Stirling's formula

Manuel Eberl

March 16, 2017

Abstract

This work contains a proof of Stirling's formula both for the factorial $n! \sim \sqrt{2\pi n}(n/e)^n$ on natural numbers and the real Gamma function $\Gamma(x) \sim \sqrt{2\pi/x}(x/e)^x$. The proof is based on work by Graham Jameson [1].

Contents

theory *Stirling-Formula*
imports
  ~~/src/HOL/Analysis
  ../Landau-Symbols/Landau-Symbols
begin

context
begin

First, we define the S_n^* from Jameson's article:

private definition S' :: *nat* ⇒ *real* ⇒ *real* **where**
  $S'\ n\ x = 1/(2*x) + (\sum r{=}1..{<}n.\ 1/(\mathit{of\text{-}nat}\ r{+}x)) + 1\ /(2*(n + x))$

Next, the trapezium (also called T in Jameson's article):

private definition T :: *real* ⇒ *real* **where**
  $T\ x = 1/(2*x) + 1/(2*(x{+}1))$

Now we define The difference $\Delta(x)$:

private definition D :: *real* ⇒ *real* **where**
  $D\ x = T\ x - \mathit{ln}\ (x + 1) + \mathit{ln}\ x$

private lemma *S′-telescope-trapezium*:
  assumes $n > 0$
  shows $S'\ n\ x = (\sum r{<}n.\ T\ (\mathit{of\text{-}nat}\ r + x))$
proof (*cases n*)
  case (*Suc m*)
  hence *m*: *Suc m = n* **by** *simp*

have ($\sum r<n.\ T$ (*of-nat* $r + x$)) =
($\sum r<Suc\ m.\ 1\ /\ (2 * real\ r + 2 * x)) + (\sum r<n.\ 1\ /\ (2 * real\ (Suc\ r)$ $+ 2 * x$))
unfolding *m* **by** (*simp add*: *T-def sum.distrib algebra-simps*)
also have ($\sum r<Suc\ m.\ 1\ /\ (2 * real\ r + 2 * x)$) =
$1/(2*x) + (\sum r<m.\ 1\ /\ (2 * real\ (Suc\ r) + 2 * x))$ (**is** - = *?a* + *?S*)
by (*subst sum-lessThan-Suc-shift*) *simp*
also have ($\sum r<n.\ 1\ /\ (2 * real\ (Suc\ r) + 2 * x)$) =
$?S + 1\ /\ (2*(real\ m + x + 1))$ (**is** - = - + *?b*) **by** (*simp add*: *Suc*)
also have $?a + ?S + (?S + ?b) = 2*?S + ?a + ?b$ **by** (*simp add*: *add-ac*)
also have $2 * ?S = (\sum r=0..<m.\ 1\ /\ (real\ (Suc\ r) + x))$
unfolding *sum-distrib-left* **by** (*intro sum.cong*) (*auto simp add*: *divide-simps*)
also have $(\sum r=0..<m.\ 1\ /\ (real\ (Suc\ r) + x)) = (\sum r=Suc\ 0..<Suc\ m.\ 1\ /$ $(real\ r + x))$
by (*subst sum.atLeast-Suc-lessThan-Suc-shift*) *simp-all*
also have $\ldots = (\sum r=1..<n.\ 1\ /\ (real\ r + x))$ **unfolding** *m* **by** *simp*
also have $\ldots + ?a + ?b = S'\ n\ x$ **by** (*simp add*: *S'-def Suc*)
finally show *?thesis* **..**
qed (*insert assms*, *simp-all*)

private lemma *stirling-trapezium*:
assumes *x*: $(x::real) > 0$
shows $D\ x \in \{0\ ..\ 1/(12*x\hat{}2) - 1/(12 * (x+1)\hat{}2)\}$
proof −
define *y* **where** $y = 1\ /\ (2*x + 1)$
from *x* **have** *y*: $y > 0$ $y < 1$ **by** (*simp-all add*: *divide-simps y-def*)

from *x* **have** $D\ x = T\ x - ln\ ((x + 1)\ /\ x)$ **by** (*subst ln-div*) (*simp-all add*: *D-def*)
also from *x* **have** $(x + 1)\ /\ x = 1 + 1\ /\ x$ **by** (*simp add*: *field-simps*)
finally have *D*: $D\ x = T\ x - ln\ (1 + 1/x)$ **.**

from *y* **have** ($\lambda n.\ y * y\hat{}n$) *sums* ($y * (1\ /\ (1 - y))$)
by (*intro geometric-sums sums-mult*) *simp-all*
hence ($\lambda n.\ y\ \hat{}\ Suc\ n$) *sums* ($y\ /\ (1 - y)$) **by** *simp*
also from *x* **have** $y\ /\ (1 - y) = 1\ /\ (2*x)$ **by** (*simp add*: *y-def divide-simps*)
finally have $*$: ($\lambda n.\ y\ \hat{}\ Suc\ n$) *sums* ($1\ /\ (2*x)$) **.**

from *y* **have** ($\lambda n.\ (-y) * (-y)\hat{}n$) *sums* ($(-y) * (1\ /\ (1 - (-y)))$)
by (*intro geometric-sums sums-mult*) *simp-all*
hence ($\lambda n.\ (-y)\ \hat{}\ Suc\ n$) *sums* ($-(y\ /\ (1 + y))$) **by** *simp*
also from *x* **have** $y\ /\ (1 + y) = 1\ /\ (2*(x+1))$ **by** (*simp add*: *y-def divide-simps*)
finally have $**$: ($\lambda n.\ (-y)\ \hat{}\ Suc\ n$) *sums* ($-(1\ /\ (2*(x+1)))$) **.**

from *sums-diff*[*OF* $*$ $**$] **have** *sum1*: ($\lambda n.\ y\ \hat{}\ Suc\ n - (-y)\ \hat{}\ Suc\ n$) *sums* $T\ x$
by (*simp add*: *T-def*)

from *y* **have** $abs\ y < 1$ $abs\ (-y) < 1$ **by** *simp-all*
from *sums-diff*[*OF this*[*THEN ln-series'*]]

have $(\lambda n.\ y\ \hat{}\ n\ /\ \mathit{real}\ n - (-y)\ \hat{}\ n\ /\ \mathit{real}\ n)$ *sums* $(\mathit{ln}\ (1 + y) - \mathit{ln}\ (1 - y))$ **by** *simp*
also from y **have** $\mathit{ln}\ (1 + y) - \mathit{ln}\ (1 - y) = \mathit{ln}\ ((1 + y)\ /\ (1 - y))$ **by** (*simp add*: *ln-div*)
also from x **have** $(1 + y)\ /\ (1 - y) = 1 + 1/x$ **by** (*simp add*: *divide-simps y-def*)
finally have $(\lambda n.\ y\hat{}n\ /\ \mathit{real}\ n - (-y)\hat{}n\ /\ \mathit{real}\ n)$ *sums* $\mathit{ln}\ (1 + 1/x)$ **.**
hence *sum2*: $(\lambda n.\ y\hat{}\mathit{Suc}\ n\ /\ \mathit{real}\ (\mathit{Suc}\ n) - (-y)\hat{}\mathit{Suc}\ n\ /\ \mathit{real}\ (\mathit{Suc}\ n))$ *sums* $\mathit{ln}\ (1 + 1/x)$
by (*subst sums-Suc-iff*) *simp*

from *sum2 sum1* **have** $\mathit{ln}\ (1 + 1/x) \le T\ x$
proof (*rule sums-le* [*OF allI*, *rotated*])
fix $n :: \mathit{nat}$
show $y\ \hat{}\ \mathit{Suc}\ n\ /\ \mathit{real}\ (\mathit{Suc}\ n) - (-y)\ \hat{}\ \mathit{Suc}\ n\ /\ \mathit{real}\ (\mathit{Suc}\ n) \le y\hat{}\mathit{Suc}\ n - (-y)\ \hat{}\ \mathit{Suc}\ n$
proof (*cases even n*)
case *True*
hence *eq*: $A - (-y)\ \hat{}\ \mathit{Suc}\ n\ /\ B = A + y\ \hat{}\ \mathit{Suc}\ n\ /\ B\ \ A - (-y)\ \hat{}\ \mathit{Suc}\ n = A + y\ \hat{}\ \mathit{Suc}\ n$
for $A\ B$ **by** *simp-all*
from y **show** *?thesis* **unfolding** *eq*
by (*intro add-mono*) (*auto simp*: *divide-simps*)
qed *simp-all*
qed
hence $D\ x \ge 0$ **by** (*simp add*: D)

define c **where** $c = (\lambda n.\ \mathit{if\ even}\ n\ \mathit{then}\ 2 * (1 - 1\ /\ \mathit{real}\ (\mathit{Suc}\ n))\ \mathit{else}\ 0)$
note *sums-diff* [*OF sum1 sum2*]
also have $(\lambda n.\ y\ \hat{}\ \mathit{Suc}\ n - (-y)\ \hat{}\ \mathit{Suc}\ n - (y\ \hat{}\ \mathit{Suc}\ n\ /\ \mathit{real}\ (\mathit{Suc}\ n) -$
$(-y)\ \hat{}\ \mathit{Suc}\ n\ /\ \mathit{real}\ (\mathit{Suc}\ n))) = (\lambda n.\ c\ n * y\ \hat{}\ \mathit{Suc}\ n)$
by (*intro ext*) (*simp add*: *c-def algebra-simps*)
finally have *sum3*: $(\lambda n.\ c\ n * y\ \hat{}\ \mathit{Suc}\ n)$ *sums* $D\ x$ **by** (*simp add*: D)

from y **have** $(\lambda n.\ y\hat{}2 * (\mathit{of\text{-}nat}\ (\mathit{Suc}\ n) * y\hat{}n))$ *sums* $(y\hat{}2 * (1\ /\ (1 - y)\hat{}2))$
by (*intro sums-mult geometric-deriv-sums*) *simp-all*
hence $(\lambda n.\ \mathit{of\text{-}nat}\ (\mathit{Suc}\ n) * y\hat{}(n+2))$ *sums* $(y\hat{}2\ /\ (1 - y)\hat{}2)$
by (*simp add*: *algebra-simps power2-eq-square*)
also from x **have** $y\hat{}2\ /\ (1 - y)\hat{}2 = 1\ /\ (4{*}x\hat{}2)$ **by** (*simp add*: *y-def divide-simps*)
finally have $*$: $(\lambda n.\ \mathit{real}\ (\mathit{Suc}\ n) * y\ \hat{}\ (\mathit{Suc}\ (\mathit{Suc}\ n)))$ *sums* $(1\ /\ (4 * x^2))$ **by** *simp*

from y **have** $(\lambda n.\ y\hat{}2 * (\mathit{of\text{-}nat}\ (\mathit{Suc}\ n) * (-y)\hat{}n))$ *sums* $(y\hat{}2 * (1\ /\ (1 - -y)\hat{}2))$
by (*intro sums-mult geometric-deriv-sums*) *simp-all*
hence $(\lambda n.\ \mathit{of\text{-}nat}\ (\mathit{Suc}\ n) * (-y)\hat{}(n+2))$ *sums* $(y\hat{}2\ /\ (1 + y)\hat{}2)$
by (*simp add*: *algebra-simps power2-eq-square*)
also from x **have** $y\hat{}2\ /\ (1 + y)\hat{}2 = 1\ /\ (2\hat{}2{*}(x+1)\hat{}2)$

unfolding *power-mult-distrib* [*symmetric*] **by** (*simp add*: *y-def divide-simps add-ac*)
finally have ∗∗: (λn. *real* (*Suc n*) ∗ (− *y*) ˆ (*Suc* (*Suc n*))) *sums* (*1* / (*4* ∗ (*x* + *1*)^2)) **by** *simp*

define *d* **where** *d* = (λn. *if even n then 2* ∗ *real n else 0*)
note *sums-diff* [*OF* ∗ ∗∗]
also have (λn. *real* (*Suc n*) ∗ *y*ˆ(*Suc* (*Suc n*)) − *real* (*Suc n*) ∗ (−*y*)ˆ(*Suc* (*Suc n*))) =
(λn. *d* (*Suc n*) ∗ *y* ˆ *Suc* (*Suc n*))
by (*intro ext*) (*simp-all add*: *d-def*)
finally have (λn. *d n* ∗ *y* ˆ *Suc n*) *sums* (*1* / (*4* ∗ *x*^{2}) − *1* / (*4* ∗ (*x* + *1*)^2))
by (*subst* (*asm*) *sums-Suc-iff*) (*simp add*: *d-def*)
from *sums-mult*[*OF this*, *of 1/3*] *x*
have *sum4*: (λn. *d n* / *3* ∗ *y* ˆ *Suc n*) *sums* (*1* / (*12* ∗ *x*ˆ*2*) − *1* / (*12* ∗ (*x* + *1*)ˆ*2*))
by (*simp add*: *field-simps*)

have *D x* ≤ (*1* / (*12* ∗ *x*ˆ*2*) − *1* / (*12* ∗ (*x* + *1*)ˆ*2*))
proof (*intro sums-le* [*OF - sum3 sum4*] *allI*)
fix *n* :: *nat*
define *c′* :: *nat* ⇒ *real*
where *c′* = (λn. *if odd n* ∨ *n* = *0 then 0 else if n* = *2 then 4/3 else 2*)
show *c n* ∗ *y* ˆ *Suc n* ≤ *d n* / *3* ∗ *y* ˆ *Suc n*
proof (*intro mult-right-mono*)
have *c n* ≤ *c′ n* **by** (*simp add*: *c-def c′-def*)
also consider *n* = *0* | *n* = *1* | *n* = *2* | *n* ≥ *3* **by** *force*
hence *c′ n* ≤ *d n* / *3* **by** *cases* (*simp-all add*: *c′-def d-def*)
finally show *c n* ≤ *d n* / *3* **.**
qed (*insert y*, *simp*)
qed

with ‹*D x* ≥ *0*› **show** *?thesis* **by** *simp*
qed

The following functions correspond to the $p_n(x)$ from the article. The special case $n = 0$ would not, strictly speaking, be necessary, but it allows some theorems to work even without the precondition $n \neq 0$:

private definition *p* :: *nat* ⇒ *real* ⇒ *real* **where**
p n x = (*if n* = *0 then 1/x else* ($\sum$ *r*<*n*. *D* (*real r* + *x*)))

We can write the Digamma function in terms of S':

private lemma *S′-LIMSEQ-Digamma*:
assumes *x*: *x* ≠ *0*
shows (λn. *ln* (*real n*) − *S′ n x* − *1*/(*2*∗*x*)) ⟶ *Digamma x*
proof −
define *c* **where** *c* = (λn. *ln* (*real n*) − ($\sum$ *r*<*n*. *inverse* (*x* + *real r*)))
have *eventually* (λn. *1* / (*2* ∗ (*x* + *real n*)) = *c n* − (*ln* (*real n*) − *S′ n x* − *1*/(*2*∗*x*))) *at-top*

using *eventually-gt-at-top*[*of 0::nat*]
proof *eventually-elim*
fix *n* :: *nat*
assume *n*: *n* > *0*
have *c n* − (*ln* (*real n*) − *S' n x* − *1*/(*2*∗*x*)) =
−($\sum$ *r*<*n*. *inverse* (*real r* + *x*)) + (*1*/*x* + ($\sum$ *r*=*1*..<*n*. *inverse* (*real r* + *x*))) + *1*/(*2*∗(*real n* + *x*))
using *x* **by** (*simp add*: *S'-def c-def field-simps*)
also have *1*/*x* + ($\sum$ *r*=*1*..<*n*. *inverse* (*real r* + *x*)) = ($\sum$ *r*<*n*. *inverse* (*real r* + *x*))
unfolding *lessThan-atLeast0* **using** *n*
by (*subst* (*2*) *sum-head-upt-Suc*) (*simp-all add*: *field-simps*)
finally show *1* / (*2* ∗ (*x* + *real n*)) = *c n* − (*ln* (*real n*) − *S' n x* − *1*/(*2*∗*x*)) **by** *simp*
qed
moreover have (λ*n*. *1* / (*2* ∗ (*x* + *real n*))) ⟶ *0*
by (*rule real-tendsto-divide-at-top tendsto-const filterlim-tendsto-pos-mult-at-top filterlim-tendsto-add-at-top filterlim-real-sequentially* | *simp*)+
ultimately have (λ*n*. *c n* − (*ln* (*real n*) − *S' n x* − *1*/(*2*∗*x*))) ⟶ *0*
by (*rule Lim-transform-eventually*)
from *tendsto-minus*[*OF this*] **have** (λ*n*. (*ln* (*real n*) − *S' n x* − *1*/(*2*∗*x*)) − *c n*) ⟶ *0* **by** *simp*
moreover from *Digamma-LIMSEQ*[*OF x*] **have** *c* ⟶ *Digamma x* **by** (*simp add*: *c-def*)
ultimately show (λ*n*. *ln* (*real n*) − *S' n x* − *1*/(*2*∗*x*)) ⟶ *Digamma x*
by (*rule Lim-transform* [*rotated*])
qed

Moreover, we can give an expansion of S' with the p as variation terms.

private lemma *S'-approx*:
S' n x = *ln* (*real n* + *x*) − *ln x* + *p n x*
proof (*cases n* = *0*)
case *True*
thus *?thesis* **by** (*simp add*: *p-def S'-def*)
next
case *False*
hence *S' n x* = ($\sum$ *r*<*n*. *T* (*real r* + *x*))
by (*subst S'-telescope-trapezium*) *simp-all*
also have ... = ($\sum$ *r*<*n*. *ln* (*real r* + *x* + *1*) − *ln* (*real r* + *x*) + *D* (*real r* + *x*))
by (*simp add*: *D-def*)
also have ... = ($\sum$ *r*<*n*. *ln* (*real* (*Suc r*) + *x*) − *ln* (*real r* + *x*)) + *p n x*
using *False* **by** (*simp add*: *sum.distrib add-ac p-def*)
also have ($\sum$ *r*<*n*. *ln* (*real* (*Suc r*) + *x*) − *ln* (*real r* + *x*)) = *ln* (*real n* + *x*) − *ln x*
by (*subst sum-lessThan-telescope*) *simp-all*
finally show *?thesis* **.**
qed

We define the limit of the p (simply called $p(x)$ in Jameson's article):

private definition P :: *real* $\Rightarrow$ *real* **where**
$P\ x = (\sum n.\ D\ (real\ n + x))$

private lemma *D-summable*:
assumes x: $x > 0$
shows *summable* $(\lambda n.\ D\ (real\ n + x))$
proof −
have $*$: *summable* $(\lambda n.\ 1\ /\ (12 * (x + real\ n)^2) - 1\ /\ (12 * (x + real\ (Suc\ n))^2))$
by (*rule telescope-summable′ real-tendsto-divide-at-top tendsto-const filterlim-tendsto-pos-mult-at-top filterlim-pow-at-top filterlim-tendsto-add-at-top filterlim-real-sequentially* | *simp*)+
show *summable* $(\lambda n.\ D\ (real\ n + x))$
proof (*rule summable-comparison-test*[*OF* - $*$], *rule exI*[*of* - *2*], *safe*)
fix n :: *nat* **assume** $n \geq 2$
show *norm* $(D\ (real\ n + x)) \leq 1\ /\ (12 * (x + real\ n)\hat{}2) - 1\ /\ (12 * (x + real\ (Suc\ n))\hat{}2)$
using *stirling-trapezium*[*of real* $n + x$] x **by** (*auto simp*: *algebra-simps*)
qed
qed

private lemma *p-LIMSEQ*:
assumes x: $x > 0$
shows $(\lambda n.\ p\ n\ x) \longrightarrow P\ x$
proof −
from *D-summable*[*OF* x] **have** $(\lambda n.\ D\ (real\ n + x))$ *sums* $P\ x$ **unfolding** *P-def*
by (*simp add*: *sums-iff*)
hence $(\lambda n.\ \sum r{<}n.\ D\ (real\ r + x)) \longrightarrow P\ x$ **by** (*simp add*: *sums-def*)
moreover from *eventually-gt-at-top*[*of 1*]
have *eventually* $(\lambda n.\ (\sum r{<}n.\ D\ (real\ r + x)) = p\ n\ x)$ *at-top*
by (*auto elim*!: *eventually-mono simp*: *p-def*)
ultimately show *?thesis* **by** (*rule Lim-transform-eventually* [*rotated*])
qed

This gives us an expansion of the Digamma function:

lemma *Digamma-approx*:
assumes x: $(x :: real) > 0$
shows *Digamma* $x = ln\ x - 1\ /\ (2 * x) - P\ x$
proof −
have *eventually* $(\lambda n.\ ln\ (inverse\ (1 + x\ /\ real\ n)) + ln\ x - 1/(2{*}x) - p\ n\ x = ln\ (real\ n) - S'\ n\ x - 1/(2{*}x))$ *at-top*
using *eventually-gt-at-top*[*of 1*::*nat*]
proof *eventually-elim*
fix n :: *nat* **assume** n: $n > 1$
have $ln\ (real\ n) - S'\ n\ x = ln\ ((real\ n)\ /\ (real\ n + x)) + ln\ x - p\ n\ x$
using *assms n* **unfolding** *S′-approx* **by** (*subst ln-div*) (*auto simp*: *algebra-simps*)
also from n **have** $real\ n\ /\ (real\ n + x) = inverse\ (1 + x\ /\ real\ n)$ **by** (*simp add*: *field-simps*)
finally show $ln\ (inverse\ (1 + x\ /\ real\ n)) + ln\ x - 1/(2{*}x) - p\ n\ x =$

ln (*real n*) − *S′ n x* − *1*/(*2*∗*x*) **by** *simp*
qed
moreover have (λn. *ln* (*inverse* (*1* + *x* / *real n*)) + *ln x* − *1*/(*2*∗*x*) − *p n x*)
$\longlongrightarrow$ *ln* (*inverse* (*1* + *0*)) + *ln x* − *1*/(*2*∗*x*) − *P x*
by (*rule tendsto-intros p-LIMSEQ x real-tendsto-divide-at-top*
filterlim-real-sequentially | *simp*)+
hence (λn. *ln* (*inverse* (*1* + *x* / *real n*)) + *ln x* − *1*/(*2*∗*x*) − *p n x*)
$\longlongrightarrow$ *ln x* − *1*/(*2*∗*x*) − *P x* **by** *simp*
ultimately have (λn. *ln* (*real n*) − *S′ n x* − *1* / (*2* ∗ *x*)) $\longlongrightarrow$ *ln x* − *1*/(*2*∗*x*) − *P x*
by (*rule Lim-transform-eventually*)
moreover from *x* **have** (λn. *ln* (*real n*) − *S′ n x* − *1* / (*2* ∗ *x*)) $\longlongrightarrow$ *Digamma x*
by (*intro S′-LIMSEQ-Digamma*) *simp-all*
ultimately show *Digamma x* = *ln x* − *1* / (*2* ∗ *x*) − *P x*
by (*rule LIMSEQ-unique* [*rotated*])
qed

Next, we derive some bounds on P:

private lemma *p-ge-0*: *x* > *0* $\Longrightarrow$ *p n x* ≥ *0*
using *stirling-trapezium*[*of real n* + *x* **for** *n*]
by (*auto simp add*: *p-def intro*!: *sum-nonneg*)

private lemma *P-ge-0*: *x* > *0* $\Longrightarrow$ *P x* ≥ *0*
by (*rule tendsto-lowerbound*[*OF p-LIMSEQ*])
(*insert p-ge-0*[*of x*], *simp-all*)

private lemma *p-upper-bound*:
assumes *x* > *0 n* > *0*
shows *p n x* ≤ *1*/(*12*∗*x*ˆ*2*)
proof −
from *assms* **have** *p n x* = ($\sum$ *r*<*n*. *D* (*real r* + *x*))
by (*simp add*: *p-def*)
also have ... ≤ ($\sum$ *r*<*n*. *1*/(*12*∗(*real r* + *x*)ˆ*2*) − *1*/(*12* ∗ (*real* (*Suc r*) + *x*)ˆ*2*))
using *stirling-trapezium*[*of real r* + *x* **for** *r*] *assms*
by (*intro sum-mono*) (*simp add*: *add-ac*)
also have ... = *1* / (*12* ∗ x^2) − *1* / (*12* ∗ (*real n* + *x*)^2)
by (*subst sum-lessThan-telescope′*) *simp*
also have ... ≤ *1* / (*12* ∗ *x*ˆ*2*) **by** *simp*
finally show *?thesis* **.**
qed

private lemma *P-upper-bound*:
assumes *x* > *0*
shows *P x* ≤ *1*/(*12*∗*x*ˆ*2*)
proof (*rule tendsto-upperbound*)
show *eventually* (λn. *p n x* ≤ *1* / (*12* ∗ *x*ˆ*2*)) *at-top*
using *eventually-gt-at-top*[*of 0*] *p-upper-bound*[*of x*] *assms*

```
    by (auto elim!: eventually-mono)
  show (λn. p n x) ⟶ P x
    by (simp add: assms p-LIMSEQ)
qed auto
```

We can now use this approximation of the Digamma function to approximate *ln-Gamma* (since the former is the derivative of the latter). We therefore define the function g from Jameson's article, which measures the difference between *ln-Gamma* and its approximation:

```
private definition g :: real ⇒ real where
  g x = ln-Gamma x − (x − 1/2) ∗ ln x + x

private lemma DERIV-g: x > 0 ⟹ (g has-field-derivative −P x) (at x)
  unfolding g-def [abs-def] using Digamma-approx[of x]
  by (auto intro!: derivative-eq-intros simp: field-simps)

private lemma isCont-P:
  assumes x > 0
  shows   isCont P x
proof −
  define D' :: real ⇒ real
    where D' = (λx. − 1 / (2 ∗ xˆ2 ∗ (x+1)ˆ2))
  have DERIV-D: (D has-field-derivative D' x) (at x) if x > 0 for x
    unfolding D-def [abs-def] D'-def T-def
    by (insert that, (rule derivative-eq-intros refl | simp)+)
       (simp add: power2-eq-square divide-simps, (simp add: algebra-simps)?)
  note this [THEN DERIV-chain2, derivative-intros]

  have (P has-field-derivative (∑ n. D' (real n + x))) (at x)
    unfolding P-def [abs-def]
  proof (rule has-field-derivative-series')
    show convex {x/2<..} by simp
  next
    fix n :: nat and y :: real assume y: y ∈ {x/2<..}
    with assms have y > 0 by simp
     thus ((λa. D (real n + a)) has-real-derivative D' (real n + y)) (at y within
{x/2<..})
      by (auto intro!: derivative-eq-intros)
  next
    from assms D-summable[of x] show summable (λn. D (real n + x)) by simp
  next
    show uniformly-convergent-on {x/2<..} (λn x. ∑ i<n. D' (real i + x))
    proof (rule weierstrass-m-test')
      fix n :: nat and y :: real
      assume y: y ∈ {x/2<..}
      with assms have y > 0 by auto
      have norm (D' (real n + y)) = (1 / (2 ∗ (y + real n)ˆ2)) ∗ (1 / (y + real
(Suc n))ˆ2)
        by (simp add: D'-def add-ac)
```

also from y assms **have** ... ≤ (1 / (2 ∗ (x/2) ^2)) ∗ (1 / (real (Suc n)) ^2)
by (intro mult-mono divide-left-mono power-mono) simp-all
also have 1 / (real (Suc n)) ^2 = inverse ((real (Suc n)) ^2) **by** (simp add: field-simps)
finally show norm (D′ (real n + y)) ≤ (1 / (2 ∗ (x/2) ^2)) ∗ inverse ((real (Suc n)) ^2) **.**
next
show summable (λn. (1 / (2 ∗ (x/2) ^2)) ∗ inverse ((real (Suc n)) ^2))
by (subst summable-Suc-iff, intro summable-mult inverse-power-summable) simp-all
qed
qed (insert assms, simp-all add: interior-open)
thus ?thesis **by** (rule DERIV-isCont)
qed

private lemma P-continuous-on [THEN continuous-on-subset]: continuous-on {0<..} P
by (intro continuous-at-imp-continuous-on ballI isCont-P) auto

private lemma P-integrable:
assumes a: a > 0
shows P integrable-on {a..}
proof −
define f **where** f = (λn x. if x ∈ {a..real n} then P x else 0)
show P integrable-on {a..}
proof (rule dominated-convergence)
fix n :: nat
from a **have** P integrable-on {a..real n}
by (intro integrable-continuous-real P-continuous-on) auto
hence f n integrable-on {a..real n}
by (rule integrable-eq [rotated]) (simp add: f-def)
thus f n integrable-on {a..}
by (rule integrable-on-superset [rotated 2]) (auto simp: f-def)
next
fix n :: nat
show ∀x∈{a..}. norm (f n x) ≤ of-real (1/12) ∗ (1 / x^2)
using a P-ge-0 P-upper-bound **by** (auto simp: f-def)
next
show (λx::real. of-real (1/12) ∗ (1 / x^2)) integrable-on {a..}
using has-integral-inverse-power-to-inf[of 2 a] a
by (intro integrable-on-cmult-left) auto
next
show ∀x∈{a..}. (λn. f n x) ⟶ P x
proof safe
fix x :: real **assume** x: x ≥ a
have eventually (λn. real n ≥ x) at-top
using filterlim-real-sequentially **by** (simp add: filterlim-at-top)
with x **have** eventually (λn. f n x = P x) at-top
by (auto elim!: eventually-mono simp: f-def)

thus $(\lambda n.\ f\ n\ x) \longrightarrow P\ x$ **by** (*simp add*: *Lim-eventually*)
qed
qed
qed

private definition *c* :: *real* **where** $c = integral\ \{1..\}\ (\lambda x.\ -P\ x) + g\ 1$

We can now give bounds on *g*:

private lemma *g-bounds*:
assumes *x*: $x \geq 1$
shows $g\ x \in \{c..c + 1/(12*x)\}$
proof −
from *assms* **have** *int-nonneg*: $integral\ \{x..\}\ P \geq 0$
by (*intro Henstock-Kurzweil-Integration.integral-nonneg P-integrable*)
(*auto simp*: *P-ge-0*)
have *int-upper-bound*: $integral\ \{x..\}\ P \leq 1/(12*x)$
proof (*rule has-integral-le*)
from *x* **show** (*P has-integral integral* $\{x..\}$ *P*) $\{x..\}$
by (*intro integrable-integral P-integrable*) *simp-all*
from *x has-integral-mult-right*[*OF has-integral-inverse-power-to-inf*[*of 2 x*], *of 1/12*]
show $((\lambda x.\ 1/(12*x\hat{}2))$ *has-integral* $(1/(12*x)))\ \{x..\}$ **by** (*simp add*: *field-simps*)
qed (*insert P-upper-bound x*, *simp-all*)

note *DERIV-g* [*THEN DERIV-chain2*, *derivative-intros*]
from *assms* **have** *int1*: $((\lambda x.\ -P\ x)$ *has-integral* $(g\ x - g\ 1))\ \{1..x\}$
by (*intro fundamental-theorem-of-calculus*)
(*auto simp*: *has-field-derivative-iff-has-vector-derivative* [*symmetric*]
intro!: *derivative-eq-intros*)
from *x* **have** *int2*: $((\lambda x.\ -P\ x)$ *has-integral integral* $\{x..\}\ (\lambda x.\ -P\ x))\ \{x..\}$
by (*intro integrable-integral integrable-neg P-integrable*) *simp-all*
from *has-integral-union*[*OF int1 int2*] *x*
have $((\lambda x.\ -\ P\ x)$ *has-integral* $g\ x - g\ 1 - integral\ \{x..\}\ P)\ (\{1..x\} \cup \{x..\})$
by (*simp add*: *max-def*)
also from *x* **have** $\{1..x\} \cup \{x..\} = \{1..\}$ **by** *auto*
finally have $((\lambda x.\ -P\ x)$ *has-integral* $g\ x - g\ 1 - integral\ \{x..\}\ P)\ \{1..\}$ **.**
moreover have $((\lambda x.\ -P\ x)$ *has-integral integral* $\{1..\}\ (\lambda x.\ -P\ x))\ \{1..\}$
by (*intro integrable-integral integrable-neg P-integrable*) *simp-all*
ultimately have $g\ x - g\ 1 - integral\ \{x..\}\ P = integral\ \{1..\}\ (\lambda x.\ -P\ x)$
by (*simp add*: *has-integral-unique*)
hence $g\ x = c + integral\ \{x..\}\ P$ **by** (*simp add*: *c-def algebra-simps*)
with *int-upper-bound int-nonneg* **show** $g\ x \in \{c..c + 1/(12*x)\}$ **by** *simp*
qed

Finally, we have bounds on *ln-Gamma*, *Gamma*, and *fact*.

private lemma *ln-Gamma-bounds-aux*:
$x \geq 1 \Longrightarrow ln\text{-}Gamma\ x \geq c + (x - 1/2) * ln\ x - x$
$x \geq 1 \Longrightarrow ln\text{-}Gamma\ x \leq c + (x - 1/2) * ln\ x - x + 1/(12*x)$

using *g-bounds*[*of x*] **by** (*simp-all add*: *g-def*)

private lemma *Gamma-bounds-aux*:
assumes *x*: $x \geq 1$
shows *Gamma x* $\geq$ *exp c* $*$ *x powr* $(x - 1/2)$ / *exp x*
Gamma x $\leq$ *exp c* $*$ *x powr* $(x - 1/2)$ / *exp x* $*$ *exp* $(1/(12*x))$
proof −
have *exp* (*ln-Gamma x*) $\geq$ *exp* $(c + (x - 1/2) *$ *ln x* $- x)$
by (*subst exp-le-cancel-iff*, *rule ln-Gamma-bounds-aux*) (*simp add*: *x*)
with *x* **show** *Gamma x* $\geq$ *exp c* $*$ *x powr* $(x - 1/2)$ / *exp x*
by (*simp add*: *Gamma-real-pos-exp exp-add exp-diff powr-def del*: *exp-le-cancel-iff*)
next
have *exp* (*ln-Gamma x*) $\leq$ *exp* $(c + (x - 1/2) *$ *ln x* $- x + 1/(12*x))$
by (*subst exp-le-cancel-iff*, *rule ln-Gamma-bounds-aux*) (*simp add*: *x*)
with *x* **show** *Gamma x* $\leq$ *exp c* $*$ *x powr* $(x - 1/2)$ / *exp x* $*$ *exp* $(1/(12*x))$
by (*simp add*: *Gamma-real-pos-exp exp-add exp-diff powr-def del*: *exp-le-cancel-iff*)
qed

private lemma *Gamma-asymp-equiv-aux*:
Gamma $\sim$ (λx. *exp c* $*$ *x powr* $(x - 1/2)$ / *exp x*)
proof (*rule asymp-equiv-sandwich*)
show *eventually* (λx. *exp c* $*$ *x powr* $(x - 1/2)$ / *exp x* $\leq$ *Gamma x*) *at-top*
eventually (λx. *exp c* $*$ *x powr* $(x - 1/2)$ / *exp x* $*$ *exp* $(1/(12*x))$ $\geq$ *Gamma x*) *at-top*
using *eventually-ge-at-top*[*of 1*::*real*] *Gamma-bounds-aux*
by (*auto elim*!: *eventually-mono*)
have ((λx::*real*. *exp* $(1 / (12 * x))$) $\longrightarrow$ *exp 0*) *at-top*
by (*rule tendsto-intros real-tendsto-divide-at-top filterlim-tendsto-pos-mult-at-top*)+
(*simp-all add*: *filterlim-ident*)
hence (λx. *exp* $(1 / (12 * x))$) $\sim$ (λx. *1*)
by (*intro asymp-equivI'*) *simp-all*
hence (λx. *exp c* $*$ *x powr* $(x - 1 / 2)$ / *exp x* $* 1$) $\sim$
(λx. *exp c* $*$ *x powr* $(x - 1 / 2)$ / *exp x* $*$ *exp* $(1 / (12 * x))$)
by (*intro asymp-equiv-mult asymp-equiv-refl*) (*simp add*: *asymp-equiv-sym*)
thus (λx. *exp c* $*$ *x powr* $(x - 1 / 2)$ / *exp x*) $\sim$
(λx. *exp c* $*$ *x powr* $(x - 1 / 2)$ / *exp x* $*$ *exp* $(1 / (12 * x))$) **by** *simp*
qed *simp-all*

private lemma *exp-1-powr-real* [*simp*]: *exp* (*1*::*real*) *powr x* = *exp x*
by (*simp add*: *powr-def*)

private lemma *fact-asymp-equiv-aux*:
fact $\sim$ (λx. *exp c* $*$ *sqrt* (*real x*) $*$ (*real x* / *exp 1*) *powr real x*)
proof −
have *fact* $\sim$ (λn. *Gamma* (*real* (*Suc n*))) **by** (*simp add*: *Gamma-fact*)
also have *eventually* (λn. *Gamma* (*real* (*Suc n*)) = *real n* $*$ *Gamma* (*real n*)) *at-top*
using *eventually-gt-at-top*[*of 0*::*nat*] *Gamma-plus1*[*of real n* **for** *n*]
by (*auto elim*!: *eventually-mono simp*: *add-ac of-nat-in-nonpos-Ints-iff*)

also have ($\lambda n.$ *Gamma* (*real n*)) $\sim$ ($\lambda n.$ *exp c* $*$ *real n powr* (*real n* $-$ *1/2*) / *exp* (*real n*))
by (*rule asymp-equiv-compose'*[*OF Gamma-asymp-equiv-aux*] *filterlim-real-sequentially*)+
also have *eventually* ($\lambda n.$ *real n* $*$ (*exp c* $*$ *real n powr* (*real n* $-$ *1* / *2*) / *exp* (*real n*)) =
exp c $*$ *sqrt* (*real n*) $*$ (*real n* / *exp 1*) *powr real n*) *at-top*
using *eventually-gt-at-top*[*of 0::nat*]
proof *eventually-elim*
fix *n* :: *nat* **assume** *n*: *n* > *0*
thus *real n* $*$ (*exp c* $*$ *real n powr* (*real n* $-$ *1* / *2*) / *exp* (*real n*)) =
exp c $*$ *sqrt* (*real n*) $*$ (*real n* / *exp 1*) *powr real n*
by (*subst powr-divide2* [*symmetric*]) (*simp-all add*: *powr-divide powr-half-sqrt field-simps*)
qed
finally show *?thesis* **by** $-$ (*simp-all add*: *asymp-equiv-mult*)
qed

We still need to determine the constant term c, which we do using Wallis' product formula:

$$\prod_{n=1}^{\infty} \frac{4n^2}{4n^2-1} = \frac{\pi}{2}$$

private lemma *powr-mult-2*: (*x::real*) > *0* $\Longrightarrow$ *x powr* (*y* $*$ *2*) = (*x^2*) *powr y*
by (*subst mult.commute*, *subst powr-powr* [*symmetric*]) (*simp add*: *powr-numeral*)

private lemma *exp-mult-2*: *exp* (*y* $*$ *2* :: *real*) = *exp y* $*$ *exp y*
by (*subst exp-add* [*symmetric*]) *simp*

private lemma *exp-c*: *exp c* = *sqrt* (*2*$*$*pi*)
proof $-$
define *p* **where** *p* = ($\lambda n.$ $\prod$ *k*=*1..n*. (*4*$*$*real k^2*) / (*4*$*$*real k^2* $-$ *1*))

have *p-0* [*simp*]: *p 0* = *1* **by** (*simp add*: *p-def*)
have *p-Suc*: *p* (*Suc n*) = *p n* $*$ (*4* $*$ *real* (*Suc n*)*^2*) / (*4* $*$ *real* (*Suc n*)*^2* $-$ *1*)
for *n* **unfolding** *p-def* **by** (*subst prod-nat-ivl-Suc'*) *simp-all*
have *p*: *p* = ($\lambda n.$ *16 ^ n* $*$ *fact n ^ 4* / (*fact* (*2* $*$ *n*))^2 / (*2* $*$ *real n* + *1*))
proof
fix *n* :: *nat*
have *p n* = ($\prod$ *k*=*1..n*. (*2*$*$*real k*)*^2* / (*2*$*$*real k* $-$ *1*) / (*2* $*$ *real k* + *1*))
unfolding *p-def* **by** (*intro prod.cong refl*) (*simp add*: *field-simps power2-eq-square*)
also have ... = ($\prod$ *k*=*1..n*. (*2*$*$*real k*)*^2* / (*2*$*$*real k* $-$ *1*)) / ($\prod$ *k*=*1..n*. (*2* $*$ *real* (*Suc k*) $-$ *1*))
by (*simp add*: *prod-dividef prod.distrib add-ac*)
also have ($\prod$ *k*=*1..n*. (*2* $*$ *real* (*Suc k*) $-$ *1*)) = ($\prod$ *k*=*Suc 1..Suc n*. (*2* $*$ *real k* $-$ *1*))
by (*subst prod.atLeast-Suc-atMost-Suc-shift*) *simp-all*
also have ... = ($\prod$ *k*=*1..n*. (*2* $*$ *real k* $-$ *1*)) $*$ (*2* $*$ *real n* + *1*)
by (*induction n*) (*simp-all add*: *prod-nat-ivl-Suc'*)
also have ($\prod$ *k* = *1..n*. (*2* $*$ *real k*)^2 / (*2* $*$ *real k* $-$ *1*)) / ... =

```
            (∏ k = 1..n. (2 * real k)^2 / (2 * real k − 1)^2) / (2 * real n + 1)
      unfolding power2-eq-square by (simp add: prod.distrib prod-dividef)
    also have (∏ k = 1..n. (2 * real k)^2 / (2 * real k − 1)^2) =
                (∏ k = 1..n. (2 * real k)^4 / ((2*real k)*(2 * real k − 1))^2)
      by (intro prod.cong refl)
       (simp add: divide-simps, (simp add: field-simps power2-eq-square eval-nat-numeral))
    also have ... = 16^n * fact n^4 / (∏ k=1..n. (2*real k) * (2*real k − 1))^2
      by (simp add: prod.distrib prod-dividef fact-prod
            prod-power-distrib [symmetric] prod-constant)
    also have (∏ k=1..n. (2*real k) * (2*real k − 1)) = fact (2*n)
      by (induction n) (simp-all add: algebra-simps prod-nat-ivl-Suc')
    finally show p n = 16 ^ n * fact n ^ 4 / (fact (2 * n))² / (2 * real n + 1) .
  qed

  have p ∼ (λn. 16 ^ n * fact n ^ 4 / (fact (2 * n))² / (2 * real n + 1))
    by (simp add: p)
  also have ... ∼ (λn. 16^n * (exp c * sqrt (real n) * (real n / exp 1) powr real
n)^4 /
                        (exp c * sqrt (real (2*n)) * (real (2*n) / exp 1) powr real
(2*n))^2 /
                      (2 * real n + 1)) (is - ∼ ?f)
   by (intro asymp-equiv-mult asymp-equiv-divide asymp-equiv-refl mult-nat-left-at-top
                fact-asymp-equiv-aux asymp-equiv-power asymp-equiv-compose'[OF
fact-asymp-equiv-aux])
       simp-all
  also have eventually (λn. ... n = exp c ^ 2 / (4 + 2/n)) at-top
    using eventually-gt-at-top[of 0::nat]
  proof eventually-elim
    fix n :: nat assume n: n > 0
     have [simp]: 16^n = 4^n * (4^n :: real) by (simp add: power-mult-distrib
[symmetric])
    from n have ?f n = exp c ^ 2 * (n / (2*(2*n+1)))
     by (simp add: power-mult-distrib divide-simps powr-mult powr-divide real-sqrt-power-even)
        (simp add: field-simps power2-eq-square eval-nat-numeral powr-mult-2
                exp-mult-2 powr-realpow)
    also from n have ... = exp c ^ 2 / (4 + 2/n) by (simp add: field-simps)
    finally show ?f n = ... .
  qed
  also have (λx. 4 + 2 / real x) ∼ (λx. 4)
    by (subst asymp-equiv-add-right) auto
  finally have p ⟶ exp c ^ 2 / 4
    by (rule asymp-equivD-const) (simp-all add: asymp-equiv-divide)
  moreover have p ⟶ pi / 2 unfolding p-def by (rule wallis)
  ultimately have exp c ^ 2 / 4 = pi / 2 by (rule LIMSEQ-unique)
  hence 2 * pi = exp c ^ 2 by simp
  also have sqrt (exp c ^ 2) = exp c by simp
  finally show exp c = sqrt (2 * pi) ..
qed
```

private lemma *c*: *c* = *ln* (*2*∗*pi*) / *2*
proof −
  note *exp-c* [*symmetric*]
  also have *ln* (*exp c*) = *c* **by** *simp*
  finally show *?thesis* **by** (*simp add*: *ln-sqrt*)
qed

This gives us the final bounds:

theorem *Gamma-bounds*:
  assumes *x* ≥ *1*
  shows *Gamma x* ≥ *sqrt* (*2*∗*pi*/*x*) ∗ (*x* / *exp 1*) *powr x* (**is** *?th1*)
    Gamma x ≤ *sqrt* (*2*∗*pi*/*x*) ∗ (*x* / *exp 1*) *powr x* ∗ *exp* (*1* / (*12* ∗ *x*)) (**is** *?th2*)
proof −
  from *assms* **have** *exp c* ∗ *x powr* (*x* − *1*/*2*) / *exp x* = *sqrt* (*2*∗*pi*/*x*) ∗ (*x* / *exp 1*) *powr x*
    by (*subst powr-divide2* [*symmetric*])
      (*simp add*: *exp-c real-sqrt-divide powr-divide powr-half-sqrt field-simps*)
  with *Gamma-bounds-aux*[*OF assms*] **show** *?th1 ?th2* **by** *simp-all*
qed

theorem *ln-Gamma-bounds*:
  assumes *x* ≥ *1*
  shows *ln-Gamma x* ≥ *ln* (*2*∗*pi*/*x*) / *2* + *x* ∗ *ln x* − *x* (**is** *?th1*)
    ln-Gamma x ≤ *ln* (*2*∗*pi*/*x*) / *2* + *x* ∗ *ln x* − *x* + *1*/(*12*∗*x*) (**is** *?th2*)
proof −
  from *ln-Gamma-bounds-aux*[*OF assms*] *assms* **show** *?th1 ?th2*
    by (*simp-all add*: *c field-simps ln-div*)
  from *assms* **have** *exp c* ∗ *x powr* (*x* − *1*/*2*) / *exp x* = *sqrt* (*2*∗*pi*/*x*) ∗ (*x* / *exp 1*) *powr x*
    by (*subst powr-divide2* [*symmetric*])
      (*simp add*: *exp-c real-sqrt-divide powr-divide powr-half-sqrt field-simps*)
qed

theorem *fact-bounds*:
  assumes *n* > *0*
  shows (*fact n* :: *real*) ≥ *sqrt* (*2*∗*pi*∗*n*) ∗ (*n* / *exp 1*) ^ *n* (**is** *?th1*)
    (*fact n* :: *real*) ≤ *sqrt* (*2*∗*pi*∗*n*) ∗ (*n* / *exp 1*) ^ *n* ∗ *exp* (*1* / (*12* ∗ *n*)) (**is** *?th2*)
proof −
  from *assms* **have** *n*: *real n* ≥ *1* **by** *simp*
  from *assms Gamma-plus1*[*of real n*]
    have *real n* ∗ *Gamma* (*real n*) = *Gamma* (*real* (*Suc n*))
    by (*simp add*: *of-nat-in-nonpos-Ints-iff add-ac*)
  also have *Gamma* (*real* (*Suc n*)) = *fact n* **by** (*subst Gamma-fact* [*symmetric*]) *simp*
  finally have ∗: *fact n* = *real n* ∗ *Gamma* (*real n*) **by** *simp*

  have *2*∗*pi*/*n* = *2*∗*pi*∗*n* / *n*^*2* **by** (*simp add*: *power2-eq-square*)

also have *sqrt* ... = *sqrt* (*2*∗*pi*∗*n*) / *n* **by** (*subst real-sqrt-divide*) *simp-all*
also have *real n* ∗ ... = *sqrt* (*2*∗*pi*∗*n*) **by** *simp*
finally have ∗∗: *real n* ∗ *sqrt* (*2*∗*pi*/*real n*) = *sqrt* (*2*∗*pi*∗*real n*) **.**

note ∗
also note *Gamma-bounds*(*2*)[*OF n*]
also have *real n* ∗ (*sqrt* (*2* ∗ *pi* / *real n*) ∗ (*real n* / *exp 1*) *powr real n* ∗ *exp* (*1* / (*12* ∗ *real n*))) =
(*real n* ∗ *sqrt* (*2*∗*pi*/*n*)) ∗ (*n* / *exp 1*) *powr n* ∗ *exp* (*1* / (*12* ∗ *n*))
by (*simp add*: *algebra-simps*)
also from *n* **have** (*real n* / *exp 1*) *powr real n* = (*real n* / *exp 1*) ^ *n*
by (*subst powr-realpow*) *simp-all*
also note ∗∗
finally show *?th2* **by** − (*insert assms*, *simp-all*)

have *sqrt* (*2*∗*pi*∗*n*) ∗ (*n* / *exp 1*) *powr n* = *n* ∗ (*sqrt* (*2*∗*pi*/*n*) ∗ (*n* / *exp 1*) *powr n*)
by (*subst* ∗∗ [*symmetric*]) (*simp add*: *field-simps*)
also from *assms* **have** ... ≤ *real n* ∗ *Gamma* (*real n*)
by (*intro mult-left-mono Gamma-bounds*(*1*)) *simp-all*
also from *n* **have** (*real n* / *exp 1*) *powr real n* = (*real n* / *exp 1*) ^ *n*
by (*subst powr-realpow*) *simp-all*
also note ∗ [*symmetric*]
finally show *?th1* **.**
qed

theorem *ln-fact-bounds*:
assumes *n* > *0*
shows *ln* (*fact n* :: *real*) ≥ *ln* (*2*∗*pi*∗*n*)/*2* + *n* ∗ *ln n* − *n* (**is** *?th1*)
ln (*fact n* :: *real*) ≤ *ln* (*2*∗*pi*∗*n*)/*2* + *n* ∗ *ln n* − *n* + *1*/(*12*∗*real n*) (**is** *?th2*)
proof −
have *ln* (*fact n* :: *real*) ≥ *ln* (*sqrt* (*2*∗*pi*∗*real n*)∗(*real n*/*exp 1*)^*n*)
using *fact-bounds*(*1*)[*OF assms*] *assms* **by** (*subst ln-le-cancel-iff*) *auto*
also from *assms* **have** *ln* (*sqrt* (*2*∗*pi*∗*real n*)∗(*real n*/*exp 1*)^*n*) = *ln* (*2*∗*pi*∗*n*)/*2* + *n* ∗ *ln n* − *n*
by (*simp add*: *ln-mult ln-div ln-realpow ln-sqrt field-simps*)
finally show *?th1* **.**
next
have *ln* (*fact n* :: *real*) ≤ *ln* (*sqrt* (*2*∗*pi*∗*real n*) ∗ (*real n*/*exp 1*)^*n* ∗ *exp* (*1*/(*12*∗*real n*)))
using *fact-bounds*(*2*)[*OF assms*] *assms* **by** (*subst ln-le-cancel-iff*) *auto*
also from *assms* **have** ... = *ln* (*2*∗*pi*∗*n*)/*2* + *n* ∗ *ln n* − *n* + *1*/(*12*∗*real n*)
by (*simp add*: *ln-mult ln-div ln-realpow ln-sqrt field-simps*)
finally show *?th2* **.**
qed

theorem *Gamma-asymp-equiv*:
Gamma ∼ (λ*x*. *sqrt* (*2*∗*pi*/*x*) ∗ (*x* / *exp 1*) *powr x* :: *real*)

```
proof -
  note Gamma-asymp-equiv-aux
  also have eventually (λx. exp c * x powr (x - 1/2) / exp x =
              sqrt (2*pi/x) * (x / exp 1) powr x) at-top
    using eventually-gt-at-top[of 0::real]
  proof eventually-elim
    fix x :: real assume x: x > 0
    thus exp c * x powr (x - 1/2) / exp x = sqrt (2*pi/x) * (x / exp 1) powr x
      by (subst powr-divide2 [symmetric])
         (simp add: exp-c powr-half-sqrt powr-divide field-simps real-sqrt-divide)
  qed
  finally show ?thesis .
qed

theorem fact-asymp-equiv:
  fact ∼ (λn. sqrt (2*pi*n) * (n / exp 1) ^ n :: real)
proof -
  note fact-asymp-equiv-aux
  also have eventually (λn. exp c * sqrt (real n) = sqrt (2 * pi * real n)) at-top
   using eventually-gt-at-top[of 0::nat] by eventually-elim (simp add: exp-c real-sqrt-mult)
  also have eventually (λn. (n / exp 1) powr n = (n / exp 1) ^ n) at-top
   using eventually-gt-at-top[of 0::nat] by eventually-elim (simp add: powr-realpow)
  finally show ?thesis .
qed

end

end
```

References

[1] G. J. O. Jameson. A simple proof of Stirling's formula for the Gamma function. *The Mathematical Gazette*, 99:68–74, 3 2015.