The Continued Fraction Approach to Real Root Isolation

A drawback with the Descartes method and the Sturm method is that the subdivision is blind to the distribution of the roots in the interval, i.e., subdividing into half is not a good option if the roots are, say clustered towards one endpoint of the interval. To improve the subdivision strategy, we want to estimate where the roots are in the interval of interest. In this lecture, we will focus on isolating the positive roots of a polynomial; using Jacobi's little observation, however, we can search on any input interval. Let $A(x) = \sum_{i=0}^{n} a_i x^i$, $a_i \in \mathbb{R}$, be an arbitrary polynomial; $A_{\text{in}}(x)$ will denote the input polynomial to various algorithms.

One approach to isolate the positive roots is to consider a unit interval at a time, map it to $(0, \infty)$ and recursively isolate the roots in it. However, when we are outputting an interval in a recursive call we have to transform it back to an interval for the original polynomial. Thus, we not only pass a polynomial A at the recursive call, but also the transformation that relates A_{in} and A. More precisely, we have the following algorithm.

$\texttt{CF}(A, M)$
INPUT: A polynomial A, a transformation $M(x)$.
OUTPUT: Isolating intervals for the positive roots of A_{in}.
1. If $\text{Var}(A) = 1$ output $M(0), M(\infty)$.
2. $A_L(x) := (x+1)^n A(1/(1+x))$. $M_L := M(1/(x+1))$
3. $A_R(x) := A(x+1)$. $M_R(x) := M(1+x)$.
4. $\texttt{CF}(A_L, M_L)$, $\texttt{CF}(A_R, M_R)$.

The procedure is invoked as $\texttt{CF}(A_{\text{in}}, x)$, and it constructs a subdivision tree, where with each node in the tree we have associated a polynomial A and a transformation M. But what is the relation between A_L, M_L and A_{in}? To understand this relation, we have to understand the transformations $M_L(x)$ and $M_R(x)$. We claim that these transformations are Möbius transformations

$$M(x) = \frac{px+q}{rx+s}, \quad pq - rs \neq 0. \tag{1}$$

A more succinct way to represent the transformation is as a matrix

$$M(x) = \begin{bmatrix} p & q \\ r & s \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix},$$

where the 2×2 matrix on the RHS is non-singular. The interval I_M outputted at step 1 of the algorithm has endpoints q/s and p/r resp. Using this notation, it is easy to see the claim that both M_L and M_R are Möbius transformations as well

$$M_L(x) = \begin{bmatrix} p & q \\ r & s \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix}, \quad M_R(x) = \begin{bmatrix} p & q \\ r & s \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix},$$

which implies that both M_L and M_R are Möbius transformations; clearly, the initial transformation x is a möbius transformation. Moreover, the relation between A and A_{in} is the following

$$A(x) = (rx+s)^n A_{\text{in}}(M(x)), \tag{2}$$

where $M(x)$ is defined as in (1). Thus the transformation with the left-most path in the tree at depth i is

$$\cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{\ddots + \cfrac{1}{1+x}}}},$$

which in terms of matrice is the product

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \cdots .$$

It is well know that If we terminate this product after i steps we get the matrix

$$\begin{bmatrix} F_{i-1} & F_i \\ F_i & F_{i+1} \end{bmatrix}$$

or in other words, the interval assoicated with the left most branch of the subdivision tree has endpoints as F_{i-1}/F_i and F_i/F_{i+1}. In general, the transformation has the form

$$q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cfrac{1}{\ddots + \cfrac{1}{q_i + x}}}}},$$

which in the matrix notation can be expressed as

$$\begin{bmatrix} 1 & q_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & q_1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & q_2 \end{bmatrix} \cdots \begin{bmatrix} 0 & 1 \\ 1 & q_i \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix},$$

or more succintly $[q_0; q_1, q_2, \ldots, q_i + x]$. Let P_i/Q_i denote the finite continued fraction obtained by substituting $x = 0$ in the transformation, i.e., the continued fraction $[q_0; q_1, \ldots, q_i]$. The following recurrence gives us the relation between P_i/Q_i and P_{i+1}/Q_{i+1}:

$$P_i = P_{i-2} + q_i P_{i-1} \text{ and } Q_i = Q_{i-2} + q_i Q_{i-1}, \tag{3}$$

where $P_{-1} := 0$, $P_1 := q_0$, $Q_{-1} := 0$ and $Q_1 := 1$; too see this recurrence, observe that P_{i+1}/Q_{i+1} is obtained from P_i/Q_i by substitutin $x = 1/q_{i+1}$ in the transformation $[q_0; q_1, q_2, \ldots, q_i + x]$. Therefore, the Möbius transformation associated with the continued fraction $[q_0; q_1, q_2, \ldots, q_i]$ is $M(x) = (P_i + P_{i-1}x)/(Q_i + Q_{i-1}x)$; the endpoints of I_M are $M(0) = P_{i-1}/Q_{i-1}$ and $M(\infty) = P_i/Q_i$. Since $M(x)$ is unimodular, the width of I_M is $1/(Q_i Q_{i-1})$. Thus at each node in the subdivision tree we have a continued fraction approximation to the real roots of A_{in} in I_M.

Now we have an understanding of the polynomials and the transformations at each node in the subdivision tree. Going back to the algorithm, are we sure that it is polynomial time? As it turns out, it is not. Consider the polynomial $(x-(a+1))(x-a)$, $a > 1$, which has two sign variations. Clearly, we will need a shifts to reach the smallest positive root, but this is clearly exponential. This algorithm was originally proposed by Vincent in 1836. The fault is that we are taking one step at a time to compute the q_i's. We want to expedite this process, that is take larger steps to get to q_i. One approach is to compute a lower bound on the smallest positive root, which is what is done in practice. Here we describe a simpler approach, a binary search on the positive axis: shift by 2^0 and check if we have dropped any sign variations; if not then shift by 2^1 and agan check if we have dropped any sign variations; in general, shifty by 2^i and check if we have dropped any sign variations; eventually, we will reach an interval $[2^i, 2^{i+1}]$ s.t. shifting by 2^i does not drop any sign variation, but shifting by 2^{i+1} does; do a binary search within this interval to get a q, such that shifting by q does not drop the sign variation, but shifting by $q+1$ does; this q is the desired quotient in the continued fraction expansion. The number of steps needed to find q is $O(\log q)$. To avoid bit-complexity blow-up, we should always perform a Taylor shift on the polynomial $A_i(x) := A_{\text{in}}(M(x))$. Thus the modified algorithm is as follows.

$\texttt{CF}(A, M)$
INPUT: A polynomial A, a transformation $M(x)$.
OUTPUT: Isolating intervals for the positive roots of A_{in}.
1. If $\text{Var}(A) = 1$ output $M(0), M(\infty)$.
2. Compute the q as described above. $A(x) \leftarrow A(x+q)$. $M(x) \leftarrow M(q+x)$.
3. $A_L(x) := (x+1)^n A(1/(1+x))$. $M_L := M(1/(x+1))$
4. $A_R(x) := A_R(x+1)$, $M_R(x) := M_R(x+1)$.
4. $\texttt{CF}(A_L, M_L)$, $\texttt{CF}(A_R, M_R)$.

We now show that the modification proposed does yield a polynomial time algorithm, starting with a bound on the size of the subdivision tree. The termination criterion is the same as the Descartes method: Let $M(x)$ be the mobius transformation associated with a node; if $\overline{C}_{I_M} \cup \underline{C}_{I_M}$ contains at most one root of A_{in} then $\text{Var}(A_{\text{in}}, I_M) \leq 1$. In particular, this implies that there exists two roots $\alpha_{I_M}, \beta_{I_M}$ such that a constant times the width of I_M is greater than the distance between these two roots, i.e.,

$$2w(I_M) = \frac{2}{Q_i Q_{i-1}} \geq |\alpha_{I_M} - \beta_{I_M}|. \tag{4}$$