

A Proof from THE BOOK: The Partial Fraction Expansion of the Cotangent

Manuel Eberl

March 24, 2023

Abstract

In this article, I formalise a proof from THE BOOK [1, Chapter 23]; namely a formula that was called ‘one of the most beautiful formulas involving elementary functions’:

$$\pi \cot(\pi z) = \frac{1}{z} + \sum_{n=1}^{\infty} \left(\frac{1}{z+n} + \frac{1}{z-n} \right)$$

The proof uses Herglotz’s trick to show the real case and analytic continuation for the complex case.

Contents

1	The Partial-Fraction Formula for the Cotangent Function	2
1.1	Auxiliary lemmas	2
1.2	Definition of auxiliary function	4
1.3	Holomorphicity and continuity	6
1.4	Functional equations	8
1.5	The limit at 0	12
1.6	Final result	12

1 The Partial-Fraction Formula for the Cotangent Function

theory *Cotangent-PFD-Formula*
imports *HOL-Complex-Analysis.Complex-Analysis HOL-Real-Asymp.Real-Asymp*

begin

1.1 Auxiliary lemmas

The following variant of the comparison test for showing summability allows us to use a ‘Big-O’ estimate, which works well together with Isabelle’s automation for real asymptotics.

lemma *summable-comparison-test-bigo*:
fixes $f :: nat \Rightarrow real$
assumes $summable (\lambda n. norm (g n)) f \in O(g)$
shows $summable f$
proof –
from $\langle f \in O(g) \rangle$ **obtain** C **where** C : *eventually* $(\lambda x. norm (f x) \leq C * norm (g x))$ *at-top*
by (*auto elim: landau-o.bigE*)
thus *?thesis*
by (*rule summable-comparison-test-ev*) (*insert assms, auto intro: summable-mult*)
qed

lemma *uniformly-on-image*:
 $uniformly-on (f ' A) g = filtercomap (\lambda h. h \circ f) (uniformly-on A (g \circ f))$
unfolding *uniformly-on-def* **by** (*simp add: filtercomap-INF*)

lemma *uniform-limit-image*:
 $uniform-limit (f ' A) g h F \longleftrightarrow uniform-limit A (\lambda x y. g x (f y)) (\lambda x. h (f x)) F$
by (*simp add: uniformly-on-image filterlim-filtercomap-iff o-def*)

lemma *Ints-add-iff1* [*simp*]: $x \in \mathbf{Z} \implies x + y \in \mathbf{Z} \longleftrightarrow y \in \mathbf{Z}$
by (*metis Ints-add Ints-diff add commute add-diff-cancel-right'*)

lemma *Ints-add-iff2* [*simp*]: $y \in \mathbf{Z} \implies x + y \in \mathbf{Z} \longleftrightarrow x \in \mathbf{Z}$
by (*metis Ints-add Ints-diff add-diff-cancel-right'*)

If a set is discrete (i.e. the difference between any two points is bounded from below), it has no limit points:

lemma *discrete-imp-not-islimgt*:
assumes $e: 0 < e$
and $d: \forall x \in S. \forall y \in S. dist\ y\ x < e \longrightarrow y = x$
shows $\neg x\ islimgt\ S$
proof
assume $x\ islimgt\ S$
hence $x\ islimgt\ S - \{x\}$

by (*meson islimpt-punctured*)
moreover from *assms* **have** *closed* ($S - \{x\}$)
 by (*intro discrete-imp-closed*) *auto*
ultimately show *False*
 unfolding *closed-limpt* **by** *blast*
qed

In particular, the integers have no limit point:

lemma *Ints-not-limpt*: $\neg((x :: 'a :: \text{real-normed-algebra-1}) \text{ islimpt } \mathbb{Z})$
 by (*rule discrete-imp-not-islimpt[of 1]*) (*auto elim!*: *Ints-cases simp: dist-of-int*)

The following lemma allows evaluating telescoping sums of the form

$$\sum_{n=0}^{\infty} (f(n) - f(n+k))$$

where $f(n) \rightarrow 0$, i.e. where all terms except for the first k are cancelled by later summands.

lemma *sums-long-telescope*:
fixes $f :: \text{nat} \Rightarrow 'a :: \{\text{topological-group-add, topological-comm-monoid-add, ab-group-add}\}$
assumes $\text{lim}: f \longrightarrow 0$
shows $(\lambda n. f\ n - f\ (n + c)) \text{ sums } (\sum k < c. f\ k)$ (*is - sums ?S*)
proof –
thm *tendsto-diff*
have $(\lambda N. ?S - (\sum n < c. f\ (N + n))) \longrightarrow ?S - 0$
by (*intro tendsto-intros tendsto-null-sum filterlim-compose[OF assms]; real-asymp*)
hence $(\lambda N. ?S - (\sum n < c. f\ (N + n))) \longrightarrow ?S$
by *simp*
moreover have *eventually* $(\lambda N. ?S - (\sum n < c. f\ (N + n)) = (\sum n < N. f\ n - f\ (n + c)))$ *sequentially*
using *eventually-ge-at-top[of c]*
proof *eventually-elim*
case (*elim N*)
have $(\sum n < N. f\ n - f\ (n + c)) = (\sum n < N. f\ n) - (\sum n < N. f\ (n + c))$
by (*simp only: sum-subtractf*)
also have $(\sum n < N. f\ n) = (\sum n \in \{.. < c\} \cup \{c.. < N\}. f\ n)$
using *elim* **by** (*intro sum.cong*) *auto*
also have $\dots = (\sum n < c. f\ n) + (\sum n \in \{c.. < N\}. f\ n)$
by (*subst sum.union-disjoint*) *auto*
also have $(\sum n < N. f\ (n + c)) = (\sum n \in \{c.. < N + c\}. f\ n)$
using *elim* **by** (*intro sum.reindex-bij-witness[of - \lambda n. n - c \lambda n. n + c]*) *auto*
also have $\dots = (\sum n \in \{c.. < N\} \cup \{N.. < N + c\}. f\ n)$
using *elim* **by** (*intro sum.cong*) *auto*
also have $\dots = (\sum n \in \{c.. < N\}. f\ n) + (\sum n \in \{N.. < N + c\}. f\ n)$
by (*subst sum.union-disjoint*) *auto*
also have $(\sum n \in \{N.. < N + c\}. f\ n) = (\sum n < c. f\ (N + n))$
by (*intro sum.reindex-bij-witness[of - \lambda n. n + N \lambda n. n - N]*) *auto*
finally show *?case*

by simp
qed
ultimately show ?thesis
unfolding sums-def by (rule Lim-transform-eventually)
qed

1.2 Definition of auxiliary function

The following function is the infinite sum appearing on the right-hand side of the cotangent formula. It can be written either as

$$\sum_{n=1}^{\infty} \left(\frac{1}{x+n} + \frac{1}{x-n} \right)$$

or as

$$2x \sum_{n=1}^{\infty} \frac{1}{x^2 - n^2} .$$

definition *cot-pfd* :: 'a :: {real-normed-field, banach} \Rightarrow 'a **where**
cot-pfd x = $(\sum n. 2 * x / (x^2 - \text{of-nat } (\text{Suc } n)^2))$

The sum in the definition of *cot-pfd* converges uniformly on compact sets. This implies, in particular, that *cot-pfd* is holomorphic (and thus also continuous).

lemma *uniform-limit-cot-pfd-complex*:

assumes $R \geq 0$

shows *uniform-limit* (cball 0 R :: complex set)

$(\lambda N x. \sum n < N. 2 * x / (x^2 - \text{of-nat } (\text{Suc } n)^2))$ *cot-pfd* sequentially

unfolding *cot-pfd-def*

proof (rule *Weierstrass-m-test-ev*)

have eventually $(\lambda N. \text{of-nat } (N + 1) > R)$ at-top

by *real-asymp*

thus $\forall_F N$ in sequentially. $\forall (x::\text{complex}) \in \text{cball } 0 R. \text{norm } (2 * x / (x^2 - \text{of-nat } (\text{Suc } N)^2)) \leq$

$2 * R / (\text{real } (N + 1)^2 - R^2)$

proof *eventually-elim*

case (elim N)

show ?case

proof *safe*

fix x :: complex **assume** x: $x \in \text{cball } 0 R$

have $(1 + \text{real } N)^2 - R^2 \leq \text{norm } ((1 + \text{of-nat } N :: \text{complex})^2) - \text{norm } (x^2)$

using x by (auto intro: *power-mono simp: norm-power simp flip: of-nat-Suc*)

also have $\dots \leq \text{norm } (x^2 - (1 + \text{of-nat } N :: \text{complex})^2)$

by (*metis norm-minus-commute norm-triangle-ineq2*)

finally show $\text{norm } (2 * x / (x^2 - (\text{of-nat } (\text{Suc } N))^2)) \leq 2 * R / (\text{real } (N + 1)^2 - R^2)$

unfolding *norm-mult norm-divide* **using** $\langle R \geq 0 \rangle$ x *elim*

```

      by (intro mult-mono frac-le) (auto intro: power-strict-mono)
    qed
  qed
next
show summable (λN. 2 * R / (real (N + 1) ^ 2 - R ^ 2))
proof (rule summable-comparison-test-bigo)
  show (λN. 2 * R / (real (N + 1) ^ 2 - R ^ 2)) ∈ O(λN. 1 / real N ^ 2)
  by real-asymp
next
show summable (λn. norm (1 / real n ^ 2))
  using inverse-power-summable[of 2] by (simp add: field-simps)
qed
qed

lemma sums-cot-pfd-complex:
  fixes x :: complex
  shows (λn. 2 * x / (x ^ 2 - of-nat (Suc n) ^ 2)) sums cot-pfd x
  using tendsto-uniform-limitI[OF uniform-limit-cot-pfd-complex[of norm x], of x]
  by (simp add: sums-def)

lemma sums-cot-pfd-complex':
  fixes x :: complex
  assumes x ∉ ℤ
  shows (λn. 1 / (x + of-nat (Suc n)) + 1 / (x - of-nat (Suc n))) sums cot-pfd
  x
proof -
  have (λn. 2 * x / (x ^ 2 - of-nat (Suc n) ^ 2)) sums cot-pfd x
    by (rule sums-cot-pfd-complex)
  also have (λn. 2 * x / (x ^ 2 - of-nat (Suc n) ^ 2)) =
    (λn. 1 / (x + of-nat (Suc n)) + 1 / (x - of-nat (Suc n))) (is ?lhs =
  ?rhs)
  proof
    fix n :: nat
    have neq1: x + of-nat (Suc n) ≠ 0
      using assms by (metis Ints-0 Ints-add-iff2 Ints-of-nat)
    have neq2: x - of-nat (Suc n) ≠ 0
      using assms by force
    have neq3: x ^ 2 - of-nat (Suc n) ^ 2 ≠ 0
      using assms by (metis Ints-of-nat eq-iff-diff-eq-0 minus-in-Ints-iff power2-eq-iff)
    show ?lhs n = ?rhs n using neq1 neq2 neq3
      by (simp add: divide-simps del: of-nat-Suc) (auto simp: power2-eq-square
    algebra-simps)
  qed
  finally show ?thesis .
qed
qed

lemma summable-cot-pfd-complex:
  fixes x :: complex
  shows summable (λn. 2 * x / (x ^ 2 - of-nat (Suc n) ^ 2))

```

using *sums-cot-pfd-complex*[of *x*] **by** (*simp add: sums-iff*)

lemma *summable-cot-pfd-real*:

fixes *x* :: *real*

shows *summable* ($\lambda n. 2 * x / (x \wedge 2 - \text{of-nat } (\text{Suc } n) \wedge 2)$)

proof –

have *summable* ($\lambda n. \text{complex-of-real } (2 * x / (x \wedge 2 - \text{of-nat } (\text{Suc } n) \wedge 2))$)

using *summable-cot-pfd-complex*[of *of-real x*] **by** *simp*

also have *?this* \longleftrightarrow *?thesis*

by (*rule summable-of-real-iff*)

finally show *?thesis* .

qed

lemma *sums-cot-pfd-real*:

fixes *x* :: *real*

shows ($\lambda n. 2 * x / (x \wedge 2 - \text{of-nat } (\text{Suc } n) \wedge 2)$) *sums cot-pfd x*

using *summable-cot-pfd-real*[of *x*] **by** (*simp add: cot-pfd-def sums-iff*)

lemma *cot-pfd-complex-of-real* [*simp*]: *cot-pfd* (*complex-of-real x*) = *of-real* (*cot-pfd x*)

using *sums-of-real*[*OF sums-cot-pfd-real*[of *x*], **where** *?'a* = *complex*]

sums-cot-pfd-complex[of *of-real x*] *sums-unique2* **by** *auto*

lemma *uniform-limit-cot-pfd-real*:

assumes $R \geq 0$

shows *uniform-limit* (*cball 0 R* :: *real set*)

($\lambda N x. \sum n < N. 2 * x / (x \wedge 2 - \text{of-nat } (\text{Suc } n) \wedge 2)$) *cot-pfd sequentially*

proof –

have *uniform-limit* (*cball 0 R*)

($\lambda N x. \text{Re } (\sum n < N. 2 * x / (x \wedge 2 - \text{of-nat } (\text{Suc } n) \wedge 2))$) ($\lambda x. \text{Re } (\text{cot-pfd } x)$) *sequentially*

by (*intro uniform-limit-intros uniform-limit-cot-pfd-complex assms*)

hence *uniform-limit* (*of-real 'cball 0 R*)

($\lambda N x. \text{Re } (\sum n < N. 2 * x / (x \wedge 2 - \text{of-nat } (\text{Suc } n) \wedge 2))$) ($\lambda x. \text{Re } (\text{cot-pfd } x)$) *sequentially*

by (*rule uniform-limit-on-subset*) *auto*

thus *?thesis*

by (*simp add: uniform-limit-image*)

qed

1.3 Holomorphicity and continuity

lemma *holomorphic-on-cot-pfd* [*holomorphic-intros*]:

assumes $A \subseteq -(\mathbb{Z} - \{0\})$

shows *cot-pfd holomorphic-on A*

proof –

have ***: *open* ($-(\mathbb{Z} - \{0\})$) :: *complex set*

by (*intro open-Compl closed-subset-Ints*) *auto*

define *f* :: *nat* \Rightarrow *complex* \Rightarrow *complex*

where $f = (\lambda N x. \sum n < N. 2 * x / (x \wedge 2 - \text{of-nat } (\text{Suc } n) \wedge 2))$
have *cot-pfd holomorphic-on* $-(\mathbf{Z}-\{0\})$
proof (*rule holomorphic-uniform-sequence*[*OF **])
fix $n :: \text{nat}$
have $**$: $x^2 - (\text{of-nat } (\text{Suc } n))^2 \neq 0$ **if** $x \in -(\mathbf{Z}-\{0\})$ **for** $x :: \text{complex}$ **and**
 $n :: \text{nat}$
proof
assume $x^2 - (\text{of-nat } (\text{Suc } n))^2 = 0$
hence $(\text{of-nat } (\text{Suc } n))^2 = x^2$
by *algebra*
hence $x = \text{of-nat } (\text{Suc } n) \vee x = -\text{of-nat } (\text{Suc } n)$
by (*subst (asm) eq-commute, subst (asm) power2-eq-iff*) *auto*
moreover have $(\text{of-nat } (\text{Suc } n) :: \text{complex}) \in \mathbf{Z} (-\text{of-nat } (\text{Suc } n) :: \text{complex})$
 $\in \mathbf{Z}$
by (*intro Ints-minus Ints-of-nat*)
ultimately show *False* **using** *that*
by (*auto simp del: of-nat-Suc*)
qed
show $f \text{ } n$ *holomorphic-on* $-(\mathbf{Z} - \{0\})$
unfolding *f-def* **by** (*intro holomorphic-intros ***)
next
fix $z :: \text{complex}$ **assume** $z: z \in -(\mathbf{Z} - \{0\})$
from $* z$ **obtain** r **where** $r: r > 0$ $\text{cball } z \ r \subseteq -(\mathbf{Z}-\{0\})$
using *open-contains-cball* **by** *blast*
have *uniform-limit* $(\text{cball } z \ r)$ f *cot-pfd sequentially*
using *uniform-limit-cot-pfd-complex*[*of norm z + r*] **unfolding** *f-def*
proof (*rule uniform-limit-on-subset*)
show $\text{cball } z \ r \subseteq \text{cball } 0 \ (\text{norm } z + r)$
unfolding *cball-subset-cball-iff* **by** (*auto simp: dist-norm*)
qed (*use <r > 0>* **in** *auto*)
with r **show** $\exists d > 0. \text{cball } z \ d \subseteq -(\mathbf{Z} - \{0\}) \wedge \text{uniform-limit } (\text{cball } z \ d) \ f$
cot-pfd sequentially
by *blast*
qed
thus *?thesis*
by (*rule holomorphic-on-subset*) *fact*
qed

lemma *continuous-on-cot-pfd-complex* [*continuous-intros*]:
assumes $A \subseteq -(\mathbf{Z}-\{0\})$
shows *continuous-on* A (*cot-pfd :: complex \Rightarrow complex*)
by (*rule holomorphic-on-imp-continuous-on holomorphic-intros assms*)
+

lemma *continuous-on-cot-pfd-real* [*continuous-intros*]:
assumes $A \subseteq -(\mathbf{Z}-\{0\})$
shows *continuous-on* A (*cot-pfd :: real \Rightarrow real*)
proof –
have *continuous-on* A (*Re \circ cot-pfd \circ of-real*)
by (*intro continuous-intros*) (*use assms in auto*)

also have $Re \circ cot\text{-}pfd \circ of\text{-}real = cot\text{-}pfd$
by *auto*
finally show *?thesis* .
qed

1.4 Functional equations

In this section, we will show three few functional equations for the function *cot-pfd*. The first one is trivial; the other two are a bit tedious and not very insightful, so I will not comment on them.

cot-pfd is an odd function:

lemma *cot-pfd-complex-minus* [*simp*]: $cot\text{-}pfd (-x :: complex) = -cot\text{-}pfd x$
proof –
have $(\lambda n. 2 * (-x) / ((-x) ^ 2 - of\text{-}nat (Suc n) ^ 2)) =$
 $(\lambda n. - (2 * x / (x ^ 2 - of\text{-}nat (Suc n) ^ 2)))$
by *simp*
also have ... *sums -cot-pfd x*
by (*intro sums-minus sums-cot-pfd-complex*)
finally show *?thesis*
using *sums-cot-pfd-complex[of -x] sums-unique2* **by** *blast*
qed

lemma *cot-pfd-real-minus* [*simp*]: $cot\text{-}pfd (-x :: real) = -cot\text{-}pfd x$
using *cot-pfd-complex-minus[of of-real x]*
unfolding *of-real-minus [symmetric] cot-pfd-complex-of-real of-real-eq-iff* .

cot-pfd is periodic with period 1:

lemma *cot-pfd-plus-1-complex*:
assumes $x \notin \mathbf{Z}$
shows $cot\text{-}pfd (x + 1 :: complex) = cot\text{-}pfd x - 1 / (x + 1) + 1 / x$
proof –
have *: $x ^ 2 \neq of\text{-}nat n ^ 2$ **if** $x \notin \mathbf{Z}$ **for** $x :: complex$ **and** n
using *that* **by** (*metis Ints-of-nat minus-in-Ints-iff power2-eq-iff*)
have **: $x + of\text{-}nat n \neq 0$ **if** $x \notin \mathbf{Z}$ **for** $x :: complex$ **and** n
using *that* **by** (*metis Ints-0 Ints-add-iff2 Ints-of-nat*)
have [*simp*]: $x \neq 0$
using *assms* **by** *auto*
have [*simp*]: $x + 1 \neq 0$
using *assms* **by** (*metis ** of-nat-1*)
have [*simp*]: $x + 2 \neq 0$
using ***[of x 2] assms* **by** *simp*

have *lim*: $(\lambda n. 1 / (x + of\text{-}nat (Suc n))) \longrightarrow 0$
by (*intro tendsto-divide-0[OF tendsto-const] tendsto-add-filterlim-at-infinity[OF tendsto-const]*
 $filterlim-compose[OF tendsto-of-nat] filterlim-Suc$)
have *sum1*: $(\lambda n. 1 / (x + of\text{-}nat (Suc n)) - 1 / (x + of\text{-}nat (Suc n + 2)))$
sums

$(\sum n < 2. 1 / (x + \text{of-nat } (\text{Suc } n)))$
using *sums-long-telescope*[*OF lim, of 2*] **by** (*simp add: algebra-simps*)

have $(\lambda n. 2 * x / (x^2 - (\text{of-nat } (\text{Suc } n))^2) - 2 * (x + 1) / ((x + 1)^2 - (\text{of-nat } (\text{Suc } (\text{Suc } n))^2)) -$
 $\text{sums } (\text{cot-pfd } x - (\text{cot-pfd } (x + 1) - 2 * (x + 1) / ((x + 1)^2 - (\text{of-nat } (\text{Suc } 0) ^ 2))))$
using *sums-cot-pfd-complex*[*of x + 1*]
by (*intro sums-diff sums-cot-pfd-complex, subst sums-Suc-iff*) *auto*
also have $2 * (x + 1) / ((x + 1)^2 - (\text{of-nat } (\text{Suc } 0) ^ 2)) = 2 * (x + 1) /$
 $(x * (x + 2))$
by (*simp add: algebra-simps power2-eq-square*)
also have $(\lambda n. 2 * x / (x^2 - (\text{of-nat } (\text{Suc } n))^2) -$
 $2 * (x + 1) / ((x + 1)^2 - (\text{of-nat } (\text{Suc } (\text{Suc } n))^2)) =$
 $(\lambda n. 1 / (x + \text{of-nat } (\text{Suc } n)) - 1 / (x + \text{of-nat } (\text{Suc } n + 2)))$
using **[of x] *[of x + 1] **[of x] **[of x + 1] assms*
apply (*intro ext*)
apply (*simp add: divide-simps del: of-nat-add of-nat-Suc*)
apply (*simp add: algebra-simps power2-eq-square*)
done
finally have *sum2*: $(\lambda n. 1 / (x + \text{of-nat } (\text{Suc } n)) - 1 / (x + \text{of-nat } (\text{Suc } n +$
 $2))) \text{sums}$
 $(\text{cot-pfd } x - \text{cot-pfd } (x + 1) + 2 * (x + 1) / (x * (x + 2)))$
by (*simp add: algebra-simps*)

have $\text{cot-pfd } x - \text{cot-pfd } (x + 1) + 2 * (x + 1) / (x * (x + 2)) =$
 $(\sum n < 2. 1 / (x + \text{of-nat } (\text{Suc } n)))$
using *sum1 sum2 sums-unique2* **by** *blast*
hence $\text{cot-pfd } x - \text{cot-pfd } (x + 1) = -2 * (x + 1) / (x * (x + 2)) + 1 / (x +$
 $1) + 1 / (x + 2)$
by (*simp add: eval-nat-numeral divide-simps*) *algebra?*
also have $\dots = 1 / (x + 1) - 1 / x$
by (*simp add: divide-simps*) *algebra?*
finally show *?thesis*
by *algebra*

qed

lemma *cot-pfd-plus-1-real*:
assumes $x \notin \mathbb{Z}$
shows $\text{cot-pfd } (x + 1 :: \text{real}) = \text{cot-pfd } x - 1 / (x + 1) + 1 / x$
proof -
have $\text{cot-pfd } (\text{complex-of-real } (x + 1)) = \text{cot-pfd } (\text{of-real } x) - 1 / (\text{of-real } x +$
 $1) + 1 / \text{of-real } x$
using *cot-pfd-plus-1-complex*[*of x*] *assms* **by** *simp*
also have $\dots = \text{complex-of-real } (\text{cot-pfd } x - 1 / (x + 1) + 1 / x)$
by *simp*
finally show *?thesis*
unfolding *cot-pfd-complex-of-real of-real-eq-iff* .

qed

cot-pfd satisfies the following functional equation:

$$2f(x) = f\left(\frac{x}{2}\right) + f\left(\frac{x+1}{2}\right) + \frac{2}{x+1}$$

lemma *cot-pfd-funeq-complex*:

fixes $x :: \text{complex}$

assumes $x \notin \mathbb{Z}$

shows $2 * \text{cot-pfd } x = \text{cot-pfd } (x / 2) + \text{cot-pfd } ((x + 1) / 2) + 2 / (x + 1)$

proof –

define $f :: \text{complex} \Rightarrow \text{nat} \Rightarrow \text{complex}$ **where** $f = (\lambda x n. 1 / (x + \text{of-nat } (\text{Suc } n)))$

define $g :: \text{complex} \Rightarrow \text{nat} \Rightarrow \text{complex}$ **where** $g = (\lambda x n. 1 / (x - \text{of-nat } (\text{Suc } n)))$

define $h :: \text{complex} \Rightarrow \text{nat} \Rightarrow \text{complex}$ **where** $h = (\lambda x n. 2 * (f x (n + 1) + g x n))$

have *sums*: $(\lambda n. f x n + g x n)$ *sums* *cot-pfd* x **if** $x \notin \mathbb{Z}$ **for** x
unfolding *f-def* *g-def* **by** (*intro sums-cot-pfd-complex'* *that*)

have $x / 2 \notin \mathbb{Z}$

proof

assume $x / 2 \in \mathbb{Z}$

hence $2 * (x / 2) \in \mathbb{Z}$

by (*intro Ints-mult*) *auto*

thus *False* **using** *assms* **by** *simp*

qed

moreover **have** $(x + 1) / 2 \notin \mathbb{Z}$

proof

assume $(x + 1) / 2 \in \mathbb{Z}$

hence $2 * ((x + 1) / 2) - 1 \in \mathbb{Z}$

by (*intro Ints-mult Ints-diff*) *auto*

thus *False* **using** *assms* **by** (*simp add: field-simps*)

qed

ultimately **have** $(\lambda n. (f (x / 2) n + g (x / 2) n) + (f ((x+1) / 2) n + g ((x+1) / 2) n))$ *sums*

$(\text{cot-pfd } (x / 2) + \text{cot-pfd } ((x + 1) / 2))$

by (*intro sums-add sums*)

also **have** $(\lambda n. (f (x / 2) n + g (x / 2) n) + (f ((x+1) / 2) n + g ((x+1) / 2) n)) =$

$(\lambda n. h x (2 * n) + h x (2 * n + 1))$

proof

fix $n :: \text{nat}$

have $(f (x / 2) n + g (x / 2) n) + (f ((x+1) / 2) n + g ((x+1) / 2) n) =$
 $(f (x / 2) n + f ((x+1) / 2) n) + (g (x / 2) n + g ((x+1) / 2) n)$

by *algebra*

also **have** $f (x / 2) n + f ((x+1) / 2) n = 2 * (f x (2 * n + 1) + f x (2 * n + 2))$

by (*simp add: f-def field-simps*)
also have $g (x / 2) n + g ((x+1) / 2) n = 2 * (g x (2 * n) + g x (2 * n + 1))$
 by (*simp add: g-def field-simps*)
also have $2 * (f x (2 * n + 1) + f x (2 * n + 2)) + \dots =$
 $h x (2 * n) + h x (2 * n + 1)$
unfolding h-def by (*simp add: algebra-simps*)
finally show $(f (x / 2) n + g (x / 2) n) + (f ((x+1) / 2) n + g ((x+1) / 2) n) =$
 $h x (2 * n) + h x (2 * n + 1) .$
qed
finally have sum1:
 $(\lambda n. h x (2 * n) + h x (2 * n + 1)) \text{ sums } (cot\text{-pfd } (x / 2) + cot\text{-pfd } ((x + 1) / 2)) .$

have $f x \longrightarrow 0$ **unfolding f-def**
by (*intro tendsto-divide-0[OF tendsto-const]*
tendsto-add-filterlim-at-infinity[OF tendsto-const]
filterlim-compose[OF tendsto-of-nat] filterlim-Suc)
hence $(\lambda n. 2 * (f x n + g x n) + 2 * (f x (Suc n) - f x n)) \text{ sums } (2 * cot\text{-pfd } x + 2 * (0 - f x 0))$
by (*intro sums-add sums sums-mult telescope-sums assms*)
also have $(\lambda n. 2 * (f x n + g x n) + 2 * (f x (Suc n) - f x n)) = h x$
by (*simp add: h-def algebra-simps fun-eq-iff*)
finally have $*$: $h x \text{ sums } (2 * cot\text{-pfd } x - 2 * f x 0)$
by simp

have $(\lambda n. \text{sum } (h x) \{n * 2 .. <n * 2 + 2\}) \text{ sums } (2 * cot\text{-pfd } x - 2 * f x 0)$
using *sums-group[OF *, of 2]* **by simp**
also have $(\lambda n. \text{sum } (h x) \{n * 2 .. <n * 2 + 2\}) = (\lambda n. h x (2 * n) + h x (2 * n + 1))$
by (*simp add: mult-ac*)
finally have sum2: $(\lambda n. h x (2 * n) + h x (2 * n + 1)) \text{ sums } (2 * cot\text{-pfd } x - 2 * f x 0) .$

have $cot\text{-pfd } (x / 2) + cot\text{-pfd } ((x + 1) / 2) = 2 * cot\text{-pfd } x - 2 * f x 0$
using *sum1 sum2 sums-unique2* **by blast**
also have $2 * f x 0 = 2 / (x + 1)$
by (*simp add: f-def*)
finally show *?thesis* **by algebra**
qed

lemma cot-pfd-funeq-real:
fixes $x :: \text{real}$
assumes $x \notin \mathbb{Z}$
shows $2 * cot\text{-pfd } x = cot\text{-pfd } (x / 2) + cot\text{-pfd } ((x + 1) / 2) + 2 / (x + 1)$
proof –
have *complex-of-real* $(2 * cot\text{-pfd } x) = 2 * cot\text{-pfd } (\text{complex-of-real } x)$
by simp

also have $\dots = \text{complex-of-real } (\text{cot-pfd } (x / 2) + \text{cot-pfd } ((x + 1) / 2) + 2 / (x + 1))$
using *assms* **by** (*subst cot-pfd-funeq-complex*) (*auto simp flip: cot-pfd-complex-of-real*)
finally show *?thesis*
by (*simp only: of-real-eq-iff*)
qed

1.5 The limit at 0

lemma *cot-pfd-real-tendsto-0: cot-pfd $-0 \rightarrow (0 :: \text{real})$*
proof –
have *filterlim cot-pfd (nhds 0) (at (0 :: real) within ball 0 1)*
proof (*rule swap-uniform-limit*)
show *uniform-limit (ball 0 1)*
 $(\lambda N x. \sum n < N. 2 * x / (x^2 - (\text{real } (\text{Suc } n))^2))$ *cot-pfd sequentially*
using *uniform-limit-cot-pfd-real[OF zero-le-one]* **by** (*rule uniform-limit-on-subset*)
auto
have $((\lambda x. 2 * x / (x^2 - (\text{real } (\text{Suc } n))^2)) \longrightarrow 0)$ (*at 0 within ball 0 1*) **for**
n
proof (*rule filterlim-mono*)
show $((\lambda x. 2 * x / (x^2 - (\text{real } (\text{Suc } n))^2)) \longrightarrow 0)$ (*at 0*)
by *real-asymp*
qed (*auto simp: at-within-le-at*)
thus $\forall_F N$ *in sequentially.*
 $((\lambda x. \sum n < N. 2 * x / (x^2 - (\text{real } (\text{Suc } n))^2)) \longrightarrow 0)$ (*at 0 within ball*
0 1)
by (*intro always-eventually allI tendsto-null-sum*)
qed *auto*
thus *?thesis*
by (*simp add: at-within-open-NO-MATCH*)
qed

1.6 Final result

To show the final result, we first prove the real case using Herglotz’s trick, following the presentation in ‘Proofs from THE BOOK’. [1, Chapter 23].

lemma *cot-pfd-formula-real:*
assumes $x \notin \mathbf{Z}$
shows $\pi * \cot (\pi * x) = 1 / x + \text{cot-pfd } x$
proof –
have *ev-not-int: eventually $(\lambda x. r x \notin \mathbf{Z})$ (at x)*
if *filterlim r (at (r x)) (at x)* **for** $r :: \text{real} \Rightarrow \text{real}$ **and** $x :: \text{real}$
proof (*rule eventually-compose-filterlim[OF - that]*)
show *eventually $(\lambda x. x \notin \mathbf{Z})$ (at (r x))*
using *Ints-not-limpt[of r x] islimpt-iff-eventually* **by** *blast*
qed

We define the function $h(z)$ as the difference of the left-hand side and right-hand side. The left-hand side and right-hand side have singularities at the

integers, but we will later see that these can be removed as h tends to 0 there.

```

define  $f :: real \Rightarrow real$  where  $f = (\lambda x. pi * cot (pi * x))$ 
define  $g :: real \Rightarrow real$  where  $g = (\lambda x. 1 / x + cot-pfd x)$ 
define  $h$  where  $h = (\lambda x. if x \in \mathbb{Z} then 0 else f x - g x)$ 

```

```

have [simp]:  $h x = 0$  if  $x \in \mathbb{Z}$  for  $x$ 
using that by (simp add: h-def)

```

It is easy to see that the left-hand side and the right-hand side, and as a consequence also our function h , are odd and periodic with period 1.

```

have odd-h:  $h (-x) = -h x$  for  $x$ 
by (simp add: h-def minus-in-Ints-iff f-def g-def)
have per-f:  $f (x + 1) = f x$  for  $x$ 
by (simp add: f-def algebra-simps cot-def)
have per-g:  $g (x + 1) = g x$  if  $x \notin \mathbb{Z}$  for  $x$ 
using that by (simp add: g-def cot-pfd-plus-1-real)
interpret  $h$ : periodic-fun-simple'  $h$ 
by standard (auto simp: h-def per-f per-g)

```

h tends to 0 at 0 (and thus at all the integers).

```

have h-lim:  $h -0 \rightarrow 0$ 
proof (rule Lim-transform-eventually)
have eventually ( $\lambda x. x \notin \mathbb{Z}$ ) (at (0 :: real))
by (rule ev-not-int) real-asymp
thus eventually ( $\lambda x::real. pi * cot (pi * x) - 1 / x - cot-pfd x = h x$ ) (at 0)
by eventually-elim (simp add: h-def f-def g-def)
next
have ( $\lambda x::real. pi * cot (pi * x) - 1 / x$ )  $-0 \rightarrow 0$ 
unfolding cot-def by real-asymp
hence ( $\lambda x::real. pi * cot (pi * x) - 1 / x - cot-pfd x$ )  $-0 \rightarrow 0 - 0$ 
by (intro tendsto-intros cot-pfd-real-tendsto-0)
thus ( $\lambda x. pi * cot (pi * x) - 1 / x - cot-pfd x$ )  $-0 \rightarrow 0$ 
by simp
qed

```

This means that our h is in fact continuous everywhere:

```

have cont-h: continuous-on  $A$   $h$  for  $A$ 
proof -
have isCont  $h x$  for  $x$ 
proof (cases  $x \in \mathbb{Z}$ )
case True
then obtain  $n$  where [simp]:  $x = of-int n$ 
by (auto elim: Ints-cases)
show ?thesis unfolding isCont-def
by (subst at-to-0) (use h-lim in ⟨simp add: filterlim-filtermap h.plus-of-int⟩)
next
case False

```

```

have continuous-on ( $-\mathbf{Z}$ ) ( $\lambda x. f x - g x$ )
  by (auto simp: f-def g-def sin-times-pi-eq-0 mult.commute[of pi] intro!:
continuous-intros)
hence isCont ( $\lambda x. f x - g x$ )  $x$ 
  by (rule continuous-on-interior)
  (use False in  $\langle$ auto simp: interior-open open-Compl[OF closed-Ints] $\rangle$ )
also have eventually ( $\lambda y. y \in -\mathbf{Z}$ ) (nhds  $x$ )
  using False by (intro eventually-nhds-in-open) auto
hence eventually ( $\lambda x. f x - g x = h x$ ) (nhds  $x$ )
  by eventually-elim (auto simp: h-def)
hence isCont ( $\lambda x. f x - g x$ )  $x \longleftrightarrow$  isCont  $h x$ 
  by (rule isCont-cong)
finally show ?thesis .
qed
thus ?thesis
  by (simp add: continuous-at-imp-continuous-on)
qed
note [continuous-intros] = continuous-on-compose2[OF cont-h]

```

Through the functional equations of the sine and cosine function, we can derive the following functional equation for f that holds for all non-integer reals:

```

have eq-f:  $f x = (f (x / 2) + f ((x + 1) / 2)) / 2$  if  $x \notin \mathbf{Z}$  for  $x$ 
proof -
  have  $x / 2 \notin \mathbf{Z}$ 
    using that by (metis Ints-add field-sum-of-halves)
  hence nz1:  $\sin (x/2 * \pi) \neq 0$ 
    by (subst sin-times-pi-eq-0) auto

  have  $(x + 1) / 2 \notin \mathbf{Z}$ 
proof
  assume  $(x + 1) / 2 \in \mathbf{Z}$ 
  hence  $2 * ((x + 1) / 2) - 1 \in \mathbf{Z}$ 
    by (intro Ints-mult Ints-diff) auto
  thus False using that by (simp add: field-simps)
qed
hence nz2:  $\sin ((x+1)/2 * \pi) \neq 0$ 
  by (subst sin-times-pi-eq-0) auto

  have nz3:  $\sin (x * \pi) \neq 0$ 
    using that by (subst sin-times-pi-eq-0) auto

  have eq:  $\sin (\pi * x) = 2 * \sin (\pi * x / 2) * \cos (\pi * x / 2)$ 
     $\cos (\pi * x) = (\cos (\pi * x / 2))^2 - (\sin (\pi * x / 2))^2$ 
    using sin-double[of pi * x / 2] cos-double[of pi * x / 2] by simp-all
show ?thesis using nz1 nz2 nz3
  apply (simp add: f-def cot-def field-simps)
  apply (simp add: add-divide-distrib sin-add cos-add power2-eq-square eq alge-
bra-simps)

```

done
qed

The corresponding functional equation for *cot-pfd* that we have already shown leads to the same functional equation for *g* as we just showed for *f*:

have *eq-g*: $g\ x = (g\ (x / 2) + g\ ((x + 1) / 2)) / 2$ **if** $x \notin \mathbf{Z}$ **for** x
using *cot-pfd-funeq-real[OF that]* **by** (*simp add: g-def*)

This then leads to the same functional equation for *h*, and because *h* is continuous everywhere, we can extend the validity of the equation to the full domain.

have *eq-h*: $h\ x = (h\ (x / 2) + h\ ((x + 1) / 2)) / 2$ **for** x
proof –
have *eventually* $(\lambda x. x \notin \mathbf{Z})$ *(at x)* *eventually* $(\lambda x. x / 2 \notin \mathbf{Z})$ *(at x)*
eventually $(\lambda x. (x + 1) / 2 \notin \mathbf{Z})$ *(at x)*
by (*rule ev-not-int; real-asymp*)
hence *eventually* $(\lambda x. h\ x - (h\ (x / 2) + h\ ((x + 1) / 2)) / 2 = 0)$ *(at x)*
proof *eventually-elim*
case (*elim x*)
thus *?case using eq-f[of x] eq-g[of x]*
by (*simp add: h-def field-simps*)
qed
hence $(\lambda x. h\ x - (h\ (x / 2) + h\ ((x + 1) / 2)) / 2) -x \rightarrow 0$
by (*simp add: tendsto-eventually*)
moreover **have** *continuous-on UNIV* $(\lambda x. h\ x - (h\ (x / 2) + h\ ((x + 1) / 2)) / 2)$
by (*auto intro!: continuous-intros*)
ultimately **have** $h\ x - (h\ (x / 2) + h\ ((x + 1) / 2)) / 2 = 0$
by (*meson LIM-unique UNIV-I continuous-on-def*)
thus *?thesis*
by *simp*
qed

Since *h* is periodic with period 1 and continuous, it must attain a global maximum *h* somewhere in the interval $[0, 1]$. Let's call this maximum *m* and let x_0 be some point in the interval $[0, 1]$ such that $h(x_0) = m$.

define *m* **where** $m = \text{Sup}\ (h\ \{0..1\})$
have $m \in h\ \{0..1\}$
unfolding *m-def*
proof (*rule closed-contains-Sup*)
have *compact* $(h\ \{0..1\})$
by (*intro compact-continuous-image cont-h*) *auto*
thus *bdd-above* $(h\ \{0..1\})$ *closed* $(h\ \{0..1\})$
by (*auto intro: compact-imp-closed compact-imp-bounded bounded-imp-bdd-above*)
qed *auto*
then obtain x_0 **where** $x_0: x_0 \in \{0..1\}$ $h\ x_0 = m$
by *blast*

```

have h-le-m:  $h\ x \leq m$  for  $x$ 
proof -
  have  $h\ x = h\ (\text{frac } x)$ 
    unfolding frac-def by (rule h.minus-of-int [symmetric])
  also have  $\dots \leq m$  unfolding m-def
proof (rule cSup-upper)
  have  $\text{frac } x \in \{0..1\}$ 
    using frac-lt-1[of x] by auto
  thus  $h\ (\text{frac } x) \in h\ \{0..1\}$ 
    by blast
next
  have compact ( $h\ \{0..1\}$ )
    by (intro compact-continuous-image cont-h) auto
  thus bdd-above ( $h\ \{0..1\}$ )
    by (auto intro: compact-imp-bounded bounded-imp-bdd-above)
qed
finally show ?thesis .
qed

```

Through the functional equation for h , we can show that if h attains its maximum at some point x , it also attains it at $\frac{1}{2}x$. By iterating this, it attains the maximum at all points of the form $2^{-n}x_0$.

```

have h-eq-m-iter-aux:  $h\ (x / 2) = m$  if  $h\ x = m$  for  $x$ 
  using eq-h[of x] that h-le-m[of x / 2] h-le-m[of (x + 1) / 2] by simp
have h-eq-m-iter:  $h\ (x_0 / 2^{\wedge} n) = m$  for  $n$ 
proof (induction n)
  case (Suc n)
  have  $h\ (x_0 / 2^{\wedge} \text{Suc } n) = h\ (x_0 / 2^{\wedge} n / 2)$ 
    by (simp add: field-simps)
  also have  $\dots = m$ 
    by (rule h-eq-m-iter-aux) (use Suc.IH in auto)
  finally show ?case .
qed (use x0 in auto)

```

Since the sequence $n \mapsto 2^{-n}x_0$ tends to 0 and h is continuous, we derive $m = 0$.

```

have  $(\lambda n. h\ (x_0 / 2^{\wedge} n)) \longrightarrow h\ 0$ 
  by (rule continuous-on-tendsto-compose[OF cont-h[of UNIV]]) (force | real-asymp)
moreover from h-eq-m-iter have  $(\lambda n. h\ (x_0 / 2^{\wedge} n)) \longrightarrow m$ 
  by simp
ultimately have  $m = h\ 0$ 
  using tendsto-unique by force
hence  $m = 0$ 
  by simp

```

Since h is odd, this means that h is identically zero everywhere, and our result follows.

```

have  $h\ x = 0$ 

```



```

    using h-le-m[of x] h-le-m[of -x] ⟨m = 0⟩ odd-h[of x] by linarith
  thus ?thesis
    using assms by (simp add: h-def f-def g-def)
qed

```

We now lift the result from the domain $\mathbb{R} \setminus \mathbb{Z}$ to $\mathbb{C} \setminus \mathbb{Z}$. We do this by noting that $\mathbb{C} \setminus \mathbb{Z}$ is connected and the point $\frac{1}{2}$ is both in $\mathbb{C} \setminus \mathbb{Z}$ and a limit point of $\mathbb{R} \setminus \mathbb{Z}$.

lemma *one-half-limit-point-Reals-minus-Ints*: $(1 / 2 :: \text{complex}) \text{ islimpt } \mathbb{R} - \mathbb{Z}$

```

proof (rule islimptI)
  fix T :: complex set
  assume 1 / 2 ∈ T open T
  then obtain r where r: r > 0 ball (1 / 2) r ⊆ T
    using open-contains-ball by blast
  define y where y = 1 / 2 + min r (1 / 2) / 2
  have y ∈ {0 < .. < 1}
    using r by (auto simp: y-def)
  hence complex-of-real y ∈ ℝ - ℤ
    by (auto elim!: Ints-cases)
  moreover have complex-of-real y ≠ 1 / 2
proof
    assume complex-of-real y = 1 / 2
    also have 1 / 2 = complex-of-real (1 / 2)
      by simp
    finally have y = 1 / 2
      unfolding of-real-eq-iff .
    with r show False
      by (auto simp: y-def)
  qed
  moreover have complex-of-real y ∈ ball (1 / 2) r
    using ⟨r > 0⟩ by (auto simp: y-def dist-norm)
  with r have complex-of-real y ∈ T
    by blast
  ultimately show ∃ y ∈ ℝ - ℤ. y ∈ T ∧ y ≠ 1 / 2
    by blast
qed

```

theorem *cot-pfd-formula-complex*:

```

  fixes z :: complex
  assumes z ∉ ℤ
  shows pi * cot (pi * z) = 1 / z + cot-pfd z
proof -
  let ?f = λz::complex. pi * cot (pi * z) - 1 / z - cot-pfd z
  have pi * cot (pi * z) - 1 / z - cot-pfd z = 0
proof (rule analytic-continuation[where f = ?f])
  show ?f holomorphic-on -ℤ
    unfolding cot-def by (intro holomorphic-intros) (auto simp: sin-eq-0)
next
  show open (-ℤ :: complex set) connected (-ℤ :: complex set)

```

```

    by (auto intro!: path-connected-imp-connected path-connected-complement-countable
countable-int)
  next
    show  $\mathbb{R} - \mathbb{Z} \subseteq (-\mathbb{Z} :: \text{complex set})$ 
      by auto
  next
    show  $(1 / 2 :: \text{complex}) \text{ islimpt } \mathbb{R} - \mathbb{Z}$ 
      by (rule one-half-limit-point-Reals-minus-Ints)
  next
    show  $1 / (2 :: \text{complex}) \in -\mathbb{Z}$ 
      using fraction-not-in-ints[of 2 1, where ?'a = complex] by auto
  next
    show  $z \in -\mathbb{Z}$ 
      using assms by simp
  next
    show ?f z = 0 if z ∈  $\mathbb{R} - \mathbb{Z}$  for z
  proof -
    have complex-of-real pi * cot (complex-of-real pi * z) - 1 / z - cot-pfd z =
      complex-of-real (pi * cot (pi * Re z) - 1 / Re z - cot-pfd (Re z))
      using that by (auto elim!: Reals-cases simp: cot-of-real)
    also have ... = 0
      by (subst cot-pfd-formula-real) (use that in ⟨auto elim!: Reals-cases⟩)
    finally show ?thesis .
  qed
qed
thus ?thesis
  by algebra
qed
end

```

References

- [1] M. Aigner and G. M. Ziegler. *Proofs from THE BOOK*. Springer, 4th edition, 2009.