Brownian Motion

Christian Pardillo-Laursen and Simon Foster

November 28, 2026

Abstract

Brownian motion is a mathematical model of continuous random movement. It starts at zero, has independent changes over disjoint time intervals, and the change over an interval of length t has a centred normal distribution with variance t. It is a fundamental example of a continuous-time stochastic process; see, for example, Klenke [1].

In this work, we construct Brownian motion in Isabelle/HOL. We first build consistent Gaussian joint distributions for its values at any finite collection of times, and use them to obtain a Gaussian process. We then apply the AFP formalisation of the Kolmogorov–Chentsov theorem [2] to construct a continuous modification of this process and prove that it starts at zero and has the required independent normal increments. In fact, its paths satisfy a stronger local Hölder-continuity property. We then construct Brownian motion in arbitrary finite-dimensional Euclidean spaces from independent coordinate processes, proving path continuity and independent centred multivariate-normal increments with covariance proportional to the elapsed time. Finally, for the real-valued process, we show that rescaling time by $c > 0$ and space by $1/\sqrt{c}$ leaves the distribution of the entire process unchanged.

Contents

1 Normal laws for increments **2**

2 Finite-dimensional laws from independent increments **3**

3 From consistent laws to a Gaussian process **10**

4 Brownian motion **15**

5 Brownian motion in finite-dimensional Euclidean space **19**

5.1 Centred Euclidean normal distributions 20

5.2 Independent families on a finite product 21

5.3 The Brownian-motion specification 21

5.4 Basic Brownian properties 21

theory *Gaussian-Increments*

imports *Kolmogorov-Chentsov.Kolmogorov-Chentsov HOL−Probability.Distributions*
begin

1 Normal laws for increments

HOL-Probability's *normal-density* takes a standard deviation, not a variance. The zero-variance case is a point mass; no zero-standard-deviation density is used. The negative-variance branch makes this a total measure-valued function, but subsequent theorems about Gaussian increments will assume nonnegative variances.

definition *centered-normal* :: *real* ⇒ *real measure* **where**
centered-normal v =
(*if v* ≤ *0 then return borel 0*
else density lborel (*normal-density 0* (*sqrt v*)))

lemma *centered-normal-zero* [*simp*]: *centered-normal 0* = *return borel 0*
⟨*proof*⟩

lemma *centered-normal-nonpositive* [*simp*]:
v ≤ *0* ⟹ *centered-normal v* = *return borel 0*
⟨*proof*⟩

lemma *prob-space-centered-normal* [*intro*]: *prob-space* (*centered-normal v*)
⟨*proof*⟩

lemma *sets-centered-normal* [*simp*]: *sets* (*centered-normal v*) = *sets borel*
⟨*proof*⟩

lemma *centered-normal-scale*:
assumes *c*: *0* < *c* **and** *v*: *0* ≤ *v*
shows *distr* (*centered-normal* (*c* ∗ *v*)) *borel* (λ*x*. *x* / *sqrt c*) =
centered-normal v
⟨*proof*⟩

lemma *integrable-centered-normal-moment*:
assumes *0* ≤ *v*
shows *integrable* (*centered-normal v*) (λ*x*. *x*^*k*)
⟨*proof*⟩

lemma *centered-normal-mean*:
assumes *0* ≤ *v*
shows (∫ *x*. *x* ∂*centered-normal v*) = *0*
⟨*proof*⟩

lemma *centered-normal-fourth-moment*:
assumes *0* ≤ *v*
shows (∫ *x*. *x*^*4* ∂*centered-normal v*) = *3* ∗ *v*^*2*
⟨*proof*⟩

lemma *centered-normal-fourth-nn-integral*:
  assumes *v*: *0* ≤ *v*
  shows (∫⁺*x*. *ennreal* (*x*^*4*) ∂*centered-normal v*) = *ennreal* (*3* ∗ *v*^*2*)
⟨*proof*⟩

lemma *centered-normal-add-positive*:
  assumes *prob*: *prob-space M*
    and *indep*: *prob-space.indep-var M borel X borel Y*
    and *measurable-X*: *X* ∈ *borel-measurable M*
    and *measurable-Y*: *Y* ∈ *borel-measurable M*
    and *v*: *0* < *v* **and** *w*: *0* < *w*
    and *X*: *distr M borel X* = *centered-normal v*
    and *Y*: *distr M borel Y* = *centered-normal w*
  shows *distr M borel* (λ*z*. *X z* + *Y z*) = *centered-normal* (*v* + *w*)
⟨*proof*⟩

lemma *centered-normal-add*:
  assumes *prob*: *prob-space M*
    and *indep*: *prob-space.indep-var M borel X borel Y*
    and *measurable-X*: *X* ∈ *borel-measurable M*
    and *measurable-Y*: *Y* ∈ *borel-measurable M*
    and *v*: *0* ≤ *v* **and** *w*: *0* ≤ *w*
    and *X*: *distr M borel X* = *centered-normal v*
    and *Y*: *distr M borel Y* = *centered-normal w*
  shows *distr M borel* (λ*z*. *X z* + *Y z*) = *centered-normal* (*v* + *w*)
⟨*proof*⟩

end
theory *Gaussian-Finite-Dimensional*
  imports *Gaussian-Increments*
begin

2 Finite-dimensional laws from independent increments

The list of times will be assumed sorted, distinct, and nonnegative in subsequent results. Index 0 represents the increment from time 0 to ts!0. Out-of-range indices are assigned variance 0 to keep this definition total.

definition *increment-variance* :: *real list* ⇒ *nat* ⇒ *real* **where**
  increment-variance ts i =
    (*if i* ≥ *length ts then 0*
     else if i = *0 then ts* ! *0*
     else ts ! *i* − *ts* ! (*i* − *1*))

lemma *increment-variance-nonnegative*:
  assumes *sorted ts* ∀ *t*∈*set ts*. *0* ≤ *t i* < *length ts*

shows $0 \leq$ *increment-variance ts i*
⟨*proof*⟩

definition *increment-law* :: *real list* ⇒ (*nat* ⇒ *real*) *measure* **where**
increment-law ts =
PiM {..<*length ts*} (λ*i*. *centered-normal* (*increment-variance ts i*))

lemma *prob-space-increment-law* [*intro*]: *prob-space* (*increment-law ts*)
⟨*proof*⟩

lemma *sets-increment-law* [*simp*]:
sets (*increment-law ts*) = *sets* (*PiM* {..<*length ts*} (λ-. *borel*))
⟨*proof*⟩

lemma *increment-law-component*:
assumes *i* < *length ts*
shows *distr* (*increment-law ts*) (*centered-normal* (*increment-variance ts i*))
(λ*w*. *w i*) = *centered-normal* (*increment-variance ts i*)
⟨*proof*⟩

definition *partial-sums* :: *nat* ⇒ (*nat* ⇒ *real*) ⇒ *nat* ⇒ *real* **where**
partial-sums n increments i =
(*if i* < *n then* (∑*j*≤*i*. *increments j*) *else undefined*)

lemma *partial-sums-increment*:
assumes 0 < *i i* < *n*
shows *partial-sums n w i* − *partial-sums n w* (*i* − 1) = *w i*
⟨*proof*⟩

definition *gaussian-finite-law* :: *real list* ⇒ (*nat* ⇒ *real*) *measure* **where**
gaussian-finite-law ts =
distr (*increment-law ts*) (*PiM* {..<*length ts*} (λ-. *borel*))
(*partial-sums* (*length ts*))

lemma *measurable-partial-sums*:
partial-sums (*length ts*) ∈
measurable (*increment-law ts*) (*PiM* {..<*length ts*} (λ-. *borel*))
⟨*proof*⟩

lemma *sets-gaussian-finite-law* [*simp*]:
sets (*gaussian-finite-law ts*) = *sets* (*PiM* {..<*length ts*} (λ-. *borel*))
⟨*proof*⟩

lemma *prob-space-gaussian-finite-law* [*intro*]: *prob-space* (*gaussian-finite-law ts*)
⟨*proof*⟩

lemma *gaussian-finite-law-empty*:
gaussian-finite-law [] = *PiM* {} (λ-. *borel*)
⟨*proof*⟩

lemma *gaussian-finite-law-initial*:
assumes *ts*: $ts \neq []$
shows *distr* (*gaussian-finite-law ts*) *borel* ($\lambda w.\ w\ 0$) =
centered-normal (*ts* ! *0*)
⟨*proof*⟩

lemma *gaussian-finite-law-starts-at-zero*:
assumes *ts*: $ts \neq []$ *ts* ! *0* = *0*
shows *AE w in gaussian-finite-law ts*. *w 0* = *0*
⟨*proof*⟩

lemma *gaussian-finite-law-increment*:
assumes *i*: *0* < *i i* < *length ts*
shows *distr* (*gaussian-finite-law ts*) *borel* ($\lambda w.\ w\ i - w\ (i - 1)$) =
centered-normal (*increment-variance ts i*)
⟨*proof*⟩

lemma *increment-law-independent*:
assumes *nonempty*: $ts \neq []$
shows *prob-space.indep-vars* (*increment-law ts*)
($\lambda i.$ *centered-normal* (*increment-variance ts i*)) ($\lambda i\ w.\ w\ i$)
{..<*length ts*}
⟨*proof*⟩

definition *coordinate-increments* :: *nat* ⇒ (*nat* ⇒ *real*) ⇒ *nat* ⇒ *real* **where**
coordinate-increments n w i =
(*if i* < *n then* (*if i* = *0 then w 0 else w i* − *w* (*i* − *1*)) *else undefined*)

lemma *coordinate-increments-partial-sums*:
coordinate-increments n (*partial-sums n w*) = *restrict w* {..<*n*}
⟨*proof*⟩

lemma *partial-sums-coordinate-increments*:
partial-sums n (*coordinate-increments n w*) = *restrict w* {..<*n*}
⟨*proof*⟩

lemma *measurable-coordinate-increments*:
coordinate-increments (*length ts*) ∈
measurable (*PiM* {..<*length ts*} (λ-. *borel*)) (*increment-law ts*)
⟨*proof*⟩

lemma *gaussian-finite-law-coordinate-increments*:
distr (*gaussian-finite-law ts*) (*increment-law ts*)
(*coordinate-increments* (*length ts*)) = *increment-law ts*
⟨*proof*⟩

lemma *gaussian-finite-law-independent-increments*:
assumes *nonempty*: $ts \neq []$

shows *prob-space.indep-vars* (*gaussian-finite-law ts*)
(λ*i*. *centered-normal* (*increment-variance ts i*))
(λ*i w*. *coordinate-increments* (*length ts*) *w i*) {..<*length ts*}
⟨*proof*⟩

lemma *gaussian-finite-law-independent-borel-increments*:
assumes *ts* ≠ []
shows *prob-space.indep-vars* (*gaussian-finite-law ts*) (λ-. *borel*)
(λ*i w*. *coordinate-increments* (*length ts*) *w i*) {..<*length ts*}
⟨*proof*⟩

lemma *gaussian-finite-law-coordinate-increment*:
assumes *i*: *i* < *length ts*
shows *distr* (*gaussian-finite-law ts*) *borel*
(λ*w*. *coordinate-increments* (*length ts*) *w i*) =
centered-normal (*increment-variance ts i*)
⟨*proof*⟩

lemma *gaussian-finite-law-adjacent-independent*:
assumes *bound*: *Suc i* < *length ts*
shows *prob-space.indep-var* (*gaussian-finite-law ts*)
borel (λ*w*. *coordinate-increments* (*length ts*) *w i*)
borel (λ*w*. *coordinate-increments* (*length ts*) *w* (*Suc i*))
⟨*proof*⟩

lemma *gaussian-finite-law-adjacent-sum*:
assumes *sorted*: *sorted ts* **and** *nonnegative*: ∀ *t*∈*set ts*. *0* ≤ *t*
and *bound*: *Suc i* < *length ts*
shows *distr* (*gaussian-finite-law ts*) *borel*
(λ*w*. *coordinate-increments* (*length ts*) *w i* +
coordinate-increments (*length ts*) *w* (*Suc i*)) =
centered-normal (*increment-variance ts i* + *increment-variance ts* (*Suc i*))
⟨*proof*⟩

definition *delete-time* :: *real list* ⇒ *nat* ⇒ *real list* **where**
delete-time ts j = *take j ts* @ *drop* (*Suc j*) *ts*

lemma *delete-time-length*:
assumes *j* < *length ts*
shows *length* (*delete-time ts j*) = *length ts* − *1*
⟨*proof*⟩

lemma *delete-time-nth*:
assumes *j* < *length ts i* < *length* (*delete-time ts j*)
shows *delete-time ts j* ! *i* = (*if i* < *j then ts* ! *i else ts* ! *Suc i*)
⟨*proof*⟩

lemma *increment-variance-delete-time-merged*:
assumes *bound*: *Suc j* < *length ts*

shows *increment-variance* (*delete-time ts j*) *j* =
increment-variance ts j + *increment-variance ts* (*Suc j*)
⟨*proof*⟩

definition *delete-coordinate* :: *nat* ⇒ *nat* ⇒ (*nat* ⇒ *real*) ⇒ *nat* ⇒ *real* **where**
delete-coordinate n j w i =
(*if i* < *n* − *1 then w* (*if i* < *j then i else Suc i*) *else undefined*)

lemma *delete-coordinate-merged-increment*:
assumes *bound*: *Suc j* < *n*
shows *coordinate-increments* (*n* − *1*) (*delete-coordinate n j w*) *j* =
coordinate-increments n w j + *coordinate-increments n w* (*Suc j*)
⟨*proof*⟩

lemma *increment-variance-delete-time-unaffected*:
assumes *j*: *j* < *length ts* **and** *i*: *i* < *length* (*delete-time ts j*)
and *different*: *i* ≠ *j*
shows *increment-variance* (*delete-time ts j*) *i* =
increment-variance ts (*if i* < *j then i else Suc i*)
⟨*proof*⟩

lemma *delete-coordinate-unaffected-increment*:
assumes *j*: *j* < *length ts* **and** *i*: *i* < *length* (*delete-time ts j*)
and *different*: *i* ≠ *j*
shows *coordinate-increments* (*length* (*delete-time ts j*))
(*delete-coordinate* (*length ts*) *j w*) *i* =
coordinate-increments (*length ts*) *w* (*if i* < *j then i else Suc i*)
⟨*proof*⟩

lemma *measurable-delete-coordinate*:
assumes *j*: *j* < *length ts*
shows *delete-coordinate* (*length ts*) *j* ∈
measurable (*PiM* {..<*length ts*} (λ-. *borel*))
(*PiM* {..<*length* (*delete-time ts j*)} (λ-. *borel*))
⟨*proof*⟩

lemma *gaussian-finite-law-deleted-time-unaffected-increment*:
assumes *j*: *j* < *length ts* **and** *i*: *i* < *length* (*delete-time ts j*)
and *different*: *i* ≠ *j*
shows *distr* (*gaussian-finite-law ts*) *borel*
(λ*w*. *coordinate-increments* (*length* (*delete-time ts j*))
(*delete-coordinate* (*length ts*) *j w*) *i*) =
centered-normal (*increment-variance* (*delete-time ts j*) *i*)
⟨*proof*⟩

lemma *gaussian-finite-law-deleted-time-merged-increment*:
assumes *sorted*: *sorted ts* **and** *nonnegative*: ∀ *t*∈*set ts*. *0* ≤ *t*
and *bound*: *Suc j* < *length ts*
shows *distr* (*gaussian-finite-law ts*) *borel*

(λw. coordinate-increments (length (delete-time ts j))
(delete-coordinate (length ts) j w) j) =
centered-normal (increment-variance (delete-time ts j) j)
⟨proof⟩

definition *deletion-block* :: *nat* ⇒ *nat* ⇒ *nat set* **where**
deletion-block j i =
(*if i* = *j then* {*j*, *Suc j*} *else* {*if i* < *j then i else Suc i*})

lemma *deletion-block-subset*:
assumes *bound*: *Suc j* < *n* **and** *i*: *i* < *n* − *1*
shows *deletion-block j i* ⊆ {..<*n*}
⟨*proof*⟩

lemma *deletion-blocks-disjoint*:
disjoint-family-on (*deletion-block j*) {..<*n* − *1*}
⟨*proof*⟩

lemma *gaussian-finite-law-deleted-time-independent-increments*:
assumes *bound*: *Suc j* < *length ts*
shows *prob-space.indep-vars* (*gaussian-finite-law ts*) (λ-. *borel*)
(λ*i w*. *coordinate-increments* (*length* (*delete-time ts j*))
(*delete-coordinate* (*length ts*) *j w*) *i*)
{..<*length* (*delete-time ts j*)}
⟨*proof*⟩

lemma *gaussian-finite-law-deleted-time-increment-law*:
assumes *sorted*: *sorted ts* **and** *nonnegative*: ∀ *t*∈*set ts*. *0* ≤ *t*
and *bound*: *Suc j* < *length ts*
shows *distr* (*gaussian-finite-law ts*) (*increment-law* (*delete-time ts j*))
(λ*w*. *coordinate-increments* (*length* (*delete-time ts j*))
(*delete-coordinate* (*length ts*) *j w*)) = *increment-law* (*delete-time ts j*)
⟨*proof*⟩

theorem *gaussian-finite-law-delete-time*:
assumes *sorted*: *sorted ts* **and** *nonnegative*: ∀ *t*∈*set ts*. *0* ≤ *t*
and *bound*: *Suc j* < *length ts*
shows *distr* (*gaussian-finite-law ts*)
(*PiM* {..<*length* (*delete-time ts j*)} (λ-. *borel*))
(*delete-coordinate* (*length ts*) *j*) = *gaussian-finite-law* (*delete-time ts j*)
⟨*proof*⟩

lemma *partial-sums-restrict*:
assumes *k* ≤ *n*
shows *restrict* (*partial-sums n w*) {..<*k*} =
partial-sums k (*restrict w* {..<*k*})
⟨*proof*⟩

lemma *increment-variance-take*:

assumes $k \leq$ *length ts* $i < k$
shows *increment-variance* (*take k ts*) i = *increment-variance ts i*
⟨*proof*⟩

lemma *increment-law-prefix*:
assumes *k*: $k \leq$ *length ts*
shows *distr* (*increment-law ts*)
(*PiM* {..<*k*} (λ*i*. *centered-normal* (*increment-variance* (*take k ts*) *i*)))
(λ*w*. *restrict w* {..<*k*}) = *increment-law* (*take k ts*)
⟨*proof*⟩

lemma *gaussian-finite-law-prefix*:
assumes *k*: $k \leq$ *length ts*
shows *distr* (*gaussian-finite-law ts*) (*PiM* {..<*k*} (λ-. *borel*))
(λ*w*. *restrict w* {..<*k*}) = *gaussian-finite-law* (*take k ts*)
⟨*proof*⟩

corollary *gaussian-finite-law-butlast*:
assumes *ts* ≠ []
shows *distr* (*gaussian-finite-law ts*)
(*PiM* {..<*length* (*butlast ts*)} (λ-. *borel*))
(λ*w*. *restrict w* {..<*length* (*butlast ts*)}) =
gaussian-finite-law (*butlast ts*)
⟨*proof*⟩

corollary *gaussian-finite-law-delete-time-any*:
assumes *sorted*: *sorted ts* **and** *nonnegative*: ∀ *t*∈*set ts*. $0 \leq t$
and *j*: $j <$ *length ts*
shows *distr* (*gaussian-finite-law ts*)
(*PiM* {..<*length* (*delete-time ts j*)} (λ-. *borel*))
(*delete-coordinate* (*length ts*) *j*) = *gaussian-finite-law* (*delete-time ts j*)
⟨*proof*⟩

lemma *increment-variance-scale*:
increment-variance (*map* (λ*t*. *c* ∗ *t*) *ts*) *i* =
c ∗ *increment-variance ts i*
⟨*proof*⟩

lemma *increment-law-scale*:
assumes *c*: $0 < c$ **and** *sorted*: *sorted ts*
and *nonnegative*: ∀ *t*∈*set ts*. $0 \leq t$
shows *distr* (*increment-law* (*map* (λ*t*. *c* ∗ *t*) *ts*)) (*increment-law ts*)
(*compose* {..<*length ts*} (λ*x*. *x* / *sqrt c*)) = *increment-law ts*
⟨*proof*⟩

lemma *gaussian-finite-law-scale*:
assumes *c*: $0 < c$ **and** *sorted*: *sorted ts*
and *nonnegative*: ∀ *t*∈*set ts*. $0 \leq t$
shows *distr* (*gaussian-finite-law* (*map* (λ*t*. *c* ∗ *t*) *ts*))

(*PiM* {..<*length ts*} (λ-. *borel*))
(*compose* {..<*length ts*} (λ*x*. *x* / *sqrt c*)) =
gaussian-finite-law ts
⟨*proof*⟩

Single-time deletion consistency for the actual joint laws. AFP stochastic-process distributions form a product of marginals, not the joint law.

end
theory *Gaussian-Process*
imports *Gaussian-Finite-Dimensional HOL−Probability.Projective-Limit*
begin

3 From consistent laws to a Gaussian process

definition *ordered-times* :: *real set* ⇒ *real list* **where**
ordered-times T = *sorted-list-of-set T*

definition *time-index* :: *real set* ⇒ *real* ⇒ *nat* **where**
time-index T t = (*THE i*. *i* < *length* (*ordered-times T*) ∧ *ordered-times T* ! *i* = *t*)

lemma *ordered-times-finite*:
assumes *finite T*
shows *set* (*ordered-times T*) = *T distinct* (*ordered-times T*)
sorted (*ordered-times T*) *length* (*ordered-times T*) = *card T*
⟨*proof*⟩

lemma *time-index-unique*:
assumes *finite*: *finite T* **and** *t*: *t* ∈ *T*
shows *time-index T t* < *length* (*ordered-times T*)
ordered-times T ! *time-index T t* = *t*
⟨*proof*⟩

lemma *ordered-times-delete*:
assumes *finite*: *finite T* **and** *t*: *t* ∈ *T*
shows *ordered-times* (*T* − {*t*}) =
delete-time (*ordered-times T*) (*time-index T t*)
⟨*proof*⟩

definition *set-coordinates* :: *real set* ⇒ (*nat* ⇒ *real*) ⇒ *real* ⇒ *real* **where**
set-coordinates T w = (λ*t*∈*T*. *w* (*time-index T t*))

lemma *set-coordinates-delete*:
assumes *finite*: *finite T* **and** *t*: *t* ∈ *T*
shows *set-coordinates* (*T* − {*t*})
(*delete-coordinate* (*length* (*ordered-times T*)) (*time-index T t*) *w*) =
restrict (*set-coordinates T w*) (*T* − {*t*})
⟨*proof*⟩

lemma *ordered-times-set*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
shows *ordered-times* (*set ts*) = *ts*
⟨*proof*⟩

lemma *time-index-nth*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts* **and** *i*: *i* < *length ts*
shows *time-index* (*set ts*) (*ts* ! *i*) = *i*
⟨*proof*⟩

definition *list-coordinates* :: *real list* ⇒ (*real* ⇒ *real*) ⇒ *nat* ⇒ *real* **where**
list-coordinates ts w = (λ*i*∈{..<*length ts*}. *w* (*ts* ! *i*))

lemma *measurable-list-coordinates*:
list-coordinates ts ∈
measurable (*PiM* (*set ts*) (λ-. *borel*))
(*PiM* {..<*length ts*} (λ-. *borel*))
⟨*proof*⟩

lemma *list-coordinates-set-coordinates*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
shows *list-coordinates ts* (*set-coordinates* (*set ts*) *w*) =
restrict w {..<*length ts*}
⟨*proof*⟩

definition *gaussian-set-law* :: *real set* ⇒ (*real* ⇒ *real*) *measure* **where**
gaussian-set-law T =
distr (*gaussian-finite-law* (*ordered-times T*)) (*PiM T* (λ-. *borel*))
(*set-coordinates T*)

lemma *measurable-set-coordinates*:
assumes *finite*: *finite T*
shows *set-coordinates T* ∈
measurable (*PiM* {..<*length* (*ordered-times T*)} (λ-. *borel*))
(*PiM T* (λ-. *borel*))
⟨*proof*⟩

lemma *prob-space-gaussian-set-law*:
assumes *finite T*
shows *prob-space* (*gaussian-set-law T*)
⟨*proof*⟩

lemma *sets-gaussian-set-law*:
sets (*gaussian-set-law T*) = *sets* (*PiM T* (λ-. *borel*))
⟨*proof*⟩

lemma *gaussian-set-law-ordered-list*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
shows *distr* (*gaussian-set-law* (*set ts*))

(*PiM* {..<*length ts*} (λ-. *borel*)) (*list-coordinates ts*) =
gaussian-finite-law ts
⟨*proof*⟩

lemma *gaussian-set-law-delete*:
assumes *finite*: *finite T* **and** *nonnegative*: $T \subseteq \{0..\}$
and *t*: $t \in T$
shows *distr* (*gaussian-set-law T*) (*PiM* ($T - \{t\}$) (λ-. *borel*))
(λ*w*. *restrict w* ($T - \{t\}$)) = *gaussian-set-law* ($T - \{t\}$)
⟨*proof*⟩

lemma *gaussian-set-law-consistent*:
assumes *subset*: $J \subseteq H$ **and** *finite*: *finite H*
and *nonnegative*: $H \subseteq \{0..\}$
shows *gaussian-set-law J* =
distr (*gaussian-set-law H*) (*PiM J* (λ-. *borel*)) (λ*w*. *restrict w J*)
⟨*proof*⟩

interpretation *gaussian*: *projective-family* {*0*..} :: *real set*
gaussian-set-law λ-. *borel* :: *real measure*
⟨*proof*⟩

interpretation *gaussian-limit*: *polish-projective* {*0*..} :: *real set gaussian-set-law*
⟨*proof*⟩

definition *gaussian-path-law* :: (*real* ⇒ *real*) *measure* **where**
gaussian-path-law =
projective-family.lim ({*0*..} :: *real set*) *gaussian-set-law* (λ-. *borel*)

lemma *prob-space-gaussian-path-law*: *prob-space gaussian-path-law*
⟨*proof*⟩

lemma *sets-gaussian-path-law*:
sets gaussian-path-law = *sets* (*PiM* {*0*..} (λ-. *borel* :: *real measure*))
⟨*proof*⟩

lemma *gaussian-limit-marginal*:
assumes *finite*: *finite J* **and** *subset*: $J \subseteq \{0..\}$
shows *distr gaussian-path-law* (*PiM J* (λ-. *borel*))
(λ*w*. *restrict w J*) = *gaussian-set-law J*
⟨*proof*⟩

interpretation *gaussian-path*: *prob-space gaussian-path-law*
⟨*proof*⟩

lemma *gaussian-path-coordinate-measurable*:
assumes $t \in \{0..\}$
shows (λ*w* :: *real* ⇒ *real*. *w t*) ∈ *borel-measurable gaussian-path-law*
⟨*proof*⟩

lemma *gaussian-path-coordinates-measurable*:
$\forall\, t{\in}\{0..\}.\ (\lambda w :: real \Rightarrow real.\ w\ t) \in$ *borel-measurable gaussian-path-law*
⟨*proof*⟩

definition *gaussian-process* :: (*real*, *real* ⇒ *real*, *real*) *stochastic-process* **where**
gaussian-process = *prob-space.process-of gaussian-path-law borel* {0..}
(λ*t w*. *w t*) *0*

lemma *gaussian-process-index*: *proc-index gaussian-process* = {0..}
⟨*proof*⟩

lemma *gaussian-process-source*: *proc-source gaussian-process* = *gaussian-path-law*
⟨*proof*⟩

lemma *gaussian-process-target*: *proc-target gaussian-process* = *borel*
⟨*proof*⟩

lemma *gaussian-process-coordinate*:
assumes $t \in \{0..\}$
shows *process gaussian-process t* = (λ*w*. *w t*)
⟨*proof*⟩

lemma *gaussian-process-joint-law*:
assumes *finite*: *finite J* **and** *subset*: $J \subseteq \{0..\}$
shows *distr* (*proc-source gaussian-process*)
(*PiM J* (λ-. *proc-target gaussian-process*))
(λ*w*. λ*t*∈*J*. *process gaussian-process t w*) = *gaussian-set-law J*
⟨*proof*⟩

lemma *measurable-gaussian-process-ordered-coordinates*:
(λ*w*. λ*i*∈{..<*length ts*}. *process gaussian-process* (*ts* ! *i*) *w*) ∈
measurable (*proc-source gaussian-process*)
(*PiM* {..<*length ts*} (λ-. *borel*))
⟨*proof*⟩

lemma *gaussian-process-ordered-list-joint*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
and *nonnegative*: $\forall\, t{\in}set\ ts.\ 0 \le t$
shows *distr* (*proc-source gaussian-process*)
(*PiM* {..<*length ts*} (λ-. *borel*))
(λ*w*. λ*i*∈{..<*length ts*}. *process gaussian-process* (*ts* ! *i*) *w*) =
gaussian-finite-law ts
⟨*proof*⟩

lemma *time-index-adjacent*:
assumes *sorted*: *sorted ts* **and** *j*: $0 < j\ j < length\ ts$
and *jump*: *ts* ! (*j* − *1*) < *ts* ! *j*
shows *time-index* (*set ts*) (*ts* ! *j*) =

Suc (*time-index* (*set ts*) (*ts* ! (*j* − *1*)))
⟨*proof*⟩

lemma *gaussian-process-ordered-increment-law*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
and *nonnegative*: ∀ *t*∈*set ts*. *0* ≤ *t*
shows *distr* (*proc-source gaussian-process*) (*increment-law ts*)
(λ*w*. *coordinate-increments* (*length ts*)
(λ*i*∈{..<*length ts*}. *process gaussian-process* (*ts* ! *i*) *w*)) =
increment-law ts
⟨*proof*⟩

lemma *gaussian-process-ordered-independent-increments*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
and *nonnegative*: ∀ *t*∈*set ts*. *0* ≤ *t* **and** *nonempty*: *ts* ≠ []
shows *prob-space.indep-vars* (*proc-source gaussian-process*) (λ-. *borel*)
(λ*i w*. *coordinate-increments* (*length ts*)
(λ*k*∈{..<*length ts*}. *process gaussian-process* (*ts* ! *k*) *w*) *i*)
{..<*length ts*}
⟨*proof*⟩

corollary *gaussian-process-independent-successive-increments*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
and *nonnegative*: ∀ *t*∈*set ts*. *0* ≤ *t* **and** *nonempty*: *ts* ≠ []
shows *prob-space.indep-vars* (*proc-source gaussian-process*) (λ-. *borel*)
(λ*i w*. *if i* = *0 then process gaussian-process* (*ts* ! *0*) *w*
else process gaussian-process (*ts* ! *i*) *w* −
process gaussian-process (*ts* ! (*i* − *1*)) *w*) {..<*length ts*}
⟨*proof*⟩

corollary *gaussian-process-jointly-independent-increments*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
and *nonnegative*: ∀ *t*∈*set ts*. *0* ≤ *t*
and *nonempty*: *ts* ≠ []
shows *prob-space.indep-vars* (*proc-source gaussian-process*) (λ-. *borel*)
(λ*i w*. *process gaussian-process* (*ts* ! *i*) *w* −
process gaussian-process (*ts* ! (*i* − *1*)) *w*) {*1*..<*length ts*}
⟨*proof*⟩

theorem *gaussian-process-indep-increments*:
indep-increments gaussian-process
⟨*proof*⟩

lemma *gaussian-process-increment-law*:
assumes *s*: *0* ≤ *s* **and** *st*: *s* ≤ *t*
shows *distr* (*proc-source gaussian-process*) *borel*
(λ*w*. *process gaussian-process t w* − *process gaussian-process s w*) =
centered-normal (*t* − *s*)
⟨*proof*⟩

lemma *gaussian-set-law-zero-component*:
distr (*gaussian-set-law* {*0*}) *borel* (λ*w*. *w 0*) = *return borel 0*
⟨*proof*⟩

lemma *gaussian-process-starts-at-zero*:
AE w in proc-source gaussian-process. *process gaussian-process 0 w* = *0*
⟨*proof*⟩

The projective-limit process starts at zero almost surely. Its independent normal increments are established above; the fourth-moment bound and continuous modification follow in the final theory.

end
theory *Brownian-Motion*
imports *Gaussian-Process*
begin

4 Brownian motion

A Brownian motion is a real-valued process on nonnegative time, starting at zero almost surely, with jointly independent centred normal increments and almost surely continuous paths.

locale *brownian-motion* =
fixes *W* :: (*real*, *'a*, *real*) *stochastic-process*
assumes *index*: *proc-index W* = {*0..*}
and *init-0*: *AE w in proc-source W*. *process W 0 w* = *0*
and *indep-increments*: *indep-increments W*
and *normal-increments*: ⋀*s t*. *0* ≤ *s* ⟹ *s* ≤ *t* ⟹
distr (*proc-source W*) *borel*
(λ*w*. *process W t w* − *process W s w*) = *centered-normal* (*t* − *s*)
and *AE-continuous*: *AE w in proc-source W*.
continuous-on {*0..*} (λ*t*. *process W t w*)

The existence argument starts from the canonical Gaussian process. Its fourth increment moment has quadratic order in the time difference, which is the estimate needed for Kolmogorov–Chentsov.

lemma *gaussian-process-fourth-increment-moment-ordered*:
assumes *s*: *0* ≤ *s* **and** *st*: *s* ≤ *t*
shows (∫⁺ *w*. *ennreal* (*dist* (*process gaussian-process t w*)
(*process gaussian-process s w*) ^ *4*) ∂*proc-source gaussian-process*) =
ennreal (*3* ∗ (*dist t s*)^*2*)
⟨*proof*⟩

lemma *gaussian-process-fourth-increment-moment*:
assumes *s*: *0* ≤ *s* **and** *t*: *0* ≤ *t*
shows (∫⁺ *w*. *ennreal* (*dist* (*process gaussian-process t w*)
(*process gaussian-process s w*) ^ *4*) ∂*proc-source gaussian-process*) =

```
    ennreal (3 * (dist t s) ^ 2)
⟨proof⟩

lemma gaussian-process-KC-bound:
  assumes 0 ≤ s 0 ≤ t
  shows (∫⁺ w. dist (process gaussian-process t w)
    (process gaussian-process s w) powr 4 ∂proc-source gaussian-process) ≤
    3 * dist t s powr 2
  ⟨proof⟩
```

The fourth-moment estimate verifies the Kolmogorov–Chentsov hypotheses and therefore yields a modification whose paths are locally Holder continuous for every exponent below one quarter.

```
lemma gaussian-process-Kolmogorov-Chentsov:
  assumes 0 < γ γ < 1 / 4
  shows Kolmogorov-Chentsov gaussian-process 4 1 3 γ
⟨proof⟩

theorem gaussian-process-continuous-modification:
  assumes 0 < γ γ < 1 / 4
  shows ∃ B. modification gaussian-process B ∧
    (∀ w. local-holder-on γ {0..} (λt. process B t w))
  ⟨proof⟩
```

Modification preserves the finite-dimensional laws. In particular, the continuous version retains independent Gaussian increments and starts at zero almost surely, so it has all the Brownian motion properties.

```
theorem brownian-continuous-modification:
  assumes gamma: 0 < γ γ < 1 / 4
  shows ∃ B. modification gaussian-process B ∧
    (∀ w. local-holder-on γ {0..} (λt. process B t w)) ∧
    indep-increments B ∧
    (AE w in proc-source B. process B 0 w = 0) ∧
    (∀ s t. 0 ≤ s ∧ s ≤ t ⟶
      distr (proc-source B) borel
        (λw. process B t w − process B s w) = centered-normal (t − s))
⟨proof⟩
```

We now choose one such modification (with the concrete Holder exponent one eighth) and give it a permanent name. The subsequent interpretation packages the transferred properties and path continuity into the *brownian-motion* locale.

```
definition brownian-process :: (real, real ⇒ real, real) stochastic-process where
  brownian-process = (SOME B. modification gaussian-process B ∧
    (∀ w. local-holder-on (1/8) {0..} (λt. process B t w)) ∧
    indep-increments B ∧
    (AE w in proc-source B. process B 0 w = 0) ∧
    (∀ s t. 0 ≤ s ∧ s ≤ t ⟶
```

distr (*proc-source B*) *borel*
(λ*w*. *process B t w* − *process B s w*) = *centered-normal* (*t* − *s*)))

lemma *brownian-process-properties*:
modification gaussian-process brownian-process ∧
(∀ *w*. *local-holder-on* (*1*/*8*) {*0*..} (λ*t*. *process brownian-process t w*)) ∧
indep-increments brownian-process ∧
(*AE w in proc-source brownian-process*. *process brownian-process 0 w* = *0*) ∧
(∀ *s t*. *0* ≤ *s* ∧ *s* ≤ *t* ⟶
distr (*proc-source brownian-process*) *borel*
(λ*w*. *process brownian-process t w* − *process brownian-process s w*) =
centered-normal (*t* − *s*))
⟨*proof*⟩

lemma *brownian-process-modification*:
modification gaussian-process brownian-process
⟨*proof*⟩

lemma *brownian-process-local-holder*:
local-holder-on (*1*/*8*) {*0*..} (λ*t*. *process brownian-process t w*)
⟨*proof*⟩

lemma *brownian-process-indep-increments*:
indep-increments brownian-process
⟨*proof*⟩

lemma *brownian-process-starts-at-zero*:
AE w in proc-source brownian-process. *process brownian-process 0 w* = *0*
⟨*proof*⟩

lemma *brownian-process-increment-law*:
assumes *0* ≤ *s s* ≤ *t*
shows *distr* (*proc-source brownian-process*) *borel*
(λ*w*. *process brownian-process t w* − *process brownian-process s w*) =
centered-normal (*t* − *s*)
⟨*proof*⟩

interpretation *constructed-brownian-motion*: *brownian-motion brownian-process*
⟨*proof*⟩

lemma *modification-gaussian-joint-law*:
assumes *mod*: *modification gaussian-process B*
and *finite*: *finite T* **and** *subset*: *T* ⊆ {*0*..}
shows *distr* (*proc-source B*) (*PiM T* (λ-. *borel*))
(λ*w*. λ*t*∈*T*. *process B t w*) = *gaussian-set-law T*
⟨*proof*⟩

lemma *brownian-process-joint-law*:
assumes *finite T T* ⊆ {*0*..}

shows *distr* (*proc-source brownian-process*) (*PiM T* (λ-. *borel*))
(λ*w*. λ*t*∈*T*. *process brownian-process t w*) = *gaussian-set-law T*
⟨*proof*⟩

lemma *brownian-process-ordered-list-joint*:
assumes *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
and *nonnegative*: ∀ *t*∈*set ts*. 0 ≤ *t*
shows *distr* (*proc-source brownian-process*)
(*PiM* {..<*length ts*} (λ-. *borel*))
(λ*w*. λ*i*∈{..<*length ts*}. *process brownian-process* (*ts* ! *i*) *w*) =
gaussian-finite-law ts
⟨*proof*⟩

lemma *brownian-process-ordered-list-scale*:
assumes *c*: 0 < *c* **and** *sorted*: *sorted ts* **and** *distinct*: *distinct ts*
and *nonnegative*: ∀ *t*∈*set ts*. 0 ≤ *t*
shows *distr* (*proc-source brownian-process*)
(*PiM* {..<*length ts*} (λ-. *borel*))
(λ*w*. λ*i*∈{..<*length ts*}.
process brownian-process (*c* ∗ (*ts* ! *i*)) *w* / *sqrt c*) =
gaussian-finite-law ts
⟨*proof*⟩

theorem *brownian-process-finite-scale-invariance*:
assumes *c*: 0 < *c* **and** *finite*: *finite T* **and** *subset*: *T* ⊆ {0..}
shows *distr* (*proc-source brownian-process*) (*PiM T* (λ-. *borel*))
(λ*w*. λ*t*∈*T*. *process brownian-process* (*c* ∗ *t*) *w* / *sqrt c*) =
gaussian-set-law T
⟨*proof*⟩

theorem *brownian-process-scale-invariance*:
assumes 0 < *c finite T T* ⊆ {0..}
shows *distr* (*proc-source brownian-process*) (*PiM T* (λ-. *borel*))
(λ*w*. λ*t*∈*T*. *process brownian-process* (*c* ∗ *t*) *w* / *sqrt c*) =
distr (*proc-source brownian-process*) (*PiM T* (λ-. *borel*))
(λ*w*. λ*t*∈*T*. *process brownian-process t w*)
⟨*proof*⟩

theorem *brownian-process-path-scale-invariance*:
assumes *c*: 0 < *c*
shows *distr* (*proc-source brownian-process*)
(*PiM* {0..} (λ-. *borel* :: *real measure*))
(λ*w*. λ*t*∈{0..}. *process brownian-process* (*c* ∗ *t*) *w* / *sqrt c*) =
gaussian-path-law
⟨*proof*⟩

corollary *brownian-process-full-scale-invariance*:
assumes 0 < *c*
shows *distr* (*proc-source brownian-process*)

(*PiM* {*0*..} (λ-. *borel* :: *real measure*))
(λ*w*. λ*t*∈{*0*..}. *process brownian-process* (*c* ∗ *t*) *w* / *sqrt c*) =
distr (*proc-source brownian-process*)
(*PiM* {*0*..} (λ-. *borel* :: *real measure*))
(λ*w*. λ*t*∈{*0*..}. *process brownian-process t w*)
⟨*proof*⟩

The named process has Gaussian joint laws, independent normal increments, locally Holder-continuous paths, and Brownian scale invariance as an equality of full path-space distributions.

end
theory *Euclidean-Brownian-Motion*
imports *Brownian-Motion*
begin

5 Brownian motion in finite-dimensional Euclidean space

The construction uses one independent copy of the real Brownian motion for every coordinate. The dimension type is finite but otherwise arbitrary.

definition *euclidean-brownian-source* ::
('*d*::*finite* ⇒ *real* ⇒ *real*) *measure* **where**
euclidean-brownian-source = *PiM UNIV* (λ-. *proc-source brownian-process*)

lemma *prob-space-euclidean-brownian-source*:
prob-space (*euclidean-brownian-source* :: ('*d*::*finite* ⇒ *real* ⇒ *real*) *measure*)
⟨*proof*⟩

interpretation *euclidean-source*: *prob-space*
euclidean-brownian-source :: ('*d*::*finite* ⇒ *real* ⇒ *real*) *measure*
⟨*proof*⟩

lemma *borel-measurable-vec-lambda*:
fixes *f* :: '*a* ⇒ '*d*::*finite* ⇒ *real*
assumes *coord*: ⋀*i*. (λ*x*. *f x i*) ∈ *borel-measurable M*
shows (λ*x*. (χ *i*. *f x i*) :: *real*^'*d*) ∈ *borel-measurable M*
⟨*proof*⟩

definition *euclidean-brownian-path* ::
real ⇒ ('*d*::*finite* ⇒ *real* ⇒ *real*) ⇒ *real*^'*d* **where**
euclidean-brownian-path t w = (χ *i*. *process brownian-process t* (*w i*))

lemma *euclidean-brownian-path-measurable* [*measurable*]:
euclidean-brownian-path t ∈
borel-measurable (*euclidean-brownian-source* ::
('*d*::*finite* ⇒ *real* ⇒ *real*) *measure*)
⟨*proof*⟩

definition *euclidean-brownian-process* ::
  (*real*, *'d*::*finite* ⇒ *real* ⇒ *real*, *real^'d*) *stochastic-process* **where**
 euclidean-brownian-process =
  euclidean-source.process-of borel {*0*..} *euclidean-brownian-path 0*

lemma *euclidean-brownian-process-index* [*simp*]:
 proc-index (*euclidean-brownian-process* ::
  (*real*, *'d*::*finite* ⇒ *real* ⇒ *real*, *real^'d*) *stochastic-process*) = {*0*..}
 ⟨*proof*⟩

lemma *euclidean-brownian-process-source* [*simp*]:
 proc-source (*euclidean-brownian-process* ::
  (*real*, *'d*::*finite* ⇒ *real* ⇒ *real*, *real^'d*) *stochastic-process*) =
  euclidean-brownian-source
 ⟨*proof*⟩

lemma *euclidean-brownian-process-target* [*simp*]:
 proc-target (*euclidean-brownian-process* ::
  (*real*, *'d*::*finite* ⇒ *real* ⇒ *real*, *real^'d*) *stochastic-process*) = *borel*
 ⟨*proof*⟩

lemma *euclidean-brownian-process-apply* [*simp*]:
 assumes *0* ≤ *t*
 shows *process* (*euclidean-brownian-process* ::
    (*real*, *'d*::*finite* ⇒ *real* ⇒ *real*, *real^'d*) *stochastic-process*) *t w* =
  (χ *i*. *process brownian-process t* (*w i*))
⟨*proof*⟩

5.1 Centred Euclidean normal distributions

definition *euclidean-centered-normal* ::
  real ⇒ (*real^'d*::*finite*) *measure* **where**
 euclidean-centered-normal v =
  distr (*PiM UNIV* (λ-::*'d*. *centered-normal v*)) *borel* (λ*x*. χ *i*. *x i*)

lemma *prob-space-euclidean-centered-normal*:
 assumes *0* ≤ *v*
 shows *prob-space* (*euclidean-centered-normal v* :: (*real^'d*::*finite*) *measure*)
 ⟨*proof*⟩

lemma *sets-euclidean-centered-normal* [*simp*]:
 sets (*euclidean-centered-normal v* :: (*real^'d*::*finite*) *measure*) = *sets borel*
 ⟨*proof*⟩

lemma *euclidean-centered-normal-coordinate*:
 assumes *0* ≤ *v*
 shows *distr* (*euclidean-centered-normal v* :: (*real^'d*::*finite*) *measure*) *borel*
    (λ*x*. *x* $ *i*) = *centered-normal v*

⟨*proof*⟩

lemma *euclidean-centered-normal-independent-coordinates*:
assumes $0 \leq v$
shows *prob-space.indep-vars* (*euclidean-centered-normal* v ::
(*real*^'*d*::*finite*) *measure*) (λ-. *borel*) (λ*i x*. *x* $ *i*) *UNIV*
⟨*proof*⟩

lemma *euclidean-centered-normal-zero*:
euclidean-centered-normal 0 = *return borel* (*0* :: *real*^'*d*::*finite*)
⟨*proof*⟩

5.2 Independent families on a finite product

lemma *indep-vars-PiM-finite-family*:
fixes I :: '*i set* **and** J :: '*j set*
and M :: '*i* ⇒ '*a measure* **and** N :: '*i* ⇒ '*j* ⇒ '*b measure*
and X :: '*i* ⇒ '*j* ⇒ '*a* ⇒ '*b*
assumes *finite-I*: *finite I* **and** *finite-J*: *finite J*
and *nonempty-I*: $I \neq \{\}$ **and** *nonempty-J*: $J \neq \{\}$
and *ps-M*: $\bigwedge i.\ i \in I \Longrightarrow$ *prob-space* (*M i*)
and *indep*: $\bigwedge i.\ i \in I \Longrightarrow$
prob-space.indep-vars (*M i*) (*N i*) (*X i*) *J*
shows *prob-space.indep-vars* (*PiM I M*) (λ(*i*,*j*). *N i j*)
(λ(*i*,*j*) *w*. *X i j* (*w i*)) (*I* × *J*)
⟨*proof*⟩

5.3 The Brownian-motion specification

locale *euclidean-brownian-motion* =
fixes W :: (*real*, '*a*, *real*^'*d*::*finite*) *stochastic-process*
assumes *index*: *proc-index W* = {*0*..}
and *init-0*: *AE w in proc-source W*. *process W 0 w* = *0*
and *indep-increments*: *indep-increments W*
and *normal-increments*: $\bigwedge s\ t.\ 0 \leq s \Longrightarrow s \leq t \Longrightarrow$
distr (*proc-source W*) *borel*
(λ*w*. *process W t w* − *process W s w*) =
euclidean-centered-normal (*t* − *s*)
and *AE-continuous*: *AE w in proc-source W*.
continuous-on {*0*..} (λ*t*. *process W t w*)

5.4 Basic Brownian properties

lemma *euclidean-brownian-process-starts-at-zero*:
AE w in proc-source (*euclidean-brownian-process* ::
(*real*, '*d*::*finite* ⇒ *real* ⇒ *real*, *real*^'*d*) *stochastic-process*).
process euclidean-brownian-process 0 w = *0*
⟨*proof*⟩

lemma *euclidean-brownian-increment-product-law*:

fixes $s\ t :: real$
assumes st: $s \leq t$
defines $X \equiv \lambda(\text{-}::'d::finite)\ p.$
$process\ brownian\text{-}process\ t\ p - process\ brownian\text{-}process\ s\ p$
shows $distr\ (euclidean\text{-}brownian\text{-}source ::$
$('d \Rightarrow real \Rightarrow real)\ measure)$
$(PiM\ UNIV\ (\lambda\text{-}::'d.\ centered\text{-}normal\ (t - s)))$
$(\lambda w.\ restrict\ (\lambda i.\ X\ i\ (w\ i))\ UNIV) =$
$PiM\ UNIV\ (\lambda\text{-}::'d.\ centered\text{-}normal\ (t - s))$
⟨*proof*⟩

lemma *euclidean-brownian-process-increment-law*:
assumes s: $0 \leq s$ **and** st: $s \leq t$
shows $distr\ (proc\text{-}source\ (euclidean\text{-}brownian\text{-}process ::$
$(real, 'd::finite \Rightarrow real \Rightarrow real, real^{\frown}'d)\ stochastic\text{-}process))\ borel$
$(\lambda w.\ process\ euclidean\text{-}brownian\text{-}process\ t\ w -$
$process\ euclidean\text{-}brownian\text{-}process\ s\ w) =$
$euclidean\text{-}centered\text{-}normal\ (t - s)$
⟨*proof*⟩

lemma *euclidean-brownian-process-local-holder*:
$local\text{-}holder\text{-}on\ (1/8)\ \{0..\}$
$(\lambda t.\ process\ (euclidean\text{-}brownian\text{-}process ::$
$(real, 'd::finite \Rightarrow real \Rightarrow real, real^{\frown}'d)\ stochastic\text{-}process)\ t\ w)$
⟨*proof*⟩

lemma *euclidean-brownian-process-indep-increments*:
$indep\text{-}increments\ (euclidean\text{-}brownian\text{-}process ::$
$(real, 'd::finite \Rightarrow real \Rightarrow real, real^{\frown}'d)\ stochastic\text{-}process)$
⟨*proof*⟩

lemma *euclidean-brownian-process-continuous*:
$continuous\text{-}on\ \{0..\}$
$(\lambda t.\ process\ (euclidean\text{-}brownian\text{-}process ::$
$(real, 'd::finite \Rightarrow real \Rightarrow real, real^{\frown}'d)\ stochastic\text{-}process)\ t\ w)$
⟨*proof*⟩

interpretation *constructed-euclidean-brownian-motion*: *euclidean-brownian-motion*
$euclidean\text{-}brownian\text{-}process ::$
$(real, 'd::finite \Rightarrow real \Rightarrow real, real^{\frown}'d)\ stochastic\text{-}process$
⟨*proof*⟩

end

References

[1] A. Klenke. *Probability Theory: A Comprehensive Course*. Springer, 3 edition, 2020.

[2] C. Pardillo-Laursen and S. Foster. The kolmogorov-chentsov theorem. *Archive of Formal Proofs*, Apr. 2025. https://isa-afp.org/entries/Kolmogorov_Chentsov.html, Formal proof development.