Lemmas for Satisfiability Modulo Transcendental Functions via Incremental Linearization (extended abstract)

Ahmed Irfan[1] Alessandro Cimatti[2] Alberto Griggio[2] Marco Roveri[2] Roberto Sebastiani[3]

[1]Stanford University, USA [2]Fondazione Bruno Kessler, Italy [3]University of Trento, Italy

Abstract

Incremental linearization is a conceptually simple, yet effective, technique that we have recently proposed for solving satisfiability problems over nonlinear real arithmetic constraints, including transcendental functions. A central step in the approach is the generation of linearization lemmas, constraints that are added during search to the SMT problem and that form a piecewise-linear approximation of the nonlinear functions in the input problem. It is crucial for both the soundness and the effectiveness of the technique that these constraints are valid (to not remove solutions) and as general as possible (to improve their pruning power). In this extended abstract, we provide more details about how linearization lemmas are generated for transcendental functions, including proofs of their soundness. Such details, which were missing in previous publications, are necessary for an independent reimplementation of the method.

1 Introduction

The field of Satisfiability Modulo Theories (SMT) has seen tremendous progress in the last decade. Nowadays, powerful and effective SMT solvers are available for a number of quantifier-free theories and their combinations, such as equality and uninterpreted functions, bit-vectors, arrays, and linear arithmetic. A fundamental challenge is to go beyond the linear case, by introducing nonlinear polynomials and transcendental functions.

Recently, we have proposed a conceptually simple, yet effective approach for dealing with the quantifier-free theory of nonlinear arithmetic over the reals, called *Incremental Linearization* [2, 3, 4, 5]. Its underlying idea is that of trading the use of expensive, exact solvers for nonlinear arithmetic for an abstraction-refinement loop on top of much less expensive solvers for linear arithmetic and uninterpreted functions. The approach is based on an abstraction-refinement loop, in which the input problem is overapproximated using an SMT formula in which nonlinear multiplications and transcendental functions are treated as uninterpreted. The initial approximation is then iteratively refined, by incrementally adding *linearization lemmas*, i.e. constraints that form a piecewise-linear approximation of the nonlinear functions in the input problem. How such lemmas are actually generated is a central aspect of the incremental linearization method. Although most of the details have been provided in previous publications (most notably [5]), some aspects of the lemma generation procedure for transcendental functions were only described at a high level of abstraction, leaving a gap between the algorithmic description and the actual implementation that might make an independent reimplementation of the method difficult. In

In: J. Abbott, A. Griggio (eds.): Proceedings of the 4th SC-square Workshop, Bern, Switzerland, 10th July 2019, published at http://ceur-ws.org

this extended abstract, we close this gap, by providing the missing details about how linearization lemmas are generated for transcendental functions, including proofs of their soundness.

2 Background

2.1 Basic definitions

We assume the standard first-order quantifier-free logical setting and standard notions of theory, satisfiability, and logical consequence. We denote formulas with φ, ψ, terms with t, variables with x, y, a, b, functions with $f, \text{TF}, f_{\text{TF}}$, each possibly with subscripts. If μ is a model and x is a variable, we write $\mu[x]$ to denote the value of x in μ, and we extend this notation to terms and formulas in the usual way.

A *transcendental function* (TF) is an analytic function that does not satisfy a polynomial equation (in contrast to an algebraic function [8, 6]). Within this paper we consider univariate exponential, logarithmic, and trigonometric functions. We denote with NTA the theory of non-linear real arithmetic (NRA) extended with these transcendental functions.

A *tangent line* to a univariate function $f(x)$ at a point of interest $x = a$ is a straight line that "just touches" the function at the point, and represents the instantaneous rate of change of the function f at that one point. The tangent line $\text{TanLine}_{f,a}(x)$ to the function f at point a is the straight line defined as follows:

$$\text{TanLine}_{f,a}(x) \doteq f(a) + \frac{d}{dx} f(a) * (x - a)$$

where $\frac{d}{dx} f$ is the first-order derivative of f wrt. x.

A *secant line* to a univariate function $f(x)$ is a straight line that connects two points on the function plot. The secant line $\text{SecLine}_{f,a,b}(x)$ to a function f between points a and b is defined as follows:

$$\text{SecLine}_{f,a,b}(x) \doteq \frac{f(a) - f(b)}{a - b} * (x - a) + f(a).$$

For a function f that is twice differentiable at point c, the *concavity* of f at c is the sign of its second derivative evaluated at c. We denote open and closed intervals between two real numbers l and u as $]l, u[$ and $[l, u]$ respectively. Given a univariate function f over the reals, the *graph* of f is the set of pairs $\{\langle x, f(x)\rangle \mid x \in \mathbb{R}\}$. We might sometimes refer to an element $\langle x, f(x)\rangle$ of the graph as a point.

Proposition 1 *Let f be a univariate function. If $f''(x) > 0$ for all $x \in [l, u]$, then for all $a, x \in [l, u]$ $\text{TanLine}_{f,a}(x) \leq f(x)$, and for all $a, b, x \in [l, u]$ $((a \neq b \land a \leq x \leq b) \rightarrow \text{SecLine}_{f,a,b}(x) \geq f(x))$.*

If $f''(x) < 0$, then the dual property holds.

Taylor Series and Taylor's Theorem.

Given a function $f(x)$ that has $n + 1$ continuous derivatives at $x = a$, the *Taylor series* of degree n centered around a is the polynomial:

$$P_{n,f,a}(x) \doteq \sum_{i=0}^{n} \frac{f^{(i)}(a)}{i!} * (x - a)^i$$

where $f^{(i)}(a)$ is the evaluation of i-th derivative of $f(x)$ at point $x = a$. The Taylor series centered around 0 is also called *Maclaurin series*.

According to *Taylor's theorem*, any continuous function $f(x)$ that is $n+1$ differentiable can be written as the sum of the Taylor series and the remainder term:

$$f(x) = P_{n,f,a}(x) + R_{n+1,f,a}(x)$$

where $R_{n+1,f,a}(x)$ is basically the Lagrange form of the remainder, and for some point b between x and a it is given by:

$$R_{n+1,f,a}(x) \doteq \frac{f^{(n+1)}(b)}{(n + 1)!} * (x - a)^{n+1}.$$

```
bool INCREMENTAL-LINEARIZATION (φ):
    φ̂ = INITIAL-ABSTRACTION(φ)
    Γ = ∅
    precision := INITIAL-PRECISION ()
    while true:
        if BUDGET-EXHAUSTED ():
            abort
        ⟨res, μ̂⟩ = CHECK-LINEAR (φ̂ ∧ ⋀Γ)
        if not res:
            return false
        ⟨sat, Γ′⟩ := CHECK-REFINE (φ, μ̂, precision)
        if sat:
            return true
        else:
            precision := MAYBE-INCREASE-PRECISION ()
            Γ′′ := REFINE-EXTRA (φ, μ̂)
            Γ := Γ ∪ Γ′ ∪ Γ′′
```

Figure 1: Solving SMT(NTA) via incremental linearization.

The value of the point b is not known, but the upper bound on the size of the remainder $R^{\text{U}}_{n+1,f,a}(x)$ at a point x can be estimated by:

$$R^{\text{U}}_{n+1,f,a}(x) \doteq \max_{c\in[\min(a,x),\max(a,x)]}(|f^{(n+1)}(c)|) * \frac{|(x-a)^{n+1}|}{(n+1)!}.$$

This allows to obtain two polynomials that are above and below the function at a given point x, by considering $P_{n,f,a}(x) + R^{\text{U}}_{n+1,f,a}(x)$ and $P_{n,f,a}(x) - R^{\text{U}}_{n+1,f,a}(x)$ respectively.

2.2 Incremental linearization

The main incremental linearization algorithm is shown in Fig. 1. In the following, we summarize its main steps. For the full details, we refer the reader to [5]. The solving procedure follows a classic abstraction-refinement loop. Initially, all non-linear multiplications and transcendental functions are treated as uninterpreted functions, resulting in a formula $\widehat{\varphi}$ over the theory of *linear* arithmetic and uninterpreted functions. Note that also π is treatead as an uninterpreted constant.[1] Then, at each iteration the current safe approximation $\widehat{\varphi}$ of the input formula φ is refined by adding new constraints Γ that rule out one (or possibly more) spurious solutions, until one of the following conditions occurs: (i) the resource budget (e.g. time, memory, number of iterations) is exhausted; or (ii) $\widehat{\varphi} \wedge \bigwedge\Gamma$ becomes unsatisfiable in the theory of linear arithmetic and uninterpreted functions (UFLRA); or (iii) the satisfiability result (in UFLRA) for $\widehat{\varphi} \wedge \bigwedge\Gamma$ can be lifted to a satisfiability result for the original formula φ. An initial current precision is set (calling the function INITIAL-PRECISION), and this value is possibly increased at each iteration (calling MAYBE-INCREASE-PRECISION) according to the result of CHECK-REFINE and some heuristic.

The core of the procedure is the CHECK-REFINE function, shown in Fig. 2. First, if the formula contains also some non-linear polynomials, CHECK-REFINE performs the refinement of non-linear multiplications as described in [5]. Then, the function iterates over all the transcendental function applications TF(x) in φ (lines 3–7), and checks whether the model $\widehat{\mu}$ computed for $\widehat{\varphi}$ is consistent with their semantics.

Intuitively, in principle, this amounts to check that TF$(\widehat{\mu}[x])$ is equal to $\widehat{\mu}[f_{\text{TF}}(x)]$. In practice, however, the check cannot be exact, since transcendental functions at rational points typically have irrational values (see e.g. [7]), which cannot be represented exactly in an SMT solver for linear arithmetic. Therefore, for each TF(x) in φ, we instead compute two polynomials, $P_l(x)$ and $P_u(x)$, with the property that TF$(\widehat{\mu}[x])$ belongs to the open interval $]P_l(\widehat{\mu}[x]), P_u(\widehat{\mu}[x])[$. The polynomials are computed using Taylor series, according to the given current precision, by the function GET-POLYNOMIAL-BOUNDS. If the model value $\widehat{\mu}[f_{\text{TF}}(x)]$ for TF(x) is outside the above interval, then the function BLOCK-SPURIOUS-NTA-TERM is used to generate some linear lemmas that will remove the spurious point $\langle\widehat{\mu}[x], \widehat{\mu}[f_{\text{TF}}(x)]\rangle$ from the graph of the current abstraction of TF(x) (line 7).

If at least one point was refined in the loop of lines 3–7, the current set of lemmas Γ is returned (line 10). If instead none of the points was determined to be spurious, the function CHECK-MODEL is called (line 9). This

[1] In this case, we add a constraint $l_\pi \leq \pi \leq u_\pi$, for two suitable values l_π and u_π, which are refined when needed (see §5).

```
⟨bool, lemma-set⟩ CHECK-REFINE (φ, μ̂, precision):
1.  Γ := CHECK-REFINE-NRA (φ, μ̂)  # NRA refinement of [2]
2.  ε := 10^−precision
3.  for all TF(x) ∈ φ:
4.      c := μ̂[x]
5.      ⟨P_l(x), P_u(x)⟩ := GET-POLYNOMIAL-BOUNDS (TF(x), c, ε)
6.      if μ̂[f_TF(x)] ≤ P_l(c) or μ̂[f_TF(x)] ≥ P_u(c):
7.          Γ := Γ∪ BLOCK-SPURIOUS-NTA-TERM (TF(x), μ̂, P_l(x), P_u(x))
8.  if Γ = ∅:
9.      if CHECK-MODEL (φ, μ̂):
10.         return ⟨true, ∅⟩
11.     else:
12.         return CHECK-REFINE (φ, μ̂, precision+1)
13. else:
14.     return ⟨false, Γ⟩
```

Figure 2: The main refinement procedure.

function tries to determine whether the abstract model $\widehat{\mu}$ does indeed imply the existence of a model for the original formula φ. If the check fails, we repeat the CHECK-REFINE call with an increased precision (line 12).

Refining a spurious point with secant and tangent lines.

Given a transcendental function application $\text{TF}(x)$, the BLOCK-SPURIOUS-NTA-TERM function generates a set of lemmas for refining the interpretation of $f_{\text{TF}}(x)$ by constructing a piecewise-linear approximation of $\text{TF}(x)$ around the point $\widehat{\mu}[x]$, using one of the polynomials $P_l(x)$ and $P_u(x)$ computed in CHECK-REFINE. The kind of lemmas generated, and which of the two polynomials is used, depend on (i) the position of the spurious value $\widehat{\mu}[f_{\text{TF}}(x)]$ relative to the correct value $\text{TF}(\widehat{\mu}[x])$, and (ii) the concavity of TF around the point $\widehat{\mu}[x]$. If the concavity is negative (resp. positive) and the point is below (resp. above) the function, the linear approximation is given by a pair of secants to the lower (resp. upper) bound polynomial P_l (resp. P_u) around $\widehat{\mu}[x]$ (lines 4–16 of Fig. 3). Otherwise, i.e. the concavity is positive (resp. negative) or equal to zero, and the point lies below (resp. above) the function, then the linear approximation is given by a tangent to the lower (resp. upper) bound polynomial P_l (resp. P_u) at $\widehat{\mu}[x]$ (lines 17–22 of Fig. 3). The two situations are illustrated in Fig. 4.

In the case of secant refinement, a second value, different from $\widehat{\mu}[x]$, is required to draw a secant line. The function GET-PREVIOUS-SECANT-POINTS returns the set of all the points at which a secant refinement was performed in the past for $\text{TF}(x)$. From this set, we take the two points closest to $\widehat{\mu}[x]$, such that $l < \widehat{\mu}[x] < u$ and that l, u do not cross any inflection point, and use those points to generate two secant lines and their validity intervals. Before returning the set of the two corresponding lemmas, we also store the new secant refinement point $\widehat{\mu}[x]$ by calling STORE-SECANT-POINT.

In the case of tangent refinement, the function GET-TANGENT-BOUNDS (line 20) returns a *validity interval* for the lemma, i.e. an interval $[l, u]$ such that the tangent line is guaranteed not to cross the transcendental function TF. In order to generate strong lemmas that can prune the search space effectively, it is important that this validity interval is as large as possible. At the same time, it is critical for correctness that the interval is not too large, i.e. that no intersection exists between the transcendental function and its tangent at c in $[l, u]$. We shall describe in detail how such validity intervals are computed for the transcendental functions that are currently supported by our implementation (namely exp and sin) in the next two Sections.

3 Lemmas for exp

3.1 Polynomial Approximation

Since $\frac{d}{dx}\exp(x) = \exp(x)$, all the derivatives of exp are positive. The polynomial $P_{n,\exp,0}(x)$ is given by the Maclaurin series

$$P_{n,\exp,0}(x) = \sum_{i=0}^{n} \frac{x^i}{i!}$$

and behaves differently depending on the sign of x. Thus, GET-POLYNOMIAL-BOUNDS (see Fig. 2) distinguishes two cases for finding the polynomials $P_l(x)$ and $P_u(x)$:[2]

[2]The case $x = 0$ is treated specially, by adding the lemma $\exp(0) = 1$ (see [5]).

lemma-set BLOCK-SPURIOUS-NTA-TERM (TF(x), $\widehat{\mu}$, $P_l(x)$, $P_u(x)$):

$c := \widehat{\mu}[x]$
$v := \widehat{\mu}[f_{\mathrm{TF}}(x)]$
conc := GET-CONCAVITY (TF(x), c)
if ($v \leq P_l(c)$ **and** conc < 0) **or** ($v \geq P_u(c)$ **and** conc > 0):
  # secant refinement
    prev := GET-PREVIOUS-SECANT-POINTS (TF(x))
    $l := \max\{p \in \text{prev} \mid p < c\}$
    $u := \min\{p \in \text{prev} \mid p > c\}$
    $P := (v \leq P_l(c))\ ?\ (P_l)\ :\ (P_u)$
    $S_l(x) := \dfrac{P(l) - P(c)}{l - c} \cdot (x - l) + P(l)$ *# secant of P between l and c*
    $S_u(x) := \dfrac{P(u) - P(c)}{u - c} \cdot (x - u) + P(u)$
    $\psi_l := (\text{conc} < 0)\ ?\ (f_{\mathrm{TF}}(x) \geq S_l(x))\ :\ (f_{\mathrm{TF}}(x) \leq S_l(x))$
    $\psi_u := (\text{conc} < 0)\ ?\ (f_{\mathrm{TF}}(x) \geq S_u(x))\ :\ (f_{\mathrm{TF}}(x) \leq S_u(x))$
    $\phi_l := (x \geq l) \wedge (x \leq c)$
    $\phi_u := (x \geq c) \wedge (x \leq u)$
    STORE-SECANT-POINT (TF(x), c)
    return $\{(\phi_l \rightarrow \psi_l), (\phi_u \rightarrow \psi_u)\}$
else: *# ($v \leq P_l(c)$ and conc ≥ 0) or ($v \geq P_u(c)$ and conc ≤ 0)*
  # tangent refinement
    $P := (v \leq P_l(c))\ ?\ (P_l)\ :\ (P_u)$
    $T(x) := P(c) + \frac{d}{dx}P(c) \cdot (x - c)$ *# tangent of P at c*
    $\langle l, u \rangle :=$ GET-TANGENT-BOUNDS (TF(x), c, $\frac{d}{dx}P(c)$)
    $\psi := (\text{conc} < 0)\ ?\ (f_{\mathrm{TF}}(x) \leq T(x))\ :\ (f_{\mathrm{TF}}(x) \geq T(x))$
    return $\{((x \geq l) \wedge (x \leq u)) \rightarrow \psi\}$

Figure 3: Piecewise-linear refinement for the transcendental function TF(x) at point c.

Case $x < 0$: we have that $P_{n,\exp,0}(x) < \exp(x)$ if n is odd and $n \geq 3$, and $P_{n,\exp,0}(x) > \exp(x)$ if n is even and $n \geq 3$; we therefore set $P_l(x) = P_{n,\exp,0}(x)$ and $P_u(x) = P_{n+1,\exp,0}(x)$ for a suitable n so that the required precision ϵ is met;

Case $x > 0$: we have that $P_{n,\exp,0}(x) < \exp(x)$ and $P_{n,\exp,0}(x) * (1 - \frac{x^{n+1}}{(n+1)!})^{-1} > \exp(x)$ when $(1 - \frac{x^{n+1}}{(n+1)!}) > 0$, therefore we set $P_l(x) = P_{n,\exp,0}(x)$ and $P_u(x) = P_{n,\exp,0}(x) * (1 - \frac{x^{n+1}}{(n+1)!})^{-1}$ for a suitable $n \geq 3$. [3]

Since the concavity of exp is always positive, the tangent refinement will always give lower bounds for $\exp(x)$, and the secant refinement will give upper bounds (see Fig. 3).

3.2 Tangent Validity Interval

Lemma 1 *Let $c \in \mathbb{Q}$ and $c \neq 0$. If the concavity of P_l at point c is positive, then the concavity of P_l is also positive for $x \in [c, \infty]$.*

Proof. We know $P_l(x) = P_{n,\exp,0}(x)$ and its second-order derivative is $P_l''(x) = P_{n-2,\exp,0}(x)$. Let us suppose that the concavity of P_l at a point c is positive, i.e. $P_l''(c) > 0$. We now show that the concavity of P_l remains postive in the interval $[c, \infty]$. We split the proof into two cases: $c > 0$ and $c < 0$.

Case $c > 0$: The statement holds because P_l'' is an increasing function – P_l''' is positive in the interval $[c, \infty]$.

Case $c < 0$: The statement also holds because P_l'' is an increasing function – $P_l'''(x) = P_{n-3,\exp,0}(x)$ is an even-degree polynomial and by Taylor's Theorem we know that it is greater than exp, which is always positive. □

Lemma 2 *The validity interval for the tangent refinement of* exp *is $[-\infty, \infty]$.*

Proof. The tangent refinment of exp is done by constructing a tangent line at a point c (TANLINE$_{P_l,c}(x)$) to P_l. Due to the way we construct P_l, the concavity of P_l is ensured to match with the concavity of exp (meaning the concavity is positive) at c. Let I_+ denote the interval $[c, \infty]$ and I_- denote the interval $[-\infty, c[$.

[3] We slightly abuse the notation: $P_u(x)$ is not a polynomial but a rational function.

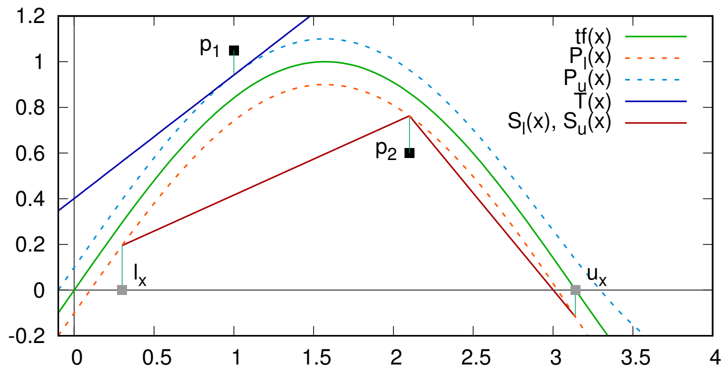

Figure 4: Piecewise-linear refinement illustration.

Case $x \geq c$: by Lemma 1 we can conclude that the concavity of $P_l(x)$ will remain positive in I_+. $\textsc{TanLine}_{P_l,c}(x)$ will be below P_l in I_+, and therefore it will be also below exp in I_+.

Case $x < c$ and $c > 0$: We know that the concavity of P_l is positive in the interval $[0, \infty]$ and $\textsc{TanLine}_{P_l,c}(x)$ will be below exp. Moreover, the slope $P_l'(c)$ is always greater than 1, which is greater than the slope $\exp(d)$ for every $d < 0$. And moving towards $-\infty$ the slope of exp is decreasing, while the slope of $\textsc{TanLine}_{P_l,c}(x)$ is constant and its values are going to decrease quicker than exp. Moreover, $\textsc{TanLine}_{P_l,c}(x)$ will cross the origin whereas exp stays positive and converges to zero at $-\infty$.

Case $x < c$ and $c < 0$: $P_l = P_{n,\exp,0}(x)$ with n being an odd number. The first-order derivative of P_l is $P_l' \doteq P_{n-1,\exp,0}(x)$, which is above exp. This means that the slope of $\textsc{TanLine}_{P_l,c}(x)$ is greater than the exp slope. So, $\textsc{TanLine}_{P_l,c}(x)$ will be below exp (same argument as above). $\square$

Thanks to Lemma 2, therefore, the implementation of GET-TANGENT-BOUNDS for exp can always return the interval $[-\infty, \infty]$.

4 Lemmas for sin

The correctness of our refinement procedure relies crucially on being able to compute the concavity of the transcendental function TF at a given point c. This is needed in order to know whether a computed tangent or secant line constitutes a valid upper or lower bound for TF around c (see 3). In the case of the sin function, computing the concavity at an arbitrary point c is problematic, since this essentially amounts to computing the value $c' \in [-\pi, \pi[$ s.t. $c = 2\pi n + c'$ for some integer n, because in $[-\pi, \pi[$ the concavity of $\sin(c')$ is the opposite of the sign of c'. This is not easy to compute because π is a transcendental number.

In order to solve this problem, we exploit another property of sin, namely its periodicity (with period 2π). More precisely, we split the reasoning about sin depending on two kinds of periods: base period, in the interval $[-\pi, \pi]$, and extended period (everywhere else in $\mathbb{R}$). For more details about the extended period, we refer to Section 4.2.2 in [5]. Here, we focus only on the base period, where all the refinement lemmas are instantiated. We can easily compute the concavity of sin in the base period by just looking at the sign of $\widehat{\mu}[x]$, provided that $-l_\pi \leq \widehat{\mu}[x] \leq l_\pi$, where l_π is the current lower bound for π.

4.1 Polynomial Approximation

For each term $\sin(x)$ that needs to be refined, we first check whether $\widehat{\mu}[x] \in [-l_\pi, l_\pi]$, where l_π is the current lower bound for π. If this is the case, then we derive the concavity of sin at $\widehat{\mu}[x]$ by just looking at the sign of $\widehat{\mu}[x]$. We can therefore perform tangent or secant refinement as shown in Fig. 3. More precisely, GET-POLYNOMIAL-BOUNDS finds the lower and upper polynomials using Taylor's theorem, which ensures that:

$$P_{n,\sin,0}(x) - R^{\text{U}}_{n+1,\sin,0}(x) \leq \sin(x) \leq P_{n,\sin,0}(x) + R^{\text{U}}_{n+1,\sin,0}(x)$$

where

$$P_{n,\sin,0}(x) = \sum_{k=0}^{n} \frac{(-1)^k * x^{2k+1}}{(2k+1)!}$$
$$R^{\text{U}}_{n+1,\sin,0}(x) = \frac{x^{2(n+1)}}{(2(n+1))!}$$

We set $P_l(x) = P_{n,\sin,0}(x) - R^{\text{U}}_{n+1,\sin,0}(x)$ and $P_u(x) = P_{n,\sin,0}(x) + R^{\text{U}}_{n+1,\sin,0}(x)$.

The remaining case to discuss is when the value of x in $\widehat{\mu}$ is not within the interval $[-l_\pi, l_\pi]$ (which means that $|\widehat{\mu}[x]| \in (l_\pi, u_\pi]$). In this case, we cannot reliably compute the concavity of sin at $\widehat{\mu}[x]$. Therefore, we refine the approximation for π instead, as described in §5.

4.2 Tangent Validity Interval

Lemma 3 *The concavity of P_u matches with the concavity of* sin *in the interval* $]0, \frac{\pi}{2}]$.

Proof. The concavity of sin is negative in the interval $]0, \pi[$. At a point of interest $c > 0$, we ensure that $P_u''(c) < 0$. This requires that we expand the Taylor series with $n > 0$, so that the resulting polynomial has degree strictly greater than 2. Using interval arithmetic, we can easily show that P_u'' is always negative in the interval $]0, \frac{\pi}{2}]$ [4]. □

Lemma 4 *The concavity of P_l matches with the concavity of* sin *in the interval* $[-\frac{\pi}{2}, 0[$.

Proof. Similar to the proof of Lemma 3. □

Lemma 5 *Let $c \in\ ]0, \pi[$. If the concavity of P_u at point c is negative, then the concavity of P_u is also negative in the interval* $]0, c]$.

Proof. Case $c \leq \frac{\pi}{2}$: the statement holds due to Lemma 3.

Case $c > \frac{\pi}{2}$: We know $P_u(x) = \sum_{k=0}^{n} \frac{(-1)^k * x^{2k+1}}{(2k+1)!} + \frac{x^{2(n+1)}}{(2(n+1))!}$, and $P_u''(x) = \sum_{k=0}^{n-2} \frac{(-1)^{k+1} * x^{2k+1}}{(2k+1)!} + \frac{x^{2(n+1)}}{(2(n+1))!}$. Moreover, $P_u''(c) < 0$. The derivative of P_u'' is given by $P_u'''(x) = \sum_{k=0}^{n-2} \frac{(-1)^{k+1} * x^{2k}}{(2k)!} + \frac{x^{2n+1}}{(2n+1)!}$, which can be rewritten as $P_u'''(x) = -(P_{n-2,\cos,0}(x) - R^{\text{U}}_{n-1,\cos,0}(x))$. By Taylor's Theorem, we know $(P_{n-2,\cos,0}(x) - R^{\text{U}}_{n-1,\cos,0}(x)) < \cos(x)$ and so $P_u'''(x) > -\cos(x)$. Since $-\cos(x) > 0$ in the interval $]\frac{\pi}{2}, \pi[$, we can conclude that P_u'' is an increasing function. Given that $P_u''(c) < 0$ and we can conclude P_u'' will also be negative in the interval $]0, c[$. □

Lemma 6 *Let $c \in\ ]-\pi, 0[$. If the concavity of P_l at point c is positive, then the concavity of P_l is also positive in the interval* $[c, 0[$.

Proof. Similar to the proof of Lemma 5. □

Lemma 7 *Let $I_+ \doteq\ ]0, \pi[$. The validity bounds for the tangent refinement of* sin *in I_+ is I_+.*

Proof. The tangent refinement of sin in I_+ is done by constructing a tangent line ($\textsc{TanLine}_{P_u,c}(x)$) at a point $c \in I_+$ to P_u. Due to the way we construct P_u, the concavity of P_u is ensured to match with the concavity of sin (meaning the concavity is negative) at c.

Case $x \leq c$: By Lemma 5, we can conclude that the concavity of $P_u(x)$ remains negative. $\textsc{TanLine}_{P_u,c}(x)$ is above P_u, and therefore it is also above sin.

Case $x > c$ and $c \leq \frac{\pi}{2}$: Due to Lemma 3, the concavity of P_u matches with the concavity of sin. So, $\textsc{TanLine}_{P_u,c}(x)$ is above in the interval $]0, \frac{\pi}{2}[$. Moreover, the slope of $\textsc{TanLine}_{P_u,c}(x)$ is positive and the line does not cross sin in the interval because the slope of sin is negative in $]\frac{\pi}{2}, \pi[$.

Case $x > c$ and $c > \frac{\pi}{2}$: The slope $P_u'(x) = \sum_{k=0}^{n} \frac{(-1)^k * x^{2k}}{(2k)!}$, which can be also rewritten as $P_u'(x) = P_{n,\cos,0}(x) + R^{\text{U}}_{n+1,\cos,0}(x)$. We know that $\sin'(x) = \cos(x)$ and by Taylor's Theorem $P_{n,\cos,0}(x) + R^{\text{U}}_{n+1,\cos,0}(x) \geq \cos(x)$. And

[4] Hint: Apply Horner's rule for P_u'' interval evaluation.

therefore $P'_u(x) \geq \sin'(x)$. The slope of sin keeps decreasing when moving from $\frac{\pi}{2}$ to π, whereas the slope of $\textsc{TanLine}_{P_u,c}(x)$ is greater than the slope of sin and is also constant. Therefore, sin will decrease and crosses the origin faster than the tangent line. □

Lemma 8 *Let* $I_- \doteq]-\pi, 0[$*. The validity interval for tangent refinement of* sin *in* I_- *is* I_-*.*

Proof. Similar to the proof of Lemma 7. □

Lemma 9 *The validity interval for tangent refinement is* $]0, \pi[$ *when* $x > 0$ *and* $]-\pi, 0[$ *when* $x < 0$*.*

Proof. The statement holds due to Lemma 7 and Lemma 8. □

Thanks to Lemma 9, therefore, the implementation of GET-TANGENT-BOUNDS for sin (see Fig. 3) returns the interval $]-\pi, 0[$ if the input tangent point c is negative, and $]0, \pi[$ if it is positive.[5]

5 Lemmas for π

The refinement of π is done by expanding the series given by Machin's formula [1]. We keep a *current precision* n (a positive integer) for π, which is used to update the bounds l_π and u_π using the inequalities (1) and (2) respectively:

$$\pi > 4 * \sum_{k=0}^{2n+1} \left(\frac{(-1)^k}{2k+1} * \left(\frac{4}{5^{2k+1}} - \frac{1}{239^{2k+1}} \right) \right) \tag{1}$$

$$\pi < 4 * \sum_{k=0}^{2(n+1)} \left(\frac{(-1)^k}{2k+1} * \left(\frac{4}{5^{2k+1}} - \frac{1}{239^{2k+1}} \right) \right). \tag{2}$$

Acknowledgement

We would like to thank Dejan Jovanovic and Andrew Reynolds for fruitful discussions about the tangent refinement, that motivated us to write this note.

References

[1] Berggren, J., Borwein, J., Borwein, P.: Pi: A Source Book. Springer New York (2013)

[2] Cimatti, A., Griggio, A., Irfan, A., Roveri, M., Sebastiani, R.: Invariant checking of NRA transition systems via incremental reduction to LRA with EUF. In: TACAS. LNCS, vol. 10205, pp. 58–75 (2017)

[3] Cimatti, A., Griggio, A., Irfan, A., Roveri, M., Sebastiani, R.: Satisfiability modulo transcendental functions via incremental linearization. In: de Moura, L. (ed.) CADE 26. LNCS, vol. 10395, pp. 95–113. Springer (2017)

[4] Cimatti, A., Griggio, A., Irfan, A., Roveri, M., Sebastiani, R.: Experimenting on solving nonlinear integer arithmetic with incremental linearization. In: SAT. Lecture Notes in Computer Science, vol. 10929, pp. 383–398. Springer (2018)

[5] Cimatti, A., Griggio, A., Irfan, A., Roveri, M., Sebastiani, R.: Incremental Linearization for Satisfiability and Verification Modulo Nonlinear Arithmetic and Transcendental Functions. ACM Trans. Comput. Log. **19**(3), 19:1–19:52 (2018). https://doi.org/10.1145/3230639

[6] Hazewinkel, M.: Encyclopaedia of Mathematics: Stochastic Approximation – Zygmund Class of Functions. Encyclopaedia of Mathematics, Springer Netherlands (1993)

[7] Niven, I.: Numbers: Rational and Irrational. Mathematical Association of America (1961)

[8] Townsend, E.: Functions of a Complex Variable. Read Books (2007)

[5] Note that the case of 0 is treated specially, by adding the lemma $\sin(0) = 0$ (see [5]).