

# The Budan–Fourier Theorem and Counting Real Roots with Multiplicity

Wenda Li

June 15, 2026

## Abstract

This entry is mainly about counting and approximating real roots (of a polynomial) with multiplicity. We have first formalised the Budan–Fourier theorem: given a polynomial with real coefficients, we can calculate sign variations on Fourier sequences to over-approximate the number of real roots (counting multiplicity) within an interval. When all roots are known to be real, the over-approximation becomes tight: we can utilise this theorem to count real roots exactly. It is also worth noting that Descartes’ rule of sign is a direct consequence of the Budan–Fourier theorem, and has been included in this entry. In addition, we have extended previous formalised Sturm’s theorem to count real roots with multiplicity, while the original Sturm’s theorem only counts distinct real roots. Compared to the Budan–Fourier theorem, our extended Sturm’s theorem always counts roots exactly but may suffer from greater computational cost.

Many problems in real algebraic geometry is about counting or approximating roots of a polynomial. Previous formalised results are mainly about counting distinct real roots (i.e. Sturm’s theorem in Isabelle/HOL [5, 2], HOL Light [4], PVS [9] and Coq [8]) and limited support for multiple real roots (i.e. Descartes’ rule of signs in Isabelle/HOL [3], HOL Light and Proof-Power<sup>1</sup>). In comparison, this entry provides more comprehensive support for reasoning about multiple real roots.

The main motivation of this entry is to cope with the roots-on-the-border issue when counting complex roots [7, 6], but the results here should be beneficial to other developments.

Our proof of the Budan–Fourier theorem mainly follows Theorem 2.35 in the book by Basu et al. [1] and that of the extended Sturm’s theorem is inspired by Theorem 10.5.6 in Rahman and Schmeisser’s book [10].

---

<sup>1</sup>According to Freek Wiedijk’s “Formalising 100 Theorems” (<http://www.cs.ru.nl/~freek/100/index.html>)

# 1 Misc results for polynomials and sign variations

**theory** *BF-Misc imports*

*HOL-Computational-Algebra.Polynomial-Factorial*

*HOL-Computational-Algebra.Fundamental-Theorem-Algebra*

*Sturm-Tarski.Sturm-Tarski*

**begin**

## 1.1 More results about sign variations (i.e. *changes*)

**lemma** *changes-0[simp]:changes (0#xs) = changes xs*

*<proof>*

**lemma** *changes-Cons:changes (x#xs) = (if filter (λx. x≠0) xs = [] then*

*0*

*else if x\* hd (filter (λx. x≠0) xs) < 0 then*

*1 + changes xs*

*else changes xs)*

*<proof>*

**lemma** *changes-filter-eq:*

*changes (filter (λx. x≠0) xs) = changes xs*

*<proof>*

**lemma** *changes-filter-empty:*

**assumes** *filter (λx. x≠0) xs = []*

**shows** *changes xs = 0 changes (a#xs) = 0 <proof>*

**lemma** *changes-append:*

**assumes** *xs≠ [] ∧ ys≠ [] → (last xs = hd ys ∧ last xs≠0)*

**shows** *changes (xs@ys) = changes xs + changes ys*

*<proof>*

**lemma** *changes-drop-dup:*

**assumes** *xs≠ [] ys≠ [] → last xs=hd ys*

**shows** *changes (xs@ys) = changes (xs@ tl ys)*

*<proof>*

**lemma** *Im-poly-of-real:*

*Im (poly p (of-real x)) = poly (map-poly Im p) x*

*<proof>*

**lemma** *Re-poly-of-real:*

*Re (poly p (of-real x)) = poly (map-poly Re p) x*

*<proof>*

## 1.2 More about *map-poly* and *of-real*

**lemma** *of-real-poly-map-pCons*[simp]: *map-poly of-real (pCons a p) = pCons (of-real a) (map-poly of-real p)*  
⟨proof⟩

**lemma** *of-real-poly-map-plus*[simp]: *map-poly of-real (p + q) = map-poly of-real p + map-poly of-real q*  
⟨proof⟩

**lemma** *of-real-poly-map-smult*[simp]: *map-poly of-real (smult s p) = smult (of-real s) (map-poly of-real p)*  
⟨proof⟩

**lemma** *of-real-poly-map-mult*[simp]: *map-poly of-real (p\*q) = map-poly of-real p \* map-poly of-real q*  
⟨proof⟩

**lemma** *of-real-poly-map-poly*:  
*of-real (poly p x) = poly (map-poly of-real p) (of-real x)*  
⟨proof⟩

**lemma** *of-real-poly-map-power*: *map-poly of-real (p<sup>n</sup>) = (map-poly of-real p)<sup>n</sup>*  
⟨proof⟩

**lemma** *of-real-poly-eq-iff* [simp]: *map-poly of-real p = map-poly of-real q  $\longleftrightarrow$  p = q*  
⟨proof⟩

**lemma** *of-real-poly-eq-0-iff* [simp]: *map-poly of-real p = 0  $\longleftrightarrow$  p = 0*  
⟨proof⟩

## 1.3 More about *order*

**lemma** *order-multiplicity-eq*:  
**assumes** *p $\neq$ 0*  
**shows** *order a p = multiplicity [-a,1:] p*  
⟨proof⟩

**lemma** *order-gcd*:  
**assumes** *p $\neq$ 0 q $\neq$ 0*  
**shows** *order x (gcd p q) = min (order x p) (order x q)*  
⟨proof⟩

**lemma** *map-poly-order-of-real*:  
**assumes** *p $\neq$ 0*  
**shows** *order (of-real t) (map-poly of-real p) = order t p* ⟨proof⟩

**lemma** *order-pcompose*:

**assumes** *pcompose p q ≠ 0*

**shows**  $\text{order } x \text{ (pcompose } p \text{ } q) = \text{order } x \text{ (} q \text{ -[:poly } q \text{ } x\text{:]) * order (poly } q \text{ } x) \text{ } p$   
*<proof>*

## 1.4 Polynomial roots / zeros

**definition** *roots-within*::*'a::comm-semiring-0 poly ⇒ 'a set ⇒ 'a set* **where**  
*roots-within p s = {x∈s. poly p x=0}*

**abbreviation** *roots*::*'a::comm-semiring-0 poly ⇒ 'a set* **where**  
*roots p ≡ roots-within p UNIV*

**lemma** *roots-def*:  $\text{roots } p = \{x. \text{poly } p \text{ } x=0\}$   
*<proof>*

**lemma** *roots-within-empty[simp]*:  
*roots-within p {} = {}* *<proof>*

**lemma** *roots-within-0[simp]*:  
*roots-within 0 s = s* *<proof>*

**lemma** *roots-withinI[intro,simp]*:  
 $\text{poly } p \text{ } x=0 \implies x \in s \implies x \in \text{roots-within } p \text{ } s$   
*<proof>*

**lemma** *roots-within-iff[simp]*:  
 $x \in \text{roots-within } p \text{ } s \iff \text{poly } p \text{ } x=0 \wedge x \in s$   
*<proof>*

**lemma** *roots-within-union*:  
 $\text{roots-within } p \text{ } A \cup \text{roots-within } p \text{ } B = \text{roots-within } p \text{ } (A \cup B)$   
*<proof>*

**lemma** *roots-within-times*:  
**fixes** *s::'a::{semiring-no-zero-divisors,comm-semiring-0}* *set*  
**shows**  $\text{roots-within } (p*q) \text{ } s = \text{roots-within } p \text{ } s \cup \text{roots-within } q \text{ } s$   
*<proof>*

**lemma** *roots-within-gcd*:  
**fixes** *s::'a::{factorial-ring-gcd,semiring-gcd-mult-normalize}* *set*  
**shows**  $\text{roots-within } (\text{gcd } p \text{ } q) \text{ } s = \text{roots-within } p \text{ } s \cap \text{roots-within } q \text{ } s$   
*<proof>*

**lemma** *roots-within-inter*:  
*NO-MATCH UNIV s ⇒ roots-within p s = roots p ∩ s*  
*<proof>*

**lemma** *roots-within-roots[simp]*:

*roots-within*  $p$   $s \subseteq$  *roots*  $p$   
 ⟨proof⟩

**lemma** *finite-roots[simp]*:  
**fixes**  $p :: 'a::idom$  *poly*  
**shows**  $p \neq 0 \implies$  *finite* (*roots-within*  $p$   $s$ )  
 ⟨proof⟩

**lemma** *roots-within-pCons-1-iff*:  
**fixes**  $a :: 'a::idom$   
**shows** *roots-within*  $[-a, 1:]$   $s =$  (*if*  $a \in s$  *then*  $\{a\}$  *else*  $\{\}$ )  
*roots-within*  $[a, -1:]$   $s =$  (*if*  $a \in s$  *then*  $\{a\}$  *else*  $\{\}$ )  
 ⟨proof⟩

**lemma** *roots-within-uminus[simp]*:  
**fixes**  $p :: 'a::comm-ring$  *poly*  
**shows** *roots-within*  $(- p)$   $s =$  *roots-within*  $p$   $s$   
 ⟨proof⟩

**lemma** *roots-within-smult*:  
**fixes**  $a :: 'a::\{semiring-no-zero-divisors, comm-semiring-0\}$   
**assumes**  $a \neq 0$   
**shows** *roots-within*  $(smult$   $a$   $p)$   $s =$  *roots-within*  $p$   $s$   
 ⟨proof⟩

## 1.5 Polynomial roots counting multiplicities.

**definition** *roots-count*:: $'a::idom$  *poly*  $\Rightarrow$   $'a$  *set*  $\Rightarrow$  *nat* **where**  
*roots-count*  $p$   $s =$   $(\sum_{r \in \text{roots-within } p \text{ } s. \text{order } r} p)$

**lemma** *roots-count-empty[simp]*:*roots-count*  $p$   $\{\} = 0$   
 ⟨proof⟩

**lemma** *roots-count-times*:  
**fixes**  $s :: 'a::idom$  *set*  
**assumes**  $p * q \neq 0$   
**shows** *roots-count*  $(p * q)$   $s =$  *roots-count*  $p$   $s +$  *roots-count*  $q$   $s$   
 ⟨proof⟩

**lemma** *roots-count-power-n-n*:  
**shows** *roots-count*  $([: - a, 1:]^n)$   $s =$  (*if*  $a \in s \wedge n > 0$  *then*  $n$  *else*  $0$ )  
 ⟨proof⟩

**lemma** *degree-roots-count*:  
**fixes**  $p :: complex$  *poly*  
**shows** *degree*  $p =$  *roots-count*  $p$  *UNIV*  
 ⟨proof⟩

**lemma** *roots-count-smult*:

**fixes**  $a::'a::\{\text{semiring-no-zero-divisors},\text{idom}\}$   
**assumes**  $a\neq 0$   
**shows**  $\text{proots-count } (\text{smult } a \ p) \ s = \text{proots-count } p \ s$   
 $\langle \text{proof} \rangle$

**lemma**  $\text{proots-count-pCons-1-iff}$ :  
**fixes**  $a::'a::\text{idom}$   
**shows**  $\text{proots-count } [:-a,1:] \ s = (\text{if } a\in s \text{ then } 1 \text{ else } 0)$   
 $\langle \text{proof} \rangle$

**lemma**  $\text{proots-count-uminus[simp]}$ :  
 $\text{proots-count } (- \ p) \ s = \text{proots-count } p \ s$   
 $\langle \text{proof} \rangle$

**lemma**  $\text{card-proots-within-leq}$ :  
**assumes**  $p\neq 0$   
**shows**  $\text{proots-count } p \ s \geq \text{card } (\text{proots-within } p \ s) \ \langle \text{proof} \rangle$

**lemma**  $\text{proots-count-0-imp-empty}$ :  
**assumes**  $\text{proots-count } p \ s = 0 \ p\neq 0$   
**shows**  $\text{proots-within } p \ s = \{\}$   
 $\langle \text{proof} \rangle$

**lemma**  $\text{proots-count-leq-degree}$ :  
**assumes**  $p\neq 0$   
**shows**  $\text{proots-count } p \ s \leq \text{degree } p \ \langle \text{proof} \rangle$

**lemma**  $\text{proots-count-union-disjoint}$ :  
**assumes**  $A \cap B = \{\} \ p\neq 0$   
**shows**  $\text{proots-count } p \ (A \cup B) = \text{proots-count } p \ A + \text{proots-count } p \ B$   
 $\langle \text{proof} \rangle$

**lemma**  $\text{proots-count-cong}$ :  
**assumes**  $\text{order-eq}:\forall x\in s. \text{order } x \ p = \text{order } x \ q \ \text{and } p\neq 0 \ \text{and } q\neq 0$   
**shows**  $\text{proots-count } p \ s = \text{proots-count } q \ s \ \langle \text{proof} \rangle$

**lemma**  $\text{proots-count-of-real}$ :  
**assumes**  $p\neq 0$   
**shows**  $\text{proots-count } (\text{map-poly of-real } p) \ ((\text{of-real}::-\Rightarrow'a::\{\text{real-algebra-1},\text{idom}\}) \ s)$   
 $= \text{proots-count } p \ s$   
 $\langle \text{proof} \rangle$

**lemma**  $\text{proots-pcompose}$ :  
**fixes**  $p \ q::'a::\text{field poly}$

**assumes**  $p \neq 0$  degree  $q=1$   
**shows**  $\text{roots-count } (p \text{compose } p \ q) \ s = \text{roots-count } p \ (\text{poly } q \ ' \ s)$   
 $\langle \text{proof} \rangle$

## 1.6 Composition of a polynomial and a rational function

**definition**  $fcompose::'a :: \text{field } \text{poly} \Rightarrow 'a \ \text{poly} \Rightarrow 'a \ \text{poly} \Rightarrow 'a \ \text{poly}$  **where**  
 $fcompose \ p \ q \ r = \text{fst } (\text{fold-coeffs } (\lambda a \ (c,d). \ (d*[:a:] + q * c,r*d)) \ p \ (0,1))$

**lemma**  $fcompose-0$  [simp]:  $fcompose \ 0 \ q \ r = 0$   
 $\langle \text{proof} \rangle$

**lemma**  $fcompose-const$ [simp]:  $fcompose \ [:a:] \ q \ r = [:a:]$   
 $\langle \text{proof} \rangle$

**lemma**  $fcompose-pCons$ :  
 $fcompose \ (pCons \ a \ p) \ q1 \ q2 = \text{smult } a \ (q2^\wedge(\text{degree } (pCons \ a \ p))) + q1 * fcompose \ p \ q1 \ q2$   
 $\langle \text{proof} \rangle$

**lemma**  $fcompose-uminus$ :  
 $fcompose \ (-p) \ q \ r = - fcompose \ p \ q \ r$   
 $\langle \text{proof} \rangle$

**lemma**  $fcompose-add-less$ :  
**assumes**  $\text{degree } p1 > \text{degree } p2$   
**shows**  $fcompose \ (p1+p2) \ q1 \ q2$   
 $= fcompose \ p1 \ q1 \ q2 + q2^\wedge(\text{degree } p1 - \text{degree } p2) * fcompose \ p2 \ q1 \ q2$   
 $\langle \text{proof} \rangle$

**lemma**  $fcompose-add-eq$ :  
**assumes**  $\text{degree } p1 = \text{degree } p2$   
**shows**  $q2^\wedge(\text{degree } p1 - \text{degree } (p1+p2)) * fcompose \ (p1+p2) \ q1 \ q2$   
 $= fcompose \ p1 \ q1 \ q2 + fcompose \ p2 \ q1 \ q2$   
 $\langle \text{proof} \rangle$

**lemma**  $fcompose-add-const$ :  
 $fcompose \ ([:a:] + p) \ q1 \ q2 = \text{smult } a \ (q2^\wedge \text{degree } p) + fcompose \ p \ q1 \ q2$   
 $\langle \text{proof} \rangle$

**lemma**  $fcompose-smult$ :  $fcompose \ (\text{smult } a \ p) \ q1 \ q2 = \text{smult } a \ (fcompose \ p \ q1 \ q2)$   
 $\langle \text{proof} \rangle$

**lemma**  $fcompose-mult$ :  $fcompose \ (p1*p2) \ q1 \ q2 = fcompose \ p1 \ q1 \ q2 * fcompose \ p2 \ q1 \ q2$   
 $\langle \text{proof} \rangle$

**lemma**  $fcompose-poly$ :  
**assumes**  $\text{poly } q2 \ x \neq 0$

**shows**  $\text{poly } p (\text{poly } q1 \ x / \text{poly } q2 \ x) = \text{poly } (\text{fcompose } p \ q1 \ q2) \ x / \text{poly } (q2 \wedge (\text{degree } p)) \ x$   
 ⟨proof⟩

**lemma** *poly-fcompose*:

**assumes**  $\text{poly } q2 \ x \neq 0$

**shows**  $\text{poly } (\text{fcompose } p \ q1 \ q2) \ x = \text{poly } p (\text{poly } q1 \ x / \text{poly } q2 \ x) * (\text{poly } q2 \ x) \wedge (\text{degree } p)$   
 ⟨proof⟩

**lemma** *poly-fcompose-0-denominator*:

**assumes**  $\text{poly } q2 \ x = 0$

**shows**  $\text{poly } (\text{fcompose } p \ q1 \ q2) \ x = \text{poly } q1 \ x \wedge \text{degree } p * \text{lead-coeff } p$   
 ⟨proof⟩

**lemma** *fcompose-0-denominator*:  $\text{fcompose } p \ q1 \ 0 = \text{smult } (\text{lead-coeff } p) (q1 \wedge \text{degree } p)$   
 ⟨proof⟩

**lemma** *fcompose-nzero*:

**fixes**  $p::'a::\text{field } \text{poly}$

**assumes**  $p \neq 0$  **and**  $q2 \neq 0$  **and**  $nconst:\forall c. q1 \neq \text{smult } c \ q2$

**and**  $\text{infi}:\text{infinite } (\text{UNIV}::'a \ \text{set})$

**shows**  $\text{fcompose } p \ q1 \ q2 \neq 0$  ⟨proof⟩

## 1.7 Bijection (*bij-betw*) and the number of polynomial roots

**lemma** *proots-fcompose-bij-eq*:

**fixes**  $p::'a::\text{field } \text{poly}$

**assumes**  $\text{bij}:\text{bij-betw } (\lambda x. \text{poly } q1 \ x / \text{poly } q2 \ x) \ A \ B$  **and**  $p \neq 0$

**and**  $nzero:\forall x \in A. \text{poly } q2 \ x \neq 0$

**and**  $\text{max-deg}:\text{max } (\text{degree } q1) (\text{degree } q2) \leq 1$

**and**  $nconst:\forall c. q1 \neq \text{smult } c \ q2$

**and**  $\text{infi}:\text{infinite } (\text{UNIV}::'a \ \text{set})$

**shows**  $\text{proots-count } p \ B = \text{proots-count } (\text{fcompose } p \ q1 \ q2) \ A$   
 ⟨proof⟩

**lemma** *proots-card-fcompose-bij-eq*:

**fixes**  $p::'a::\text{field } \text{poly}$

**assumes**  $\text{bij}:\text{bij-betw } (\lambda x. \text{poly } q1 \ x / \text{poly } q2 \ x) \ A \ B$  **and**  $p \neq 0$

**and**  $nzero:\forall x \in A. \text{poly } q2 \ x \neq 0$

**and**  $\text{max-deg}:\text{max } (\text{degree } q1) (\text{degree } q2) \leq 1$

**and**  $nconst:\forall c. q1 \neq \text{smult } c \ q2$

**and**  $\text{infi}:\text{infinite } (\text{UNIV}::'a \ \text{set})$

**shows**  $\text{card } (\text{proots-within } p \ B) = \text{card } (\text{proots-within } (\text{fcompose } p \ q1 \ q2) \ A)$   
 ⟨proof⟩

**lemma** *proots-pcompose-bij-eq*:

**fixes**  $p::'a::\text{idom } \text{poly}$

**assumes**  $\text{bij}:\text{bij-betw } (\lambda x. \text{poly } q \ x) \ A \ B$  **and**  $p \neq 0$

**and** *q-deg*: degree  $q = 1$   
**shows** *roots-count*  $p B = \text{roots-count } (p \circ_p q) A$  *<proof>*

**lemma** *roots-card-pcompose-bij-eq*:

**fixes**  $p::'a::\text{idom poly}$

**assumes** *bij*:*bij-betw*  $(\lambda x. \text{poly } q x) A B$  **and**  $p \neq 0$

**and** *q-deg*: degree  $q = 1$

**shows**  $\text{card } (\text{roots-within } p B) = \text{card } (\text{roots-within } (p \circ_p q) A)$  *<proof>*

**end**

## 2 Budan–Fourier theorem

**theory** *Budan-Fourier* **imports**

*BF-Misc*

**begin**

The Budan–Fourier theorem is a classic result in real algebraic geometry to over-approximate real roots of a polynomial (counting multiplicity) within an interval. When all roots of the the polynomial are known to be real, the over-approximation becomes tight – the number of roots are counted exactly. Also note that Descartes’ rule of sign is a direct consequence of the Budan–Fourier theorem.

The proof mainly follows Theorem 2.35 in Basu, S., Pollack, R., Roy, M.-F.: Algorithms in Real Algebraic Geometry. Springer Berlin Heidelberg, Berlin, Heidelberg (2006).

### 2.1 More results related to *sign-r-pos*

**lemma** *sign-r-pos-nzero-right*:

**assumes** *nzero*: $\forall x. c < x \wedge x \leq d \longrightarrow \text{poly } p x \neq 0$  **and**  $c < d$

**shows** if *sign-r-pos*  $p c$  then  $\text{poly } p d > 0$  else  $\text{poly } p d < 0$

*<proof>*

**lemma** *sign-r-pos-at-left*:

**assumes**  $p \neq 0$

**shows** if *even*  $(\text{order } c p) \longleftrightarrow \text{sign-r-pos } p c$  then eventually  $(\lambda x. \text{poly } p x > 0)$  *(at-left c)*

else eventually  $(\lambda x. \text{poly } p x < 0)$  *(at-left c)*

*<proof>*

**lemma** *sign-r-pos-nzero-left*:

**assumes** *nzero*: $\forall x. d \leq x \wedge x < c \longrightarrow \text{poly } p x \neq 0$  **and**  $d < c$

**shows** if *even*  $(\text{order } c p) \longleftrightarrow \text{sign-r-pos } p c$  then  $\text{poly } p d > 0$  else  $\text{poly } p d < 0$

*<proof>*

## 2.2 Fourier sequences

**function** *pders*::*real poly*  $\Rightarrow$  *real poly list* **where**  
*pders* *p* = (if *p* = 0 then [] else Cons *p* (*pders* (*pderiv* *p*)))  
 ⟨*proof*⟩

**termination**  
 ⟨*proof*⟩

**declare** *pders.simps*[*simp del*]

**lemma** *set-pders-nzero*:  
**assumes** *p* ≠ 0 *q* ∈ *set* (*pders* *p*)  
**shows** *q* ≠ 0  
 ⟨*proof*⟩

## 2.3 Sign variations for Fourier sequences

**definition** *changes-itv-der*:: *real*  $\Rightarrow$  *real*  $\Rightarrow$  *real poly*  $\Rightarrow$  *int* **where**  
*changes-itv-der* *a* *b* *p* = (let *ps* = *pders* *p* in *changes-poly-at* *ps* *a* − *changes-poly-at* *ps* *b*)

**definition** *changes-gt-der*:: *real*  $\Rightarrow$  *real poly*  $\Rightarrow$  *int* **where**  
*changes-gt-der* *a* *p* = *changes-poly-at* (*pders* *p*) *a*

**definition** *changes-le-der*:: *real*  $\Rightarrow$  *real poly*  $\Rightarrow$  *int* **where**  
*changes-le-der* *b* *p* = (*degree* *p* − *changes-poly-at* (*pders* *p*) *b*)

**lemma** *changes-poly-pos-inf-pders*[*simp*]: *changes-poly-pos-inf* (*pders* *p*) = 0  
 ⟨*proof*⟩

**lemma** *changes-poly-neg-inf-pders*[*simp*]: *changes-poly-neg-inf* (*pders* *p*) = *degree* *p*  
 ⟨*proof*⟩

**lemma** *pders-coeffs-sgn-eq*: *map* ( $\lambda p. \text{sgn}(\text{poly } p \ 0)$ ) (*pders* *p*) = *map* *sgn* (*coeffs* *p*)  
 ⟨*proof*⟩

**lemma** *changes-poly-at-pders-0*: *changes-poly-at* (*pders* *p*) 0 = *changes* (*coeffs* *p*)  
 ⟨*proof*⟩

## 2.4 Budan–Fourier theorem

**lemma** *budan-fourier-aux-right*:  
**assumes** *c* < *d2* and *p* ≠ 0  
**assumes**  $\forall x. c < x \wedge x \leq d2 \longrightarrow (\forall q \in \text{set } (\text{pders } p). \text{poly } q \ x \neq 0)$   
**shows** *changes-itv-der* *c* *d2* *p* = 0  
 ⟨*proof*⟩

**lemma** *budan-fourier-aux-left'*:  
**assumes** *d1* < *c* and *p* ≠ 0

**assumes**  $\forall x. d1 \leq x \wedge x < c \longrightarrow (\forall q \in \text{set } (pders\ p). \text{poly } q\ x \neq 0)$   
**shows**  $\text{changes-itv-der } d1\ c\ p \geq \text{order } c\ p \wedge \text{even } (\text{changes-itv-der } d1\ c\ p - \text{order } c\ p)$   
 $\langle \text{proof} \rangle$

**lemma** *budan-fourier-aux-left*:

**assumes**  $d1 < c$  **and**  $p \neq 0$   
**assumes**  $nzero: \forall x. d1 < x \wedge x < c \longrightarrow (\forall q \in \text{set } (pders\ p). \text{poly } q\ x \neq 0)$   
**shows**  $\text{changes-itv-der } d1\ c\ p \geq \text{order } c\ p$  **even**  $(\text{changes-itv-der } d1\ c\ p - \text{order } c\ p)$   
 $\langle \text{proof} \rangle$

**theorem** *budan-fourier-interval*:

**assumes**  $a < b$   $p \neq 0$   
**shows**  $\text{changes-itv-der } a\ b\ p \geq \text{roots-count } p\ \{x. a < x \wedge x \leq b\} \wedge$   
**even**  $(\text{changes-itv-der } a\ b\ p - \text{roots-count } p\ \{x. a < x \wedge x \leq b\})$   
 $\langle \text{proof} \rangle$

**theorem** *budan-fourier-gt*:

**assumes**  $p \neq 0$   
**shows**  $\text{changes-gt-der } a\ p \geq \text{roots-count } p\ \{x. a < x\} \wedge$   
**even**  $(\text{changes-gt-der } a\ p - \text{roots-count } p\ \{x. a < x\})$   
 $\langle \text{proof} \rangle$

Descartes' rule of signs is a direct consequence of the Budan–Fourier theorem

**theorem** *descartes-sign*:

**fixes**  $p::\text{real poly}$   
**assumes**  $p \neq 0$   
**shows**  $\text{changes } (\text{coeffs } p) \geq \text{roots-count } p\ \{x. 0 < x\} \wedge$   
**even**  $(\text{changes } (\text{coeffs } p) - \text{roots-count } p\ \{x. 0 < x\})$   
 $\langle \text{proof} \rangle$

**theorem** *budan-fourier-le*:

**assumes**  $p \neq 0$   
**shows**  $\text{changes-le-der } b\ p \geq \text{roots-count } p\ \{x. x \leq b\} \wedge$   
**even**  $(\text{changes-le-der } b\ p - \text{roots-count } p\ \{x. x \leq b\})$   
 $\langle \text{proof} \rangle$

## 2.5 Count exactly when all roots are real

**definition** *all-roots-real*::  $\text{real poly} \Rightarrow \text{bool}$  **where**

$\text{all-roots-real } p = (\forall r \in \text{roots } (\text{map-poly of-real } p). \text{Im } r = 0)$

**lemma** *all-roots-real-mult[simp]*:

$\text{all-roots-real } (p * q) \longleftrightarrow \text{all-roots-real } p \wedge \text{all-roots-real } q$   
 $\langle \text{proof} \rangle$

**lemma** *all-roots-real-const-iff*:

**assumes** *all-real:all-roots-real* *p*  
**shows** *degree*  $p \neq 0 \iff (\exists x. \text{poly } p \ x = 0)$   
 $\langle \text{proof} \rangle$

**lemma** *all-roots-real-degree*:  
**assumes** *all-roots-real* *p*  
**shows** *roots-count* *p* *UNIV* = *degree* *p*  $\langle \text{proof} \rangle$

**lemma** *all-real-roots-mobius*:  
**fixes** *a b::real*  
**assumes** *all-roots-real* *p* **and**  $a < b$   
**shows** *all-roots-real* (*fcompose* *p*  $[:a,b:]$   $[:1,1:]$ )  $\langle \text{proof} \rangle$

If all roots are real, we can use the Budan–Fourier theorem to EXACTLY count the number of real roots.

**corollary** *budan-fourier-real*:  
**assumes**  $p \neq 0$   
**assumes** *all-roots-real* *p*  
**shows** *roots-count* *p*  $\{x. x \leq a\}$  = *changes-le-der* *a* *p*  
 $a < b \implies \text{roots-count } p \{x. a < x \wedge x \leq b\} = \text{changes-itv-der } a \ b \ p$   
 $\text{roots-count } p \{x. b < x\} = \text{changes-gt-der } b \ p$   
 $\langle \text{proof} \rangle$

Similarly, Descartes’ rule of sign counts exactly when all roots are real.

**corollary** *descartes-sign-real*:  
**fixes** *p::real poly* **and** *a b::real*  
**assumes**  $p \neq 0$   
**assumes** *all-roots-real* *p*  
**shows** *roots-count* *p*  $\{x. 0 < x\}$  = *changes* (*coeffs* *p*)  
 $\langle \text{proof} \rangle$

**end**

### 3 Extension of Sturm’s theorem for multiple roots

**theory** *Sturm-Multiple-Roots*  
**imports**  
*BF-Misc*  
**begin**

The classic Sturm’s theorem is used to count real roots WITHOUT multiplicity of a polynomial within an interval. Surprisingly, we can also extend Sturm’s theorem to count real roots WITH multiplicity by modifying the signed remainder sequence, which seems to be overlooked by many textbooks.

Our formal proof is inspired by Theorem 10.5.6 in Rahman, Q.I., Schmeisser, G.: Analytic Theory of Polynomials. Oxford University Press (2002).

### 3.1 More results for *smods*

**lemma** *last-smods-gcd*:  
**fixes**  $p\ q :: \text{real poly}$   
**defines**  $pp \equiv \text{last } (\text{smods } p\ q)$   
**assumes**  $p \neq 0$   
**shows**  $pp = \text{smult } (\text{lead-coeff } pp) (\text{gcd } p\ q)$   
 $\langle \text{proof} \rangle$

**lemma** *last-smods-nzero*:  
**assumes**  $p \neq 0$   
**shows**  $\text{last } (\text{smods } p\ q) \neq 0$   
 $\langle \text{proof} \rangle$

### 3.2 Alternative signed remainder sequences

**function** *smods-ext* ::  $\text{real poly} \Rightarrow \text{real poly} \Rightarrow \text{real poly list}$  **where**  
 $\text{smods-ext } p\ q = (\text{if } p=0 \text{ then } [] \text{ else}$   
 $\quad (\text{if } p \bmod q \neq 0$   
 $\quad \quad \text{then } \text{Cons } p (\text{smods-ext } q (- (p \bmod q)))$   
 $\quad \quad \text{else } \text{Cons } p (\text{smods-ext } q (pderiv\ q)))$   
 $\quad )$   
 $\langle \text{proof} \rangle$   
**termination**  
 $\langle \text{proof} \rangle$

**lemma** *smods-ext-prefix*:  
**fixes**  $p\ q :: \text{real poly}$   
**defines**  $pp \equiv \text{last } (\text{smods } p\ q)$   
**assumes**  $p \neq 0\ q \neq 0$   
**shows**  $\text{smods-ext } p\ q = \text{smods } p\ q @ \text{tl } (\text{smods-ext } pp (pderiv\ pp))$   
 $\langle \text{proof} \rangle$

**lemma** *no-0-in-smods-ext*:  $0 \notin \text{set } (\text{smods-ext } p\ q)$   
 $\langle \text{proof} \rangle$

### 3.3 Sign variations on the alternative signed remainder sequences

**definition** *changes-itv-smods-ext* ::  $\text{real} \Rightarrow \text{real} \Rightarrow \text{real poly} \Rightarrow \text{real poly} \Rightarrow \text{int}$   
**where**  
 $\text{changes-itv-smods-ext } a\ b\ p\ q = (\text{let } ps = \text{smods-ext } p\ q \text{ in changes-poly-at } ps\ a$   
 $\quad - \text{changes-poly-at } ps\ b)$

**definition** *changes-gt-smods-ext* ::  $\text{real} \Rightarrow \text{real poly} \Rightarrow \text{real poly} \Rightarrow \text{int}$  **where**  
 $\text{changes-gt-smods-ext } a\ p\ q = (\text{let } ps = \text{smods-ext } p\ q \text{ in changes-poly-at } ps\ a$   
 $\quad - \text{changes-poly-pos-inf } ps)$

**definition** *changes-le-smods-ext* ::  $\text{real} \Rightarrow \text{real poly} \Rightarrow \text{real poly} \Rightarrow \text{int}$  **where**  
 $\text{changes-le-smods-ext } b\ p\ q = (\text{let } ps = \text{smods-ext } p\ q \text{ in changes-poly-neg-inf } ps)$

– *changes-poly-at ps b*)

**definition** *changes-R-smods-ext*:: *real poly*  $\Rightarrow$  *real poly*  $\Rightarrow$  *int* **where**  
*changes-R-smods-ext p q* = (*let ps* = *smods-ext p q* in *changes-poly-neg-inf ps*  
– *changes-poly-pos-inf ps*)

### 3.4 Extension of Sturm’s theorem for multiple roots

**theorem** *sturm-ext-interval*:

**assumes** *a < b poly p a  $\neq$  0 poly p b  $\neq$  0*

**shows** *roots-count p {x. a < x  $\wedge$  x < b}* = *changes-itv-smods-ext a b p (pderiv p)*  
(*proof*)

**theorem** *sturm-ext-above*:

**assumes** *poly p a  $\neq$  0*

**shows** *roots-count p {x. a < x}* = *changes-gt-smods-ext a p (pderiv p)*  
(*proof*)

**theorem** *sturm-ext-below*:

**assumes** *poly p b  $\neq$  0*

**shows** *roots-count p {x. x < b}* = *changes-le-smods-ext b p (pderiv p)*  
(*proof*)

**theorem** *sturm-ext-R*:

**assumes** *p  $\neq$  0*

**shows** *roots-count p UNIV* = *changes-R-smods-ext p (pderiv p)*  
(*proof*)

**end**

## 4 Descartes Roots Test

**theory** *Descartes-Roots-Test* **imports** *Budan-Fourier*  
**begin**

The Descartes roots test is a consequence of Descartes’ rule of signs: through counting sign variations on coefficients of a base-transformed (i.e. Taylor shifted) polynomial, it can over-approximate the number of real roots (counting multiplicity) within an interval. Its ability is similar to the Budan–Fourier theorem, but is far more efficient in practice. Therefore, this test is widely used in modern root isolation procedures.

More information can be found in the wiki page about Vincent’s theorem: [https://en.wikipedia.org/wiki/Vincent%27s\\_theorem](https://en.wikipedia.org/wiki/Vincent%27s_theorem) and Collins and Akritas’s classic paper of root isolation: Collins, G.E., Akritas, A.G.: Polynomial real root isolation using Descarte’s rule of signs. SYMSACC. 272–275 (1976). A more modern treatment is available from a recent implementation of isolating real roots: Kobel, A., Rouillier, F., Sagraloff, M.: Computing Real Roots of Real Polynomials ... and now For Real! Proceedings of ISSAC

'16, New York, New York, USA (2016).

**lemma** *bij-betw-pos-interval*:

**fixes**  $a b::\text{real}$

**assumes**  $a < b$

**shows**  $\text{bij-betw } (\lambda x. (a+b * x) / (1+x)) \{x. x > 0\} \{x. a < x \wedge x < b\}$

*<proof>*

**lemma** *proots-sphere-pos-interval*:

**fixes**  $a b::\text{real}$

**defines**  $q1 \equiv [:a, b:]$  **and**  $q2 \equiv [:1, 1:]$

**assumes**  $p \neq 0$   $a < b$

**shows**  $\text{proots-count } p \{x. a < x \wedge x < b\} = \text{proots-count } (\text{fcompose } p \ q1 \ q2) \{x. 0 < x\}$

*<proof>*

**definition** *descartes-roots-test::real  $\Rightarrow$  real  $\Rightarrow$  real poly  $\Rightarrow$  nat* **where**

*descartes-roots-test*  $a \ b \ p = \text{nat } (\text{changes } (\text{coeffs } (\text{fcompose } p \ [:a, b:] \ [:1, 1:])))$

**theorem** *descartes-roots-test*:

**fixes**  $p::\text{real poly}$

**assumes**  $p \neq 0$   $a < b$

**shows**  $\text{proots-count } p \{x. a < x \wedge x < b\} \leq \text{descartes-roots-test } a \ b \ p \wedge$   
*even*  $(\text{descartes-roots-test } a \ b \ p - \text{proots-count } p \{x. a < x \wedge x < b\})$

*<proof>*

The roots test *descartes-roots-test* is exact if its result is 0 or 1.

**corollary** *descartes-roots-test-zero*:

**fixes**  $p::\text{real poly}$

**assumes**  $p \neq 0$   $a < b$   $\text{descartes-roots-test } a \ b \ p = 0$

**shows**  $\forall x. a < x \wedge x < b \longrightarrow \text{poly } p \ x \neq 0$

*<proof>*

**corollary** *descartes-roots-test-one*:

**fixes**  $p::\text{real poly}$

**assumes**  $p \neq 0$   $a < b$   $\text{descartes-roots-test } a \ b \ p = 1$

**shows**  $\text{proots-count } p \{x. a < x \wedge x < b\} = 1$

*<proof>*

Similar to the Budan–Fourier theorem, the Descartes roots test result is exact when all roots are real.

**corollary** *descartes-roots-test-real*:

**fixes**  $p::\text{real poly}$

**assumes**  $p \neq 0$   $a < b$

**assumes** *all-roots-real*  $p$

**shows**  $\text{proots-count } p \{x. a < x \wedge x < b\} = \text{descartes-roots-test } a \ b \ p$

*<proof>*

**end**

## 5 Acknowledgements

The work was supported by the ERC Advanced Grant ALEXANDRIA (Project 742178), funded by the European Research Council and led by Professor Lawrence Paulson at the University of Cambridge, UK.

## References

- [1] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [2] M. Eberl. Sturm’s theorem. *Archive of Formal Proofs*, Jan. 2014. [http://isa-afp.org/entries/Sturm\\_Sequences.html](http://isa-afp.org/entries/Sturm_Sequences.html), Formal proof development.
- [3] M. Eberl. Descartes’ rule of signs. *Archive of Formal Proofs*, Dec. 2015. [http://isa-afp.org/entries/Descartes\\_Sign\\_Rule.html](http://isa-afp.org/entries/Descartes_Sign_Rule.html), Formal proof development.
- [4] J. Harrison. Verifying the accuracy of polynomial approximations in HOL. In E. L. Gunter and A. Felty, editors, *Theorem Proving in Higher Order Logics: 10th International Conference, TPHOLs’97*, volume 1275 of *Lecture Notes in Computer Science*, pages 137–152, Murray Hill, NJ, 1997. Springer-Verlag.
- [5] W. Li. The Sturm–Tarski Theorem. *Archive of Formal Proofs*, Sept. 2014.
- [6] W. Li. Count the Number of Complex Roots. *Archive of Formal Proofs*, Oct. 2017.
- [7] W. Li and L. C. Paulson. Evaluating Winding Numbers and Counting Complex Roots through Cauchy Indices in Isabelle/HOL. *CoRR*, abs/1804.03922, 2018.
- [8] A. Mahboubi and C. Cohen. Formal proofs in real algebraic geometry: from ordered fields to quantifier elimination. *Logical Methods in Computer Science*, 8(1), 2012.
- [9] A. Narkawicz, C. A. Muñoz, and A. Dutle. Formally-Verified Decision Procedures for Univariate Polynomial Computation Based on Sturm’s and Tarski’s Theorems. *Journal of Automated Reasoning*, 54(4):285–326, 2015.
- [10] Q. I. Rahman and G. Schmeisser. *Analytic Theory of Polynomials*. Oxford University Press, 2002.