Real Stability Testing*

Prasad Raghavendra[1], Nick Ryder[2], and Nikhil Srivastava[3]

1 **University of California, Berkeley, USA**
raghavendra@berkeley.edu
2 **University of California, Berkeley, USA**
nick.ryder@berkeley.edu
3 **University of California, Berkeley, USA**
nikhil@math.berkeley.edu

Abstract

We give a strongly polynomial time algorithm which determines whether or not a bivariate polynomial is real stable. As a corollary, this implies an algorithm for testing whether a given linear transformation on univariate polynomials preserves real-rootedness. The proof exploits properties of hyperbolic polynomials to reduce real stability testing to testing nonnegativity of a finite number of polynomials on an interval.

1998 ACM Subject Classification F.2.1 Numerical Algorithms and Problems

Keywords and phrases real stable polynomials, hyperbolic polynomials, real rootedness, moment matrix, sturm sequence

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.5

1 Introduction

A univariate polynomial with real coefficients is called *real-rooted* if all of its roots are real. Multivariate generalizations of this concept, known as *hyperbolic* and *real stable* polynomials, were defined in the 50's and in the 80's in the context of Partial Differential Equations and Control Theory, respectively[1], and have since made contact with several areas of mathematics. In particular, a polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is called *real stable* if it has no zeros with all coordinates in the open upper half of the complex plane. These polynomials have played a central role in several recent advances in theoretical computer science and combinatorics — for instance, [1, 14, 13, 7, 5]. Each of these works relies in a critical way on (1) understanding which polynomials are real stable (2) understanding which linear operators *preserve* real-rootedness and real stability. Motivated by (1) and (2), this paper studies the following two fundamental algorithmic problems:

Problem 1. Given a bivariate polynomial[2] $p \in \mathbb{R}_n[x, y]$, is p real stable?

Problem 2. Given a linear operator $T : \mathbb{R}_n[x] \to \mathbb{R}_m[x]$, does T preserve real-rootedness?

The main result of this paper is a strongly polynomial time algorithm that solves Problem 1.

* This research was supported by NSF Grants CCF-1553751, NSF CCF-1343104 and NSF CCF-1407779.
[1] See [17] for a more detailed history.
[2] We use $\mathbb{R}_n[x_1, \ldots, x_k]$ to denote the vector space of real polynomials in $x_1, \ldots, x_k$ of degree at most n in each variable.

▶ **Theorem 1** (Main). *Given the coefficients of a bivariate polynomial $p \in \mathbb{R}_n[x, y]$, there is a deterministic algorithm which decides whether or not p is real stable in at most $O(n^5)$ arithmetic operations, assuming exact arithmetic.*

Part of the motivation for solving Problem 1 is the following theorem of Borcea and Branden, which shows that Problem 2 can be reduced to Problem 1.

▶ **Theorem 2** (Borcea-Branden [4]). *For every linear transformation $T : \mathbb{R}_n[x] \to \mathbb{R}_m[x]$, there is a bivariate polynomial $p \in \mathbb{R}_{\max(n,m)}[x, y]$ such that T preserves real-rootedness if and only if p is real stable. Moreover, the coefficients of p can be computed from the matrix of T in linear time.*

Thus, our main theorem immediately implies a solution to Problem 2 as well.

To give the reader a feel for the objects at hand, we remark that the set of real stable polynomials in any number of variables is a nonconvex set with nonempty interior [16]. In the univariate case, the interior of the set of real-rooted polynomials simply corresponds to polynomials with distinct roots, and its boundary contains polynomials which have roots with multiplicity greater than one. With regards to Problem 2, the prototypical example of an operator which preserves real rootedness is differentiation. Recent applications such as [15] rely on finding more elaborate differential operators with this property.

We now describe the main ideas in our algorithm. It turns out that testing bivariate real stability is equivalent to testing whether a certain *two parameter* family of polynomials is real rooted. It is not clear how to check this continuum of real-rootedness statements in strongly polynomial, or even in exponential time. To circumvent this, we use a deep convexity result from the theory of hyperbolic polynomials to reduce the two parameter family to a one parameter family of degree n polynomials, whose coefficients are themselves polynomials of degree n in the parameter. We then use a characterization of real-rootedness as postive semidefiniteness of certain moment matrices to further reduce this to checking that a finite number of univariate polynomials are *nonnegative* an interval. Finally, we solve each instance of the nonnegativity problem using Sturm sequences and a bit of algebra.

The set of polynomials nonnegative on an interval forms a closed convex cone, so the last step of our algorithm may be viewed as a strongly polynomial time membership oracle for this cone. We would not be surprised if such a result is already known (at least as folklore) but we were unable to find a concrete reference in the literature, so this component of our method may be of independent interest.

We see this result as being both mathematically fundamental, as well as useful for researchers who work with stable polyomials, particularly since many of their known applications so far (e.g.[14]) put special emphasis on properties of bivariate restrictions. More speculatively, it is possible that being able to test membership in the set of real stable polynomials is a step towards being able to optimize over them.

1.1 Related Work

Problem 1 was solved in the univariate case by C. Sturm in 1835 [19], who described a now well-known method that can be turned into a strongly polynomial quadratic time algorithm given the coefficients of p [2]. We are unaware of any published work regarding algorithms for the bivariate case or for Problem 2. We remark that following the release of this paper, Thorsten Theobald has observed (informal communication) that the quantifier elimination techniques of [3] can be used to obtain *weakly* polynomial time algorithms for Problems 1 and 2.

The paper [9] studied the problem of testing whether a bivariate polynomial is *real zero* (a special case of real stability). It reduced that problem to testing PSDness of a one-parameter family of matrices which it then suggested could be solved using semidefinite programming, but without quite proving a theorem to that effect. This work is partly inspired by ideas in [9].

The paper [12] gives semidefinite programming based algorithms that can test whether certain restricted classes of *multiaffine* polynomials are real stable (in more than 2 variables).

The problem of certifying that a univariate polynomial is nonnegative is typically stated (for instance, in lecture notes) as being the solution to a semidefinite program. If one were able to work out the appropriate error to which the SDP has to be solved, this could give a weakly polynomial time algorithm for nonnegativity, which we suspect must be known as folklore. The paper [18] analyzes a semidefinite programming based algorithm in the special case when the polynomial is nondegenerate in an appropriate sense.

2 Real Stable and Hyperbolic Polynomials

We recall below the definition of a real stable polynomial in an arbitrary number of variables.

▶ **Definition 3.** A polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is called *real stable* if it is identically zero[3] or if $p(z_1, \ldots, z_n) \neq 0$ whenever $\mathrm{Im}(z_i) > 0$ for all $i = 1, \ldots, n$. Equivalently, p is real stable if and only if the univariate restrictions

$$t \mapsto p(te_1 + x_1, te_2 + x_2, \ldots, te_n + x_n)$$

are real rooted whenever $e_1, \ldots, e_n > 0$ and $x_1, \ldots, x_n \in \mathbb{R}$.

The equivalence between the two formulations above is an easy exercise. Note that a univariate polynomial is real stable if and only if it is real rooted. Note that we consider the zero polynomial to be real-rooted.

We will frequently use the elementary fact that a limit of real-rooted polynomials is real-rooted, which follows from Hurwitz's theorem (see, e.g. [20, Sec. 2]), or from the argument principle.

Real Stable polynomials are closely related to the following more general class of polynomials.

▶ **Definition 4.** A homogeneous polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is called *hyperbolic* with respect to a point $e = (e_1, \ldots, e_n) \in \mathbb{R}^n$ if $p(e) > 0$ and the univariate restrictions

$$t \mapsto p(te + x)$$

are real rooted for all $x \in \mathbb{R}^n$. The connected component of $\{x \in \mathbb{R}^n : p(x) \neq 0\}$ containing e is called the *hyperbolicity cone* of p with respect to e, and will be denoted $K(p, e)$.

Perhaps the most familiar example of a hyperbolic polynomial is the determinant of a symmetric matrix:

$$X \mapsto \det(X)$$

[3] Some works (e.g. [4]) consider only nonzero polynomials to be stable, while others [20] include the zero polynomial. We find the latter convention more convenient.

for real symmetric X, which is hyperbolic with respect to the identity matrix since the characteristic polynomial of a symmetric matrix is always real rooted. The corresponding hyperbolicity cone is the cone of positive semidefinite matrices.

The most important theorem regarding hyperbolic polynomials says that hyperbolicity cones are *always* convex, and that hyperbolicity at one point in the cone implies hyperbolicity at every other point. Thus, hyperbolic polynomials and hyperbolicity cones may be viewed as generalizing determinants and PSD cones.

▶ **Theorem 5** (Garding [6]). *If $p \in \mathbb{R}[x_1, \ldots, x_n]$ is hyperbolic with respect to $e \in \mathbb{R}^n$ then:*

1. *$K(p, e)$ is an open convex cone.*
2. *p is hyperbolic with respect to every point $y \in K(p, e)$.*

The reason hyperbolic polynomials are relevant in this work is that real stable polynomials are essentially a special case of them.

▶ **Theorem 6** (Borcea-Branden [4]). *A nonzero bivariate polynomial $p(x, y)$ of total degree at most m is real stable if and only if its homogenization*

$$p_H(x, y, z) := z^m p(x/z, y/z)$$

is hyperbolic with respect to every point in

$$\mathbb{R}^2_{>0} \times \{0\} = \{(e_1, e_2, 0) : e_1, e_2 > 0\}.$$

Thus, real stable polynomials enjoy the strong structural properties guaranteed by Theorem 5 as well, and we exploit these in our algorithm.

3 Parameter Reduction via Hyperbolicity

In this section we use the properties of hyperbolic polynomials to reduce real stability of a bivariate polynomial to testing real rootedness of a one parameter family of polynomials.

▶ **Theorem 7** (Reduction to One-Parameter Family). *A nonzero bivarite polynomial $p \in \mathbb{R}_n[x, y]$ ofis real stable if and only if following two conditions hold:*

1. *The one-parameter family of univariate polynomials $q_\gamma \in \mathbb{R}[t]$ given by,*

   $$q_\gamma(t) = p(\gamma + t, t) \in \mathbb{R}[t]$$

   are real rooted for all $\gamma \in \mathbb{R}$.
2. *The univariate polynomial*

   $$t \mapsto p_H(t, 1 - t, 0)$$

   is strictly positive on the interval $(0, 1)$,

Proof. (*real-stability of p* $\implies$ (1) & (2))

By Theorem 6, p_H is hyperbolic with respect to the positive orthant $\mathbb{R}^2_{>0} \times \{0\}$. Since $(1, 1, 0) \in \mathbb{R}^2_{>0} \times \{0\}$, this implies that for all $(x, y, z) \in \mathbb{R}^3$,

$$q(t) = p_H(x + t, y + t, z)$$

is real-rooted. Setting $x = \gamma$, $y = 0$ and $z = 1$ we get that $q_\gamma(t) = p_H(\gamma + t, t, 1) = p(\gamma + t, t)$ is real-rooted for all $\gamma \in \mathbb{R}$ which is condition (1). Finally, since

$$\{(t, 1 - t, 0) | t \in (0, 1)\} \subset \mathbb{R}^2_{>0} \times \{0\}$$

and p_H is hyperbolic with respect to $\mathbb{R}^2_{>0} \times \{0\}$, it follows that $p_H(t, 1-t, 0) > 0$ for all $t \in (0, 1)$.

((1) & (2) $\implies$ *real-stability of* p)

First, we claim that the polynomial p_H is hyperbolic with respect to $(1, 1, 0)$. By (2) we have $p_H(1/2, 1/2, 0) > 0$ so homogeneity implies that $p_H(1, 1, 0) > 0$. It remains to show that $q_{x,y,z}(t) = p_H(x + t, y + t, z)$ is real-rooted for all $(x, y, z) \in \mathbb{R}^3$. First, consider the case of $(x, y, z) \in \mathbb{R}^3$ with $z \neq 0$.

$$\begin{aligned}
&\forall (x, y, z) \in \mathbb{R}^3 \text{ with } z \neq 0, p_H(x + t, y + t, z) \text{ is real-rooted} \\
\iff &\forall (x, y, z) \in \mathbb{R}^3 \text{ with } z \neq 0, p_H(\frac{x}{z} + \frac{t}{z}, \frac{y}{z} + \frac{t}{z}, 1) \text{ is real-rooted} \\
\iff &\forall (x, y, z) \in \mathbb{R}^3 \text{ with } z \neq 0, p_H(\frac{x}{z} + t, \frac{y}{z} + t, 1) \text{ is real-rooted (replacing } t/z \text{ with } t) \\
\iff &\forall (x, y) \in \mathbb{R}^2, p_H(x + t, y + t, 1) \text{ is real-rooted} \\
\iff &\forall (x, y) \in \mathbb{R}^2, p_H(x + t, t, 1) \text{ is real-rooted (replacing } t \text{ with } t - y) \\
\iff &\forall \gamma \in \mathbb{R}, p(\gamma + t, t) \text{ is real-rooted}
\end{aligned}$$

By Hurwitz's theorem, the limit of any sequence of real-rooted polynomials is real-rooted. Therefore, if $q_{x,y,z}(t)$ is real-rooted for all $(x, y, z) \in \mathbb{R}^3$ with $z \neq 0$ then $q_{x,y,z}(t)$ is real-rooted for all $(x, y, z) \in \mathbb{R}^3$.

Given that p_H is hyperbolic with respect to $e = (\frac{1}{2}, \frac{1}{2}, 0)$, its hyperbolicity cone $K(p_H, e)$ is a convex cone containing $(1, 1, 0)$. Condition (2) implies that the connected component of $\{x | p(x) \neq 0\}$ containing $(1, 1, 0)$ contains the open line segment from $(1, 0, 0)$ to $(0, 1, 0)$. Together, this implies that the positive quadrant $\mathbb{R}^2 \times \{0\} \subseteq K(p_H, e)$. By Theorem 6, this implies that p is real-stable. ◀

Thus, our algorithmic goal is reduced to testing whether a one-parameter family is real-rooted, and whether a given univariate polynomial is positive on an interval. We solve these problems in the sequel.

4 Real-rootedness of one-parameter families

In this section we present two algorithms for testing real-rootedness of a one-parameter family of polynomials. Both algorithms reduce this problem to verifying nonnegativity of a finite number of polynomials on the real line. The first algorithm produces n polynomials of degree roughly $O(n^3)$, and has the advantage of being very simple, relying only on elementary techniques and standard algorithms such as fast matrix multiplication and the discrete Fourier transform. The second algorithm produces n polynomials of degree roughly $O(n^2)$ and runs significantly faster, but uses somewhat more specialized (but nonetheless classical) machinery from the theory of resultants.

4.1 A Simple $O(n^{3+\omega})$ Algorithm

The first algorithm is based on the observation that real-rootedness of a single polynomial is equivalent to testing positive semidefiniteness of its moment matrix, which in turn is equivalent to testing nonnegativity of the elementary symmetric polynomials of that matrix. In the more general case of a one-parameter family, the latter polynomials turn out to be polynomials of bounded degree in the parameter, and it therefore suffices to verify that these are nonnegative everywhere.

We begin by recalling the Newton Identities, which express the moments of a polynomial in terms of its coefficients.

▶ **Lemma 8** (Newton Identities). *If*

$$p(x) = \sum_{k=0}^{n} (-1)^k x^{n-k} c_k = c_0 \prod_{i=1}^{n} (x - x_i) \in \mathbb{R}[x]$$

with $c_0 \neq 0$ is a univariate polynomial with roots $x_1, \dots, x_n$, then the moments

$$m_k := \sum_{i=1}^{n} x_i^k$$

satisfy the recurrence:

$$m_k = (-1)^{k-1} \frac{c_k}{c_0} + \sum_{i=1}^{k-1} (-1)^{k-1+i} \frac{c_{k-i}}{c_0} m_i \qquad 0 \leq k \leq n,$$

$$m_k = \sum_{k-n}^{k-1} (-1)^{k-1+i} \frac{c_{k-i}}{c_0} m_i \qquad k > n,$$

$$m_0 = n.$$

The following consequences of Lemma 8 will be relevant to analyzing our algorithm.

▶ **Corollary 9.**

1. *The moments $m_0, \dots, m_{2n-2}$ of a degree n polynomial can be computed from its coefficients in $O(n^2)$ arithmetic operations.*
2. *Suppose $p(x) = \sum_{k=0}^{n} (-1)^k x^{n-k} c_k(\gamma)$ is a polynomial whose coefficients are polynomials $c_0(\gamma), \dots, c_n(\gamma) \in \mathbb{R}_d[\gamma]$ in a parameter γ. Then the moments of p are given by*

   $$m_k(\gamma) = r_k(\gamma)/c_0(\gamma)^k,$$

   for some polynomials $r_k \in \mathbb{R}_{dk}[\gamma]$.

Proof. The first claim follows because each application of the recurrence requires at most n arithmetic operations. For the second claim, observe that each ratio $c_{k-i}(\gamma)/c_0(\gamma)$ is a rational function with a numerator of degree at most d and denominator $c_0(\gamma)$. Thus, each application of the recurrence increases the degree of the numerator by at most d and introduces an additional c_0 in the denominator. ◀

As a subroutine, we will also need the following standard result in linear algebra.

▶ **Theorem 10** (Keller-Gehrig [11]). *Given an $n \times n$ complex matrix A, there is an algorithm which computes the characteristic polynomial of A in time $O(n^\omega \log n)$.*

We now specify the algorithm and prove its correctness.

▶ **Theorem 11.** *A polynomial $p_\gamma(x) = \sum_{k=0}^{n} (-1)^k x^{n-k} c_k(\gamma)$ is real-rooted for all $\gamma \in \mathbb{R}$ if and only if the polynomials $q_0, \dots, q_n$ output by **SimpleRR** are nonnegative on $\mathbb{R}$. Moreover, **SimpleRR** runs in time $\tilde{O}(dn^{2+\omega} + d^2 n^3)$.*

Proof. We first show correctness. Let $m_k(p)$ denote the k^{th} moment of the roots of a polynomial. By Sylvester's theorem [2, Theorem 4.58], a real polynomial

$$p_\gamma(x) = \sum_{k=0}^{n} (-1)^k x^{n-k} c_k(\gamma)$$

is real-rooted if and only if the corresponding moment matrix

$$M(\gamma)_{k,l} := m_{k+l-2}(p_\gamma)$$

is positive semidefinite. Since ν is even and c_0 has real coefficients, we have for every $\gamma \in \mathbb{R}$ that is not a root of c_0:

$$M(\gamma) \succeq 0 \iff c_0(\gamma)^\nu M(\gamma) = H(\gamma) \succeq 0.$$

Since c_0 has only finitely many roots and a limit of PSD matrices is PSD, we conclude that

$$M(\gamma) \succeq 0 \quad \forall \gamma \in \mathbb{R} \iff H(\gamma) \succeq 0 \forall \gamma \in \mathbb{R}.$$

Note that by Corollary 9 the entries of $H(\gamma)$ are polynomials of degree at most $d(\nu + 2n - 2)$ in γ.

We now recall a well-known[4] (e.g., [10]) characterization of positive semidefiniteness as a semialgebraic condition: an $n \times n$ real symmetric matrix A is PSD if and only if $e_k(A) \geq 0$ for all $k = 1, \ldots, n$, where

$$e_k(A) = \sum_{|S|=k} \det(A_{S,S})$$

is the sum of all $k \times k$ principal minors of A. Thus, p_γ is real-rooted for all $\gamma \in \mathbb{R}$ if and only if the polynomials

$$q_k(\gamma) := e_k(H(\gamma))$$

for $k = 1, \ldots, n$ are nonnegative on $\mathbb{R}$.

Since each q_k is a sum of determinants of order at most n in $H(\gamma)$ it has degree at most n in the entries of $H(\gamma)$, and we conclude that $q_1, \ldots, q_n \in \mathbb{R}_N[\gamma]$. Thus, the q_k can be recovered by interpolating them at the N^{th} roots of unity. Since the k^{th} elementary symmetric function of a matrix is the coefficient of z^{n-k} in its characteristic polynomial, this is precisely what is achieved in Step 2.

For the complexity analysis, it is clear that Step 1 takes $O(dn^2)$ time. Constructing each Hankel matrix $H(s_i)$ takes time $O(dn + n^2)$ by Corollary 9, and computing its elementary symmetric functions via the characteristic polynomial takes time $O(n^\omega \log n)$, according to Theorem 10. Thus, the total time for each iteration is $O(n^\omega \log n + dn)$, so the time for all iterations is $O(dn^{2+\omega} \log n + d^2 n^3)$. The final step requires $O(N \log N)$ time for each e_k using fast polynomial interpolation via the discrete Fourier transform, for a total of $O(dn^3 \log n)$. Thus, the total running time is $\tilde{O}(dn^{2+\omega} + d^2 n^3)$, suppressing logarithmic factors. ◀

[4] Here is a short proof: A is PSD iff $\det(zI - A)$ has only nonnegative roots. Since A is symmetric we know the roots are real. We now observe that a real-rooted polynomial has nonnegative roots if and only if its coefficients alternate in sign.

Algorithm 1: SimpleRR

Input: $(n+1)$ univariate polynomials $c_0, \ldots, c_n \in \mathbb{R}_d[\gamma]$ with $c_0 \not\equiv 0$.
Output: n univariate polynomials $q_1, \ldots, q_n \in \mathbb{R}_{3n^2d}[\gamma]$

1. Let ν be the first even integer greater than or equal to n and let $N = nd(2n-2+\nu) = O(dn^2)$. Let $s_1, \ldots, s_N \in \mathbb{C}$ be the N^{th} roots of unity.
2. For each $i = 1, \ldots N$:
   - Compute the $n \times n$ Hankel matrix $H(s_i)$ with entries
   $$H(s_i)_{k,l} := c_0(s_i)^\nu m_{k+l-2}(p_{s_i}),$$
   by applying the Newton identities (Lemma 8).
   - Compute the characteristic polynomial
   $$\det(zI - H(s_i)) = \sum_{k=0}^{n} (-1)^k z^{n-k} e_k(H(s_i))$$
   using the Keller-Gehrig algorithm (Theorem 10).
3. For each $k = 1, \ldots, n$: Use the points $e_k(H(s_1)), \ldots, e_k(H(s_N))$ to interpolate the coefficients of the polynomial
   $$q_k(\gamma) := e_k(H(\gamma)).$$

Output $q_1, \ldots, q_n$.

4.2 A Faster $O(n^4)$ Algorithm Using Subresultants

The algorithm of the previous section is based on the generic fact that a matrix is PSD if and only if its elementary symmetric polynomials are nonnegative. In this section we exploit the fact that our matrices have a special structure – namely, they are moment matrices – to find a different finite set of polynomials whose nonnegativity suffices to certify their PSDness. These polynomials are called *subdiscriminants*, and turn out to be related to another class of polynomials called *subresultants*, for which there are known fast symbolic algorithms.

Let M_p denote the $n \times n$ moment matrix corresponding to a polynomial p of degree n. Recall that $M_p = VV^T$ where V is the Vandermonde matrix formed by the roots of p. Let $(M_p)_i$ denote the leading principal $i \times i$ minor of M_p. We define subdiscriminants of a polynomial, and then show their relation to the leading principal minors of the moment matrix. For the remainder of this section it will be more convenient to use the notation

$$p(x) = \sum_{k=0}^{n} a_k x^k$$

for the coefficients of a polynomial, with roots $x_1, \ldots, x_n$ and $a_n \neq 0$.

▶ **Definition 12.** The k^{th} *subdiscriminant* of a polynomial p is defined as

$$\texttt{sDisc}_k(p) = a_n^{2k-2} \sum_{S \subset \{1,\ldots,n\}, |S|=k} \prod_{\{i,j\} \subset S} (x_i - x_j)^2$$

▶ **Lemma 13.** *The leading principal minors of the moment matrix are multiples of the* *subdiscriminants,*

$$(M_p)_i = a_n^{2-2k}\mathtt{sDisc}_k(p) = \sum_{S\subset\{1,\ldots,n\},|S|=k}\ \prod_{\{i,j\}\subset S}(x_i - x_j)^2$$

Proof. Let

$$V_i = \begin{bmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_n \\ \vdots & \vdots & \vdots \\ x_1^{i-1} & \ldots & x_n^{i-1} \end{bmatrix}.$$

Then $(M_p)_i = \det(V_i V_i^T)$. By Cauchy-Binet, this determinant is the sum over the determinants of all submatrices of size $i \times i$. These submatrices are exactly the Vandermonde matrices formed by subsets of the roots of size i. Then the identity follows from the formula for the determinant of a Vandermonde matrix. ◀

Equipped with this we can provide an alternative characterization of real rootedness. Define the sign of a number, denoted $\mathtt{sgn}$ to be $+1$ if it is positive, -1 if it is negative, and 0 otherwise.

▶ **Lemma 14.** *p is real-rooted if and only if the sequence $\mathtt{sgn}(\mathtt{sDisc}_1(p)), \ldots, \mathtt{sgn}(\mathtt{sDisc}_n(p))$ is first 1's and then 0's.*

Proof. Note that since $a_n \neq 0$ we have $\mathtt{sgn}(Disc_k) = \mathtt{sgn}(a_n^{2(1-k)}Disc_k) = \mathtt{sgn}((M_p)_i)$. It is clear from the definition of the subdiscriminants that if p is real-rooted with k distinct roots then $\mathtt{sDisc}_i$ is positive if $i \leq k$ and $\mathtt{sDisc}_i = 0$ if $i > k$.

Conversely, given a polynomial p with k distinct roots, then if $i > k$ we have all the minors of size i in V_i^T contain two identical rows, and hence V_i^T does not have full rank, so $V_i V_i^T$ is singular. Let $x_1, x_2, \ldots, x_j$ be the real distinct roots of p and $y_1, \bar{y_1}, \ldots, y_l, \bar{y_l}$ be the distinct complex conjugate pairs of p where $j + l = k$. Suppose the multiplicities of x_i are n_i and y_i are m_i. Then the top left $k \times k$ submatrix of M_p is

$$= \sum_i n_i \begin{bmatrix} 1 \\ x_i \\ \vdots \\ x_i^{k-1} \end{bmatrix} \begin{bmatrix} 1 & x_i & \cdots & x_i^{k-1} \end{bmatrix} + \sum_i m_i \begin{bmatrix} 1 \\ y_i \\ \vdots \\ y_i^{k-1} \end{bmatrix} \begin{bmatrix} 1 & y_i & \cdots & y_i^{k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ \bar{y_i} \\ \vdots \\ \bar{y_i}^{k-1} \end{bmatrix} \begin{bmatrix} 1 & \bar{y_i} & \cdots & \bar{y_i}^{k-1} \end{bmatrix}$$

$$= \sum_i n_i \begin{bmatrix} 1 \\ x_i \\ \vdots \\ x_i^{k-1} \end{bmatrix} \begin{bmatrix} 1 & x_i & \cdots & x_i^{k-1} \end{bmatrix} + \sum_i m_i \begin{bmatrix} 1 & 1 \\ \mathrm{Re}(y_i) & \mathrm{Im}(y_i) \\ \vdots & \vdots \\ \mathrm{Re}(y_i^{k-1}) & \mathrm{Im}(y_i^{k-1}) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ \mathrm{Re}(y_i) & \mathrm{Im}(y_i) \\ \vdots & \vdots \\ \mathrm{Re}(y_i^{k-1}) & \mathrm{Im}(y_i^{k-1}) \end{bmatrix}^T$$

This shows that this submatrix is positive definite if and only if the distinct roots are all real. Note that by Sylvester's criterion this submatrix is positive definite if and only if all the leading principal minors of size $\leq k$ are positive. ◀

We now obtain a formula for the subdiscriminants of a polynomial in terms of its coefficients. The connection is provided by another family of polynomials called the *subresultants*.

▶ **Definition 15.** Let $p = \sum_{k=0}^{n} a_k x^k$ where $a_n \neq 0$. The *kth subresultant* of p, denoted $\mathtt{sRes}_k(p, p')$ is the determinant of the submatrix obtained from the first $2n-1-2k$ columns of the following $(2n-1-2k) \times (2n-1-k)$ matrix:

$$\begin{bmatrix} a_n & \cdots & \cdots & \cdots & \cdots & a_0 & 0 & 0 \\ 0 & \ddots & & & & & \ddots & 0 \\ \vdots & \ddots & a_n & \cdots & \cdots & \cdots & \cdots & a_0 \\ \vdots & & 0 & na_n & \cdots & \cdots & \cdots & a_1 \\ \vdots & \iddots & \iddots & & & & \iddots & 0 \\ 0 & \iddots & & & & \iddots & \iddots & \vdots \\ na_n & \cdots & \cdots & \cdots & a_1 & 0 & \cdots & 0 \end{bmatrix}$$

We will use two properties of subresultants. The first is a good bound on their degree as a consequence of the determinantal formula above. The second is quick algorithm to compute them. We refer the reader to [2] for a more detailed discussion of subresultants.

In this paper we will only be interested in subresultants of a polynomial with its derivative We are interested in this because of its relation to our leading principal minors:

▶ **Lemma 16** ([2] Proposition 4.27). *Let $p(x) = \sum_{k=0}^{n} a_k x^k$ where $a_n \neq 0$*

$$\mathtt{sRes}_k(p, p') = a_n \mathtt{sDisc}_{n-k}(p)$$

▶ **Corollary 17.** *Since the first column of the determinant used to define the subresultant is divisible by a_n, we get $\mathtt{sDisc}_k(p)$ is a polynomial in our coefficients $a_n, \ldots, a_0$ of degree at most $2n$.*

The benefit of studying the principal minors instead of the coefficients of the characteristic polynomial for our moment matrix is that we can use an algorithm from subresultant theory to quickly calculate all the minors at once.

▶ **Theorem 18** ([2] Algorithm 8.21). *There exists an algorithm which, given a polynomial p of degree n returns a list of all of its subresultants $\mathtt{sRes}_k(p, p')$ for $k = 1, \ldots, n$ in $O(n^2)$ time.*

▶ Remark. Many computer algebra systems (e.g., Mathematica, Macaulay2) have built-in efficient algorithms to compute subresultants.

We now combine the above facts to obtain a crisp condition for real-rootedness of a one-parameter family. Recall that by Theorem 7, we are interested in testing when a family of polynomials $p_\gamma(x)$ are real-rooted for all $\gamma \in \mathbb{R}$, where

$$p_\gamma(x) = \sum_{k=0}^{n} a_k(\gamma) x^k$$

with $c_k \in \mathbb{R}_n[\gamma]$. Let $c_m(\gamma)$ be the highest coefficient that is not identically zero. We are only interested in the case when $m \geq 2$.

▶ **Proposition 19.** *If $p_\gamma(x) = \sum_{k=0}^{n} x^k c_k(\gamma)$ with $c_k \in \mathbb{R}_d[\gamma]$, then $\mathtt{sDisc}_k(p_\gamma)$ is a polynomial in γ of degree at most $2dn$.*

Proof. From our previous lemma, we know that $\mathtt{sDisc}_k$ is a polynomial in the coefficients of p of degree at most $2n$. Since each of these coefficients $c_k(\gamma)$ is a polynomial in γ of degree at most d, our result follows. ◀

We now extend our characterization of real-rootedness in terms of the signs of the principal minors of a fixed polynomial to a characterization for coefficients which are polynomials in γ.

▶ **Theorem 20.** *$p_\gamma(x)$ is real-rooted for all $\gamma \in \mathbb{R}$ if and only if there exists a k such that $\mathtt{sDisc}_i(p_\gamma)$ is a nonnegative polynomial which is not identically zero for all $i \le k$ and $\mathtt{sDisc}_i(p_\gamma)$ is identically zero for $i > k$.*

Proof. First suppose that $p_\gamma(x)$ is real rooted for all $\gamma \in \mathbb{R}$. Observe that $c_m(\gamma)$ vanishes for at most finitely many points Z_1. Moreover, the degree m discriminant of p_γ is a polynomial in γ, and is zero for at most finitely many points — call them Z_2. Thus, for $\gamma \notin Z_1 \cup Z_2$, we know that p_γ has exactly m distinct real roots, so by Lemma 14 $\mathtt{sDisc}_i(p_\gamma)$ is strictly positive for $i \le m$ and zero for $i > m$ on this set. By continuity this implies that $\mathtt{sDisc}_i(p_\gamma)$ is nonnegative and not identically zero on $\mathbb{R}$ for $i \le m$, and $\mathtt{sDisc}_i(p_\gamma)$ is identically zero for $i > m$, as desired.

To prove the converse, note that for $i \le k$, $\mathtt{sDisc}_i(p_\gamma(t))$ is not identically zero, and hence there are finitely many γ away from which $\mathtt{sDisc}_i(p_\gamma)$ is positive for all $i \le k$, and then all zero. By Lemma 14 we get that $p_\gamma(x)$ is real rooted for all these γ. Since real-rootedness is preserved by taking limits (by Hurwitz's theorem), we conclude that $p_\gamma(x)$ is real rooted for all $\gamma \in \mathbb{R}$. ◀

Combining these observations, and using the $O(n^2)$ time algorithm to compute the subdiscriminants, we arrive at the following $O(n^4)$ time algorithm for computing all the subdiscriminants.

Algorithm 2: FastRR

Input: $(n+1)$ univariate polynomials $c_0, \ldots, c_n \in \mathbb{R}_d[\gamma]$ with $c_0 \not\equiv 0$.
Output: n univariate polynomials $q_1, \ldots, q_n \in \mathbb{R}_{2dn}[\gamma]$

1. Find distinct points $\gamma_1, \ldots, \gamma_{2dn} \in$ such that $c_m(\gamma_i) \neq 0$.
2. For each γ_i use the subresultant algorithm (Theorem 18) to compute all of the $\mathtt{sRes}_k(p_{\gamma_i})$, with $k = 1, \ldots, n$.
3. Use the above values to compute $2dn$ different values $q_k(\gamma_1), \ldots, q_k(\gamma_{2dn})$ for each of the polynomials
$$q_k(\gamma) := \mathtt{sDisc}_k(p_\gamma) = c_m(\gamma)^{-1}\mathtt{sRes}_{m-k}(p_\gamma)),$$
$k = 1, \ldots, n$.
4. Use fast interpolation to compute the coefficients of $q_1, \ldots, q_n$.

Output $q_1, \ldots, q_n$.

▶ **Theorem 21.** ***FastRR*** *runs in $O(n^4)$ time.*

Proof. Since $c_n(\gamma)$ is of degree at most d we can test $2dn + d$ points to find $2dn$ points on which $c_n(\gamma)$ doesnt vanish. Each evaluation takes $O(d)$ times, so total this takes $O(d^2n)$ time. To compute $\mathtt{sRes}_k(p_{\gamma_i})$ for each $0 \le k \le n-1$ and $1 \le i \le 2dn$ takes $O(dn^3)$ time by Theorem 18. Then to scale all the subresultants, since we have $O(dn^2)$ data points and have already computed $c_n(\gamma_i)$ takes $O(dn^2)$ time. Finally, since the degrees of the q_k are at most $2dn$, the total time to interpolate all of them is $O(dn^2 \log n)$. ◀

5 Univariate Nonnegativity Testing

In this section, we describe an algorithm to test non-negativity of a univariate polynomial over the real line.

Let $p \in \mathbb{R}[x]$ denote a univariate polynomial of degree d. The goal of the algorithm is to test if $p(x) \geq 0$ for all $x \in \mathbb{R}$. A canonical approach for the problem would be to use a Sum-of-Squares semidefinite program to express p as a sum of squares of low-degree polynomials. Unfortunately, the resulting algorithm is not a symbolic algorithm, i.e., its runtime is not strongly polynomial in the degree d, since semidefinite programming is not known to be strongly polynomial.

We will now describe a strongly polynomial time algorithm to test non-negativity of the polynomial p. Our starting point is an algorithm to count the number of real roots of a polynomial using Sturm sequences. We refer the reader to Basu et al. [2] for a detailed presentation of Sturm sequences and algorithms to compute them. For our purposes, we will need the following lemma.

▶ **Lemma 22.** *Given a univariate polynomial $p \in \mathbb{R}[x]$, the algorithm based on computing Sturm sequences uses $O(\deg(p)^2)$ arithmetic operations to determine the number of real roots of p.*

The polynomial p is positive, i.e., $p(x) > 0$ for all $x \in \mathbb{R}$, if and only if it has no real roots. Therefore, Lemma 22 yields an algorithm to test positivity using in $O(d^2)$ arithmetic operations. To test non-negativity, the only additional complication stems from the roots of the polynomial p. We begin with a simple observation.

▶ **Fact 23.** *If $p \in \mathbb{R}[x]$ is monic then $p(x) \geq 0$ for all $x \in \mathbb{R}$ if and only if p has no real roots of odd multiplicity.*

▶ **Definition 24.** A square-free decomposition of a polynomial $p \in \mathbb{R}[x]$ of degree d, is a set of polynomials $\{a_1, \ldots, a_d\} \in \mathbb{R}[x]$ such that

$$p(x) = \prod_{i=1}^{d} a_i(x)^i \ ,$$

and each a_i has no roots with multiplicity greater than one. Alternately, for each $i \in [d]$, $a_i(x)^i$ consists of all roots of p with multiplicity exactly i.

Square-free decompositions can be computed efficiently using gcd computations. Yun [21] carries out a detailed analysis of square-free decomposition algorithms. In particular, he shows that an algorithm due to Musser can be used to compute square-free decompositions at the cost of constantly many gcd computations.

Now, we are ready to describe an algorithm to test non-negativity.

Algorithm 3: Nonnegative

Input: A monic polynomial $p \in \mathbb{R}[x]$, $\deg(p) = d$
Goal: Test if $p(x) \geq 0$ for all $x \in \mathbb{R}$.

1. Using Musser's algorithm, compute the square-free decomposition of p given by,
$$p = \prod_{i \in [d]} a_i^i$$
where $a_i \in \mathbb{R}[x]$ has no roots with multiplicity greater than 1.
2. For each $i \in [\lceil \frac{d}{2} \rceil]$
   - Using Sturm sequences, test if a_{2i-1} has real roots. If a_{2i-1} has real roots p is NOT non-negative.

Runtime

Let $T_{gcd}(d)$ denote the time-complexity of computing the gcd of two univariate polynomials of degree d. The runtime of Musser's square-free decomposition algorithm is within constant factors of $T_{gcd}(d)$. Let $S_{real}(\ell)$ denote the time-complexity of determining if a degree ℓ polynomial has no real roots. Observe that

$$\sum_i \deg(a_i) \leq \deg(p) = d$$

Since $S_{real}(\ell)$ is super-linear in ℓ, we have $\sum_{i \in [d]} S_{real}(a_i) \leq S_{real}(d)$. The run-time of the algorithm is given by $O(T_{gcd}(d) + S_{real}(d))$. Using Sturm sequences, $S_{real}(d) = O(d^2)$ elementary operations on real numbers (see [2]). Using Euclid's algorithm, $T_{gcd}(d) = O(d^2)$ elemenetary operations on real numbers. This yields an algorithm for non-negativity that incurs at most $O(d^2)$ elementary operations.

6 Conclusion and Discussion

Finally, we combine the ingredients from sections 3, 4, and 5 to obtain the proof of our main theorem.

Proof of Theorem 1. Given the coefficients of p, we can compute the coefficients of the one-parameter family in (1) of Theorem 7 in time at most $O(n^3)$. By Theorem 21, **FastRR** produces the polynomials $q_1, \ldots, q_n$ in time $O(n^4)$. We check that some final segment of these polynomials are identically zero by evaluating each one at $O(n^2)$ points. These polynomials have degree $O(n^2)$, so **Nonnegative** requires time $O(n^4)$ to check nonnegativity of each remaining one, for a total running time of $O(n^5)$.

For part (2) of Theorem 7, we simply use a Sturm sequence to ensure that there are no roots in $(0, 1)$, and then evaluate the polynomial at a single point to check that the sign is positive. ◀

The algorithm in this paper offers a starting point in the area of polynomial time algorithms for real stability. In addition to the obvious possibility of improving the running time to say $O(n^4)$ or below, several natural open questions remain:

- Can the algorithm be generalized to 3 or more variables? The bottleneck to doing this is that we do not know how to check real rootedness of 2-parameter families, or equivalently, nonnegativity of bivariate polynomials.

- Is there an algorithm for testing whether a given polynomial is hyperbolic with respect to *some* direction, without giving the direction as part of the input?
- Is there an algorithm for testing stability of bivariate polynomials with *complex* coefficients?

Perhaps leaving the realm of strongly polynomial time algorithms, the major open question in this area is the following: a famous theorem of Helton and Vinnikov [8] asserts that every bivariate real stable polynomial can be written as

$$p(x, y) = \det(xA + yB + C)$$

for some positive semidefinite matrices A, B and real symmetric C. Unfortunately, their proof does not give an efficient algorithm for finding these matrices. Can the ideas in this paper, perhaps via using SDPs to find sum-of-squares representations of certain nonnegative polynomials derived from p, be used to obtain such an algorithm?

Acknowledgments. We thank Eric Hallman, Navin Goyal, Jonathan Leake, and Bernd Sturmfels for valuable discussions. We also thank Didier Henrion for valuable correspondence regarding other algorthimic approaches to these problems, and Thorsten Theobald for pointing out the reference [3].

References

1 Nima Anari and Shayan Oveis Gharan. The kadison-singer problem for strongly rayleigh measures and applications to asymmetric tsp. *arXiv preprint arXiv:1412.1143*, 2014.

2 Saugata Basu, Richard Pollack, and Marie-Francoise Roy. *Algorithms in real algebraic geometry*, volume 20033. Springer, 2005.

3 Michael Ben-Or, Dexter Kozen, and John Reif. The complexity of elementary algebra and geometry. *Journal of Computer and System Sciences*, 32(2):251–264, 1986.

4 Julius Borcea and Petter Brändén. The lee-yang and pólya-schur programs. i. linear operators preserving stability. *Inventiones mathematicae*, 177(3):541–569, 2009.

5 Julius Borcea, Petter Brändén, and Thomas Liggett. Negative dependence and the geometry of polynomials. *Journal of the American Mathematical Society*, 22(2):521–567, 2009.

6 Lars Garding. An inequality for hyperbolic polynomials. *Journal of Mathematics and Mechanics*, 8(6):957–965, 1959.

7 Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 550–559. IEEE, 2011.

8 J William Helton and Victor Vinnikov. Linear matrix inequality representation of sets. *Communications on pure and applied mathematics*, 60(5):654–674, 2007.

9 Didier Henrion. Detecting rigid convexity of bivariate polynomials. *Linear Algebra and its Applications*, 432(5):1218–1233, 2010.

10 Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

11 Walter Keller-Gehrig. Fast algorithms for the characteristics polynomial. *Theoretical computer science*, 36:309–317, 1985.

12 Mario Kummer, Daniel Plaumann, and Cynthia Vinzant. Hyperbolic polynomials, interlacers, and sums of squares. *Mathematical Programming*, 153(1):223–245, 2015.

13 Adam Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families i: Bipartite ramanujan graphs of all degrees. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 529–537. IEEE, 2013.

14 Adam W Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families ii: Mixed characteristic polynomials and the kadison–singer problem. *Annals of Mathematics*, 182(1):327–350, 2015.

15 Adam W Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families iv: Bipartite ramanujan graphs of all sizes. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1358–1377. IEEE, 2015.

16 Wim Nuij. A note on hyperbolic polynomials. *Mathematica Scandinavica*, 23(1):69–72, 1969.

17 Robin Pemantle. Hyperbolicity and stable polynomials in combinatorics and probability. *arXiv preprint arXiv:1210.3231*, 2012.

18 Helfried Peyrl and Pablo A Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science*, 409(2):269–281, 2008.

19 Charles Sturm. *Mémoire sur la résolution des équations numériques.* 1835.

20 David Wagner. Multivariate stable polynomials: theory and applications. *Bulletin of the American Mathematical Society*, 48(1):53–84, 2011.

21 David Y.Y. Yun. On square-free decomposition algorithms. In *Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation*, SYMSAC'76, pages 26–35, New York, NY, USA, 1976. ACM. doi:10.1145/800205.806320.