful general research works to focus on approximation and interpolation. Recently, the progressive iteration approximation (Abbreviated as PIA) method which utilizes NTP basis to generate limitation curve and surface interpolating given scattered data points attracts researchers’ great attention, the corresponding early and up to date researchers include Qi et al. [13], de Boor [14], Lin et al. [15] [16], Lu [17], Lin [18], Chen et al. [19], etc. It should be noted that Delgado and Peña [20] compared the PIA convergence rates of different NTP bases and proved that the convergence of normalized B-basis is fastest.

This paper deduces the PIA limitation form of NTP curve and surface, and proves the uniqueness of the limitation curve and surface. In the meantime, by deducing or summing up many transformation matrices from SBGB basis to power basis, DP basis to Bernstein basis, and Bernstein basis to power basis, by utilizing the explicit inverse matrix algorithm of Vandermonde matrix, we obtain matrix expressions of PIA limitation curve and surface when interpolation basis is SBGB basis or DP basis. The examples are used to demonstrate the validity and rationality of our algorithm in the end.

II. NTP BASIS AND ITS TRANSFORMATION WITH POWER BASIS

A. SBGB Basis and Its Transformation

Definition 1 [5] In general, SBGB basis is defined as

\[ \alpha_i(t; n, K) = \begin{cases} \binom{n/2}{i} K + i \end{cases} \alpha(t, \cdot, \cdot) \alpha(t, \cdot, \cdot), \]

\[ 0 \leq i \leq \left[ \frac{n/2}{2} \right] - K - 1; \]

\[ \alpha_i(t; n, K) = \begin{cases} \binom{n}{i} (1-t)^{n-i} \alpha(t, \cdot, \cdot), \end{cases} \]

\[ \left[ \frac{n/2}{2} \right] - K \leq i \leq \left[ \frac{n/2}{2} \right]; \]

\[ \alpha_{n-i}(1-t; n, K), \]

\[ \left[ \frac{n}{2} \right] + 1 \leq i \leq n \]
in which \( \lceil n/2 \rceil \) and \( \lfloor n/2 \rfloor \) are the maximum integer not more than \( n/2 \) and the minimum integer not less than \( n/2 \), respectively. SBGB basis blending the control points \( \{ p_i \}_{i=0}^n \) generates a degree \( n \) Said-Bézier type generalized Ball (Abbreviated as SBGB) curve:

\[
P(t; n, K) = \sum_{i=0}^n \alpha_i(t; n, K)p_i, \quad 0 \leq t \leq 1.
\]

From the definition, the curves \( P(t; n, 0), P\left(t; n, \lfloor n/2 \rfloor\right) \) are Said-Ball curve and Bézier curve respectively. When \( 1 \leq K \leq \lfloor n/2 \rfloor - 1 \), \( P(t; n, K) \) is located between them.

Theorem 1 \([8]\) SBGB basis is an NTP basis.

**Lemma 1** The transpose matrix of the matrix \( F_{n,K}^{n,K} \) can transform SBGB basis to Power basis, i.e.,

\[
(1, t, ..., t^n) = \left( \alpha_0(t; n, K), \alpha_1(t; n, K), ..., \alpha_n(t; n, K) \right)(F_{n,K}^{n,K})^T.
\]

Here

\[
f_{i'} = \begin{cases} \sum_{i=s}^{n-i} \left( \begin{array}{c} i+1 \\ i \end{array} \right) \left( \begin{array}{c} \lfloor n/2 \rfloor + j \\\ n \end{array} \right) \left( \begin{array}{c} j+1 \\ j \end{array} \right) - f(0), & 0 \leq j \leq \lfloor n/2 \rfloor - K - 1; \\
\sum_{i=s}^{n-i} \left( \begin{array}{c} i+1 \\ i \end{array} \right) \left( \begin{array}{c} j+1 \\ j \end{array} \right) - f(0), & \lfloor n/2 \rfloor - K \leq j \leq \lfloor n/2 \rfloor; \\
\sum_{i=s}^{n-i} \left( \begin{array}{c} i+1 \\ i \end{array} \right) \left( \begin{array}{c} j+1 \\ j \end{array} \right) - f(1), & \lfloor n/2 \rfloor + 1 \leq i \leq \lfloor n/2 \rfloor + K; \\
\sum_{i=s}^{n-i} \left( \begin{array}{c} i+1 \\ i \end{array} \right) \left( \begin{array}{c} j+1 \\ j \end{array} \right) - f(1), & \lfloor n/2 \rfloor + K + 1 \leq i \leq n.
\end{cases}
\]

Proof: Simplifying the dual functionals of SBGB basis mentioned in Ref. \([6]\), we can get

\[
f(t) = \begin{cases} \sum_{i=s}^{n-i} \left( \begin{array}{c} i+1 \\ i \end{array} \right) f(0), & 0 \leq i \leq \lfloor n/2 \rfloor - K - 1; \\
\sum_{i=s}^{n-i} \left( \begin{array}{c} i+1 \\ i \end{array} \right) f(0), & \lfloor n/2 \rfloor - K \leq i \leq \lfloor n/2 \rfloor; \\
\sum_{i=s}^{n-i} \left( \begin{array}{c} i+1 \\ i \end{array} \right) f(1), & \lfloor n/2 \rfloor + 1 \leq i \leq \lfloor n/2 \rfloor + K; \\
\sum_{i=s}^{n-i} \left( \begin{array}{c} i+1 \\ i \end{array} \right) f(1), & \lfloor n/2 \rfloor + K + 1 \leq i \leq n.
\end{cases}
\]

Taking \( f(t) \) as \( t' \), we have

\[
f^{(s)}(0) = \begin{cases} 0, & i \neq s; \\
i!, & i = s.
\end{cases}
\]

\[
f^{(s)}(1) = \begin{cases} i!, & s \leq i; \\
(i-s)!, & s > i.
\end{cases}
\]

Thus, this lemma can be proved by substituting the above equation to the dual function.

**B. DP Basis and Its Transformation**

**Definition 2** \([9]\) In general, DP basis can be expressed as

\[
D_n^t = \begin{cases} (1-t)^n, & i = 0; \\
(t(1-t)^{i-1} - (1-t)^{i-1}), & 1 \leq i \leq \lfloor n/2 \rfloor - 1; \\
\left( \frac{1}{2} \right)^{i-1} \left( 1 - \left( \frac{1}{2} \right)^{i-1} \right), & 1 \leq i < \lfloor n/2 \rfloor; \\
\left( \frac{1}{2} \right)^{i-1} \left( 1 - \left( \frac{1}{2} \right)^{i-1} \right) + \left( \frac{1}{2} \right)^{i-1} t \left( 1 - \left( \frac{1}{2} \right)^{i-1} \right), & i = \lfloor n/2 \rfloor.
\end{cases}
\]

Thus, this lemma can be proved by substituting the above equation to the dual function.

**Theorem 2** \([9]\) DP basis \( \{ D_n^t \}_{i=0}^n \) mentioned in Definition 2 is an NTP basis.

Next, we introduce the following notations \([10]\):

\[
K_i(n, j) = 1 - \sum_{i=j}^{\lfloor n/2 \rfloor + 1} (-1)^{i-j} \binom{n}{i-j} i^{j-1};
\]

\[
K_i(n, j) = 1 - \sum_{i=j}^{\lfloor n/2 \rfloor + 1} (-1)^{i-j} \binom{n}{i-j} j^{i-1} i.
\]

**Lemma 2** \([10]\) The transpose matrix of the matrix \( H_{(n+1)\times(n+1)}^T = (\{ h_{ij} \}_{i,j=0}^{n+1}) \) can transform DP basis to Bernstein basis, i.e.,

\[
(B_0^n(t), B_1^n(t), ..., B_n^n(t)) = (D_0^n(t), D_1^n(t), ..., D_n^n(t)) H_{(n+1)\times(n+1)}^T,
\]

here the non-zero elements in the matrix \( H_{(n+1)\times(n+1)} \) can be written as

\[
h_{ij} = \begin{cases} 1, & i = j = 0; \\
\left( \frac{1}{2} \right)^{i-j} \binom{n}{i-j} \left( 1 - \left( \frac{1}{2} \right)^{i-j} \right), & 0 \leq j \leq i \leq \lfloor n/2 \rfloor; \\
\left( \frac{1}{2} \right)^{j-i} \binom{n}{j-i} \left( 1 - \left( \frac{1}{2} \right)^{j-i} \right), & 0 \leq i \leq j \leq \lfloor n/2 \rfloor; \\
\left( \frac{1}{2} \right)^{j-i} \binom{n}{j-i} \left( 1 - \left( \frac{1}{2} \right)^{j-i} \right) + \left( \frac{1}{2} \right)^{i-j} \binom{n}{i-j} \left( 1 - \left( \frac{1}{2} \right)^{i-j} \right), & i = \lfloor n/2 \rfloor; \\
\left( \frac{1}{2} \right)^{j-i} \binom{n}{j-i} \left( 1 - \left( \frac{1}{2} \right)^{j-i} \right), & i = \lfloor n/2 \rfloor; \\
\left( \frac{1}{2} \right)^{j-i} \binom{n}{j-i} \left( 1 - \left( \frac{1}{2} \right)^{j-i} \right) + \left( \frac{1}{2} \right)^{i-j} \binom{n}{i-j} \left( 1 - \left( \frac{1}{2} \right)^{i-j} \right), & j = \lfloor n/2 \rfloor + 1; \\
\left( \frac{1}{2} \right)^{j-i} \binom{n}{j-i} \left( 1 - \left( \frac{1}{2} \right)^{j-i} \right), & j = \lfloor n/2 \rfloor + 1; \\
\left( \frac{1}{2} \right)^{j-i} \binom{n}{j-i} \left( 1 - \left( \frac{1}{2} \right)^{j-i} \right) + \left( \frac{1}{2} \right)^{i-j} \binom{n}{i-j} \left( 1 - \left( \frac{1}{2} \right)^{i-j} \right), & i \geq \lfloor n/2 \rfloor + 1;
\end{cases}
\]
in which  
\[ A(n) = \left\lfloor \frac{n}{2} \right\rfloor - \left\lfloor n/2 \right\rfloor. \]

C. The Transformation between Bernstein Basis and Power Basis

**Lemma 3** [21] The transpose matrices of the matrices  
\[ Q_{(n+1)\times(n+1)} = (q_{i,j})_{(n+1)\times(n+1)} \]  
and  
\[ S_{(n+1)\times(n+1)} = (s_{i,j})_{(n+1)\times(n+1)} \]  
can transform power basis into Bernstein basis, and vice versa:

\[ (B_0^n(t), B_1^n(t), \ldots, B_n^n(t)) = (1, t, \ldots, t^n)Q^T_{(n+1)\times(n+1)} \]  
and  
\[ (1, t, \ldots, t^n) = (B_0^n(t), B_1^n(t), \ldots, B_n^n(t))S^T_{(n+1)\times(n+1)}. \]

Here  
\[ q_{i,j} = \begin{cases} 0, & j < i; \\ (-1)^{i-j} \binom{n}{j} \binom{j}{i}, & j \geq i; \end{cases} \]

\[ s_{i,j} = \begin{cases} 0, & j < i; \\ \binom{n-i}{i-j} \binom{n}{j}, & j \geq i. \end{cases} \]

D. The Inverse Matrix of the Vandermonde Matrix

**Lemma 4** The coefficients of a degree \( n \) polynomial written as  
\[ F(x) = (x-a_0)(x-a_1)\ldots(x-a_{n-1}) = z_0^n x^n + z_1^n x^{n-1} + \ldots + z_n^n \]
can be recursively computed by using the following generalized Yanghui triangle:

\[
\begin{align*}
  z_0^n & = 1, \\
  z_1^n & = \sum_{i=0}^{n-1} (-a_i), \\
  z_2^n & = \sum_{i=0}^{n-2} (-a_i) \sum_{j=0}^{n-1} (-a_j), \\
  & \quad \ldots \\
  z_n^n & = \sum_{i=0}^{n-m} (-a_i) \sum_{j=0}^{n-m+1} (-a_j) \ldots \sum_{k=0}^{n-1} (-a_k), \\
  z_{n+1}^n & = \prod_{i=0}^{n} (-a_i).
\end{align*}
\]

in which  
\[ z_0^n = 1, \quad z_1^n = \sum_{i=0}^{n-1} (-a_i), \quad z_k^n = \prod_{i=0}^{k-1} (-a_i), \]  
and the recursion formula is

\[ z_k^n = z_{k-1}^n + z_{k-1}^{n-1}(-a_k), \quad k = 3, 4, \ldots, n, \quad i = 2, \ldots, k-1. \]

**Proof:** The coefficients of the polynomial \( F(x) \) can be summarized as

\[
\begin{align*}
  z_0^n & = 1, \\
  z_1^n & = \sum_{i=0}^{n-1} (-a_i), \\
  z_2^n & = \sum_{i=0}^{n-2} (-a_i) \sum_{j=0}^{n-1} (-a_j), \\
  & \quad \ldots \\
  z_n^n & = \sum_{i=0}^{n-m} (-a_i) \sum_{j=0}^{n-m+1} (-a_j) \ldots \sum_{k=0}^{n-1} (-a_k), \\
  z_{n+1}^n & = \prod_{i=0}^{n} (-a_i).
\end{align*}
\]

Therefore Lemma 4 can be proved by a direct computation.

**Lemma 5** [22] Given \( n \) different parameters \( \alpha = \{a_i\}_{i=0}^{n-1} \), the elements in the inverse matrix \( R_{n-1}^\alpha = (r_{ij})_{n\times n} \) of the following \( n \times n \) non-degenerate Vandermonde matrix

\[
V_{n-1}^\alpha = \begin{bmatrix}
1 & a_0 & \cdots & a_0^{n-1} \\
1 & a_1 & \cdots & a_1^{n-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & a_{n-1} & \cdots & a_{n-1}^{n-1}
\end{bmatrix}_{n\times n}
\]

have the following recursive relation:

\[ r_{n-1,j}^\alpha = \frac{z_0^n}{F'(a_{n-j})} - a_{n-j}^\alpha - \frac{z_{n-m}^n}{F'(a_j)} \]

\[ j = 0, 1, \ldots, n-1, \quad m = 0, 1, \ldots, n-2. \]

**Proof:** If Denote \( V^* \), \( |V| \) as the adjoint matrix and determinant of the matrix \( V_{n-1}^\alpha \), respectively, express the matrix generated when the \( j \)-th row of the matrix \( V_{n-1}^\alpha \) is replaced by the vector \((1, x, \ldots, x^{n-1})\) as \( V_j \), and let

\[ (p_0(x), p_1(x), \ldots, p_{n-1}(x)) = (1, x, \ldots, x^{n-1})(V_{n-1}^\alpha)^{-1} = (1, x, \ldots, x^{n-1})V^*/|V|, \]

then

\[ p_j(x) = |V_j|/|V| = \prod_{i=0}^{n-1} \frac{x-a_i}{a_j-a_i} = \frac{F(x)}{F'(a_j)(x-a_j)}. \]

Since \( R_{n-1}^\alpha \) is the inverse matrix of the matrix \( V_{n-1}^\alpha \), we have

\[ p_j(x) = (1, x, \ldots, x^{n-1})(r_{0,j}^\alpha, r_{1,j}^\alpha, \ldots, r_{n-1,j}^\alpha)^T. \]

i.e.,

\[
\frac{z_0^n x^n + z_1^n x^{n-1} + \ldots + z_n^n}{F'(a_j)(x-a_j)} = r_{n-1,j}^\alpha x^{n-1} + \ldots + r_{1,j}^\alpha x + r_{0,j}^\alpha.
\]

Comparing the coefficients of \( x^i \), \( i = 0, 1, \ldots, n-1 \), which is in the both sides of the above expression, the proof of the lemma can be completed.

III. THE PIA ALGORITHM OF NTP CURVE AND SURFACE AND ITS INTERPOLATION

A. An Explicit Algorithm of PIA

Given an NTP basis \( U = \{u_i(t)\}_{i=0}^{n} \) and a sequence of the data points in \( R^2 \) or \( R^3 \), and given \( T = \{t_i\}_{i=0}^{n} \), a group of increasing parameters, \( t_0 < t_1 < \cdots < t_n \), in which \( t_i \) is assigned to the points \( p_i \), \( i = 0, 1, \ldots, n \). Firstly, we
generate an initial curve \( C^0(t) = \sum_{i=0}^{n} u_i(t) p_i^0, \) \( i = 0, 1, \ldots, n. \) Then we iteratively compute a sequence of the curve \( C^{k+1}(t) = \sum_{i=0}^{n} p_i^{k+1} u_i(t), \) where

\[
p_i^{k+1} = p_i^k + \Delta_i^k, \quad \Delta_i^k = p_i - C^k(t_i), \quad i = 0, 1, \ldots, n.
\]

Thus the control points of the iteration sequence can be written as the following matrix form:

\[
P^{k+1} - P = (I - B)P^k,
\]

in which

\[
P^j = [p_0^j, p_1^j, \ldots, p_n^j]^T, \quad j = 0, 1, \ldots, k + 1,
\]

and the collocation matrix is

\[
B = \begin{bmatrix}
    (u_0(t_0)) & (u_1(t_0)) & \cdots & (u_n(t_0)) \\
    (u_0(t_1)) & (u_1(t_1)) & \cdots & (u_n(t_1)) \\
    \vdots & \vdots & \ddots & \vdots \\
    (u_0(t_n)) & (u_1(t_n)) & \cdots & (u_n(t_n))
\end{bmatrix}.
\]

(1)

Considering that \( B \) is an invertible matrix \(^{[16]}\), we have

\[
P^{k+1} - B^{-1}P = (I - \omega B)(P^k - B^{-1}P), \quad k = 0, 1, \ldots.
\]

Since \( U \) is an NTP basis \(^{[16]}\), the above iteration sequence is convergent. Thus \( \lim_{k \to \infty} P^{k+1} = B^{-1}P \).

The case of tensor product surface can be analyzed in a similar way. Given two NTP bases \( U = \{u_i(t)\}_{i=0}^{m}, \quad V = \{v_j(s)\}_{j=0}^{n}, \) given the data points \( \{p_{i,j}\}_{i=0,j=0}^{m,n} \) in \( R^3 \) and two group of increasing parameters \( T = \{t_i\}_{i=0}^{m}, \quad S = \{s_j\}_{j=0}^{n}, \quad 0 < t_0 < t_1 < \cdots < t_m, \quad 0 < s_0 < s_1 < \cdots < s_n. \) Here every pair of the parameters \( (t_i, s_j) \) are assigned to the point \( p_{i,j}, \quad i = 0, 1, \ldots, m; \quad j = 0, 1, \ldots, n. \) Above all, the initial surface \( S^0(t) = \sum_{j=0}^{n} \sum_{i=0}^{m} u_i(t) v_j(s) p_{i,j}^0 \) is generated, where \( p_{i,j}^0 = p_{i,j}, \quad i = 0, 1, \ldots, m; \quad j = 0, 1, \ldots, n. \) Then, a sequence of the surfaces \( S^{k+1}(t) = \sum_{i=0}^{m} \sum_{j=0}^{n} u_i(t) v_j(s) p_{i,j}^{k+1} \) is generated iteratively, in which

\[
p_{i,j}^{k+1} = p_{i,j} + \Delta_{i,j}^k, \quad \Delta_{i,j}^k = p_{i,j} - S^k(t_i, s_j).
\]

The control points of the iteration sequence can be expressed as the following matrix form:

\[
P^{k+1} = P + (I - B)P^k,
\]

in which

\[
P^j = [p_{0,0}^j, p_{0,1}^j, \ldots, p_{m,0}^j, p_{0,1}^j, \ldots, p_{m,n}^j, \ldots, p_{i,j}^j, \ldots, p_{m,n}^j]^T, \quad j = 0, 1, \ldots, k + 1,
\]

\[
P = [p_{0,0}, p_{0,1}, \ldots, p_{m,0}, \ldots, p_{0,1}, \ldots, p_{m,n}, \ldots, p_{m,0}, \ldots, p_{m,n}]^T,
\]

\[
B = B_1 \otimes B_2, \quad B_1 = (u_i(t_j))_{i=0, \ldots, m}, \quad B_2 = (v_j(s_k))_{j=0, \ldots, n}
\]

Considering \( B \) is an invertible matrix \(^{[16]}\), we have

\[
P^{k+1} - B^{-1}P = (I - \omega B)(P^k - B^{-1}P), \quad k = 0, 1, \ldots.
\]

Since \( U, \) \( V \) are NTP bases \(^{[16]}\), the iteration sequence is convergent. Then \( \lim_{k \to \infty} P^{k+1} = B^{-1}P. \)

B. An Interpolation Algorithm of NTP Curve and Surface

**Theorem 3** Given an ordered data points set \( \{p_i \in R^3 | i = 0, 1, \ldots, n\} \) and \( \tau = \{t_i\}_{i=0}^{n}, \quad t_0 < t_1 < \cdots < t_n, \) a sequence of the real increasing parameters, then the NTP curve interpolating the \( i \)-th point \( p_i \) at the parameter \( t_i (i = 0, 1, \ldots, n) \) can be expressed as

\[
P(t) = \sum_{i=0}^{n} N^u_i(t) \tilde{p}_i,
\]

in which

\[
(\tilde{p}_0, \tilde{p}_1, \ldots, \tilde{p}_n)^T = Z_{(n+1) \times (n+1)} (p_0, p_1, \ldots, p_n)^T.
\]

Furthermore, if there is the following relation

\[
(1, t, \cdots, t^n) = (N^u_0(t), N^u_1(t), \cdots, N^u_n(t)) A^T
\]

between the NTP basis \( \{N^u_j(t)\}_{j=0}^{n} \) and the power basis \( \{t^j\}_{j=0}^{n}, \) then we have

\[
Z_{(n+1) \times (n+1)} = A^T (R_n^u)^T.
\]

Specially, if the NTP interpolation curve is SBGB curve or DP curve, then we have

\[
Z_{(n+1) \times (n+1)} = (F_{(n+1) \times (n+1)}\tau) (R_n^u)^T, \quad Z_{(n+1) \times (n+1)} = H^T_{(n+1) \times (n+1)} S^T_{(n+1) \times (n+1)} (R_n^u)^T.
\]

**Proof:** Suppose

\[
(N^u_0(t), N^u_1(t), \cdots, N^u_n(t)) = (1, t, \cdots, t^n) C^T,
\]

it is obvious that \( C^T = (A^T)^{-1}. \) Noting that the curve

\[
P(t) = \sum_{i=0}^{n} N^u_i(t) \tilde{p}_i
\]

interpolates the point \( p_i \) at the parameter \( t_i, \quad i = 0, 1, \ldots, n, \) we have
we can then there exist two matrices respectively.

Theorem 4 Given a space data points set \( \{ \mathbf{p}_{i,j} \mid i = 0,1,\ldots,m; \ j = 0,1,\ldots,n \} \) and a real parameter sequence \( (t_i,s_j), i = 0,1,\ldots,m; \ j = 0,1,\ldots,n \), satisfying \( t_0 < t_1 < \cdots < t_m \), \( s_0 < s_1 < \cdots < s_n \), then the tensor product NTP surface interpolating the data point \( \mathbf{p}_{i,j} \) at the parameter point \( (t_i,s_j), i = 0,1,\ldots,m; \ j = 0,1,\ldots,n \) can be expressed as

\[
\mathbf{P}(t,s) = \sum_{i=0}^{m} \sum_{j=0}^{n} M_i^m(t) N_j^n(s) \tilde{\mathbf{p}}_{i,j},
\]

in which \( M_i^m(t), i = 0,1,\ldots,m; \ N_j^n(s), j = 0,1,\ldots,n \) are two NTP bases. If we write the control point set \( \{ \tilde{\mathbf{p}}_{i,j} \}_{i,j=0}^{m,n} \) and the data point set \( \{ \mathbf{p}_{i,j} \}_{i,j=0}^{m,n} \) as the following vector forms:

\[
\tilde{\mathbf{P}} = \left[ \tilde{\mathbf{p}}_{0,0}, \tilde{\mathbf{p}}_{0,1}, \ldots, \tilde{\mathbf{p}}_{m,n} \right]^T, \quad \mathbf{P} = \left[ \mathbf{p}_{0,0}, \mathbf{p}_{0,1}, \ldots, \mathbf{p}_{m,n} \right]^T.
\]

This paper deduces the PIA limitation form of NTP curve and then there exist an \((m+1)(n+1)\times((m+1)(n+1))\) matrix \( Z \) such that

\[
\tilde{\mathbf{P}} = Z \mathbf{P},
\]

in which \( Z = Z_{(m+1)\times(m+1)}^{M} \otimes Z_{(n+1)\times(n+1)}^{N} \). If \( M_i^m(t), i = 0,1,\ldots,m; \ N_j^n(s), j = 0,1,\ldots,n \) are SBGB basis or DP basis, the matrices \( Z_{(m+1)\times(m+1)}^{M} \), \( Z_{(n+1)\times(n+1)}^{N} \) have the same form as in (2).

Corollary 2 Given a space data points set \( \{ \mathbf{p}_{i,j} \mid i = 0,1,\ldots,m; \ j = 0,1,\ldots,n \} \) and a real parameter sequence \( (t_i,s_j), i = 0,1,\ldots,m; \ j = 0,1,\ldots,n \), the PIA limitation interpolation tensor product surface is unique if we choose the NTP bases as some basis of polynomial function space with degree not more than \( m \) and \( n \) respectively.
TABLE 1. THE CONTROL POINTS OF THE INTERPOLATION CURVES

<table>
<thead>
<tr>
<th>( K = 0 ) (Said-Ball)</th>
<th>SBGB ( K = 1 )</th>
<th>SBGB ( K = 3 )</th>
<th>( K = 5 ) (Bézier)</th>
<th>DP curve</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>(1.05, -1.45)</td>
<td>(0.90, -1.25)</td>
<td>(0.70, -0.97)</td>
<td>(0.63, -0.87)</td>
<td>(-1.07e+02, -2.00e+03)</td>
</tr>
<tr>
<td>(1.80, 0.78)</td>
<td>(1.57, 0.28)</td>
<td>(1.26, -0.22)</td>
<td>(1.26, -0.22)</td>
<td>(4.11e+02, 5.63e+03)</td>
</tr>
<tr>
<td>(1.60, -6.90)</td>
<td>(1.59, -4.50)</td>
<td>(1.54, -3.16)</td>
<td>(1.54, -3.16)</td>
<td>(-5.38e+02, -5.26e+03)</td>
</tr>
<tr>
<td>(0.85, 15.90)</td>
<td>(1.15, 7.74)</td>
<td>(1.15, 7.74)</td>
<td>(1.15, 7.74)</td>
<td>(2.41e+02, 1.63e+03)</td>
</tr>
<tr>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>(-0.85, -15.90)</td>
<td>(-1.15, -7.74)</td>
<td>(-1.15, -7.74)</td>
<td>(-1.15, -7.74)</td>
<td>(-2.41e+02, -1.63e+03)</td>
</tr>
<tr>
<td>(-1.60, 6.90)</td>
<td>(-1.59, 4.50)</td>
<td>(-1.54, 3.16)</td>
<td>(-1.54, 3.16)</td>
<td>(5.38e+02, 5.26e+03)</td>
</tr>
<tr>
<td>(-1.80, -0.78)</td>
<td>(-1.57, -0.28)</td>
<td>(-1.26, 0.22)</td>
<td>(-1.26, 0.22)</td>
<td>(-4.11e+02, -5.63e+03)</td>
</tr>
<tr>
<td>(-1.05, 1.45)</td>
<td>(-0.90, 1.25)</td>
<td>(-0.70, 0.97)</td>
<td>(-0.63, 0.87)</td>
<td>(1.07e+02, 2.00e+03)</td>
</tr>
<tr>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
<td>(0, 0)</td>
</tr>
</tbody>
</table>

IV. EXAMPLES

**Example 1** In this example, we sample 11 points \( \{p_i\}_{i=0}^{10} \) from the Lemniscate of Gerono, \((x(t), y(t)) = (\cos(t), \sin(t) \cos(t)), \quad t \in [0, 2\pi] \) in the following way: \( p_i = (x(t_i), y(t_i)), \quad t_i = \pi/2 + \pi i/5, \quad i = 0, 1, \ldots, 10. \) For the sake of convenience, we choose uniform parameter \( \{t_i = i/10\}_{i=0}^{10}. \) Thus by Theorem 3, the control points of SBGB curves and DP curve interpolating the above sample points can be computed directly (See Table 1).

**Example 2** We uniformly choose 36 points from the surface \( z(x, y) = \sqrt{x^2 + y^2} \) in the interval \([0, 5; 0, 5] \). The corresponding sample points are

\[
\begin{array}{ccccccc}
(0, 0) & (1, 0) & (2, 0) & (3, 0) & (4, 0) & (5, 0) \\
(0, 1) & (1, 1.71) & (2, 1.09) & (3, 0.95) & (4, 0.97) & (5, 1.98) \\
(0, 2) & (1, 2.09) & (2, 2.14) & (3, 2.16) & (4, 2.17) & (5, 2.16) \\
(0, 3) & (1, 3.95) & (2, 3.66) & (3, 3.21) & (4, 3.24) & (5, 3.27) \\
(0, 4) & (1, 4.97) & (2, 4.79) & (3, 4.24) & (4, 2.83) & (5, 4.31) \\
(0, 5) & (1, 5.98) & (2, 5.86) & (3, 5.27) & (4, 5.32) & (5, 5.35) \\
\end{array}
\]

Then by Theorem 4, we can directly obtain the NTP interpolation surface, to be seen as in Fig.2. If we choose two Bernstein bases to generate a surface interpolating the given sample points, then the control points of interpolation surface are

\[
\begin{array}{ccccccc}
(0, 0) & (1, 0) & (2, 0) & (3, 0) & (4, 0) & (5, 0) \\
(0, 1) & (1, 2.94) & (2, 1.05) & (3, 1.77) & (4, 1.76) & (5, 1.00) \\
(0, 2) & (1, 2.04) & (2, 2.25) & (3, 2.07) & (4, 2.42) & (5, 2.02) \\
(0, 3) & (1, 1.77) & (2, 0.76) & (3, 3.00) & (4, 3.24) & (5, 3.70) \\
(0, 4) & (1, 0.76) & (2, 2.42) & (3, 4.34) & (4, 4.30) & (5, 4.18) \\
(0, 5) & (1, 1.00) & (2, 2.02) & (3, 5.27) & (4, 5.31) & (5, 5.35) \\
\end{array}
\]

If we choose one Bernstein basis and one Said-Ball basis to generate a surface interpolating the given sample points, then the control points are

\[
\begin{array}{ccccccc}
(0, 0) & (1, 0) & (2, 0) & (3, 0) & (4, 0) & (5, 0) \\
(0, 1) & (1, 1.67) & (2, 1.67) & (3, 1.67) & (4, 1.67) & (5, 1.67) \\
(0, 2) & (1, 2.50) & (2, 2.50) & (3, 2.50) & (4, 2.50) & (5, 2.50) \\
(0, 3) & (1, 3.33) & (2, 3.33) & (3, 3.33) & (4, 3.33) & (5, 3.33) \\
(0, 4) & (1, 5.00) & (2, 5.00) & (3, 5.00) & (4, 5.00) & (5, 5.00) \\
\end{array}
\]

Fig. 2. Up: the 6×6 sample data points.
V. CONCLUSIONS

This paper sufficiently investigates the matrices transformation between some usual NTP bases and power basis as well as the invertible matrix formula of the Vandermonde matrix. By expressing the PIA limitation of NTP curves or surfaces, we obtain some explicit matrix forms of the curves and surfaces generated by these NTP bases which can interpolate the given scattered data points, to make the goal of inverse engineering accomplished quickly and accurately.

Parameter choices of PIA make a big difference to the final interpolation form of the curves and surfaces, thus the influence effect and control will be our future research work.

ACKNOWLEDGMENT

This work was supported by the National Nature Science foundations of China under Grant No.60933007 and No.61070065.

REFERENCES