

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

Manuel Eberl

March 19, 2025

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	3
1.3	Holomorphicity and continuity	6
1.4	Functional equations	10
1.5	The limit at 0	14
1.6	Final result	14

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

lemma *uniformly-on-image*:

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

lemma *uniform-limit-image*:

uniform-limit ($f \text{ ' } A$) $g \ h \ F \longleftrightarrow \text{uniform-limit } A$ ($\lambda x \ y. g \ x \ (f \ y)$) ($\lambda x. h \ (f \ x)$) 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 $d: \forall x \in S. \forall y \in S. \text{dist } y \ x < e \longrightarrow y = x$

shows $\neg x \text{ islimgt } S$

proof

assume $x \text{ islimgt } S$

hence $x \text{ islimgt } S - \{x\}$

by (*meson islimgt-punctured*)

moreover from *assms* **have** *closed* ($S - \{x\}$)

by (*intro discrete-imp-closed*) *auto*

ultimately show *False*

unfolding *closed-limgt* **by** *blast*

qed

In particular, the integers have no limit point:

lemma *Ints-not-limgt*: $\neg((x :: 'a :: \text{real-normed-algebra-1}) \text{ islimgt } \mathbf{Z})$

by (*rule discrete-imp-not-islimgt[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 $cot\text{-}pfd :: 'a :: \{real\text{-}normed\text{-}field, banach\} \Rightarrow 'a$ **where**
 $cot\text{-}pfd\ x = (\sum n. 2 * x / (x \wedge 2 - of\text{-}nat (Suc\ n) \wedge 2))$

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

lemma $uniform\text{-}limit\text{-}cot\text{-}pfd\text{-}complex$:

assumes $R \geq 0$
shows $uniform\text{-}limit (cball\ 0\ R :: complex\ set)$
 $(\lambda N x. \sum n < N. 2 * x / (x \wedge 2 - of\text{-}nat (Suc\ n) \wedge 2))\ cot\text{-}pfd$ *sequentially*
unfolding $cot\text{-}pfd\text{-}def$
proof (*rule Weierstrass-m-test-ev*)
have *eventually* $(\lambda N. of\text{-}nat (N + 1) > R)$ *at-top*
by *real-asymp*
thus $\forall_F N$ *in sequentially*. $\forall (x :: complex) \in cball\ 0\ R. norm (2 * x / (x \wedge 2 - of\text{-}nat (Suc\ N) \wedge 2)) \leq$
 $2 * R / (real (N + 1) \wedge 2 - R \wedge 2)$
proof *eventually-elim*
case (*elim N*)
show *?case*
proof *safe*
fix $x :: complex$ **assume** $x \in cball\ 0\ R$
have $(1 + real\ N)^2 - R^2 \leq norm ((1 + of\text{-}nat\ N :: complex) \wedge 2) - norm$
 $(x \wedge 2)$
using x **by** (*auto intro: power-mono simp: norm-power simp flip: of-nat-Suc*)
also have $\dots \leq norm (x^2 - (1 + of\text{-}nat\ N :: complex)^2)$
by (*metis norm-minus-commute norm-triangle-ineq2*)
finally show $norm (2 * x / (x^2 - (of\text{-}nat (Suc\ N))^2)) \leq 2 * R / (real (N$
 $+ 1) \wedge 2 - R \wedge 2)$
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* $(\lambda N. 2 * R / (real (N + 1) \wedge 2 - R \wedge 2))$
proof (*rule summable-comparison-test-bigo*)
show $(\lambda N. 2 * R / (real (N + 1) \wedge 2 - R \wedge 2)) \in O(\lambda N. 1 / real\ N \wedge 2)$
by *real-asymp*
next
show *summable* $(\lambda n. norm (1 / real\ n \wedge 2))$
using *inverse-power-summable[of 2]* **by** (*simp add: field-simps*)
qed
qed

lemma $sums\text{-}cot\text{-}pfd\text{-}complex$:

fixes $x :: complex$
shows $(\lambda n. 2 * x / (x \wedge 2 - of\text{-}nat (Suc\ n) \wedge 2))\ sums\ cot\text{-}pfd\ x$
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'-aux*:
fixes $x :: 'a :: \{\text{banach, real-normed-field, field-char-0}\}$
assumes $x \notin \mathbb{Z} - \{0\}$
shows $2 * x / (x^{\wedge} 2 - \text{of-nat } (\text{Suc } n)^{\wedge} 2) =$
 $1 / (x + \text{of-nat } (\text{Suc } n)) + 1 / (x - \text{of-nat } (\text{Suc } n))$
proof –
have *neq1*: $x + \text{of-nat } (\text{Suc } n) \neq 0$
using *assms* **by** (*subst add-eq-0-iff2*) (*auto simp del: of-nat-Suc*)
have *neq2*: $x - \text{of-nat } (\text{Suc } n) \neq 0$
using *assms* **by** (*auto simp del: of-nat-Suc*)
have *neq3*: $x^{\wedge} 2 - \text{of-nat } (\text{Suc } n)^{\wedge} 2 \neq 0$
using *assms* **by** (*auto simp del: of-nat-Suc simp: power2-eq-iff*)
show *?thesis* **using** *neq1 neq2 neq3*
by (*simp add: divide-simps del: of-nat-Suc*) (*auto simp: power2-eq-square alge-*
bra-simps)
qed

lemma *sums-cot-pfd-complex'*:
fixes $x :: \text{complex}$
assumes $x \notin \mathbb{Z} - \{0\}$
shows $(\lambda n. 1 / (x + \text{of-nat } (\text{Suc } n)) + 1 / (x - \text{of-nat } (\text{Suc } n))) \text{ sums cot-pfd}$
 x
using *sums-cot-pfd-complex*[*of x*] *sums-cot-pfd-complex'-aux*[*OF assms*] **by** *simp*

lemma *summable-cot-pfd-complex*:
fixes $x :: \text{complex}$
shows *summable* $(\lambda n. 2 * x / (x^{\wedge} 2 - \text{of-nat } (\text{Suc } n)^{\wedge} 2))$
using *sums-cot-pfd-complex*[*of x*] **by** (*simp add: sums-iff*)

lemma *summable-cot-pfd-real*:
fixes $x :: \text{real}$
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 $x :: \text{real}$
shows $(\lambda n. 2 * x / (x^{\wedge} 2 - \text{of-nat } (\text{Suc } n)^{\wedge} 2)) \text{ sums cot-pfd } x$
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 *has-field-derivative-cot-pfd-complex*:

fixes z :: *complex*

assumes $z \in -(\mathbb{Z} - \{0\})$

shows (*cot-pfd has-field-derivative* ($-\text{Polygamma } 1 (1 + z) - \text{Polygamma } 1 (1 - z)$)) $(\text{at } z)$

proof –

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

where $f = (\lambda N x. \sum n < N. 2 * x / (x \wedge 2 - \text{of-nat } (\text{Suc } n) \wedge 2))$

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

where $f' = (\lambda N x. \sum n < N. -1 / (x + \text{of-nat } (\text{Suc } n)) \wedge 2 - 1 / (x - \text{of-nat } (\text{Suc } n)) \wedge 2)$

have $\exists g'. \forall x \in -(\mathbb{Z} - \{0\}). (\text{cot-pfd has-field-derivative } g' x) (\text{at } x) \wedge (\lambda n. f' n x) \longrightarrow g' x$

proof (*rule has-complex-derivative-uniform-sequence*)

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

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

next

fix n :: *nat* **and** x :: *complex*

assume $x \in -(\mathbb{Z} - \{0\})$

have $nz: x^2 - (\text{of-nat } (\text{Suc } n))^2 \neq 0$ **for** n

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 $(\text{of-nat } (\text{Suc } n) :: \text{complex}) \in \mathbb{Z} (-\text{of-nat } (\text{Suc } n) :: \text{complex})$

$\in \mathbb{Z}$
by (*intro Ints-minus Ints-of-nat*) +
ultimately show *False* **using** *x*
by (*auto simp del: of-nat-Suc*)
qed

have *nz1: x + of-nat (Suc k) \neq 0 for k*
using *x* **by** (*subst add-eq-0-iff2*) (*auto simp del: of-nat-Suc*)
have *nz2: x - of-nat (Suc k) \neq 0 for k*
using *x* **by** (*auto simp del: of-nat-Suc*)

have ($(\lambda x. 2 * x / (x^2 - (of\text{-}nat\ (Suc\ k))^2))$ *has-field-derivative*
 $- 1 / (x + of\text{-}nat\ (Suc\ k))^2 - 1 / (x - of\text{-}nat\ (Suc\ k))^2$) (*at x*) **for** *k ::*
nat

proof -
have ($(\lambda x. inverse\ (x + of\text{-}nat\ (Suc\ k)) + inverse\ (x - of\text{-}nat\ (Suc\ k)))$
has-field-derivative
 $- inverse\ ((x + of\text{-}nat\ (Suc\ k)) \wedge 2) - inverse\ ((x - of\text{-}nat\ (Suc\ k)) \wedge$
 $2)$) (*at x*)
by (*rule derivative-eq-intros refl nz1 nz2*) + (*simp add: power2-eq-square*)
also have *?this \longleftrightarrow ?thesis*
proof (*intro DERIV-cong-ev*)
have *eventually* ($\lambda t. t \in -(\mathbb{Z} - \{0\})$) (*nhds x*) **using** *x*
by (*intro eventually-nhds-in-open open-Compl closed-subset-Ints*) *auto*
thus *eventually* ($\lambda t. inverse\ (t + of\text{-}nat\ (Suc\ k)) + inverse\ (t - of\text{-}nat\ (Suc$
 $k)) =$
 $2 * t / (t^2 - (of\text{-}nat\ (Suc\ k))^2)$) (*nhds x*)
proof *eventually-elim*
case (*elim t*)
thus *?case*
using *sums-cot-pfd-complex'-aux[of t k]* **by** (*auto simp add: field-simps*)
qed
qed (*auto simp: field-simps*)
finally show *?thesis* .
qed

thus (*f n has-field-derivative f' n x*) (*at x*)
unfolding *f-def f'-def* **by** (*intro DERIV-sum*)

next
fix *x :: complex* **assume** *x: x \in $-(\mathbb{Z} - \{0\})$*
have *open $-(\mathbb{Z} - \{0\}) :: complex\ set$*
by (*intro open-Compl closed-subset-Ints*) *auto*
then obtain *r* **where** *r: r > 0 cball x r \subseteq $-(\mathbb{Z} - \{0\})$*
using *x open-contains-cball* **by** *blast*

have *uniform-limit (cball x r) f cot-pfd sequentially*
using *uniform-limit-cot-pfd-complex[of norm x + r]* **unfolding** *f-def*
proof (*rule uniform-limit-on-subset*)
show *cball x r \subseteq cball 0 (cmod x + r)*
by (*subst cball-subset-cball-iff*) *auto*

qed (*use* $\langle r > 0 \rangle$ *in* *auto*)
thus $\exists d > 0. \text{cball } x \ d \subseteq -(\mathbf{Z} - \{0\}) \wedge \text{uniform-limit } (\text{cball } x \ d) \ f \ \text{cot-pfd}$
sequentially
using r **by** (*intro* $\text{exI}[\text{of} - r]$) *auto*
qed
then obtain g' **where** $g': \bigwedge x. x \in -(\mathbf{Z} - \{0\}) \implies (\text{cot-pfd has-field-derivative } g' \ x) \ (\text{at } x)$

$$\bigwedge x. x \in -(\mathbf{Z} - \{0\}) \implies (\lambda n. f' \ n \ x) \longrightarrow g' \ x \ \text{by } \text{blast}$$

have $(\lambda n. f' \ n \ z) \longrightarrow -\text{Polygamma } 1 \ (1 + z) - \text{Polygamma } 1 \ (1 - z)$
proof $-$
have $(\lambda n. -\text{inverse } (((1 + z) + \text{of-nat } n) \wedge \text{Suc } 1) - \text{inverse } (((1 - z) + \text{of-nat } n) \wedge \text{Suc } 1)) \ \text{sums}$
 $(-((-1) \wedge \text{Suc } 1 * \text{Polygamma } 1 \ (1 + z) / \text{fact } 1) - (-1) \wedge \text{Suc } 1 * \text{Polygamma } 1 \ (1 - z) / \text{fact } 1)$
using z **by** (*intro* $\text{sums-diff } \text{sums-minus } \text{Polygamma-LIMSEQ}$) (*auto simp: add-eq-0-iff*)
also have $\dots = -\text{Polygamma } 1 \ (1 + z) - \text{Polygamma } 1 \ (1 - z)$
by *simp*
also have $(\lambda n. -\text{inverse } (((1 + z) + \text{of-nat } n) \wedge \text{Suc } 1) - \text{inverse } (((1 - z) + \text{of-nat } n) \wedge \text{Suc } 1)) =$
 $(\lambda n. -1/(z + \text{of-nat } (\text{Suc } n)) \wedge 2 - 1/(z - \text{of-nat } (\text{Suc } n)) \wedge 2)$
by (*simp add: f'-def field-simps power2-eq-square*)
finally show *?thesis*
unfolding $\text{sums-def } f'\text{-def}$.
qed
with $g'(\mathcal{Q})[\text{OF } z]$ **have** $g' \ z = -\text{Polygamma } 1 \ (1 + z) - \text{Polygamma } 1 \ (1 - z)$
using LIMSEQ-unique **by** *blast*
with $g'(\mathcal{I})[\text{OF } z]$ **show** *?thesis*
by *simp*
qed

lemma *has-field-derivative-cot-pfd-complex'* [*derivative-intros*]:
assumes $(g \ \text{has-field-derivative } g') \ (\text{at } x \ \text{within } A)$ **and** $g \ x \notin \mathbf{Z} - \{0\}$
shows $((\lambda x. \text{cot-pfd } (g \ x :: \text{complex})) \ \text{has-field-derivative } (-\text{Polygamma } 1 \ (1 + g \ x) - \text{Polygamma } 1 \ (1 - g \ x)) * g') \ (\text{at } x \ \text{within } A)$
using $\text{DERIV-chain2}[\text{OF } \text{has-field-derivative-cot-pfd-complex } \text{assms}(1)] \ \text{assms}(2)$
by *auto*

lemma *Polygamma-real-conv-complex*: $x \neq 0 \implies \text{Polygamma } n \ x = \text{Re } (\text{Polygamma } n \ (\text{of-real } x))$
by (*simp add: Polygamma-of-real*)

lemma *has-field-derivative-cot-pfd-real* [*derivative-intros*]:
assumes $(g \ \text{has-field-derivative } g') \ (\text{at } x \ \text{within } A)$ **and** $g \ x \notin \mathbf{Z} - \{0\}$
shows $((\lambda x. \text{cot-pfd } (g \ x :: \text{real})) \ \text{has-field-derivative } (-\text{Polygamma } 1 \ (1 + g \ x) - \text{Polygamma } 1 \ (1 - g \ x)) * g') \ (\text{at } x \ \text{within } A)$

proof –
have *: *complex-of-real* $(g\ x) \notin \mathbf{Z} - \{0\}$
using *assms(2)* **by** *auto*
have **: $(1 + g\ x) \neq 0 \ (1 - g\ x) \neq 0$
using *assms(2)* **by** (*auto simp: add-eq-0-iff*)
have $((\lambda x. \text{Re} ((\text{cot-pfd} \circ (\lambda x. \text{of-real} (g\ x)))\ x)) \text{ has-field-derivative}$
 $(-\text{Polygamma } 1 (1 + g\ x) - \text{Polygamma } 1 (1 - g\ x)) * g')$ (*at x within*
A)
by (*rule derivative-eq-intros has-vector-derivative-real-field*
*field-vector-diff-chain-within assms refl *)* +
*(use ** in <auto simp: Polygamma-real-conv-complex>)*)
thus *?thesis*
by *simp*
qed

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

assumes $A \subseteq -(\mathbf{Z} - \{0\})$
shows *cot-pfd holomorphic-on A*

proof –

have *cot-pfd holomorphic-on* $(-(\mathbf{Z} - \{0\}))$
unfolding *holomorphic-on-def*
using *has-field-derivative-cot-pfd-complex field-differentiable-at-within*
field-differentiable-def **by** *fast*

thus *?thesis*

by (*rule holomorphic-on-subset*) (*use assms in auto*)

qed

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

assumes *f holomorphic-on A* $\bigwedge x. x \in A \implies f\ x \notin \mathbf{Z} - \{0\}$

shows $(\lambda x. \text{cot-pfd} (f\ x)) \text{ holomorphic-on } A$

using *holomorphic-on-compose[OF assms(1) holomorphic-on-cot-pfd] assms(2)*

by (*auto simp: o-def*)

lemma *continuous-on-cot-pfd-complex* [*continuous-intros*]:

assumes *continuous-on A f* $\bigwedge z. z \in A \implies f\ z \notin \mathbf{Z} - \{0\}$

shows *continuous-on A* $(\lambda x. \text{cot-pfd} (f\ x :: \text{complex}))$

by (*rule continuous-on-compose2[OF holomorphic-on-imp-continuous-on[OF*
holomorphic-on-cot-pfd[OF order.refl]] assms(1)]) (*use assms(2) in auto*)

lemma *continuous-on-cot-pfd-real* [*continuous-intros*]:

assumes *continuous-on A f* $\bigwedge z. z \in A \implies f\ z \notin \mathbf{Z} - \{0\}$

shows *continuous-on A* $(\lambda x. \text{cot-pfd} (f\ x :: \text{real}))$

proof –

have *continuous-on A* $(\lambda x. \text{Re} (\text{cot-pfd} (\text{of-real} (f\ x))))$

by (*rule continuous-intros assms*) + (*use assms in auto*)

thus *?thesis*

by *simp*

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]: $\text{cot-pfd } (-x :: \text{complex}) = -\text{cot-pfd } x$

proof –

have $(\lambda n. 2 * (-x) / ((-x) ^ 2 - \text{of-nat } (\text{Suc } n) ^ 2)) =$
 $(\lambda n. - (2 * x / (x ^ 2 - \text{of-nat } (\text{Suc } n) ^ 2)))$

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]: $\text{cot-pfd } (-x :: \text{real}) = -\text{cot-pfd } x$

using *cot-pfd-complex-minus[of of-real x]*

unfolding *of-real-minus [symmetric] cot-pfd-complex-of-real of-real-eq-iff* .

$1 / x + \text{cot-pfd } x$ is periodic with period 1:

lemma *cot-pfd-plus-1-complex*:

assumes $x \notin \mathbf{Z}$

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

proof –

have *: $x ^ 2 \neq \text{of-nat } n ^ 2$ **if** $x \notin \mathbf{Z}$ **for** $x :: \text{complex}$ **and** n

using *that* **by** (*metis Ints-of-nat minus-in-Ints-iff power2-eq-iff*)

have **: $x + \text{of-nat } n \neq 0$ **if** $x \notin \mathbf{Z}$ **for** $x :: \text{complex}$ **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*: $(\lambda n. 1 / (x + \text{of-nat } (\text{Suc } n))) \longrightarrow 0$

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*: $(\lambda n. 1 / (x + \text{of-nat } (\text{Suc } n)) - 1 / (x + \text{of-nat } (\text{Suc } n + 2)))$
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 $(\lambda n. 2 * x / (x^2 - (\text{of-nat } (\text{Suc } n))^2) - 2 * (x + 1) / ((x + 1)^2 - (\text{of-nat } (\text{Suc } (\text{Suc } n)))^2))$

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*: $(\lambda n. 1 / (x + \text{of-nat } (\text{Suc } n)) - 1 / (x + \text{of-nat } (\text{Suc } n + 2))) \text{ sums } (cot-pfd\ x - cot-pfd\ (x + 1) + 2 * (x + 1) / (x * (x + 2)))$
by (*simp add: algebra-simps*)

have $cot-pfd\ x - cot-pfd\ (x + 1) + 2 * (x + 1) / (x * (x + 2)) = (\sum n < 2. 1 / (x + \text{of-nat } (\text{Suc } n)))$
using *sum1 sum2 sums-unique2 by blast*
hence $cot-pfd\ x - cot-pfd\ (x + 1) = -2 * (x + 1) / (x * (x + 2)) + 1 / (x + 1) + 1 / (x + 2)$
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 $cot-pfd\ (x + 1 :: \text{real}) = cot-pfd\ x - 1 / (x + 1) + 1 / x$

proof –

have $cot-pfd\ (\text{complex-of-real } (x + 1)) = cot-pfd\ (\text{of-real } x) - 1 / (\text{of-real } x + 1) + 1 / \text{of-real } x$

using *cot-pfd-plus-1-complex*[*of x*] *assms by simp*

also have $\dots = \text{complex-of-real } (cot-pfd\ x - 1 / (x + 1) + 1 / x)$

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* **using** *that* **by** *(intro sums-cot-pfd-complex')* *auto*

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*:
 $(\lambda n. h\ x\ (2 * n) + h\ x\ (2 * n + 1))\ sums\ (cot-pfd\ (x / 2) + cot-pfd\ ((x + 1) / 2)) .$

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 $(\lambda n. 2 * (f\ x\ n + g\ x\ n) + 2 * (f\ x\ (Suc\ n) - f\ x\ n))\ sums\ (2 * cot-pfd\ x + 2 * (0 - f\ x\ 0))$
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 $*$: $h\ x\ sums\ (2 * cot-pfd\ x - 2 * f\ x\ 0)$
by *simp*

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

have $cot-pfd\ (x / 2) + cot-pfd\ ((x + 1) / 2) = 2 * cot-pfd\ x - 2 * f\ x\ 0$
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 $x :: real$
assumes $x \notin \mathbb{Z}$
shows $2 * cot-pfd\ x = cot-pfd\ (x / 2) + cot-pfd\ ((x + 1) / 2) + 2 / (x + 1)$
proof –
have *complex-of-real* $(2 * cot-pfd\ x) = 2 * cot-pfd\ (complex-of-real\ x)$
by *simp*
also have $\dots = complex-of-real\ (cot-pfd\ (x / 2) + cot-pfd\ ((x + 1) / 2) + 2 / (x + 1))$
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*: $\text{cot-pfd } -0 \rightarrow (0 :: \text{real})$
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 $\text{pi} * \text{cot } (\text{pi} * x) = 1 / x + \text{cot-pfd } x$
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 $f :: \text{real} \Rightarrow \text{real}$ **where** $f = (\lambda x. \text{pi} * \text{cot } (\text{pi} * x))$
define $g :: \text{real} \Rightarrow \text{real}$ **where** $g = (\lambda x. 1 / x + \text{cot-pfd } x)$

define h **where** $h = (\lambda x. \text{if } x \in \mathbb{Z} \text{ then } 0 \text{ else } f\ x - g\ x)$

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* $(0 :: \text{real})$)
by (*rule ev-not-int*) *real-asymp*
thus *eventually* $(\lambda x::\text{real}. \pi * \cot(\pi * x) - 1 / x - \cot\text{-pfd } x = h\ x)$ (*at* 0)
by *eventually-elim* (*simp add: h-def f-def g-def*)
next
have $(\lambda x::\text{real}. \pi * \cot(\pi * x) - 1 / x) - 0 \rightarrow 0$
unfolding *cot-def* **by** *real-asymp*
hence $(\lambda x::\text{real}. \pi * \cot(\pi * x) - 1 / x - \cot\text{-pfd } x) - 0 \rightarrow 0 - 0$
by (*intro tendsto-intros cot-pfd-real-tendsto-0*)
thus $(\lambda x. \pi * \cot(\pi * x) - 1 / x - \cot\text{-pfd } x) - 0 \rightarrow 0$
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** $[simp]: x = \text{of-int } n$
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* $(-\mathbb{Z})$ $(\lambda x. f\ x - g\ x)$
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 ⟨auto simp: interior-open open-Compl[OF closed-Ints]⟩)
    also have eventually (λy. y ∈ -Z) (nhds x)
      using False by (intro eventually-nhds-in-open) auto
    hence eventually (λx. f x - g x = h x) (nhds x)
      by eventually-elim (auto simp: h-def)
    hence isCont (λx. f x - g x) x ⟷ 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 ∉ Z for x
proof -
  have x / 2 ∉ Z
    using that by (metis Ints-add field-sum-of-halves)
  hence nz1: sin (x/2 * pi) ≠ 0
    by (subst sin-times-pi-eq-0) auto

  have (x + 1) / 2 ∉ Z
  proof
    assume (x + 1) / 2 ∈ Z
    hence 2 * ((x + 1) / 2) - 1 ∈ 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) ≠ 0
    by (subst sin-times-pi-eq-0) auto

  have nz3: sin (x * pi) ≠ 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 - pdf 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 ...  $\leq m$  unfolding m-def
proof (rule cSup-upper)
  have  $\text{frac } x \in \{0..1\}$ 
    using frac-lt-1[of x] by auto
  thus  $h (\text{frac } x) \in h \text{ ' } \{0..1\}$ 
    by blast
next
  have compact ( $h \text{ ' } \{0..1\}$ )
    by (intro compact-continuous-image cont-h) auto
  thus bdd-above ( $h \text{ ' } \{0..1\}$ )
    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 ... =  $m$ 
    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-asympt) +
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]  $\langle m = 0 \rangle$  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 $T :: \text{complex set}$

assume $1 / 2 \in T$ open T

then obtain r where $r: r > 0$ ball $(1 / 2) r \subseteq T$

using *open-contains-ball* by *blast*

define y where $y = 1 / 2 + \min r (1 / 2) / 2$

have $y \in \{0 < .. < 1\}$

using r by (auto simp: *y-def*)

hence *complex-of-real* $y \in \mathbb{R} - \mathbb{Z}$

by (auto elim!: *Ints-cases*)

moreover have *complex-of-real* $y \neq 1 / 2$

proof

assume *complex-of-real* $y = 1 / 2$

also have $1 / 2 = \text{complex-of-real } (1 / 2)$

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* $y \in \text{ball } (1 / 2) r$

using $\langle r > 0 \rangle$ by (auto simp: *y-def dist-norm*)

with r have *complex-of-real* $y \in T$

by *blast*

ultimately show $\exists y \in \mathbb{R} - \mathbb{Z}. y \in T \wedge y \neq 1 / 2$

by *blast*

qed

theorem *cot-pfd-formula-complex*:

fixes $z :: \text{complex}$

assumes $z \notin \mathbb{Z}$

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

proof –

let $?f = \lambda z :: \text{complex}. \text{pi} * \text{cot } (\text{pi} * z) - 1 / z - \text{cot-pfd } z$

have $\text{pi} * \text{cot } (\text{pi} * z) - 1 / z - \text{cot-pfd } z = 0$

proof (rule *analytic-continuation*[where $f = ?f$])

show $?f$ *holomorphic-on* $-\mathbb{Z}$

unfolding *cot-def* by (intro *holomorphic-intros*) (auto simp: *sin-eq-0*)

next

show *open* $(-\mathbb{Z} :: \text{complex set})$ *connected* $(-\mathbb{Z} :: \text{complex set})$

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 ∈ ℝ - 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.