

CAGD Winter 2001
Project # 2
2-D Curve Plotting Package
Due: Wednesday, 6 February

Dr. Thomas W. Sederberg

February 7, 2001

Add the following commands to CPLOT:

BEZI All subsequent curves will be plotted as Bézier curves, until a **BFUN** command is given. **BEZI** is the default.

BFUN

deg,n

$f_0^1, f_1^1, \dots, f_{deg}^1$

$f_0^2, f_1^2, \dots, f_{deg}^2$

...

...

$f_0^n, f_1^n, \dots, f_{deg}^n$

Use the user defined blending functions in plotting subsequent curves, where blending function $f^i(t)$ is defined: $f^i(t) = f_0^i + f_1^i t + f_2^i t^2 + \dots + f_{deg}^i t^{deg}$ and the curve is defined $\mathbf{P}(t) = \mathbf{P}_0 f^0(t) + \mathbf{P}_1 f^1(t) + \dots + \mathbf{P}_n f^n(t)$.

BPRInt

ncurve

Write to a file the control points and weights of the indicated rational Bézier curve. Format the output so that a classmate would understand it.

CPPL

ncurve, radius

Plot the control polygon for curve number **ncurve**. Plot the control points as curves with radius **radius**, where the units for **radius** are in terms of the 0—1 screen coordinate system. Thus, a **radius** of .5 would fill the entire page. If **radius** = 0, do not plot control points.

CURVature

ncurve, tmin, tmax, nval

Write to a file the curvature at **nval** parameter values, beginning at **tmin** and ending at **tmax**.

ELEVate**ncurve, number of elevations, new curve number**

Degree elevate the specified Bézier curve the specified number of times. Store the degree elevated curve in the indicated curve number.

OFFSet**ncurve, radius**

Plot the offset of the specified rational Bézier curve. For a positive radius, the offset should be on the left side of the tangent vector

SUBDdivide**ncurve, t, nleft, nright**

Subdivide curve number **ncurve** at parameter value **t** using the de Casteljau algorithm. Store the left portion of the curve as curve number **nleft** and the right portion as curve number **nright**.

Create and run the following six data files. Each data file will have four viewports.

test1.dat

In the top-left viewport, plot a Timmer curve. Using the same four control points, plot a Ball curve in the top-right viewport. Using the same four control points, plot an Overhauser curve in the lower left viewport, and using the same four control points, plot a cubic Bézier curve in the bottom right viewport.

test2.dat

Top-left: Ball cubic with $\mathbf{P}_1 = \mathbf{P}_2$

Top-right: Quadratic Bézier with same three control point locations that you used for the Ball cubic. .

Bottom-left: A two-segment Timmer curve (ie., for the first segment, use control points $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ and for the second segment, use control points $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4$).

Bottom-right: A four-segment Timmer curve which is a closed loop (ie., for the first segment, use control points $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$, for the second segment, use control points $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_0$, for the third segment, use control points $\mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_0, \mathbf{P}_1$, and for the fourth segment, use control points $\mathbf{P}_3, \mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$).

test3.dat

Top-left: A degree four Bézier curve, with its control polygon (zero radius in the control points).

Top-right: Subdivide the curve in half and plot the two curves and their control polygons.

Bottom-left: Subdivide the curve into fourths and plot the four curves and their control polygons.

Bottom-right: Subdivide the curve into eights and plot the eight curves and their control polygons.

test4.dat

Top-left: Plot the same Bézier curve, with its control polygon, as used in test3.dat.

Top-right: Plot the same curve, with control polygon, degree elevated to degree 5.

Bottom-left: Plot the same curve, with control polygon, degree elevated to degree 6. Bottom-right: Plot the same curve, with control polygon, degree elevated to degree 20.

test5.dat

Top-left: Plot a degree four Bézier curve and three offset curves of any radius (that fit in the window).

Top-right: Plot a circle and a degree three Bézier curve which closely approximates a quarter circle (as found in your homework assignment).

Bottom-left: Plot a unit circle at the origin and a degree four Bézier curve with control points $(1, 0, 1)$, $(1, 0.3927, 1)$, $(0.790485, 0.790485, 1)$, $(0.3927, 1, 1)$, $(0, 1, 1)$. Print curvature and derivative values at 11 points along the Bézier curve.

Bottom-right: Plot a circle and a degree two rational Bézier curve which exactly expresses a quarter circle. Print curvature and derivative values at 11 points along the Bézier curve.

test6.dat

Top-left: Plot a polynomial quadratic Bézier curve along with its control polygon. Then set $w_1 = 2$ and plot the resulting curve. Do it again for $w_1 = 0$, $w_1 = 5$, $w_1 = 10$ and $w_1 = 100$.

Top-right: Do the same thing with a cubic Bézier curve.

Bottom-left: Plot a quadratic Bézier curve with control points $P_0 = (1, 0)$, $P_1 = (1, 1)$, $P_2 = (0, 1)$ and weights $w_0 = 1$, $w_1 = -1$ (that's right, negative 1), and $w_2 = 2$.

Bottom right: Make up your own set of user-defined blending functions. The functions should be degree four, should involve four control points, should interpolate the two end control points, and should NOT be symmetric. Plot a curve using those blending functions, and plot it a second time using the same control points in reverse order.

Hand in the plot of each file, with your name plotted on each sheet, and also a printout showing the curvature and derivative values for indicated curves in test5.dat.