

# Introducing Fundamentals of Computer Graphics Using MATLAB™

Shih-Liang (Sid) Wang

Department of Mechanical Engineering  
North Carolina A&T State University  
Greensboro, NC 27411

## Introduction

In a technical elective CAD course (MEEN 619) in Department of Mechanical Engineering, North Carolina A&T State University, fundamentals of computer graphics typically account for about one-third of the course material. CAD software-IDEAS and finite element analysis are the other subjects in this course. Introduction of computer graphics addresses, among other topics, geometric transformation and parametric curves and surfaces, including B-spline and Bezier curves. These subjects applied to the design of airfoils, auto bodies and ship hulls, as well as to commercial advertising and movie making. Without good understanding of these graphics fundamentals, CAD users can not effectively use associated tools.

Most textbooks [1-4] covering these graphics subjects are primarily written for computer science majors. Algorithms to implement these concepts are efficient but difficult to be programmed in the conventional programming languages that engineering students are familiar with. Many engineering students feel the complex mathematical expressions and programs hinder the learning of these concepts.

MATLAB [5] is a powerful environment for linear algebra with graphical presentation [6], and is available on a wide range of computer platforms. Unlike a general-purpose language, MATLAB development goes much faster and code is dramatically shorter. In some regards, it is a higher language than most common programming languages like C or FORTRAN. MATLAB is therefore a great computation environment for learning the fundamentals of computer graphics. Many MATLAB files have been developed in the past few years by the author and his students to help effectively presenting key concepts and visualizing these mathematical expressions.

## Parametric Representation of Analytical Curves

Parametric representation of curves and surfaces is a foundation of computer graphics. It is easier to introduce this format with an analytical curve, an arc of a circle for example. Cartesian form of the equation of an arc of a circle can be expressed as the equation of a circle with joint angle constraints:

$$\begin{aligned}x &= r \cos \theta \\y &= r \sin \theta \\ \theta_0 &\leq \theta \leq \theta_1\end{aligned}\tag{1}$$

The angle range can also be represented in a parametric form as:

$$\begin{aligned}\theta &= \theta_0 + u(\theta_1 - \theta_0) \\ 0 &\leq u \leq 1\end{aligned}$$

Equation (1) can then be rewritten in the parametric form:

$$\begin{aligned}x &= r \cos(\theta_0 + u(\theta_1 - \theta_0)) \\ y &= r \sin(\theta_0 + u(\theta_1 - \theta_0)) \\ 0 &\leq u \leq 1\end{aligned}\tag{2}$$

Equation (2) shows that parametric form is effective and elegant to represent a segment of an analytic curve.

### Parametric Representation of Synthetic Curves

Different forms of spline curves, which are parametric representation of synthetic curves, are important parts of all computer drawing or computer-aided design software packages which include: Microsoft Office Draw, Microsoft Paint, CorelDraw, AutoCAD, I-DEAS, and ProEngineer. Additionally, Postscript fonts are stored as spline curves. Although these curves are widely available, they are not intuitive to many users. On-line help of these software packages do not effectively guide users on the characteristics of these curves.

### Bezier Curves

A Bezier curve can be expressed as:

$$\begin{aligned}\mathbf{p}(u) &= \sum_{i=0}^n \mathbf{p}_i B_{i,n}(u) \quad 0 \leq u \leq 1 \\ B_{i,n}(u) &= \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}\end{aligned}$$

where  $\mathbf{p}_i$  is the position vector of a control point. The order of the curve,  $n$ , is the highest order of parameter  $u$ . The number of control points and their positions define a curve. Figures 1-3 represent three different Greek alphabetical characters. Figures 4-5 show two different mathematical symbols.

Through these examples, characteristics of a Bezier curve like endpoint interpolation, tangent conditions, and convex hull property can be demonstrated. Note that the lines connecting the control points are shown in these figures as the control polygons. Figures 6 and 7 show the same Greek alphabetical character (delta) represented by seven and nine control points respectively. Note that the with additional control points, the curve becomes smoother.

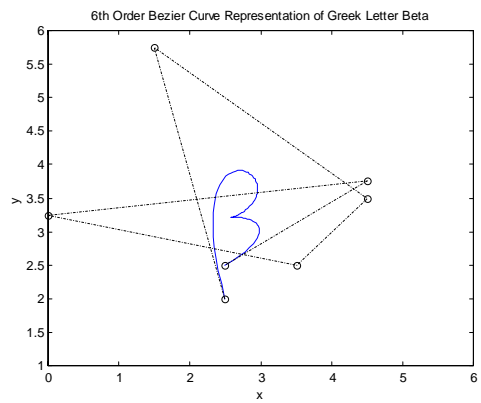
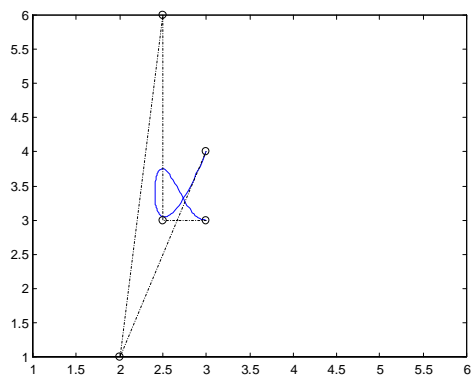


Figure 1 Alpha – with Five Control Points      Figure 2 Beta – with Seven Control Points

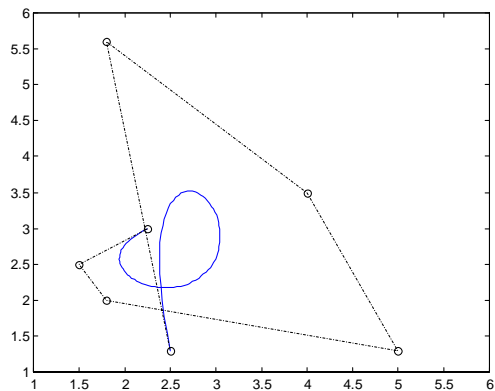


Figure 3 Phi – with Seven Control Points

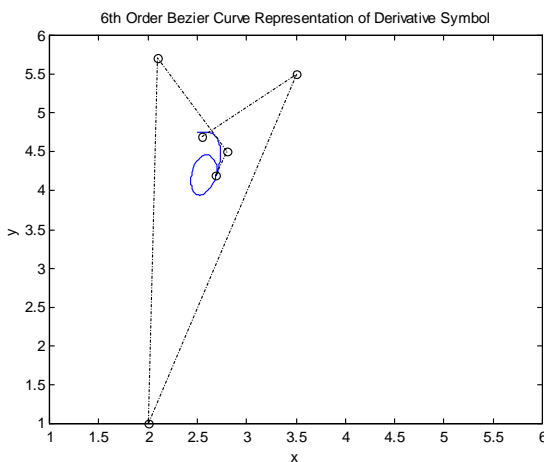


Figure 4 Partial Differential Sign – with Seven Control Points

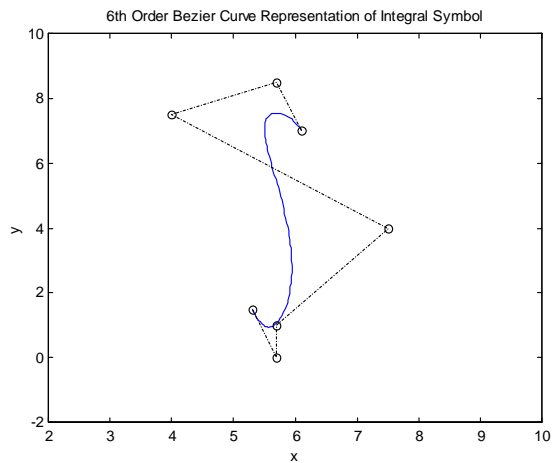


Figure 5 Integral Sign – with Seven Control Points

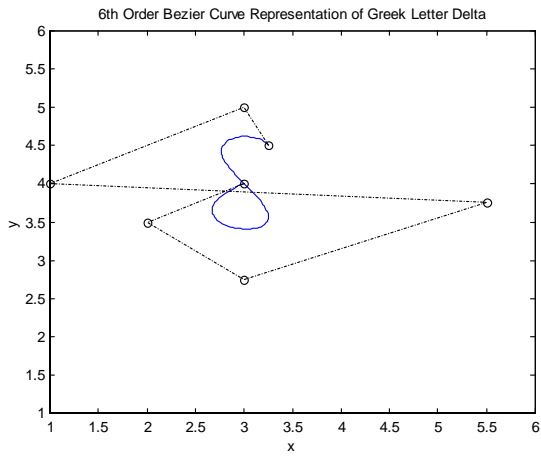


Figure 6 Delta – with Seven Control Points

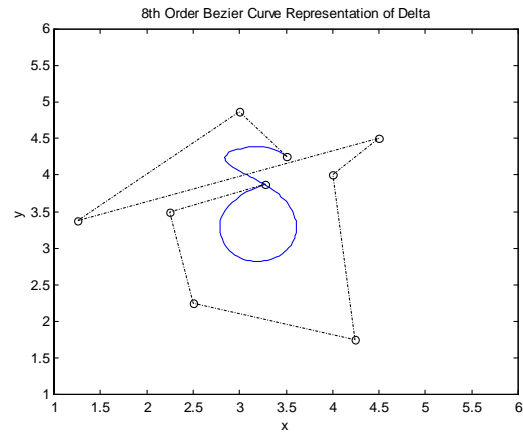


Figure 7 Delta – with Nine Control Points

### Bezier Surfaces

A Bezier surface is an extension of a one-dimensional Bezier curve to two parameters  $u$  and  $v$ , and can be expressed as:

$$\mathbf{p}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{p}_{ij} B_{i,n}(u) B_{j,m}(v) \quad 0 \leq u \leq 1, 0 \leq v \leq 1$$

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

$$B_{j,m}(v) = \frac{m!}{j!(m-j)!} v^j (1-v)^{m-j}$$

Figure 8 shows a sunvisor modeled with a Bezier surface. MATLAB can display a surface with a mesh grid, as in this figure, or by a smooth surface with different colors, similar to what commercial CAD packages can provide.

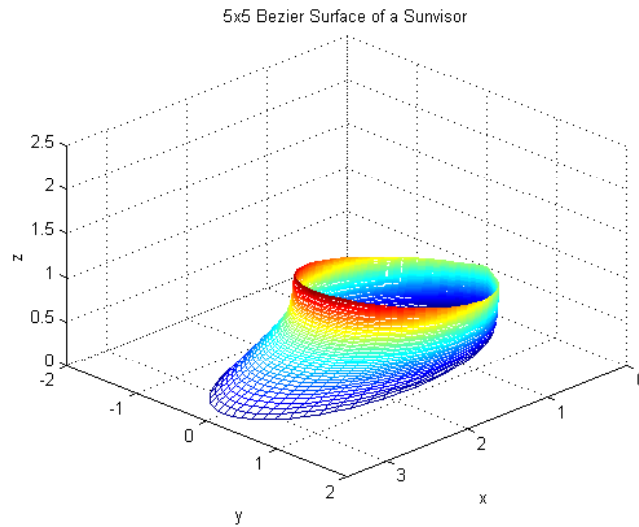


Figure 8 A Sunvisor with 36 control points

## B-Spline Curves

B-spline curves have properties similar to those of Bezier curves. The parametric form of the equation can be expressed as:

$$\mathbf{p}(u) = \sum_{i=0}^n N_{i,k}(u) \mathbf{p}_i \quad 0 \leq u \leq u_{n+k}$$

The order of curve is  $k-1$ .

The number of control points is  $n+1$ .

The control points are  $\mathbf{p}_0, \dots, \mathbf{p}_n$ .

The upper bound of the parameter  $u$  is  $u_{n+k}$ .

Knots vector is from  $u_0$  to  $u_{n+k}$  and is defined as :

$$u_j = \begin{cases} 0 & j < k \\ j-k+1 & k \leq j \leq n \\ n-k+2 & j > n \end{cases}$$

Re cursive formulation of the coefficient can be expressed as :

$$N_{i,k}(u) = (u-u_i) \frac{N_{i,k-1}(u)}{(u_{i+k-1}-u_i)} + (u_{i+k}-u) \frac{N_{i+1,k-1}(u)}{(u_{i+k}-u_{i+1})}$$

One important feature of B-spline curves is the order of a curve is independent of the number of control points. Because of this characteristic, local control of a curve, as composed to global control, is possible.

Figures 9 and 10 respectively show a fourth and a second order B-spline curve representing a Greek alphabetical character (beta) with identical control points. Note that the fourth order curve is smoother, neater, and farther away from the control polygon than the second order curve.

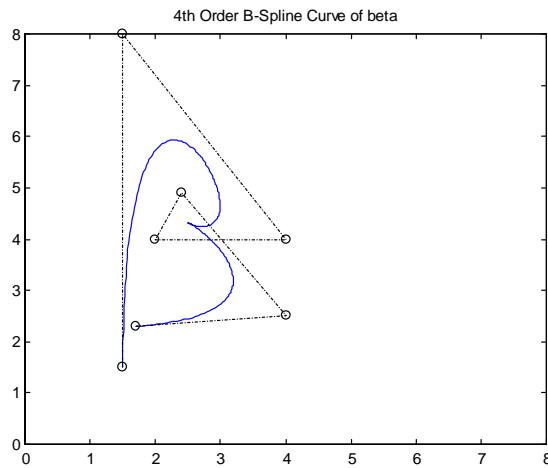


Figure 9 Beta – 4<sup>th</sup> order B-Spline

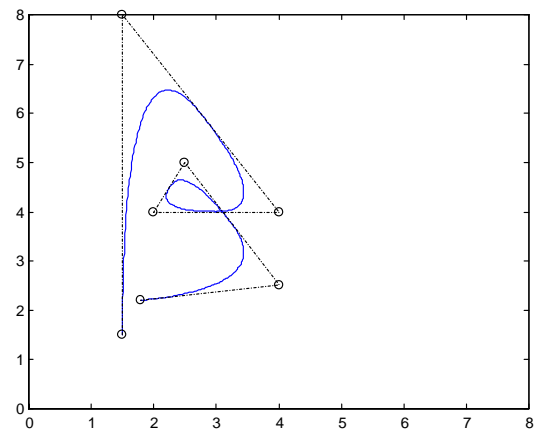


Figure 10 Beta – 2<sup>nd</sup> order B-Spline

## Homogeneous Transformation

$$\mathbf{P}^* = [\mathbf{T}] \mathbf{P}$$

(3)

$\mathbf{P}$  can be a single position vector, or a collection of vectors defining curves, surfaces, and solids. The transformation matrix  $[\mathbf{T}]$  can be represented as:

$$[\mathbf{T}] = \left[ \begin{array}{ccc|c} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ \hline t_{41} & t_{42} & t_{43} & t_{44} \end{array} \right] = \left[ \begin{array}{c|c} \mathbf{T}_1 & \mathbf{T}_2 \\ \mathbf{T}_3 & 1 \end{array} \right]$$

The 3x3 submatrix  $[\mathbf{T}_1]$  is used for scaling, reflection, or rotation; the 3x1 column vector  $[\mathbf{T}_2]$  generates translation; the 1x3 row vector  $[\mathbf{T}_3]$  produces perspective projection. Figure 11 shows a wedge reflected about  $xz$ -plane with the following transformation matrix:

$$[\mathbf{T}] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

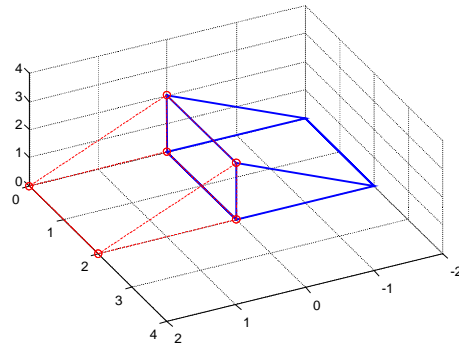


Figure 11 Reflection of a wedge about  $xz$ -plane.

Two consecutive transformations with Equation (3) can be combined as:

$$\mathbf{P}^* = [\mathbf{T}_1] [\mathbf{T}_2] \mathbf{P} \tag{4}$$

If the order of transformation  $[\mathbf{T}_1]$  and  $[\mathbf{T}_2]$  is exchanged, the result is different. In Figure 12, the wedge is rotated  $30^\circ$  anti-clockwise about the  $y$ -axis and then translated 2 units in the  $x$ -direction and 1 unit in the  $z$ -direction. Figure 13 shows the wedge being translated the same distance and then rotated the same angle. Note the difference of final wedge position and orientation of these two different transformations.

### Website for Computer Graphics

To ease the learning on fundamentals of computer graphics, many MATLAB files have been posted on a website established for this CAD course. Students can download these files as a basis for their assignments. Interesting files created by students are then uploaded to the website to broaden the collection.

The website is built with Microsoft Internet Explorer 4.0 to integrate Word documents and MATLAB files. The advantage of this web browser is that there is no need to convert Word documents to the HTML format. Therefore, class notes and homework assignments can be easily posted on the web.

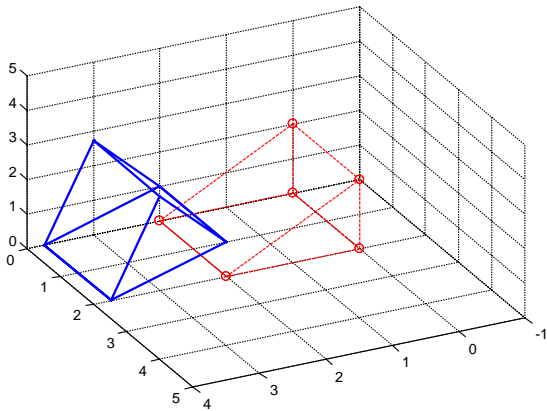


Figure 12 Rotation Followed by Translation

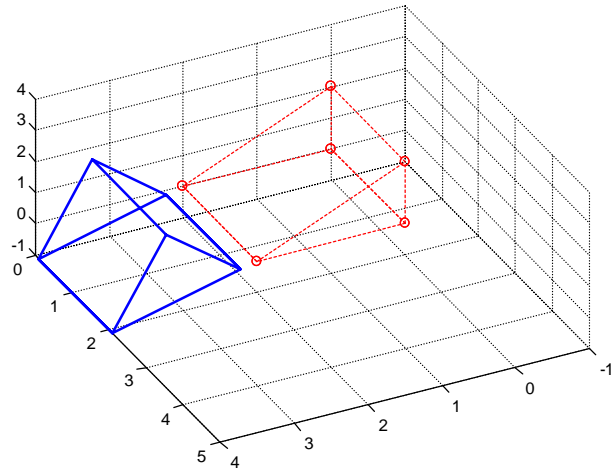


Figure 13 Translation Followed by Rotation

## Discussion

All files illustrated in this paper are derived from students' work. All MATLAB files can be accessed from the website through the hyperlink. The website invites students to explore the geometric transformation and characteristics of the synthetic curves and surfaces, and the feedback has been positive. Students are primarily enthused by the interactive learning which can be pursued at their leisure and by files that can be downloaded to their home PCs. Their feedback helped guide the further development of this website.

Sample files help students who do not have sufficient background in MATLAB. Through these examples, students developed more interesting work. Future work is toward interactive files similar to those by Rockwood and Chambers [7], which is developed in a different language environment, and the result will be collected in a CD for dissemination

## Bibliography

[1] Foley, J.D., van Dam, S.K., van Dam, J.F., Phillips, R.L., *Introduction to Computer Graphics*, Addison-Wesley, Reading, MA, 1993.

[2] Farin, G., *Computer Aided Geometric Design; From Theory to Applications*, 3rd edition, Academic Press, Boston, MA, 1995.

[3] Ibrahim Zeid, *CAD/CAM Theory and Practice*, McGraw-Hill, Inc., 1991

[4] Shah, J.J., and Mantyla M., *Parametric and Feature-Based CAD/CAM*, John Wiley & Sons, New York, NY, 1995.

[5] *MATLAB 5.2 User's Guide*, The MathWorks, Inc., Natick, MA, 1998.

[6] S. Nakamura, *Numerical Analysis and Graphical Visualization with MATLAB*, Prentice Hall, 1996

[7] A. Rockwood and P. Chambers, *Interactive Curves and Surfaces*, Morgan Kaufmann Publisher, 1996.

### **Shih-Liang Wang**

Shih-Liang Wang is Associate Professor in the Department of Mechanical Engineering at North Carolina A&T State University. Dr. Wang received his Ph.D. in mechanical engineering at Ohio State University in 1986. His research interest includes motion control and dynamic simulation of mechanical systems, and design of machines and mechanisms.