

# A Comparison of Three Curve Intersection Algorithms

Thomas W. Sederberg  
Scott R. Parry  
Brigham Young University  
Provo, Utah 84602

## *EXTENDED ABSTRACT*

An empirical comparison is made between three algorithms for computing the points of intersection of two planar Bezier curves. The algorithms compared are:

- A1. The well known Bezier subdivision algorithm, which is discussed in [Lane'80]
- A2. A subdivision algorithm based on interval analysis due to Koparkar and Mudur [Koparkar'83]
- A3. An algorithm due to Sederberg, Anderson and Goldman which reduces the problem to one of finding the roots of a univariate polynomial [Sederberg'84].

The details of these three algorithms are presented in their respective references. This abstract mentions major implementation choices, and assumes that the reader is familiar with the three algorithms.

The execution times of A1 and A2 are highly sensitive to implementation. We observed that the decision on when to terminate the subdivision can effect execution speed by an order of magnitude. We based our termination criterion on a suggestion from [Wang'84] that a curve be subdivided until its deviation from a straight line segment is no larger than a value  $\epsilon$ . If  $(x_i, y_i) \quad i = 0, 1, \dots, n$  are the control points of a degree  $n$  Bezier curve and

$$L_0 = \max_{0 \leq i \leq n-2} (|x_i - 2x_{i+1} + x_{i+2}|, |y_i - 2y_{i+1} + y_{i+2}|),$$

then

$$r_0 = \log_4 \frac{\sqrt{3}n(n-1)L_0}{8\epsilon}$$

where  $r_0$  is the number of times the curve must be subdivided in order for a curve segment to approximate a line segment to within a tolerance  $\epsilon$ .

For A1 and A2, the convex hull was taken to be a bounding rectangle. For A1, subdivision was performed using the geometric construction algorithm and the bounding rectangle was determined by the new Bezier control points. For A2, the polynomial equations were converted to standard power basis, and subdivision was performed by simply evaluating the curve segment at its parametric midpoint, using Horner's algorithm.

For A3, the implicitization procedure was performed using Bernstein polynomials. It was noted that standard polynomials introduced excessive roundoff error for curves of degree larger than three. A critical part of the implementation is a polynomial root finder which finds only real roots. We developed our own root finder for this purpose, which usually runs significantly faster than, for example, the standard IMSL root finder which finds *all* roots (real and complex) of a polynomial.

We tested these three algorithms on pairs of Bezier curves of degree three, four, and five, with several different numbers of intersection points. Our goal was to compute the intersection points to five decimal places of accuracy, and we experimented to find the optimal value of  $r_0$  for this purpose.

**Results:** The tests were run on a VAX11/750, with floating point accelerator, under UNIX 4.2 BSD. Our test results indicate that for two cubic Bezier curves which intersect once, A3 is at least twice as fast as the other two algorithms. For more than one intersection, A3 does even better -- if the two cubic curves intersect nine times, A3 is typically five or six times faster than the others.

For curves of degree four, A2 and A3 are comparable, and A1 is the slowest. For curves of degree five, A2 is typically twice as fast as the other two algorithms.

Multiple intersections were not thoroughly tested. Our implementation of A1 and A2 did not properly detect the presence of tangency. A3 handled tangency with no problem by reporting a multiple polynomial root.

## References

- Koparkar, P.A. and Mudur, S.P.(1983), "A new class of algorithms for processing parametric curves", *Computer-Aided Design*, Vol. 15, No. 1, 41-45.
- Lane, J.M., and Riesenfeld, R.F.(1980), "A theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI, No. 1,35-46.
- Sederberg, T.W., Anderson, D.C. and Goldman, R.N.(1984), "Implicit representation of parametric curves and surfaces", *Computer Vision, Graphics and Image Processing*, Vol. 28, 72-84.
- Wang, G. (1984), "The Subdivision Method for Finding the Intersection between Two Bezier Curves or Surfaces", *Zhejiang University Journal*, Special Issue on Computational Geometry.