The Budan-Fourier Theorem and Counting Real Roots with Multiplicity

Wenda Li

September 4, 2018

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 ProofPower\(^1\)). 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].

\(^1\)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 Misc

lemma lead-coeff-pderiv:
  fixes p :: 'a::{comm-semiring-1, semiring-no-zero-divisors, semiring-char-0} poly
  shows lead-coeff (pderiv p) = of-nat (degree p) * lead-coeff p
⟨proof⟩

lemma gcd-degree-le-min:
  assumes p ≠ 0 q ≠ 0
  shows degree (gcd p q) ≤ min (degree p) (degree q)
⟨proof⟩

lemma lead-coeff-normalize-field:
  fixes p::'a::{field, semidom-divide-unit-factor} poly
  assumes p ≠ 0
  shows lead-coeff (normalize p) = 1
⟨proof⟩

lemma smult-normalize-field-eq:
  fixes p::'a::{field, semidom-divide-unit-factor} poly
  shows p = smult (lead-coeff p) (normalize p)
⟨proof⟩

lemma lead-coeff-gcd-field:
  fixes p q::'a::{field, semidom-divide-unit-factor, factorial-ring-gcd} poly
  assumes p ≠ 0 ∨ q ≠ 0
  shows lead-coeff (gcd p q) = 1
⟨proof⟩

lemma poly-gcd-0-iff:
  poly (gcd p q) x = 0 ←→ poly p x=0 ∧ poly q x=0
⟨proof⟩

lemma order-multiplicity-eq:
  assumes p ≠ 0
  shows order a p = multiplicity [-a,1:] p
⟨proof⟩

lemma order-gcd:
  assumes p ≠ 0 q ≠ 0
  shows order x (gcd p q) = min (order x p) (order x q)
⟨proof⟩
1.2 More results about sign variations (i.e. changes)

**Lemma changes-0**: \( \text{changes (0#xs)} = \text{changes xs} \)

**Proof**

**Lemma changes-Cons**: \( \text{changes (x#xs)} = (\text{if filter } (\lambda x. x\neq 0) \text{ xs }\{\} \text{ then 0 else if x* hd (filter } (\lambda x. x\neq 0) \text{ xs} < 0 \text{ then 1 + changes xs else changes xs}) \)

**Proof**

**Lemma changes-filter-eq**: \( \text{changes (filter } (\lambda x. x\neq 0) \text{ xs) = changes xs} \)

**Proof**

**Lemma changes-filter-empty**: \( \text{assumes filter } (\lambda x. x\neq 0) \text{ xs }\{\} \text{ shows changes xs } = 0 \text{ changes (a#xs) } = 0 \)

**Proof**

**Lemma changes-append**: \( \text{assumes xs }\{\} \land ys \{\} \rightarrow (\text{last xs } = \text{hd ys} \land \text{last xs}\neq 0) \text{ shows changes (xs@ys) = changes xs + changes ys} \)

**Proof**

**Lemma changes-drop-dup**: \( \text{assumes xs }\{\} \land ys \{\} \rightarrow \text{last xs = hd ys} \text{ shows changes (xs@ys) = changes (xs@ tl ys)} \)

**Proof**

1.3 Induction on polynomial roots

**Lemma poly-root-induct-alt [case-names 0 no-proots root]**:

**Proof**

1.4 Polynomial roots / zeros

**Definition proots-within**: \( \text{proots-within } p \text{:: comm-semiring-0 poly } \Rightarrow 'a \text{ set } \Rightarrow 'a \text{ set where} \)

**Proof**

**Abbreviation proots**: \( \text{proots } p \text{:: comm-semiring-0 poly } \Rightarrow 'a \text{ set } \Rightarrow 'a \text{ set where} \)

**Proof**

**Lemma proots-def**: \( \text{proots } p = \{ x. \text{ poly p } x=0 \} \text{)
lemma proots-within-empty [simp]:
proots-within p {} = {} ⟨proof⟩

lemma proots-within-0 [simp]:
proots-within 0 s = s ⟨proof⟩

lemma proots-withinI [intro, simp]:
poly p x = 0 ⇒ x ∈ s ⇒ x ∈ proots-within p s ⟨proof⟩

lemma proots-within-iff [simp]:
x ∈ proots-within p s ←→ poly p x = 0 ∧ x ∈ s ⟨proof⟩

lemma proots-within-union:
proots-within p A ∪ proots-within p B = proots-within p (A ∪ B) ⟨proof⟩

lemma proots-within-times:
fixes s :: 'a :: {semiring-no-zero-divisors, comm-semiring-0} set
shows proots-within (p * q) s = proots-within p s ∪ proots-within q s ⟨proof⟩

lemma proots-within-gcd:
fixes s :: 'a :: factorial-ring-gcd set
shows proots-within (gcd p q) s = proots-within p s ∩ proots-within q s ⟨proof⟩

lemma proots-within-inter:
NO-MATCH UNIV s ⇒ proots-within p s = proots p ∩ s ⟨proof⟩

lemma proots-within-proots [simp]:
proots-within p s ⊆ proots p ⟨proof⟩

lemma finite-proots [simp]:
fixes p :: 'a :: idom poly
shows p ≠ 0 ⇒ finite (proots-within p s) ⟨proof⟩

lemma proots-within-pCons-1-iff:
fixes a :: 'a :: idom
shows proots-within [−a, 1:] s = (if a ∈ s then {a} else {})
proots-within [a, −1:] s = (if a ∈ s then {a} else {}) ⟨proof⟩

lemma proots-within-uminus [simp]:

4
fixes $p :: 'a::comm-ring poly$
shows $\text{proots-within} (-p) s = \text{proots-within} p s$
⟨proof⟩

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

1.5 Polynomial roots counting multiplicities.

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

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

lemma $\text{proots-count-times}$:
fixes $s :: 'a::idom set$
assumes $p \ast q \neq 0$
shows $\text{proots-count} (p \ast q) s = \text{proots-count} p s + \text{proots-count} q s$
⟨proof⟩

lemma $\text{proots-count-power-n-n}$:
  shows $\text{proots-count} \left[\begin{array}{c} -a \\ 1 \end{array}\right]^n s = (\text{if } a \in s \land n > 0 \text{ then } n \text{ else } 0)$
⟨proof⟩

lemma $\text{degree-proots-count}$:
  fixes $p :: \text{complex poly}$
  shows $\text{degree} p = \text{proots-count} p \text{ UNIV}$
⟨proof⟩

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

lemma $\text{proots-count-pCons-1-iff}$:
  fixes $a :: \text{idom}$
  shows $\text{proots-count} \left[\begin{array}{c} -a \\ 1 \end{array}\right] s = (\text{if } a \in s \text{ then } 1 \text{ else } 0)$
⟨proof⟩

lemma $\text{proots-count-uminus}$ [simp]:
  $\text{proots-count} (-p) s = \text{proots-count} p s$
⟨proof⟩
lemma card-proots-within-leq:
  assumes p ≠ 0
  shows proots-count p s ≥ card (proots-within p s) ⟨proof⟩

lemma proots-count-leq-degree:
  assumes p ≠ 0
  shows proots-count p s ≤ degree p ⟨proof⟩

lemma proots-count-union-disjoint:
  assumes A ∩ B = {} p ≠ 0
  shows proots-count p (A ∪ B) = proots-count p A + proots-count p B ⟨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 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.


2.1 More results related to sign-r-pos

lemma sign-r-pos-nzero-right:
  assumes nzero:∀ x. c < x ∧ x ≤ d ⟹ poly p x ≠ 0 and c < d
  shows if sign-r-pos p c then poly p d > 0 else poly p d < 0 ⟨proof⟩

lemma sign-r-pos-at-left:
  assumes p ≠ 0
  shows if even (order c p) ⟷ sign-r-pos p c then eventually (λx. poly p x > 0)
  (at-left c) else eventually (λx. poly p x < 0) (at-left c) ⟨proof⟩

lemma sign-r-pos-nzero-left:
  assumes nzero:∀ x. d ≤ x ∧ x < c ⟹ poly p x ≠ 0 and d < c
shows if even (order c p) \(\longleftrightarrow\) sign-r-pos p c then poly p d \(\geq\) 0 else poly p d \(\leq\) 0
⟨proof⟩

### 2.2 Fourier sequences

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

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

**lemma** set-pders-nzero:
  assumes \(p \neq 0\) \(q \in \text{set}(\text{pders} p)\)
  shows \(q \neq 0\)
⟨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 (λp. sgn (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 \(c < d^2\) and \(p \neq 0\)
  assumes \(\forall x. c < x \land x \leq d^2 \rightarrow (\forall q \in \text{set}(\text{pders} p). \text{poly} \ q \ x \neq 0)\)
  shows changes-itv-der c d^2 p = 0
lemma budan-fourier-aux-left:\n  \textbf{assumes} \ d_1 < c \ \textbf{and} \ p \not= 0
  \textbf{assumes} \ \forall x. \ d_1 \leq x \land x < c \ \rightarrow \ (\forall q \in \text{set \ (pders \ p)}. \ poly \ q \ x \not= 0)
  \textbf{shows} \ \text{changes-ivt-der} \ d_1 \ c \ p \ \geq \ \text{order} \ c \ p \ \land \ \text{even} \ (\text{changes-ivt-der} \ d_1 \ c \ p \ - \ \text{order} \ c \ p)
\langle \text{proof} \rangle

lemma budan-fourier-aux-left:\n  \textbf{assumes} \ d_1 < c \ \textbf{and} \ p \not= 0
  \textbf{assumes} \ nzero: \ \forall x. \ d_1 < x \land x < c \ \rightarrow \ (\forall q \in \text{set \ (pders \ p)}. \ poly \ q \ x \not= 0)
  \textbf{shows} \ \text{changes-ivt-der} \ d_1 \ c \ p \ \geq \ \text{order} \ c \ p \ \land \ \text{even} \ (\text{changes-ivt-der} \ d_1 \ c \ p \ - \ \text{order} \ c \ p)
\langle \text{proof} \rangle

theorem budan-fourier-interval:\n  \textbf{assumes} \ a < b \ p \not= 0
  \textbf{shows} \ \text{changes-ivt-der} \ a \ b \ p \ \geq \ \text{proots-count} \ \{x. \ a < x \land x \leq b\} \ \land \ \text{even} \ (\text{changes-ivt-der} \ a \ b \ p \ - \ \text{proots-count} \ \{x. \ a < x \land x \leq b\})
\langle \text{proof} \rangle

theorem budan-fourier-gt:\n  \textbf{assumes} \ p \not= 0
  \textbf{shows} \ \text{changes-gt-der} \ a \ p \ \geq \ \text{proots-count} \ \{x. \ a < x\} \ \land \ \text{even} \ (\text{changes-gt-der} \ a \ p \ - \ \text{proots-count} \ \{x. \ a < x\})
\langle \text{proof} \rangle

Descartes’ rule of signs is a direct consequence of the Budan-Fourier theorem.

\textbf{theorem} \ \text{descartes-sign}:
  \textbf{fixes} \ p::\text{real \ poly}
  \textbf{assumes} \ p \not= 0
  \textbf{shows} \ \text{changes} \ (\text{coeffs} \ p) \ \geq \ \text{proots-count} \ \{x. \ 0 < x\} \ \land \ \text{even} \ (\text{changes} \ (\text{coeffs} \ p) \ - \ \text{proots-count} \ \{x. \ 0 < x\})
\langle \text{proof} \rangle

\textbf{theorem} \ \text{budan-fourier-le}:
  \textbf{assumes} \ p \not= 0
  \textbf{shows} \ \text{changes-le-der} \ b \ p \ \geq \ \text{proots-count} \ \{x. \ x \leq b\} \ \land \ \text{even} \ (\text{changes-le-der} \ b \ p \ - \ \text{proots-count} \ \{x. \ x \leq b\})
\langle \text{proof} \rangle

If we knowing that all roots of a polynomial are real, we can use the Budan-Fourier theorem to EXACTLY count the number of real roots.

\textbf{corollary} \ \text{budan-fourier-real}:
  \textbf{assumes} \ p \not= 0 \ a < b
  \textbf{assumes} \ \text{proots-deg}: \ \text{proots-count} \ \text{p \ UNIV} \ = \text{degree} \ p \ — \ \text{All \ of} \ p’S \ \text{roots \ are \ real.}
  \textbf{shows} \ \text{proots-count} \ \{x. \ x \leq a\} = \text{changes-le-der} \ a \ p
\begin{align*}
\text{proots-count } p \{ x. \ a < x \land x \leq b \} &= \text{changes-ite-der } a \ b \ p \\
\text{proots-count } p \{ x. \ b < x \} &= \text{changes-gt-der } b \ p
\end{align*}

\begin{proof}
\end{proof}

\section{Extension of Sturm’s theorem for multiple roots}

\begin{proof}
\end{proof}

3.1 More results for \textit{smods}

\begin{proof}
\end{proof}

3.2 Alternative signed remainder sequences

\begin{proof}
\end{proof}

\begin{proof}
\end{proof}

\begin{proof}
\end{proof}

\begin{proof}
\end{proof}

\begin{proof}
\end{proof}

\begin{proof}
\end{proof}

\begin{proof}
\end{proof}

\begin{proof}
\end{proof}

\begin{proof}
\end{proof}
fixes $p$, $q :: \text{real poly}$
defines $\text{pp} \equiv \text{last} \ (\text{smods } p \ q)$
assumes $p \neq 0 \ q \neq 0$
shows $\text{smods-ext } p \ q = \text{smods } p \ q \ @ \ \text{tl} \ (\text{smods-ext pp } \ (\text{pderiv pp}))$

(\text{proof})

\begin{itemize}
  \item \textbf{lemma} \text{no-0-in-smods-ext:} $0 \notin \text{set} \ (\text{smods-ext } p \ q)$
  \item (\text{proof})
\end{itemize}

\subsection*{3.3 Sign variations on the alternative signed remainder sequences}

\begin{itemize}
  \item \textbf{definition} \text{changes-itv-smods-ext::} \text{real} \Rightarrow \text{real poly} \Rightarrow \text{real poly} \Rightarrow \text{int}
    \item where
    \begin{align*}
      \text{changes-itv-smods-ext } a \ b \ p \ q &= (\text{let } \text{ps} = \text{smods-ext } p \ q \ \text{in} \ \text{changes-poly-at ps } a \\
      &- \ \text{changes-poly-at ps } b)
    \end{align*}
  \item (\text{proof})
\end{itemize}

\begin{itemize}
  \item \textbf{definition} \text{changes-gt-smods-ext::} \text{real} \Rightarrow \text{real poly} \Rightarrow \text{real poly} \Rightarrow \text{int}
    \item where
    \begin{align*}
      \text{changes-gt-smods-ext } a \ p \ q &= (\text{let } \text{ps} = \text{smods-ext } p \ q \ \text{in} \ \text{changes-poly-pos-inf ps} \\
      &- \ \text{changes-poly-at ps } b)
    \end{align*}
  \item (\text{proof})
\end{itemize}

\begin{itemize}
  \item \textbf{definition} \text{changes-le-smods-ext::} \text{real} \Rightarrow \text{real poly} \Rightarrow \text{real poly} \Rightarrow \text{int}
    \item where
    \begin{align*}
      \text{changes-le-smods-ext } b \ p \ q &= (\text{let } \text{ps} = \text{smods-ext } p \ q \ \text{in} \ \text{changes-poly-neg-inf ps} \\
      &- \ \text{changes-poly-at ps } b)
    \end{align*}
  \item (\text{proof})
\end{itemize}

\begin{itemize}
  \item \textbf{definition} \text{changes-R-smods-ext::} \text{real poly} \Rightarrow \text{real poly} \Rightarrow \text{int}
    \item where
    \begin{align*}
      \text{changes-R-smods-ext } p \ q &= (\text{let } \text{ps} = \text{smods-ext } p \ q \ \text{in} \ \text{changes-poly-neg-inf ps} \\
      &- \ \text{changes-poly-pos-inf ps})
    \end{align*}
  \item (\text{proof})
\end{itemize}

\subsection*{3.4 Extension of Sturm’s theorem for multiple roots}

\begin{itemize}
  \item \textbf{theorem} \text{sturm-ext-interval:}
    \item assumes $a < b$ \text{poly } p \ a \neq 0 \ \text{poly } p \ b \neq 0$
    \item shows \text{proots-count } p \ \{x. \ a < x \land x < b\} = \text{changes-itv-smods-ext } a \ b \ p \ (\text{pderiv } p)$
    \item (\text{proof})
  \item (\text{proof})
\end{itemize}

\begin{itemize}
  \item \textbf{theorem} \text{sturm-ext-above:}
    \item assumes \text{poly } p \ a \neq 0$
    \item shows \text{proots-count } p \ \{x. \ a < x\} = \text{changes-gt-smods-ext } a \ p \ (\text{pderiv } p)$
    \item (\text{proof})
  \item (\text{proof})
\end{itemize}

\begin{itemize}
  \item \textbf{theorem} \text{sturm-ext-below:}
    \item assumes \text{poly } p \ b \neq 0$
    \item shows \text{proots-count } p \ \{x. \ x < b\} = \text{changes-le-smods-ext } b \ p \ (\text{pderiv } p)$
    \item (\text{proof})
  \item (\text{proof})
\end{itemize}

\begin{itemize}
  \item \textbf{theorem} \text{sturm-ext-R:}
    \item assumes $p \neq 0$
    \item shows \text{proots-count } p \ \text{UNIV} = \text{changes-R-smods-ext } p \ (\text{pderiv } p)$
    \item (\text{proof})
\end{itemize}
proof

end

References


