2000-04-12

Bounding Interval Rational Bézier Curves with Interval Polynomial Bézier Curves

Thomas W. Sederberg
tom@cs.byu.edu

Falai Chen

Wenping Lou

Bounding Interval Rational Bézier Curves with Interval Polynomial Bézier Curves

Falai Chen
Department of Mathematics
University of Science and Technology of China
Hefei, Anhui, 230026, China
chenfl@ustc.edu.cn

Thomas W. Sederberg
Computer Department
Brigham Young University
Provo, Utah 84602, USA
tom@cs.byu.edu

Wenping Lou
Department of Mathematics
University of Science and Technology of China
Hefei, Anhui, 230026, China
wplou@263.net

Abstract

In this paper, we put forward and study the problem of bounding an interval rational Bézier curve with an interval polynomial Bézier curve. We propose three different methods—Hybrid Method, Perturbation Method and Linear Programming Method to solve this problem. Examples are illustrated to compare the three different methods. The empirical results show that the Perturbation Method and the Linear Programming Method produce much tighter bounds than the Hybrid Method, though they are computationally several times more expensive.

1 Introduction

In the communities of Approximation Theory and Computer Aided Geometric Design(CAGD), there is considerable interest in approximating functions (or curves/surfaces) with functions (or curves/surfaces) of simpler forms, for example, in approximating rational functions (or curves) with polynomial functions (or curves). Much literature has focused on these problems from different points of views, e.g., from pure approximation theory to applications in CAGD. However, as far as the authors are aware, almost all the related work is concerned with how accurate the approximation is and seldom consider the problem of how to transfer the approximation errors into subsequent applications. This problem can be important in some applications, e.g., in tolerance analysis in CAD and in numerical analysis. To solve this problem, Sederberg, et. al. [2] introduced interval forms of curves and surfaces. Based on the new representations of curves and surfaces, several authors developed robust algorithms for geometric operations such as curve/curve intersections in CAD/CAM systems ([3]–[8]).

In this paper, we are interested in the following problem, a problem that has potential applications in CAD and numerical analysis:

Given an interval rational function (or curve), bound it with an interval polynomial function (or curve) such that the bound is as tight as possible.

We will develop three different methods–Hybrid Method, Perturbation Method and Linear Programming Method to solve the problem in the following sections. Testing and comparisons are made between the three different approaches through illustration of examples. The empirical results show that the Hybrid Method generally produces much looser bounds than the other two methods, though it is computationally less expensive.

2. Interval Bézier Curves

An *interval* $[a, b]$ is the set of real numbers $\{x | a \leqslant x \leqslant b\}$. Interval arithmetic operations are defined by

$$[a, b] * [c, d] = \{x * y | x \in [a, b] \text{ and } y \in [c, d]\}, \qquad (1)$$

where $*$ represents an arithmetic operation, $* \in \{+, -, \cdot, /\}$. One can verify that

$$\begin{aligned}
[a, b] + [c, d] = &\quad [a + c, b + d] \\
[a, b] - [c, d] = &\quad [a - d, b - c] \\
[a, b] \cdot [c, d] = &\quad [\min(ac, ad, bc, bd), \\
&\quad \max(ac, ad, bc, bd)] \\
[a, b]/[c, d] = &\quad [a, b] \cdot [1/d, 1/c], \quad 0 \notin [c, d].
\end{aligned} \qquad (2)$$

Interval arithmetic is an important tool in numerical analysis, and it has many applications in other areas. For details, the reader is referred to Moore's book *Interval Analysis* [1].

An *interval polynomial* is a polynomial whose coefficients are intervals:

$$[p](t) := \sum_{k=0}^{n} [a_k, b_k] B_k^n(t), \quad 0 \leqslant t \leqslant 1, \qquad (3)$$

where $B_k^n(t) = \binom{n}{k} t^k (1-t)^{n-k}$, $k = 0, 1, \ldots, n$ are Bernstein bases.

An interval polynomial can be also expressed in the form

$$[p](t) = [p_{\min}(t), p_{\max}(t)], \qquad 0 \leqslant t \leqslant 1, \qquad (4)$$

where

$$p_{\min}(t) = \sum_{k=0}^{n} a_k B_k^n(t)$$

and

$$p_{\max}(t) = \sum_{k=0}^{n} b_k B_k^n(t)$$

$p_{\min}(t)$ and $p_{\max}(t)$ are called *lower bound* (denoted by $lb([p](t))$) and *upper bound* (denoted by $ub([p](t))$) of $[p](t)$ respectively.

The *width* of an interval polynomial can be defined by

$$W([p](t)) = ||p_{\max}(t) - p_{\min}(t)||, \qquad (5)$$

where the norm $||.||$ is the standard norm, such as $||.||_1$, $||.||_\infty$ and $||.||_2$.

An *interval rational function* is defined

$$[r](t) := \frac{\sum_{k=0}^{n} [a_k, b_k] \omega_k B_k^n(t)}{\sum_{k=0}^{n} \omega_k B_k^n(t)}, \quad 0 \leqslant t \leqslant 1, \qquad (6)$$

where ω_k, $k = 0, 1, \ldots$ are weights (It is also possible to define the weights to be intervals. Since the algorithms in the following sections are similar for both cases, we assume the weights are real numbers in this paper). It is similar to define the upper bound, the lower bound and the width of an interval rational function.

An *interval polynomial Bézier curve* is a polynomial Bézier curve whose control points are vector-valued intervals (i.e., rectangular regions in a plane):

$$[\mathbf{P}](t) = \sum_{i=0}^{n} [\mathbf{P}_i] B_i^n(t), \qquad (7)$$

where $[\mathbf{P}_i] = [a_i, b_i] \times [c_i, d_i] = ([a_i, b_i], [c_i, d_i])$, $i = 0, 1, \ldots, n$ are the interval control points.

Similarly, an *interval rational Bézier curve* is defined

$$[\mathbf{R}](t) = \frac{\sum_{i=0}^{n} [\mathbf{R}_i] \omega_i B_i^n(t)}{\sum_{i=0}^{n} w_i B_i^n(t)}, \qquad (8)$$

where $[\mathbf{R}_i]$ and $w_i > 0$, $i = 0, 1, \ldots, n$ are interval control points and weights respectively.

An interval (polynomial or rational) Bézier curve defines a region (a slender tube) in the plane which consists of all the Bézier curves whose control points satisfy $\mathbf{P}_i \in [\mathbf{P}_i]$ for $i = 0, 1, \ldots, n$. Figure 1 shows a sample cubic interval polynomial Bézier curve. An interval polynomial (or rational) Bézier curve can be viewed as a vector-valued interval polynomial (or rational function). For example, the interval polynomial Bézier curve (7) can be rewritten in the form

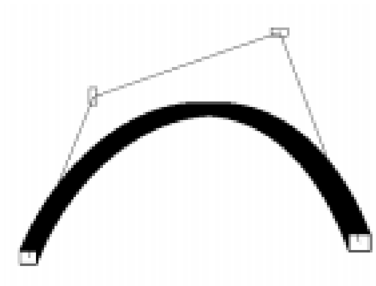

Figure 1. A sample interval Bézier curve

$$[\mathbf{P}](t) = ([x](t), [y](t)), \qquad (9)$$

where $[x](t)$ and $[y](t)$ are interval polynomials:

$$[x](t) = \sum_{k=0}^{n} [a_k, b_k] B_k^n(t) \qquad (10)$$

$$[y](t) = \sum_{k=0}^{n} [c_k, d_k] B_k^n(t) \qquad (11)$$

With this form, the problem of bounding interval rational Bézier curves with interval polynomial Bézier curves can be conveniently converted to the problem of bounding interval rational functions with interval polynomials.

3. Bounding Interval Rational Bézier Curves with Interval Polynomial Bézier Curves

Before dealing with the problem of bounding interval rational Bézier curves with interval polynomial Bézier curves, we solve the problem of bounding an interval rational function with an interval polynomial.

Problem 1 *Given an interval rational function*

$$[r](t) := \frac{\sum_{k=0}^{n} [r_k] \omega_k B_k^n(t)}{\sum_{k=0}^{n} w_k B_k^n(t)}, \qquad (12)$$

find an interval polynomial

$$[p](t) := \sum_{k=0}^{m} [p_k] B_k^n(t) \qquad (13)$$

such that $[p](t)$ *bounds* $[r](t)$

$$[r](t) \subset [p](t), \quad \textit{for } 0 \leqslant t \leqslant 1, \tag{14}$$

and that the width of $[p](t)$ *is as tight as possible*. $[p](t)$ is called an *interval polynomial bound* of $[r](t)$.

Since the process of finding an upper bound is similar to that of finding a lower bound, we can solve the following problem instead

Problem 2 *Given a rational function of degree* n

$$R(t) := \frac{\sum_{k=0}^{n} R_k \omega_k B_k^n(t)}{\sum_{k=0}^{n} \omega_k B_k^n(t)}, \tag{15}$$

find a polynomial of degree m

$$P(t) := \sum_{k=0}^{m} P_k B_k^n(t) \tag{16}$$

such that

$$P(t) \geqslant R(t), \qquad 0 \leqslant t \leqslant 1 \tag{17}$$

and

$$||P(t) - R(t)|| \tag{18}$$

is minimized.

In the following, we will propose three different methods–Hybrid Method, Perturbation Method and Linear Programming Method to solve the above problem.

3.1. Hybrid Method

In [2], Sederberg, et al proposed a method called the *hybrid curve method* to approximate rational functions with polynomials. Indeed, this method also provides a way to compute an upper polynomial bound of a rational function. The main idea of the Hybrid Method is as follows.

Any rational function $R(t)$ can be expressed as a polynomial, one of whose coefficients is a rational function:

$$\begin{aligned} R(t) &= \frac{\sum_{k=0}^{n} r_k \omega_k B_k^n(t)}{\sum_{k=0}^{n} \omega_k B_k^n(t)} \\ &\equiv \sum_{k=0, k\neq l}^{m} P_k B_k^m(t) + M(t) B_l^m(t), \end{aligned} \tag{19}$$

where $M(t)$ is a rational function of degree n:

$$M(t) = \frac{\sum_{k=0}^{n} M_k \omega_k B_k^n(t)}{\sum_{k=0}^{n} \omega_k B_k^n(t)}, \tag{20}$$

and $l = [m/2]$ is an integer.

The coefficients of P_i, $i = 0, 1, \ldots, m, i \neq l$ and M_i, $i = 0, 1, \ldots, n$ can be computed as follows:

For $i = 0, 1, \ldots, l-1$,

$$\begin{aligned} P_0 &= R_0, \\ P_1 &= R_0 + \frac{n\omega_1(R_1 - P_0)}{m\omega_0}, \\ &\vdots \\ P_i &= R_0 + \frac{\sum_{j=1}^{\min(n,i)} \binom{n}{j}\binom{m}{i-j}\omega_j(R_j - P_{i-j})}{\binom{m}{i}\omega_0}. \end{aligned} \tag{21}$$

For $i = m+n, \ldots, n+l+1$,

$$\begin{aligned} P_m &= R_n, \\ P_{m-1} &= R_n + \frac{n\omega_{n-1}(R_{n-1} - P_m)}{m\omega_n}, \\ &\vdots \\ P_{i-n} &= R_n + \frac{\sum_{j=\max(0,i-m)}^{n-1} \binom{n}{j}\binom{m}{i-j}\omega_j(R_j - P_{i-j})}{\binom{m}{i-n}\omega_n}, \end{aligned} \tag{22}$$

and

$$M_i = \sum_{j=\max(0,i+l-m)}^{\min(i+l,n)} \frac{\binom{n}{j}\binom{m}{i+l-j}\omega_j(R_j - P_{i+l-j})}{\binom{m}{l}\binom{n}{i}\omega_i}, \quad i = 0, 1, \ldots, n. \tag{23}$$

Theorem 1 *Let* $M_{max} = \max_{0\leqslant k\leqslant n} M_k$ *and* $M_{min} = \min_{0\leqslant k\leqslant n} M_k$. *Then polynomial*

$$P(t) = \sum_{k=0, k\neq l}^{m} P_k B_k^m(t) + M_{max} B_l^m(t) \tag{24}$$

is an upper bound of $R(t)$*, and*

$$||P(t) - R(t)||_\infty \leqslant c(M_{max} - M_{min}), \tag{25}$$

where c *is a constant:*

$$c = \binom{m}{l}\left(\frac{l}{m}\right)^l \left(\frac{m-l}{m}\right)^{m-l}. \tag{26}$$

Proof: Since

$$M(t) \leqslant M_{\max}, \qquad 0 \leqslant t \leqslant 1,$$

from (19), one has

$$R(t) \leqslant \sum_{k=0, k\neq l}^{m} P_k B_k^m(t) + M_{\max} B_l^m(t) = P(t),$$

i.e., P(t) is a polynomial upper bound of $R(t)$.

On the other hand,

$$||P(t) - R(t)||_\infty \leqslant (M_{\max} - M_{\min}) \max_{0\leqslant t\leqslant 1} B_l^m(t).$$

Since $B_l^m(t)$ attains maximum value at $t = \frac{l}{m}$ in $[0, 1]$, (25) follows immediately.

The Hybrid Method generally provides a loose bound for a rational function, and it has the disadvantage that there is a restricted condition under which the polynomial converges to the rational function when $m \to \infty$. For the specific convergence condition, the reader is referred to [10].

3.2. Perturbation Method

The Hybrid Method presented in the last subsection generally produces quite a loose bound. In this subsection, we will propose an improved algorithm to solve Problem 2.

Given rational function (15), we perturb $R(t)$ with another rational function $\epsilon(t)$ such that $R(t)+\epsilon(t)$ is a polynomial

$$R(t)+\epsilon(t)=P(t):=\sum_{k=0}^{m} P_k B_k^m(t), \tag{27}$$

where

$$\epsilon(t)=\frac{\sum_{k=0}^{m+n}\epsilon_k\omega'_k B_k^{m+n}(t)}{\sum_{k=0}^{m+n}\omega'_k B_k^{m+n}(t)} \tag{28}$$

and

$$\omega'_k=\frac{1}{\binom{m+n}{m}}\sum_{i+j=k}\omega_i\binom{k}{i}\binom{m+n-k}{m-i}. \tag{29}$$

From (27), we have

$$\sum_{k=0}^{n} R_k\omega_k B_k^n(t)+\sum_{k=0}^{m+n}\epsilon_k\omega'_k B_k^{m+n} = \sum_{i=0}^{m} P_i B_i^m(t)\sum_{j=0}^{n}\omega_j B_j^n(t). \tag{30}$$

Writing both sides of the above equation in Bernstein form and comparing the coefficients, one get

$$\epsilon_k=\sum_{i=\max(k-n,0)}^{\min(k,m)}(P_i-R_{k-i})\alpha_{k,i}, \tag{31}$$

$k=0,1,\ldots,m+n$, where for $\max(k-n,0)\leqslant i\leqslant \min(k,m)$,

$$\alpha_{k,i}=\frac{\omega_{k-i}\binom{k}{i}\binom{m+n-k}{m-i}}{\sum_{j+l=k}\omega_l\binom{k}{j}\binom{m+n-k}{m-j}}, \tag{32}$$

otherwise, $\alpha_{k,i}=0$. Now we wish to make the perturbation $\epsilon(t)$ is as small as possible. To this end, we minimize $\sum_{k=0}^{m+n}\epsilon_k^2$. Let

$$L(P_0,\ldots,P_m):=\sum_{k=0}^{m+n}\epsilon_k^2 := \left(\sum_{i=\max(k-n,0)}^{\min(k,m)}(P_i-R_{k-i})\alpha_{k,i}\right)^2 \tag{33}$$

From

$$\frac{\partial L}{\partial P_l} = 2\sum_{k=0}^{m+n}\sum_{i=\max(k-n,0)}^{\min(k,m)}(P_i-R_{k-i})\alpha_{k,i}\alpha_{k,l} = 0,\quad l=0,1,\ldots,m, \tag{34}$$

we obtain a system of linear equations for P_i, $i=0,1,\ldots,m$:

$$\sum_{i=0}^{m}\left(\sum_{k=\max(i,l)}^{\min(i+n,l+n)}\alpha_{k,i}\alpha_{k,l}\right)P_i = \sum_{i=0}^{m}\left(\sum_{k=\max(i,l)}^{\min(i+n,l+n)}\alpha_{k,i}\alpha_{k,l}\right)R_{k-i}, \tag{35}$$

$l=0,1,\ldots,m$. By solving the above equations and substituting P_i into (31), the values of ϵ_i are obtained.

Now a polynomial upper bound of $R(t)$ can be computed as follows.

Theorem 2 *Let $\epsilon_{max} = max_{0\leqslant i\leqslant m+n}\epsilon_i$ and $\epsilon_{min} = min_{0\leqslant i\leqslant m+n}\epsilon_i$. Then polynomial $P(t)-\epsilon_{min}$ is an upper bound of $R(t)$, and*

$$||P(t)-\epsilon_{min}-R(t)||_\infty\leqslant\epsilon_{max}-\epsilon_{min}. \tag{36}$$

Proof: From (27), we have

$$R(t)\leqslant P(t)-\epsilon_{\min},$$

which means $P(t)-\epsilon_{\min}$ is an upper bound of $R(t)$.

To get (36), one need only notice that $P(t)-R(t)=\epsilon(t)$.

3.3. Linear Programming Method

In this approach, we take

$$||P(t)-R(t)||_1=\frac{1}{m+1}\sum_{k=0}^{m}P_k-c \tag{37}$$

as the minimization target, where $c=\int_0^1 R(t)dt$ is a constant.

To satisfy condition (17), one must have

$$\sum_{k=0}^{m}P_kB_k^m(t)\sum_{k=0}^{n}\omega_kB_k^n(t)\geqslant\sum_{k=0}^{n}R_k\omega_kB_k^n(t), \tag{38}$$

for any $t\in[0,1]$. Or equivalently,

$$\sum_{k=0}^{m+n}\sum_{i+j=k}P_i\omega_j\binom{k}{i}\binom{m+n-k}{m-i}B_k^{m+n}(t) \geqslant\sum_{k=0}^{m+n}\sum_{i+j=k}R_j\omega_j\binom{k}{i}\binom{m+n-k}{m-i}B_k^{m+n}(t).$$

Thus a sufficient condition for (17) to hold is

$$d_k := \sum_{i+j=k} (P_i - R_j)\omega_j \binom{k}{i}\binom{m+n-k}{m-i} \geqslant 0, \quad (39)$$

$k = 0, 1, \ldots, m+n$. An immediate consequence of the above discussion is the following

Theorem 3 *Let $P_i, i = 0, 1, \ldots, m$ be the solutions of the following linear programming problem*

$$\begin{cases} Min & \sum_{i=0}^{m} P_i \\ s.t. & d_k \geqslant 0, \quad k = 0, 1, \ldots, m+n. \end{cases} \quad (40)$$

Then polynomial $P(t)$ defined in (16) is an upper bound of $R(t)$, and

$$||P(t) - R(t)||_\infty \leqslant max_{\,0\leqslant k\leqslant m+n} \frac{d_k}{\omega''_k}, \quad (41)$$

where

$$\omega''_k = \sum_{i=max(k-n,0)}^{min(k,m)} \omega_{k-i}\binom{k}{i}\binom{m+n-k}{m-i}, \quad (42)$$

k=0,1, . . . ,m+n.

Proof: We need only prove the second part of the theorem. To estimate $||P(t) - R(t)||_\infty$, we write $P(t) - R(t)$ in a rational Bernstein polynomial:

$$P(t) - R(t) = \frac{\sum_{k=0}^{m+n} \frac{1}{\binom{m+n}{m}} d_k B_k^{m+n}(t)}{\sum_{k=0}^{m+n} \omega'_k B_k^{m+n}(t)},$$

where w'_k is defined as in (29). Now (41) follows directly from the above equation.

Remarks: The Hybrid Method automatically solves the *constrained upper bound problem*, i.e., in addition to satisfying (15—18), the following constraints must be held:

$$\begin{cases} P^{(i)}(0) = R^{(i)}(0) \\ P^{(i)}(1) = P^{(i)}(1) \end{cases} \quad i = 0, 1, \ldots, \mu, \quad (43)$$

where $\mu \leqslant [\frac{m-1}{2}]$. The Perturbation Method and the Linear Programming Method can also be adapted to solve the constrained upper bound problem. For the Perturbation Method, just set $\epsilon_k = 0$, $k = 0, 1, \ldots, \mu, m-\mu, \ldots, m$, and solve for the corresponding P_k from (31). The remaining P_k can be obtained from (35) with $l = \mu + 1, \ldots, m-\mu-1$. For the Linear Programming Method, P_k, $k = 0, 1, \ldots, \mu, m-\mu, \ldots, m$ can be solved by setting the corresponding $d_k = 0$, and the remaining P_k are obtained by solving the linear programming problem (40).

3.4. Bounding Interval Rational Bézier Curves with Interval Polynomial Bézier Curves

In the previous subsections, we derived three different methods to bound an interval rational function with an interval polynomial. These three methods lead directly to algorithms of bounding an interval rational Bézier curve with an interval polynomial Bézier curve. The main results are based on the following fact.

Theorem 4 *Given an interval rational Bézier curve of degree n*

$$[\mathbf{R}](t) = ([x](t), [y](t)). \quad (44)$$

If

$$[\bar{x}](t) = \sum_{i=0}^{m} [\bar{x}_i] B_i^m(t) \quad (45)$$

and

$$[\bar{y}](t) = \sum_{i=0}^{m} [\bar{y}_i] B_i^m(t) \quad (46)$$

are degree m interval polynomial bounds of $[x](t)$ and $[y](t)$ respectively, then interval polynomial Bézier curve

$$[\mathbf{P}](t) = ([\bar{x}](t), [\bar{y}](t)) = \sum_{i=0}^{m} ([\bar{x}_i], [\bar{y}_i]) B_i^m(t) \quad (47)$$

bounds interval rational Bézier curve $[\mathbf{R}](t)$, i.e., $[\mathbf{R}](t) \subset [\mathbf{P}](t)$.

Proof: Straightforward.

To measure how tight a bound is, we define the *bounding error* as follows

$$\begin{aligned} e([x](t), [\bar{x}](t)) &= \max(||ub([x](t)) - ub([\bar{x}](t))||_\infty, \\ &\qquad ||lb([x](t)) - lb([\bar{x}](t))||_\infty) \\ e([y](t), [\bar{y}](t)) &= \max(||ub([y](t)) - ub([\bar{y}](t))||_\infty, \\ &\qquad ||lb([y](t)) - lb([\bar{y}](t))||_\infty) \\ e([\mathbf{R}](t), [\mathbf{P}](t)) &= \max(e([x](t), [\bar{x}](t)), \\ &\qquad e([y](t), [\bar{y}](t))) \end{aligned} \quad (48)$$

It is a little hard to compute the bounding errors by the above definitions. Fortunately, Theorems 1, 2 and 3 give a good estimation for the bounding error by each of the three methods.

For a given interval rational Bézier curve $[\mathbf{R}](t)$ and tolerance $\epsilon > 0$, we use one of the three methods to find an interval polynomial Bézier curve $[\mathbf{P}](t)$ to bound $[\mathbf{R}](t)$. If the bounding error is larger than ϵ, we subdivide $[\mathbf{R}](t)$ at parameter value $t = 1/2$ and then bound each segment with an interval polynomial Bézier curve respectively. This process is continued until the bounding error is less than ϵ for each segment. At last, we find a piecewise interval polynomial Bézier curve which bound the original interval rational Bézier curve $[\mathbf{R}](t)$.

4. Examples and Comparison

In this section, we will compare the bounding errors and the computational costs between the three methods–Hybrid Method (HM), Perturbation Method (PM) and Linear Programming Method (LPM) for bounding an interval rational Bézier curve with an interval polynomial Bézier curve through illustration of examples. We implemented these examples on a 266 MHZ Pentium PC.

Example 1 Let $[r](t)$ be a cubic interval rational function with the following interval coefficients and weights:

$$\begin{array}{ll} [a_0, b_0] = [0, 1/4], & \omega_0 = 1, \\ [a_1, b_1] = [1, 6/5], & \omega_1 = 2, \\ [a_2, b_2] = [-1, -3/4], & \omega_2 = 2, \\ [a_3, b_3] = [1, 7/6], & \omega_3 = 1 \end{array}$$

We use degree $m = 4, 5$ and 6 interval polynomials to bound $[r](t)$ respectively. The bounding errors (B.E.) and the computational time (C.T.) in CUP seconds for the three methods are list in Table.1. Figure.2–Figure.10 show the corresponding figures.

Table 1. Comparison of the bounding errors and the computational costs (I)

m	H M		P M		L P M	
	B.E.	C.T.	B.E.	C.T.	B.E.	C.T.
$m = 4$	3.00	0.02	0.53	0.11	1.02	0.17
$m = 5$	3.68	0.02	0.33	0.11	0.13	0.22
$m = 6$	2.25	0.02	0.22	0.16	0.10	0.22

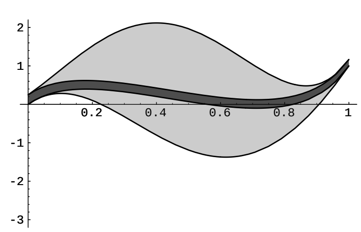

Figure 2. Hybrid Method $m = 4$

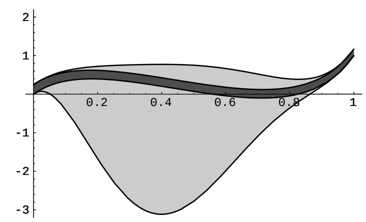

Figure 3. Hybrid Method $m = 5$

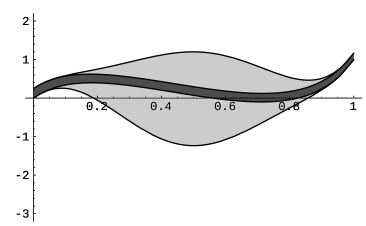

Figure 4. Hybrid Method $m = 6$

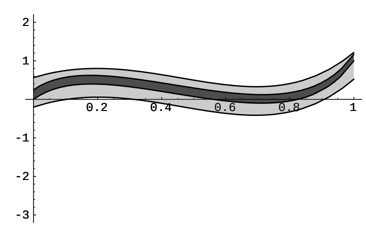

Figure 5. Perturbation Method $m = 4$

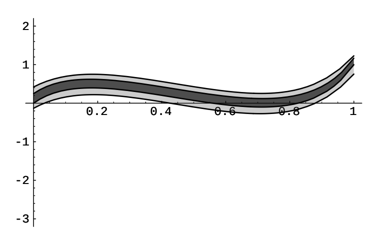

Figure 6. Perturbation Method $m = 5$

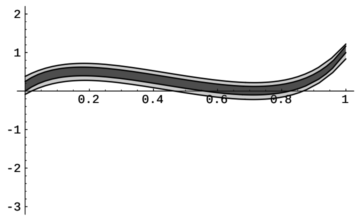

Figure 7. Perturbation Method $m = 6$

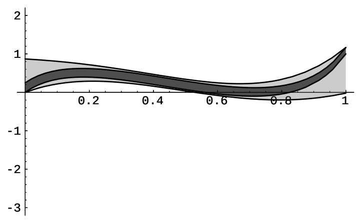

Figure 8. Linear Programming Method $m = 4$

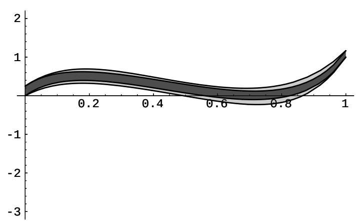

Figure 9. Linear Programming Method $m = 5$

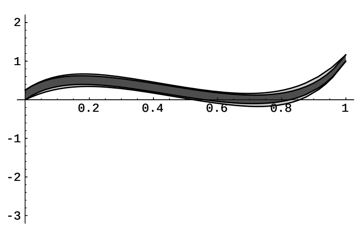

Figure 10. Linear Programming Method $m = 6$

Example 2 In this example, we consider a case where the Hybrid Method diverges while the other two methods converges. The interval rational function $[r](t)$ is defined by

$$\begin{aligned}
[a_0, b_0] &= [-1/8, 1/8], & \omega_0 &= 1,\\
[a_1, b_1] &= [7/8, 1], & \omega_1 &= 3,\\
[a_2, b_2] &= [7/4, 2], & \omega_2 &= 8,\\
[a_3, b_3] &= [7/8, 1], & \omega_3 &= 2,\\
[a_4, b_4] &= [-1/8, 1/8], & \omega_4 &= 1
\end{aligned}$$

The degree 4, 5 and 6 interval polynomial bounds by the three methods are shown in Figure 11-Figure 19. The corresponding bounding errors and computational time are list in Table 2.

Table 2. Comparison of the bounding errors and the computational costs (II)

m	H M		P M		L M	
	errors	time	errors	time	errors	time
$m = 4$	4.32	0.02	0.68	0.10	1.18	0.24
$m = 5$	12.99	0.02	0.54	0.14	0.97	0.24
$m = 6$	5.54	0.02	0.46	0.16	0.22	0.28

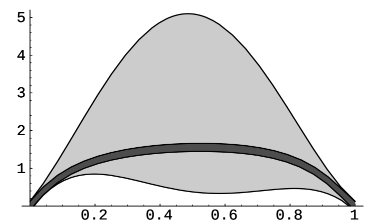

Figure 11. Hybrid Method $m = 4$

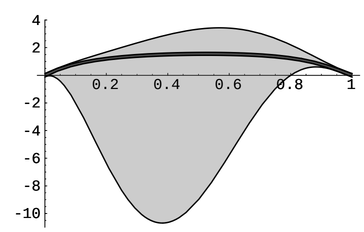

Figure 12. Hybrid Method $m = 5$

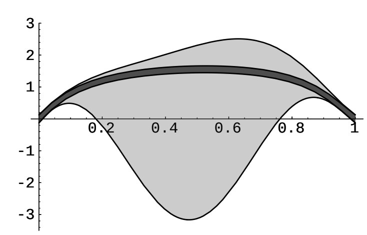

Figure 13. Hybrid Method $m = 6$

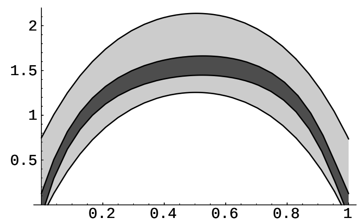

Figure 14. Perturbation Method $m = 4$

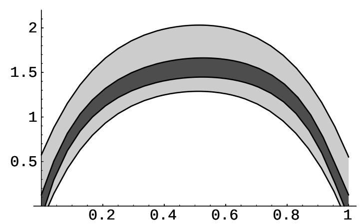

Figure 15. Perturbation Method $m = 5$

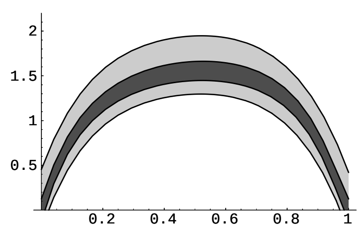

Figure 16. Perturbation Method $m = 6$

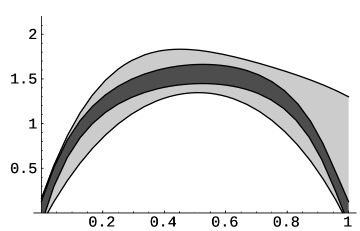

Figure 17. Linear Programming Method $m = 4$

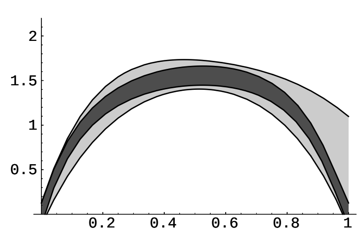

Figure 18. Linear Programming Method $m = 5$

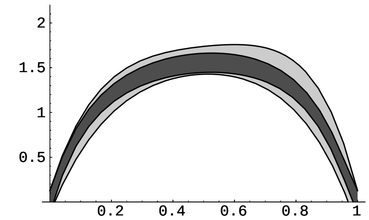

Figure 19. Linear Programming Method $m = 6$

Example 3 In the last example, we use piecewise interval polynomial curves to bound an interval rational curve. Let the control points and weights of the interval rational curve be:

$$
\begin{aligned}
[\mathbf{R}_0] &= ([50, 70], [340, 355]), \quad & \omega_0 = 1, \\
[\mathbf{R}_1] &= ([190, 220], [100, 120]), \quad & \omega_1 = 2, \\
[\mathbf{R}_2] &= ([340, 360], [330, 340]), \quad & \omega_2 = 2, \\
[\mathbf{R}_3] &= ([430, 455], [120, 130]), \quad & \omega_3 = 1
\end{aligned}
$$

We recursively subdivide the interval rational curve at the parameter value $t = \frac{1}{2}$ and then bound each segment using an interval polynomial curve of degree $m = 4$ and 5 respectively. Table 3 and 4 list the corresponding bounding errors and the computational time for the three methods.

Table 3. Comparison of the bounding errors and computational costs (III) ($m = 4$**)**

number of	H M		P M		L M	
subdivisions	errors	time	errors	time	errors	time
0	423.75	0.03	74.68	0.22	123.3	0.44
1	31.82	0.05	11.06	0.33	6.89	0.71
2	3.24	0.11	1.28	0.71	0.73	1.38
3	0.23	0.27	0.093	1.32	0.051	2.69
4	0.012	0.44	0.0051	2.69	0.0026	5.44
5	0.0012	0.88	0.00049	5.38	0.00026	10.93

From the above examples, we can draw a conclusion that the bounding errors by the Perturbation Method and the Linear Programming Method are much smaller than those by the Hybrid Method, even though they are computationally several times more expensive. Consequently, if we use a piecewise interval polynomial curve to bound an interval rational curve such that the bounding error is less than some given tolerance, the Hybrid Method will generally need more segments than the other two methods. Furthermore, examples show that the Hybrid Method has a much more restrict convergence condition than the other two methods do.

Table 4. Comparison of the bounding errors and computational costs (III) ($m = 5$)

number of subdivisions	H M		P M		L M	
	errors	time	errors	time	errors	time
0	386.73	0.03	36.77	0.22	15.83	0.38
1	29.74	0.05	5.41	0.44	3.19	0.77
2	1.53	0.11	0.40	0.88	0.25	1.59
3	0.055	0.27	0.017	1.76	0.011	3.13
4	0.0018	0.49	0.00059	3.46	0.00042	6.32
5	.00012	1.05	.000041	6.97	.000029	12.58

5. Conclusions

In this paper, we put forward a new problem in CAD and interval analysis communities–bounding an interval rational function (or Bézier curve) with an interval polynomial function (or Bézier curve). We proposed three different methods to solve the problem, and comparisons of bounding errors and computational costs are made between these three methods. The experimental results show that the Perturbation Method and the Linear Programming Method produce much tighter bounds than the Hybrid Method does, although they are computationally several times more expensive.

Further research problems include: (1) for the Perturbation Method and the Linear Programming Method, explore whether the interval polynomial bounds converge to the original interval rational curve when m goes to infinity; and (2) generalizing the results in this paper to surface cases.

Acknowledgement

The first and third authors are supported by the National Natural Science Foundation of China (grants 69603009 and 19971087) and the 973 Project on Mathematical Mechanics (grant G1998030600). The authors are sincerely thankful for the financial support of K.C. Wong Foundation.

References

[1] R.E. Moore, *Interval Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1966.

[2] T.W. Sederberg, and R.T. Farouki, ”Approximation by interval Bézier curves”, *IEEE Comput. Graph. Appl.*, 1992, 15(2), 87-95.

[3] C.Y. Hu, T. Maekawa, E.C. Sherbrooke, and N.M. Patrikalakis, ”Robust interval algorithm for curve intersections”, *Computer Aided Design*, 1996, 28(6/7), 495-506.

[4] C.Y. Hu, N.M. Patrikalakis, and X. Ye, ”Robust interval solid modeling: Part I: representations”, *Computer Aided Design*, 1996, 28(10), 807-817.

[5] C.Y. Hu, N.M. Patrikalakis, and X. Ye, ”Robust interval solid modeling: Part II: boundary evaluation”, *Computer Aided Design*, 1996, 28(10), 819-830.

[6] C.Y. Hu, T. Maekawa, N.M. Patrikalakis, and X. Ye, ”Robust interval algorithm for surface intersections”, *Computer Aided Design*, 1997, 29(9), 617-627.

[7] S.T. Tuohy, T. Maekawa, G. Shen, and N.M. Patrikalakis, ”Approximation of measured data with interval B-splines”, *Computer Aided Design*, 1997, 29(11), 791-799.

[8] F. Chen, and W. Lou, ”Degree Reduction of Interval Bézier curves”, accepted by *Computer Aided Design*.

[9] T.W. Sederberg, and M. Kakimoto, ”Approximating rational curves using polynomial curves”, *NURBS for Curve and Surface Design*, G. Farin, eds., SIAM, Philadelphia, 1991, 149-158.

[10] G. Wang, T.W. Sederberg, and F. Chen, ”On the convergence of polynomial approximation of rational functions”, *Journal of Approximation Theory*, 1997, 89, 267-288.