

# Methods of Graphic Representation of Curves in CAD systems in Knitting Industry

Elena Iv. Zaharieva-Stoyanova<sup>1</sup>

**Abstract** – This paper treats the problems related to curve generation and its application in knitting industry CAD/CAM systems. The methods can be applied for a curve generation in knitting object pattern design. Most commonly used methods are reviewed and some functions for curve generation created. They can be used as programming modules in a CAD system for knitting pattern design.

**Keywords** – Computer graphics, CAD/CAM systems, parametric cubic curves, Bezier curves, B-splines, FF knitting method.

## I. INTRODUCTION

The development of CAD/CAM systems needs a solution of some problems related to graphics representation of designed objects in digital format. Generally, application of computer graphics in knitting industry CAD/CAM systems has two aspects:

- knitting structures design;
- design of knitting pattern shapes.

The second one is related to knitting machine's capacity to make products by fully-fashion (FF) method. It means that the machine knits the cuts of products, or the whole products. This method allows to avoid cutting the materials, reduces the number of the operations and it leads to reducing the waste products to a minimum.

Usually, as designed objects the patterns of knitting products are represented with straight-line polygons. [1],[2] This manner of pattern representation is used because of the following reasons: it is possible for the knitting products size to have a little tolerance within of  $\pm 1$  cm; it is easier to process straight-lines than curves; knitting machines are not so precise in material production. This precision depends on a knitting machine performance called fine or gauche. The performance is related to the number of needles per inch or cm. It determines a loop size and a thread thickness, too. All these features distinguish a knitting patterns' presentation from cloths patterns' one.

Because of these reasons the patterns of knitting products are described as straight-lines polygons without curves. If there is a curved section, the workers form it additionally by cutting material.

This paper treats the problems related to curve representation as object in computer graphics. These methods can be applied for a curve's generation in knitting object pattern design. The most commonly used methods are reviewed. Similarly, the methods of curve generation used in cloths patterns representation are given in [3] but as was already mentioned knitting patterns shapes are more flexible. What is aimed here is to apply more methods in the design of knitting pattern objects. Functions drawing curves are created by these methods. Said functions will be used for graphic representation of knitting products' patterns. Their development is strongly related to the requirement of knitting product realization i.e. to obtain a form of knitting pattern which will match the design one. However, it is the main problem of FF knitting method, hence our effort to show some possibilities of usage in CAD/CAM systems in knitting industry automation.

## II. CUBIC CURVES REPRESENTATION

Cubic curves are commonly used in computer graphics because they are quite flexible and not so complex as well [4],[5],[6]. These features determined the way the paper reviews the application of said curves.

### A. Parametric Cubic Curves

The parametric form of curve representation is more usable:

$$\begin{aligned} x(t) &= a_3 t^3 + a_2 t^2 + a_1 t + a_0 \\ y(t) &= b_3 t^3 + b_2 t^2 + b_1 t + b_0 \end{aligned} \quad (1)$$

Using a vector form, the first equation can be represented as followed:

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \quad (2)$$

Using a vector form equation (1) is transformed into:

$$\begin{aligned} x(t) &= T \cdot A \\ y(t) &= T \cdot B \end{aligned} \quad (3)$$

Where T, A, and B are vectors.

<sup>1</sup>Elena Ivanova Zaharieva-Stoyanova is with the Technical University of Gabrovo, 4 H. Dimitar str, 5300 Gabrovo, Bulgaria, E-mail: zaharieva@tugab.bg

The derivatives of the curve with respect to  $t$  can be expressed as follows:

$$\begin{aligned} x'(t) &= [3t^2 + 2t + 1 + 0]A \\ y'(t) &= [3t^2 + 2t + 1 + 0]B \end{aligned} \quad (4)$$

Cubic curve definition needs 3 point – two endpoints and one middle point, as well as the tangent vector at the middle point. Let the points P0, P2 fix the endpoints; P1 – the midpoint; and T1 – the tangent at the P1.

The parametric curve can be determined as followed formula:

$$\begin{aligned} x(t) &= T.M.X \\ y(t) &= T.M.Y \end{aligned} \quad (5)$$

Using the first equation, the full formula is obtained as:

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -4 & 0 & 4 & 4 \\ 8 & -4 & 6 & -4 \\ -5 & 4 & -2 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_1 \\ x_1' \\ x_0 \end{bmatrix} \quad (6)$$

The product of multiplying T by M is so called basis (or blending) functions:

$$\begin{aligned} f_1(t) &= -4t^3 + 8t^2 - 5t + 1 \\ f_2(t) &= -4t^2 + 4t \\ f_3(t) &= -4t^3 + 6t^2 - 2t \\ f_4(t) &= 4t^3 + 4t^2 - 1 \end{aligned} \quad (7)$$

The basis functions are so called “weighting” function for the points P0, P1, P2, and tangent T1 because they determine how these points exert influence on the curve. The curve will be represented as follows:

$$\begin{aligned} x(t) &= f_1(t)x_0 + f_2(t)x_1 + f_3(t)x_1' + f_4(t)x_2 \\ y(t) &= f_1(t)y_0 + f_2(t)y_1 + f_3(t)y_1' + f_4(t)y_2 \end{aligned} \quad (8)$$

### B. Bezier Curves

A Bezier curve in its most common form is a simple cubic equation that can be used in any number of useful ways. Originally, it was developed by Pierre Bezier for CAD/CAM operations.

Bezier curves are also a variation of the parametric cubic curves specified by four control points P0, P1, P2, P3. The expression for the matrix corresponding to formula (6) is:

$$Mb = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (9)$$

The Bezier basis functions are as follows:

$$\begin{aligned} f_1(t) &= -t^3 - 3t^2 - 3t + 1 \\ f_2(t) &= 3t^3 - 6t^2 + 3t \\ f_3(t) &= -3t^3 + 3t^2 \\ f_4(t) &= t^3 \end{aligned} \quad (10)$$

Alternatively, the basis functions are:

$$\begin{aligned} f_1(t) &= (1-t)^3 \\ f_2(t) &= 3t(1-t)^2 \\ f_3(t) &= 3t^2(1-t) \\ f_4(t) &= t^3 \end{aligned} \quad (11)$$

Bezier curve given in parametric format:

$$P(t) = \sum_{i=0}^m C_m^i t^i (1-t)^{m-i} P_i \quad (12)$$

For better graphics interpretation let us transform this formula as:

$$P(t) = (1-t)^m P_0 + \sum_{i=1}^{m-1} C_m^i t^i (1-t)^{m-i} P_i + t^m P_m \quad (13)$$

### C. B-splines

A B-spline is a spline-function (part-polynomial function) which equals zero in all sub-segments except in the interval  $[m, m+1]$  ([5],[6]).

To represent a B-spline in sub-segment  $i$ , the following formula is used:

$$N_{i,0}(x) = \begin{cases} 1, & x_i \leq x \leq x_{i+1} \\ 0, & x \notin (x_i, x_{i+1}) \end{cases} \quad (14)$$

An  $m$ -degree B-spline in  $[x_i, x_{i+m+1}]$  can be represented as:

$$N_{i,m}(x) = \frac{x - x_i}{x_{i+m} - x_i} N_{i,m-1}(x) + \frac{x_{i+m+1} - x}{x_{i+m+1} - x_{i+1}} N_{i+1,m-1}(x) \quad (15)$$

Normally, square and cubic B-splines are used. Should B-spline can be of higher order, almost cubic B-splines are applied. The cubic B-spline basis functions work over four control points. The basis functions appears below (equation 19, 20). Note that special basis function are needed for the first and last two sections of the B-spline so that it passes through the first and last points.

The basis functions for first section are:

$$\begin{aligned} B_1(t) &= (1-t)^3 \\ B_2(t) &= \frac{21}{12}t^3 - \frac{9}{2}t^2 + 3t \\ B_3(t) &= -\frac{11}{12}t^3 + \frac{3}{2}t^2 \\ B_4(t) &= \frac{t^3}{6} \end{aligned} \quad (16)$$

The basis functions for the second section are:

$$\begin{aligned} B_1(t) &= \frac{(1-t)^3}{4} \\ B_2(t) &= \frac{7}{12}t^3 - \frac{5}{4}t^2 + \frac{t}{4} + \frac{7}{12} \\ B_3(t) &= -\frac{1}{2}t^3 + \frac{1}{2}t^2 + \frac{1}{2}t + \frac{1}{6} \\ B_4(t) &= \frac{t^3}{6} \end{aligned} \quad (17)$$

The basis functions for the middle sections are:

$$\begin{aligned} B_1(t) &= \frac{(1-t)^3}{4} \\ B_2(t) &= \frac{1}{2}t^3 - t^2 + \frac{2}{3} \\ B_3(t) &= -\frac{1}{2}t^3 + \frac{1}{2}t^2 + \frac{1}{2}t + \frac{1}{6} \\ B_4(t) &= \frac{t^3}{6} \end{aligned} \quad (18)$$

### III. FUNCTION FOR CURVE GENERATION

Using the formule given in previous section some functions for curve generation are created. Some of them have already been presented in [7], [8]. This paper gives their development and a creation of some new functions.

The following function creates a Bezier curve by using of 3 control points:

```
void CBezierCurvesView::BezierCurves3(CDC *pDC, CPoint *P)
{
    double t, dt, x, y;
    double dx=0, dy=0;
    pDC->MoveTo(P[0]);
    dt=0.01;
    t=0;
    int i = 0;
    while (t<=1)
    {
        x=pow((1-t),3)*P[0].x+3*pow((1-t),2)*t*P[1].x+3*t*t*(1-t)*P[2].x+t*t*t*P[3].x;
        y=pow((1-t),3)*P[0].y+3*pow((1-t),2)*t*P[1].y+3*t*t*(1-t)*P[2].y+t*t*t*P[3].y;
        pDC->LineTo(CPoint((long)x,(long)y));
        t+=dt;
    }
}
```

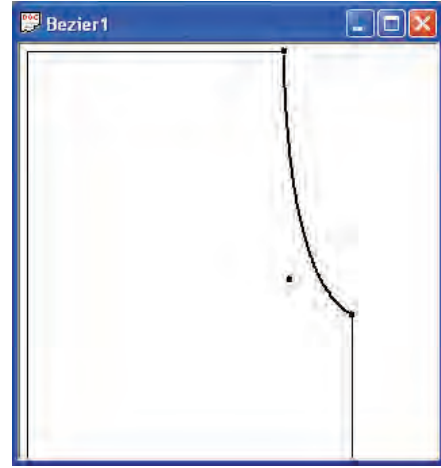


Fig. 1. Curve generation to form sleeve in knitting pattern using 2 control points

A function generating a Bezier curve by using of 2 control points is created, too. These functions are applied in knitting pattern design. Fig. 1 and 2 are examples of their application.

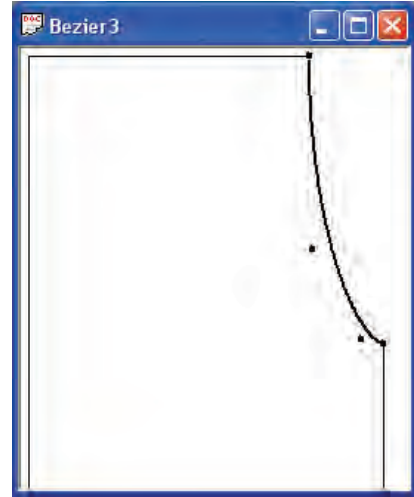


Fig. 2. Curve generation to form sleeve in knitting pattern using 3 control points

To generate a Bezier curve by n control points the following function is created:

```
void CBezierCurvesView::BezierCurvesDraw(CDC *pDC)
{
    double t, dt, x, y;
    double C, xk, yk;
    int i;
    pDC->MoveTo(P[0]);
    dt=0.01;
    t=0;
    while (t<=1)
    {
        xk=0;
        yk=0;
        for (int i=1; i<n-1;i++)
        {
            C = (double)facturiel(n)/(facturiel(i)*facturiel(n-i));
            xk += C*pow(t,i)*pow(1-t,n-i)*P[i].x;
            yk += C*pow(t,i)*pow(1-t,n-i)*P[i].y;
        }
    }
}
```

```

x=pow((1-t),n)*P[0].x + xk + pow(t,n)*P[n].x;
y=pow((1-t),n)*P[0].y + yk + pow(t,n)*P[n].y;
pDC->LineTo(CPoint((long)x,(long)y));
t+=dt;
}
}

```

This function can be applied for curve generation of knitting pattern but it has a disadvantage – the curve doesn't go through the control points. To avoid it the function of curve generation splits the control points into some groups and applies the previous function simultaneously. The result is given on fig. 3.

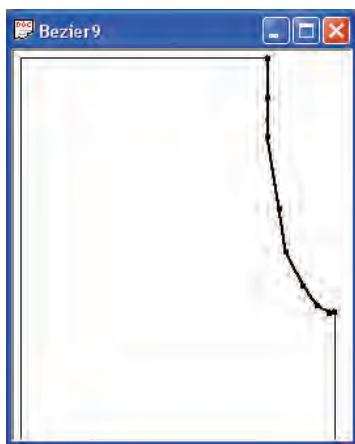


Fig. 3. Curve generation to form sleeve in knitting pattern using Bezier method

The function for curve generation using B-spline method is created, too. It uses the equation 15, 16, 17. The result of its application is represented in fig. 4.

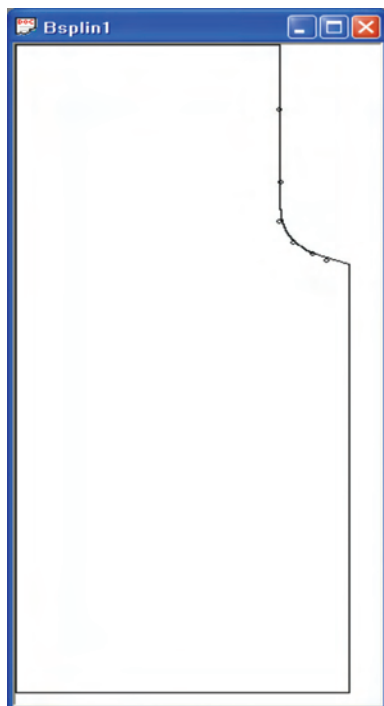


Fig. 4. Knitted pattern generated by using a cube B-spline curve

#### IV. CONCLUSION

This paper treats the problems related to curve generation and its application in knitting industry CAD/CAM systems. The methods can be applied for a curve's generation in knitting object pattern design. The most commonly used methods are reviewed, especially parametric cubic curve generation, generation of Bezier curves and B-spline method. The paper traces out some methods for curve generation. Some functions for curve generation are created. Their application is as a part of CAD system for knitting pattern design. The development of curve generation functions are strongly related to the algorithm for an accuracy of knitting pattern shape form suggested in [1]. The project is realized as an application in MS Visual C++ 6.00 using MFC.

#### REFERENCES

- [1] Zaharieva E., J. Angelova, Methodics of Fully Fashioned Knitting on Cotton, Textil & Obleklo, 5/1998, pp. 15-17.
- [2] Angelova, J., E. Zaharieva-Stoyanova, Shape Precision at Set Outline Knitting, Scientific conference EMF'2005, 22-24. September 2005, Varna, pp. 143-148.
- [3] Rakova R., Hr. Petrov, Methods for Curve Generation of Cloths Patterns by Polynomial Interpolation, Textil & Obleklo, 11-12/2000.
- [4] Broucke P., Bézier curves, December 1996.
- [5] Lukipudis E., Computer Graphics and Geometric Modeling, Sofia, Tehnika, 1996.
- [6] Pavlidis T., Algorithms for Graphics and Image Processing, Bell laboratories, Computer Science Press, 1982.
- [7] Zaharieva-Stoyanova E., Application of Bezier Curves in Knitting Industry CAD/CAM Systems, International conference ICEST'2005, 29.06-1.07.2005, Nish, Serbia and Montenegro, pp.687-690.
- [8] Zaharieva-Stoyanova E., Application of Spline Functions in Knitting Products Graphic Design, International scientific conference UNITECH'05, 24-25 November 2005, Gabrovo, pp. I-273 – I-276.
- [9] Zaharieva-Stoyanova E., Algorithm for Computer Aided Design Curve Shape Form Generation of Knitting Patterns, International scientific conference on Automation, Quality and Testing, Robots, AQTR 2006 (THETA15), May 25-28 2006, Cluj-Napoka, Romania, pp TI-327-331.