A formal study of Bernstein coefficients and polynomials

Yves Bertot, Guilhot Frédérique, Assia Mahboubi

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A formal study of Bernstein coefficients and polynomials

Yves Bertot — Fréderique Guilhot — Assia Mahboubi

N° ????

July 2010

Domaine 2

ISSN 0249-6399 ISRN INRIA/RR--????--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SACLAY - ÎLE-DE-FRANCE

A formal study of Bernstein coefficients and polynomials

Yves Bertot*†, Fréderique Guilhot , Assia Mahboubi †

Domaine : Algorithmique, programmation, logiciels et architectures
Équipes-Projets Marelle , TypiCal

Rapport de recherche n° ???? — July 2010 — 31 pages

Abstract: Bernstein coefficients provide a discrete approximation of the behavior of a polynomial inside an interval. This can be used for example to isolate real roots of polynomials. We prove a criterion for the existence of a single root in an interval and the correctness of the de Casteljau algorithm to compute efficiently Bernstein coefficients.

Key-words: Polynomials, de Casteljau, Bernstein polynomials, Coq, Constructive mathematics, Real root isolation

* This work has been partially funded by the Galapagos project, of the French ANR.
† This work has been partially funded by the FORMATH project, nr. 243847, of the FET program within the 7th Framework program of the European Commission

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

A formal study of Bernstein coefficients and polynomials

Résumé : Les coefficients de Bernstein fournissent une approximation discr‘ete du comportement d'un polynôme sur un intervalle donné. Ils peuvent être par exemple utilisés pour isoler les racines réelles d'un polynôme. Nous prouvons un critère suffisant à l'existence d'une unique racine dans un intervalle ainsi que la correction de l'algorithme de de Casteljau pour calculer efficacement les coefficients de Bernstein.

Mots-clés : Polynômes, de Casteljau, polynômes de Bernstein, Coq, Mathématiques constructives, isolation de racines réelles

1 Introduction

Bernstein coefficients provide a discrete approximation of polynomials inside a bounded interval. As such they are useful tools to solve problems like locating the roots of polynomials, isolating these roots or solving systems of inequations with polynomial members. In computer aided design, they are also used intensively as they give an efficient tool to draw curves that are controlled by points that users can grab and drag, with instantaneous and intuitive feedback on the shape the curve will take as the control points move around. Bernstein coefficients are closely related to splines and Bezier curves and they have a very simple geometrical interpretation, which we illustrate in section 2.

Bernstein coefficients are defined for a given polynomial, a given degree, and a given interval. If the degree is n, then the coefficients form a sequence of size $n+1$. In this paper, we are interested in three important properties of these coefficients:

1. if the coefficients taken in order exhibit exactly one sign change, then the polynomial is guaranteed to have exactly one root inside the interval.

2. if all coefficients have the same sign, then the polynomial is guaranteed to have no root inside the interval for which they have been computed.

3. there is an easy method to compute Bernstein coefficients for the two intervals obtained when splitting a larger interval from the Bernstein coefficients for this larger interval.

We describe a formal proof of these three properties, concentrating on the second and third properties. These proofs will be done in the setting of polynomials with the rational coefficients and rational values.

In the following, we will assume that we are working with polynomials whose roots are all simple, called separable polynomials. Starting from an arbitrary polynomial it is easy to produce a separable polynomial with the same roots by computing the greatest common divisor of this polynomial and its derivative.

The main plan of the proof of the first property is to describe a sequence of pairs (I_0, P_0) to (I_3, P_3), each pair containing an interval and a polynomial, so that every root of polynomial P_i inside I_i is in bijective correspondance with a root of polynomial P_{i+1} in the interval I_{i+1}. If we study the roots of the polynomial P on the interval (l, r), then I_0 and P_0 are respectively (l, r) and P. The last interval I_3 is $(0, +\infty)$ and the last polynomial P_3 is $c_0+c_1X+\cdots+c_nX^n$, where the coefficients c_i have the same sign as the Bernstein coefficients. Going from P_i to P_{i+1} we apply a given transformation. The first transformation is a change of variable so that I_1 is $(0, 1)$ and $P_1(x) = P_0(x\times r+(1-x)l)$. The second transformation is so that I_2 is $(1, +\infty)$ and $P_2(x) = 0$ exactly when $P_1(1/x) = 0$, as long as $x \neq 0$. The third transformation is a translation so that $I_3 = (0, +\infty)$ and $P_3(x) = P_2(1+x)$. We will show that the condition on Bernstein coefficients simply boils down to Descartes' law of sign [Des69, BPR06] for polynomial P_3 in the case where there is exactly one sign change in this polynomial's coefficients. This path from one polynomial to another is described in section 5.

Descartes' law of signs provides a sufficient criterion for the existence of exactly one root for a polynomial between 0 and $+\infty$. In its most general form, this law expresses a relation between the number of roots of a polynomial

between 0 and $+\infty$ and the number of sign changes in the coefficients of this polynomial. The number of sign changes is larger than the number of roots and the difference between the two numbers is a multiple of 2. Thus, if the number of sign changes is 1, there is exactly one root between 0 and $+\infty$.

For our development, we only prove the corollary of Descartes' law of signs for the case where there is only one sign change. Expressing Descartes' law on the coefficients of polynomial P_3 yields directly a law expressed in terms of sign changes for Bernstein's coefficients of P with respect to the interval (l, r). This proof is done in section 4.

Another part of our work is to describe dichotomy. Knowing Bernstein coefficients for a polynomial and a given interval, it is easy to obtain the Bernstein coefficients for the two half intervals, using an algorithm due to de Casteljau [dC85]. In the process, we increase the precision of the approximation given by the Bernstein coefficients. De Casteljau's algorithm is a simple combinatorial algorithm based on taking arithmetic means of successive coefficients. To justify this combinatorial process we show in section 6 that Bernstein coefficients actually are the coefficients of the polynomial in a different basis from the usual monomials, called the *Bernstein basis*.

Most of our proofs were made using only rational numbers as numeric values. Thus, we work with a type of numbers where equality and comparison are decidable and the process we describe can effectively be used in a decision procedure.

When considering only rational numbers, the existence of roots takes a different meaning: if a polynomial has a single simple real root in an interval, this root may not be rational. However, we can use a corresponding property on rational numbers: there exists a sub-interval inside which the absolute value of the slope is bounded below, and such that the values of the polynomial at the sub-interval bounds have opposite signs. In a similar vein, the intermediate value theorem does not hold with rational numbers, but a corresponding statement, expressed as a bounded-value property, does. Our proof development relies on this approach. We describe the formal aspects of this approach to describing roots in section 3.

Bernstein coefficients provide an important stepping stone to address various aspects of real algebraic numbers, decision procedures for real arithmetic, and more ambitious algorithms like cylindrical decomposition [BPR06, Mah07].

The formal work described in this paper has been performed using the Coq system[BC04], with SSReflect extension [GM08]. We think some characteristics of the proof system played a key role in making this development possible. We describe these key aspects in section 7.

2 Bernstein coefficients

Bézier curves [Béz86] are parametric curves that are widely used to construct smooth plane curves whose shape are governed by a finite finite number of *control points*. A Bézier curve controlled by $n+1$ points is a polynomial expression of degree n in its parameter t. For instance, given two points P_0 and P_1, the corresponding Bézier curve is the segment $B_{(P_0,P1)}(t) = tP_0 + (1-t)P_1$. For three control points P_0, P_1, P_3, the Bézier curve is $B_{(P_0,P_1,P_2)}(t) = (1-t)^2P_0 + 2(1-t)tP_1 + t^3P_2$. We already see in this case that a Bézier curve does not

meet all its control points. In fact, it is only guaranteed to pass through the first and the last control point. In the case of the quadratic Bézier curve, the middle control point P_1 is the intersection of the tangents to the curve at P_0 and P_2. The general formula giving the Bézier curve at n control points is:

$$B_{(P_0,\ldots,P_n}(t) = \sum_{k=0}^{n} \binom{n}{k} (1-t)^{n-k} t^k P_i$$

which satisfies the recursive relation:

$$B_{(P_0,\ldots,P_n)} = (1,t) B_{(P_0,\ldots,P_{n-1})} + t B_{(P_1,\ldots,P_n)}$$

Bézier curves are named after the engineer Paul Bézier who was working on coachbuilding. These curves have very interesting properties for interpolation but also for computer graphics: a Bézier curve is contained in the convex hull of its control points and uniform transformations on the control points like translation or rotation have the same effect on the curve. Points control the shape of the curve since the k-th derivative of the curve at its extreme points is governed by the $k+1$ nearest control points.

Computer graphics algorithms usually use piecewise polynomial paths (called splines) of low degree. Most modern vector graphics standards, like for instance SVG, feature support for Bézier splines. TrueType fonts use quadratic Bézier splines, whereas PostScript or MetaFont [Knu86] use cubic Bézier splines.

Bernstein polynomials are defined as the weight assigned to each control point: the k-th Bernstein polynomial $P_b(n,k)$ is defined by:

$$B_{(P_0,\ldots,P_n)}(t) = \sum_{k=0}^{n} P_b(n,k)(t) \binom{n}{k} P_i$$

hence:

$$P_b(n,k)(t) = \binom{n}{k} (1-t)^{n-k} t^k$$

For arbitrary numbers l and r, we can also consider the following polynomials, called the *Bernstein polynomials for degree n and the interval (l,r)* for $0 \leq k \leq n$

$$P_b(n,l,r,k) = \binom{n}{k} \frac{(x-l)^{n-k}(r-x)^k}{r-l}^n .$$

These polynomials also constitue a basis of the vector space of polynomial of degree n, and we will usually call it the *Bernstein basis* leaving the degree and the values l and r unspecified. Every polynomial p hence has a sequence of coefficients b_i, so that $p(x) = \sum_{i=0}^{n} P_b(n,l,r,i)(x)$. The coefficients b_i are the Bernstein coefficients.

When $l < r$, the Bernstein polynomials are positive and each polynomial of index k reaches its maximum at the point $d_k = l + \frac{(r-l)k}{n}$, so that each coefficient b_k somehow has a dominant influence on the value of the polynomial around d_k. Moreover, the coefficients $\binom{n}{k}$ included in the definition of $P_b(n,k,r,k)$ are chosen in such a way that the coefficient b_k would tend to have a value close to the value of the polynomial in d_k. For instance, if the p_1 is the constant polynomial with value 1, then all its Bernstein coefficients are equal to 1; similarly, if p_2 is

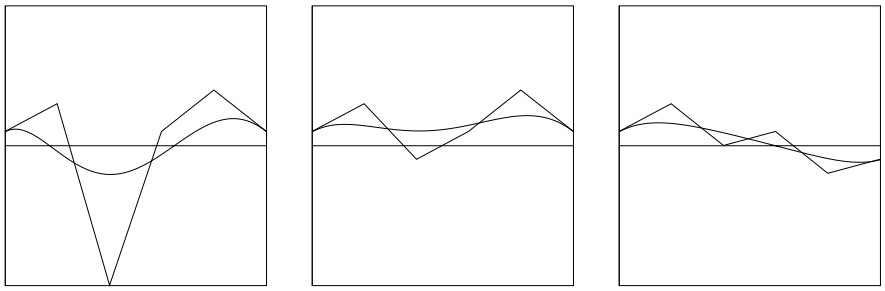
the identity polynomial, and n is larger than 0, then the Bernstein coefficients for p_2 are $l + \frac{(r-l)k}{n}$, as can be verified using the following computation:

$$\begin{aligned}
&\sum_{i=0}^{n} \left(l + \frac{(r-l)i}{n}\right) \binom{n}{i} \frac{(x-l)^i (r-x)^{n-i}}{(r-l)^n} \\
&= \sum_{i=0}^{n} l \binom{n}{i} \frac{(x-l)^i (r-x)^{n-i}}{(r-l)^n} + \sum_{i=0}^{n} \frac{i}{n} \binom{n}{i} \frac{(x-l)^i (r-x)^{n-i}}{(r-l)^{n-1}} \\
&= l \frac{((x-l)+(r-x))^n}{(r-l)^n} + \sum_{i=1}^{n} \binom{n-1}{i-1} \frac{(x-l)^i (r-x)^{n-i}}{(r-l)^{n-1}} \\
&= l + (x-l) \sum_{i=0}^{n-1} \binom{n-1}{i} \frac{(x-l)^i (r-x)^{(n-1)-i}}{(r-l)^{n-1}} \\
&= l + (x-l) \frac{((x-l)+(r-x))^{n-1}}{(r-l)^{n-1}} = x
\end{aligned}$$

At the first equality sign, we distribute inside the first sum; in the second term, we simplify the denominator with the numerator $(r-l)$. At the second equality sign, we recognize that the first term contains a binomial formula corresponding to $((x-l)+(r-x))^n$; in the second term, we recognize that the first element of the sum can be removed because it is 0, also we recognize that $\frac{i}{n}\binom{n}{i}$ is $\binom{n-1}{i-1}$ when $i \neq 0$. At the third equality sign, we use the equality $(x-l)+(r-x) = r-l$ for the first term and we factor out $(x-l)$ from the remaining indexed sum and re-index that sum. We then recognize another binomial formula and can conclude.

The Bernstein coefficients are related to a broken line (made of contiguous straightline segments) which gives a rough approximation of the polynomial's function graph. More precisely, given the bounds (l, r) of the interval, and the Bernstein coefficients $(b_0, \ldots, b_n)$ of polynomial p, the $n+1$ points with coordinates $(l + i\frac{r-l}{n}), b_i)$ are the control points the polynomial is a Bézier curve of. Each of these points describes the behavior of the polynomial when the input x is close to $l + i\frac{r-l}{n}$. In this sense, Bernstein coefficients can be said to *control* the behavior of the polynomial in some part of the interval. These points can be move about the vertical line $x = l + i\frac{r-l}{n}$. When b_i is close to the average between b_{i-1} and b_{i+1} (in other words, when c_i is close to the segment joining c_{i-1} and c_{i+1}), the behavior of the polynomial in this area is quite eventless. But when c_i is significantly removed from this segment, c_i appears to be pulling the polynomial's curve in its direction, with the curve of the polynomial usually crossing the vertical line $x = l + i\frac{r-l}{n}$ between the segment and the point c_i. This is illustrated in the Figure 1.

In Figure (1-a) the illustration shows that a peak in the disposition of the control points corresponds to a bend in the polynomial's curve (the Bernstein coefficients are 1, 3, -10, 1, 4, 1 and -10 corresponds to a downward peak). In this case, the peak provokes two sign changes, which are reproduced in the shape of the curve and correspond to the existence of two roots inside the interval. In Figure (1-b), the coefficients still exhibit a downward peak with a negative coefficient, but the polynomial's curve stays away from the x-axis and the two sign changes in the Bernstein coefficients do not correspond to any real root for the polynomial (this is a false alert). In Figure (1-c), there is on sign change,



(a) 1, 3, -10, 1, 4, 1 (b) 1, 3, -1, 1, 4, 1 (c) 3, 0, 1, 10, -2, -1

Figure 1: Bernstein Control points and corresponding polynomial curves

so that the first and last coefficients have opposite signs. In fact, the first and the last Bernstein coefficients are equal to the values of the polynomial at the bounds of the interval, so this imposes the existence of at least one root. But because there is exactly one sign change in the coefficients, we can be sure that any other bend in the curve stays away of the x-axis.

3 Describing roots of a polynomial in the rational setting

Roots of a polynomial with coefficients in a given field are not necessary in this field. This is only the case with algebraically closed fields like the field of complex numbers. Real closed fields, like the field of real numbers, give weaker properties on the existence of roots: only polynomials with odd degree have at least one root in the real closed field of their coefficients. These fields are ordered and contain enough elements so that they enjoy the existence of the *intermediate value theorem*: any polynomial that has a negative value in one point a and a positive value in another point b is guaranteed to have a root at some point between a and b.

Then, we can consider even smaller fields, like the field of rational numbers. Here we don't have the *intermediate value theorem* anymore. Still, it makes sense to say that a polynomial with rational coefficients has a single *real* root in a given interval with rational endpoints, even without defining real points. In that case indeed the polynomial crosses the real x-axis in exactly one point, which can be approximated arbitrarily precisely by rational numbers. This is what we want to make precise in the next two sections.

3.1 Criteria for the existence of a unique root

We concentrate on a sufficient criterion for the existence of a root inside an interval. This criterion is strong enough to build a Cauchy sequence whose limit in the real numbers would be the root. Our criterion is based on slopes.

Ensuring that the slope is positive or negative is some interval helps making sure that there are not two roots. In our setting, where the polynomials we consider have only simple roots, we have the stronger property that the slope is

separated from 0 by a given ratio. In the case of positive slopes, we write the slope requirement for a polynomial p inside a given interval I as follows:

$$\exists k, 0 < k \quad \wedge \quad \forall x\ y, [x \in I \wedge y \in I \wedge x < y \Rightarrow k(y - x) < p(y) - p(x)]$$

Depending on the kind of interval that we will consider, we will have two different ways to express the existence of a single root in the interval.

1. If the interval is bounded, we express that the interval can be decomposed into three parts, the first part where the polynomial's value is always negative (I_1 in Figure 1, the second part where the polynomial's value goes from negative to positive with a requirement on the slope (I_2 in Figure 1), and the third part where the polynomial's value is always positive (I_3 in Figure 1).

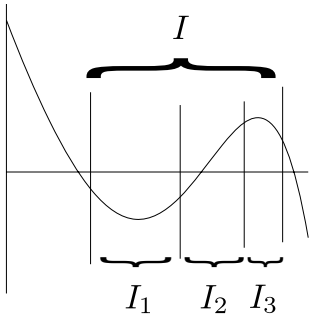


Figure 2: A sufficient criterion for the existence of a single root in a bounded interval

2. If the interval is unbounded, we express that the interval can be decomposed into two parts, the first part where the polynomials value is always negative (I_1 in Figure 2), and the second part where there is a requirement on the slope with a positive slope (I_2 in Figure 2).

3.2 Finding locations where a polynomial's value is arbitrarily small

While we can't be sure to produce a rational value on which the polynomial of interest returns the zero value, we at least need to be able to produce an input for which the polynomial's absolute value is arbitrarily small. In classical mathematics once we know that the polynomial takes values of opposite sign at the bounds of an interval, we know that there is a root for this polynomial in this interval, thanks to the *intermediate value theorem*. For this work, we establish a simplified constructive, real point free, replacement of the intermediate value theorem specialized for polynomials. The statement we prove has the following form:

$$\begin{array}{l} \forall p\ x\ y\ \varepsilon, \\ \quad 0 < \varepsilon \quad \wedge \quad p(x) < 0 \leq p(y) \quad \Rightarrow \quad [\exists x'\ y', -\varepsilon \leq p(x') < 0 \leq p(y') \leq \varepsilon \wedge \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad x \leq x' < y' \leq y] \end{array}$$

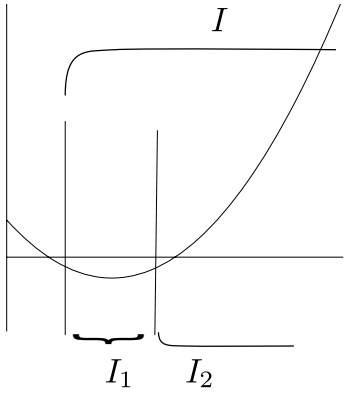


Figure 3: A sufficient criterion for the existence of a single root in an unbounded interval

We call this theorem the *constructive intermediate value theorem.*

We again rely on reasoning about slopes. Without loss of generality, we can assume that the two values x and y are positive. Assuming that the polynomial has the shape $a+X\times p'$, we construct another polynomial P'' whose coefficients are the absolute values of the coefficients of P''. This polynomial is increasing and its maximum value in $[x,y]$ is reached in y. We prove that the slope of the polynomial between any two points inside $[x,y]$ is smaller than $k=P''(y)$. Thus, we establish that the slope of any polynomial is bounded in absolute value on any bounded interval. In particular, for any $z,t\in[x,y]$, we have

$$|p(z)-p(t)|\leq|k\times(z-t)|.$$

For a given ε, and assuming $p(x)<0<p(y)$ we can choose an n so that $\frac{k(y-x)}{n}<\varepsilon$ We then consider the $n+1$ values $a_i=x+\frac{i\times(y-x)}{n}$ and we solve a discrete problem over the values a_i. We simply need to find the largest prefix $a_0,\ldots,a_{j-1}$ so that all values $p(a_k)$ in this prefix are negative. We can set $x'=a_{j-1}$, because the next value a_j is necessarily non-negative and $p(a_j)-p(x')<\varepsilon$, thus $-\varepsilon<p(x')<0$. In a similar way, we can set $y'=a_j$ because $0\leq p(y')<\varepsilon$.

Our algorithm is illustrated in figure 3.2, where the distance between the a_i's is chosen according to the maximal slope occurring between $l=a_0$ and $r=a_{12}$. The point selected by our algorithm is a_8, even though there are more roots in the vicinity of a_1 and a_2 but neither a_1 nor a_2 is a point where the polynomial takes a positive value.

4 A simple form of Descartes' law of signs

One of the main results studied in this paper is that having only one sign changes in the sequence of Bernstein coefficients for the polynomial p and the interval (l,r) ensures that there is only one root of p inside (l,r). The proof of this result relies on a similar property for the standard coefficients of another polynomial q: if there is only one sign change in the coefficients of q then q has only one

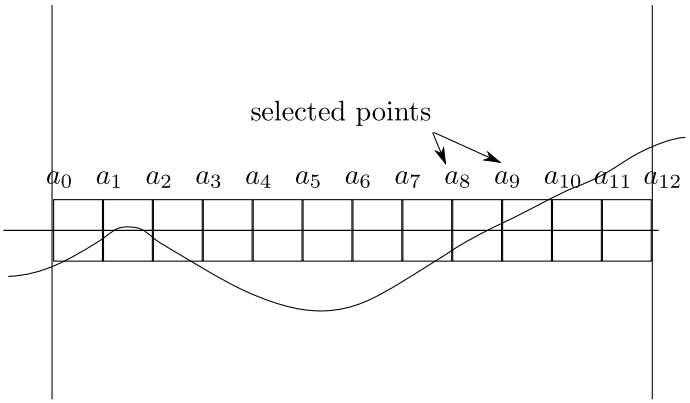


Figure 4: Bounding a polynomials value

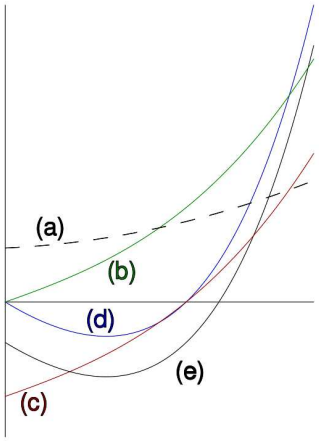
root inside the interval (l, r). In this section, we discuss how this property is proved formally.

4.1 A Geometrical explanation of the proof

Let's first describe a simple graphical argument based on curves for polynomial functions between 0 and $+\infty$, as shown in figure 5. To describe our proof, we assume that new polynomials are built from existing ones by multiplying them by the polynomial X and adding a constant; an operation that is known as "Horner's scheme". Polynomials with one sign change and a positive leading coefficient are obtained by starting with a positive constant, applying Horner's scheme a certain number of times with non-negative constants, then applying it a negative constant, and then applying it again a certain number of times with non-positive constants.

Polynomials with only non-negative coefficients have curves which look like the curves (5-a) or (5-b) depending on whether the first coefficient is 0, adding a positive coefficient to a polynomial of the form (5-a) or (5-b) yields a polynomial of the form (5-a), multiplying a polynomial of the form (5-a) or (5-b) by the polynomial X yields a new polynomial of the form (5-b). Thus, Horner's scheme with non-negative constants keeps polynomials in the (a-b) form. Then, when applying Horner's scheme with a negative coefficient (thus introducing a sign change), the multiplication by X first builds a polynomial of the (5-b) form, and adding a negative constant, one obtains a curve whose shape is given by (5-c). From then on, multiplying a polynomial of the form (5-c), (5-d), or (5-e) by X yields a a polynomial of the (5-d) form; adding a negative constant to a polynomial of the form (5-d) or (5-e) yields a polynomial of the (5-e) form. Polynomials of the form (5-d) or (5-e) share the following characteristic: there exists a positive value x, so that the polynomial has a negative value between 0 and x, and the slope of the curve is strictly positive above x. Because of the slope condition, we can also find a point where the polynomial is positive.

Let us now give a more precise proof, outlining the concepts that are used in the formal proof.



(a) non-negative coefficients, first one non-zero,
(b) non-negative coefficients, first one zero,
(c) first coefficient negative, all others non-negative,
(d) one sign change, first coefficient zero,
(e) one sign change, first coefficient negative

Figure 5: Classes of polynomials with or without sign change

4.2 Lemmas for polynomials with non-negative coefficients

Polynomials are simply encoded by their lists of coefficients, evaluating a polynomials on a given number is done recursively following Horner's scheme, and recognizing polynomials with only non-negative coefficients is also done using a simple recursive function, written in the following form:

```
Fixpoint all_pos_or_zero (l:seq Qcb) : bool :=
if l is a::tl then (0 <= a) && all_pos_or_zero tl else true.
```

As a reminder, the Coq syntax variant that we use, SSReflect, privilegies boolean predicates, so that `(0 <= a)` stands for a boolean value computed by a recursive function instead of a proposition as in standard Coq litterature. Also, the type `Qcb` stands for a representation of rational number as fractions in canonical form (hence the letter `c`), where the verification that the fraction is in canonical form is also expressed using a boolean function (hence the letter `c`). Since these numbers are in canonical form, the equality test between two fractions is simply based on syntactic equality, in other words Leibnitz equality. The type constructor `seq` is a type for lists, with extra constraints on the types that can be stored in such lists: equality must be decidable, a feature that can then be exploited intensively by the SSReflect package [GM08].

We should remark that polynomials satisfying the boolean predicate `all_pos_or_zero` may not contain any positive coefficients: for this reason, we cannot guarantee that they are increasing or strictly positive anywhere between 0 and $+\infty$.

We prove easily by induction on lists that if they contain only non-negative coefficients, then the corresponding polynomial always has a postive value in $(0, +\infty)$ and from then, we also prove by induction that any polynomial with only non-negative coefficients is increasing.

We then prove that for every polynomial P with non-negative coefficients, the product $x \times P(x)$ can be made arbitrarily close to 0 while x stays in $(0, +\infty)$.

4.3 Two lemmas on slopes

A first lemma on slopes concerns the existence of points where a polynomial P takes a value above an arbitrary bound a. If the slope is bounded below by a positive ratio k, this is guaranteed as it suffices to take a value that is large enough. As the proof is constructive, we need to be more precise: assuming the slope is larger than k for any y larger than x_1, it suffices to take any value larger than $x_1 + \frac{a-P(x_1)}{k}$. This result is remembered in our development under the name `above_slope`.

A second lemma on slopes concerns the slope of a product of the form $x \times P(x)$. This lemma reproduces the known formulas for the derivative of products of derivable functions, but is expressed solely in terms of lower bounds of slopes: If a function f has a slope larger than or equal to a non-negative ratio k_f when x is larger than a certain bound a, then then the slope of the product $x \times f(x)$ is larger than $ak_f + f(x)$.

This statement requires f to have a positive slope, but it leaves open whether $f(x)$ is positive or not. In particular, the values a and k_r can be fixed for a large interval: we intend a to be the lower bound of interval I_2 as used in the criterion for existence of a unique root in an unbounded interval (see Figure 2).

4.4 Polynomials with exactly one sign change

We can then address the case of polynomials with exactly one sign change. We want to show that these polynomials have exactly one root. We exhibit the two intervals described in the criterion for unbounded intervals (see Figure 2) the positive value x_1 and the positive ratio k such that the polynomial is negative in the interval $(0, x_1)$ and the slope between any two values above x_1 is larger than k.

To detect polynomials with exactly one sign changes, we use two recursive functions. The first one, which we call `alternate_1`, recognizes polynomials with at least one positive coefficient, preceded by any number of non-positive coefficients (possibly 0), and followed by only non-negative coefficients, as checked by `all_pos_or_zero`. This function is defined as follows:

```
Fixpoint alternate_1 (l:seq Qcb) : bool :=
  if l is a::tl then
     if 0 < a then all_pos_or_zero tl else alternate_1 tl
  else false.
```

The second function, which we call `alternate`, checks for the presence of at least one negative coefficient and then calls `alternate_1`. Thus, `alternate` calls itself recursively as long as it finds zero coefficients, the function `alternate_1` also calls itself recursively as long as it finds non-positive arguments.

As we have two recursive functions, `alternate_1` and `alternate`, we actually need to perform two proofs by induction. Each proof by induction shows that some invariant is satisfied.

The invariant for `alternate_1` must be satisfied by a polynomial P that may or may not contain a negative coefficient, so that this invariant cannot guarantee the existence of places where the polynomial takes a negative value. Instead, this invariant guarantees for any positive ε the existence of a positive `x` and a k so that:

1. for any y between 0 and x, $P(y) \leq P(x)$,
2. the slope between two points larger x is guaranteed to be larger than k,
3. the number $x \times P(x)$ is between 0 and ε.

The invariant for `alternate` is exactly the criterion we use to describe the existence of exactly one root in an unbounded interval. This proof by induction is done by induction on the list. The empty list does not satisfy the predicate `alternate` so that this case is taken care of easily. The other case is when the polynomial is described by a list of the form $a::l$, so that l represents another polynomial P_l and $P(x) = a + x \times P_l(x)$. Here another case distinction mut be studied, depending on whether a is zero or negative.

If a is negative, we cannot use an induction hypothesis, because in this case l is only guaranteed to satisfy the predicate `alternate_1`. On the other hand, the invariant for `alternate_1` guarantees the existence of an x so that $x \times P_l(x)$ is positive and smaller than $-a$, this x is the right witness and the slope is $P_l(x)$. Since P_l is negative at the left of x it is easy to prove that $y \times P_l(y)$ is negative when $0 < y \leq x$, and thus $P(y)$ is negative. To reason on the slope, we use our lemma about the slope of $x \times P(x)$, using 0 for the slope of P (we only know that it is increasing).

If a is zero, we have by induction hypothesis that there exists an x and k so that P_l is negative on the left of x and has a slope larger than k on the right of x. However, this does not guarantee that x is the right witness for P because the slope of $x \times P_l(x)$ is only larger than $P_l(x) + x \times k$, and $P_l(x)$ is negative. The solution is to note that P_l necessary takes a positive value in some point v_1 on the right of x and to use our *constructive intermediate value theorem* from section 3.2 to build a new value x_1 so that $-\frac{kv_1}{2} \leq P_l(x_1) \leq 0$. Now P_l is still guaranteed to be negative between x and x_1, because of the slope condition and now the slope on the right of x_1 is guaranteed to be larger than $\frac{kv_1}{2}$, which is positive.

5 From Bernstein to Descartes

In this section, we clarify the polynomial transformations that link the problem of finding the roots of a polynomial inside an arbitrary bounded interval (l, r) successively with the problem of finding the roots of an other polynomial inside the interval $(0, 1)$ and with the problem of finding the roots of yet another polynomial inside the interval $(0, +\infty)$. These transformations make it possible to compute another collection of coefficients, which happen to be very simply related to Bernstein coefficients.

Proving the properties of Bernstein coefficients works by establishing a route from Descartes' law of signs to Bernstein coefficients. Descartes' law of signs works for the interval $(0, +\infty)$. This criterion can easily be adapted to any half-line interval $(a, +\infty)$ and more precisely to $(1, +\infty)$. Then a criterion on $(1, +\infty)$ can be transformed into a criterion on $(0, 1)$. This can, in turn, be transposed to any interval. It happens that this path gives a way to reason on Bernstein coefficients.

5.1 A criterion for the interval $(1, +\infty)$

The law of signs gives us a sufficient condition to determine when the unbounded interval $(0, +\infty)$ contains exactly one root for a polynomial. Through a change of variable, we obtain a similar criterion for the interval $(1, +\infty)$.

In the following, we will call θ_v the transformation that maps any polynomial P to the polynomial $y \mapsto P(y + v)$. If $P = \sum_{i=0}^{n} a_i x^i$, We have the following formula:

$$P(y+v) = \sum_{i=0}^{n} a_i (y+v)^i = \sum_{k=0}^{n} (\sum_{i=k}^{n} a_i \binom{i}{k} v^{i-k}) y^k$$

The polynomial P has exactly one root in the interval $(v, +\infty)$ if and only if the polynomial $\theta_v(P)$ has exactly one root in the interval $(0, +\infty)$. We proved this lemma, using our criterion for a unique root in an unbounded interval to express the existence of root.

Thus, if we apply Descartes's law of signs on the coefficients $\sum_{i=k}^{n} a_i \binom{i}{k}$ we can obtain a sufficient criterion for the existence of exactly one root of polynomial $P = \sum_{i=0}^{n} a_i x^i$ in the interval $(1, +\infty)$.

5.2 A criterion for the interval $(0, 1)$

Descartes' law of signs works for unbounded intervals. In this section, we see how to cover also bounded intervals. The trick here relies on reversing the polynomial's list of coefficients. Obviously, the number of sign changes in a list of coefficients is the same as the number of sign changes in the reversed list.

However, the roots of a polynomial on the interval $(1, +\infty)$ are in one-to-one correspondence with the roots of the reversed polynomial between zero and one. This is due to the following equation:

$$\sum_{i=0}^{n} a_i x^i = x^n \times \sum_{i=0}^{n} a_i x^{i-n}$$

We can now perform another change of variable, here $y = 1/x$ and a change of index $j = n - i$ in the sum.

$$\sum_{i=0}^{n} a_i x^l = (\frac{1}{y})^n \sum_{j=0}^{n} a_{n-j} y^j$$

The polynomial $\sum_{j=0}^{n} a_{n-j} y^j$ is exactly the reversed polynomial, and the expression $(\frac{1}{y})^n$ never becomes 0 for $y \in (0, 1)$. Thus, x is a root of the polynomial between 1 and $+\infty$ if and only if $y = x^{-1}$ is a root of the reversed polynomial between 0 and 1.

Let us note ρ the function that computes the reverse of a polynomial. Here we need to be precise: the coefficients of a polynomial of degree n actually are the coefficients of a vector in an $n + 1$ dimensional space, whose basis is made of the monomials X^i where $i \in \{0, \ldots, n\}$. Seen as an operation on this vector space, ρ is an involutive automorphism. But polynomials of degree less than n are also elements of this vector space and the reverse operation must be understood as reversing the list of coefficients of length $n + 1$ obtained by completing the polynomials description with enough 0 coefficients.

To illustrate the correspondance between a polynomial and its reverse, we can consider the polynomial $P(x) = x^2 + \frac{3}{2}x - 1$, the reversed polynomial is $Q(x) = -x^2 + \frac{3}{2} + 1$ and after the variable change we obtain the polynomial $-x^2 - 2x + 1$ which exhibits only one sign change. This predicts that the polynomial has exactly one root between 0 and 1, and indeed the two roots of the initial polynomial are -2 and $1/2$. This is illustrated in figure 5.2 where the curve with a solid line is the curve for the polynomial P, while the curve with a dashed line is the curve for the polynomial Q, which has a single root between 1 and $+\infty$.

As a conclusion, we can also establish a correspondance between unique roots in $(1, +\infty)$ unique roots in $(0, 1)$, but not for the same polynomials. When working on rational numbers, this correspondance works by linking the criterion for unbounded intervals with the criterion for bounded intervals. This proof involves the computation of slope for a $x^n p(1/x)$ from the slope of p, which makes it trickier than the rest. This is again a place where our constructive intermediate value theorem plays a role.

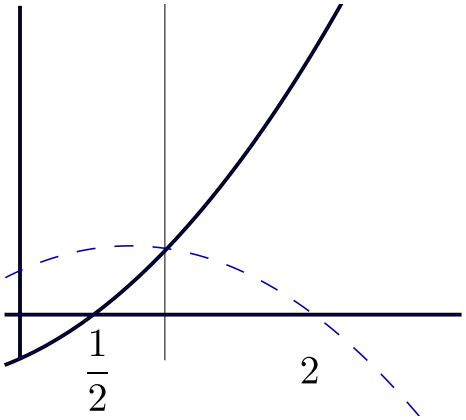


Figure 6: Curves of $x^2 + \frac{3}{2}x1$ (solid line) and its reverse $-x^2 + \frac{3}{2}x - 1$ (dashed)

5.3 Handling arbitrary bounded intervals

The next step is to relate the roots of any polynomial inside an arbitrary interval (l, r) with the roots of another polynomial inside the interval $(0, 1)$. This is done with another change of variable, this time $x = (r - l)y + l$. In other words, the polynomial function which maps any x to $p(x)$ has a root between l and r if and only if the polynomial function which maps any y to $p((r - l)y + l)$ has a root between 0 and 1.

Here again, we can define a generic transformation on polynomials, named χ_k that corresponds to expanding with a given ratio. For an arbitrary polynomial $P = \sum_{i=0}^{n} a_i X^i$, the polynomial $\chi_k(p)$ is defined as follows:

$$\chi_k(p) = \sum_{i=0}^{n} a_i(kX^i) = \sum_i i = 0^n a_i k^i x^i$$

Thus, the change of variable to study the roots of polynomial p is actually represented by $\chi_{r-l} \circ \theta_l$.

The geometric effect of the polynomial transformation is illustrated in figure 5.3, where the shape of the curve for the polynomial $\frac{x^3}{8} - \frac{x^2}{8} + 3x$ inside the interval $(2, 4)$ is reproduced by the shape of the curve for the polynomial $x^3 - \frac{5}{2}x^2 - 2x + \frac{3}{2}$ inside the theorem $(0, 1)$.

5.4 Recapitulating operations

In our formal development, we defined the three operations for translating (θ), expanding (χ), and reversing the list of coefficients (ρ). We can then compute a sequence of coefficients by applying the transformation

$$\tau = \theta_1 \circ \rho \circ \chi_{r-l} \circ \theta_l.$$

When the coefficients we obtain have exactly one sign change, we know that the polynomial has exactly one root inside the interval (l, r).

By construction, each of the operation θ, ρ, χ actually is a linear application of the vector space of polynomials of degree less than n into itself. The inverse

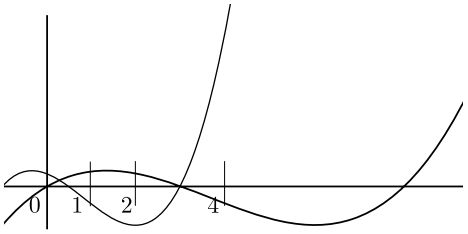


Figure 7: Curves of a polynomial inside $(2, 4)$ and the corresponding transformed polynomial between $(0, 1)$

of θ_a is θ_{-a}, and this is easily proved, so that θ_a is always bijective. When k is nonzero, the inverse of χ_k is $\chi_{\frac{1}{k}}$, so this linear application is also bijective. The inverse of ρ is itself. As a result, the whole transformation is also inversible and its inverse is

$$\tau^{-1}\theta_{-l} \circ \chi_{\frac{1}{r-l}} \circ \rho \circ \theta_{-1}.$$

We proved that the images of the monomials X^i by this inverse transformation are multiples of the Bernstein polynomials, in reverse order. Let us first observe the effect of $\rho \circ \theta_{-1}$:

$$\begin{aligned}
\tau^{-1}(X^i) = \theta_{-l} \circ \chi_{\frac{1}{r-l}} \circ \rho \left(\sum_{j=0}^{i} \binom{i}{j} (-1)^j x^{i-j} \right) &= \theta_{-l} \circ \chi_{\frac{1}{r-l}} \left(\sum_{j=0}^{i} \binom{i}{j} (-1)^j x^{n-i+j} \right) \\
&= \theta_{-l} \circ \chi_{\frac{1}{r-l}} \left(x^{n-i} \sum_{j=0}^{i} \binom{i}{j} (-x)^j \right) \\
&= \theta_{-l} \circ \chi_{\frac{1}{r-l}} \left(x^{n-i} (1-x)^i \right)
\end{aligned}$$

Then let's observe the effect of $\theta_{-l} \circ \chi_{\frac{1}{r-l}}$.

$$\begin{aligned}
\tau^{-1}(X^i) &= \theta_{-l} \left(\frac{x^{n-i}}{(r-l)^{n-i}} \left(1 - \frac{x}{r-l} \right)^i \right) \\
&= \theta_{-l} \left(\frac{x^{n-i}}{(r-l)^{n-i}} \frac{(r-l-x)^i}{(r-l)^i} \right) \\
&= \theta_{-l} \left(\frac{x^{n-i}(r-l-x)^i}{(r-l)^n} \right) \\
&= \frac{(x-l)^{n-i}(r-l-(x-l))^i}{(r-l)^n} \\
&= \frac{1}{\binom{n}{i}} (P_b(n, l, r, n-i))
\end{aligned}$$

If the transformation $\tau(p)$ leads to a sequence of coefficients c_i, this means $\tau(p) = \sum_{i=0}^{n} c_i x^i$. Now, using the fact that both τ and τ^{-1} are linear, we can

see that the polynomial p as

$$\begin{aligned} p &= \tau^{-1}\left(\sum_{i=0}^{n} c_i X^i\right) \\ &= \sum_{i=0}^{n} c_i \tau^{-1}(x^i) \\ &= \sum_{i=0}^{n} c_i \frac{1}{\binom{n}{i}} P_b(n,l,r,n-i) \end{aligned}$$

Thus, the Bernstein coefficients are obtained in the following manner:

$$b_i = \binom{n}{n-i}^{-1} c_{n-i} = \binom{n}{i}^{-1} c_{n-i}$$

Since the number of sign changes does not depend on the order in which the list is observed, we obtain the proof that one sign change in the sequence of Bernstein coefficients implies the existence of a root in the interval (l, r).

6 Dichotomy

Bernstein coefficients give precise information when they exhibit either zero or one sign change. In the first case, we have the guarantee that there are no roots of the considered polynomial in the considered interval. In the second case, we have the guarantee that there is exactly one root.

When Bernstein coefficients exhibit more than one sign change, no conclusion can be drawn about the existence and unicity of roots in the interval. For instance, in Figure (1.b), the Bernstein coefficients exhibit two sign changes, but there is no root inside the interval. When facing this kind of unconclusive information, the solution is to refine the approximation given by the control line.

6.1 Geometric intuition for dichotomy

When cutting an interval in two halves, the number of control points is approximately doubled, because each of the new half-intervals receives a new sequence of n Bernstein coefficients. As a result, the control points are closer to each other and to the polynomial's curve and they give a more accurate account of the curve's position with respect to the x-axis. This is illustrated in Figure 8, where the initial Bernstein coefficients exhibit a sign change, which is needed to account for the bend in the first half of the interval (a positive local minimal, but expressed by a negative Bernstein coefficient). In the halved interval two more points are added in the vicinity of the bend, and none of the control points needs to be negative anymore.

In Figure 8, the dotted line represents the polynomial's curve, the solid line links the control points for the largest interval, marked by round bullets (the Bernstein coefficients are 1, 3, -1, 1, 4 1 for this interval). The dashed line links the control points for the two half intervals, marked by square boxes (the Bernstein coefficients are 1, 2, 1.5, 1, 0.9375, 1.15625 for the first interval, and

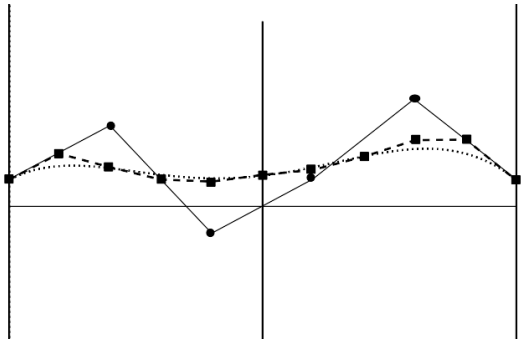


Figure 8: Bernstein control points for halved intervals

1.15625, 1.375, 1.875, 2.5, 2.5, 1 for the second interval). This figure illustrates that the control line really gets closer to the polynomial's curve, and provides a much better approximation of the polynomial.

The formula given in section 2 is useful to compute an initial series of Bernstein coefficients, and the correctness of the conditions for existence of roots based on these coefficients can be justified using the transformation described in section 5.4.

It may seem that computing Bernstein coefficients is a costly process. Around 1950, while studying Bézier curves, De Casteljau noticed that the coefficients for the sub intervals were easy to compute from the coefficients for the big interval through a simple recursive process, exploiting the recurrence relation already given in section 2. We have also proved the correctness of this algorithm. This proof is the topic of the next section.

6.2 Initialization

Given an arbitrary polynomial non constant polynomial p of degree n, defined by $p = \sum_{i=0}^{n} a_i X^i \in \mathbb{Q}[X]$ it is actually possible to bound the absolute values of its roots by a simple constant defined from the coefficients $(a_i)i = 0 \dots n$, called the Cauchy bound [BPR06]:

$$\forall x \in \mathbb{R}, p(x) = 0 \Rightarrow |x| \leq C(p) \quad \text{with} \quad C(p) = \sum_{i=0}^{n} \frac{|a_i|}{|a_n|}$$

Indeed, let x be a root of p. If $|x| \leq 1$, since $1 \leq C(p)$, the inequality trivialy holds. Then if $|x| > 1$, since x is a root, and $a_n \neq 0$

$$x^n = -\frac{1}{a_n} \sum_{i=0}^{n-1} a_i x^i$$

Hence:

$$|x|^n \leq \frac{1}{|a_n|} \sum_{i=0}^{n-1} |a_i||x|^i \leq \frac{1}{|a_n|} \sum_{i=0}^{n} |a_i||x|^i$$

first by triangular inequality, then by adding a non-negative constant on the right (otherwise the sum might be empty). Then:

$$|x| \leq \frac{1}{|a_n|} \sum_{i=0}^{n} |a_i||x|^{i-n} \leq \frac{1}{|a_n|} \sum_{i=0}^{n} |a_i|$$

since $|x| > 1$ implies that forall $i = 0 \ldots n$, $|x|^{i-n} \leq 1$.

This means that to start studying the roots of a polynomial p we can restrict the infinite real line to a bounded interval $(-C(p), C(p))$. This justifies we can start a real root isolation process by providing the initial interval of interest. On this first interval, we compute Bernstein coefficients from the transformations presented in the previous section. Then in case of more that one sign change, we continue by invoking the splitting de Casteljau algorithm exposed in the next subsection.

6.3 Splitting algorithm

Given three pairwise distinct rational numbers l, r, m, there exists an efficient algorithm to deduce the two respective lists of Bernstein coefficients of a polynomial P on intervals (l, m) and (m, r) from the list of Bernstein coefficients of P on interval (l, r).

Let `b` be the sequence of Bernstein coefficients of a polynomial P for an interval (l, r) with $l, r \in \mathbb{Q}$ and degree n. Let $m \in \mathbb{Q}$ be a rational number distinct from l and r. We pose $\alpha = \frac{m-l}{r-l}$ and $\beta = \frac{r-m}{r-l}$. The de_casteljau algorithm is defined recursively by:

```
Fixpoint de_casteljau (alpha beta : Qcb) (b : nat -> Qcb) (n :
    nat) :=
  match n with
  | 0 => b
  | i.+1 => fun j =>
    (alpha * de_casteljau c i j + beta * de_casteljau c i j.+1)
end.
```

where the initial sequence of coefficients `b` is represented by an infinite sequence of rational numbers, for which only the first n elements are relevant. The following function gives the Bernstein coefficients of P on the finite interval (l, m).

```
Definition dicho' alpha beta c i :=
  de_casteljau alpha beta c i 0.
```

The following function gives the Bernstein coefficients of P on the finite interval (m, r).

```
Definition dicho alpha beta p c i :=
   de_casteljau alpha beta c (p - i) i.
```

Observing the function de_casteljau more precisely, we see that the algorithm actually proceeds by creating a succession of lines where the element at rank j in a given line is obtained by computing a weighted sum of the two elements at rank j and $j + 1$ on the previous line.

This process can be illustrated geometrically by a succession of broken lines. For the first line, we take the control line of the initial interval. Then, for each of the segments that compose this control line, we cut this segment in the same proportion as the the proportion in which the interval is split between (l, m) and (m, r). This gives us a new collection of points. We started with $n + 1$ control points and thus had n segments, we now have n new points, defining $n - 1$ new segments. We repeat this process with the new segments, until we reach a situation where there is only one segment and we again split this segment into two parts in proportion of (l, m) and (m, r). The last point is guaranteed to lie on the polynomial's curve.

Although we actually only use de Casteljau's algorithm when m is the midpoint of the initial interval, it works for any relative positions of l, m, and r, as long as they are pairwise distinct.

The different points computed by the de Casteljau algorithm are represented on figure 9. The innermost points are the control points in the two new bases, computed from the originial control points $\{B0, \ldots, B5\}$. The middle innermost control point, on the curve, belongs to the two new lists of control points. Points $\{C0, \ldots, C5\}$ are the control points in the left half, $\{D0, \ldots, D5\}$ are the control points on the right half. De Casteljau algorithm is extensively used in computer graphics for rasterizing Bézier curves.

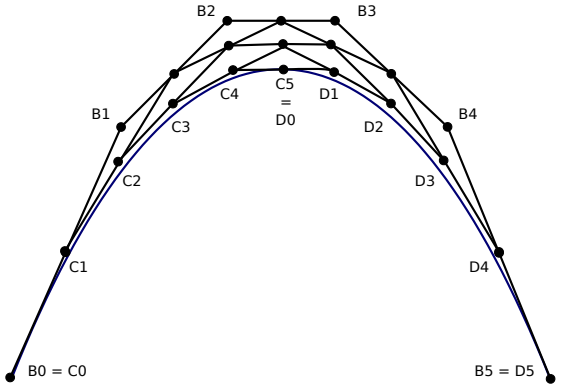


Figure 9: Intermediate points computed by de Casteljau splittling algorithm applied at the middle of $[B0, B5]$

The aim of this section is to prove that this algorithms is correct, i.e. that the dicho and dicho' function indeed computes the expected Bernstein coefficients. The correctness theorems as stated in Coq are:

```
Lemma dicho'_correct : forall (l r m : Qcb)(q : {poly Qcb})(p :
    nat)
(c : nat -> Qcb)
(alpha := (r - m) * (r - l)^-1) (beta := (m - l) * (r - l)^-1),
  m != l ->
  q = \sum_(i < p.+1)(c i) * bernp l r p i ->
  q = \sum_(j < p.+1)(dicho' alpha beta c j) * bernp l m p j.
```

```
Lemma dicho_correct : forall (l r m : Qcb)(q : {poly Qcb})(p :
    nat)
(c : nat -> Qcb)
(alpha := (r - m) * (r - l)^-1) (beta := ((m - l) * (r - l)^-1)),
  m != r ->
  q = \sum_(i < p.+1)(c i) * bernp l r p i ->
  q = \sum_(j < p.+1)(dicho alpha beta p c j) * bernp m r p j.
```

where `(bernp l r p i)` is the i-th polynomial in the Bernstein basis of degree p with parameters l and r.

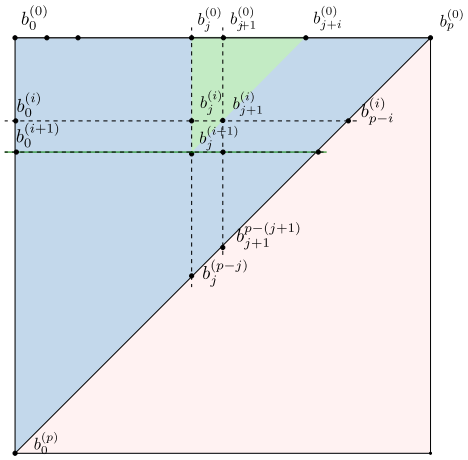


Figure 10: Properties of de Casteljau computations

The properties of computations performed by the de Casteljau algorithm are summarized on Figure 6.3. Starting from the input list $b = (b_0^(0) \dots b_p^{(0)})$ of coefficients in the basis with parameters l and r, on the upper side of the triangle, it computes the full triangle, so that in the end the two expected output lists can be read on the two other sides of the triangle. The list $b' = b_0^{(0)} \dots b_0^{(j)} \dots b_0^{(p)}$ is the list of coefficients in the basis with parameters l and m output by `dicho'`. The list $b'' = b_0^{(p)} \dots b_j^{(p-j)} \dots b_p^0$ is the list of coefficients in the basis with parameters m and r output by `dicho`. The green area on Figure 6.3 show which values the computation of an arbitrary given point in the triangle relies on. This structure is imposed by the fixpoint equation of the recursive definition of the de Casteljau algorithm:

```
de_casteljau alpha beta b n.+1 i =
(de_casteljau alpha beta b n i) + (de_casteljau alpha beta b n
    i.+1)
```

which looks very similar to the recursive relation governing the Pascal triangle.

Let us first notice that the shape of Bernstein polynomials implies that:

```
Lemma bern_swap :
 forall n i l r,
 (i <= n) -> r != l -> bernp r l n i = bernp l r n (n - i).
```

This remark implies that if b is the list of coefficients of the polynomial p in the Bernstein basis of degree n with parameters l and r, then the reverse of b is the list of coefficients of the same polynomial p in the Bernstein basis of degree n with parameters r and l: reversing the list swaps the parameters:

```
Lemma bern_rev_coef : forall (n : nat)(l r : Qcb)(b : nat -> Qcb)
  ,
 \sum_(i < n.+1)(b i) * (bernp l r n i) =
 \sum_(i < n.+1)(b (n - i)) * (bernp r l n i).
```

This remark shows that the correctness of the `dicho'` function is enough to get a certified computation of both Bernstein coefficient lists: if b is the initial list of Bernstein coefficients with parameters l and r, then reversing b gives the coefficients with parameters r and l, applying `dicho'` on the reverse of b using r, l and m computes the coefficients with parameter r and m, hence reversing this output gives the result expected for `dicho` on b using l, r and m. Using a similar symmetry on the `de_casteljau` algorithm, we in fact reduce the proof of the `dicho_correct` specification to the proof of `dicho'_correct`.

By linearity, we can also reduce the proof of the `dicho'_correct` specification to the case where the input polynomial p is in fact itself a Bernstein polynomial. This means that the input coefficient list b only contains zeros except at one position where the coefficient is one.

Let us first compute the expected output of the `dicho'` function on a such a list. In other words, for any distinct rational numbers l, r, m and any $n \in \mathbb{N}$, given $i \leq n$, we want to compute the coefficients of:

$$P_b(n, l, r, i) = \binom{n}{i} \frac{(X-l)^i (X-r)^{n-i}}{(r-l)^n}$$

in the basis $(P_b(n, l, m, i))_{i=0,\ldots,n}$. We pose $\alpha = \frac{r-m}{r-l}$ and $\beta = \frac{m-l}{r-l}$. In the polynomials of the new basis, formal denominators are of the form $(m-l)$. By noticing that:

$$\frac{X-l}{r-l} = \beta \frac{X-l}{m-l} \quad \text{and} \quad \frac{r-X}{r-l} = \alpha \frac{X-l}{m-l} + \frac{m-X}{m-l}$$

and by using the binomial identity:

$$\binom{n}{i}\binom{n-i}{j-i} = \binom{j}{i}\binom{n}{j}$$

we obtain that:

$$P_b(n, l, r, i) = \sum_{j=i}^{n} \binom{j}{i} \alpha^{j-i} \beta^i P_b(n, l, m, j) \quad (*)$$

Now to achieve the proof of the `dicho'_correct` lemma, it is sufficient to prove that the values output by the `dicho'` function coincide with the ones of $(*)$, which boils down to an induction on i.

7 Formalization issues

The sources of this development are available from
`http://www.inria.fr/sophia/members/Yves.Bertot/proofs/bernstein.html`

7.1 Numbers

The work we describe in this article was intentionally not based on any implementation of real numbers, but expressed solely in terms natural numbers and rational numbers. Our intention is to provide a description of Bernstein polynomials and their characteristics that will be usable in any real closed field, but without assuming the intermediate value theorem.

We proved a minimal set of properties of polynomial functions by expressing these properties in the first-order theory of rational numbers. Thus, we do not treat a general notion of Cauchy sequences, continuous functions, or derivability. As a result, we do not need to rely on "book equality" or setoids and express all equalities using "Leibniz equality".

As a result, our work is versatile and can be embedded in any choice of formalization: it is compatible with constructive mathematics, as for instance implemented in Coq in [CFGW04, O'C07, O'C08] as it relies solely on intuitionistic mathematics and it is also readily usable with the "classical real numbers" of the standard Coq library since it relies on plain equality.

It was instrumental in our work that we could reason on rational numbers by assuming that they are in a field with decidable Leibniz equality. In our first experiments, started around 2005, we used a model of rational numbers that only provided a setoid structure. This approach turned out to be too limiting and the development pace has improved drastically once we came back to decidable equality on rational numbers. The current formalization `Qcb` relies on reduced fractions, composed of a numerator, a denominator, and a proof that they are relatively prime, expressed using a boolean predicate to benefit from the unicity of equality proofs in the boolean type. Alternative representation of rational numbers with Leibniz equality can also be found in [Ber03].

The criteria we have chosen to describe the existence of roots are not faithful to the usual notion of having a root. For instance, they are not satisfied for the polynomials $x^4 - x^2 + \frac{1}{4}$ and $x^3 - \frac{3}{2}x^2 + \frac{3}{4}x - \frac{1}{8}$, which both would be thought of having roots inside $(0, 1)$ in the usual sense, these root being respectively $\frac{1}{\sqrt{2}}$ and $\frac{1}{2}$. The curve of the first polynomial only touches the x-axis in a place that is not a rational number, but stays on the positive side. The curve of the second one really crosses the x-axis, moreover it does so in a rational number. The problem is that in both case the root is multiple and the slope does not satisfy the property of staying away from 0. This shows that the criteria we used to express to represent the existence of roots are specially designed for this proof where we concentrate on polynomials with only simple roots and should probably not be used in over settings.

On the other hand, our "constructive intermediate value theorem" is only specialized to work on polynomials and could actually be easily generalized to any uniformly continuous function over a bounded interval, as it is standard in constructive real analysis [Bis67]. Still, this theorem alone does not give a simple way to construct a Cauchy sequence representing a root, whereas a root isolation process does. Thus, these three aspects of our formalization —characterization of roots and intermediate value theorem— are designed especially for the purpose of this development.

7.2 Algebraic structures

This work is based on the hierarchy of algebraic structures available in the SSReflect repository. This hierarchy, described in [GMR+07], has been since enriched with interfaces for ordered integral domains and fields. The ordered integral domains instances we use here are integers and rational numbers. For this purpose, we have populated the hierarchy with the adequate structure instances on the types provided by the standard library of the Coq system, namely `Z`, the type of integers as binary words, and `Qcb`, the type of rational numbers as irreducible fractions. All SSReflect algebraic structures require their carrier type to be equipped with a boolean equality test reflecting the structural Leibniz equality available by default on the type. In our case, since the only concrete types we manipulate (integers and rational numbers) constructively fall into this category, this was not a problem.

The libraries already present in the standard distribution of the system were providing a sufficient body of lemmas to achieve this formalization. However, the SSReflect layer of interfaces introduces a very uniform framework to deal with algebraic manipulations. In particular, instances of a same interface share theory, hence lemma names, and notations. The theory we describe in this paper involves for instance several ring structures, namely the ring of integers, the ring of rational numbers, and the ring of polynomial with rational coefficients. Being all instances of the ring abstract interface (and also integral domains, and even field for rationals and integers), all these structures share the same symbols for operations, like `(x * y)` for multiplication of `x` by `y`, `(x *+ n)` for the product of `x` by a natural number (which is in fact defined as iterated additions). This latter operation and generic notation is of special interest to deal with binomial coefficients in a transparent way since the theory of binomial coefficients, defined as natural numbers, directly applies without specifying any specialized injection from natural numbers to each ring structure. In fact once rationals and integers are equipped with the appropriate structures, our proofs no more involve any argument specific to the representation of arithmetic.

7.3 Representations of polynomials

This work is devoted to the study of roots of univariate polynomials, from an analytic point on view. Therefore, a polynomial P is here seen as a function $P : \mathbb{Q} \to \mathbb{Q}$. For this purpose we can simply represent a polynomial by a big endian list of coefficients in the monomial basis. This representation can be simply interpreted as a function $\mathbb{Q} \to \mathbb{Q}$ using the Horner evaluation scheme.

This representation is however not canonical since the same polynomial can be represented by an infinite number of coefficient lists, only differing by the number of tail zeroes. The ring structure of polynomials of arbitrary degree is actually based on a normalized version of this representation, namely by the type:

```
Record polynomial (R : ringType) :=
  Polynomial {polyseq :> seq R; _ : last 1 polyseq != 0}.
```

that is a pair of a list `polyseq : (seq R)` with a proof that the last element of this list is non zero. The type `(polynomial R)` is used under the notation `{poly R}`.

The constant zero polynomial is hence represented by the empty list. The `:>` symbol indicates that the `polyseq` constructor is declared as a coercion: a polynomial can at any time be seen as a list of coefficient, forgetting the proof that it is in normal form.

Note that polynomials require a ring structure on their coefficients, which ensures that arithmetic operations on polynomials will be properly defined and specified. The polynomial X is represented by the list $[::0,1]$, where 1 and 0 are the one and zero constant of the underlying coefficient ring. Moreover, since the SSReflect ring structure embeds a type with boolean equality, polynomials inherit from a canonical boolean equality test.

At many places however, we need to consider polynomials as lists of coefficients in a basis for the vector space of polynomials of degree less than a fixed value. In the case of a monomial basis, we need to consider lists of coefficients with a meaningful number of tail zeros. This is for instance the case in the proof that the images of the monomials X^i by a certain transformation are positive multiples of the Bernstein polynomials (see section 5.4). Before being transformed, the list representing the normal form of polynomial X^i should be padded with $(p-i)$ tail zeroes to obtain the coefficient list of X^i in the monomial basis of polynomials of degree less than p. This amounts to formalizing the fact that polynomials of degree less than a fixed value form a finite dimensional vector space and to specify the changes in the basis. Unfortunately, the linear algebra part of the SSReflect archive is not yet sufficiently well integrated to get this easily from an existing infrastructure. We hence only define these linear algebra operation at a low level (catenating zeros,...) and prove them correct, which certainly leaves room for improvement.

Correctness proofs of the transformations involve exact identities between polynomials and not extensional identities between polynomial functions. For this purpose we need to use normalizing constructors to interpret an arbitrary list into a polynomial. The constructor:

```
Definition poly_cons (c : R)(p : {poly R}) :=
  if p is Polynomial ((_ :: _) as s) ns then
    @Polynomial R (c :: s) ns
  else c%:P.
```

builds the normal form of the polynomial $(c +' X * p)$, provided that `p` has type `{poly R}` and is hence in normal form. Since `p` is in normal form is it either non-zero, and in that case, formed with a list `s` that has a non-zero element, or zero, and in that case formed with the empty list. In the first case, the new normal form is simply `(c :: s)` and the proof of well-formedness is up to conversion the initial proof that `p` was well-formed. In the second case, we output the constant polynomial with value `c`, denoted `c%:P`.

Now the operator:

```
Definition Poly := foldr poly_cons 0%:P.
```

iterates this construction and builds a normal polynomial out of an arbitrary list. It is straightforward and useful to have lemmas specifying the effects of lists iterator (like catenation, mapping,...) under the `Poly` construction, specially when dealing with the correctness proofs of the Bernstein transformation (zero-padding, expansion,..).

An other important consequence of the ring structure available on polynomials is that we are allowed to define a polynomials by providing its coefficient function `nat -> R`. This facility is in fact a direct application of the iterated operator facilities available in SSReflect [GGMR09]: given a coefficient function `c : nat -> R`, and a natural number `n` the expression:

```
\sum_(i < n) (c i)%:P * 'X^i
```

represents the polynomial $\sum_{i<n} c_i X^i$, defined as a sum of products of powers of the polynomial `'X` by constant polynomials built out of the first values of `c`.

To avoid dealing with out of bound accesses in coefficient lists, the de Casteljau algorithm works on infinite lists of coefficients represented as functions `nat -> Qcb`. The correctness lemmas however reinterpret these coefficient functions as polynomials using the above iterated sum facilities.

7.4 Automation issues

The automation given by the type inference based infrastructure is relieving the user from many painful formalization issues. Beside the sharing of notation and theory mentioned above, which is automatically infered on the fly, algebraic manipulation on indexed sums is a routine work. The correctness proofs of the transformations mentioned in section 5.4 involve a substantial work on iterated sum handling.

Since ring operations are automatically equipped with the iterated operation facilities [GGMR09], proof steps like distributing constant or natural number product on an iterated sum, replacing an iterated sum of zeroes elements by a zero, replacing $\sum_{i=0}^{n} r_i$ by $\sum_{i=0}^{n} r_{n-i}$ only require a single rewriting step. Even a more complex operation combining associativity and reindexing like transforming $\sum_{i=0}^{n+m} r_i$ into $\sum_{i=0}^{n} r_i + \sum_{i=0}^{m} r_{n+i}$ is again a single rewriting step of a generic lemma about iteration of associative operators. This SSReflect library on indexed iterations has been of crucial importance in a large part of the proofs.

On the other side, a significant part of scripts is devoted to too many atomic rewrite steps to normalize ring expressions, or prove trivial consequences of the properties of order like transitivity or compatibility with field operations. The SSReflect libraries still lack the standard automated proof producing decision procedures available in the Coq system, like ring normalization or linear arithmetic decision. The SSReflect structures are indeed still not connected to these mechanisms. The Coq tactics on linear arithmetic are hardcoding the representation of integers and coefficients and should rely on a more abstract structure like the one of ordered field. Moreover an SSReflect ring structure instance is not automatically registered as a valid ring structure for the `ring` tactic. This issue should be solved by bridging the gap between structures infered by type inference (which is the case of our ring structures) and the OCaml database used by Coq tactics.

In the case of the automation of ring identities, it would significantly help the user if normalization could handle simultaneously the various ring and semi-ring structures that can occur in an expression, like the rings of polynomial coefficients, polynomials themselves and possibly the semi-ring of integers.

In the case of the automation of ordered arithmetic, proof steps often involve non linear expressions, for which it is quite difficult to get a truely generic and

efficient proof producing decision procedure. This is in fact part of the long term objectives of this work, namely to certify a complete decision procedure for the full first order theory of real closed fields. Yet incomplete but lightweight tools could probably be crafted to relieve the user from pedestrian steps when possible.

7.5 Current state of the formalization

In this section, we recapitulate the main results described in this paper that have a formal proof in our development.

- The absolute values of the real roots of a polynomial is bounded by the Cauchy bound, which is expressed only using the absolute values of the coefficients of the polynomial.
- If a polynomial function p has a negative value in x and a positive value in y, with $x < y$, then for any ε one can exhibit x' and y' so that $-\varepsilon < p(x') < p(y') < \varepsilon$.
- If a polynomial has only one sign change in its coefficients for the standard monomial basis, then this polynomial has exactly one root between 0 (excluded) and $+\infty$.
- If a polynomial has only one sign change in its Bernstein coefficients for a given interval (l, r), then this polynomial has exactly one root between l and r (excluded).
- The inverse images of monomials are the Bernstein polynomials divided by the corresponding binomial coefficient.
- De Casteljau's algorithm computes correctly the Bernstein coefficients for the intervals (l, m) and (m, r) from the Bernstein coefficients for the interval (l, r).

This work is part of a more ambitious plan, aiming at providing an efficient procedure to isolate the roots of any polynomial. It remains to develop the connections between these various results that will constitute this procedure and its proof of correctness. To certify an algorithmically naive version of such a procedure, we still need to describe the proceure to reduce the multiplicity of roots (dividing by the greatest common divisor between the polynomial and its derivative) and to describe the termination of procedure based on successive dichotomy. The reduction to separable polynomials should not require too much effort considering the libraries already available in the SSReflect package. The study of termination might however require a subtantial work.

An other issue will be to connect the correctness proof of such a naive implementation with more realistic programs, like an implementation of de Casteljau linear in the degree of the input, as implemented in [Mah07] or even more optimized codes like the ones of [MRR05].

8 Conclusion

Real root isolation methods by sign changes based methods is a classical topic, extensively studied (see [RZ03] for a review of the related litterature) after

the pioneering work of Uspensky [Usp48]. Bernstein polynomials are used to provide efficient implementations of these methods [MRR05, RZ03]. To our knowledge, this work is the first mechanized proofs of de Casteljau algorithm,and of the building blocks of a real root isolation procedure. The closest work to ours is probably the study of global optimization methods in Coq lead by Roland Zumkeller [Zum08]. Indeed, Bernstein polynomial bases are also used to approximate continuous functions on a closed domain. This last work results in an implementation in the Coq system of tool to find optimums of multivariate continuous real functions. Yet we could not find a mention of a formalization of the correctness proofs of this tool.

This work on Bernstein polynomials combines techniques coming from analysis, algebra, and geometry. For instance, the properties of reversing the list of coefficients of a polynomial are studied by looking at the polynomial as a function from rational numbers to rational numbers. Similarly, the proof of Descartes' law of signs works by looking at functions and bounds on their values in various intervals. On the other hand, the definition of Bernstein coefficients relies on concepts that come from linear algebra: vector spaces, bases, or morphisms. Last, de Casteljau's algorithm relies on geometry with midpoints, or segments. It is particularly exciting that we can now study formally mathematical algorithms that use all these aspects of mathematics.

This development is not made just for the beauty of it. The initial goal is to provide one of the basic blocks required for cylindrical algebraic decomposition [BPR06, Mah07]. In the short term, we want to complete this into a full algorithm to isolate the roots of an arbitrary polynomial. This involves proving the technique to reduce the multiplicity of roots that we already described in the introduction, initializing the search xfor roots with an interval large enough to contain all the roots, xprogramming the recursive dichotomy process, and proving that this process always terminates.

For the proof of termination, we already know a mathematical argument, described in (book reference here) under the name "theorem of three circles". However, this theorem uses arguments based on complex numbers and we wish to find a more elementary proof, as we still want to express our result using mainly rational numbers. Our proof of the law of signs already is a more elementary one than the ones found in the literature. This point is debatable. The concept that we use to describe the existence of roots are contrived, because we restrict ourselves to manipulating rational numbers. This is reasonable if we consider that our development is a stepping stone in the path towards defining algebraic numbers.

Having Bernstein polynomials also makes it possible to consider adding plotting facilities to the theorem proving tool. Thus we could develop a tool to study mathematics where users could easily visualize the curves or surfaces associated to the objects they define, define new objects by direct graphical manipulation, and prove properties in the same environment. The idea is tempting, but it is not obvious that the high quality brought by formal verification is needed for such an application.

In the long run, a good knowledge of Bernstein polynomials and coefficients opens the door to a wide variety of tools that are commonplace in computer aided design and possibly robotics. Splines and Bezier curves which are often used in drawing tools share a lot of properties with Bernstein control points. Thus, we can envision that theorem provers equipped with a library on Bern-

stein coefficients could be useful to reason on designs, for instance to check that several parts do not collide or that some surface has the right topological properties. Concerning robotics, splines and Bezier curves can also be used to describe the trajectory of moving vehicles. Here, we can dream of a time where theorem proving may play a role in verifying that robots will not run into trouble or endanger people. Of course, not all geometrical objects are described using polynomials and many objects are often described in the computer as a collections of flat objects grouped together, like broken lines, triangulations, or more generally simplicial complexes. However, curvy objects are needed if we want to represent faithfully natural objects. It is exciting that we are getting closer to using formal methods on models of the real world.

References

[BC04] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development, Coq'Art:the Calculus of Inductive Constructions*. Springer-Verlag, 2004.

[Ber03] Yves Bertot. Simple canonical representation of rational numbers. *Electr. Notes Theor. Comput. Sci.*, 85(7), 2003.

[Béz86] Pierre Bézier. *Courbes et Surfaces*. Hermès, 1986.

[Bis67] Errett Bishop. *Foundations of Constructive Analysis*. McGraw-Hill Book Company, 1967.

[BPR06] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Rela ALgebraic Geometry*, volume 10 of *Algorithms and Computations in Mathematics*. Springer, second edition, 2006.

[CFGW04] Luís Cruz-Filipe, Herman Geuvers, and Freek Wiedijk. C-corn, the constructive coq repository at nijmegen. In Andrea Asperti, Grzegorz Bancerek, and Andrzej Trybulec, editors, *MKM*, volume 3119 of *Lecture Notes in Computer Science*, pages 88–103. Springer, 2004.

[dC85] Paul de Castleljau. *Formes à pôles*. Hermès, 1985.

[Des69] René Descartes. *Géométrie (1636). A source book in Mathematics*. Harvard University Press, 1969.

[GGMR09] François Garillot, Georges Gonthier, Assia Mahboubi, and Laurence Rideau. Packaging mathematical structures. In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel, editors, *TPHOLs*, volume 5674 of *Lecture Notes in Computer Science*, pages 327–342. Springer, 2009.

[GM08] Georges Gonthier and Assia Mahboubi. A Small Scale Reflection Extension for the Coq system. Research Report RR-6455, INRIA, 2008.

[GMR+07] Georges Gonthier, Assia Mahboubi, Laurence Rideau, Enrico Tassi, and Laurent Théry. A modular formalisation of finite group theory. In Klaus Schneider and Jens Brandt, editors, *TPHOLs*, volume 4732 of *Lecture Notes in Computer Science*, pages 86–101. Springer, 2007.

[Knu86] Donald Knuth. *Metafont: the Program.* Addison Wesley, 1986.

[Mah07] Assia Mahboubi. Implementing the cylindrical algebraic decomposition within the coq system. *Mathematical Structures in Computer Science*, 17(1):99–127, 2007.

[MRR05] Bernard Mourrain, Fabrice Rouillier, and Marie-Françoise Roy. Bernstein's basis and real root isolation. *Mathematical Sciences Research Institute Publications*, 2005.

[O'C07] Russell O'Connor. A monadic, functional implementation of real numbers. *Mathematical Structures in Computer Science*, 17(1):129–159, 2007.

[O'C08] Russell O'Connor. Certified exact transcendental real number computation in coq. In *TPHOLs*, Lecture Notes in Computer Science, pages 246–261. Springer, 2008.

[RZ03] Fabrice Rouillier and Paul Zimmermann. Efficient isolation of polynomial real roots. *Journal of Computational and Applied Mathematics*, 162(1):33–50, 2003.

[Usp48] James Victor Uspensky. *Theory of Equations.* MacGraw-Hill Bok Company, 1948.

[Zum08] Roland Zumkeller. *Global Optimization in Type Theory.* PhD thesis, École Polytechnique, 2008.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399