Algorithm for Computer Aided Design Curve Shape Form Generation of Knitting Patterns

Elena Zaharieva-Stoyanova
Technical University of Gabrovo
4 H. Dimirat str.
5300 Gabrovo, Bulgaria, phone +359 223 529

Abstract - **The knitting industry CAD/CAM system's development needs a solution of some problems related to the computer representation of the designed objects. One of them is a generation of curve shape form knitting pattern and its transformation to knitting rows and stitches. This is an important problem for the FF knitting method because the knitted product's shape must be the same as the pattern's one. This paper suggests an algorithm for curve forms knitting patterns generation and its transformation to knitting rows and stitches. The algorithm's idea is to use the stitch size and geometry, applying the Bezier and B-spline methods. The proposed algorithm has been realized as an application, implemented with MS Visual C++ and can be used in knitting industry CAD/CAM systems.**

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

I. Introduction

The CAD/CAM systems' development needs a solution of some problems related to the computer graphics and computer representation of the designed objects. The application of computer graphics in knitting industry CAD/CAM system has two aspects: design of knitting structures, and design of knitting patterns' forms.

The latter is related to the knitting machine's ability to make products by fully-fashion (FF) method. It means that the machine knits the cuts of products, or the whole products as a whole. This method allows avoiding cutting the materials reduces the number of the operations and gets the waste products to a minimum.

The main problem in the FF knitting method is that the shape of the product must be the same as that of the pattern. It means that the metric shape has to be converted into knitting rows and stitches, in order to be realized. Therefore, the knitting structures and the shape of the knitting pattern are closely related to each other.

To convert a metric form to a number of rows and stitches the designer needs to know the stitch size and geometry. They depend on the kind of the knitting structure and also on the specifications of the material (the thread) it will be produced from and the machine gauge.

The conversion of a pattern into stitches and rows is easier, when the pattern is represented by a piece-wise straight line shape, rather than a curve. That is why the designers often use only straight-line shape forms to create knitting patterns. The method for conversion of these patterns to stitches and rows is given in [1], [2].

The new knitting machines allow designers to use curve-shaped forms. These forms need to be transformed into stitches, so that the knitting pattern preserves the form. This is not an easy task and often without immediate solution – the straight lines and curves of the knitting structure are not smooth. Thus, the transformation task is to represent a knitting pattern, as close as possible to the form.

[3] suggests a solution and implementation in MS Visio and MS Visual Basic. The same paper also suggests using Adobe Photoshop for knitting pattern generation. This solves the problem partially without taking the knitting pattern stitch shape into consideration and the transformation of the pattern to stitches is not done automatically.

This paper suggests an algorithm for curve forms knitting pattern generation and its transformation to knitting rows and stitches. The algorithm is based on curve generation, according to the shapes and sizes of the stitches. The purpose is to produce a knitting pattern with the same form as the designed shape, which is important for designers [4], [5]. The algorithm is implemented as functions, which generate knitting patterns with curve shapes.

II. Methods for Curve Generation

There are different mathematical methods for curve generation, which can be used for producing patterns for knitting threads. The proposed algorithm uses two of them, namely the Bezier curve generation and B-Splines.

A. Berier 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 [7], [8], [9], [10].

Bezier curves are defined using **m** control points, known as **knots**. Two of these are the end points of the curve, while the others effectively define the gradient at the end points. These points control the shape of the curve. The curve is actually a **blend** of the knots. This is a recurring theme of approximation curves; defining a curve as a blend of the values of several control points [5], [6], [7], [8].

A Bezier curve is represented in parametric form as:

$$x = p_x(t) \quad (1)$$

$$y = p_y(t)$$

Let (x_0,y_0), (x_1,y_1), ...(x_m,y_m) be control points. The Bezier parametric curve function is of the form:

$$p_x(t) = \sum_{i=0}^{m} C_m^i t^i (1-t)^{m-i} x_i \quad (2)$$

$$p_y(t) = \sum_{i=0}^{m} C_m^i t^i (1-t)^{m-i} y_i$$

The value of the parameter **t** value varies from **0** to **1**. The value of the coefficient $\boldsymbol{C_m}$ is:

$$C_m^i = \frac{m!}{i!(m-i)!} \quad (3)$$

Using the vector form the Bezier curve (2) is represented as:

$$P(t) = \sum_{i=0}^{m} C_m^i t^i (1-t)^{m-i} P_i \quad (4)$$

For better graphics interpretation let us transform this formula as:

$$P(t) = (1-t)^m P_o + \sum_{i=1}^{m-1} C_m^i t^i (1-t)^{m-i} P_i + t^m P_m \quad (5)$$

For computer presentation the equation (5) is better than (4) because it avoids the division by when **i=0**, or **i=m** and avoids having to compute **n!** for **n=0**, which is undefined.

The proposed algorithm uses equation (5). The example of its application is given on fig. 1 which represents a knitting pattern drawing. A sleeve shape is produced via a Bezier curve with three control points.

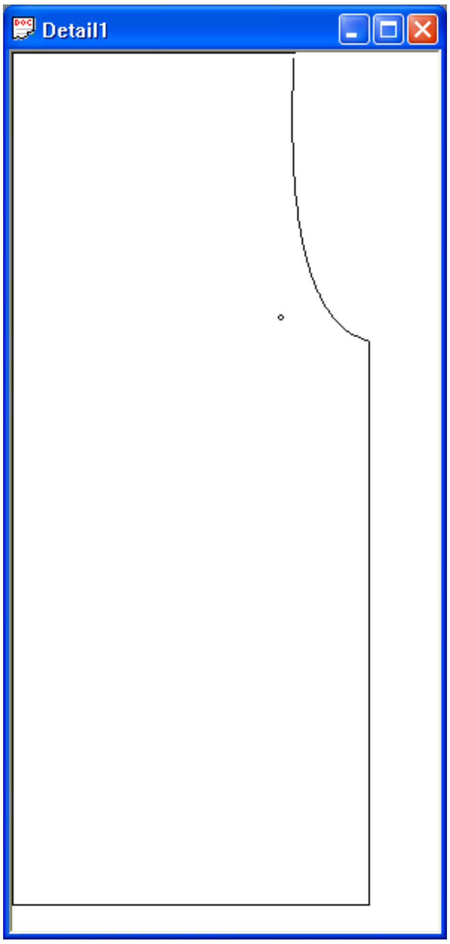

Figure 1. Knitted pattern generated by Bezier curve with three control points

B. B-spline curves

A B-spline is a spline-function (part-polynomial function) which equals zero in all sub-segments except in the interval [m, m+1] ([7],[8], [11]).

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

$$N_{i,0}(x) = \begin{cases} 1, x_i \le x \le x_{i+1} \\ 0, x \notin (x_i, x_{i+1}) \end{cases} \quad (6)$$

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

$$N_{i,m}(x) = \frac{x - x_i}{x_{i+m} - x_i} N_{i,m-1}(x) + \frac{x_{i+m+1} - x}{x_{i+m+1} - x_{i+1}} N_{i+1,m-1}(x) \quad (7)$$

The B-spline representation formulas are given in [7], [8], [11]. The programming modules given in this paper use uniform B-splines. The formula describing a linear uniform B-spline is:

$$U_{i,1}(u(x)) = \begin{cases} u, 0 \le u \le 1 \\ 2-u, 1 \le u \le 2 \end{cases} \quad (8)$$

To represent a square uniform B-spline, we can use the following formula:

$$U_{i,2}(u(x)) = \begin{cases} \frac{1}{2}u^2, 0 \le u \le 1 \\ \frac{3}{4} - \left(u - \frac{3}{2}\right)^2, 1 \le u \le 2 \\ \frac{1}{2}(3-u)^2, 2 \le u \le 3 \end{cases} \quad (9)$$

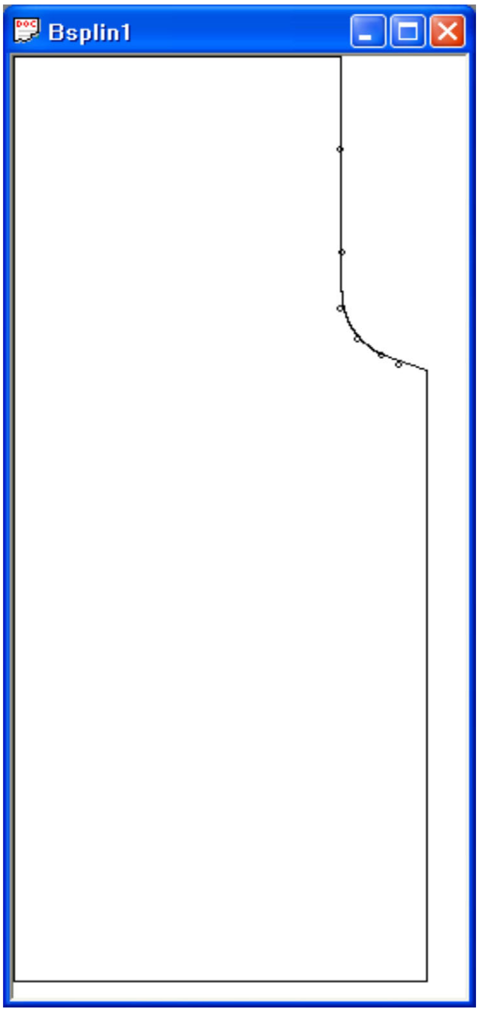

Figure 2. Knitted pattern generated by using a uniform cube B-slpine curve

To represent a cubic uniform B-spline the following formula is used:

$$U_{i,3}(u(x)) = \begin{cases} \frac{1}{6}u^3, 0 \le u \le 1 \\ \frac{2}{3} - \frac{1}{2}(u-2)^3 - (u-2)^2, 1 \le u \le 2 \\ \frac{2}{3} - \frac{1}{2}(u-2)^3 - (u-2)^2, 2 \le u \le 3 \\ \frac{1}{6}(4-u)^2, 3 \le u \le 4 \end{cases} \quad (10)$$

The uniform B-splines may be represented in a matrix form. The matrix representations of formulas (9), (10) are:

$$S_i(x) = \begin{bmatrix} x^2 & x & 1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \end{bmatrix} \quad (11)$$

$$S_i(x) = \begin{bmatrix} x^3 & x^2 & x & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \\ p_{i+2} \end{bmatrix} \quad (12)$$

Equation (11) represents a uniform square B-spline; the equation (12) represents a uniform cubic B-spline. The points p_{i-1}, p_i, p_{i+1}, p_{i+2} are control points.

The proposed algorithm uses equations (11) and (12). Fig. 2 is an example of a knitting pattern, represented by applying these equations. It shows a sleeve curve, produced by using the uniform cubic B-spline.

III. ALGORITHM FOR CURVE GENERATION OF KNITTED PATTERNS

The knitting pattern shown on fig. 1 and fig. 2 can be realized by changing the number of knitted needles during the knitting process. To produce the shape, where a sleeve will be attached, for example, the knitting machine must reduce the number of needles. The problems related to the realization of the knitting pattern by FF method, namely the kinds of pattern shape forms and how to knit them are presented in [1], [2].

The discussion topic of this paper is the use of the stitch dimension and geometry in the proposed algorithm. The stitch outline is shown on fig. 3. **A** value represents the stitch width; the **B** is the stitch height (see fig. 3a.). To reduce the number of the needles the machine makes a curve knitting two, three, or more stitches together (see fig. 3b, 3c, 3d). The vice-versa situation is possible too. The number of needles in the knitted row (see fig. 3e) may be increased. It is important to knit a pattern following the set pattern form. The precision is better when the knitted pattern is close to the set pattern form.

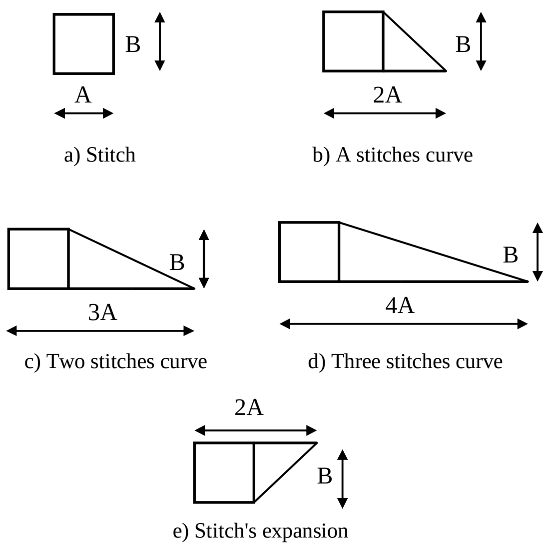

Figure 3. Stitch dimensions and geometry

The suggested algorithm generates a curve shape knitting pattern by tracing the set shape form and using stitch dimensions and geometry. Let the point set (Cx_i, Cy_i) describe the position of the end of the knitting stitch. The first point corresponding to the curve's start is (Cx_0, Cy_0) point.

After it generates a point, approximating the curve, the algorithm checks the **x** variation (**dx**). It is a difference between the next and previous point (x1,y1) corresponding to the knitting stitch:

$$dx=x-x1 \quad dy=y-y1 \tag{13}$$

If the absolute value of **dx** is greater than stitch width **A**. It means that the machine must change the number of needles – the number will go up, when **dx** is greater than 0, and will go down if **dx** is less than 0.

The value of **dy** and the ratio **dx/dy** determine the new point position. If the **dx/dy ratio** is equal to 1, the next point position is found, using this equation:

$$Cx_i=Cx_{i-1} \pm A \quad Cy_i=Cy_{i-1} - B \tag{14}$$

If the **dx/dy** ratio is less than 1, it means that two new points will be inserted. To determine the first point, formula (14) is used. The next position is determined by the following equation:

$$Cx_i=Cx_{i-1} \quad Cy_i=Cy_{i-1} - (int(dy/B) -1)*B \tag{15}$$

If **dx/dy** ratio is greater than 1, it means that the knitting machine must reduce the number of knitting stitches with 2, 3, or more needles. There is a technological restriction: usually the machine tears the thread when knitting more than 3 stitches together. Because of this, we should not allow the ratio **dx/dy** to be greater than 3.

The algorithm works as follows:

```
Cx0 = x0;
Cy0 = y0;
while ((Cx[i], Cy[i]) is between [(x0,y0), (xn,yn)] )
{
   dx = dx + x;
   dx = dy + y;
   if ( fabs(dx) >= A )
      {
         if(fabs(dx/dy) < 1)
            {
                  i=i+1;
                  if (dx<0)  Cx[i] = Cx[i-1] - A;
                  else  Cx[i] = Cx[i-1] + A;
                  Cy[i] = Cy[i-1] - B;

                  i=i+1;
                  Cx[i] = Cx[i-1];
                  Cy[i] = Cy[i-1] - (int(dy/B)-1)*B;
            }
         else if (fabs(dx/dy)  = 1 )
         {
               i=i+1;
               Cx[i] = Cx[i-1] - A;
               Cy[i] = Cy[i-1] - B;
         }
            else
             if((fabs(dx/dy)>1.5)&&(fabs(dx/dy)<=2.5)
               {
                i=i+1;
                Cx[i] = Cx[i-1] - 2*A;
                Cy[i] = Cy[i-1] - B;
               }
         else
             if((fabs(dx/dy)>2.5)&&(fabs(dx/dy)<=3.5)
               {
                i=i+1;
                Cx[i] = Cx[i-1] - 3*A;
                Cy[i] = Cy[i-1] - B;
               }
               else exit();

      dx=0;
      dy=0;
   }
}
```

According to algorithm, if the **dx/dy** ratio is greater than 3.5 the program stops. This algorithm is applied to produce functions generating a knitting pattern with a curve shape and transforming it into knitting stitches and rows. Fig. 4 and fig. 5 are examples of its application.

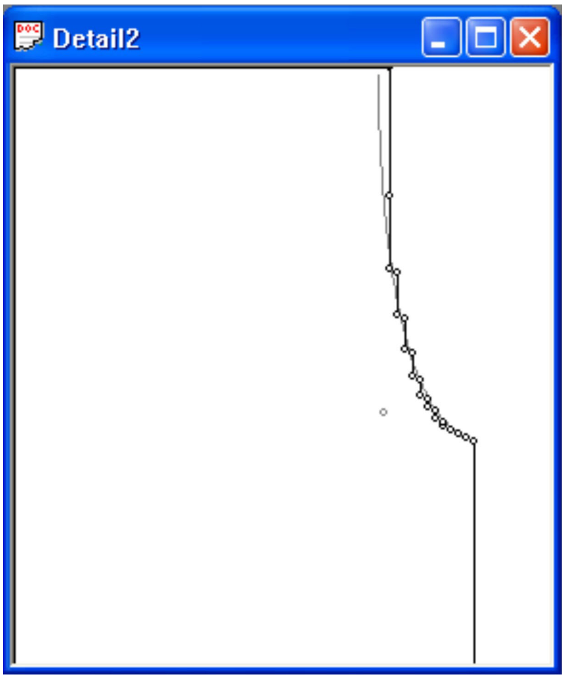

Figure 4. The application of the algorithm using a knitted pattern generated by Bezier curve with three control points

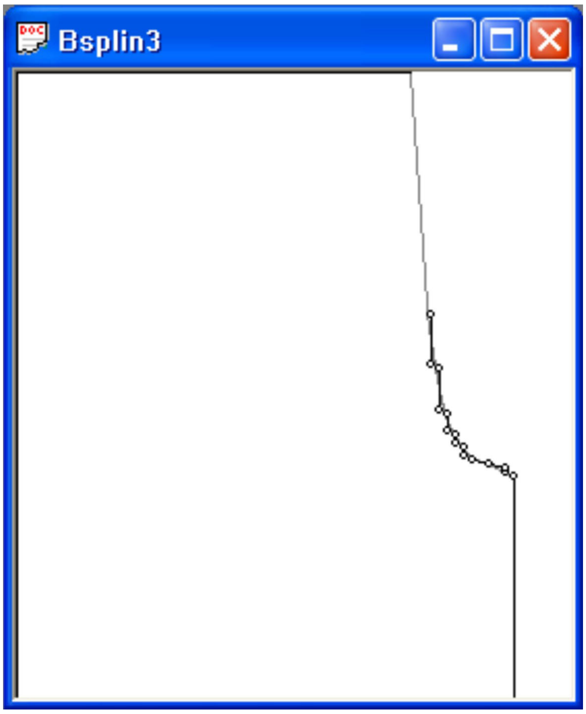

Figure 5. The application of the algorithm using a knitted pattern generated by a uniform cube B-slpine curve

The Bezier curve generation method is used for the first example. The points, marking the end points of the stitches are shown in black. They trace a shape of really knitted pattern. It can be seen that the black line is close to the set pattern line, shown in gray. The sizes of **A** and **B** stitch dimensions is determined according to the number of rows and stitches per 5 centimeters.

The second example is an algorithm's application using a uniform cube B-spline for curve generation. The shape of really knitted pattern colored in black is near to the gray colored set pattern line.

IV. Conclusion

This paper treats the problems related to curve generation and usage in knitting industry CAD/CAM systems. Because of the abilities of the new knitting machines, it is possible to make knitting products by Fully-Fashion (FF) method. It is possible to knit patterns with different kind of shapes. The FF method requires certain precision - the knitted pattern must be the same as the set form. The paper traces out some methods for curve generation. It proposes an algorithm for generation of the curve of really knitted pattern. It can be used with different curve generation methods. The algorithm is realized as an application in MS Visual C++ 6.00 using MFC.

References

[1] Zaharieva E., St. Milchev, Automatc Generation and Adjustment of Programs for Cotton Knitting Control, *International conference AUTOMATICS and INFORNATICS'97, 13-17 October, 1997, Sofia*, pp. 75-78.

[2] Zaharieva E., J. Angelova, Methodics of Fully Fashioned Knitting on Cotton, *Textil & Obleklo, 5/1998*, pp. 15-17.

[3] Olga Diordiev, Costea Budulan, *Straightforward methods for computer aied shape generation of knitted patterns, 5th international conference -Texsci 2003, Liberec*, pp. 372 - 375.

[4] Angelova, J., E. Zaharieva-Stoyanova, Shape Precision at Set Outline Knitting, *Scientific conference EMF'2005, 22-24. September 2005, Varna*, pp. 143-148.

[5] Zaharieva-Stoyanova E., J. Angelova, Computer Aided Design of Knitting Products with Curve Form of View, *International scientific conference UNITECH'05, 24-25 November 2005, Gabrovo*, pp. I-277 – I-281.

[6] Rakova R., Hr. Petrov, Methods for Curve Generation of Cloths Patterns by Polynomial Interpolation, *Textil & Obleklo, 11-12/2000*.

[7] Lukipudis E., Computer Graphics and Geometric Modeling, *Sofia, Tehnika, 1996*.

[8] Pavlidis T., Algorithms for Graphics and Image Processing, *Bell laboratories, Computer Science Press, 1982*.

[9] Brourke P., Bézier curves, *December 1996*.

[10] 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.

[11] 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.