Efficient Intersection Computation of the Bezier and Hermite Curves with Axis Aligned Bounding Box

VACLAV SKALA
Dept. of Computer Science and Engineering
University of West Bohemia
Pilsen, CZECH REPUBLIC
ORCID 0000-0001-8886-4281

Abstract: Cubic parametric curves are used in many applications including the CAD/CAM systems. Especially the Hermite, Bezier and Coons formulations of a cubic parametric curve are used in E^2 and E^3 space. This paper presents efficient algorithm for the intersection computation of a cubic parametric curve with the Axis Aligned Bounding Box (AAB Box). Usual solution is to represent the cubic curve by a polyline, i.e. actually by sampled points of the given curve. However, this approach is dependent on the sampling frequency and can lead to problems especially in CAD/CAM systems and numerically controlled machines use.

Keywords: CAD/CAM systems, cubic parametric curves, Hermite curve, Bezier curve, Coons curve, parametric curve clipping, intersection, clipping, Axis Aligned Bounding Box, AAB Box, computer graphics, geometric modelling.

Received: April 30, 2021. Revised: September 28, 2021. Accepted: November 25, 2021. Published: December 21, 2021.

1. Introduction

This contribution describes a novel approach to intersection computation of a cubic parametric curve with the Axis Aligned Bounding Box (AAB Box) without resample the cubic curve to a polyline which is then clipped by the AAB Box using some standard clipping algorithms, e.g. Cohens-Sutherland clipping in E^2 and E^3 with a significant computational overhead.

In computer graphics and geometric modelling courses, mostly only the mathematical definition of the Bézier [1], Hermite and Coons cubic parametric curves are presented, see Solomon [2]. However, simple and understandable formal derivation of the Hermite is crucial for understanding, see the Appendix for relevant books.

The presented approach is based on the Bézier curve formulation. Note, that the Hermite form can be easily reformulated to the Bézier curve formulation.

If a parametric curve is to be rendered, the restriction of the AAB Box is to be respected, i.e. a rectangular area in the E^2 case or a volume in the E^3 case.

Usually, the drawing packages just sample the points of the given parametric curve forming a polyline, which is rendered and each segment of the polyline is clipped by the given AAB Box. This approach is simple, however, computations of the curve parts outside of the AAB Box is lost and the result depends on the parameter sampling frequency as well, e.g. cubic curve clipping by the AAB Box Xuefeng [3].

However, in some application such "simple" approach cannot be taken and the segments of a parametric curve inside of the AAB Box have to be determined exactly, e.g. in the case of numerically controlled machines etc.

Of course, the intersections can be computed as roots of a cubic equation WiKi [4] using Cardano's formula, however, such computation requires $\sqrt{}$ and $\sqrt[3]{}$ computations, casus irreducibilis, complex number representation and the algorithm is quite complex, etc., see WiKi [4] for details.

Generally, the intersection computation of a cubic curve with a line in the E^2 case leads to a system of nonlinear equations Eq.1:

$$\begin{aligned} x(u) &= a_x u^3 + b_x u^2 + c_x u + d_x \\ y(u) &= a_y u^3 + b_y u^2 + c_y u + d_y \\ curve \in E^2 &: \mathbf{c}(u) = (x(u), y(u)) \\ line \in E^2 &: \alpha x + \beta y + \gamma = 0 \end{aligned} \tag{1}$$

In the case of a cubic curve intersection with the AAB Box computation is simplified as the intersection computation with the horizontal lines and vertical lines which form the AAB Box is needed, only.

In the E^3 case, the AAB Box is formed by 6 bounding planes and it leads to a system of nonlinear equations Eq.2

$$\begin{aligned} x(u) &= a_x u^3 + b_x u^2 + c_x u + d_x \\ y(u) &= a_y u^3 + b_y u^2 + c_y u + d_y \\ z(u) &= a_z u^3 + b_z u^2 + c_z u + d_z \\ curve \in E^3 &: \mathbf{c}(u) = (x(u), y(u), z(u)) \\ plane \in E^3 &: \alpha x + \beta y + \gamma z + \delta = 0 \end{aligned} \tag{2}$$

It should be noted that for the Bézier cubic curve or the Hermite cubic curve the parameter u is restricted to $u \in < 0, 1 >$. It means that the intersections computations can be further simplified

It should be noted, that the Bézier curve of the degree n is defined as:

$$^{(B)}x(u) = \sum_{i=0}^{n} \binom{n}{i} u^i (1-u)^{n-i} x_{i+1} \tag{3}$$

where $x_1, \ldots, x_n$ are control points of the Bézier curve of the degree n.

The fundamental property of the Bézier curve is that all points

Supported by the University of West Bohemia - Institutional research support No.1311.

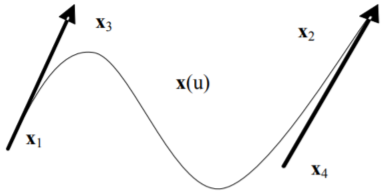

Fig. 1. Hermite cubic curve (tangential vectors shortened)

of the curve are inside of the convex hull of the Bézier control points for all parameter values $u \in < 0, 1 >$.

A. Hermite form

The Hermite form is based on cubic curve segment's end-points and tangential vectors in those points, see Fig.1 (tangential vectors are shortened). The Hermite form is not practical for interaction, but it is useful in smooth joining of cubic segments.

The tangential vectors of the Bézier form are defined as:

$$x'(0) = n(x_2 - x_1) \quad x'(1) = n(x_n - x_{n-1}) \tag{4}$$

From the Eq.4, it can be seen direct connection with the Hermite cubic curve formulation as the tangential vectors of the Hermite form are given as:

$$x'(0) = 3(x_2 - x_1) \quad x'(1) = 3(x_4 - x_3) \tag{5}$$

Therefore the "inner" control points of the Bézier form using the Hermite tangential vectors are given as:

$$\{u, x_2,\} = \{\frac{1}{3}, x_1 + \frac{1}{3}x'(0)\}, \quad x_3 = \{\frac{2}{3}, x_4 - \frac{1}{3}x'(1)\} \tag{6}$$

where $x'(0)$ and $x'(1)$ are tangential vectors of the Hermite form; similarly for the y-coordinate, resp. z-coordinate.
It means, that the points x_2, resp. x_3 have u coordinate $u = \frac{1}{3}$, resp. $u = \frac{2}{3}$ in the Bézier cubic curve formulation.

B. General form

There is a general form for common description of the Bézier, Hermite and Coons curves. It should be noted that the parametric curve represents only the $x(u)$-coordinate and for the other coordinates, i.e. $y(u)$, $z(u)$, it is similar.

Generally for the E^3 case, for a curve $\mathbf{C}(u)$ we can write:

$$\begin{aligned}\mathbf{C}(u) = [\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4]^T \ \mathbf{M}_F \ [u^3, u^2, u, 1]^T = \\ [\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4]^T \ \mathbf{M}_F \ \mathbf{u}\end{aligned} \tag{7}$$

where $\mathbf{P}_1, \dots, \mathbf{P}_4$ are control values of the given form, i.e. Bézier, resp. Hermite form.

- In the case of the Bézier form
$\mathbf{P}_1 = [x_1, y_1, z_1]^T$, $\mathbf{P}_2 = [x_2, y_2, z_2]^T$
$\mathbf{P}_3 = [x_3, y_3, z_3]^T$, $\mathbf{P}_4 = [x_4, y_4, z_4]^T$ are vectors of the curve control points
- In the case of the Hermite form
$\mathbf{P}_1 = [x_1, y_1, z_1]^T$, $\mathbf{P}_2 = [x_4, y_4, z_4]^T$ are vectors of the curve end-points,
$\mathbf{P}_3 = [x'_1, y'_1, z'_1]^T$, $\mathbf{P}_4 = [x'_4, y'_4, z'_4]^T$ are the tangential vectors at the curve end-points(notation at Fig.1 is used).

The Eq.7 can be rewritten as:

$$\mathbf{C}(u) = [\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4]^T \ \mathbf{M}_H \ \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \tag{8}$$

, i.e.

$$\begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = \begin{bmatrix} x_1^{(u)} & x_2^{(u)} & x_1^{(u)} & x_2^{(u)} \\ y_2^{(u)} & y_2^{(u)} & y_1^{(u)} & y_2^{(u)} \\ z_3^{(u)} & z_2^{(u)} & z_1^{(u)} & z_2^{(u)} \end{bmatrix} \ \mathbf{M}_H \ \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \tag{9}$$

It means that the Bézier and Hermite can be mutually transferable using linear operation using the transformation matrix $\mathbf{M}_{B \mapsto H}$, which is invertible.

$$\begin{aligned} \mathbf{M}_H &= \mathbf{M}_{B \mapsto H} \mathbf{M}_B \\ \mathbf{M}_B &= \mathbf{M}_{H \mapsto B} \mathbf{M}_H \\ \mathbf{M}_{B \mapsto H} &= \mathbf{M}_{H \mapsto B}^{-1} \end{aligned} \tag{10}$$

Similarly for other similar forms.

In the following, the algorithm for intersection of the Bézier cubic parametric curve with the AAB Box is analyzed.

2. Analysis of Parameter Intervals

The intersection computation of the Bézier cubic curve with the AAB Box in the E^2 space is presented at Fig.2. On the left part, the cubic curve $y(u)$ is shown with the blending functions, on the bottom part the cubic curve $x(u)$ is shown with the blending functions and in the right top part, the

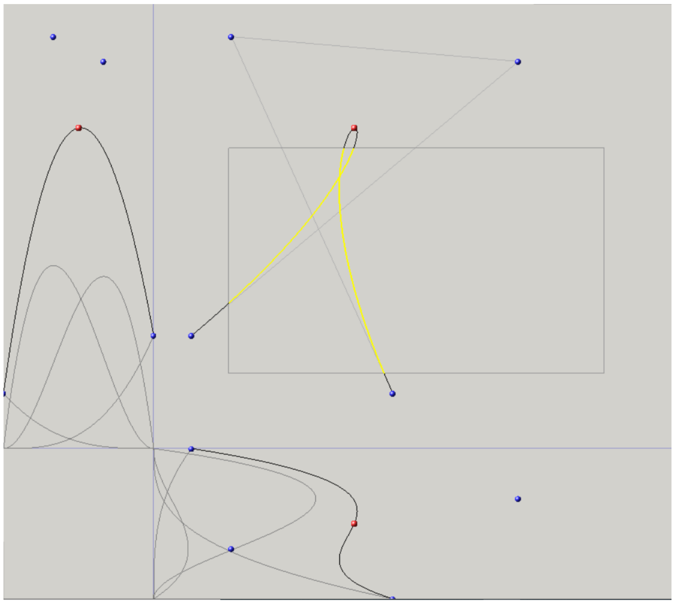

Fig. 2. Bézier cubic curve

parametric curve given by points $\mathbf{c}(u) = (x(u), y(u)) \in E^2$, $u \in< 0, 1 >$ is shown, together with the control points.

If the **all** control points are inside of the AAB Box then the whole curve is $\mathbf{c}(u)$ is inside, otherwise intersection computation is needed.

It can be seen, that the curve $\mathbf{c}(u)$ intersects the AAB Box several times and those points need to be determined. Lines forming the AAB Box are vertical or horizontal only. It means that intersections can be split to two separate computation for $x(u)$ and $y(u)$, leading to formulation:

$$\begin{aligned} y(u) &= a_y u^3 + b_y u^2 + c_y u + d_y \\ y &= y_{min} \quad or \quad y = y_{max} \\ &\qquad\qquad and \\ x(u) &= a_x u^3 + b_x u^2 + c_x u + d_x \\ x &= x_{min} \quad or \quad x = x_{max} \end{aligned} \tag{11}$$

where the AAB Box is given as $\mathbf{x}_{min} \times \mathbf{x}_{max}$ and $\mathbf{x}_{min} = (x_{min}, y_{min})$, $\mathbf{x}_{max} = (x_{max}, y_{max})$.

It means, that the equation Eq.11 is to be solve generally for two values min and max and for the each parametric curve, i.e $\mathbf{c}(u) = (x(u), y(u))$ in the E^2 case, and $\mathbf{c}(u) = (x(u), y(u), z(u))$ in the E^3 case.

$$\begin{aligned} \xi(u) &= au^3 + bu^2 + cu + d \\ \xi &= min \quad or \quad \xi = max \end{aligned} \tag{12}$$

It should be noted that the coefficients $\mathbf{a} = [a, b, c, d]$ differs for x, y, resp. z coordinates.

The parametric curve $\xi(u)$ has three stationary points, i.e. two extremes and one point of inflection given as $x'(u) = 0$, generally. The stationary points are defined as:

$$\begin{aligned} \xi'(u) = \frac{d}{du}\{au^3 + bu^2 + cu + d\} = \\ 3au^2 + 2bu + c = 0 \end{aligned} \tag{13}$$

Solving the quadratic equation Eq.13 two values u_1 and u_2 are obtain if exist.

For the extreme values of $\xi(u)$

- if $\xi''(u) < 0$ then it is a local maximum
- if $\xi''(u) > 0$ then it is a local minimum
- if $\xi''(u) = 0$ then it is an inflection

where the second derivative $\xi''(u)$ is given as:

$$\xi''(u) = \frac{d^2\xi(u)}{du^2} = 6au + 2b \tag{14}$$

Then the inflection point of the parametric curve

$$\begin{aligned} \xi''(u) &= 6au + 2b = 0 \\ u_{inflection} &= -\frac{2b}{6a} = -\frac{a}{3b} \end{aligned} \tag{15}$$

Note, that the $u_{inflection}$ value is different for $x(u)$, $y(u)$, resp. $z(u)$.

It means, that there are three intervals $(-\infty, u_1 >$, $< u_1, u_2 >, < u_2, \infty)$ and the cubic curve can have just one intersection in each interval with the parametric curve. However, the Bézier, Hermite and Coons cubic curves are defined for the interval $< 0, 1 >$ and therefore only the non-empty intervals $(-\infty, u_1 > \cap < 0, 1 >$, $< u_1, u_2 > \cap < 0, 1 >$ and $< u_2, +\infty) \cap < 0, 1 >$ has to be explored for an intersection in detail.

This analysis has to be done for the $x(u)$, $y(u)$, resp. $z(u)$ intervals of parameters in the E^2 case, resp. in the E^3 case. Now, three intervals of u are obtained for the each parametric curve, i.e.

- I_1^x, I_2^x, I_3^x for the curve $x(u)$
- I_1^y, I_2^y, I_3^y for the curve $y(u)$
- I_1^z, I_2^z, I_3^z for the curve $z(u)$ - in the case of E^3

Of course, only non-empty intervals, i.e. respecting that the parameter u is limited to $u \in< 0, 1 >$ are considered for the next processing and only non empty intervals given by the Eq.16 are used for finding parameters of intersection of the AAB Box and the parametric curve $\mathbf{c}(u) = (x(u), y(u))$ in the E^2case, resp. $\mathbf{c}(u) = (x(u), y(u), z(u))$ in the E^3 case.

$$\begin{aligned} \bigcap_{i,j,k=1}^{3,3,3} I_i^x \cap I_j^y \quad in\ the\ E^2\ case \\ \bigcap_{i,j,k=1}^{3,3,3} I_i^x \cap I_j^y \cap I_k^z \quad in\ the\ E^3\ case \end{aligned} \tag{16}$$

If the parametric curve $\xi(u)$ has different sign of the value at the given interval end-points, there is one and only one value of the parameter u for which $\xi(u) = 0$. It means, that range of numerical methods for finding a root of a non-linear function $f(x) = 0$ can be considered, e.g.:

- Binary search (bisection) WiKi [5]
- Regula falsi WiKi [6]
- Secant method WiKi [7]
- Newton's method WiKi [8]

It should be noted that the Newton's method [8] is not applicable directly, as $f'(x) = 0$ at the extreme points, see Eq.17, in spite of the high convergence promises.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{17}$$

Also the applicability criterion is required, see Eq.18 lead to some restrictions as well.

$$\left| \frac{f(x) f''(x)}{[f''(x)]^2} \right| \leq m < 1 \tag{18}$$

As the point of inflection is easily determined the Secant method WiKi [7] is the best choice.

The proposed method for finding intersections of the given parametric cubic curve with the AAB Box is simple as it requires solution of a quadratic equation Eq.13, computation of common parts of the parameters intervals Eq.16 and application of the Secant method WiKi [7]. Analysis of the secant method is described in Wikiversity [9].

It is important, that precise intersection points with the AAB Box are found precisely.

3. Conclusion

This contribution presents a new approach for for the Bézier and Hermite cubic parametric curve intersection with the Axis Aligned Bounding Box (AAB Box) using intersection of the curve computation. It eliminates problems with sampling of the curve and replacing the cubic curve by a polyline clipped by the AAB Box used in practice and causing precision problem at segments of the polyline partially inside of the AAB Box. Also, it eliminates need to repeatedly use the clipping algorithm, e.g. Cohen-Sutherland's clipping in E^2, resp. E^3 space causing unnecessary computational overhead.

Acknowledgement

The author would like to thank colleagues at the Shandong University (China) and University of West Bohemia for hints provided, to students and colleagues at the University of West Bohemia, Plzen and VSB-Technical University, Ostrava for their recent suggestions. Critical comments and recommendations given by anonymous reviewers led to significant improvements of this paper.

References

[1] P. Bézier, *The mathematical basis of the UNISURF CAD system*, ser. The Morgan Kaufmann Series in Computer Graphics. London: Butterworth Publ., 1986.

[2] D. Salomon, *Curves and Surfaces for Computer Graphics*. Berlin, Heidelberg: Springer-Verlag, 2005.

[3] X. Ao, Q. Fu, Z. Wu, X. Wang, M. Zhou, Q. Chen, and H. S. Seah, "An intersection algorithm for disk b-spline curves," *Computers & Graphics*, vol. 70, pp. 99–107, 2018, cAD/Graphics 2017.

[4] Wikipedia contributors, "Cubic equation — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Cubic_equation

[5] ——, "Binary search algorithm — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Binary_search_algorithm

[6] ——, "Regula falsi — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Regula_falsi

[7] ——, "Secant method — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Secant_method

[8] ——, "Newton's method — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Newton's_method

[9] Wikiversity, "Numerical analysis/the secant method — wikiversity,," 2020, [Online; accessed 4-October-2020]. [Online]. Available: https://en.wikiversity.org/wiki/Numerical_Analysis/The_Secant_Method

[10] D. Salomon, *The Computer Graphics Manual*. Springer, 2011.

[11] M. K. Agoston, *Computer Graphics and Geometric Modelling: Mathematics*. Berlin, Heidelberg: Springer-Verlag, 2005.

[12] ——, *Computer Graphics and Geometric Modelling: Implementation & Algorithms*. Berlin, Heidelberg: Springer-Verlag, 2004.

[13] E. Lengyel, *Mathematics for 3D Game Programming and Computer Graphics*. USA: Charles River Media, Inc., 2001.

[14] J. Vince, *Introduction to the Mathematics for Computer Graphics*, 3rd ed. Berlin, Heidelberg: Springer-Verlag, 2010.

[15] J. D. Foley, A. van Dam, S. Feiner, and J. F. Hughes, *Computer graphics - principles and practice, 2nd Edition*. Addison-Wesley, 1990.

[16] J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, J. D. Foley, S. K. Feiner, and K. Akeley, *Computer Graphics - Principles and Practice, 3rd Edition*. Addison-Wesley, 2014.

[17] R. S. Ferguson, *Practical Algorithms for 3D Computer Graphics*, 2nd ed. USA: A. K. Peters, Ltd., 2013.

[18] P. Shirley and S. Marschner, *Fundamentals of Computer Graphics*, 3rd ed. USA: A. K. Peters, Ltd., 2009.

[19] S. Marschner and P. Shirley, *Fundamentals of Computer Graphics, Fourth Edition*, 4th ed. USA: A. K. Peters, Ltd., 2016.

[20] T. Theoharis, N. Platis, G. Papaioannou, and N. Patrikalakis, *Graphics and Visualization: Principles & Algorithms (1st ed.)*. A K Peters/CRC Press, 2008.

[21] P. J. Schneider and D. H. Eberly, *Geometric Tools for Computer Graphics*, ser. The Morgan Kaufmann Series in Computer Graphics. San Francisco: Morgan Kaufmann, 2003.

[22] L. Ammeraal and K. Zhang, "Computer graphics for java programmers," *Computer Graphics for Java Programmers*, pp. 1–387, 2017.

[23] A. Thomas, *Integrated Graphic and Computer Modelling*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[24] D. D. Hearn, M. P. Baker, and W. Carithers, *Computer Graphics with OpenGL*, 4th ed. USA: Prentice Hall Press, 2010.

[25] F. S. Hill and S. M. Kelley, *Computer Graphics Using OpenGL (3rd Edition)*. USA: Prentice-Hall, Inc., 2006.

Appendix A

Details on parametric cubic curves and bicubic patches can be found, e.g., in:

- Salomon,D.: The Computer Graphics Manual [10],
- Salomon,D.: Computer Graphics and Geometric Modeling [2],
- Agoston,M.K.: Computer Graphics and Geometric Modelling: Mathematics [11],
- Agoston,M.K.: Computer Graphics and Geometric Modelling: Implementation & Algorithms [12],
- Lengyel,E.: Mathematics for 3D Game Programming and Computer Graphics [13],
- Vince,J.: Introduction to the Mathematics for Computer Graphics [14],
- Foley,J.D., van Dam,A., Feiner,S., Hughes,J.F.: Computer graphics - principles and practice [15],
- Hughes,J.F., van Dam,A., McGuire,M., Sklar,D.F., Foley,J.D., Feiner,S.K., Akeley,K.: Computer Graphics - Principles and Practice [16],
- Ferguson,R.S.: Practical Algorithms for 3D Computer Graphics [17],
- Shirley,P., Marschner,S.: Fundamentals of Computer Graphics [18],
- Marschner,S., Shirley,P.: Fundamentals of Computer Graphics Marschner [19],
- Theoharis,T., Platis,N., Papaioannou,G., Patrikalakis,N.: Graphics and Visualization: Principles & Algorithms [20],
- Schneider,P.J., Eberly,D.H.: Geometric Tools for Computer Graphics [21],
- Ammeraal,L., Zhang,K.: Computer graphics for Java programmers [22],
- Vince,J: Mathematics for Computer Graphics [14],
- Thomas,A.: Integrated Graphic and Computer Modelling [23],
- Hearn,D. and Baker,M.P., Warren,C.: Computer Graphics with OpenGL [24],
- F. S. Hill and S. M. Kelley: Computer Graphics Using OpenGL [25].

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US