Chapter 20

POLYNOMIAL APPROXIMATION OF RATIONAL CURVES

A degree r plane rational Bézier curve

$$\begin{aligned}\mathbf{R}(t) &= \frac{\sum_{i=0}^{r} B_i^r(t) w_i \mathbf{R}_i}{\sum_{i=0}^{r} B_i^r(t) w_i} \\ &= (x_R(t), y_R(t)) \qquad (\mathbf{R}_i = (X_i, Y_i))\end{aligned} \tag{20.1}$$

can be expressed as a degree n plane polynomial Bézier curve $\mathbf{H}(t)$ with a *moving control point* $\mathbf{M}(t)$:

$$\begin{aligned}\mathbf{H}(t) &\equiv \mathbf{R}(t) \\ &= \sum_{i=0, i\neq m}^{n} B_i^n(t)\mathbf{P}_i + \mathbf{M}(t) B_m^n(t) \\ &= (x_H(t), y_H(t)) \quad (0 < m < n),\end{aligned} \tag{20.2}$$

where

$$\mathbf{M}(t) = \frac{\sum_{i=0}^{r} B_i^r(t) w_i \mathbf{M}_i}{\sum_{i=0}^{r} B_i^r(t) w_i} \tag{20.3}$$

is a degree r plane rational Bézier curve. The curve $\mathbf{H}(t)$ is called a *Hybrid Curve* [SK91]. Figure 20.1.a shows a rational cubic curve, and Figure 20.1.b shows that same curve represented as a degree two hybrid curve. Figure 20.1.c shows how to evaluate a point $\mathbf{H}(t)$ on a hybrid curve — you first evaluate the moving control point at $\mathbf{M}(t)$. The polynomial Bézier curve can then be evaluated, treating point $\mathbf{M}(t)$ as a stationary control point ($\mathbf{P}_1$ in this case). Figure 20.1.d shows the curve in Figure 20.1.a represented as a degree four hybrid curve.

As derived in [SK91], the $\mathbf{P}_i$ and $\mathbf{M}_i$ can be computed:

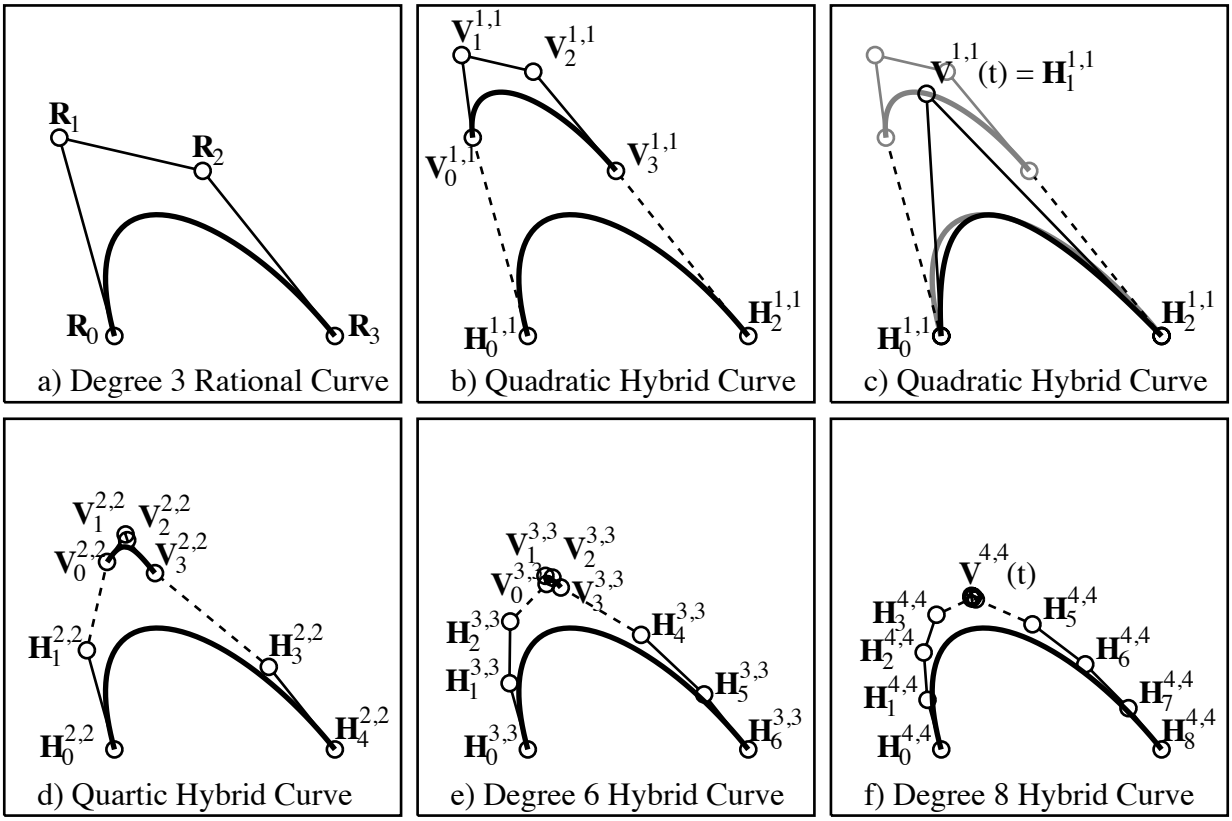

Figure 20.1: Hybrid curves

For $i = 0, \ldots, m-1$:

$$\begin{aligned} \mathbf{P}_0 &= \mathbf{R}_0, \\ \mathbf{P}_i &= \mathbf{R}_0 + \frac{\sum_{j=1}^{\min(r,i)} \binom{r}{j}\binom{n}{i-j} w_j[\mathbf{R}_j - \mathbf{P}_{i-j}]}{w_0\binom{n}{i}}. \end{aligned} \tag{20.4}$$

For $i = n+r, \ldots, m+1+r$:

$$\begin{aligned} \mathbf{P}_n &= \mathbf{R}_r, \\ \mathbf{P}_{i-r} &= \mathbf{R}_r + \frac{\sum_{j=\max(0,i-n)}^{r-1} \binom{r}{j}\binom{n}{i-j} w_j[\mathbf{R}_j - \mathbf{P}_{i-j}]}{w_r\binom{n}{i-r}}. \end{aligned} \tag{20.5}$$

For $i = 0, \ldots, r$:

$$\mathbf{M}_i = \sum_{\substack{j = \max(0, i+m-n) \\ j \neq i}}^{\min(i+m,r)} \frac{\binom{r}{j}\binom{n}{i+m-j} w_j[\mathbf{R}_j - \mathbf{P}_{i+m-j}]}{w_i\binom{n}{m}\binom{r}{i}} + \mathbf{R}_i. \tag{20.6}$$

The moving control point is always a rational curve of the same degree as the original rational curve. Also, the control point weights of the moving control point curve is the same as those of the original rational curve.

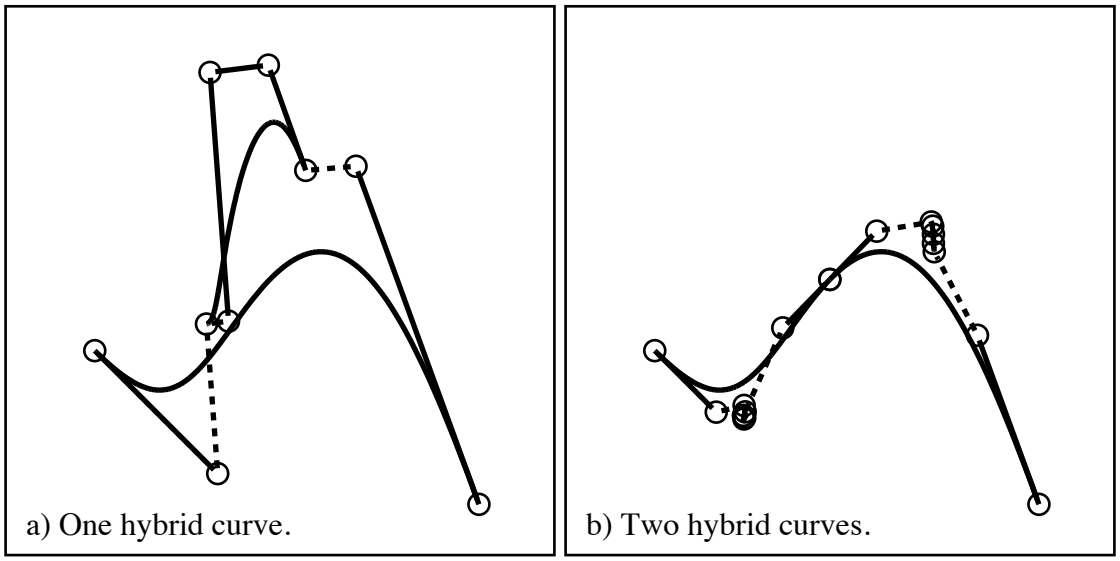

Figure 20.2: Hybrid curves after subdivision

If we simply replace the moving control point $\mathbf{M}(t)$ with a stationary control point $\mathbf{P}_m$, the hybrid representation $\mathbf{H}(t)$ of the rational curve $\mathbf{R}(t)$ leads directly to a polynomial approximation $\mathbf{P}(t)$ (as in equation 20.10). An intelligent choice of point $\mathbf{P}_m$ is the midpoint of the x/y min-max

box which bounds the moving control point. Denote the width of such a bounding box as $2|\triangle_x|$, the heighth $2|\triangle_y|$, and the diagonal $2||\triangle||$. If n is an even number, $m = n/2$, and $||\mathbf{M}(t) - \mathbf{P}_m|| \leq ||\triangle||$, then

$$||\mathbf{P}(t) - \mathbf{H}(t)|| \leq ||\triangle|| \left(\begin{array}{c} n \\ \frac{n}{2} \end{array} \right) \frac{1}{2^n}, \tag{20.7}$$

$$\begin{aligned} |x_P(t) - x_H(t)| &\leq B_{xPH} \\ &= |\triangle_x| \left(\begin{array}{c} n \\ \frac{n}{2} \end{array} \right) \frac{1}{2^n}, \end{aligned} \tag{20.8}$$

$$\begin{aligned} |y_P(t) - y_H(t)| &\leq B_{yPH} \\ &= |\triangle_y| \left(\begin{array}{c} n \\ \frac{n}{2} \end{array} \right) \frac{1}{2^n}. \end{aligned} \tag{20.9}$$

Note that $||\triangle||$ for the quartic hybrid curve is smaller than $||\triangle||$ for the quadratic hybrid curve. This is often, though not always, the case, depending on the radius of convergence. Conditions under which convergence does occur is discussed in [WS93]. However, as illustrated in Figure 20.2, each half of a subdivided rational curve always has a smaller value of $||\triangle||$ than does the original rational curve.

20.1 PLANAR AREAS

Shape design often calls for the computation of integral values such as the cross-sectional area of ducts or the center of gravity of a ship's cross-section. For plane regions bounded by polynomial Bézier curves, closed form solutions exist. This chapter reviews those solutions and also presents a numerical algorithm for the case of where the bounding curves are in *rational* Bézier form. The earliest such algorithm for polynomial curves is probably due to Faux and Pratt [FP79]. General algorithms have been developed by Liu, Nowacki, and Lu [Liu87, NLL90].

Most algorithms for polynomial curves, such as evaluation, subdivision and degree elevation, extend readily to rational curves, while other algorithms, such as those involving derivatives and integrals, do not. Ultimately, the problem of computing areas, centroids, or volumes of revolution for regions bounded by rational Bézier curves reduces to the problem of integrating a rational function (that is, a polynomial divided by another polynomial) for which there exist basically two approaches: the method of partial fractions, and numerical integration.

Recall that partial fractions is a basic technique from calculus in which a rational function of t is written as the sum of a polynomial and fractions of the form

$$\frac{A}{(at+b)^k} \quad \text{or} \quad \frac{At+B}{(at^2+bt+c)^k},$$

where A, B, a, b and c are constants and k is a positive integer. This method leads to straightforward closed-form integration formulae for degree two rational functions, and significantly more complicated formulae for degree three and four cases. Closed-form integration formulae do not generally exist for degree greater than four.

Numerical integration [BF89], using methods such as the trapezoid rule, Simpson's rule, etc., is the technique most often applied in practice. But for rational functions, determining an error bound is expensive, since it involves finding a bound on high order derivatives of a rational function.

20.2 Integrals Involving Plane Bézier Curves

This section reviews general algorithms for computing integral values on plane Bézier curves, as given by D. Liu and H. Nowacki [Liu87, NLL90].

Let $\mathbf{P}_i = (x_i, y_i) \in R^2, i = 0, 1, \ldots, n$, then

$$\begin{aligned} \mathbf{P}(t) &= \sum_{i=0}^{n} B_i^n(t)\mathbf{P}_i \\ &= (x_P(t), y_P(t)) \quad (\mathbf{P}_i = (x_i, y_i)) \end{aligned} \tag{20.10}$$

is a degree n plane Bézier curve.

First, define the following:

$$\begin{aligned} C_{i,j}^{(n)} &= \int_0^1 B_i^n(t) B_j^{n-1}(t) dt, \\ D_{i,j}^{(n)} &= C_{i,j-1}^{(n)} - C_{i,j}^{(n)} \\ &= \begin{cases} -\frac{1}{2n} & (i,j) = (0,0) \\ \frac{1}{2n} & (i,j) = (n,n) \\ \dfrac{(j-i)\binom{n}{i}\binom{n}{j}}{(i+j)(2n-i-j)\binom{2n}{i+j}} & 0 \le i,j \le n, (i,j) \ne (0,0), (n,n). \end{cases} \end{aligned} \tag{20.11}$$

$$\begin{aligned} C_{i,j,k}^{(n)} &= \int_0^1 B_i^n(t) B_j^n(t) B_k^{n-1}(t) dt, \\ D_{i,j,k}^{(n)} &= C_{i,j,k-1}^{(n)} - C_{i,j,k}^{(n)} \\ &= \begin{cases} -\frac{1}{3n} & (i,j,k) = (0,0,0) \\ \frac{1}{3n} & (i,j,k) = (n,n,n) \\ \dfrac{(2k-i-j)\binom{n}{i}\binom{n}{j}\binom{n}{k}}{(i+j+k)(3n-i-j-k)\binom{3n}{i+j+k}} & 0 \le i,j,k \le n, (i,j,k) \ne (0,0,0), (n,n,n). \end{cases} \end{aligned} \tag{20.12}$$

The **area under a curve** given by equation 20.10 over the interval $[x_0, x_n]$ (see Figure 20.3) is:

$$\begin{aligned} Area &= \int_{x_0}^{x_n} y_P dx_P \\ &= n \sum_{i,j=0}^{n} D_{i,j}^{(n)} y_i x_j, \qquad (\frac{dx_P(t)}{dt} \ne 0, \quad 0 \le t \le 1). \end{aligned} \tag{20.13}$$

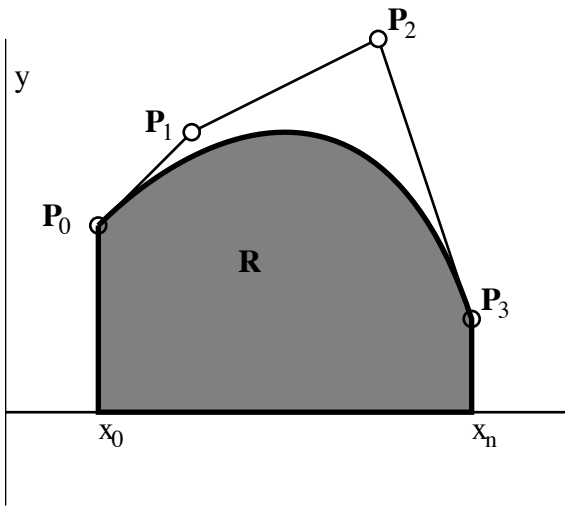

Figure 20.3: Area under cubic Bézier curve

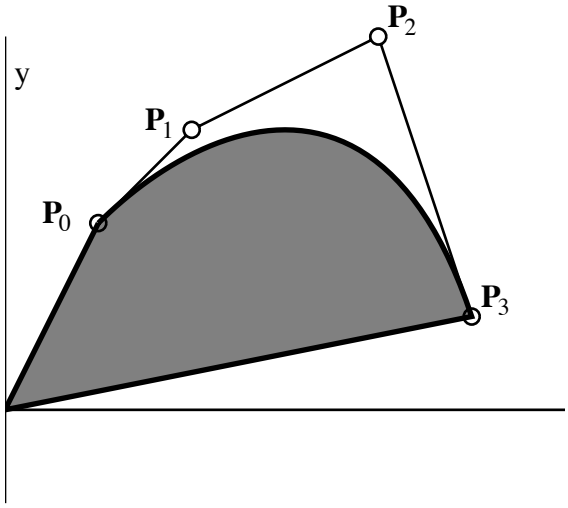

Figure 20.4: Area of cubic Bézier sector

The **area contained in the sector** formed by the origin and a curve given by equation 20.10 (see Figure 20.4) is:

$$\begin{aligned} \textit{Directed Area} &= \frac{1}{2}\int_0^1 \mathbf{P}(t) \wedge \frac{d\mathbf{P}(t)}{dt} dt \\ &= n \sum_{\substack{i,j=0 \\ i<j}}^{n} D_{i,j}^{(n)} (\mathbf{P}_i \wedge \mathbf{P}_j), \qquad (\mathbf{P}(t) \wedge \frac{d\mathbf{P}(t)}{dt} \neq 0, \quad 0 \leq t \leq 1), \end{aligned} \tag{20.14}$$

where symbol $\wedge$ denotes a wedge product, and

$$\mathbf{P}_i \wedge \mathbf{P}_j = \begin{vmatrix} x_i & y_i \\ x_j & y_j \end{vmatrix}. \tag{20.15}$$

The **first moments of area** of R (in Figure 20.3) about the x- and y-axes are:

$$\begin{aligned} I_x &= \int_{x_0}^{x_n} \frac{1}{2} y_P^2 dx_P \\ &= \frac{n}{2} \sum_{i,j,k=0}^{n} D_{i,j,k}^{(n)} y_i y_j x_k. \end{aligned} \tag{20.16}$$

$$\begin{aligned} I_y &= \int_{x_0}^{x_n} x_P y_P dx_P \\ &= n \sum_{i,j,k=0}^{n} D_{i,j,k}^{(n)} x_i y_j x_k. \end{aligned} \tag{20.17}$$

The **volume** of a solid of revolution generated by revolution of region R (in Figure 20.3) about the x-axis is:

$$\begin{aligned} \textit{Volume} &= \int_{x_0}^{x_n} \pi y_P^2 dx_P \\ &= 2\pi I_x. \end{aligned} \tag{20.18}$$

20.3 Error Bounds on Integration

Computing areas, etc., of rational Bézier curves can be done by first approximating the rational curve with a polynomial curve (as discussed in chapter 20), then applying the closed form equations for polynomial curves. This section discusses how to determine error bounds on the resulting integration values.

From [Flo92], bounds for $\mathbf{P}'(t)$, $\mathbf{R}'(t)$, $\mathbf{P}(t)$, and $\mathbf{R}(t)$ are

$$\begin{aligned}
|x'_P(t)| &\leq B_{x'P} \\
&= n \cdot max_{0\leq i\leq n-1}|\triangle x_i|, && (20.19)\\
|y'_P(t)| &\leq B_{y'P} \\
&= n \cdot max_{0\leq i\leq n-1}|\triangle y_i|, && (20.20)\\
|x'_R(t)| &\leq B_{x'R} \\
&= r \cdot \left(\frac{max_{0\leq i\leq r}w_i}{min_{0\leq i\leq r}w_i}\right)^2 \cdot max_{0\leq i\leq r-1}|\triangle X_i|, && (20.21)\\
|y'_R(t)| &\leq B_{y'R} \\
&= r \cdot \left(\frac{max_{0\leq i\leq r}w_i}{min_{0\leq i\leq r}w_i}\right)^2 \cdot max_{0\leq i\leq r-1}|\triangle Y_i|, && (20.22)\\
|x_P(t)| &\leq B_{xP} \\
&= max_{0\leq i\leq n}|x_i|, && (20.23)\\
|y_P(t)| &\leq B_{yP} \\
&= max_{0\leq i\leq n}|y_i|, && (20.24)\\
|x_R(t)| &\leq B_{xR} \\
&= max_{0\leq i\leq r}|X_i|, && (20.25)\\
|y_R(t)| &\leq B_{yR} \\
&= max_{0\leq i\leq r}|Y_i|. && (20.26)
\end{aligned}$$

Of special interest is the error bound $\left|\int_{x_0}^{x_n} y_R dx_R - \int_{x_0}^{x_n} y_P dx_P\right|$ which expresses the error in applying integration formulae directly to polynomial approximations of rational curves:

$$\begin{aligned}
\int_{x_0}^{x_n} y_R dx_R - \int_{x_0}^{x_n} y_P dx_P &= \int_0^1 y_R x'_R dt - \int_0^1 y_P x'_P dt \\
&= \int_0^1 y_H x'_H dt - \int_0^1 y_P x'_P dt \\
&= \int_0^1 (y_H x'_H - y_H x'_P + y_H x'_P - y_P x'_P) dt \\
&= \int_0^1 y_H d(x_H - x_P) + \int_0^1 (y_H - y_P) x'_P dt \\
&= -\int_0^1 (x_H - x_P) y'_H dt + \int_0^1 (y_H - y_P) x'_P dt,
\end{aligned}$$

and hence

$$\begin{aligned}
\left|\int_{x_0}^{x_n} y_R dx_R - \int_{x_0}^{x_n} y_P dx_P\right| &\leq \int_0^1 |x_H - x_P| \cdot |y'_H| dt + \int_0^1 |y_H - y_P| \cdot |x'_P| dt \\
&\leq B_{xPH} \cdot B_{y'R} + B_{yPH} \cdot B_{x'P}. && (20.27)
\end{aligned}$$

Likewise,

$$\begin{aligned}
\left|\int_0^1 \mathbf{R}\wedge\frac{d\mathbf{R}}{dt}dt-\int_0^1 \mathbf{P}\wedge\frac{d\mathbf{P}}{dt}dt\right| &= \left|\int_0^1 \mathbf{H}\wedge\frac{d\mathbf{H}}{dt}dt-\int_0^1 \mathbf{P}\wedge\frac{d\mathbf{P}}{dt}dt\right| \\
&= \left|\int_0^1 \left\{\mathbf{H}\wedge(\frac{d\mathbf{H}}{dt}-\frac{d\mathbf{P}}{dt})\right\}dt+\int_0^1\left\{(\mathbf{H}-\mathbf{P})\wedge\frac{d\mathbf{P}}{dt}\right\}dt\right| \\
&= \left|\int_0^1 \begin{vmatrix} x_H & y_H \\ x'_H-x'_P & y'_H-y'_P\end{vmatrix}dt+\int_0^1\begin{vmatrix} x_H-x_P & y_H-y_P \\ x'_P & y'_P\end{vmatrix}dt\right| \\
&= \left|\int_0^1 x_H(y'_H-y'_P)dt-\int_0^1 y_H(x'_H-x'_P)dt+\int_0^1(x_H-x_P)y'_Pdt\right. \\
&\quad \left.-\int_0^1(y_H-y_P)x'_Pdt\right| \\
&= \left|-\int_0^1(y_H-y_P)x'_Hdt+\int_0^1(x_H-x_P)y'_Hdt+\int_0^1(x_H-x_P)y'_Pdt\right. \\
&\quad \left.-\int_0^1(y_H-y_P)x'_Pdt\right| \\
&= \left|\int_0^1(x_H-x_P)(y'_H+y'_P)dt-\int_0^1(y_H-y_P)(x'_H+x'_P)dt\right| \\
&\leq \int_0^1|x_H-x_P|(|y'_H|+|y'_P|)dt+\int_0^1|y_H-y_P|(|x'_H|+|x'_P|)dt \\
&\leq B_{xPH}\cdot(B_{y'R}+B_{y'P})+B_{yPH}\cdot(B_{x'R}+B_{x'P}). \qquad (20.28)
\end{aligned}$$

$$\begin{aligned}
\left|\int_{x_0}^{x_n} y_R^2dx_R-\int_{x_0}^{x_n}y_P^2dx_P\right| &= \left|\int_0^1 y_H^2x'_Hdt-\int_0^1 y_P^2x'_Pdt\right| \\
&= \left|\int_0^1 y_H^2(x'_H-x'_P)dt+\int_0^1 y_H(y_H-y_P)x'_Pdt+\right. \\
&\quad \left.\int_0^1 y_P(y_H-y_P)x'_Pdt\right| \\
&= \left|-2\int_0^1(x_H-x_P)y_Hy'_Hdt+\int_0^1(y_H-y_P)(y_H+y_P)x'_Pdt\right| \\
&\leq 2\int_0^1|x_H-x_P||y_H||y'_H|dt+\int_0^1|y_H-y_P|(|y_H|+|y_P|)|x'_P|dt \\
&\leq 2(B_{xPH}\cdot B_{y'R}\cdot B_{yR})+B_{yPH}\cdot B_{x'P}\cdot(B_{yR}+B_{yP}). \qquad (20.29)
\end{aligned}$$

$$\begin{aligned}
\left|\int_{x_0}^{x_n} x_R y_R dx_R - \int_{x_0}^{x_n} x_P y_P dx_P\right| &= \left|\int_0^1 x_H y_H x'_H dt - \int_0^1 x_P y_P x'_P dt\right| \\
&= \left|\int_0^1 x_H y_H (x'_H - x'_P) dt + \int_0^1 x_H (y_H - y_P) x'_P dt + \right. \\
&\quad \left. \int_0^1 (x_H - x_P) y_P x'_P dt\right| \\
&= \left|-\int_0^1 (x_H - x_P)(x'_H y_H + x_H y'_H) dt + \int_0^1 (y_H - y_P) x_H x'_P dt + \right. \\
&\quad \left. \int_0^1 (x_H - x_P) y_P x'_P dt\right| \\
&= \left|\int_0^1 (x_H - x_P)(-x'_H y_H - x_H y'_H + x'_P y_P) dt + \right. \\
&\quad \left. \int_0^1 (y_H - y_P) x_H x'_P dt\right| \\
&\leq \int_0^1 |x_H - x_P|(|x'_H||y_H| + |x_H||y'_H| + |x'_P||y_P|) dt + \\
&\quad \int_0^1 |y_H - y_P||x_H||x'_P| dt \\
&\leq B_{xPH} \cdot (B_{x'R} \cdot B_{yR} + B_{y'R} \cdot B_{xR} + B_{x'P} \cdot B_{yP}) + \\
&\quad B_{yPH} \cdot B_{x'P} \cdot B_{xR}. \qquad (20.30)
\end{aligned}$$

20.4 Examples

Since the integration error is a function of the hybrid curve bounding box size, that error can be controlled by recursively splitting $\mathbf{R}(t)$ in half until the bounding boxes are small enough to satisfy the ultimate integration error limit. Then, the exact integration formulae for polynomial curves can be used.

We have implemented this algorithm and compared it against two standard methods of numerical integration: the trapezoid rule, and Simpson's rule. It turns out that the trapezoid rule can be easily modified to return an error bound when dealing with rational Bézier curves. Imagine we are computing the area between the x axis and a rational Bézier curve. Then the top of each trapezoid is a line segment which approximates a segment of the rational Bézier curve. The convex hull of the control points of that curve segment then serves as a bound on how accurately a given trapezoid approximates the differential area.

The traditional way to determine an error bound for Simpson's rule requires us to obtain a bound on the fourth derivative of the integrand $x'(t)y(t)$ (for areas) or $\pi x'(t)y^2(t)$ (for volumes). This is possible, but for a degree r rational Bézier curve, the fourth derivative of the area (volume) integrand is a rational function of degree $48r$ ($64r$). Instead of using this robust error bound in our implementation of Simpson's rule, we used the heuristic of comparing the answer obtained using m intervals with the answer obtained using $2m$ intervals. This, of course, provides only a general indication of the error, not a guarantee of precision.

Figures 20.5 and 20.6 show two test cases in which the shaded area and volume of revolution about the x axis were computed. Figure 20.6 is actually a semi-circle expressed as a cubic rational

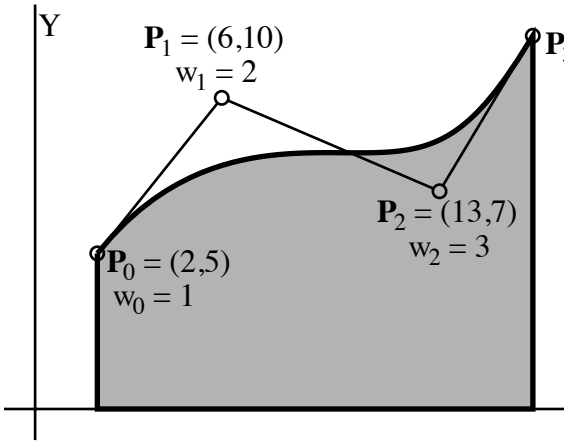

Figure 20.5: Example 1

Table 20.1: Execution times for Figure 5 (10^{-3} seconds)

Method	Tolerance = 10^{-3}		Tolerance = 10^{-6}	
	Area	Volume	Area	Volume
Degree 8 Hybrid	1	6	4	12
Degree 6 Hybrid	2	5	5	14
Simpson	4	15	7	61
Trapezoid	6	210	40	1560

Bézier curve.

Tables 20.4 and 20.4 list the execution time in milliseconds. Tests were run on a workstation rated at 65 Specmarks. The exact area of the semi-circle is $\pi/8 = .39269908\ldots$, and the computed area using hybrid curves is 0.39269906 using a tolerance of 10^{-6}, and 0.39272477 using a tolerance of 10^{-3}.

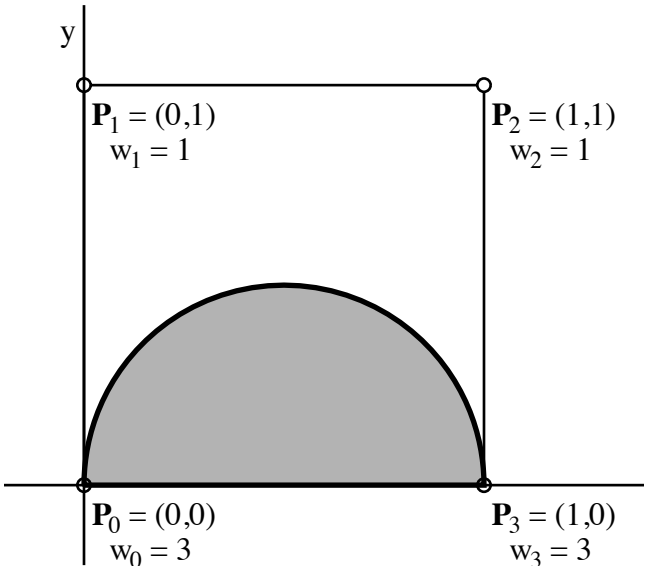

Figure 20.6: Example 2

Table 20.2: Execution times for Figure 6 (10^{-3} seconds)

Method	Tolerance = 10^{-3}		Tolerance = 10^{-6}	
	Area	Volume	Area	Volume
Degree 8 Hybrid	0.7	1.8	1.9	3.4
Degree 6 Hybrid	0.7	1.0	2.2	3.2
Simpson	0.2	0.5	1.0	1.0
Trapezoid	0.7	1.0	23	34