

Actuarial Mathematics

Yosuke Ito

March 24, 2023

Abstract

Actuarial Mathematics is a theory in applied mathematics, which is mainly used for determining the prices of insurance products and evaluating the liability of a company associating with insurance contracts. It is related to calculus, probability theory and financial theory, etc.

In this entry, I formalize the very basic part of Actuarial Mathematics in Isabelle/HOL. The first formalization is about the theory of interest which deals with interest rates, present value factors, an annuity certain, etc.

I have already formalized the basic part of Actuarial Mathematics in Coq (<https://github.com/Yosuke-Ito-345/Actuary>). This entry is currently the partial translation and a little generalization of the Coq formalization. The further translation in Isabelle/HOL is now proceeding.

Contents

1 Preliminary Definitions and Lemmas	1
2 List of Actuarial Notations (Global Scope)	5
3 Theory of Interest	7

theory *Preliminaries*

imports *HOL-Analysis.Analysis*

begin

notation *powr* (**infixr** $\hat{\ } 80$)

1 Preliminary Definitions and Lemmas

lemma *seq-part-multiple*: **fixes** $m\ n :: nat$ **assumes** $m \neq 0$ **defines** $A \equiv \lambda i :: nat. \{i * m ..< (i+1) * m\}$

shows $\forall i\ j. i \neq j \longrightarrow A\ i \cap A\ j = \{\}$ **and** $(\bigcup_{i < n. A\ i}) = \{..< n * m\}$

proof –

{ **fix** $i\ j :: nat$

```

have  $i \neq j \implies A\ i \cap A\ j = \{\}$ 
proof (erule contrapos-np)
  assume  $A\ i \cap A\ j \neq \{\}$ 
  then obtain  $k$  where  $k \in A\ i \cap A\ j$  by blast
  hence  $i*m < (j+1)*m \wedge j*m < (i+1)*m$  unfolding A-def by force
  hence  $i < j+1 \wedge j < i+1$  using mult-less-cancel2 by blast
  thus  $i = j$  by force
qed }
thus  $\forall i\ j. i \neq j \longrightarrow A\ i \cap A\ j = \{\}$  by blast
next
show  $(\bigcup_{i < n}. A\ i) = \{.. < n*m\}$ 
proof
  show  $(\bigcup_{i < n}. A\ i) \subseteq \{.. < n*m\}$ 
  proof
    fix  $x::nat$ 
    assume  $x \in (\bigcup_{i < n}. A\ i)$ 
    then obtain  $i$  where  $i-n: i < n$  and  $i-x: x < (i+1)*m$  unfolding A-def by
force
    hence  $i+1 \leq n$  by linarith
    hence  $x < n*m$  by (meson less-le-trans mult-le-cancel2 i-x)
    thus  $x \in \{.. < n*m\}$ 
    using diff-mult-distrib mult-1 i-n by auto
  qed
next
show  $\{.. < n*m\} \subseteq (\bigcup_{i < n}. A\ i)$ 
proof
  fix  $x::nat$ 
  let  $?i = x\ div\ m$ 
  assume  $x \in \{.. < n*m\}$ 
  hence  $?i < n$  by (simp add: less-mult-imp-div-less)
  moreover have  $?i*m \leq x \wedge x < (?i+1)*m$ 
    using assms div-times-less-eq-dividend dividend-less-div-times by auto
  ultimately show  $x \in (\bigcup_{i < n}. A\ i)$  unfolding A-def by force
qed
qed
qed

```

lemma (*in field*) *divide-mult-cancel[simp]*: fixes $a\ b$ assumes $b \neq 0$
 shows $a / b * b = a$
 by (*simp add: assms*)

lemma *inverse-powr*: $(1/a).\ ^b = a.\ ^{-b}$ if $a > 0$ for $a\ b :: real$
 by (*smt that powr-divide powr-minus-divide powr-one-eq-one*)

lemma *powr-eq-one-iff-gen[simp]*: $a.\ ^x = 1 \iff x = 0$ if $a > 0\ a \neq 1$ for $a\ x :: real$
 by (*metis powr-eq-0-iff powr-inj powr-zero-eq-one that*)

lemma *powr-less-cancel2*: $0 < a \implies 0 < x \implies 0 < y \implies x.\ ^a < y.\ ^a \implies x <$

```

y
  for a x y ::real
  proof -
    assume a-pos: 0 < a and x-pos: 0 < x and y-pos: 0 < y
    show x.^a < y.^a  $\implies$  x < y
    proof (erule contrapos-pp)
      assume  $\neg$  x < y
      hence x  $\geq$  y by fastforce
      hence x.^a  $\geq$  y.^a
      proof (cases x = y)
        case True
          thus ?thesis by simp
        next
          case False
            hence x.^a > y.^a
              using  $\langle$ x  $\geq$  y $\rangle$  powr-less-mono2 a-pos y-pos by auto
            thus ?thesis by auto
      qed
    thus  $\neg$  x.^a < y.^a by fastforce
  qed
qed

lemma geometric-increasing-sum-aux: (1-r)^2 * ( $\sum$  k<n. (k+1)*r^k) = 1 -
(n+1)*r^n + n*r^(n+1)
  for n::nat and r::real
  proof (induct n)
    case 0
      thus ?case by simp
    next
      case (Suc n)
        thus ?case
          by (simp add: distrib-left power2-diff field-simps power2-eq-square)
  qed

lemma geometric-increasing-sum: ( $\sum$  k<n. (k+1)*r^k) = (1 - (n+1)*r^n +
n*r^(n+1)) / (1-r)^2
  if r  $\neq$  1 for n::nat and r::real
  by (subst geometric-increasing-sum-aux[THEN sym], simp add: that)

lemma Reals-UNIV[simp]:  $\mathbb{R}$  = {x::real. True}
  unfolding Reals-def by auto

lemma DERIV-fun-powr2:
  fixes a::real
  assumes a-pos: a > 0
    and f: DERIV f x :> r
  shows DERIV ( $\lambda$ x. a.^(f x)) x :> a.^(f x) * r * ln a
  proof -
    let ?g = ( $\lambda$ x. a)

```

have g : *DERIV* ? g x $>$ 0 **by** *simp*
have pos : ? g x $>$ 0 **by** (*simp add*: a -*pos*)
show ?*thesis*
using *DERIV-powr*[*OF* g pos f] a -*pos* **by** (*auto simp add*: *field-simps*)
qed

lemma *has-real-derivative-powr2*:
assumes a -*pos*: $a > 0$
shows $((\lambda x. a. \hat{x})$ *has-real-derivative* $a. \hat{x} * \ln a$) (at x)
proof –
let ? f = $(\lambda x. x::real)$
have f : *DERIV* ? f x $>$ 1 **by** *simp*
thus ?*thesis* **using** *DERIV-fun-powr2*[*OF* a -*pos* f] **by** *simp*
qed

lemma *has-integral-powr2-from-0*:
fixes a c :: *real*
assumes a -*pos*: $a > 0$ **and** a -*neg-1*: $a \neq 1$ **and** c -*nneg*: $c \geq 0$
shows $((\lambda x. a. \hat{x})$ *has-integral* $((a. \hat{c} - 1) / (\ln a))$) {0.. c }
proof –
have $((\lambda x. a. \hat{x})$ *has-integral* $((a. \hat{c}) / (\ln a) - (a. \hat{0}) / (\ln a))$) {0.. c }
proof (*rule fundamental-theorem-of-calculus*[*OF* c -*nneg*])
fix x ::*real*
assume $x \in \{0..c\}$
show $((\lambda y. a. \hat{y} / \ln a)$ *has-vector-derivative* $a. \hat{x}$) (at x within {0.. c })
using *has-real-derivative-powr2*[*OF* a -*pos*, of x]
apply –
apply (*drule* *DERIV-cdivide*[**where** $c = \ln a$], *simp add*: *assms*)
apply (*rule* *has-vector-derivative-within-subset*[**where** $S = UNIV$ **and** $T = \{0..c\}$],
auto)
by (*rule* *iffD1*[*OF* *has-real-derivative-iff-has-vector-derivative*])
qed
thus ?*thesis*
using *assms powr-zero-eq-one* **by** (*simp add*: *field-simps*)
qed

lemma *integrable-on-powr2-from-0*:
fixes a c :: *real*
assumes a -*pos*: $a > 0$ **and** a -*neg-1*: $a \neq 1$ **and** c -*nneg*: $c \geq 0$
shows $(\lambda x. a. \hat{x})$ *integrable-on* {0.. c }
using *has-integral-powr2-from-0*[*OF* *assms*] **unfolding** *integrable-on-def* **by** *blast*

lemma *integrable-on-powr2-from-0-general*:
fixes a c :: *real*
assumes a -*pos*: $a > 0$ **and** c -*nneg*: $c \geq 0$
shows $(\lambda x. a. \hat{x})$ *integrable-on* {0.. c }
proof (*cases* $a = 1$)
case *True*
thus ?*thesis*

```

    using has-integral-const-real by auto
next
case False
thus ?thesis
    using has-integral-powr2-from-0 False assms by auto
qed

```

```

lemma has-integral-null-interval: fixes a b :: real and f::real ⇒ real assumes a
≥ b
shows (f has-integral 0) {a..b}
using assms content-real-eq-0 by blast

```

```

lemma has-integral-interval-reverse: fixes f :: real ⇒ real and a b :: real
assumes a < b
and continuous-on {a..b} f
shows ((λx. f (a+b-x)) has-integral (integral {a..b} f)) {a..b}
proof -
let ?g = λx. a + b - x
let ?g' = λx. -1
have g-C0: continuous-on {a..b} ?g using continuous-on-op-minus by simp
have Dg-g': ∧x. x∈{a..b} ⇒ (?g has-field-derivative ?g' x) (at x within {a..b})
by (auto intro!: derivative-eq-intros)
show ?thesis
using has-integral-substitution-general
[of { } a b ?g a b f, simplified, OF assms g-C0 Dg-g', simplified]
apply (simp add: has-integral-null-interval[OF assms(1), THEN integral-unique])
by (simp add: has-integral-neg-iff)
qed

```

```

end
theory Interest
imports Preliminaries
begin

```

2 List of Actuarial Notations (Global Scope)

```

definition i-nom :: real ⇒ nat ⇒ real ($i[-] ^{-} [0,0] 200)
where $i[i] ^{m} ≡ m * ((1+i). ^{1/m} - 1) — nominal interest rate
definition i-force :: real ⇒ real ($δ[-] [0] 200)
where $δ[i] ≡ ln (1+i) — force of interest
definition d-nom :: real ⇒ nat ⇒ real ($d[-] ^{-} [0,0] 200)
where $d[i] ^{m} ≡ $i[i] ^{m} / (1 + $i[i] ^{m}/m) — discount rate
abbreviation d-nom-yr :: real ⇒ real ($d[-] [0] 200)
where $d[i] ≡ $d[i] ^{1} — Post-fix "yr" stands for "year".
definition v-pres :: real ⇒ real ($v[-] [0] 200)
where $v[i] ≡ 1 / (1+i) — present value factor
definition ann :: real ⇒ nat ⇒ nat ⇒ real ($a[-] ^{-}'-- [0,0,101] 200)
where $a[i] ^{m}-n ≡ ∑ k<n*m. $v[i]. ^{(k+1::nat)/m} / m
— present value of an immediate annuity

```

abbreviation $ann\text{-}yr :: real \Rightarrow nat \Rightarrow real$ ($\$a[-]'\text{-} [0,101]$ 200)
where $\$a[i]\text{-}n \equiv \$a[i] \hat{\sim} \{1\}\text{-}n$

definition $acc :: real \Rightarrow nat \Rightarrow nat \Rightarrow real$ ($\$s[-] \hat{\sim} \{-}\text{-} [0,0,101]$ 200)
where $\$s[i] \hat{\sim} \{m\}\text{-}n \equiv \sum_{k < n * m}. (1+i). \hat{\sim}((k::nat)/m) / m$
— future value of an immediate annuity
— The name "acc" stands for "accumulation".

abbreviation $acc\text{-}yr :: real \Rightarrow nat \Rightarrow real$ ($\$s[-]'\text{-} [0]$ 200)
where $\$s[i]\text{-}n \equiv \$s[i] \hat{\sim} \{1\}\text{-}n$

definition $ann\text{-}due :: real \Rightarrow nat \Rightarrow nat \Rightarrow real$ ($\$a''''[-] \hat{\sim} \{-}\text{-} [0,0,101]$ 200)
where $\$a''[i] \hat{\sim} \{m\}\text{-}n \equiv \sum_{k < n * m}. \$v[i]. \hat{\sim}((k::nat)/m) / m$
— present value of an annuity-due

abbreviation $ann\text{-}due\text{-}yr :: real \Rightarrow nat \Rightarrow real$ ($\$a''''[-]'\text{-} [0,101]$ 200)
where $\$a''[i]\text{-}n \equiv \$a''[i] \hat{\sim} \{1\}\text{-}n$

definition $acc\text{-}due :: real \Rightarrow nat \Rightarrow nat \Rightarrow real$ ($\$s''''[-] \hat{\sim} \{-}\text{-} [0,0,101]$ 200)
where $\$s''[i] \hat{\sim} \{m\}\text{-}n \equiv \sum_{k < n * m}. (1+i). \hat{\sim}((k+1::nat)/m) / m$
— future value of an annuity-due

abbreviation $acc\text{-}due\text{-}yr :: real \Rightarrow nat \Rightarrow real$ ($\$s''''[-]'\text{-} [0,101]$ 200)
where $\$s''[i]\text{-}n \equiv \$s''[i] \hat{\sim} \{1\}\text{-}n$

definition $ann\text{-}cont :: real \Rightarrow real \Rightarrow real$ ($\$a''[-]'\text{-} [0,101]$ 200)
where $\$a''[i]\text{-}n \equiv integral \{0..n\} (\lambda t::real. \$v[i]. \hat{\sim} t)$
— present value of a continuous annuity

definition $acc\text{-}cont :: real \Rightarrow real \Rightarrow real$ ($\$s''[-]'\text{-} [0,101]$ 200)
where $\$s''[i]\text{-}n \equiv integral \{0..n\} (\lambda t::real. (1+i). \hat{\sim} t)$
— future value of a continuous annuity

definition $perp :: real \Rightarrow nat \Rightarrow real$ ($\$a[-] \hat{\sim} \{-}\text{-}'\text{-}\infty [0,0]$ 200)
where $\$a[i] \hat{\sim} \{m\}\text{-}\infty \equiv 1 / \$i[i] \hat{\sim} \{m\}$
— present value of a perpetual annuity

abbreviation $perp\text{-}yr :: real \Rightarrow real$ ($\$a[-]'\text{-}\infty [0]$ 200)
where $\$a[i]\text{-}\infty \equiv \$a[i] \hat{\sim} \{1\}\text{-}\infty$

definition $perp\text{-}due :: real \Rightarrow nat \Rightarrow real$ ($\$a''''[-] \hat{\sim} \{-}\text{-}'\text{-}\infty [0,0]$ 200)
where $\$a''[i] \hat{\sim} \{m\}\text{-}\infty \equiv 1 / \$d[i] \hat{\sim} \{m\}$
— present value of a perpetual annuity-due

abbreviation $perp\text{-}due\text{-}yr :: real \Rightarrow real$ ($\$a''''[-]'\text{-}\infty [0]$ 200)
where $\$a''[i]\text{-}\infty \equiv \$a''[i] \hat{\sim} \{1\}\text{-}\infty$

definition $ann\text{-}incr :: nat \Rightarrow real \Rightarrow nat \Rightarrow nat \Rightarrow real$
($\$(I \hat{\sim} \{-} a')[-] \hat{\sim} \{-}\text{-}'\text{-} [0,0,0,101]$ 200)
where $\$(I \hat{\sim} \{l\} a)[i] \hat{\sim} \{m\}\text{-}n \equiv \sum_{k < n * m}. \$v[i]. \hat{\sim}((k+1::nat)/m) * [l*(k+1::nat)/m] / (l*m)$
— present value of an increasing annuity
— This is my original definition.
— Here, "l" represents the number of increments per unit time.

abbreviation $ann\text{-}incr\text{-}lvl :: real \Rightarrow nat \Rightarrow nat \Rightarrow real$
($\$(Ia')[-] \hat{\sim} \{-}\text{-}'\text{-} [0,0,101]$ 200)
where $\$(Ia)[i] \hat{\sim} \{m\}\text{-}n \equiv \$(I \hat{\sim} \{1\} a)[i] \hat{\sim} \{m\}\text{-}n$
— The post-fix "lvl" stands for "level".

abbreviation $ann\text{-}incr\text{-}yr :: real \Rightarrow nat \Rightarrow real$ ($\$(Ia')[-]'\text{-} [0,101]$ 200)
where $\$(Ia)[i]\text{-}n \equiv \$(Ia)[i] \hat{\sim} \{1\}\text{-}n$

definition $acc\text{-}incr :: nat \Rightarrow real \Rightarrow nat \Rightarrow nat \Rightarrow real$
($\$(I \hat{\sim} \{-} s')[-] \hat{\sim} \{-}\text{-}'\text{-} [0,0,0,101]$ 200)

where $\$(I\{l\}s)[i]\{m\}-n \equiv \sum_{k < n * m}. (1+i). \wedge(n-(k+1::nat)/m) * [l*(k+1::nat)/m]$
 $/ (l*m)$
— future value of an increasing annuity
abbreviation *acc-incr-lvl* :: *real* \Rightarrow *nat* \Rightarrow *nat* \Rightarrow *real*
 $(\$(Is')[-]\{-}\{-}'-- [0,0,101] 200)$
where $\$(Is)[i]\{m\}-n \equiv \$(I\{1\}s)[i]\{m\}-n$
abbreviation *acc-incr-yr* :: *real* \Rightarrow *nat* \Rightarrow *real* $(\$(Is')[-]'-- [0,101] 200)$
where $\$(Is)[i]-n \equiv \$(Is)[i]\{1\}-n$
definition *ann-due-incr* :: *nat* \Rightarrow *real* \Rightarrow *nat* \Rightarrow *nat* \Rightarrow *real*
 $(\$(I\{-}a''''')[-]\{-}\{-}'-- [0,0,0,101] 200)$
where $\$(I\{l\}a'')[i]\{m\}-n \equiv \sum_{k < n * m}. \$v[i]. \wedge(k::nat)/m * [l*(k+1::nat)/m]$
 $/ (l*m)$
abbreviation *ann-due-incr-lvl* :: *real* \Rightarrow *nat* \Rightarrow *nat* \Rightarrow *real*
 $(\$(Ia''''')[-]\{-}\{-}'-- [0,0,101] 200)$
where $\$(Ia'')[i]\{m\}-n \equiv \$(I\{1\}a'')[i]\{m\}-n$
abbreviation *ann-due-incr-yr* :: *real* \Rightarrow *nat* \Rightarrow *real* $(\$(Ia''''')[-]'-- [0,101] 200)$
where $\$(Ia'')[i]-n \equiv \$(Ia'')[i]\{1\}-n$
definition *acc-due-incr* :: *nat* \Rightarrow *real* \Rightarrow *nat* \Rightarrow *nat* \Rightarrow *real*
 $(\$(I\{-}s''''')[-]\{-}\{-}'-- [0,0,0,101] 200)$
where $\$(I\{l\}s')[i]\{m\}-n \equiv \sum_{k < n * m}. (1+i). \wedge(n-(k::nat)/m) * [l*(k+1::nat)/m]$
 $/ (l*m)$
abbreviation *acc-due-incr-lvl* :: *real* \Rightarrow *nat* \Rightarrow *nat* \Rightarrow *real*
 $(\$(Is''''')[-]\{-}\{-}'-- [0,0,101] 200)$
where $\$(Is'')[i]\{m\}-n \equiv \$(I\{1\}s'')[i]\{m\}-n$
abbreviation *acc-due-incr-yr* :: *real* \Rightarrow *nat* \Rightarrow *real* $(\$(Is''''')[-]'-- [0,101] 200)$
where $\$(Is'')[i]-n \equiv \$(Is'')[i]\{1\}-n$
definition *perp-incr* :: *nat* \Rightarrow *real* \Rightarrow *nat* \Rightarrow *real* $(\$(I\{-}a')[-]\{-}\{-}'-\infty [0,0,0]$
 $200)$
where $\$(I\{l\}a)[i]\{m\}-\infty \equiv \lim (\lambda n. \$(I\{l\}a)[i]\{m\}-n)$
abbreviation *perp-incr-lvl* :: *real* \Rightarrow *nat* \Rightarrow *real* $(\$(Ia)[-]\{-}\{-}'-\infty [0,0] 200)$
where $\$(Ia)[i]\{m\}-\infty \equiv \$(I\{1\}a)[i]\{m\}-\infty$
abbreviation *perp-incr-yr* :: *real* \Rightarrow *real* $(\$(Ia)[-]'-\infty [0] 200)$
where $\$(Ia)[i]-\infty \equiv \$(Ia)[i]\{1\}-\infty$
definition *perp-due-incr* :: *nat* \Rightarrow *real* \Rightarrow *nat* \Rightarrow *real* $(\$(I\{-}a''''')[-]\{-}\{-}'-\infty$
 $[0,0,0] 200)$
where $\$(I\{l\}a'')[i]\{m\}-\infty \equiv \lim (\lambda n. \$(I\{l\}a'')[i]\{m\}-n)$
abbreviation *perp-due-incr-lvl* :: *real* \Rightarrow *nat* \Rightarrow *real* $(\$(Ia''''')[-]\{-}\{-}'-\infty [0,0]$
 $200)$
where $\$(Ia'')[i]\{m\}-\infty \equiv \$(I\{1\}a'')[i]\{m\}-\infty$
abbreviation *perp-due-incr-yr* :: *real* \Rightarrow *real* $(\$(Ia''''')[-]'-\infty [0] 200)$
where $\$(Ia'')[i]-\infty \equiv \$(Ia'')[i]\{1\}-\infty$

3 Theory of Interest

locale *interest* =

fixes $i :: real$ — i stands for an interest rate.

assumes *v-futr-pos*: $1 + i > 0$ — Assume that the future value is positive.

context *interest*

begin

abbreviation $i\text{-nom}' :: \text{nat} \Rightarrow \text{real} (\$i\{-\} [0] 200)$
 where $\$i\{m\} \equiv \$i[i]\{m\}$
abbreviation $i\text{-force}' :: \text{real} (\$\delta)$
 where $\$\delta \equiv \$\delta[i]$
abbreviation $d\text{-nom}' :: \text{nat} \Rightarrow \text{real} (\$d\{-\} [0] 200)$
 where $\$d\{m\} \equiv \$d[i]\{m\}$
abbreviation $d\text{-nom-yr}' :: \text{real} (\$d)$
 where $\$d \equiv \$d[i]$
abbreviation $v\text{-pres}' :: \text{real} (\$v)$
 where $\$v \equiv \$v[i]$
abbreviation $ann' :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real} (\$a\{-\}'-- [0,101] 200)$
 where $\$a\{m\}\text{-}n \equiv \$a[i]\{m\}\text{-}n$
abbreviation $ann\text{-yr}' :: \text{nat} \Rightarrow \text{real} (\$a'\text{-} [101] 200)$
 where $\$a\text{-}n \equiv \$a[i]\text{-}n$
abbreviation $acc' :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real} (\$s\{-\}'-- [0,101] 200)$
 where $\$s\{m\}\text{-}n \equiv \$s[i]\{m\}\text{-}n$
abbreviation $acc\text{-yr}' :: \text{nat} \Rightarrow \text{real} (\$s'\text{-} [101] 200)$
 where $\$s\text{-}n \equiv \$s[i]\text{-}n$
abbreviation $ann\text{-due}' :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real} (\$a'''\{-\}'-- [0,101] 200)$
 where $\$a''\{m\}\text{-}n \equiv \$a''[i]\{m\}\text{-}n$
abbreviation $ann\text{-due-yr}' :: \text{nat} \Rightarrow \text{real} (\$a'''\text{-} [101] 200)$
 where $\$a''\text{-}n \equiv \$a''[i]\text{-}n$
abbreviation $acc\text{-due}' :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real} (\$s'''\{-\}'-- [0,101] 200)$
 where $\$s''\{m\}\text{-}n \equiv \$s''[i]\{m\}\text{-}n$
abbreviation $acc\text{-due-yr}' :: \text{nat} \Rightarrow \text{real} (\$s'''\text{-} [101] 200)$
 where $\$s''\text{-}n \equiv \$s''[i]\text{-}n$
abbreviation $ann\text{-cont}' :: \text{real} \Rightarrow \text{real} (\$a'''\text{-} [101] 200)$
 where $\$a'\text{-}n \equiv \$a'[i]\text{-}n$
abbreviation $acc\text{-cont}' :: \text{real} \Rightarrow \text{real} (\$s'''\text{-} [101] 200)$
 where $\$s'\text{-}n \equiv \$s'[i]\text{-}n$
abbreviation $perp' :: \text{nat} \Rightarrow \text{real} (\$a\{-\}'\text{-}\infty [0] 200)$
 where $\$a\{m\}\text{-}\infty \equiv \$a[i]\{m\}\text{-}\infty$
abbreviation $perp\text{-yr}' :: \text{real} (\$a'\text{-}\infty)$
 where $\$a\text{-}\infty \equiv \$a[i]\text{-}\infty$
abbreviation $perp\text{-due}' :: \text{nat} \Rightarrow \text{real} (\$a'''\{-\}'\text{-}\infty [0] 200)$
 where $\$a''\{m\}\text{-}\infty \equiv \$a''[i]\{m\}\text{-}\infty$
abbreviation $perp\text{-due-yr}' :: \text{real} (\$a'''\text{-}\infty)$
 where $\$a''\text{-}\infty \equiv \$a''[i]\text{-}\infty$
abbreviation $ann\text{-incr}' :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real} (\$(I\{-\}a')\{-\}'-- [0,0,101] 200)$
 where $\$(I\{l\}a)\{m\}\text{-}n \equiv \$(I\{l\}a)[i]\{m\}\text{-}n$
abbreviation $ann\text{-incr-lvl}' :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real} (\$(Ia')\{-\}'-- [0,101] 200)$
 where $\$(Ia)\{m\}\text{-}n \equiv \$(Ia)[i]\{m\}\text{-}n$
abbreviation $ann\text{-incr-yr}' :: \text{nat} \Rightarrow \text{real} (\$(Ia')'\text{-} [101] 200)$
 where $\$(Ia)\text{-}n \equiv \$(Ia)[i]\text{-}n$
abbreviation $acc\text{-incr}' :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real} (\$(I\{-\}s')\{-\}'-- [0,0,101] 200)$

where $\$(I\{l\}s)\{m\}-n \equiv \$(I\{l\}s)[i]\{m\}-n$
abbreviation $acc-incr-lvl' :: nat \Rightarrow nat \Rightarrow real (\$(Is')\{-}\{-- [0,101] 200)$
where $\$(Is)\{m\}-n \equiv \$(Is)[i]\{m\}-n$
abbreviation $acc-incr-yr' :: nat \Rightarrow real (\$(Is')\{-- [101] 200)$
where $\$(Is)-n \equiv \$(Is)[i]-n$
abbreviation $ann-due-incr' :: nat \Rightarrow nat \Rightarrow nat \Rightarrow real (\$(I\{-}a''''')\{-}\{-- [0,0,101] 200)$
where $\$(I\{l\}a')\{m\}-n \equiv \$(I\{l\}a')[i]\{m\}-n$
abbreviation $ann-due-incr-lvl' :: nat \Rightarrow nat \Rightarrow real (\$(Ia''''')\{-}\{-- [0,101] 200)$
where $\$(Ia'')\{m\}-n \equiv \$(Ia'')[i]\{m\}-n$
abbreviation $ann-due-incr-yr' :: nat \Rightarrow real (\$(Ia''''')\{-- [101] 200)$
where $\$(Ia'')-n \equiv \$(Ia'')[i]-n$
abbreviation $acc-due-incr' :: nat \Rightarrow nat \Rightarrow nat \Rightarrow real (\$(I\{-}s''''')\{-}\{-- [0,0,101] 200)$
where $\$(I\{l\}s'')\{m\}-n \equiv \$(I\{l\}s'')[i]\{m\}-n$
abbreviation $acc-due-incr-lvl' :: nat \Rightarrow nat \Rightarrow real (\$(Is''''')\{-}\{-- [0,101] 200)$
where $\$(Is'')\{m\}-n \equiv \$(Is'')[i]\{m\}-n$
abbreviation $acc-due-incr-yr' :: nat \Rightarrow real (\$(Is''''')\{-- [101] 200)$
where $\$(Is'')-n \equiv \$(Is'')[i]-n$
abbreviation $perp-incr' :: nat \Rightarrow nat \Rightarrow real (\$(I\{-}a')\{-}\{--\infty [0,0] 200)$
where $\$(I\{l\}a)\{m\}-\infty \equiv \$(I\{l\}a)[i]\{m\}-\infty$
abbreviation $perp-incr-lvl' :: nat \Rightarrow real (\$(Ia')\{-}\{--\infty [0] 200)$
where $\$(Ia)\{m\}-\infty \equiv \$(Ia)[i]\{m\}-\infty$
abbreviation $perp-incr-yr' :: real (\$(Ia')\{--\infty)$
where $\$(Ia)-\infty \equiv \$(Ia)[i]-\infty$
abbreviation $perp-due-incr' :: nat \Rightarrow nat \Rightarrow real (\$(I\{-}a''''')\{-}\{--\infty [0,0] 200)$
where $\$(I\{l\}a'')\{m\}-\infty \equiv \$(I\{l\}a'')[i]\{m\}-\infty$
abbreviation $perp-due-incr-lvl' :: nat \Rightarrow real (\$(Ia''''')\{-}\{--\infty [0] 200)$
where $\$(Ia'')\{m\}-\infty \equiv \$(Ia'')[i]\{m\}-\infty$
abbreviation $perp-due-incr-yr' :: real (\$(Ia''''')\{--\infty)$
where $\$(Ia'')-\infty \equiv \$(Ia'')[i]-\infty$

lemma $v-futr-m-pos: 1 + \$(i)\{m\}/m > 0$ **if** $m \neq 0$ **for** $m::nat$
using $v-futr-pos$ $i-nom-def$ **by force**

lemma $i-nom-1[simp]: \$(i)\{1\} = i$
using $v-futr-pos$ $i-nom-def$ **by force**

lemma $i-nom-eff: (1 + \$(i)\{m\}/m)^m = 1 + i$ **if** $m \neq 0$ **for** $m::nat$
unfolding $i-nom-def$ **using** $less-imp-neq$ $v-futr-pos$ **that**
apply ($simp$, $subst$ $powr-realpow[THEN sym]$, $simp$)
by ($subst$ $powr-powr$, $simp$)

lemma $i-nom-i: 1 + \$(i)\{m\}/m = (1+i).\{1/m\}$ **if** $m \neq 0$ **for** $m::nat$
unfolding $i-nom-def$ **by** ($simp$ $add: that$)

lemma $i-nom-0-iff-i-0: \$(i)\{m\} = 0 \iff i = 0$ **if** $m \neq 0$ **for** $m::nat$
proof

```

assume  $i^{\wedge}\{m\} = 0$ 
hence  $\star: (1+i).\wedge(1/m) = (1+i).\wedge 0$ 
  unfolding i-nom-def using v-futr-pos that by simp
show  $i = 0$ 
proof (rule ccontr)
  assume  $i \neq 0$ 
  hence  $1/m = 0$  using powr-inj  $\star$  v-futr-pos by smt
  thus False using that by simp
qed
next
assume  $i = 0$ 
thus  $i^{\wedge}\{m\} = 0$ 
  unfolding i-nom-def by simp
qed

lemma i-nom-pos-iff-i-pos:  $i^{\wedge}\{m\} > 0 \iff i > 0$  if  $m \neq 0$  for  $m::\text{nat}$ 
proof
  assume  $i^{\wedge}\{m\} > 0$ 
  hence  $\star: (1+i).\wedge(1/m) > 1.\wedge(1/m)$ 
    unfolding i-nom-def using v-futr-pos that by (simp add: zero-less-mult-iff)
  thus  $i > 0$ 
    using powr-less-cancel2[of  $1/m$   $1$   $1+i$ ] v-futr-pos that by simp
next
  assume  $i > 0$ 
  hence  $(1+i).\wedge(1/m) > 1.\wedge(1/m)$ 
    using powr-less-mono2 v-futr-pos that by simp
  thus  $i^{\wedge}\{m\} > 0$ 
    unfolding i-nom-def using that by (simp add: zero-less-mult-iff)
qed

lemma e-delta:  $\text{exp } \$\delta = 1 + i$ 
  unfolding i-force-def by (simp add: v-futr-pos)

lemma delta-0-iff-i-0:  $\delta = 0 \iff i = 0$ 
proof
  assume  $\delta = 0$ 
  thus  $i = 0$ 
    using e-delta by auto
next
  assume  $i = 0$ 
  thus  $\delta = 0$ 
    unfolding i-force-def by simp
qed

lemma lim-i-nom:  $(\lambda m. i^{\wedge}\{m\}) \longrightarrow \delta$ 
proof -
  let  $?f = \lambda h. ((1+i).\wedge h - 1) / h$ 
  have D1ipwr: DERIV  $(\lambda h. (1+i).\wedge h) 0 :=> \delta$ 
    unfolding i-force-def

```

using *has-real-derivative-powr2*[*OF v-futr-pos*, **where** $x=0$] *v-futr-pos* **by** *simp*
hence $\lim f: (?f \longrightarrow \mathbb{R})$ (*at 0*)
unfolding *DERIV-def* **using** *v-futr-pos* **by** *auto*
hence $(\lambda m. \mathbb{R}^{\wedge}\{Suc\ m\}) \longrightarrow \mathbb{R}$
unfolding *i-nom-def* **using** *tendsto-at-iff-sequentially*[*of ?f* \mathbb{R} 0 \mathbb{R} , *THEN iffD1*]
apply *simp*
apply (*drule-tac* $x=\lambda m. 1 / Suc\ m$ **in** *spec*, *simp*, *drule* *mp*)
subgoal **using** *lim-1-over-n LIMSEQ-Suc* **by** *force*
by (*simp* *add: o-def mult.commute*)
thus *?thesis*
by (*simp* *add: LIMSEQ-imp-Suc*)
qed

lemma *d-nom-0-iff-i-0*: $\mathbb{R}^{\wedge}\{m\} = 0 \iff i = 0$ **if** $m \neq 0$ **for** $m::nat$
proof –
have $\mathbb{R}^{\wedge}\{m\} = 0 \iff \mathbb{R}^{\wedge}\{m\} = 0$
unfolding *d-nom-def* **using** *v-futr-m-pos* **by** (*smt* (*verit*) *divide-eq-0-iff of-nat-0*)
thus *?thesis*
using *i-nom-0-iff-i-0* **that** **by** *auto*
qed

lemma *d-nom-pos-iff-i-pos*: $\mathbb{R}^{\wedge}\{m\} > 0 \iff i > 0$ **if** $m \neq 0$ **for** $m::nat$
proof –
have $\mathbb{R}^{\wedge}\{m\} > 0 \iff \mathbb{R}^{\wedge}\{m\} > 0$
unfolding *d-nom-def* **using** *zero-less-divide-iff i-nom-pos-iff-i-pos v-futr-m-pos*
that **by** *smt*
thus *?thesis*
using *i-nom-pos-iff-i-pos* **that** **by** *auto*
qed

lemma *d-nom-i-nom*: $1 - \mathbb{R}^{\wedge}\{m\}/m = 1 / (1 + \mathbb{R}^{\wedge}\{m\}/m)$ **if** $m \neq 0$ **for** $m::nat$
proof –
have $1 - \mathbb{R}^{\wedge}\{m\}/m = 1 - (\mathbb{R}^{\wedge}\{m\}/m) / (1 + \mathbb{R}^{\wedge}\{m\}/m)$
by (*simp* *add: d-nom-def*)
also **have** $\dots = 1 / (1 + \mathbb{R}^{\wedge}\{m\}/m)$
using *v-futr-m-pos*
by (*smt* (*verit*, *ccfv-SIG*) *add-divide-distrib* **that** *div-self*)
finally **show** *?thesis* .
qed

lemma *lim-d-nom*: $(\lambda m. \mathbb{R}^{\wedge}\{m\}) \longrightarrow \mathbb{R}$
proof –
have $(\lambda m. \mathbb{R}^{\wedge}\{m\}/m) \longrightarrow 0$
using *lim-i-nom tendsto-divide-0 tendsto-of-nat* **by** *blast*
hence $(\lambda m. 1 + \mathbb{R}^{\wedge}\{m\}/m) \longrightarrow 1$
by (*metis* *add.right-neutral tendsto-add-const-iff*)
thus *?thesis*
unfolding *d-nom-def* **using** *lim-i-nom tendsto-divide div-by-1* **by** *fastforce*

qed

lemma *v-pos*: $\$v > 0$
unfolding *v-pres-def* using *v-futr-pos* by *auto*

lemma *v-1-iff-i-0*: $\$v = 1 \longleftrightarrow i = 0$

proof

assume $\$v = 1$

thus $i = 0$

unfolding *v-pres-def* by *simp*

next

assume $i = 0$

thus $\$v = 1$

unfolding *v-pres-def* by *simp*

qed

lemma *v-lt-1-iff-i-pos*: $\$v < 1 \longleftrightarrow i > 0$

proof

assume $\$v < 1$

thus $i > 0$

unfolding *v-pres-def* by (*simp add: v-futr-pos*)

next

assume $i > 0$

thus $\$v < 1$

unfolding *v-pres-def* by (*simp add: v-futr-pos*)

qed

lemma *v-i-nom*: $\$v = (1 + \$i^{\wedge}\{m\}/m)^{\wedge}-m$ if $m \neq 0$ for $m::nat$

proof -

have $\$v = (1 + i)^{\wedge}-1$

unfolding *v-pres-def* using *v-futr-pos* *powr-real-def* that by (*simp add: powr-neg-one*)

also have $\dots = ((1 + \$i^{\wedge}\{m\}/m)^{\wedge}m)^{\wedge}-1$

using *i-nom-eff* that by *presburger*

also have $\dots = (1 + \$i^{\wedge}\{m\}/m)^{\wedge}-m$

using *powr-powr* *powr-realpow* [THEN *sym*] *v-futr-m-pos* that by *simp*

finally show ?thesis .

qed

lemma *i-v*: $1 + i = \$v^{\wedge}-1$

unfolding *v-pres-def* *powr-real-def* using *v-futr-pos* *powr-neg-one* by *simp*

lemma *i-v-powr*: $(1 + i)^{\wedge}a = \$v^{\wedge}-a$ for $a::real$

by (*subst i-v*, *subst powr-powr*, *simp*)

lemma *v-delta*: $\ln \$v = - \δ

unfolding *i-force-def* *v-pres-def* using *v-futr-pos* by (*simp add: ln-div*)

lemma *is-derive-vpow*: *DERIV* ($\lambda t. \$v^{\wedge}t$) $t := - \$\delta * \$v^{\wedge}t$

using *v-delta* *has-real-derivative-powr2* *v-pos* by (*metis mult.commute*)

lemma *d-nom-v*: $\$d\{m\} = m * (1 - \$v.\wedge(1/m))$ **if** $m \neq 0$ **for** $m::nat$

proof –

have $\$d\{m\} = m * (1 - 1 / (1 + \$i\{m\}/m))$

using *d-nom-i-nom*[*THEN sym*] **that** **by** *force*

also have $\dots = m * (1 - 1 / (1 + i).\wedge(1/m))$

using *i-nom-i* **that** *powr-minus-divide* **by** *simp*

also have $\dots = m * (1 - \$v.\wedge(1/m))$

using *v-pres-def* *v-futr-pos* *powr-divide* **by** *simp*

finally show *?thesis* .

qed

lemma *d-nom-i-nom-v*: $\$d\{m\} = \$i\{m\} * \$v.\wedge(1/m)$ **if** $m \neq 0$ **for** $m::nat$

unfolding *d-nom-def* *v-pres-def* **using** *i-nom-i* *powr-divide* *v-futr-pos* **that** **by** *auto*

lemma *a-calc*: $\$a\{m\}-n = (1 - \$v\wedge n) / \$i\{m\}$ **if** $m \neq 0$ $i \neq 0$ **for** $n m :: nat$

proof –

have $\wedge l::nat. l/m = (1/m) * l$ **by** *simp*

hence $\star: \wedge l::nat. \$v.\wedge(l/m) = (\$v.\wedge(1/m))\wedge l$

using *powr-powr* *powr-realpow* *v-pos* **by** (*metis powr-gt-zero*)

hence $\$a\{m\}-n = (\sum k < n * m. (\$v.\wedge(1/m))\wedge(k+1::nat)) / m$

unfolding *ann-def* **by** *presburger*

also have $\dots = \$v.\wedge(1/m) * (\sum k < n * m. (\$v.\wedge(1/m))\wedge k) / m$

by (*simp*, *subst sum-divide-distrib*[*THEN sym*], *subst sum-distrib-left*[*THEN sym*], *simp*)

also have $\dots = \$v.\wedge(1/m) * (((\$v.\wedge(1/m))\wedge(n * m) - 1) / (\$v.\wedge(1/m) - 1)) / m$

apply (*subst geometric-sum*[*of* $\$v.\wedge(1/m)$ $n * m$]; *simp?*)

using *powr-zero-eq-one*[*of* $\$v$] *v-pos* *v-1-iff-i-0* *powr-inj* **that**

by (*smt* (*verit*, *del-insts*) *divide-eq-0-iff* *of-nat-eq-0-iff*)

also have $\dots = ((\$v.\wedge(1/m))\wedge(n * m) - 1) / (m * (\$v.\wedge(1/m) - 1) / \$v.\wedge(1/m))$

by (*simp add: field-simps*)

also have $\dots = (\$v\wedge n - 1) / (m * (1 - 1 / \$v.\wedge(1/m)))$

apply (*subst* \star [*of* $n * m::nat$, *THEN sym*], *simp only: of-nat-simps*)

apply (*subst nonzero-mult-div-cancel-right*[**where** $'a = \text{real}$, *of* m n], *simp add: that*)

apply (*subst powr-realpow*[*OF* *v-pos*])

apply (*subst times-divide-eq-right*[*of* $-$ $\$v.\wedge(1/m)$, *THEN sym*])

using *v-pos* **by** (*subst diff-divide-distrib*[*of* $-$ $\$v.\wedge(1/m)$], *simp*)

also have $\dots = (1 - \$v\wedge n) / (m * (1 / \$v.\wedge(1/m) - 1))$

using *minus-divide-divide* **by** (*smt mult-minus-right*)

also have $\dots = (1 - \$v\wedge n) / \$i\{m\}$

unfolding *i-nom-def* *v-pres-def* **using** *v-futr-pos* *powr-divide* **by** *auto*

finally show *?thesis* .

qed

lemma *a-calc-i-0*: $\$a\{m\}-n = n$ **if** $m \neq 0$ $i = 0$ **for** $n m :: nat$

unfolding *ann-def* *v-pres-def* **using** *that* **by** *simp*

lemma *s-calc-i-0*: $\$s\{\!m\}-n = n$ if $m \neq 0$ $i = 0$ for $n\ m :: \text{nat}$
unfolding *acc-def* **using** *that* **by** *simp*

lemma *s-a*: $\$s\{\!m\}-n = (1+i)\hat{n} * \$a\{\!m\}-n$ if $m \neq 0$ for $n\ m :: \text{nat}$

proof –

have $(1+i)\hat{n} * \$a\{\!m\}-n = (\sum k < n*m. (1+i)\hat{n} * (\$v.\hat{((k+1::\text{nat})/m)} / m))$

unfolding *ann-def* **using** *sum-distrib-left* **by** *blast*

also have $\dots = (\sum k < n*m. (1+i).\hat{((n*m - \text{Suc } k)/m)} / m)$

proof –

have $\bigwedge k :: \text{nat}. k < n*m \implies (1+i)\hat{n} * (\$v.\hat{((k+1::\text{nat})/m)} / m) = (1+i).\hat{((n*m - \text{Suc } k)/m)} / m$

unfolding *v-pres-def*

apply (*subst powr-realpow*[*THEN sym*], *simp add: v-futr-pos*)

apply (*subst inverse-powr*, *simp add: v-futr-pos*)

apply (*subst times-divide-eq-right*, *subst powr-add*[*THEN sym*], *simp add:*

that)

by (*subst of-nat-diff*, *simp add: Suc-le-eq*, *simp add: diff-divide-distrib that*)

thus *?thesis* **by** (*meson lessThan-iff sum.cong*)

qed

also have $\dots = (\sum k < n*m. (1+i).\hat{(k/m)} / m)$

apply (*subst atLeast0LessThan*[*THEN sym*])+

by (*subst sum.atLeastLessThan-rev*[*THEN sym*, *of - n*m 0*, *simplified add-0-right*], *simp*)

also have $\dots = \$s\{\!m\}-n$

unfolding *acc-def* **by** *simp*

finally show *?thesis* ..

qed

lemma *s-calc*: $\$s\{\!m\}-n = ((1+i)\hat{n} - 1) / \$i\{\!m\}$ if $m \neq 0$ $i \neq 0$ for $n\ m :: \text{nat}$

using *that v-futr-pos*

apply (*subst s-a*, *simp*, *subst a-calc*; *simp?*)

apply (*rule disjI2*)

apply (*subst right-diff-distrib*, *simp*)

apply (*rule left-right-inverse-power*)

unfolding *v-pres-def* **by** *auto*

lemma *a''-a*: $\$a''\{\!m\}-n = (1+i).\hat{(1/m)} * \$a\{\!m\}-n$ if $m \neq 0$ for $m :: \text{nat}$

unfolding *ann-def ann-due-def*

apply (*subst sum-distrib-left*, *subst times-divide-eq-right*, *simp*)

by (*subst i-v*, *subst powr-powr*, *subst powr-add*[*THEN sym*], *simp*, *subst add-divide-distrib*, *simp*)

lemma *a-a''*: $\$a\{\!m\}-n = \$v.\hat{(1/m)} * \$a''\{\!m\}-n$ if $m \neq 0$ for $m :: \text{nat}$

unfolding *ann-def ann-due-def*

apply (*subst sum-distrib-left*, *subst times-divide-eq-right*, *simp*)

by (*subst powr-add*[*THEN sym*], *subst add-divide-distrib*, *simp*)

lemma $a''\text{-calc-i-0}$: $\$a''\{m\}\text{-}n = n$ **if** $m \neq 0$ $i = 0$ **for** $n\ m :: \text{nat}$
unfolding $\text{ann-due-def v-pres-def}$ **using** that by simp

lemma $s''\text{-calc-i-0}$: $\$s''\{m\}\text{-}n = n$ **if** $m \neq 0$ $i = 0$ **for** $n\ m :: \text{nat}$
unfolding acc-due-def **using** that by simp

lemma $a''\text{-calc}$: $\$a''\{m\}\text{-}n = (1 - \$v\hat{n}) / \$d\{m\}$ **if** $m \neq 0$ $i \neq 0$ **for** $n\ m :: \text{nat}$

proof –

have $\$a''\{m\}\text{-}n = (1+i).\hat{(1/m)} * ((1 - \$v\hat{n}) / \$i\{m\})$

using $a''\text{-a a-calc times-divide-eq-right}$ **that by simp**

also have $\dots = (1 - \$v\hat{n}) / (\$v.\hat{(1/m)} * \$i\{m\})$

by $(\text{subst i-v, subst powr-powr, simp, subst powr-minus-divide, simp})$

also have $\dots = (1 - \$v\hat{n}) / \$d\{m\}$

using $d\text{-nom-i-nom-v}$ **that by simp**

finally show $?thesis$.

qed

lemma $s''\text{-s}$: $\$s''\{m\}\text{-}n = (1+i).\hat{(1/m)} * \$s\{m\}\text{-}n$ **if** $m \neq 0$ **for** $m :: \text{nat}$
unfolding $\text{acc-def acc-due-def}$
by $(\text{simp add: sum-distrib-left add-divide-distrib powr-add})$

lemma $s\text{-}s''$: $\$s\{m\}\text{-}n = \$v.\hat{(1/m)} * \$s''\{m\}\text{-}n$ **if** $m \neq 0$ **for** $m :: \text{nat}$
unfolding $\text{acc-def acc-due-def v-pres-def}$ **using** v-futr-pos
apply $(\text{simp add: sum-distrib-left inverse-powr add-divide-distrib})$
by $(\text{metis (no-types) add-diff-cancel-left' powr-add uminus-add-conv-diff})$

lemma $s''\text{-calc}$: $\$s''\{m\}\text{-}n = ((1+i)\hat{n} - 1) / \$d\{m\}$ **if** $m \neq 0$ $i \neq 0$ **for** $n\ m :: \text{nat}$

proof –

have $\$s''\{m\}\text{-}n = (1+i).\hat{(1/m)} * ((1+i)\hat{n} - 1) / \$i\{m\}$

using $s''\text{-s s-calc times-divide-eq-right}$ **that by simp**

also have $\dots = ((1+i)\hat{n} - 1) / (\$v.\hat{(1/m)} * \$i\{m\})$

by $(\text{subst i-v, subst powr-powr, simp, subst powr-minus-divide, simp})$

also have $\dots = ((1+i)\hat{n} - 1) / \$d\{m\}$

using $d\text{-nom-i-nom-v}$ **that by simp**

finally show $?thesis$.

qed

lemma $s''\text{-}a''$: $\$s''\{m\}\text{-}n = (1+i)\hat{n} * \$a''\{m\}\text{-}n$ **if** $m \neq 0$ **for** $m :: \text{nat}$
using $\text{that s''-s a''-a s-a}$ **by simp**

lemma $a'\text{-calc}$: $\$a'\text{-}n = (1 - \$v.\hat{n}) / \$\delta$ **if** $i \neq 0$ $n \geq 0$ **for** $n :: \text{real}$
unfolding ann-cont-def
apply $(\text{rule integral-unique})$
using $\text{has-integral-powr2-from-0[OF v-pos - that(2)] v-delta v-1-iff-i-0}$ **that**
by $(\text{smt minus-divide-divide})$

lemma $a'\text{-calc-i-0}$: $\$a'\text{-}n = n$ **if** $i = 0$ $n \geq 0$ **for** $n :: \text{real}$

```

unfolding ann-cont-def
apply (subst iffD2[OF v-1-iff-i-0], simp add: that)
by (simp add: integral-cong that)

lemma s'-calc:  $\$s'-n = ((1+i).\widehat{n} - 1) / \$\delta$  if  $i \neq 0$   $n \geq 0$  for  $n::real$ 
unfolding acc-cont-def
apply (rule integral-unique)
using has-integral-powr2-from-0[OF v-futr-pos - that(2)] i-force-def that
by simp

lemma s'-calc-i-0:  $\$s'-n = n$  if  $i = 0$   $n \geq 0$  for  $n::real$ 
unfolding acc-cont-def
apply (subst ⟨i = 0⟩, simp)
by (simp add: integral-cong that)

lemma s'-a':  $\$s'-n = (1+i).\widehat{n} * \$a'-n$  if  $n \geq 0$  for  $n::real$ 
proof –
have  $(1+i).\widehat{n} * \$a'-n = \text{integral } \{0..n\} (\lambda t. (1+i).\widehat{(n-t)})$ 
unfolding ann-cont-def
using integrable-on-powr2-from-0-general[of $v n] v-pos v-futr-pos that
apply (subst integral-mult, simp)
apply (rule integral-cong)
unfolding v-pres-def using inverse-powr powr-add[THEN sym] by smt
also have  $\dots = \$s'-n$ 
unfolding acc-cont-def using v-futr-pos that
apply (subst has-integral-interval-reverse[of 0 n, simplified, THEN integral-unique];
simp?)
by (rule continuous-on-powr; auto)
finally show ?thesis ..
qed

lemma lim-m-a:  $(\lambda m. \$a^{\widehat{\{m\}}-n}) \longrightarrow \$a'-n$  for  $n::nat$ 
proof (rule LIMSEQ-imp-Suc)
show  $(\lambda m. \$a^{\widehat{\{Suc\ m\}}-n}) \longrightarrow \$a'-n$ 
proof (cases i = 0)
case True
show ?thesis
using a-calc-i-0 a'-calc-i-0 True by simp
next
case False
show ?thesis
using False v-pos delta-0-iff-i-0
apply (subst a-calc; simp?)
apply (subst a'-calc; simp?)
apply (subst powr-realpow, simp)
apply (rule tendsto-divide; simp?)
by (rule LIMSEQ-Suc[OF lim-i-nom])
qed
qed

```



```

lemma lim-m-a'': ( $\lambda m. \$a''\{m\}-n$ )  $\longrightarrow$   $\$a'-n$  for  $n::nat$ 
proof (rule LIMSEQ-imp-Suc)
  show ( $\lambda m. \$a''\{Suc\ m\}-n$ )  $\longrightarrow$   $\$a'-n$ 
  proof (cases i = 0)
    case True
      show ?thesis
      using a''-calc-i-0 a'-calc-i-0 True by simp
    next
      case False
      show ?thesis
      using False v-pos delta-0-iff-i-0
      apply (subst a''-calc; simp?)
      apply (subst a'-calc; simp?)
      apply (subst powr-realpow, simp)
      apply (rule tendsto-divide; simp?)
      by (rule LIMSEQ-Suc[OF lim-d-nom])
  qed
qed

```

```

lemma lim-m-s: ( $\lambda m. \$s\{m\}-n$ )  $\longrightarrow$   $\$s'-n$  for  $n::nat$ 
proof (rule LIMSEQ-imp-Suc)
  show ( $\lambda m. \$s\{Suc\ m\}-n$ )  $\longrightarrow$   $\$s'-n$ 
  proof (cases i = 0)
    case True
      show ?thesis
      using s-calc-i-0 s'-calc-i-0 True by simp
    next
      case False
      show ?thesis
      using False v-futr-pos delta-0-iff-i-0
      apply (subst s-calc; simp?)
      apply (subst s'-calc; simp?)
      apply (subst powr-realpow, simp)
      apply (rule tendsto-divide; simp?)
      by (rule LIMSEQ-Suc[OF lim-i-nom])
  qed
qed

```

```

lemma lim-m-s'': ( $\lambda m. \$s''\{m\}-n$ )  $\longrightarrow$   $\$s'-n$  for  $n::nat$ 
proof (rule LIMSEQ-imp-Suc)
  show ( $\lambda m. \$s''\{Suc\ m\}-n$ )  $\longrightarrow$   $\$s'-n$ 
  proof (cases i = 0)
    case True
      show ?thesis
      using s''-calc-i-0 s'-calc-i-0 True by simp
    next
      case False
      show ?thesis

```

using *False v-futr-pos delta-0-iff-i-0*
apply (*subst s''-calc; simp?*)
apply (*subst s'-calc; simp?*)
apply (*subst powr-realpow, simp*)
apply (*rule tendsto-divide; simp?*)
by (*rule LIMSEQ-Suc[OF lim-d-nom]*)
qed
qed

lemma *lim-n-a*: $(\lambda n. \$a^{\wedge}\{m\}\text{-}n) \longrightarrow \$a^{\wedge}\{m\}\text{-}\infty$ **if** $m \neq 0$ **i** > 0 **for** $m::\text{nat}$
proof –
have $\$i^{\wedge}\{m\} \neq 0$ **using** *i-nom-pos-iff-i-pos that by smt*
moreover have $(\lambda n. \$v^{\wedge}n) \longrightarrow 0$
using *LIMSEQ-realpow-zero[of \$v] v-pos v-lt-1-iff-i-pos that by simp*
ultimately show *?thesis*
using that apply (*subst a-calc; simp?*)
unfolding perp-def apply (*rule tendsto-divide; simp?*)
using tendsto-diff[where a=1 and b=0] by auto
qed

lemma *lim-n-a''*: $(\lambda n. \$a''^{\wedge}\{m\}\text{-}n) \longrightarrow \$a''^{\wedge}\{m\}\text{-}\infty$ **if** $m \neq 0$ **i** > 0 **for** $m::\text{nat}$
proof –
have $\$d^{\wedge}\{m\} \neq 0$ **using** *d-nom-pos-iff-i-pos that by smt*
moreover have $(\lambda n. \$v^{\wedge}n) \longrightarrow 0$
using *LIMSEQ-realpow-zero[of \$v] v-pos v-lt-1-iff-i-pos that by simp*
ultimately show *?thesis*
using that apply (*subst a''-calc; simp?*)
unfolding perp-due-def apply (*rule tendsto-divide; simp?*)
using tendsto-diff[where a=1 and b=0] by auto
qed

lemma *lsm-lam*: $\$(I^{\wedge}\{l\}s)^{\wedge}\{m\}\text{-}n = (1+i)^{\wedge}n * \$(I^{\wedge}\{l\}a)^{\wedge}\{m\}\text{-}n$
if $l \neq 0$ $m \neq 0$ **for** $l\ n\ m :: \text{nat}$
unfolding acc-incr-def ann-incr-def v-pres-def using v-futr-pos powr-realpow
apply (*subst inverse-powr, simp*)
apply (*subst sum-distrib-left*)
by (*subst minus-real-def, subst powr-add, subst times-divide-eq-right, subst mult.assoc, simp*)

lemma *Iam-calc*: $\$(Ia)^{\wedge}\{m\}\text{-}n = (\sum j < n. (j+1)/m * (\sum k=j*m..<(j+1)*m. \$v.^{\wedge}((k+1)/m)))$
if $m \neq 0$ **for** $n\ m :: \text{nat}$
proof –
let $?I = \{..<n\}$
let $?A = \lambda j. \{j*m..<(j+1)*m\}$
let $?g = \lambda k. \$v.^{\wedge}((k+1::\text{nat})/m) * \lceil(k+1::\text{nat})/m\rceil / m$
have $\$(Ia)^{\wedge}\{m\}\text{-}n = (\sum j < n. \sum k=j*m..<(j+1)*m. \$v.^{\wedge}((k+1)/m) * \lceil(k+1)/m\rceil / m)$
unfolding ann-incr-def using seq-part-multiple that
apply (*simp only: mult-1*)

by (*subst sum.UNION-disjoint*[of ?I ?A ?g, THEN *sym*]; *simp*)
 also have ... = $(\sum j < n. (j+1)/m * (\sum k=j*m..<(j+1)*m. \$v.\wedge((k+1)/m)))$
 proof –
 { **fix** *j k*
 assume $j*m \leq k \wedge k < (j+1)*m$
 hence $j*m < k+1 \wedge k+1 \leq (j+1)*m$ **by** *force*
 hence $j < (k+1)/m \wedge (k+1)/m \leq j+1$
 using *pos-less-divide-eq pos-divide-le-eq of-nat-less-iff of-nat-le-iff* that
 by (*smt (verit) of-nat-le-0-iff of-nat-mult*)
 hence $\lceil (k+1)/m \rceil = j+1$
 by (*simp add: ceiling-unique*) }
hence $\bigwedge j k. j*m \leq k \wedge k < (j+1)*m \implies \lceil (k+1)/m \rceil = j+1$
by (*metis (no-types) of-nat-1 of-nat-add*)
with *v-pos* **show** ?thesis
 apply (*intro sum.cong, simp*)
 apply (*subst sum-distrib-left, rule sum.cong; simp*)
 by (*smt (verit, ccfv-SIG) of-int-1 of-int-diff of-int-of-nat-eq*)
qed
finally show ?thesis .
qed

lemma *Ism-calc*: $\$(I\wedge\{m\})-n = (\sum j < n. (j+1)/m * (\sum k=j*m..<(j+1)*m. (1+i).\wedge(n-(k+1)/m)))$
if $m \neq 0$ **for** $n m :: nat$
using *v-pos* that
apply (*subst Iism-Ilam; simp*)
apply (*subst Iam-calc[simplified]; simp?*)
apply ((*subst sum-distrib-left, rule sum.cong; simp*))+
unfolding *v-pres-def* **using** *v-futr-pos*
apply (*subst inverse-powr; simp*)
apply (*subst powr-realpow[THEN sym], simp*)
by (*subst powr-add[THEN sym]; simp*)

lemma *Imam-calc-aux*: $\$(I\wedge\{m\}a)\wedge\{m\}-n = (\sum k < n*m. \$v.\wedge((k+1)/m) * (k+1) / m^2)$
if $m \neq 0$ **for** $m::nat$
unfolding *ann-incr-def power-def*
apply (*rule sum.cong, simp*)
apply (*subst of-nat-mult*)
using *v-pos* that
apply (*subst nonzero-mult-div-cancel-left, simp*)
by (*subst ceiling-of-nat; simp*)

lemma *Imam-calc*:
 $\$(I\wedge\{m\}a)\wedge\{m\}-n = (\$v.\wedge(1/m) * (1 - (n*m+1)*\$v.\wedge n + n*m*\$v.\wedge(n+1/m))) / (m*(1-\$v.\wedge(1/m)))^2$
if $i \neq 0$ $m \neq 0$ **for** $n m :: nat$
proof –
have $\star: \$v.\wedge(1/m) > 0$ **using** *v-pos* **by** *force*
hence $\$(I\wedge\{m\}a)\wedge\{m\}-n = (\sum k < n*m. (k+1)*(\$v.\wedge(1/m))^{\wedge(k+1)}) / m^2$

using *that*
apply (*subst Imam-calc-aux, simp*)
apply (*subst sum-divide-distrib[THEN sym], simp*)
apply (*rule sum.cong; simp*)
using *powr-realpow[THEN sym] powr-powr by (simp add: add-divide-distrib powr-add)*
also have $\dots = \$v.\hat{\sim}(1/m) * (\sum k < n*m. (k+1)*(\$v.\hat{\sim}(1/m))^k) / m^2$
by (*subst sum-distrib-left, simp add: that, rule sum.cong; simp*)
also have $\dots = \$v.\hat{\sim}(1/m) * ((1 - (n*m+1)*(\$v.\hat{\sim}(1/m))^{n*m}) + n*m*($v.\hat{\sim}(1/m))^{n*m+1}) / (1 - \$v.\hat{\sim}(1/m))^2 / m^2$
using *v-pos v-1-iff-i-0 that by (subst geometric-increasing-sum; simp?)*
also have $\dots = (\$v.\hat{\sim}(1/m) * (1 - (n*m+1)*\$v.\hat{\sim}^n + n*m*\$v.\hat{\sim}^{n+1/m})) / (m*(1-\$v.\hat{\sim}(1/m)))^2$
using \star
apply (*subst powr-realpow[of \$v.\hat{\sim}(1/m), THEN sym], simp*)
apply (*subst powr-powr*)
apply (*subst times-divide-eq-right[THEN sym], subst divide-divide-eq-left*)
apply (*subst power-mult-distrib*)
using *powr-eq-one-iff-gen v-pos v-1-iff-i-0 apply (simp add: field-simps)*
by (*(subst powr-realpow, simp)+, simp*)
finally show *?thesis .*
qed

lemma *Imam-calc-i-0*: $\$(I\hat{\sim}\{m\}a)\hat{\sim}\{m\}-n = (n*m+1)*n / (2*m)$ **if** $i = 0$ $m \neq 0$
for $n\ m :: \text{nat}$

proof –
have $\$(I\hat{\sim}\{m\}a)\hat{\sim}\{m\}-n = (\sum k < n*m. \$v.\hat{\sim}((k+1)/m) * (k+1) / m^2)$
by (*subst Imam-calc-aux, simp-all add: that*)
also have $\dots = (\sum k < n*m. k+1) / m^2$
apply (*subst v-1-iff-i-0[THEN iffD2], simp-all add: that*)
by (*subst sum-divide-distrib[THEN sym], simp*)
also have $\dots = (n*m*(n*m+1) \text{ div } 2) / m^2$
apply (*subst Suc-eq-plus1[THEN sym], subst sum-bounds-lt-plus1[of id, simplified]*)
by (*subst Sum-Icc-nat, simp*)
also have $\dots = (n*m+1)*n / (2*m)$
apply (*subst real-of-nat-div, simp*)
using *that by (subst power2-eq-square, simp add: field-simps)*
finally show *?thesis .*
qed

lemma *Imsm-calc*:

$\$(I\hat{\sim}\{m\}s)\hat{\sim}\{m\}-n = ((1+i).\hat{\sim}(n+1/m) - (n*m+1)*(1+i).\hat{\sim}(1/m) + n*m) / (m*((1+i).\hat{\sim}(1/m)-1))^2$
if $i \neq 0$ $m \neq 0$ **for** $n\ m :: \text{nat}$

proof –
have $\$(I\hat{\sim}\{m\}a)\hat{\sim}\{m\}-n = (\$v.\hat{\sim}^n * ((1+i).\hat{\sim}(n+1/m) - (n*m+1)*(1+i).\hat{\sim}(1/m) + n*m)) / (m*((1+i).\hat{\sim}(1/m)-1))^2$

proof –
have $\$(I\{m\}a)\{m\}-n =$
 $(\$v.\wedge(1/m) * (1 - (n*m+1)*\$v\wedge n + n*m*\$v.\wedge(n+1/m))) / (m*(1-\$v.\wedge(1/m)))\wedge 2$
using that by (*subst Imam-calc; simp*)
also have $\dots = (1 - (n*m+1)*\$v\wedge n + n*m*\$v.\wedge(n+1/m)) / (\$v.\wedge(1/m)*(m*(\$v.\wedge(-1/m)-1))\wedge 2)$
apply (*subgoal-tac* $\$v.\wedge(-1/m) = 1 / \$v.\wedge(1/m)$, *erule ssubst*)
apply (*(subst power2-eq-square)+, simp add: field-simps that*)
by (*simp add: powr-minus-divide*)
also have $\dots =$
 $(\$v.\wedge(n+1/m) * (\$v.\wedge(-n-1/m) - (n*m+1)*\$v.\wedge(-1/m) + n*m)) /$
 $(\$v.\wedge(1/m)*(m*(\$v.\wedge(-1/m)-1))\wedge 2)$
apply (*subgoal-tac* $\$v.\wedge(-n-1/m) = 1 / \$v.\wedge(n+1/m)$ $\$v.\wedge(-1/m) = \$v\wedge n$
 $/ \$v.\wedge(n+1/m)$)
apply (*(erule ssubst)+, simp-all add: field-simps*)
using *v-pos*
apply (*simp add: powr-diff[THEN sym] powr-realpow[THEN sym]*)
by (*smt powr-minus-divide*)
also have $\dots =$
 $(\$v\wedge n * (\$v.\wedge(-n-1/m) - (n*m+1)*\$v.\wedge(-1/m) + n*m)) / ((m*(\$v.\wedge(-1/m)-1))\wedge 2)$
apply (*subst powr-add[of - n 1/m]*)
using *v-pos powr-realpow by simp*
also have $\dots =$
 $(\$v\wedge n * ((1+i).\wedge(n+1/m) - (n*m+1)*(1+i).\wedge(1/m) + n*m)) / ((m*((1+i).\wedge(1/m)-1))\wedge 2)$
apply (*subgoal-tac* $-n-1/m = -(n+1/m) - 1/m = -(1/m)$, *(erule ssubst)+*)
apply (*subst i-v-powr[THEN sym]*)
by *simp-all*
finally show *?thesis* .
qed
thus *?thesis*
apply –
using that *v-futr-pos*
apply (*subst Ilsm-Ilam, simp*)
apply (*erule ssubst, simp*)
apply (*rule disjI2*)
by (*subst power-mult-distrib[THEN sym], simp add: v-pres-def*)
qed

lemma *Imsm-calc-i-0*: $\$(I\{m\}s)\{m\}-n = (n*m+1)*n / (2*m)$ **if** $i = 0$ $m \neq 0$
for $n\ m :: nat$
using that
apply (*subst Ilsm-Ilam, simp*)
by (*subst Imam-calc-i-0; simp*)

lemma *Ila''m-Ilam*: $\$(I\{l\}a'')\{m\}-n = (1+i).\wedge(1/m) * \$(I\{l\}a)\{m\}-n$
if $l \neq 0$ $m \neq 0$ **for** $l\ m\ n :: nat$
unfolding *ann-incr-def ann-due-incr-def* **using that**
apply (*subst i-v, subst powr-powr, simp*)
apply (*subst sum-distrib-left*)
apply (*rule sum.cong; simp*)

apply (rule disjI2)
by (smt (verit) add-divide-distrib powr-add)

lemma $Ia''m\text{-calc}$: $\$(Ia'')^{\wedge\{m\}}\text{-}n = (\sum j < n. (j+1)/m * (\sum k=j*m..<(j+1)*m. \$v.^{\wedge(k/m)}))$

if $m \neq 0$ **for** $n\ m :: \text{nat}$
using that
apply (subst $Ia''m\text{-}Iam$; simp del: One-nat-def)
apply (subst $Iam\text{-}calc$; simp)
apply (subst sum-distrib-left)
apply (rule sum.cong; simp)
apply (subst sum-distrib-left)+
apply (rule sum.cong; simp)
apply (subst $i\text{-}v\text{-}powr$)
using $powr\text{-}add$ [of $\$v$, THEN sym] **by** (simp add: field-simps)

lemma $Ima''m\text{-}calc\text{-}aux$: $\$(I^{\wedge\{m\}}a'')^{\wedge\{m\}}\text{-}n = (\sum k < n*m. \$v.^{\wedge(k/m)} * (k+1) / m^{\wedge 2})$

if $m \neq 0$ **for** $m :: \text{nat}$
using that
apply (subst $Ia''m\text{-}Iam$, simp)
apply (subst $Imam\text{-}calc\text{-}aux$, simp)
apply (subst sum-distrib-left)
apply (rule sum.cong; simp)
using $powr\text{-}add$ [of $\$v$, THEN sym] $i\text{-}v\text{-}powr$ **by** (simp add: field-simps)

lemma $Ima''m\text{-}calc$: $\$(I^{\wedge\{m\}}a'')^{\wedge\{m\}}\text{-}n = (1 - (n*m+1)*\$v^{\wedge n} + n*m*\$v.^{\wedge(n+1/m)}) / (m*(1-\$v.^{\wedge(1/m)}))^{\wedge 2}$

if $i \neq 0$ $m \neq 0$ **for** $n\ m :: \text{nat}$
using that $v\text{-}pos$
apply (subst $Ia''m\text{-}Iam$, simp)
apply (subst $Imam\text{-}calc$; simp)
by (smt (verit, del-Insts) $i\text{-}v\text{-}powr$ $powr\text{-}add$ $powr\text{-}zero\text{-}eq\text{-}one$)

lemma $Ils''m\text{-}Ilsm$: $\$(I^{\wedge\{l\}}s')^{\wedge\{m\}}\text{-}n = (1+i).^{\wedge(1/m)} * \$(I^{\wedge\{l\}}s)^{\wedge\{m\}}\text{-}n$

if $l \neq 0$ $m \neq 0$ **for** $l\ m\ n :: \text{nat}$
unfolding $acc\text{-}incr\text{-}def$ $acc\text{-}due\text{-}incr\text{-}def$ sum-distrib-left **using** that
apply (intro sum.cong; simp)
by (smt (verit, ccfv-SIG) add-divide-distrib powr-add)

lemma $Ims''m\text{-}calc$:

$\$(I^{\wedge\{m\}}s'')^{\wedge\{m\}}\text{-}n =$
 $(1+i).^{\wedge(1/m)} * ((1+i).^{\wedge(n+1/m)} - (n*m+1)*(1+i).^{\wedge(1/m)} + n*m) /$
 $(m*((1+i).^{\wedge(1/m)}-1))^{\wedge 2}$

if $i \neq 0$ $m \neq 0$ **for** $n\ m :: \text{nat}$
using that **by** (simp add: $Ils''m\text{-}Ilsm$ $Imsm\text{-}calc$)

lemma $lim\text{-}Imam$: $(\lambda n. \$(I^{\wedge\{m\}}a)^{\wedge\{m\}}\text{-}n) \longrightarrow 1 / (\$i^{\wedge\{m\}}*\$d^{\wedge\{m\}})$ **if** $m \neq 0$ $i > 0$ **for** $m :: \text{nat}$

proof –

have $(\lambda n. \$ (I^{\wedge\{m\}} a)^{\wedge\{m\}-n} =$
 $(\lambda n. \$ v. \wedge(1/m) * (1 - (n*m+1)*\$v^{\wedge n} + n*m*\$v. \wedge(n+1/m)) / (m*(1-\$v. \wedge(1/m)))^{\wedge 2})$
using that by $(subst\ Imam\ calc; simp)$
moreover have $(\lambda n. \$ v. \wedge(1/m) * (1 - (n*m+1)*\$v^{\wedge n} + n*m*\$v. \wedge(n+1/m))$
 $/ (m*(1-\$v. \wedge(1/m)))^{\wedge 2}$
 $\longrightarrow 1 / (\$i^{\wedge\{m\}}*\$d^{\wedge\{m\}})$

proof –

have $\star: |\$v| < 1$
using $v\text{-lt-1-iff-i-pos } v\text{-pos}$ **that by force**
hence $(\lambda n. (n*m+1)*\$v^{\wedge n} \longrightarrow 0$
apply $(subst\ tendsto\ cong[of - (\lambda n. n*m*\$v^{\wedge n} + \$v^{\wedge n})])$
apply $(rule\ always\ eventually, rule\ allI)$
apply $(simp\ add: distrib\ right)$
apply $(subgoal\ tac\ 0 = 0 + 0, erule\ ssubst, intro\ tendsto\ intros; simp)$
apply $(subst\ mult.\ commute, subst\ mult.\ assoc)$
apply $(subgoal\ tac\ 0 = real\ m * 0, erule\ ssubst, intro\ tendsto\ intros; simp?)$
by $(rule\ powser\ times\ n\ limit\ 0; simp)$
moreover have $(\lambda n. n*m*\$v. \wedge(n+1/m)) \longrightarrow 0$
apply $(subst\ tendsto\ cong[of - (\lambda n. (m*\$v. \wedge(1/m))*(n*\$v^{\wedge n})])$
apply $(rule\ always\ eventually, rule\ allI)$
apply $(simp\ add: powr\ add\ powr\ realpow\ v\ pos)$
apply $(subgoal\ tac\ 0 = m*\$v. \wedge(1/m) * 0, erule\ ssubst, intro\ tendsto\ intros;$
 $simp?)$
by $(rule\ powser\ times\ n\ limit\ 0, simp\ add: \star)$
ultimately have $(\lambda n. \$ v. \wedge(1/m) * (1 - (n*m+1)*\$v^{\wedge n} + n*m*\$v. \wedge(n+1/m))$
 $/ (m*(1-\$v. \wedge(1/m)))^{\wedge 2}$
 $\longrightarrow \$v. \wedge(1/m) * (1 - 0 + 0) / (m*(1-\$v. \wedge(1/m)))^{\wedge 2}$
using $v\text{-lt-1-iff-i-pos } v\text{-pos}$ **that by** $(intro\ tendsto\ intros; simp)$
thus $?thesis$
unfolding $i\text{-nom-def}$ **using** $v\text{-pos}$ **that**
apply $(subst\ i\text{-v-powr}, subst\ powr\ minus\ divide, subst\ d\ nom\ v; simp)$
by $(subst(asm)(2)\ power2\ eq\ square, simp\ add: field_simps)$
qed
ultimately show $?thesis$ **by** $simp$
qed

lemma $perp\ incr\ calc: \$ (I^{\wedge\{m\}} a)^{\wedge\{m\}-\infty} = 1 / (\$i^{\wedge\{m\}}*\$d^{\wedge\{m\}})$ **if** $m \neq 0$ $i > 0$ **for** $m::nat$

unfolding $perp\ incr\ def$ **by** $(rule\ limI, rule\ lim\ Imam; simp\ add: that)$

lemma $lim\ Ima''m: (\lambda n. \$ (I^{\wedge\{m\}} a'')^{\wedge\{m\}-n} \longrightarrow 1 / (\$d^{\wedge\{m\}})^{\wedge 2}$ **if** $m \neq 0$ $i > 0$ **for** $m::nat$

unfolding $perp\ due\ incr\ def$ **using that**

apply $(subst\ Ila''m\ Ilam, simp, subst\ mult.\ commute, subst\ i\text{-v-powr}, subst\ powr\ minus\ divide)$
apply $(subgoal\ tac\ 1 / (\$d^{\wedge\{m\}})^{\wedge 2} = (1 / (\$i^{\wedge\{m\}}*\$d^{\wedge\{m\}}))*(1 / \$v. \wedge(1/m)), erule\ ssubst)$

apply $(intro\ tendsto\ intros, simp\ add: lim\ Imam)$

by $(simp\ add: d\ nom\ i\text{-nom}\ v\ power2\ eq\ square)$

```

lemma perp-due-incr-calc:  $(I^{\{m\}a'})^{\{m\}-\infty} = 1 / (d^{\{m\}})^2$  if  $m \neq 0$  i  $>$ 
0 for  $m::nat$ 
  unfolding perp-due-incr-def by (rule limI, rule lim-Ima''m; simp add: that)

end

end

```