Views of Pi: definition and computation

Yves Bertot
INRIA
and
Guillaume Allais
INRIA, Ecole Normale Supérieure de Lyon

We study several formal proofs and algorithms related to the number π in the context of Coq's standard library. In particular, we clarify the relation between roots of the *cosine* function and the limit of the alternated series whose terms are the inverse of odd natural numbers (known as Leibnitz' formula). We give a formal description of the arctangent function and its expansion as a power series. We then study other possible descriptions of π, first as the surface of the unit disk, second as the limit of perimeters of regular polygons with an increasing number of sides. In a third section, we concentrate on techniques to effectively compute approximations of π in the proof assistant by relying on rational numbers and decimal representations.

1. INTRODUCTION

The number π has been a fascinating object for mathematicians for many centuries. It is both a very concrete number and an abstract one. It is concrete because it is a simple ratio between either the respective surfaces of a circle and a square or between the perimeter of a circle and its diameter. It is abstract because its transcendantal status places it beyond the reach of most constructions starting from natural numbers (through integers, rational numbers, and algebraic numbers). For formalized mathematics, which thrive on elementary approaches to objects, this number represents a small challenge of its own.

There are several approaches to the π number. This variety of approaches imposes a dilemma on the teacher or the developer of a formalized mathematics library: what should be used as the definition, and what should be understood as provable properties of the number?

The investigations in this paper started with a study of the library for real numbers in the Coq system [4] between 2002 and 2012. In this library, the trigonometric functions were initially (in 2002) given by axiomatic properties, mainly about the sine and cosine of added angles, derivatives, and values in π. In 2003, definitions of sine and cosine as limits of power series were provided.

$$\sin x = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{2i+1!} \qquad \cos x = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i}}{2i!}.$$

At that point, most of the axioms were removed. With these definitions as limits of power series, these two functions seem to have no relation with the measure of triangles or with circles. However, two of the main properties were proved from that time:

(1) The value of $\cos(x+y)$ is $\cos x \cos y - \sin x \sin y$ (lemma named `cos_plus`).

(2) for a given x the sum $\sin^2 x + \cos^2 x$ is always 1 (lemma named `sin2_cos2`)

These two properties express that $(\cos x, \sin x)$ can really be understood as the Cartesian coordinates of a point on the unit circle, and that the value x can be viewed as an angle. It feels natural that the number π should be defined from the sin and cos functions directly, but instead the designer of the Reals library chose to define this number as a multiple of the limit of the infinite sum whose terms are the alternated inverse of odd numbers:

$$\pi \triangleq 4 \times (1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \cdots) = 4 \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1}$$

This formula revolves around the series expansion of atan. The history of this formula dates back to the 17th century in Europe and to the 14th century in India. In Europe, it is often referred to as Leibnitz' formula for π, but precedence can probably be given to James Gregory's work on the computation of surfaces of circles and hyperboles. In India, the whole theory of power series to approximate trigonometric functions and their inverses was probably established by Madhava of Sangamagrama.

The relation between this value and the surface of the circle is not immediate, and for this reason the properties of π with respect to the sin and cos functions is not immediate either. The initial developer of the library thus chose to leave the relation between π and the trigonometric functions as an axiom.

```
Axiom sin_PI2 : sin (PI / 2) = 1.
```

The main contribution in this paper is the description of the work performed to remove this axiom, in such a way that all facts that were formally proved for the value of π remain formally proved.

We performed this task by changing the definition of π: we showed that the cosine function changes sign between 1 and 2 and, using the intermediate value theorem, we proved the existence of a value α between 1 and 2 such that $\cos \alpha = 0$. We also showed that this value α was the unique root of cos in this interval. We then defined π as $\pi = 2\alpha$ and we were able to prove that $\sin \frac{\pi}{2} = 1$. Thus, the axiom was removed and replaced by a proved property, but the task was shifted to proving the equality:

$$\pi = 4 \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1}.$$

This proof relied on $\tan \frac{\pi}{4} = 1$, in other words $\pi = 4 \times \mathrm{atan}(1)$, using a study of derivatives for inverse functions in general and atan in particular, and on a power series expansion for atan's derivative:

$$\frac{\mathrm{d}(\mathrm{atan}\ x)}{\mathrm{d}x} = \frac{1}{1+x^2} = \sum_{i=0}^{\infty} (-1)^i x^{2i}.$$

Taking primitives on both sides of the equality gives the following equality:

$$\mathrm{atan}\ x = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{2i+1}.$$

And then the case for $x = 1$ yields the desired result. This piece of mathematics has to be done carefully, because the right-hand side actually diverges for values larger than 1.

After this first effort, we extended our reflection to other possible approaches to the number π and we added the formal proofs needed to show that these other approaches are equivalent.

(1) The first relies on computing the surface of the upper half of a circle centered in (0,0) with radius 1. It is simple to notice that the upper boundary of the circle is the graph of the function satisfying $x^2 + y^2 = 1$ with $0 \leq y$, in other words $y = \sqrt{1 - x^2}$, or more precisely we want to compute the integral of the function $x \mapsto \sqrt{1 - x^2}$. This approach requires that we formalize another reciprocal function, namely the arcsine function, but this is easily done after the ground work has been done for the arctangent.

(2) The second approach uses inscribed or tangential polygons to the circle, thus reproducing a process that was proposed by Archimedes. The computation of the perimeter for these polynomials requires some geometric reasoning that boils down to computing sine and tangent values for half angles, so this is naturally related to trigonometric functions.

Then we turned our attention to opportunities to compute good approximations of π with reasonable efficiency inside the Coq system. The description of π by Leibnitz' formula is not good for this purpose, because it takes $4 \times n$ terms of the infinite sum to get an approximation of π to a precision of $\frac{1}{n}$. However, the series for computing atan between -1 and 1 is the following one:

$$\text{atan } x = \sum_{i=0}^{\infty} \frac{(-1)^i}{(2i+1)} x^{2i+1}$$

It appears that this series converges pleasantly fast for values of x such that $|x| < 1$, while it converges slowly for $|x| = 1$. Because the series is alternated, it is also easy to show that truncated sums with an even number of initial terms will provide under-approximations, while truncated sums with an odd number of initial terms will provide over-approximations. We used this to construct a function that provides an interval with rational bounds containing $\text{atan}(x)$ when x is rational.

Then, we proved the simple formulas about atan:

$$\text{atan}(u) + \text{atan}(v) = \text{atan}\left(\frac{u+v}{1-uv}\right).$$

This makes it possible to compute atan(1) as the sum of values of atan in other rational positions, all of them smaller than 1. For instance, starting with $u = 1$ using $v = \frac{-1}{5}$ repeatedly, we can prove the equality used by Machin in 1706:

$$\frac{\pi}{4} = 4 \times \text{atan}\frac{1}{5} - \text{atan}\frac{1}{239}$$

We used this to define a function that returns certified approximations of π to a certain number of decimal places, by relying on exact computations of rational numbers.

2. REMOVING THE UNWANTED AXIOM

In this section, we concentrate on the path followed to provide a new definition of the number π that aims directly at removing the axiom and then on the path followed to provide a proof that the property that was used in the old definition is a consequence of the new definition.

2.1 Context information about series and trigonometry in the Coq library

The description of finite sums relies on a function `sum_f_R0` of type `(nat -> R) -> nat -> R`, when applied on a function f and a number n, this function computes the value

$$\sum_{i=0}^{n} f(i).$$

This computation is simply described by a recursive function on natural numbers. Note that the design choice imposes that the base case is not an empty sum, but the value $f(0)$.

To express that an infinite sum of real numbers converges to a limit l, the Coq library uses a predicate `infinite_sum` of type `(nat -> R) -> R -> Prop`. As one could expect, this predicate is based on `sum_f_R0` and relies on the same ε-N scheme as would be needed to express the convergence towards l of the sequence $(u_n)_{n\in\mathbb{N}}$ of partial sums defined as follows:

$$u_n = \sum_{i=0}^{n} f(i).$$

The usual concept to reason on this kind of convergence is `Un_cv`, a predicate of type `(nat -> R) -> R -> Prop`. Due to this strong similarity, the useful theorems are either theorems about the `infinite_sum` concept or theorems about the `Un_cv` concept. The definitions of `infinite_sum` and `Un_cv` are given as follows:

```
Definition infinite_sum (s:nat -> R) (l:R) : Prop :=
  forall eps:R,
    eps > 0 ->
    exists N : nat,
      (forall n:nat, (n >= N)%nat -> R_dist (sum_f_R0 s n) l < eps).

Definition Un_cv (Un : nat -> R) (l:R) : Prop :=
  forall eps:R,
    eps > 0 ->
    exists N : nat,
      (forall n:nat, (n >= N)%nat -> R_dist (Un n) l < eps).
```

Since `sum_f_R0` is described as a recursive function on natural numbers, this imposes that the index used for each term of the sum is given as a natural number. When this index is used in the body of the coefficient, it needs to be coerced into the type of real numbers. The main type coercion is given by the function `INR`, which describes the injection of natural numbers in the type of real numbers. This injection is itself described as a recursive function over the structure of the unary natural numbers effectively converting successor constructors in +1 operations. As a result, the formal transcription of power series is often cluttered with coercion functions, sometimes inconveniently distinguishing *obviously equal* values (e.g. INR $(m \times n)$ and INR $m \times$ INR n) thus forcing the user to explicitly manipulate them.

For instance, let's study how the sine function is described. The first step is to define the generic coefficient $\frac{(-1)^n}{(2n+1)!}$.

```
Definition sin_n (n:nat) : R := (-1) ^ n / INR (fact (2 * n + 1)).
```

The library relies on infinite_sum to state that the corresponding series converges towards a given limit thus having an infinite sum.

```
Definition sin_in (x l:R) : Prop :=
  infinite_sum (fun i:nat => sin_n i * x ^ i) l.
```

There is a general theorem, called *d'Alembert's criterion* (Alembert_C1 in Coq's library) stating that if $\left|\frac{a_{n+1}}{a_n}\right|$ converges towards 0, then for any x, $\sum_{n=0}^{\infty} a_n x^n$ converges. This is used to conclude that the power series defined by sin_n is a total function on $\mathbb{R}$:

```
Lemma exist_sin : forall x:R, { l:R | sin_in x l }.
```

However, this states the existence of a limit for the infinite sum

$$\sum_{i=0}^{\infty} \frac{(-1)^i}{(2i+1)!} x^i$$

while $\sin x$ is the following different sum:

$$\sin x = \sum_{i=0}^{\infty} \frac{(-1)^i}{(2i+1)!} x^{2i+1}$$

The two can be reconciled by using the following definition, where Rsqr denotes the squaring function.

```
Definition sin (x:R) := let (a,_) := exist_sin (Rsqr x) in x * a.
```

Since 2003, Coq's standard library for real numbers also contains a lemma expressing the known trigonometric formula for the cosine of the sum of two angles:

```
Theorem cos_plus :
   forall x y:R, cos (x + y) = cos x * cos y - sin x * sin y.
```

The proof of this lemma is ad hoc, but it could have been described as an instanciation of a more general lemma about the convergence of the Cauchy product of two series, a lemma known as Mertens' theorem[1]. A direct consequence of this theorem requiring only knowledge of sine and cosine's respective parities is that the sum of squares of sine and cosine for the same input is always equal to 1.

[1] We proved this theorem formally, but it has not been included in the library yet.

2.2 The axiom and its pervasive influence on proofs

Up to Coq version 8.4, there was an axiom stating $\sin\frac{\pi}{2} = 1$ in the standard library of Coq. This property is central in understanding the properties of the trigonometric functions. The first consequences, in combination with the lemmas `sin2_cos2` and `cos_plus` are

$$\cos\frac{\pi}{2} = 0 \qquad \sin x = -\cos(x + \frac{\pi}{2}).$$

Then obviously, as soon as one has a description of the derivative of the cosine function, one can deduce a description of the derivative of the sine function. The derivation formulas are important, because the derivative for the tangent function is obtained from the derivatives for the cosine and sine functions. In the organization of the standard library dating from 2003, the lemmas describing derivatives for these two functions both relied on the property $\sin\frac{\pi}{2} = 1$. But it is crucial to know the formula:

$$\frac{\mathrm{d}\tan x}{\mathrm{d}x} = 1 + \tan^2 x.$$

With this, we can establish that the arctangent function has derivative $x \mapsto \frac{1}{1+x^2}$, which is useful to prove the power series formula for arctangent.

2.3 The new definition of π

In this section, we will show how to describe $\frac{\pi}{2}$ as the value between 1 and 2 where cos is 0. Our first step will be to show that such a value exists.

D'Alembert's criterion is very powerful, as it guarantees absolute convergence over the whole real line. However, we could also see that the series for $\sin x$ and $\cos x$ are alternated series when x is positive. These types of series converge as soon as the sequence of terms $a_n x^n$ is decreasing in absolute value, and converges towards 0. Moreover, the term $|a_n x^n|$ gives a bound to the value of $\Sigma_{i=n+1}^{\infty} a_i x^i$. So we define a function for approximating values of the cosine and sine based on these results. Basically, `sin_approx x n` computes the first terms of the series for the sine function up to the term of exponent $2n+1$ and `cos_approx x n` computes the first terms of the series for the cosine function up to the term of exponent $2n$.

```
Definition sin_term (a:R) (i:nat) : R :=
 (-1) ^ i * (a ^ (2 * i + 1) / INR (fact (2 * i + 1))).

Definition cos_term (a:R) (i:nat) : R :=
  (-1) ^ i * (a ^ (2 * i) / INR (fact (2 * i))).

Definition sin_approx (a:R) (n:nat) : R := sum_f_R0 (sin_term a) n.

Definition cos_approx (a:R) (n:nat) : R := sum_f_R0 (cos_term a) n.
```

We have two theorems to express that these approximation functions make it possible to find bounds around $\sin x$ and $\cos x$ respectively. These theorems provide bounds only because the absolute value `sin_term x` is not guaranteed to be decreasing for any `x`, it is only ultimately decreasing when `x` is very large. When `x` is small enough, the decreasing property is given from the first terms.

```
Theorem pre_sin_bound :
  forall (a:R) (n:nat),
    0 <= a -> a <= 4 ->
  sin_approx a (2 * n + 1) <= sin a <= sin_approx a (2 * (n + 1)).

Lemma pre_cos_bound :
  forall (a:R) (n:nat),
    - 2 <= a -> a <= 2 ->
    cos_approx a (2 * n + 1) <= cos a <= cos_approx a (2 * (n + 1)).
```

When `a` is a rational number, the bounds provided by both theorems are rational numbers, too.

We use these theorems to find bounds for $\sin \frac{7}{8}$ and $\cos \frac{7}{8}$ and thus establish that $\cos 78$ is positive and $\cos 74$ is negative, in preparation for a use of the *intermediate value theorem*. With $n = 0$, the theorem `pre_sin_bound` makes it possible to construct a proof for the following comparisons, since $3 = 2 \times (2 \times 0 + 1) + 1$ and $5 = 2 \times (2 \times (0 + 1)) + 1$.

$$\frac{7}{8} - \frac{1}{3!}\left(\frac{7}{8}\right)^3 \leq \sin\frac{7}{8} \leq \frac{7}{8} - \frac{1}{3!}\left(\frac{7}{8}\right)^3 + \frac{1}{5!}\left(\frac{7}{8}\right)^5.$$

Again with $n = 0$ the theorem `pre_cos_bound` makes it possible to construct a proof for the following comparisons:

$$1 - \frac{1}{2}\left(\frac{7}{8}\right)^2 \leq \cos\frac{7}{8} \leq 1 - \frac{1}{2}\left(\frac{7}{8}\right)^2 + \frac{1}{4!}\left(\frac{7}{8}\right)^4$$

We can show that $\sin \frac{7}{8} > \cos \frac{7}{8}$ by comparing

$$1 - \frac{1}{2}\left(\frac{7}{8}\right)^2 + \frac{1}{4!}\left(\frac{7}{8}\right)^4 \quad \text{and} \quad \frac{7}{8} - \frac{1}{3!}\left(\frac{7}{8}\right)^3.$$

The manipulation of this kind of comparisons is covered by some of the recent tactics provided in Coq. Let's illustrate this in the following proof.

```
Lemma cmp_sin_approx :
  1 - 1/2 * (7/8)^2 + 1/INR(fact 4) * (7/8)^4
  <= 7/8 - 1/INR(fact 3) * (7/8)^3.
simpl INR; field_simplify.

  ============================
   8073344 / 12582912 <= 18760 / 24576
```

The tactic `simpl` reduces the two calls to `fact` to plain formulas expressed only with repeated additions of `1` (but `fact 4` is a small number). The tactic `field_simplify` then computes the fractions, in no noticeable time. The best tactic to address the resulting goal currently is the `psatzl` [2, 3] tactic. We apply this tactic because this comparison between constants obviously falls in the category of linear comparisons.

```
psatzl R.
No more subgoals
```

```
Qed.
```

Because the theorems we want to establish are meant to be included in the standard library, it is sensible to find a proof that does not rely on the advanced tactic `psatzl`, which may itself rely on the main library. Here is an alternative approach to the proof.

We start by proving the following elementary lemmas, which rely directly on the properties of multiplication and addition with respect to order.

```
Lemma s1 : forall x, 0 < x -> 0 < 2 * x.
Proof.
intros; apply Rmult_lt_0_compat;[apply Rlt_0_2 | assumption].
Qed.

Lemma s2 : forall x, 0 < x -> 0 < 1 + x.
Proof.
intros; apply Rplus_lt_0_compat; [apply Rlt_0_1 | assumption].
Qed.
```

We then transform the comparison between constants so that all computations appear in only one side, then we require the computation of constants using the tactic `field_simplify`.

```
Lemma cmp_sin_approx' :
  1 - 1/2 * (7/8)^2 + 1/INR(fact 4) * (7/8)^4
   <= 7/8 - 1/INR(fact 3) * (7/8)^3.
assert (t : forall x y, 0 <= y - x -> x <= y) by (intros; fourier).
apply t.
  ============================
   0 <=
   7 / 8 - 1 / INR (fact 3) * (7 / 8) ^ 3 -
   (1 - 1 / 2 * (7 / 8) ^ 2 + 1 / INR (fact 4) * (7 / 8) ^ 4)
simpl; field_simplify.
============================
   0 / 1 <= 37644926976 / 3092376453 12
```

At this point we simplify the formula `0 / 1` and prove that both the numerator and the denominator in the right-hand-side fraction are positive.

```
unfold Rdiv; rewrite Rmult_0_l.
apply Rlt_le, Rmult_lt_0_compat, Rinv_0_lt_compat;
 repeat (apply s1 || apply s2); apply Rlt_0_1.
Qed.
```

Note that the number of theorem applications in the `repeat` statement is proportional to the logarithm of the numbers being scrutinized, so this repeat statement is actually very fast to execute.

Using lemma `pre_cos_bound` we can also verify that $\cos \frac{7}{8}$ is positive, and using $\sin \frac{7}{8} > \cos \frac{7}{8}$ together with the lemma `cos_plus` about the cosine of the sum of two angles, we can deduce that $\cos \frac{7}{4}$ is negative.

It is then possible to use the *intermediate value theorem*, already provided in the library, with the following statement:

```
Lemma IVT :
  forall (f:R -> R) (x y:R),
    continuity f ->
    x < y -> f x < 0 -> 0 < f y -> { z:R | x <= z <= y /\ f z = 0 }.
```

In plain english, this lemma says : *if f is continuous, if $x < y$ and $f(x) < 0$ and $0 < f(y)$, then we can obtain a z between x and y such that $f(z) = 0$.* Note that this theorem is not directly applicable to the cos function, because cos has a negative value for $\frac{7}{4}$, which is larger than $\frac{7}{8}$ where it has a positive value, but the opposite function is suitable. We can then conclude that this function has a root between these values.

```
Definition PI_2_aux : {z | 7/8 <= z <= 7/4 /\ -cos z = 0}.
```

In a sense, the value `PI_2_aux` is a *qualified* real value, that is, a plain value together with proofs that this value is between $\frac{7}{8}$ and $\frac{7}{4}$ and that the function $-\cos$ returns 0 for that input. The number π can then be defined as the double of the value described in this definition, and the fact that $\cos\frac{\pi}{2} = 0$ is a consequence of the definition.

```
Definition PI2 := proj1_sig PI_2_aux.

Definition PI := 2 * PI2.

Lemma cos_pi2 : cos (PI/2) = 0.
```

Then, using the lemma `cos_plus`, it is easy to prove the following facts:

$$\begin{aligned}\cos(\frac{\pi}{2} - x) &= \sin x\\ \sin(\frac{\pi}{2}) &= 1\\ \sin(x+y) &= \sin x\cos y + \cos x \sin y\end{aligned}$$

It is also possible to prove that the derivative of sin is cos and the derivative of cos is $-\sin$. Also, `pre_sin_bound` guarantees that sin is positive between 0 and $\frac{7}{4}$, thus cos is decreasing between 0 and $\frac{7}{4}$, so cos is positive between 0 and $\frac{\pi}{2}$ and so is sin. Then we can deduce that sin is positive between 0 and π, and thus cos is decreasing between 0 and π, as stated by the following lemma:

```
Lemma cos_decreasing_0 :
  forall x y:R, 0 <= x -> x <= PI -> 0 <= y -> y <= PI ->
    cos x < cos y -> y < x.
```

This guarantees that there is only one value between $\frac{7}{8}$ and $\frac{7}{4}$ such that $\cos x = 0$, this value is $x = \frac{\pi}{2}$.

2.4 First techniques to compute approximations of π

The polynomial function provided by `cos_approx` can already be used to compute approximants of π. The standard library does that to establish that $\frac{3}{2} < \frac{\pi}{2}$. We shall now see how to use this for closer approximations.

For instance, any value a inside the interval $(\frac{7}{8},\frac{7}{4})$ such that cos_approx a (2 * (n + 1)) < 0 is guaranteed to be an over-approximant. Similarly, if cos_approx a (2 * n + 1) is positive, the value a is guaranteed to be an under-approximant.

When performing the numeric computations, we need to be careful not to compute the factorial function using natural numbers, because such computations have a cost proportional to the end value. If the computations are done in real numbers by well-designed tactics or in integers by the direct definition of multiplication in integers, then the proofs can be performed fairly quickly.

A good illustration is the proof that $\pi/2 < \frac{31416}{20000}$, which can be done in the following way.

```
Lemma over1 : PI/2 < 31416/20000.
assert (t := PI2_3_2).
  t : 3 / 2 < PI / 2
  ============================
   PI / 2 < 31416 / 20000
```

The first tactic adds the fact $\frac{\pi}{2} > \frac{3}{2}$ to the context, to make sure the next calls to psatzl R will succeed for most formulas that entail π. The next theorems we use are cos_decreasing_0 and cos_PI2. The former has a collection of side-conditions to satisfy, but they are mostly taken care of by psatzl R.

```
apply cos_decreasing_0; try psatzl R; rewrite cos_PI2.

  t : 3 / 2 < PI / 2
  ============================
   cos (31416 / 20000) < 0
```

We can then use pre_cos_bound, so that we only have to compare the value of a polynomial in 31416/2000 to 0. We determined by trials and errors that a polynomial of degree 12 was enough to conclude, so we give the parameter $2 = \frac{12}{4} - 1$ to the theorem.

```
destruct (pre_cos_bound (31416/20000) 2) as [_ t’]; try psatzl R.
apply Rle_lt_trans with (1 := t’).
  t : 3 / 2 < PI / 2
  t’ : cos (31416/20000) <= cos_approx (31416 / 20000) (2 * (2 + 1))
  ============================
   cos_approx (31416 / 20000) (2 * (2 + 1)) < 0
```

We need to expand the definition of cos_approx to obtain mostly pure expressions. Notice that we do not expand the computation of the factorial function using the simpl tactic, which would run us into a complexity wall. So we use simpl before expanding cos_term. This only expands the function sum_f_R0.

```
unfold cos_approx; simpl; unfold cos_term.
  t : 3 / 2 < PI / 2
  t’ : cos (31416/20000) <= cos_approx (31416 / 20000) (2 * (2 + 1))
  ============================
   (-1) ^ 0 * ((31416 / 20000) ^ (2 * 0) / INR (fact (2 * 0))) +
   (-1) ^ 1 * ((31416 / 20000) ^ (2 * 1) / INR (fact (2 * 1))) +
   (-1) ^ 2 * ((31416 / 20000) ^ (2 * 2) / INR (fact (2 * 2))) +
   (-1) ^ 3 * ((31416 / 20000) ^ (2 * 3) / INR (fact (2 * 3))) +
   (-1) ^ 4 * ((31416 / 20000) ^ (2 * 4) / INR (fact (2 * 4))) +
   (-1) ^ 5 * ((31416 / 20000) ^ (2 * 5) / INR (fact (2 * 5))) +
   (-1) ^ 6 * ((31416 / 20000) ^ (2 * 6) / INR (fact (2 * 6))) < 0
```

We then want to perform some natural number computations (up to `2 * 6`) but not all (we do not want to compute 12! in natural number representation). So we want to expand natural number multiplications before expanding the factorial function, then expand the factorial function to express its behavior using multiplications that we do not want to expand. This is carefully done as follows:

```
simpl mult; rewrite !fact_simpl; simpl fact.
rewrite !mult_1_r, !mult_INR.
  t : 3 / 2 < PI / 2
  t' : cos (31416/20000) <= cos_approx (31416 / 20000) (2 * (2 + 1))
  ============================
   (-1) ^ 0 * ((31416 / 20000) ^ 0 / INR 1) +
   (-1) ^ 1 * ((31416 / 20000) ^ 2 / INR 2) +
   (-1) ^ 2 * ((31416 / 20000) ^ 4 / (INR 4 * (INR 3 * INR 2))) +
  ...
   (-1) ^ 6 *
   ((31416 / 20000) ^ 12 /
    (INR 12 *
     (INR 11 *
      (INR 10 *
       (INR 9 *
        (INR 8 *
         (INR 7 *
          (INR 6 * (INR 5 * (INR 4 * (INR 3 * INR 2)))))))))))
```

We do not want to print the whole 24 lines of text for the resulting goal. It is enough to say that the biggest argument of `INR` in this expression is 12, so that the complexity of expanding these functions remains manageable. We can conclude the proof with the following tactics.

```
simpl INR; psatzl R.
Qed.
```

The whole proof takes around 3 seconds on a laptop, so it is manageable. A similar proof makes it possible to prove that $\frac{\pi}{2}$ is larger than $\frac{31415}{20000}$.

This staging of computation is necessary because `simpl` relies on the built-in notion computation provided for all recursive functions defined on inductive types, and this notion of computation would normally compute any natural number until it is expressed solely with `O` and repeated applications of `S`. In other theorems where there is no such built-in notion of computation, one can choose to use other "standard forms", on which all computation is represented by rewrite theorems. For instance, in HOL-light, the standard form actually uses a binary representation, and a collection of theorems is provided to show how most basic arithmetic operations act with respect with this binary representation. The parsing and display machinery is also fine-tuned to handle this binary representation, so that the usual computation tactic for arithmetic actually can compute a number like `!12` without difficulty. To the regular user, HOL-light may feel simpler to user, but on the other hand, one needs to know one computation tactic for arithmetic, another for computing with lists, and so on. In Coq, computations capabilities are more powerful, but in the case of computation with the natural numbers, this extra power runs in a complexity wall as soon as the considered numbers are little hight. To alleviate this problem, Coq also provides an alternative type noted `N`, where the structure of the data follows the binary representation. However, using this extra data-type is still unwieldy because one need to navigate between this type and the type `nat`.

3. PROVING THE ALTERNATED SERIES FOR π

In spite of its beautiful simplicity, the Leibnitz formula is not very useful because its convergence speed is rather poor. However, most alternatives to compute approximations of π rely on the power series expansion of the arctangent function and the Leibnitz formula is a particular case. Defining this power series expansion is the main contribution of this section.

3.1 Defining atan and studying its derivative

To define the atan function, we rely on the intermediate value theorem, but we need a more general statement than the one used earlier in section 2.3. Indeed, Lemma `IVT` requires a function that is continuous over the whole real line, but the tan function does not satisfy this condition, since it is discontinuous in every point of the form $\frac{\pi}{2} + k\pi$. For the purpose of defining atan, we proved a stronger version of the intermediate value theorem, with the following statement, which only requires the function to be continuous inside the interval of interest:

```
Lemma IVT_interv : forall (f : R -> R) (x y : R),
       (forall a, x <= a <= y -> continuity_pt f a) ->
       x < y -> f x < 0 -> 0 < f y ->
       {z : R | x <= z <= y /\ f z = 0}.
```

For every value t in the real line, we only need to find two values x and y inside the open interval $(-\frac{\pi}{2}, \frac{\pi}{2})$ such that $\tan x < t$ and $\tan y > t$. Because the tangent function is symmetric, it is enough to find a value y such that $\tan y > |t|$ and we can then choose $x = -y$. We can then distinguish cases between whether t is larger than 1 or not. If t is smaller than 1, it suffices to use the value 1. If t is larger than 1, then we can use the following value:

$$\frac{\pi}{2} - \frac{1}{2 \times (|t| + 1)}.$$

We use the functions `sin_approx` and `cos_approx` to show that the tangent of this value satisfies the needed comparison. We can thus establish the following definitions.

```
Definition frame_tan t : {x | 0 < x < PI/2 /\ Rabs t < tan x}.
Definition pre_atan (y : R) :
   {x : R | -PI/2 < x < PI/2 /\ tan x = y}.
```

The function `frame_tan` returns an x such that $|t| < \tan x$. The function `pre_atan` returns an x such that $\tan x = y$.

These are enough to define the function `atan` and to show that the returned value is between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$.

```
Definition atan x := let (v, _) := pre_atan x in v.

Lemma atan_bound : forall x, -PI/2 < atan x < PI/2.
Proof.
intros x; unfold atan; destruct (pre_atan x) as [v [i _]]; exact i.
Qed.

Lemma atan_right_inv : forall x, tan (atan x) = x.
Proof.
intros x; unfold atan; destruct (pre_atan x) as [v [_ q]]; exact q.
Qed.
```

As a side note, let's remark that it is possible to define the arctangent with only `IVT` by looking for roots of the function $x \mapsto \sin x - t \cos x$, which is continuous in x over the whole line and increasing for $x \in (0, \frac{\pi}{2})$. Nevertheless, `IVT_interv` is a useful addition to the standard library, even if it is not strictly necessary for our purpose.

3.2 Derivatives for tan and atan

To define the power series for the arctangent function, we rely on the property that this function's derivative has a very simple form, which also has a very simple infinite series.

The derivation formula for the tangent function is as follows:

$$\tan' x = 1 + \tan^2 x$$

This formula is valid only for $x \neq \frac{\pi}{2} + k \times \pi$ for any integer k. We proved this formally using the formal definition of tan as the quotient of cos and sin. The division operation over real numbers is defined in such a way that $\frac{x}{0}$ exists for any x, even though there is no way to prove any properties of this number. As a consequence, the tangent's function's type is `R -> R` even though the mathematical object would be ill-defined on countably many points. Correspondingly, the only intervals for which we can establish properties of this function are the intervals for which the cosine, i.e. the denominator, is non-zero.

This is illustrated by the following lines in the standard library:

```
Definition tan (x:R) : R := sin x / cos x.

Lemma tan_plus :
  forall x y:R,
    cos x <> 0 ->
    cos y <> 0 ->
    cos (x + y) <> 0 ->
    1 - tan x * tan y <> 0 ->
    tan (x + y) = (tan x + tan y) / (1 - tan x * tan y).
```

For derivation, we have chosen to describe the derivability of the tan function only on the open interval bounded by $-\frac{\pi}{2}$ and $\frac{\pi}{2}$.

```
Lemma derivable_pt_tan :
    forall x, -PI/2 < x < PI/2 -> derivable_pt tan x.

Lemma derive_pt_tan : forall (x:R),
 forall (Pr1: -PI/2 < x < PI/2),
 derive_pt tan x (derivable_pt_tan x Pr1) = 1 + (tan x)^2.
```

To describe the derivative of a reciprocal function, like the arctangent function, Coq's standard library already contained a general formula. It could only be applied under a collection of important conditions which became quite intricate when restricting the statement to intervals. We added two theorems in the library to handle this case. The first theorem indicates under which conditions the reciprocal function of a derivable function is derivable. the second one indicates what the value of the reciprocal's derivative is. Here is the statement of the first theorem[2].

```
Lemma derivable_pt_recip_interv :
 forall (f g:R->R) (lb ub x : R)
   (lb_lt_ub:lb < ub) (x_encad:f lb < x < f ub)
   (f_eq_g:forall x : R, f lb <= x -> x <= f ub ->
      comp f g x = id x)
   (g_wf:forall x : R, f lb <= x -> x <= f ub -> lb <= g x <= ub)
   (f_incr:forall x y : R, lb <= x -> x < y -> y <= ub -> f x < f y)
   (f_derivable:
     forall a : R, lb <= a <= ub -> derivable_pt f a),
       derive_pt f (g x)
         (derivable_pt_recip_interv_prelim1 f g lb ub x lb_lt_ub
         x_encad f_eq_g g_wf f_incr f_derivable)
         <> 0 ->
         derivable_pt g x.
```

The conditions of this theorem express that we work between two bounds `lb` and `ub`, and with two functions `f` and `g`, where `f` is the right inverse of `g` inside the interval `(f lb, f ub)`, the image of the interval `(f lb, f ub)` by the function `g` is inside the interval `[lb, ub]`, the function `f` is increasing over the interval `[lb, ub]`, and the derivative of `f` is different from 0. Under these conditions `g` is derivable. We note here that Coq's standard library makes a heavy use of depend types to describe the derivative of functions only at places where this derivative is proved to exist.

The second theorem relies on the same conditions. Note that the final equality of this theorem expresses the equality between derivatives, but both the left-hand

[2] In `Coq 8.4pl2`, these two theorems are not loaded by default with the `Reals` library, one must require the `Ranalysis5` sub-library.

side and the right-hand side need to contain proof components to express that the derivatives being considered do exist.

```
Lemma derive_pt_recip_interv :
  forall (f g:R->R) (lb ub x:R)
   (lb_lt_ub:lb < ub) (x_encad:f lb < x < f ub)
   (f_incr:forall x y : R, lb <= x -> x < y -> y <= ub -> f x < f y)
   (g_wf:forall x : R, f lb <= x -> x <= f ub -> lb <= g x <= ub)
   (Prf:forall a : R, lb <= a <= ub -> derivable_pt f a)
   (f_eq_g:forall x, f lb <= x -> x <= f ub -> (comp f g) x = id x)
   (Df_neq:
        derive_pt f (g x)
          (derivable_pt_recip_interv_prelim1 f g lb ub x
                 lb_lt_ub x_encad f_eq_g g_wf f_incr Prf)
         <> 0),
       derive_pt g x
         (derivable_pt_recip_interv f g lb ub x lb_lt_ub x_encad
           f_eq_g g_wf f_incr Prf Df_neq)
       =
       1 / (derive_pt f (g x)
              (Prf (g x)
                (derive_pt_recip_interv_prelim1_1 f g lb ub x
                lb_lt_ub x_encad f_incr g_wf f_eq_g))).
```

With the help of these new theorems we were able to establish a theorem with the following statements:

```
Lemma derivable_pt_atan : forall x, derivable_pt atan x.

Lemma derive_pt_atan : forall x,
      derive_pt atan x (derivable_pt_atan x) =
        1 / (1 + x ^ 2).
```

Note that the sub-formula `x ^ 2` in the second theorem's statement comes from the simplification of `tan (atan x) ^ 2`.

3.3 The power series for atan

The next step is to describe the power series for the function $x \mapsto \frac{1}{1+x^2}$ and for the arctangent function. Part of this work was copied from Guillaume Melquiond's interval package, where the `atan` function was already studied.

It requires a simple proof by induction to show that

$$\sum_{i=0}^{n}(-1)^i x^{2i} = \frac{1-(-x^2)^{n+1}}{1+x^2}.$$

When n goes to infinity, the left-hand side of this equality is an alternated series whose generic term is decreasing in absolute value and has limit 0. There is a generic theorem in Coq's standard library to show that such an infinite sum converges (and the modulus of convergence is given by the last term). In the right-hand side of the equality above, we also see that the limit is easy to compute. Moreover, we show that the function $x \mapsto x^n$ is increasing for every positive natural number n, and this implies that the convergence of the series towards $\frac{1}{1+x^2}$ is uniform in any closed interval inside the interval $(-1, 1)$. We define a function `ps_atan` that coincides with the limit of this power series on the interval [-1,1] and coincides with `atan` outside this interval. In this name the `ps` prefix is an acronym for *power series*.

Then we rely on a general theorem about the derivatives of uniformly converging sequences of functions to prove that the derivative of `ps_atan` inside the open interval $(-1, 1)$ is the function $x \mapsto \frac{1}{1+x^2}$, relying on a general theorem about the derivatives of uniformly converging sequences of functions, which we added to Coq's standard library.

```
Lemma derivable_pt_lim_CVU : forall (fn fn':nat -> R -> R)(f g:R->R)
  (x:R) c r, Boule c r x ->
  (forall y n, Boule c r y -> derivable_pt_lim (fn n) y (fn' n y))->
  (forall y, Boule c r y -> Un_cv (fun n => fn n y) (f y)) ->
  (CVU fn' g c r) ->
  (forall y, Boule c r y -> continuity_pt g y) ->
  derivable_pt_lim f x (g x).
```

Because we wanted to state this theorem at the right level of abstraction, we took care of expressing it with concepts taken from metric spaces and topology (hence the word `Boule`, which is French for *Ball* and was already present in Coq's standard library).

With the help of this theorem, we prove the following lemma about `ps_atan`:

```
Lemma derivable_pt_lim_ps_atan : forall x, -1 < x < 1 ->
  derivable_pt_lim ps_atan x ((fun y => /(1 + y ^ 2)) x).
```

Moreover, we prove that `ps_atan` has value 0 in 0, like `atan`, and since they have the same derivative over the open interval (-1,1), they coincide inside this interval. On the other hand, we also proved that the function `ps_atan` is continuous in 1. All combined, we can use the mean value theorem to express that `ps_atan - atan` must coincide with the 0 function everywhere in the interval [0,1]. Since `atan` is related to the trigonometric functions, it is related to the new definition of π. Moreover, `ps_atan` is related to the power series and thus it is related to the old definition of π. This reconciles the new definitions with the old one and we can prove the property, now named `PI_ineq`. In this definition `/ x` is the notation for the inverse $\frac{1}{x}$.

```
Definition PI_tg (n:nat) := / INR (2 * n + 1).

Lemma PI_ineq :
  forall N : nat,
    sum_f_R0 (tg_alt PI_tg) (S (2 * N)) <= PI / 4 <=
    sum_f_R0 (tg_alt PI_tg) (2 * N).
```

In mathematical notation, after expanding the definitions of `tg_alt` and `PI_tg`, this means

$$\forall N, \sum_{n=0}^{2N+1} \frac{-1^n}{2n+1} \leq \frac{\pi}{4} \leq \sum_{n=0}^{2N} \frac{-1^n}{2n+1}$$

4. ALTERNATIVE APPROACHES TO DESCRIBING π

We now look at three other methods to describe π and we describe the formal proofs that show that these other methods return the same value of π.

4.1 The surface of the circle as an integral

An elementary way to describe π is to compute the surface of a disk by simple means. For instance, the equation of the unit circle is given by:

$$x^2 + y^2 = 1$$

Looking only at the upper half of the circle, this means computing y as a function of x, we simply have $y = \sqrt{1 - x^2}$.

The value of the surface for the half disk can be computed approximately by taking regularly spaced numbers $x_0 = -1$, $x_1 = -1 + 2/k$, …, $x_k = -1 + 2k/k = 1$, and adding the corresponding y values for each of these numbers, and then dividing by k. As k grows, the approximation gets better and better. This actually corresponds to computing the integral of step functions that get closer and closer to the function $\sqrt{1 - x^2}$. This is the principle of Riemann integration.

Riemann integration is already described in Coq's standard library, and its relation to computing derivatives is also covered. In particular, every continuous function is Riemann integrable and we could thus have defined π by the following lines:

```
Definition circle_aux x := Rabs (1 - x ^ 2).

Definition circle_curve x := sqrt (circle_aux x).

Lemma circle_curve_ct : forall x, continuity_pt circle_curve x.
...

Lemma cmpm1_1 : -1 <= 1.
...

Definition hPI_pr :=
  continuity_implies_RiemannInt cmpm1_1
   (fun x _ => circle_curve_ct x).

Definition PI_sf := 2 * RiemannInt hPI_pr.
```

In principle, the value `PI_sf`[3] should be exactly π. However, it has not been defined using trigonometric functions. The exercise we study in this section is to show the equality between the values.

It boils down to studying the following integral:

$$\pi_{sf} = \int_{-1}^{1} \sqrt{1 - x^2} dx$$

Computing this integral relies on the arcsine function, asin.

[3] The suffix `sf` refers to *surface*.

To define asin, we use the same approach as for defining atan. We first prove for every x between -1 and 1 the existence of a value y between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ such that $\sin y = x$. This proof of existence is quite simple: if x is -1 or 1 we return $-\frac{\pi}{2}$ or $\frac{\pi}{2}$ respectively. For other values, we simply use the intermediate value theorem between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. The existence statement is written in Coq in the following manner:

```
Lemma exists_asin : forall x,
  -1 <= x <= 1 -> {t | -PI/2 <= t <= PI/2 /\ sin t = x}.
```

We can then define the asin function by extracting the value from this existential statement for inputs between -1 and 1. To make this function easy to use, it is better to define it with type `R -> R`. In this case, we must also define the value the function takes for inputs smaller than -1 and for inputs larger than 1. We choose to return $-\frac{\pi}{2}$ when the input is smaller than -1 and $\frac{\pi}{2}$ when the input is larger than 1. We also provided a proof that the function defined in this way is continuous.

Since the arcsine function is the reciprocal function of the sine function and since the sine function has positive derivative between -1 and 1, we can prove that the arcsine function is continuous and derivable in that open interval and that its derivative is

$$\mathrm{asin}' x = \frac{1}{\sqrt{1-x^2}}.$$

The usual approach to computing the integral π_{sf} is to note that we can perform a variable change $x = \sin u$, so $dx = cosu\ du$, and $\sqrt{1-x^2} = \cos u$. So the integral becomes:

$$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^2 u\ du = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{1+\cos 2u}{2} du$$

The primitive of $\frac{1+\cos 2u}{2}$ is $\frac{x}{2} + \frac{sin2u}{4}$. Computing with this primitive gives the desired result.

In practice, variable change is simply justified by the derivation rule for composed functions:

$$(f \circ g)' = f' \circ g \times g'.$$

This derivation rule is provided in the standard library of Coq and we simply formalize the computation by relying on the function

```
 (fun x => sin (2 * asin x)/4 + asin x/2)
```

and showing that its derivative is `circle_curve` between -1 and 1.

4.2 The perimeter of the circle as a limit of polygons

Archimedes computed approximations of π around 250 B.C. His technique relies on computing the perimeter of regular polygons, starting with hexagons. There are actually two approaches, the first one uses a sequence of inscribed polygons and computes under-approximations of π. The second uses tangential polygons and computes over-approximations. Each polygon in the sequence is obtained by multiplying the number of sides of the previous polygon by 2. Archimedes himself repeated this process 4 times (thus computing 5 polygons in each sequence), so that he actually computed approximations of π by computing perimeters of regular polygons with 96 sides.

We formalize the computations of both sequences, showed that each of them converges, and showed that one is strictly increasing and the other strictly decreasing. Let's take a closer look at the computation for the first sequence.

We start with an hexagon, given by the points at angles $\frac{\pi}{6}$, $\frac{\pi}{2}$, $\frac{5\pi}{6}$, $\frac{7\pi}{6}$, $\frac{3\pi}{2}$, and $\frac{11\pi}{6}$. This configuration makes it easy to see that the length of each side is $2 \sin \frac{\pi}{6}$. When dividing angles by 2 to obtain a dodecagon, we get a new polygon with sides of length $2 \sin \frac{\pi}{12}$, and so on. At each step we need to compute $\sin \frac{\alpha}{2}$ while we already know the value of $\sin \alpha$ and the comparisons $0 < \alpha < \frac{\pi}{2}$.

The standard library already contains the formula for computing $\sin \alpha$ if we already know $\sin \frac{\alpha}{2}$:

```
Lemma sin_2a : forall x:R, sin (2 * x) = 2 * sin x * cos x.
```

So we start with the following equality:

$$\sin \alpha = 2 \sin \frac{\alpha}{2} \cos \frac{\alpha}{2}.$$

For α between 0 and $\frac{\pi}{2}$, we know that $\cos \frac{\alpha}{2} = \sqrt{1 - \sin^2 \frac{\alpha}{2}}$. So we need to raise this equation to the square to obtain a second degree equation. This equation has two roots, but the one we are looking for can be distinguished thanks to the property $0 < \sin \frac{\alpha}{2} < \sin \alpha$. This results condenses in the following formalized lemma:

```
Lemma sin_half :
  forall x, 0 <= x < PI/2 ->
    sin (x/2) = sqrt (1/2 - sqrt(1 - sin x^2)/2).
```

To reproduce Archimedes' approach to describing π, we abstract away from $\sin \frac{\alpha}{2}$ and $\sin \alpha$ and make it possible to repeat the process an arbitrary number of times:

```
Fixpoint archimedes_l_seq n : R :=
 match n with
   0%nat => 1/2
 | S p => sqrt (1/2 - sqrt(1 - archimedes_l_seq p ^ 2)/2)
 end.
```

Since we start with $\frac{1}{2} = \sin \frac{\pi}{6}$, this function is designed to map any n to $\sin \frac{\pi}{3 \times 2^{n+1}}$, it remains to multiply this value by $3 \times 2^{n+1}$ to obtain approximations of π.

We proved that this value converges towards π, with the following formalized statement:

```
Lemma archimedes_l_pi :
  Un_cv (fun n => 3 * 2 ^ (n + 1) * archimedes_l_seq n) PI.
```

To make this result useful, it is interesting also to show that every value in the sequence provides a lower bound to π. This is done by showing that the difference between consecutive terms of the sequence is positive.

```
Lemma archimedes_l_growing :
  forall n, 3 * 2 ^ (n + 1) * archimedes_l_seq n <
          3 * 2 ^ (S n + 1) * archimedes_l_seq (S n).
```

```
...

Lemma archimedes_l_bound_pi :
  forall n, 3 * 2 ^ (n + 1) * archimedes_l_seq n < PI.
```

For the sequence based on tangential polygons, we also need find the reciprocal to the function computing $\tan 2\alpha$ from $\tan\alpha$. This function is described in the following formalized lemma:

```
Lemma tan_2a :
  forall x:R, cos x <> 0 -> cos (2 * x) <> 0 ->
    1 - tan x * tan x <> 0 ->
    tan (2 * x) = 2 * tan x / (1 - tan x * tan x).
```

The use of this formula is constrained by the necessity of ensuring that the various divisors are non-zero. When all the conditions are met, the equation can again be transformed into a second degree equation, where we need to select the relevant root. Again, selecting the right root is done by choosing the one that is between 0 and 1, this gives us for the tangents of positive angles smaller than $\frac{\pi}{2}$.

The result can then be generalized to all relevant values between $\frac{-\pi}{2}$ and $\frac{\pi}{2}$.

```
Lemma tan_half : forall x : R, x <> 0 -> -PI/2 < x < PI/2 ->
  tan (x / 2) = (sqrt (1 + tan x^2) - 1)/tan x.
```

Again, we abstract away from $\tan\alpha$ and $\tan\frac{\alpha}{2}$ and define a function that iterates the process, starting from the value $\tan\frac{\pi}{6}$.

```
Lemma archimedes_u_seq_step :
  forall n, archimedes_u_seq n =
  match n with
    0%nat => 1/sqrt 3
  | S p => (sqrt (1 + archimedes_u_seq p^2) - 1)/archimedes_u_seq p
  end.
```

Similarly to previous computations, we showed that the computed value converges to π and provides upper bounds.

```
Lemma archimedes_u_pi :
  Un_cv (fun n => 3 * 2 ^ (n + 1) * archimedes_u_seq n) PI.


Lemma archimedes_u_bound_pi :
  forall n, PI < 3 * 2 ^ (n + 1) * archimedes_u_seq n.
```

5. COMPUTING π

All the proofs presented so far manipulate mainly real numbers, which are presented as an abstract data-type in Coq's standard library. To effectively compute approximations, it is interesting to concentrate on a data-type that supports computation and is at the same time dense in the real line. An obvious choice is the data-type of rational numbers, for which some support is already provided in the standard library.

5.1 Exact computations with rational numbers

The standard library provides a module called `QArith` for computing with rational numbers. Rational numbers are constructed as pairs of an integer and a positive number, and both types use a binary representation for numerals. As a result, Coq actually provides means to perform exact computations of rational numbers. This is done entirely symbolically using inductive types, so that the efficiency of computation does not compare to finely tuned libraries for high-speed arbitrary computation libraries, but it still makes it possible to compute around a hundred decimals of the π number in a matter of seconds.

The `QArith` module provides ways to describe rational values, simply writing `(a#b)` for $\frac{a}{b}$. Then it provides the operations of addition, multiplication, subtraction, division, and exponentiation. However, none of the values returned by these operations are reduced fractions. For instance, multiplying $\frac{1}{3}$ with $\frac{3}{2}$ returns $\frac{3}{6}$, which is equal as a rational number with $\frac{1}{2}$, but is not represented in the same manner in memory. To produce reduced fractions, we could use the function `Qred` provided by the `QArith` library.

The correspondance with real numbers is established by a function `Q2R` that maps rational values (of type `Q`) to real values (of type `R`), together with a collection of morphism theorems, which state that `Q2R (0#1) =0`, `Q2R (1#1)=1`, `Q2R (a + b) = Q2R a + Q2R b`, and similar theorems for multiplication, subtraction, division, and exponentiation.

When it comes to "seeing" that we can compute π, the rational values are not very telling because it is rather difficult to recognize that a number is approximately the third of another when both numbers are written with several dozens of digits.

To recover the possibility to see the digits of the π number in its fractional decimal representation, we added a function that computes p digits of a rational number. The method is to multiply the numerator by 10^p and to divide by the denominator.

```
Definition dec (x: Q) p := Zdiv (Qnum x * 10 ^ p) (QDen x).
```

We prove that this function is correct, in the sense that the result is the value of the input rounded by default at precision 10^{-p}.

```
Lemma dec_correct : forall x p, (0 <= p)%Z ->
  (IZR(dec x p))/IZR (10 ^ p) <= Q2R x
    < IZR(dec x p + 1)/IZR (10 ^ p).
```

5.2 Machin-like formulas

To compute π to a high precision, it is unreasonable to use Leibnitz' formula. The n^{th} term in the sum is $\frac{\pm 1}{n}$, so that computing $\frac{\pi}{4}$ with precision 10^{-p} requires 10^p terms in the sum. However, computing atan at other rational values converges much faster, thanks to the exponent in the general term for the power series of atan. It is useful to find formulas that relate values of atan 1 with values of atan at smaller rational inputs.

The general formula for the tangent of a difference of angles is:

$$\tan(x-y) = \frac{\tan x - \tan y}{1 + \tan x \tan y}.$$

Replace x with atan a and y with atan b and apply atan on both sides of the equation, this formula can be rephrased using the arctangent function to obtain the following formulas:

$$\mathrm{atan}\ a - \mathrm{atan}\ b = \mathrm{atan} \frac{a-b}{1+ab}$$

For instance, using $a = 1$, $b = \frac{1}{2}$ we obtain the following formula:

$$\frac{\pi}{4} = \mathrm{atan}\frac{1}{2} + \mathrm{atan}\frac{1}{3}.$$

We gave a formal proof of the atan subtraction formula and used it repeatedly to prove a variety of formulas involving inverses of integers.

```
Definition atan_sub u v := (u - v)/(1 + u * v).

Lemma atan_sub_correct :
 forall u v, 1 + u * v <> 0 -> -PI/2 < atan u - atan v < PI/2 ->
   -PI/2 < atan (atan_sub u v) < PI/2 ->
   atan u = atan v + atan (atan_sub u v).

Lemma Machin_2_3 : PI/4 = atan(/2) + atan(/3).

Lemma Machin_4_5_239 : PI/4 = 4 * atan (/5) - atan(/239).

Lemma Machin_2_3_7 : PI/4 = 2 * atan(/3) + (atan (/7)).
```

In particular, the formula `Machin_4_5_239` is the one used by John Machin in 1706 to compute π to the hundredth decimal place. It is easy to understand why he preferred this formula: it involves computing many powers of $\frac{1}{5}$, but each of these powers is easily computed in decimal notation, since dividing by 5 is the same as multiplying by 2 (which is fairly easy) and dividing by 10 (which is very easy).

We developed a few functions to encode the computation of Machin's formula using rational inputs and outputs.

The first function, `atan_rat_aux`, computes the arctangent value for x by adding n terms of the power series. It returns the sum, the sign of the last term in the sum, the number $2n+1$ as represented in the type `positive`, and the rational number x^{2n+1}. This is designed to avoid recomputing powers of x from scratch at each term.

```
Fixpoint atan_rat_aux x n : Q * Z * positive * Q :=
  match n with
    0%nat => (x, 1%Z, 1%positive, x)
  | S p =>
    let '(v, sign, rank, power) := atan_rat_aux x p in
    let new_sign := Zopp sign in
    let new_rank := (rank + 2)%positive in
    let new_power := x * x * power in
     (v + (Qmake new_sign new_rank) * new_power, new_sign, new_rank,
      new_power)
   end.
```

The second function atan_rat returns the total sum of the first n terms and the absolute value of the last term. This second component is useful to evaluate the quality of the approximation. This function is then used to compute approximations of π and to return an estimate of the error made when truncating the infinite sum to a finite sum.

```
Fixpoint div3 n :=
  match n with S (S (S p)) => S (div3 p) | _ => 0%nat end.

Definition pi_approx rank pr :=
  let (v1, p1) := atan_rat (1#5) rank in
  let (v2, p2) := atan_rat (1#239) (div3 rank) in
  let v := dec ( (4#1) * ((4#1) * v1 - (v2 + p2))) pr in
    (v, dec ((4#1) * ((4#1) * p1 + p2)) pr).
```

If we compute n terms of the series for atan$\frac{1}{5}$, we compute only $\frac{n}{3}$ terms of the series for atan$\frac{1}{239}$. We made this choice because $5^3 < 239 < 5^4$.

We then proved the correctness of our approximation functions, to obtain the following statement:

```
Lemma pi_approx_correct :
  forall r p v e, (0 <= p)%Z -> pi_approx r p = (v, e) ->
  IZR v/IZR (10 ^ p) < PI < IZR (v + e + 2)/IZR (10 ^ p).
```

Note that the error is taken into account in the result, but with +2 added, so that the last digit computed is not guaranteed, but it may only be wrong by one unit. To compute the first hundred digits of the decimal representation of π, we estimated that it suffices to compute 35 terms of the power series for atan$\frac{1}{5}$ and the corresponding number for the power series for atan$\frac{1}{239}$. This computation takes slightly less than a minute on a standard laptop computer.

In practice when performing proofs, it is seldom necessary to obtain an approximation of π that is more precise than a few digits. But even for such a low precision, Leibnitz' formula is not adapted: it requires a thousand terms to obtain the approximation $3.14 < \pi < 3.15$. On the other hand, using Machin's formula, only 2 terms of the series for $\frac{1}{5}$ and one term of the series for $\frac{1}{239}$ are needed at that precision.

6. CONCLUSION

The work described in Section 1 of this paper has been added in Coq's standard library, together with the description of Machin-like formulas (Section 5.2). The work on computing the surface of the unit disk using an integral, the work on Archimedes' technique, and the work on computing rational approximations and decimal representations are available from the authors and from open archives.

CCorn provide another library for real analysis in Coq, with an emphasis on constructive mathematics and efficient computations for a variety of mathematical functions. In that library, π is defined in another way, as a limit using the cos function

$$u_0 = 0 \qquad u_{n+1} = u_n + \cos u_n.$$

CCorn also provides algorithms to compute π relying on Machin-like formulas, with a more advanced formula suited for faster computation at higher precisions. In particular, Krebbers Spitters [10] advertise computing a collection of combined formulas, among which $\sqrt{\pi}$ to 500 digits, in less than 6 seconds. Our implementation does not compete with this, but this was not really the objective of this work.

Most other theorem provers provide a descriptions of π, but very few consider the question of computing approximations of this mathematical constant beyond the first few digits in the theorem prover itself. In Mizar [11] the definition was added in 1998 [15], first by defining sine and cosine from the exponential function in complex numbers, so the definition is closely related to the series expansion. The number π (the circle ratio) is then defined by $\frac{\sin}{\cos}(\frac{\pi}{4}) = 1$ and $\pi \in (0, 4)$. This definition does not attempt to provide better approximations of π. In ACL2 [9], more precisely in [5], the cosine function is defined by its taylor expansion and π is defined as the only value between 0 and 2 where this function takes the value 0. It is then proved that any value between 0 and 2 where the cosine function is negative is an upper bound of π and likewise that any value where this function is positive is a lower bound. In PVS [13], neither the trigonometric functions nor the π number seem to be provided by the default standard library, on the other hand the NASA PVS library [12] provides descriptions of trigonometric functions and the number π is given with lower and upper bounds to a precision of 8 digits. By studying the sequence of lemmas, one can infer that these bounds are obtained by computing the Machin formula. In Isabelle [14], π is defined as twice the root of cos between 0 and 2. Machin's formula is also provided in the library together with the presentation of arctangent as a power series and its instantiation at 1. Work has also been done in Isabelle to provide formally proved computations of real functions, including the exponential and trigonometric function and the π constant [8]. In this work, an extensive language of formulas containing transcendental functions and constants is defined and an interpreter for this langage is described, so that computations can be performed to a prescribed precision. The computations are performed using Taylor expansions. The value of π is explicitely computed using the original Machin formula.

In HOL-light [6] the trigonometric functions are defined once and for all at the same time as complex exponentials. The constant π is then described as the first positive value where sin has value 0. An approximation tool is provided with the help of a function that uses the monotonicity of sin around $\frac{1}{2}$. More precisely, if a, b are two positive real numbers smaller than 4 such that $\sin \frac{a}{6} \leq \frac{1}{2} \leq \sin \frac{b}{6}$, then they provide bound for π. This tool is then used to provide an approximation of π to a precision of $\frac{1}{2^{32}}$ (more than 9 digits of accuracy). In [7], a more systematic approach is also described, which relies on the Machin formula, and automatically produces approximations of π to any precision (more precisely given an input n it produces an approximation up to 2^{-n}).

In an extra piece of work about π, we also verified formally an algorithm using arithmetic-geometric means, which makes it possible to obtain a precision of one million digits in a matter of hours inside a computation-able proof system like Coq. Close colleagues also verified the correctness of an algorithm based on the BBP formula [1], which makes it possible to compute isolated digits (in hexadecimal format). This work will be described in another publication.

References

[1] David Bailey, Peter Borwein, and Simon Plouffe. On the rapid computation of various polylogarithmic constants. *Mathematics of Computation*, 66:903–913, 1997.

[2] Frédéric Besson. Fast reflexive arithmetic tactics the linear case and beyond. In Thorsten Altenkirch and Conor McBride, editors, *TYPES*, volume 4502 of *Lecture Notes in Computer Science*, pages 48–62. Springer, 2006.

[3] Frédéric Besson, Pierre-Emmanuel Cornilleau, and David Pichardie. Modular SMT proofs for fast reflexive checking inside Coq. In Jean-Pierre Jouannaud and Zhong Shao, editors, *CPP*, volume 7086 of *Lecture Notes in Computer Science*, pages 151–166. Springer, 2011.

[4] Coq development team. The Coq proof assistant, 2008. `http://coq.inria.fr`.

[5] Ruben A. Gamboa. *Mechanically Verifying Real-Valued Algorithms in ACL2*. PhD thesis, University of Texas at Austin, 1999.

[6] John Harrison. HOL light: A tutorial introduction. In *Proceedings of the First International Conference on Formal Methods in Computer-Aided Design (FMCAD96), volume 1166 of Lecture Notes in Computer Science*, pages 265–269. Springer-Verlag, 1996.

[7] John Harrison. Formal verification of floating point trigonometric functions. In Warren A. Hunt and Steven D. Johnson, editors, *Formal Methods in Computer-Aided Design: Third International Conference FMCAD 2000*, volume 1954 of *Lecture Notes in Computer Science*, pages 217–233. Springer-Verlag, 2000.

[8] Johannes Hölzl. Proving inequalities over reals with computation in Isabelle/HOL. In Gabriel Dos Reis and Laurent Théry, editors, *Proceedings of the ACM SIGSAM 2009 International Workshop on Programming Languages for Mechanized Mathematics Systems (PLMMS'09)*, pages 38–45, Munich, August 2009.

[9] Matt Kaufmann, Panagiotis Manolios, and J Strother Moore. *Computer-aided reasoning: an approach.* Kluwer Academic Publishing, 2000.

[10] Robbert Krebbers and Bas Spitters. Type classes for efficient exact real arithmetic in Coq. *Logical Methods in Computer Science*, 9(1), 2011.

[11] Roman Matuszewski and Piotr Rudnicki. Mizar: The first 30 years. *Mechanized Mathematics and Its Applications*, 4:3–24, 2005.

[12] NASA Langley Formal Methods Team. NASA PVS library, 2013. `http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/pvslib.html`.

[13] Sam Owre, John M. Rushby, and Natarajan Shankar. PVS: A prototype verification system. In Deepak Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752, Saratoga, NY, jun 1992. Springer-Verlag.

[14] Lawrence C. Paulson. *Isabelle, a generic theorem prover.* Springer, 1994.

[15] Yuguan Yang and Yasunari Shidama. Trigonometric functions and existence of circle ratio. *Formalized Mathematics*, 7(2), 1998.