World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:13, No:8, 2019

Fast and Efficient Algorithms for Evaluating Uniform and Nonuniform Lagrange and Newton Curves

Taweechai Nuntawisuttiwong, Natasha Dejdumrong

Abstract—Newton-Lagrange Interpolations are widely used in numerical analysis. However, it requires a quadratic computational time for their constructions. In computer aided geometric design (CAGD), there are some polynomial curves: Wang-Ball, DP and Dejdumrong curves, which have linear time complexity algorithms. Thus, the computational time for Newton-Lagrange Interpolations can be reduced by applying the algorithms of Wang-Ball, DP and Dejdumrong curves. In order to use Wang-Ball, DP and Dejdumrong algorithms, first, it is necessary to convert Newton-Lagrange polynomials into Wang-Ball, DP or Dejdumrong polynomials. In this work, the algorithms for converting from both uniform and non-uniform Newton-Lagrange polynomials into Wang-Ball, DP and Dejdumrong polynomials are investigated. Thus, the computational time for representing Newton-Lagrange polynomials can be reduced into linear complexity. In addition, the other utilizations of using CAGD curves to modify the Newton-Lagrange curves can be taken.

Keywords—Newton interpolation, Lagrange interpolation, linear complexity.

I. Introduction

THERE are many interpolation methods used in numerical analysis. The popular one is Newton's divided-difference interpolating polynomial. The Newton's method was given by Isaac Newton written in Lemma 5 of Book III of his Principia Mathematica of 1687 [1] but it was known to him before from the letter he wrote to Oldenburg [2]. The simple forms of Newton interpolating polynomials, Lagrange interpolations, were published by Joseph-Louis Lagrange under his name. However, the same formula had been produced by Waring sixteen years earlier.

Newton-Lagrange interpolations can be beneficial to estimate intermediate data between precise data points or to predict the future data. In addition, Newton-Lagrange interpolations can be used to vectorize a raster-image into a vector image. However, a quadratic computational time, O(n^2), must be taken to construct a Newton-Lagrange interpolation.

In computer aided geometric design (CAGD), there are many polynomial curves, e.g. Bézier [3], Said-Ball [4], Wang-Ball [5], DP [6], NB1 [7], [8] and Dejdumrong [9] curves. The difference between Newton-Lagrange and CAGD curves is the utilization of CAGD curves that is for CAD/CAM modeling. In CAGD curves, Bézier curves are widely used in CAD software because they are better in shape preserving and possess simple forms. Unfortunately, Bézier curve has a quadratic computation time, which is slower than the others. However, there are some curves in CAGD, Wang-Ball, DP and Dejdumrong curves with linear complexity. In order to reduce their computational time, Bézier curve is converted into any of Wang-Ball, DP and Dejdumrong curves. Hence, the computational time for Newton-Lagrange interpolations can be reduced by converting them into Wang-Ball, DP and Dejdumrong algorithms. Thus, it is necessary to investigate the conversion from Newton-Lagrange interpolation into the curves with linear complexity, Wang-Ball, DP and Dejdumrong curves. An application of this work is to modify sketched image in CAD application with the computational time reduction for plotting Newton-Lagrange curves.

T. Nuntawisuttiwong and N. Dejdumrong are with the Department of Computer Engineering, King Mongkut's University of Technology Thonburi, Bangkok, Thailand (e-mail: taweechai.wee@gmail.com, natasha@cpe.kmutt.ac.th).

II. Newton & Lagrange Polynomials

Although Newton and Lagrange polynomials are different representations, their simple forms are the same polynomials. Newton applies Divided Difference method to represent its polynomial while Lagrange uses simpler formula. At the beginning, it is important to introduce both methods. Newton and Lagrange polynomials can be shown as follows:

A. Newton Polynomial

Let $\mathcal{N}(t)$ be the Newton polynomial of control points, $\{\mathbf{b}_i\}_{i=0}^n$, and the weight, $\{t_i\}_{i=0}^n$, then

$$\mathcal{N}(t) = \sum_{i=0}^{n} f[t_i, t_{i-1}, .., t_0] \prod_{j=0}^{i} (t - t_j), \;\; t\epsilon[t_0, t_n] \quad (1)$$

where

$$f[t_i, t_{i-1}, .., t_k] = \frac{f[t_i, t_{i-1}, .., t_{k+1}] - f[t_{i-1}, t_{i-2}, .., t_k]}{t_i - t_k}, \quad (2)$$

and

$$f[t_i] = \mathbf{b}_i. \quad (3)$$

B. Lagrange Polynomial

Let $\mathcal{L}(t)$ be the Lagrange polynomial of control points, $\{\mathbf{v}_i\}_{i=0}^n$, and the weight, $\{t_i\}_{i=0}^n$, then

$$\mathcal{L}(t) = \sum_{i=0}^{n} \mathbf{v}_i \prod_{j=0, j\neq i}^{n} \frac{t - t_j}{t_i - t_j}, \;\; t\epsilon[t_0, t_n]. \quad (4)$$

In CAGD curves, there exists the power basis form for representing curves. In this work, the power basis

representations for Newton and Lagrange polynomial are investigated. Thus, Newton and Lagrange polynomial can be computed by the following definition.

Definition 1. *Newton-Lagrange Power Basis: Let $\mathcal{N}(t)$ and $\mathcal{L}(t)$ be Newton and Lagrange polynomial with their control points, $\{\mathbf{b}_i\}_{i=0}^n$. Newton and Lagrange polynomial can be represented by using power basis as follows:*

$$\mathcal{N}(t) = \mathcal{L}(t) = \sum_{i=0}^{n}\sum_{j=0}^{n} \mathbf{b}_i \cdot f_{i,j} \cdot t^i, \tag{5}$$

where $f_{i,j}$ is Newton-Lagrange power basis defined by

$$f_{i,j} = \frac{\mathcal{F}_{i,n-j}^n(0)}{\prod_{i=0,i\neq j}^n t_j - t_i} \tag{6}$$

and

$$\mathcal{F}_{i,n-j}^n(u) = \begin{cases} 1 & , \quad \text{for } j = 0, \\ \sum_{k=u,k\neq i}^{n-j+1} \mathcal{F}_{i,j-1}^n(k+1)t_k & , \quad \text{for } j > 0. \end{cases} \tag{7}$$

Definition 2. *Newton-Lagrange Monomial Matrices: Newton and Lagrange polynomials can be represented by using monomial matrix forms as follows:*

$$\mathcal{N}(t) = \mathcal{L}(t) = \begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n \end{bmatrix} \cdot \mathbf{M} \cdot \begin{bmatrix} 1 \\ t \\ \vdots \\ t^n \end{bmatrix} \tag{8}$$

where $\mathbf{M}$ is Newton-Lagrange monomial matrix defined as follows:

$$\mathbf{M} = \begin{bmatrix} f_{0,0} & f_{0,1} & \cdots & f_{0,n} \\ f_{1,0} & f_{1,1} & \cdots & f_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n,0} & f_{n,1} & \cdots & f_{n,n} \end{bmatrix}. \tag{9}$$

The matrix inverse of Newton-Lagrange monomial matrix can be expressed as follows:

$$\begin{aligned} \mathbf{M}^{-1} &= \begin{bmatrix} f_{0,0} & f_{0,1} & \cdots & f_{0,n} \\ f_{1,0} & f_{1,1} & \cdots & f_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n,0} & f_{n,1} & \cdots & f_{n,n} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mathbf{g}_{0,0} & \mathbf{g}_{0,1} & \cdots & \mathbf{g}_{0,n} \\ \mathbf{g}_{1,0} & \mathbf{g}_{1,1} & \cdots & \mathbf{g}_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{g}_{n,0} & \mathbf{g}_{n,1} & \cdots & \mathbf{g}_{n,n} \end{bmatrix}, \end{aligned} \tag{10}$$

where

$$\mathbf{g}_{i,j} = \begin{cases} 1 & , \quad \text{for } i = 0, \\ t_j^i & , \quad \text{for } i > 0. \end{cases} \tag{11}$$

Newton and Lagrange polynomial are very useful for polynomial interpolation. However, they take a lot of computational time, $O(n^2)$. In CAGD curves, Wang-Ball, DP and Dejdumrong curves consume linear computational time, $O(n)$. Thus, the computational time can also be reduced by using the conversion from Newton-Lagrange polynomials into Wang-Ball, DP or Dejdumrong curves.

III. Conversion from Uniform Newton-Lagrange Polynomials into CAGD Curve

A. Conversion from Uniform Newton-Lagrange Curve into Wang-Ball Curve

Wang-Ball control points, $\{\mathbf{p}_i\}_{i=0}^n$, can be obtained from Newton control points, $\{\mathbf{b}_i\}_{i=0}^n$, and Lagrange control points, $\{\mathbf{v}_i\}_{i=0}^n$, as follows:

$$\begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n \end{bmatrix} = \begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n \end{bmatrix} \cdot \mathbf{A}^{-1} \tag{12}$$

and

$$\begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{bmatrix} \cdot \mathbf{A}^{-1} \tag{13}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{0,0} & \mathbf{a}_{0,1} & \cdots & \mathbf{a}_{0,n} \\ \mathbf{a}_{1,0} & \mathbf{a}_{1,1} & \cdots & \mathbf{a}_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{n,0} & \mathbf{a}_{n,1} & \cdots & \mathbf{a}_{n,n} \end{bmatrix} \tag{14}$$

and

$$\mathbf{a}_{i,j}^n(t) = \begin{cases} (2t_j)^i(1-t_j)^{i+2} & , \quad \text{for } 0 \leq i \leq \left\lfloor \frac{n}{2} \right\rfloor - 1 \\ (2t_j)^i(1-t_j)^{n-i} & , \quad \text{for } i = \left\lfloor \frac{n}{2} \right\rfloor \\ (2(1-t_j))^{n-i}(t_j)^i & , \quad \text{for } i = \left\lceil \frac{n}{2} \right\rceil \\ \mathbf{a}_{n-i,j}^n(1-t_j) & , \quad \text{for } \left\lceil \frac{n}{2} \right\rceil + 1 \leq i \leq n \end{cases} \tag{15}$$

B. Conversion from Uniform Newton-Lagrange Curve into DP Curve

DP control points, $\{\mathbf{q}_i\}_{i=0}^n$, can be obtained from Newton control points, $\{\mathbf{b}_i\}_{i=0}^n$, and Lagrange control points, $\{\mathbf{v}_i\}_{i=0}^n$, as follows:

$$\begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_n \end{bmatrix} = \begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n \end{bmatrix} \cdot \mathbf{C}^{-1} \tag{16}$$

and

$$\begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{bmatrix} \cdot \mathbf{C}^{-1} \tag{17}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_{0,0} & \mathbf{c}_{0,1} & \cdots & \mathbf{c}_{0,n} \\ \mathbf{c}_{1,0} & \mathbf{c}_{1,1} & \cdots & \mathbf{c}_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{n,0} & \mathbf{c}_{n,1} & \cdots & \mathbf{c}_{n,n} \end{bmatrix} \tag{18}$$

and

$$\mathbf{c}_{i,j}^{n}(t)=\begin{cases}(-1)^{n-i}(1-t_j)^n & ,\text{for} \quad i=0\\ (-1)^{n-i}(1-t_j)^{n-i}t_j & ,\text{for} \quad 0<i\leq \lfloor\frac{n}{2}\rfloor-1\\ (-1)^{n-i}(n-2i)(1-t_j)^{i+1}t_j+ & \\ (\frac{1}{2})^{\lceil\frac{n}{2}\rceil-\lfloor\frac{n}{2}\rfloor}(1-t_j^{\lfloor\frac{n}{2}\rfloor+1}+ & ,\text{for} \quad i=\lfloor\frac{n}{2}\rfloor\\ (-1)^{\lfloor\frac{n}{2}\rfloor}(1-t_j)^{\lfloor\frac{n}{2}\rfloor+1}) & \\ (n-2i)(\frac{t_j-b}{b-a})(\frac{t_j-a}{b-a})^{n-i+1}+ & \\ (\frac{1}{2})^{\lceil\frac{n}{2}\rceil-\lfloor\frac{n}{2}\rfloor}(1-(\frac{t_j-a}{b-a})^{\lfloor\frac{n}{2}\rfloor+1}+ & ,\text{for} \quad i=\lceil\frac{n}{2}\rceil\\ (-1)^{\lfloor\frac{n}{2}\rfloor}(1-t_j)^{\lfloor\frac{n}{2}\rfloor+1}) & \\ (-1)(1-t_j)t_j^i & ,\text{for} \quad \lceil\frac{n}{2}\rceil+1\leq i<n\\ (-1)^{n-i}t_j^i & ,\text{for} \quad i=n\end{cases} \tag{19}$$

C. Conversion from Uniform Newton-Lagrange Curve into Dejdumrong Curve

Dejdumrong control points, $\{\mathbf{d}_i\}_{i=0}^n$, can be obtained from Newton control points, $\{\mathbf{b}_i\}_{i=0}^n$, and Lagrange control points, $\{\mathbf{v}_i\}_{i=0}^n$, as follows:

$$\begin{bmatrix}\mathbf{d}_0 & \mathbf{d}_1 & \cdots & \mathbf{d}_n\end{bmatrix}=\begin{bmatrix}\mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n\end{bmatrix}\cdot\mathbf{D}^{-1} \tag{20}$$

and

$$\begin{bmatrix}\mathbf{d}_0 & \mathbf{d}_1 & \cdots & \mathbf{d}_n\end{bmatrix}=\begin{bmatrix}\mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix}\cdot\mathbf{D}^{-1} \tag{21}$$

where

$$\mathbf{D}=\begin{bmatrix}\mathbf{d}_{0,0} & \mathbf{d}_{0,1} & \cdots & \mathbf{d}_{0,n}\\ \mathbf{d}_{1,0} & \mathbf{d}_{1,1} & \cdots & \mathbf{d}_{1,n}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{d}_{n,0} & \mathbf{d}_{n,1} & \cdots & \mathbf{d}_{n,n}\end{bmatrix} \tag{22}$$

and

$$\mathbf{d}_{i,j}^{n}(t)=\begin{cases}(-1)^{i+1}3^i(1-t_j)^{i+3} & ,\text{for} \quad i=0\\ (-1)^{i+1}3^i(1-t_j)^{i+3}t_j^i & ,\text{for} \quad 0<i<\lceil\frac{n}{2}\rceil-1\\ (-1)^{n-i}3^i(1-t_j)^{n-i}t_j^i & ,\text{for} \quad i=\lceil\frac{n}{2}\rceil-1\\ (-1)^{n-i}2\times 3^{i-1}(1-t_j)^it_j^i & ,\text{for} \quad i=\frac{n}{2}\text{ and } n\text{ is even}\\ (-3)^{n-i}(1-t_j)^{n-i}t_j^i & ,\text{for} \quad i=\lfloor\frac{n}{2}\rfloor+1\\ (-3)^{n-i}(1-t_j)^{n-i}t_j^{n-i+3} & ,\text{for} \quad \lfloor\frac{n}{2}\rfloor+1<i<n\\ (-3)^{n-i}t_j^{n-i+3} & ,\text{for} \quad i=n\end{cases} \tag{23}$$

IV. Conversion from Nonuniform Newton-Lagrange into CAGD curves

A. Conversion from Nonuniform Newton-Lagrange curve into Nonuniform Wang-Ball curve

Wang-Ball control points, $\{\mathbf{p}_i\}_{i=0}^n$, can be transformed from Newton control points, $\{\mathbf{b}_i\}_{i=0}^n$, and Lagrange control points, $\{\mathbf{v}_i\}_{i=0}^n$, as follows:

$$\begin{bmatrix}\mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n\end{bmatrix}=\begin{bmatrix}\mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n\end{bmatrix}\cdot A^{-1} \tag{24}$$

and

$$\begin{bmatrix}\mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n\end{bmatrix}=\begin{bmatrix}\mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix}\cdot A^{-1} \tag{25}$$

where

$$A=\begin{bmatrix}a_{0,0} & a_{0,1} & \cdots & a_{0,n}\\ a_{1,0} & a_{1,1} & \cdots & a_{1,n}\\ \vdots & \vdots & \ddots & \vdots\\ a_{n,0} & a_{n,1} & \cdots & a_{n,n}\end{bmatrix} \tag{26}$$

and

$$a_{i,j}^{n}(t,a,b)=\begin{cases}(-2)^i\beta^{i+2} & \text{for} \quad i=0\\ (-2)^i\beta^{i+2}\alpha^i & \text{for} \quad 0<i\leq\lfloor\frac{n}{2}\rfloor\\ (-1)^{n-i}2^i\beta^{n-i}\alpha^i & \text{for} \quad i=\lfloor\frac{n}{2}\rfloor\\ (-2)^{n-i}\beta^{n-i}\alpha^i & \text{for} \quad i=\lceil\frac{n}{2}\rceil\\ (-2)^{n-i}\beta^{n-i}\alpha^{n-i+2} & \text{for} \quad \lceil\frac{n}{2}\rceil+1\leq i<n\\ (-2)^{n-i}\alpha^{n-i+2} & \text{for} \quad i=n\end{cases} \tag{27}$$

where

$$\alpha=\frac{t_j-a}{b-a} \tag{28}$$

and

$$\beta=\frac{t_j-b}{b-a} \tag{29}$$

B. Conversion from Nonuniform Newton-Lagrange curve into Nonuniform DP curve

DP control points, $\{\mathbf{q}_i\}_{i=0}^n$, can be derived from Newton control points, $\{\mathbf{b}_i\}_{i=0}^n$, and Lagrange control points, $\{\mathbf{v}_i\}_{i=0}^n$, as follows:

$$\begin{bmatrix}\mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_n\end{bmatrix}=\begin{bmatrix}\mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n\end{bmatrix}\cdot C^{-1} \tag{30}$$

and

$$\begin{bmatrix}\mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_n\end{bmatrix}=\begin{bmatrix}\mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix}\cdot C^{-1} \tag{31}$$

where

$$C=\begin{bmatrix}c_{0,0} & c_{0,1} & \cdots & c_{0,n}\\ c_{1,0} & c_{1,1} & \cdots & c_{1,n}\\ \vdots & \vdots & \ddots & \vdots\\ c_{n,0} & c_{n,1} & \cdots & c_{n,n}\end{bmatrix} \tag{32}$$

and

$$c_{i,j}^{n}(t,a,b)=\begin{cases}(-1)^{n-i}\beta^{n} & ,\text{for} \quad i=0\\ (-1)^{n-i}\beta^{n-i}\alpha & ,\text{for} \quad 0<i\leq \lfloor\frac{n}{2}\rfloor-1\\ (-1)^{n-i}(n-2i)\beta^{i+1}\alpha+ & \\ (\frac{1}{2})^{\lceil\frac{n}{2}\rceil-\lfloor\frac{n}{2}\rfloor}(1-\alpha^{\lfloor\frac{n}{2}\rfloor+1} & ,\text{for} \quad i=\lfloor\frac{n}{2}\rfloor\\ +(-1)^{\lfloor\frac{n}{2}\rfloor}\beta^{\lfloor\frac{n}{2}\rfloor+1}) & \\ (n-2i)\beta\alpha^{n-i+1}+ & \\ (\frac{1}{2})^{\lceil\frac{n}{2}\rceil-\lfloor\frac{n}{2}\rfloor}(1-\alpha^{\lfloor\frac{n}{2}\rfloor+1}+ & ,\text{for} \quad i=\lceil\frac{n}{2}\rceil\\ (-1)^{\lfloor\frac{n}{2}\rfloor}\beta^{\lfloor\frac{n}{2}\rfloor+1}) & \\ (-1)\beta\alpha^{i} & ,\text{for} \quad \lceil\frac{n}{2}\rceil+1\leq i<n\\ (-1)^{n-i}\alpha^{i} & ,\text{for} \quad i=n\end{cases} \tag{33}$$

C. Conversion from Nonuniform Newton-Lagrange Curve into Nonuniform Dejdumrong Curve

Dejdumrong control points, $\{\mathbf{d}_i\}_{i=0}^{n}$, can be expressed by Newton control points, $\{\mathbf{b}_i\}_{i=0}^{n}$, and Lagrange control points, $\{\mathbf{v}_i\}_{i=0}^{n}$, as follows:

$$\begin{bmatrix}\mathbf{d}_0 & \mathbf{d}_1 & \cdots & \mathbf{d}_n\end{bmatrix}=\begin{bmatrix}\mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n\end{bmatrix}\cdot D^{-1} \tag{34}$$

and

$$\begin{bmatrix}\mathbf{d}_0 & \mathbf{d}_1 & \cdots & \mathbf{d}_n\end{bmatrix}=\begin{bmatrix}\mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix}\cdot D^{-1} \tag{35}$$

where

$$D=\begin{bmatrix}d_{0,0} & d_{0,1} & \cdots & d_{0,n}\\ d_{1,0} & d_{1,1} & \cdots & d_{1,n}\\ \vdots & \vdots & \ddots & \vdots\\ d_{n,0} & d_{n,1} & \cdots & d_{n,n}\end{bmatrix} \tag{36}$$

and

$$d_{i,j}^{n}(t,a,b)=\begin{cases}(-1)^{i+1}3^{i}\beta^{i+3} & ,\text{for} \quad i=0\\ (-1)^{i+1}3^{i}\beta^{i+3}\alpha^{i} & ,\text{for} \quad 0<i<\lceil\frac{n}{2}\rceil-1\\ (-1)^{n-i}3^{i}\beta^{n-i}\alpha^{i} & ,\text{for} \quad i=\lceil\frac{n}{2}\rceil-1\\ (-1)^{n-i}2\cdot 3^{i-1}\beta^{i}\alpha^{i} & ,\text{for} \quad i=\frac{n}{2}\text{ and } n\text{ is even}\\ (-3)^{n-i}\beta^{n-i}\alpha^{i} & ,\text{for} \quad i=\lfloor\frac{n}{2}\rfloor+1\\ (-3)^{n-i}\beta^{n-i}\alpha^{n-i+3} & ,\text{for} \quad \lfloor\frac{n}{2}\rfloor+1<i<n\\ (-3)^{n-i}\alpha^{n-i+3} & ,\text{for} \quad i=n\end{cases} \tag{37}$$

V. Fast & Efficient Algorithms

An efficient algorithm used to plot uniform and nonuniform Newton-Lagrange with linear complexity can be defined by Algorithm as follows:

Algorithm 1: Algorithm of Uniform Newton-Lagrange curves

1 Convert from the control points of uniform Newton-Lagrange into Wang-Ball, DP or Dejdumrong control points;
2 Use Wang-Ball, DP or Dejdumrong algorithm to represent uniform Newton-Lagrange polynomials;

Algorithm 2: Algorithm of Nonuniform Newton-Lagrange curves

1 Convert from the control points of nonuniform Newton-Lagrange into Wang-Ball, DP or Dejdumrong control points;
2 Use Wang-Ball, DP or Dejdumrong algorithm to represent nonuniform Newton-Lagrange polynomials;

VI. Conclusion

In this work, the conversion formulae from both uniform and non-uniform Newton-Lagrange curve into Wang-Ball, DP and Dejdumrong curves are well-proposed in the formal ways with an example of proof in appendix. Hence, the computational time for the calculation of Newton-Lagrange curve can be reduced from quadratic into linear computational complexity. Even though the conversion algorithm is cubic computation, but it will be computed only once. An example of using our new proposed techniques is when we want to convert a sketched image that can be recognized by using Newton-Lagrange either uniform and non-uniform polynomials.

Appendix A
Proof for the Conversion from Uniform Newton-Lagrange into Wang-Ball Polynomials

In order to understand obtaining of the formula for the conversion from uniform Newton-Lagrange into Wang-Ball polynomials, the proof is defined as follows:

Given $\mathcal{N}(t)$ be Newton polynomial and $\mathcal{L}(t)$ be Lagrange polynomial. They can be represented using monomial matrix form as follows:

$$\mathcal{N}(t)=\mathcal{L}(t)=\begin{bmatrix}\mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n\end{bmatrix}\cdot\mathbf{M}\cdot\begin{bmatrix}1\\ t\\ \vdots\\ t^{n}\end{bmatrix}. \tag{38}$$

Moreover, Wang-Ball curve, $A(t)$, with its control points,

$\{\mathbf{p}_i\}_{i=0}^n$, can be represented using monomial matrix form as follows:

$$A(t) = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n \end{bmatrix} \cdot \mathcal{A} \cdot \begin{bmatrix} 1 \\ t \\ \vdots \\ t^n \end{bmatrix}, \tag{39}$$

where $\mathcal{A}$ is Wang-Ball monomial matrix.

Proof

The proof is started by equating $\mathcal{N}(t)$, $\mathcal{L}(t)$ and $A(t)$, then

$$\begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n \end{bmatrix} \cdot \mathbf{M} \cdot \begin{bmatrix} 1 \\ t \\ \vdots \\ t^n \end{bmatrix} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n \end{bmatrix} \cdot \mathcal{A} \cdot \begin{bmatrix} 1 \\ t \\ \vdots \\ t^n \end{bmatrix}. \tag{40}$$

Simplifying equation 40, we get

$$\begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n \end{bmatrix} \cdot \mathbf{M} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n \end{bmatrix} \cdot \mathcal{A}. \tag{41}$$

Finally we obtain,

$$\begin{bmatrix} \mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n \end{bmatrix} \cdot \mathcal{A} \cdot \mathbf{M}^{-1}. \tag{42}$$

Thus, it needs to investigate the formula for $\mathcal{A} \cdot \mathbf{M}^{-1}$. From Wang-Ball curve,

$$A(t) = \sum_{i=0}^{n} \cdot \mathbf{p}_i \cdot A_i^n(t) = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_n \end{bmatrix} \cdot \mathcal{A} \cdot \begin{bmatrix} 1 \\ t \\ \vdots \\ t^n \end{bmatrix}. \tag{43}$$

Consider that

$$A_i^n(t) = \mathcal{A} \cdot \begin{bmatrix} 1 \\ t \\ \vdots \\ t^n \end{bmatrix}, \tag{44}$$

and $A_i^n(t)$ is already defined as follows:

$$A_i^n(t) = \begin{cases} (2t)^i(1-t)^{i+2} & , \quad \text{for } 0 \le i \le \left\lfloor \frac{n}{2} \right\rfloor - 1, \\ (2t)^i(1-t)^{n-i} & , \quad \text{if } i = \left\lfloor \frac{n}{2} \right\rfloor, \\ (2(1-t))^{n-i}t^i & , \quad \text{if } i = \left\lceil \frac{n}{2} \right\rceil, \\ A_{n-i}^n(1-t) & , \quad \text{for } \left\lceil \frac{n}{2} \right\rceil + 1 \le i \le n. \end{cases} \tag{45}$$

Since

$$\mathcal{A} \cdot \mathbf{M}^{-1} = \mathcal{A} \cdot \begin{bmatrix} 1 & 1 & \dots & 1 \\ t_0 & t_1 & \dots & t_n \\ t_0^2 & t_1^2 & \dots & t_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ t_0^n & t_1^n & \dots & t_n^n \end{bmatrix}, \tag{46}$$

it is obvious that

$$\mathbf{a}_{i,j}^n(t) = \mathcal{A} \cdot \mathbf{M}^{-1} = \begin{cases} (2t_j)^i(1-t_j)^{i+2} & , \quad \text{for } 0 \le i \le \left\lfloor \frac{n}{2} \right\rfloor - 1 \\ (2t_j)^i(1-t_j)^{n-i} & , \quad \text{for } i = \left\lfloor \frac{n}{2} \right\rfloor \\ (2(1-t_j))^{n-i}(t_j)^i & , \quad \text{for } i = \left\lceil \frac{n}{2} \right\rceil \\ \mathbf{a}_{n-i,j}^n(1-t_j) & , \quad \text{for } \left\lceil \frac{n}{2} \right\rceil + 1 \le i \le n \end{cases} \tag{47}$$

REFERENCES

[1] I. Newton, *Philosophiae Naturalis Principia Mathematica.* London, 1687.
[2] I. Newton, *Letter to Oldenburg (24 october 1676).* in The Correspondence of Isaac Newton, vol. 2, pp.110-161, 1960.
[3] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, 5th ed. Academic Press, Morgan Kaufman Publishers, San Francisco, 2002.
[4] H. B. Said, *Generalized Ball Curve and Its Recursive Algorithm.* ACM Transactions on Graphics, vol. 8, pp. 360-371, 1989.
[5] G. J. Wang, *Ball Curve of High Degree and Its Geometric Properties.* Appl. Math.: A Journal of Chinese Universities, vol. 2, pp. 126-140, 1987.
[6] J. Delgado and J. M. Peña, *A Shape Preserving Representation with an Evaluation Algorithm of Linear Complexity.* Computer Aided Geometric Design, vol. 20(1), pp. 1-20, 2008.
[7] W. Hongyi, *Unifying Representation of Bézier Curve And Genaralized Ball Curves.* Appl. Math. J. Chinese Univ. Ser. B, vol. 5(1), pp. 109-121, 2000.
[8] Y. Dan and C. Xinmeng, *Another Type Of Generalized Ball Curves And Surfaces.* Acta Mathematica Scientia, vol. 27B(4), pp. 897-907, 2007.
[9] N. Dejdumrong, *Efficient Algorithms for Non-rational and Rational Bézier Curves.* Fifth International Conference on Computer Graphics, Imaging and Visualisation, 2008.