A Fast Algorithm for Rational Function Approximations

Michael Rosenblum

November 30, 1999

Abstract

I present the rational function approximation problem, and show how to modify a classic polynomial-GCD algorithm to solve it quickly. First, I discuss the problem and its motivation. Next, I prove that an optimal solution to the continued fraction approximation problem is also an optimal solution to the rational function approximation problem (relying on notes from Madhu Sudan). My contribution to the problem is the second half of the paper: I show how to use a polynomial-GCD-finding algorithm due to Aho, Hopcroft, and Ullman [1] and a variation due to Don Knuth, to solve the former problem. Last, I prove the correctness of this algorithm and bound its running time.

Introduction

My motivation for studying the rational function approximation problem, posed to me by Madhu Sudan, is to gain insight into a generalization of this problem which has applications in coding theory. In particular, the generalization to be discussed briefly at the end of this paper could be used to speed up decoding of Reed-Solomon error-correcting codes.

The rational function approximation problem asks, given a rational function in one variable, $\alpha(x) = \frac{g(x)}{h(x)}$, and an integer w, find another rational function that is "close" to $\alpha(x)$ but whose denominator is a polynomial of degree $< w$. I attack this problem by reducing it to the continued fraction approximation problem (defined below), and then giving an algorithm to solve this latter problem.

The algorithm I present to find the continued fraction approximation (defined below) of a given rational function uses ideas from Euclid's algorithm to find polynomial GCDs. I will show that one can construct the continued fraction approximation to a rational function $\alpha(x) = g(x)/h(x)$ based on the sequence of quotients produced by running Euclid's algorithm on $h(x), g(x)$. The naive Euclidean algorithm takes time $O(n^2 logn)$. The improved algorithm I present, following closely techniques from [1], improves on the naive Euclidean algorithm using a divide-and-conquer strategy to achieve running time $O(nlog^2 n)$.

The faster algorithm capitalizes on the observation that "many" high-order terms in the quotient and remainder of $h(x)$ divided by $g(x)$ do not depend on the low order terms of $h(x)$ and $g(x)$. Thus, we can temporarily ignore some of the low order terms when doing synthetic divisions in Euclid's algorithm, thereby reducing the size of polynomials we divide (and so reducing the running time of the algorithm).

Though the proofs I present require digesting a bit of notation and definitions, all the ideas involved are elementary and require only a knowledge of how the Euclidean algorithm for polynomial GCDs works.

Part I: Reduction of Rational Function Approximation Problem to Continued Fraction Approximation Problem

I owe much of this section to Madhu Sudan's notes. My contribution includes filling in the proofs for propositions 1 and 2.

First, some notation and definitions:

1. Throughout this paper, $a_i, b_i, d, e, f, g_i, h_i, p_i, q_i, t, u$ will represent polynomial functions in one variable x, over a field k.
2. α will represent a rational function over the same field.
3. The degree of a rational function $\alpha(x) = g(x)/h(x)$ is defined to be $deg[g(x)] - deg[h(x)]$.
4. Given a rational function $\alpha(x)$, define its continued fraction representation $CF(\alpha(x)) = (q_1(x), \ldots, q_l(x))$ where the $q_i(x)$ form the unique sequence of polynomials (which happen to be those generated by Euclid's algorithm run on h, g) such that
$$\alpha(x) = \cfrac{1}{q_1(x) + \cfrac{1}{q_2(x) + \cfrac{1}{q_3(x) + \cfrac{1}{\ldots + \cfrac{1}{q_l(x)}}}}}.$$

Now I can formally state the rational function approximation problem:

Definition 1: rational function approximation problem
Given integer w and rational function $\alpha(x) = g(x)/h(x)$
Minimize $deg[\alpha(x) - p_0(x)/p_1(x)] + deg[p_1(x)]$ s.t. $deg[p_1(x)] \leq w$

By proposition 1 below, this is equivalent to minimizing $deg[g(x)p_1(x) - h(x)p_0(x)]$.

I show in this section that any optimal solution to the following problem is also an optimal solution to the corresponding rational function approximation problem.

Definition 2: continued fraction approximation problem
Given integer w and rational function $\alpha(x) = \frac{g(x)}{h(x)}$, find the rational function $a(x)/b(x) = CF^{-1}(q_1(x), \ldots, q_k(x))$, where $(q_1(x), \ldots, q_l(x)) = CF(\alpha(x))$, and k the largest index such that $CF^{-1}(q_1(x), \ldots, q_k(x))$ has degree of denominator at most w.

I first prove some elementary propositions to be used in this section.

Proposition 1: For $\alpha_0(x), \alpha_1(x)$ rational functions,
i. $deg[\alpha_0(x)\alpha_1(x)] = deg[\alpha_0(x)] + deg[\alpha_1(x)]$
ii. if $deg[\alpha_0(x)] \neq deg[\alpha_1(x)]$, then $deg[\alpha_0(x) \pm \alpha_1(x)] = max(deg[\alpha_0(x)], deg[\alpha_1(x)])$

This is easily verified by the rules of multiplication and addition of rational functions.

For a sequence $q_1, \ldots, q_l$, with $\forall i, deg[q_i] > 0$ define inductively the sequences g_i and h_i:
$g_0 = 0$; $h_0 = 1$

$g_1 = 1$; $h_1 = q_1$
$\forall i > 1, g_i = g_{i-2} + q_i g_{i-1}; h_i = h_{i-2} + q_i h_{i-1}$

Proposition 2
i. The degrees of the h_i are strictly monotone increasing.
ii. $\forall i > 0, h_i g_{i-1} - g_i h_{i-1} = \pm 1$
iii. $\forall i \geq 1, g_i/h_i = CF^{-1}(q_1, \ldots, q_i)$

Proof:
Part i: Since we assumed $deg[q_i] > 0$, we can apply the above proposition regarding degrees of sums and products of polynomials. Note that since we assume $deg[q_i] > 0$, this immediately implies $deg[h_i] = \sum_{j=1}^{i} deg[q_j]$
Part ii: This requires a straightforward induction on i, with inductive step using the identity $h_i g_{i-1} - g_i h_{i-1} = h_i(g_{i-1} + q_{i+1} g_i) - g_i(h_{i-1} + q_{i+1} h_i)$ and substituting $g_{i+1} = g_{i-1} + q_{i+1} g_i$ and $h_{i+1} = h_{i-1} + q_{i+1} h_i$ to prove the property for $i+1$. In particular this shows that g_i is relatively prime to h_i.
Part iii: Use induction on the length of the sequence $(q_1, \ldots, q_i)$. If i =1 or 2, the property holds trivially. Assume the property holds for all quotient sequences of length $< i$, where $i > 2$. Now, take quotient sequence $(q_1, \ldots, q_i)$. Then $CF^{-1}(q_1, \ldots, q_i) = \frac{1}{q_1 + CF^{-1}(q_2, \ldots, q_i)}$, so we apply the inductive hypothesis to the sequence $(q_2, \ldots, q_i)$, with corresponding g'_i, h'_i, to get $CF^{-1}(q_2, \ldots, q_i) = g'_i/h'_i$, and similarly to get $CF^{-1}(q_2, \ldots, q_{i-1}) = g'_{i-1}/h'_{i-1}$,$CF^{-1}(q_2, \ldots, q_{i-2}) = g'_{i-2}/h'_{i-2}$
We would like to show $\frac{1}{q_1 + g'_i/h'_i} = \frac{g_i}{h_i}$, in order to finish the inductive step.
Applying inductive hypothesis twice, we get:

$$g_{i-1}/h_{i-1} = CF^{-1}(q_1, \ldots, q_{i-1}) = \frac{1}{q_1 + CF^{-1}(q_2, \ldots, q_{i-1})} = \frac{1}{q_1 + g'_{i-1}/h'_{i-1}} = \frac{h'_{i-1}}{q_1 h'_{i-1} + g'_{i-1}}$$
$$g_{i-2}/h_{i-2} = CF^{-1}(q_1, \ldots, q_{i-2}) = \frac{1}{q_1 + CF^{-1}(q_2, \ldots, q_{i-2})} = \frac{1}{q_1 + g'_{i-2}/h'_{i-2}} = \frac{h'_{i-2}}{q_1 h'_{i-2} + g'_{i-2}}$$

Using part ii. we can conclude:
$g_{i-1} = h'_{i-1}$ and $h_{i-1} = (q_1 h'_{i-1} + g'_{i-1})$
$g_{i-2} = h'_{i-2}$ and $h_{i-2} = (q_1 h'_{i-2} + g'_{i-2})$

Now our work pays off:
$h'_i = h'_{i-2} + q_i h'_{i-1} = g_{i-2} + q_i g_{i-1} = g_i$
and

$$\begin{aligned} q_1 h'_i + g'_i &= q_1 h'_i + (g'_{i-2} + q_i g'_{i-1}) \\ &= q_1 h'_i + (h_{i-2} - q_1 h'_{i-2}) + q_i(h_{i-1} - q_1 h'_{i-1}) \\ &= h_{i-2} + q_i h_{i-1} + q_1(h'_i - h'_{i-2} - q_i h'_{i-1}) \\ &= h_{i-2} + q_i h_{i-1} \\ &= h_i \end{aligned}$$

Thus, $\frac{1}{q_1 + g'_i/h'_i} = \frac{h'_i}{q_1 h'_i + g'_i} = \frac{g_i}{h_i}$, finishing the inductive step. ■

From iii. we immediately get the following crucial ingredient in showing g_i/h_i is a good approximation to α: Let $\alpha_i = CF^{-1}(q_{i+1}, \ldots, q_l)$ (so that $\alpha_{i+1} = \frac{1}{\alpha_i} - q_{i+1}$ and $\forall i \ deg[\alpha_i] < 0$). Then

$$\alpha = CF^{-1}(q_1, \ldots, q_{i-1}, q_i + \alpha_i) = \frac{g_{i-2} + (q_i + \alpha_i) g_{i-1}}{h_{i-2} + (q_i + \alpha_i) h_{i-1}}$$

Theorem 1 ***(reduction of rational function approximation problem to continued fraction approximation problem)*** *Given integer w, let k be the unique integer such that $deg[h_k] \leq w < deg[h_{k+1}]$. Then g_k/h_k is an optimum solution to the rational function approximation problem for $\alpha = CF^{-1}(q_1, \ldots, q_l)$*

Proof:

We first bound the degree of $\alpha - \frac{g_k}{h_k}$:

$$\begin{aligned}
\alpha - \frac{g_k}{h_k} &= \frac{g_{k-2} + (q_k + \alpha_k) g_{k-1}}{h_{k-2} + (q_k + \alpha_k) h_{k-1}} - \frac{g_{k-2} + q_k g_{k-1}}{h_{k-2} + q_k h_{k-1}} \\
&= \frac{\alpha_k (g_{k-1} h_{k-2} - h_{k-1} g_{k-2})}{(h_{k-2} + (q_k + \alpha_k) h_{k-1})(h_{k-2} + q_k h_{k-1})} \\
&= \frac{\pm \alpha_k}{(h_k + \alpha_k h_{k-1})(h_k)} \\
&= \frac{\pm 1}{(\alpha^{-1} h_k + h_{k-1})(h_k)} \\
&= \frac{\pm 1}{((q_{k+1} + \alpha_{k+1}) h_k + h_{k-1})(h_k)} \\
&= \frac{\pm 1}{(h_{k+1} + \alpha_{k+1} h_k)(h_k)}
\end{aligned}$$

Thus $deg[\alpha - \frac{g_k}{h_k}]$ is at most $-(deg[h_k] + deg[h_{k+1}])$.

We can now prove that g_k/h_k is the best rational function approximation to α with degree of denominator between $deg[h_k]$ and $deg[h_{k+1}] - 1$ inclusive. Let a/b be a candidate for approximating α, so it must have $deg[b] \leq w$. We need to check that $deg[\alpha - a/b] + deg[b] \geq -deg[h_{k+1}]$ to show that a/b is no better an approximation than g_k/h_k. Define

$$\delta = \alpha - \frac{a}{b} = \frac{g_k}{h_k} \pm \frac{1}{(h_{k+1} + \alpha_{k+1} h_k)(h_k)} - \frac{a}{b} = \frac{(h_{k+1} + \alpha_{k+1} h_k)[b g_k - a h_k] \pm b}{(h_{k+1} + \alpha_{k+1} h_k)(h_k) b}$$

We have two cases.
Case 1: If $bg_k - ah_k \neq 0$, then δ has degree at least $-(deg[h_k] + deg[b])$ since $deg[\alpha_{k+1}] < 0$. Further, $deg[\delta] + deg[b] \geq -deg[h_k] > -deg[h_{k+1}]$, so a/b is no better an approximant than g_k/h_k.
Case 2: If $bg_k - ah_k = 0$, then $\delta = \alpha - \frac{g_k}{h_k}$, and since g_k, h_k relatively prime by proposition 2, $deg[b] \geq deg[h_k]$, so again a/b is no better an approximant than g_k/h_k. ■

Now that I have reduced the rational function approximation problem to the continued fraction approximation problem, I'll give an algorithm to efficiently solve the latter.

Part II: A Fast Algorithm to Solve the Continued Fraction Approximation Problem

The algorithm I present to solve the continued fraction approximation problem uses techniques from the Euclidean algorithm to generate the sequence $(g_0/h_0, g_1/h_1, \ldots)$ starting from a rational function $\alpha(x) = g(x)/h(x)$. As mentioned in the introduction, this could naively be accomplished by running Euclid's algorithm on h, g to produce the quotient sequence $(q_1, \ldots, q_l)$, and then computing $CF^{-1}(q_1), CF^{-1}(q_1, q_2), \ldots$ until a rational function $CF^{-1}(q_1, \ldots, q_k)$ with degree of denominator greater than w was found; at this point we would output the rational function in our list appearing just before this function: $CF^{-1}(q_1, \ldots, q_{k-1})$. This would take total time $O(n^2 logn)$ for $O(n)$ polynomial divisions/multiplications, each taking time $O(nlogn)$.

I present a divide-and-conquer strategy to solve the problem in $O(nlog^2 n)$ time. The idea is that we can compute parts of the quotient and remainder without using low order terms of the divisor and dividend (see lemma 1 below for a formal statement). In particular, if we had an algorithm that could compute, starting with h, g, all remainders and quotients that would result from Euclid's algorithm up to the first remainder with degree less than $\frac{1}{2}deg[h]$, we could use this recursively (with the help of proposition 3 proved below) to compute the quotient and remainder sequences to $\frac{3}{4}deg[h]$, $\frac{7}{8}deg[h]$, ...

We will implicitly be keeping track of the degrees of the h_i generated–which can be easily computed by keeping track of the degrees of the q_i generated and using $deg[h_i] = \sum_{j=1}^{i} deg[q_j]$–so that we stop as soon as we reach an h_i with degree greater than w.

How does this mysterious algorithm that computes "half" the quotient and remainder sequence work? This algorithm, *Half-CF*, uses divide-and-conquer itself to recursively compute 1/4 of the remainder sequence, then 1/3 of the remaining remainder sequence (which is only 3/4 the original remainder sequence, so that gets us 1/2 way down the original remainder sequence). The time-saving trick is that when *half-CF* calls itself, it does so on inputs of degree no more than half the degree of its input, allowing us to get a nice time-bound on its performance. Of course, we have to prove that *Half-CF* performs correctly, which takes up a large part of this section.

To understand the algorithm, I first introduce some notation.

Let a_0 and a_1 be polynomials s.t. $deg[a_0] > deg[a_1]$ and with remainder sequence $a_0, a_1, ..., a_l$ under the Euclidean algorithm, and corresponding quotient sequence $q_1, q_2, ..., q_l$ (so that in particular, $a_1/a_0 = CF^{-1}(q_1, q_2, ..., q_l)$).

1. Define $l(k)$ = the unique integer such that $deg[a_{l(k)}] > k$ and $deg[a_{l(k)+1}] \leq k$.
2. Define recursively: for $0 \leq i$, $R^{a_0,a_1}_{i,i} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

   and for $i < j$, $R^{a_0,a_1}_{i,j} = \begin{pmatrix} 0 & 1 \\ 1 & -q_j \end{pmatrix} * \begin{pmatrix} 0 & 1 \\ 1 & -q_{j-1} \end{pmatrix} * \ldots * \begin{pmatrix} 0 & 1 \\ 1 & -q_{i+1} \end{pmatrix}$

I present the following propositions which show the usefulness of the above definitions (the proofs are simple verifications based on the definitions):

Proposition 3:
i. We can evaluate $R^{f,g}_{0,l(k)}$ by using the Euclidean algorithm to generate a quotient sequence and

remainder sequence, stopping when we find a remainder of degree less than or equal to k. More precisely, if $deg[g] > k$, we have $R^{f,g}_{0,l(k)} = R^{g,r}_{0,l(k)} * \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix}$ where r is the remainder and q is the quotient of f divided by g; that is $f = q_1 g + r$ where $deg[r] < deg[g]$.
ii. We can use the matrices $R^{h,g}_{i,j}$ to compute the continued fraction generated by $\alpha(x) = g(x)/h(x)$ as follows: Recall the inductive definition of the sequences g_i and h_i from section 1:
$g_0 = 0$; $h_0 = 1$
$g_1 = 1$; $h_1 = q_1$
$\forall i > 1, g_i = g_{i-2} + q_i g_{i-1}; h_i = h_{i-2} + q_i h_{i-1}$
We can evaluate g_i, h_i by computing $R^{h,g}_{0,i+1}$, and then extracting the upper row:
$\forall i \geq 0$, $R_{0,i+1} = \pm \begin{pmatrix} -g_i & h_i \\ g_{i+1} & -h_{i+1} \end{pmatrix}$
iii. The matrices $R^{h,g}_{i,j}$ can also be used to compute terms in the remainder sequence generated by h,g. In particular, for $i < j < k$,

$$\begin{pmatrix} a_j \\ a_{j+1} \end{pmatrix} = R_{i,j} \begin{pmatrix} a_i \\ a_{i+1} \end{pmatrix}$$

Algorithm

The following algorithm computes the continued fraction approximation of $\alpha(x) = g(x)/h(x)$ and degree bound w, where $deg[g(x)] < deg[h(x)]$. The output is a matrix $\begin{pmatrix} a(x) & b(x) \\ c(x) & d(x) \end{pmatrix}$ where $a(x)/b(x) = g_k(x)/h_k(x)$ is an optimal continued fraction approximation.
It makes calls to a function to be defined below called *Half-CF* (half-continued-fraction) which behaves as follows: *Half-CF* takes as input two polynomials h and g and integer w, with $deg[h] > deg[g]$. Let $(q_1(x), \ldots, q_l(x)) = CF(\alpha(x))$, and k be the largest index such that $CF^{-1}(q_1(x), \ldots, q_k(x))$ has degree of denominator at most w. Then *Half-CF* returns $R^{h,g}_{0,l(deg[h]/2)}$ (unless $k < l(deg[h]/2)$, in which case it stops at the first remainder corresponding to r_k in the original sequence).

Continued-Fraction$(\alpha(x) = g(x)/h(x),\ w)$

1. if $deg[h(x)] < w$ then return $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
2. else let $F = \begin{pmatrix} a(x) & b(x) \\ c(x) & d(x) \end{pmatrix} \leftarrow$ *Half-CF*(h, g, w)
3. $\begin{pmatrix} t \\ u \end{pmatrix} \leftarrow F \begin{pmatrix} h \\ g \end{pmatrix}$;
4. $r \leftarrow t \bmod u$; that is, r is the remainder of t divided by u
5. $q \leftarrow \lfloor \frac{t}{u} \rfloor$; that is, q is the quotient of t divided by u
6. let $G = \begin{pmatrix} a'(x) & b'(x) \\ c'(x) & d'(x) \end{pmatrix} \leftarrow \begin{pmatrix} 0 & 1 \\ 1 & -q(x) \end{pmatrix} * F$
7. return *Continued-Fraction*$(\alpha' = r(x)/u(x),\ w) * G$

On the assumption that *Half-CF* acts as claimed, we can prove that *Continued-Fraction* is correct:

Theorem 2 *Correctness of Continued Fraction: Continued-Fraction($\alpha(x) = g(x)/h(x), w$) returns* $\begin{pmatrix} a(x) & b(x) \\ c(x) & d(x) \end{pmatrix}$ *where $a(x)/b(x) = g_k(x)/h_k(x)$ is an optimal continued fraction approximation. That is, $a(x)/b(x) = CF^{-1}(q_1(x), \ldots, q_k(x))$, where $(q_1(x), \ldots, q_l(x)) = CF(\alpha(x))$, and k the largest index such that $CF^{-1}(q_1(x), \ldots, q_k(x))$ has degree of denominator at most w.*

Proof: This is an induction on $deg[f] = n$. If $k < \lfloor n/2 \rfloor$, *Half-CF* returns the desired $a(x)/b(x)$ by hypothesis; otherwise, at lines 4 and 5 one division is done to produce the next q in the quotient sequence, by proposition 3.ii. At line 6 is computed the next pair of continued fraction approximants. *Continued-Fraction* calls itself on lower degree input, and by inductive hypothesis returns the matrix required to complete the continued fraction approximation, using proposition 3.iii. ■

Now I present the algorithm *Half-CF*, where it is implicit that once a remainder is reached corresponding to remainder r_k in the original remainder sequence, it does not compute any more quotients. This can be accomplished by keeping track of the degrees of the h_i generated–which can be easily computed by keeping track of the degrees of the q_i generated and using $deg[h_i] = \sum_{j=1}^{i} deg[q_j]$–so that we stop as soon as we reach an h_i with degree greater than w.

Half-CF(f, g, w)

1. if $deg[g] \leq deg[f]/2$ then return $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
2. else let $m = \lfloor deg[f]/2 \rfloor$
3. $\quad f = f_1 x^m + f_0$; where $deg[f_0] < m$
4. $\quad g = g_1 x^m + g_0$; where $deg[g_0] < m$
5. $\quad D \leftarrow$ *Half-CF*(f_1, g_1, w)
6. $\quad \begin{pmatrix} a \\ b \end{pmatrix} \leftarrow D \begin{pmatrix} f \\ g \end{pmatrix}$
7. $\quad c \leftarrow a \bmod b$; that is, c is remainder of a divided by b
8. $\quad q \leftarrow \lfloor a/b \rfloor$; that is, q is the quotient
9. $\quad b = b_1 x^{\lfloor m/2 \rfloor} + b_0$; where $deg[b_0] < \lfloor m/2 \rfloor$
10. $\quad c = c_1 x^{\lfloor m/2 \rfloor} + c_0$; where $deg[c_0] < \lfloor m/2 \rfloor$
11. $\quad E \leftarrow$ *Half-CF*(b_1, c_1, w)
12. $\quad$ return $E * \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix} * D$

Lemma 1 *Let $f = f_1 x^k + f_0; g = g_1 x^k + g_0$ where $deg[f_0] < k$ and $deg[g_0] < k$. Let q and q_1 be the quotients $\lfloor f(x)/g(x) \rfloor$ and $\lfloor f_1(x)/g_1(x) \rfloor$, and let r and r_1 by the remainders $f(x) - q(x)g(x)$ and $f_1(x) - q_1(x)g_1(x)$, respectively. If $deg[f] > deg[g]$ and $deg[g] > \lceil \frac{deg[f]+k}{2} \rceil$ [i.e. $deg[g_1] \geq \frac{1}{2} deg[f_1]$], then,*

i. *$q = q_1$ and*

ii. *r and $r_1 x^k$ agree in all terms of degree $k + deg[f] - deg[g]$ and higher.*

Proof: The lemma is a consequence of the ordinary synthetic division algorithm for polynomials. In the quotient of $f(x)$ divided by $g(x)$, the first $deg[g] - k$ terms (which is all the terms since $deg[f] - deg[g] \leq deg[g] - k$ by hypothesis) of the quotient depend neither on $g_0(x)$ nor on $f_0(x)$. Thus, the quotients q and q_1 are identical. Similar reasoning shows that the remainder terms of degree $deg[f] - (deg[g] - k)$ or greater do not depend on $g_0(x)$; also the terms of degree k or greater do not depend on $f_0(x)$. But $deg[f] - deg[g] + k > k$ by hypothesis, so all we can assert is $r(x)$ and $r_1(x)x^k$ agree in all terms of degree $(deg[f] - deg[g] + k)$ and higher. ■

We are now ready to prove the main theorem, which ensures correctness of the algorithm *Half-CF*. I fully flesh out this proof, which in [1] consisted of a single sentence.

Theorem 3 *Let* n *denote* $deg[f(x)]$*. Then* $\forall f(x), g(x), k < n, s.t. deg[f(x)] > deg[g(x)]$*, and* $f = f_1 x^k + f_0; g = g_1 x^k + g_0$ *where* $deg[f_0] < k$ *and* $deg[g_0] < k$ *we have*

$$R^{f,g}_{0,l(\lceil \frac{n+k}{2} \rceil)} = R^{f_1,g_1}_{0,l(\lceil \frac{n-k}{2} \rceil)}$$

Proof: I will induct on n, the degree of the polynomial $f(x)$.

Base Case: $n = 1$
If $n = 1$, then the only non-negative value of k for which the theorem applies is $k = 0$ (since we require $k < n$). For $k = 0$ the theorem holds trivially.

Inductive Step: Assume that for all $f'(x)$ with $deg[f'(x)] < n$ the above theorem holds. Now, take any polynomial $f(x)$ with $deg[f(x)] = n$, $g(x)$ with $deg[g(x)] < deg[f(x)]$, $k < n$, and set $f = f_1 x^k + f_0; g = g_1 x^k + g_0$ where $deg[f_0] < k$ and $deg[g_0] < k$.
We have two cases:
Case 1: $deg[g(x)] \leq \lceil \frac{n+k}{2} \rceil$
In this case, since by construction $deg[g_1] = deg[g] - k$, we have
$deg[g_1(x)] \leq \lceil \frac{n+k}{2} \rceil - k = \lceil \frac{n-k}{2} \rceil$
so both
$R^{f,g}_{0,l(\lceil \frac{n+k}{2} \rceil)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$R^{f_1,g_1}_{0,l(\lceil \frac{n-k}{2} \rceil)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
and so the theorem holds.

Case 2: $deg[g(x)] > \lceil \frac{n+k}{2} \rceil$
In this case, again since by construction $deg[g_1] = deg[g] - k$, we have
$deg[g_1(x)] > \lceil \frac{n+k}{2} \rceil - k = \lceil \frac{n-k}{2} \rceil$.

We now apply Lemma 1 above, and conclude that
1) the quotients $\lfloor f/g \rfloor$ and $\lfloor f_1/g_1 \rfloor$ are equal, and
2) for the remainders $r = f/g - \lfloor f/g \rfloor$ and $r_1 = f_1/g_1 - \lfloor f_1/g_1 \rfloor$, we have r and $r_1 x^k$ agree at all terms of order $k + n - deg[g]$ and higher.

and so we can write
$g = g_1 x^k + g_0; r = r_1 x^k + r_0$ where $deg[g_0] < k$ and $deg[r_0] < k + n - deg[g]$

Now, by proposition 3.i. above,

$$R^{f,g}_{0,l(\lceil \frac{n+k}{2} \rceil)} = R^{g,r}_{0,l(\lceil \frac{n+k}{2} \rceil)} * \begin{pmatrix} 0 & 1 \\ 1 & -\lfloor f/g \rfloor \end{pmatrix}$$

$$R^{f_1,g_1}_{0,l(\lceil \frac{n-k}{2} \rceil)} = R^{g_1,r_1}_{0,l(\lceil \frac{n-k}{2} \rceil)} * \begin{pmatrix} 0 & 1 \\ 1 & -\lfloor f_1/g_1 \rfloor \end{pmatrix}$$

So in particular, if we had $R^{g,r}_{0,l(\lceil \frac{n+k}{2} \rceil)} = R^{g_1,r_1}_{0,l(\lceil \frac{n-k}{2} \rceil)}$, we would be done.

Unfortunately, since $n > deg[g]$ we cannot immediately apply the inductive hypothesis to the pair of polynomials g, r and constant k. The problem is that we only know that r and r_1 agree at terms of degree $k + n - deg[g]$ and higher. But this is enough information for us to apply the inductive hypothesis to g, r and larger constant $k' = k+n-deg[g]$ (where $k' < deg[g]$ because $deg[g] > \lceil \frac{n+k}{2} \rceil$) giving:

$R^{g,r}_{0,l(\lceil \frac{n+k}{2} \rceil)} = R^{g_2,r_2}_{0,l(\lceil \frac{2degg-k-n}{2} \rceil)}$
where
$g_1 = g_2 x^{n-deg[g]} + a; r_1 = r_2 x^{n-deg[g]} + b$ where $deg[a] < n - deg[g]$ and $deg[b] < n - deg[g]$;
so that,
$g = g_2 x^{k'} + c; r = r_2 x^{k'} + d$; where $deg[c] < k'$ and $deg[d] < k'$.

We can apply the inductive hypothesis again to the polynomials g_1, r_1 and constant $k'' = n - deg[g]$ (where $k'' < deg[g_1]$ because $deg[g_1] = deg[g] - k$ and we have $deg[g] > \lceil \frac{n+k}{2} \rceil$) to get

$R^{g_1,r_1}_{0,l(\lceil \frac{n-k}{2} \rceil)} = R^{g_2,r_2}_{0,l(\lceil \frac{2degg-k-n}{2} \rceil)}$

Thus, putting the two together we get

$R^{g,r}_{0,l(\lceil \frac{n+k}{2} \rceil)} = R^{g_1,r_1}_{0,l(\lceil \frac{n-k}{2} \rceil)}$

as desired, and we are done. ■

With this key theorem, we can now prove that *Half-CF* works as claimed.

Theorem 4 *Correctness of Half-CF: Half-CF(f, g, w), with $deg[f] > deg[g]$, returns $R^{f,g}_{0,l(n/2)}$, where $n = deg[f]$.*
Proof: The proof is an induction on n, using Theorem 2 at the inductive step to show $D = R^{f,g}_{0,l\lceil \frac{3m}{2} \rceil}$ and $E = R^{f,g}_{l\lceil \frac{3m}{2} \rceil+1,l(m)}$, where $m = \lfloor \frac{n}{2} \rfloor$. More precisely, we have by inductive hypothesis that
$D = R^{f_1,g_1}_{0,l(\lceil \frac{n-m}{2} \rceil)}$. Theorem 2 gives,
$= R^{f,g}_{0,l\lceil \frac{n+m}{2} \rceil}$
$= R^{f,g}_{0,l\lceil \frac{3m}{2} \rceil}$

For the second part, again by inductive hypothesis
$E = R^{b_1,c_1}_{0,l(\lceil \frac{deg[b_1]}{2} \rceil)}$
$= R^{b_1,c_1}_{0,l(\lceil \frac{deg[b]-\lfloor m/2 \rfloor}{2} \rceil)}$ We can apply Theorem 2 since b_1, c_1 of degree $m+1$ at most.
$= R^{b,c}_{0,l(\lceil \frac{deg[b]+\lfloor m/2 \rfloor}{2} \rceil)}$
$= R^{f,g}_{l\lceil \frac{3m}{2} \rceil+1,l(m)}$
The last line follows since, according to propositions proved at the beginning of this section, b is the first term in the remainder sequence of f, g with $deg[b] < \frac{3m}{2}$. ■

Having proved the correctness of *Half-CF*, and by theorem 2 the correctness of *Continued-Fraction*, all that remains is to show the time bound $O(nlog^2 n)$ for the latter.

Theorem 5 *Continued-Fraction runs in time $O(nlog^2 n)$ on inputs of degree at most n*

Proof: I first show that *Half-CF* runs in time $O(nlog^2n)$ on input of degree at most n. *Half-CF* makes two recursive calls, both on polynomials of degree at most $\lceil n/2 \rceil$, and also does one division and three matrix multiplications on polynomials of degree at most n. Thus, the running time obeys the recurrence (where we allow $O(nlogn)$ time for a multiplication/division):
$T(n) = 2T(n/2) + O(nlogn)$, which solves as $T(n) = O(nlog^2n)$.
Now I can bound the time for *Continued-Fraction*:
It calls *Half-CF* once on polynomials of degree at most n, does one division and three matrix multiplications on polynomials of degree also at most n, and recursively calls itself on polynomials of degree at most $n/2$. Thus, the running time obeys the recurrence:
$T(n) = T(n/2) + O(nlogn) + O(nlog^2n)$, which solves as $T(n) = O(nlog^2n)$.■

What Next? The motivation for studying the rational function approximation problem is that its natural generalization has applications in decoding Reed-Solomon codes. This generalization, proposed by Madhu Sudan, is the following:

> Given integers w_0, w_1 and polynomials in one variable $p_0(x), p_1(x), p_2(x)$
> Minimize $Deg[a_0(x)p_0(x) + a_1(x)p_1(x) + a_2(x)p_2(x)]$
> s.t. $Deg[a_0(x)] \leq w_0$, $Deg[a_1(x)] \leq w_1$ (with $Deg[a_2(x)]$ unconstrained),
> and $a_0(x)p_0(x) + a_1(x)p_1(x) + a_2(x)p_2(x) \neq 0$

I hope that the ideas presented in this paper will shed light on this problem, or at least give a direction in which to pursue its solution.

Acknowledgments

I would like to thank Professor Madhu Sudan for his guidance, both in proposing the problem and providing ideas/resources/help on how to solve it. I would like to thank Professor Don Knuth for his suggestions, and for giving an algorithm based on polynomial-GCD finding whose ideas I used in creating the algorithm in this paper.

References

[1] Aho, Hopcroft, and Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley Series in Computer Science and Information Processing. pages 302-308, 1974