Buffon's Needle Problem

Manuel Eberl

February 6, 2026

Abstract

In the 18th century, Georges-Louis Leclerc, Comte de Buffon posed and later solved the following problem [1, 2], which is often called the first problem ever solved in geometric probability: Given a floor divided into vertical strips of the same width, what is the probability that a needle thrown onto the floor randomly will cross two strips?

This entry formally defines the problem in the case where the needle's position is chosen uniformly at random in a single strip around the origin (which is equivalent to larger arrangements due to symmetry). It then provides proofs of the simple solution in the case where the needle's length is no greater than the width of the strips and the more complicated solution in the opposite case.

Contents

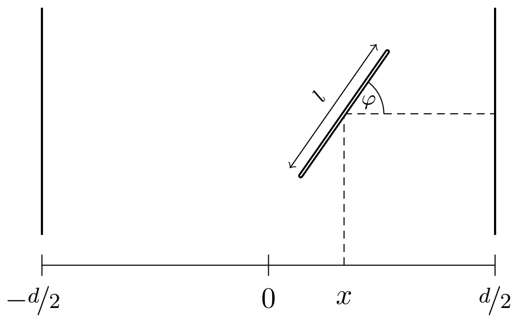

Figure 1: A sketch of the situation in Buffon's needle experiment. There is a needle of length l with its centre at a certain x coordinate, angled at an angle φ off the horizontal axis. The two vertical lines are a distance of d apart, each being $d/2$ away from the origin.

1 Buffon's Needle Problem

theory *Buffons-Needle*
  imports *HOL−Probability.Probability*
begin

1.1 Auxiliary material

lemma *sin-le-zero′*: *sin x ≤ 0* **if** *x ≥ −pi x ≤ 0* **for** *x*
  by (*metis minus-le-iff neg-0-le-iff-le sin-ge-zero sin-minus that*(*1*) *that*(*2*))

1.2 Problem definition

Consider a needle of length l whose centre has the x-coordinate x. The following then defines the set of all x-coordinates that the needle covers (i.e. the projection of the needle onto the x-axis.)

definition *needle* :: *real* ⇒ *real* ⇒ *real* ⇒ *real set* **where**
  needle l x φ = *closed-segment* (*x* − *l* / *2* ∗ *sin φ*) (*x* + *l* / *2* ∗ *sin φ*)

Buffon's Needle problem is then this: Assuming the needle's x position is chosen uniformly at random in a strip of width d centred at the origin, what is the probability that the needle crosses at least one of the left/right boundaries of that strip (located at $x = \pm\frac{1}{2}d$)?
We will show that, if we let $x := l/d$, the probability of this is

$$\mathcal{P}_{l,d} = \begin{cases} 2/\pi \cdot x & \text{if } l \leq d \\ 2/\pi \cdot (x - \sqrt{x^2 - 1} + \arccos(1/x)) & \text{if } l \geq d \end{cases}$$

A plot of this function can be found in Figure 2.

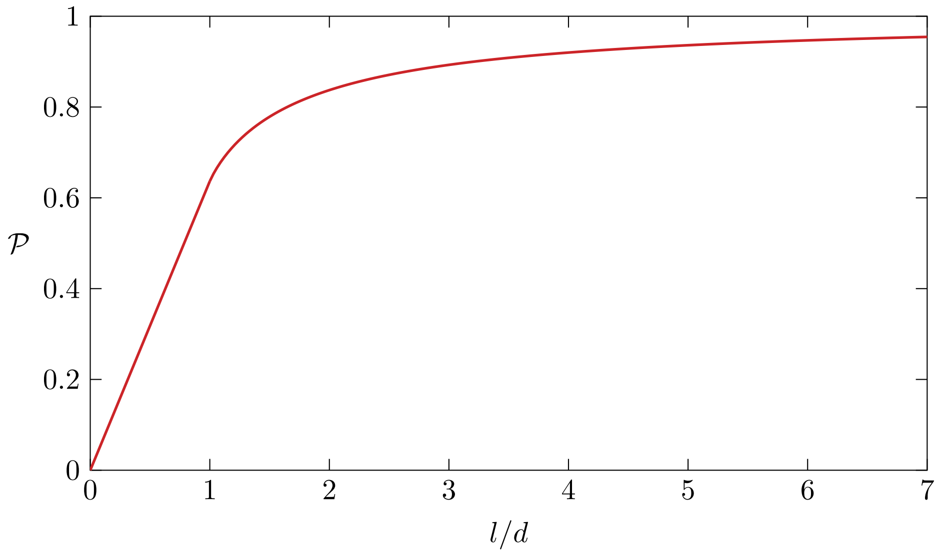

Figure 2: The probability $\mathcal{P}$ of the needle hitting one of the lines, as a function of the quotient l/d (where l is the length of the needle and d the horizontal distance between the lines).

locale *Buffon* =
  fixes *d l* :: *real*
  assumes *d*: *d* > *0* **and** *l*: *l* > *0*
begin

definition *Buffon* :: (*real* × *real*) *measure* **where**
  Buffon = *uniform-measure lborel* ({−*d*/*2*..*d*/*2*} × {−*pi*..*pi*})

lemma *space-Buffon* [*simp*]: *space Buffon* = *UNIV*
  by (*simp add*: *Buffon-def*)

definition *Buffon-set* :: (*real* × *real*) *set* **where**
  Buffon-set = {(*x*,*φ*) ∈ {−*d*/*2*..*d*/*2*} × {−*pi*..*pi*}. *needle l x φ* ∩ {−*d*/*2*, *d*/*2*} ≠ {}}

1.3 Derivation of the solution

The following form is a bit easier to handle.

lemma *Buffon-set-altdef1*:
  Buffon-set =
    {(*x*,*φ*) ∈ {−*d*/*2*..*d*/*2*} × {−*pi*..*pi*}.
      let a = *x* − *l* / *2* ∗ *sin φ*; *b* = *x* + *l* / *2* ∗ *sin φ*
       in min a b + *d*/*2* ≤ *0* ∧ *max a b* + *d*/*2* ≥ *0* ∨ *min a b* − *d*/*2* ≤ *0* ∧ *max a b* − *d*/*2* ≥ *0*}

```
proof -
  have (λ(x,φ). needle l x φ ∩ {-d/2, d/2} ≠ {}) =
      (λ(x,φ). let a = x - l / 2 * sin φ; b = x + l / 2 * sin φ
              in  -d/2 ≥ min a b ∧ -d/2 ≤ max a b ∨ min a b ≤ d/2 ∧ max a b
≥ d/2)
    by (auto simp: needle-def Let-def closed-segment-eq-real-ivl min-def max-def)
  also have ... =
    (λ(x,φ). let a = x - l / 2 * sin φ; b = x + l / 2 * sin φ
            in  min a b + d/2 ≤ 0 ∧ max a b + d/2 ≥ 0 ∨ min a b - d/2 ≤ 0 ∧
max a b - d/2 ≥ 0)
    by (auto simp add: algebra-simps Let-def)
  finally show ?thesis unfolding Buffon-set-def case-prod-unfold
    by (intro Collect-cong conj-cong refl) meson
qed

lemma Buffon-set-altdef2:
  Buffon-set = {(x,φ) ∈ {-d/2..d/2} × {-pi..pi}. abs x ≥ d / 2 - abs (sin φ)
* l / 2}
  unfolding Buffon-set-altdef1
proof (intro Collect-cong prod.case-cong refl conj-cong)
  fix x φ
  assume *: (x, φ) ∈ {-d/2..d/2} × {-pi..pi}
  let ?P = λx φ. let a = x - l / 2 * sin φ; b = x + l / 2 * sin φ
              in  min a b + d/2 ≤ 0 ∧ max a b + d/2 ≥ 0 ∨ min a b - d/2 ≤ 0
∧ max a b - d/2 ≥ 0

  show ?P x φ ⟷ (d / 2 - |sin φ| * l / 2 ≤ |x|)
  proof (cases φ ≥ 0)
    case True
    have x - l / 2 * sin φ ≤ x + l / 2 * sin φ using l True *
      by (auto simp: sin-ge-zero)
    moreover from True and * have sin φ ≥ 0 by (auto simp: sin-ge-zero)
    ultimately show ?thesis using * True
      by (force simp: field-simps Let-def min-def max-def case-prod-unfold abs-if)
  next
    case False
    with * have x - l / 2 * sin φ ≥ x + l / 2 * sin φ using l
      by (auto simp: sin-le-zero' mult-nonneg-nonpos)
    moreover from False and * have sin φ ≤ 0 by (auto simp: sin-le-zero')
    ultimately show ?thesis using * False l d
      by (force simp: field-simps Let-def min-def max-def case-prod-unfold abs-if)
  qed
qed
```

By using the symmetry inherent in the problem, we can reduce the problem to the following set, which corresponds to one quadrant of the original set:

```
definition Buffon-set' :: (real × real) set where
  Buffon-set' = {(x,φ) ∈ {0..d/2} × {0..pi}. x ≥ d / 2 - sin φ * l / 2}
```

```
lemma closed-buffon-set [simp, intro, measurable]: closed Buffon-set
  proof -
  have Buffon-set = ({−d/2..d/2} × {−pi..pi}) ∩
          (λz. abs (fst z) + abs (sin (snd z)) ∗ l / 2 − d / 2) −‘ {0..}
    (is - = ?A) unfolding Buffon-set-altdef2 by auto
  also have closed . . .
   by (intro closed-Int closed-vimage closed-Times) (auto intro!: continuous-intros)
  finally show ?thesis by simp
qed

lemma closed-buffon-set′ [simp, intro, measurable]: closed Buffon-set′
proof -
  have Buffon-set′ = ({0..d/2} × {0..pi}) ∩
          (λz. fst z + sin (snd z) ∗ l / 2 − d / 2) −‘ {0..}
    (is - = ?A) unfolding Buffon-set′-def by auto
  also have closed . . .
   by (intro closed-Int closed-vimage closed-Times) (auto intro!: continuous-intros)
  finally show ?thesis by simp
qed

lemma measurable-buffon-set [measurable]: Buffon-set ∈ sets borel
  by measurable

lemma measurable-buffon-set′ [measurable]: Buffon-set′ ∈ sets borel
  by measurable


sublocale prob-space Buffon
  unfolding Buffon-def
proof -
  have emeasure lborel ({− d / 2..d / 2} × {− pi..pi}) = ennreal (2 ∗ d ∗ pi)
    unfolding lborel-prod [symmetric] using d
    by (subst lborel.emeasure-pair-measure-Times)
       (auto simp: ennreal-mult mult-ac simp flip: ennreal-numeral)
  also have . . . ≠ 0 ∧ . . . ≠ ∞
    using d by auto
 finally show prob-space (uniform-measure lborel ({− d / 2..d / 2} × {− pi..pi}))
    by (intro prob-space-uniform-measure) auto
qed

lemma buffon-prob-aux:
  emeasure Buffon {(x,φ). needle l x φ ∩ {−d/2, d/2} ≠ {}} =
     emeasure lborel Buffon-set / ennreal (2 ∗ d ∗ pi)
proof -
  have [measurable]: A × B ∈ sets borel if A ∈ sets borel B ∈ sets borel
    for A B :: real set using that unfolding borel-prod [symmetric] by simp
  have {(x, φ). needle l x φ ∩ {− d / 2, d / 2} ≠ {}} ∈ sets borel
    by (intro pred-Collect-borel)
     (simp add: borel-prod [symmetric] needle-def closed-segment-eq-real-ivl case-prod-unfold)
```

hence *emeasure Buffon* $\{(x,\varphi).$ *needle l x* $\varphi \cap \{-d/2,\ d/2\} \neq \{\}\} =$
emeasure lborel $((\{-d/2..d/2\} \times \{-\ pi..pi\}) \cap \{(x,\varphi).$ *needle l x* $\varphi \cap \{-d/2,\ d/2\} \neq \{\}\}) /$
emeasure lborel $(\{-(d/2)..d/2\} \times \{-pi..pi\})$
unfolding *Buffon-def Buffon-set-def* **by** (*subst emeasure-uniform-measure*) *simp-all*
also have $(\{-d/2..d/2\} \times \{-\ pi..pi\}) \cap \{(x,\ \varphi).$ *needle l x* $\varphi \cap \{-d/2,\ d/2\} \neq \{\}\} =$ *Buffon-set*
unfolding *Buffon-set-def* **by** *auto*
also have *emeasure lborel* $(\{-(d/2)..d/2\} \times \{-pi..pi\}) =$ *ennreal* $(2 * d * pi)$
using *d* **by** (*simp flip*: *lborel-prod ennreal-mult add*: *lborel.emeasure-pair-measure-Times*)
finally show *?thesis* **.**
qed

lemma *emeasure-buffon-set-conv-buffon-set′*:
emeasure lborel Buffon-set = *4* ∗ *emeasure lborel Buffon-set′*
proof −
have *distr-lborel* [*simp*]: *distr M lborel f* = *distr M borel f* **for** *M* **and** *f* :: *real* ⇒ *real*
by (*rule distr-cong*) *simp-all*

define *A* **where** *A* = *Buffon-set′*
define *B C D* **where** *B* = (λ*x*. (−*fst x*, *snd x*)) −' *A* **and** *C* = (λ*x*. (*fst x*, −*snd x*)) −' *A* **and**
D = (λ*x*. (−*fst x*, −*snd x*)) −' *A*
have *meas* [*measurable*]:
(λ*x*::*real* × *real*. (−*fst x*, *snd x*)) ∈ *borel-measurable borel*
(λ*x*::*real* × *real*. (*fst x*, −*snd x*)) ∈ *borel-measurable borel*
(λ*x*::*real* × *real*. (−*fst x*, −*snd x*)) ∈ *borel-measurable borel*
unfolding *borel-prod* [*symmetric*] **by** *measurable*
have *meas′* [*measurable*]: *A* ∈ *sets borel B* ∈ *sets borel C* ∈ *sets borel D* ∈ *sets borel*
unfolding *A-def B-def C-def D-def* **by** (*rule measurable-buffon-set′ measurable-sets-borel meas*)+

have ∗: *Buffon-set* = *A* ∪ *B* ∪ *C* ∪ *D*
proof (*intro equalityI subsetI*, *goal-cases*)
case (*1 z*)
show *?case*
proof (*cases fst z* ≥ *0*; *cases snd z* ≥ *0*)
assume *fst z* ≥ *0 snd z* ≥ *0*
with *1* **have** *z* ∈ *A*
by (*auto split*: *prod.splits simp*: *Buffon-set-altdef2 Buffon-set′-def sin-ge-zero A-def B-def*)
thus *?thesis* **by** *blast*
next
assume ¬(*fst z* ≥ *0*) *snd z* ≥ *0*
with *1* **have** *z* ∈ *B*
by (*auto split*: *prod.splits simp*: *Buffon-set-altdef2 Buffon-set′-def sin-ge-zero*

A-def B-def)
    thus *?thesis* **by** *blast*
  next
    assume *fst z* $\geq$ *0* $\neg$(*snd z* $\geq$ *0*)
    with *1* **have** *z* $\in$ *C*
     by (*auto split*: *prod.splits simp*: *Buffon-set-altdef2 Buffon-set′-def sin-le-zero′ A-def B-def C-def*)
    thus *?thesis* **by** *blast*
  next
    assume $\neg$(*fst z* $\geq$ *0*) $\neg$(*snd z* $\geq$ *0*)
    with *1* **have** *z* $\in$ *D*
     by (*auto split*: *prod.splits simp*: *Buffon-set-altdef2 Buffon-set′-def sin-le-zero′ A-def B-def D-def*)
    thus *?thesis* **by** *blast*
  qed
next
  case (*2 z*)
  thus *?case* **using** *d l*
   by (*auto simp*: *Buffon-set-altdef2 Buffon-set′-def sin-ge-zero sin-le-zero′ A-def B-def C-def D-def*)
qed

have $A \cap B = \{0\} \times (\{0..pi\} \cap \{\varphi.\ sin\ \varphi * l - d \geq 0\})$
  using *d l* **by** (*auto simp*: *Buffon-set′-def A-def B-def C-def D-def*)
moreover have *emeasure lborel* ... = *0*
 unfolding *lborel-prod* [*symmetric*] **by** (*subst lborel.emeasure-pair-measure-Times*) *simp-all*
ultimately have *AB*: $(A \cap B) \in$ *null-sets lborel*
  unfolding *lborel-prod* [*symmetric*] **by** (*simp add*: *null-sets-def*)

have $C \cap D = \{0\} \times (\{-pi..0\} \cap \{\varphi.\ -sin\ \varphi * l - d \geq 0\})$
  using *d l* **by** (*auto simp*: *Buffon-set′-def A-def B-def C-def D-def*)
moreover have *emeasure lborel* ... = *0*
 unfolding *lborel-prod* [*symmetric*] **by** (*subst lborel.emeasure-pair-measure-Times*) *simp-all*
ultimately have *CD*: $(C \cap D) \in$ *null-sets lborel*
  unfolding *lborel-prod* [*symmetric*] **by** (*simp add*: *null-sets-def*)

have $A \cap D = \{\}$ $B \cap C = \{\}$ **using** *d l*
  by (*auto simp*: *Buffon-set′-def A-def D-def B-def C-def*)
moreover have $A \cap C = \{(d/2,\ 0)\}$ $B \cap D = \{(-d/2,\ 0)\}$
   using *d l* **by** (*auto simp*: *case-prod-unfold Buffon-set′-def A-def B-def C-def D-def*)
ultimately have *AD*: $A \cap D \in$ *null-sets lborel* **and** *BC*: $B \cap C \in$ *null-sets lborel* **and**
  AC: $A \cap C \in$ *null-sets lborel* **and** *BD*: $B \cap D \in$ *null-sets lborel* **by** *auto*

note $*$
also have *emeasure lborel* $(A \cup B \cup C \cup D)$ = *emeasure lborel* $(A \cup B \cup C)$ +

emeasure lborel D
using *AB AC AD BC BD CD* **by** (*intro emeasure-Un′*) (*auto simp*: *Int-Un-distrib2*)
also have *emeasure lborel* (*A* ∪ *B* ∪ *C*) = *emeasure lborel* (*A* ∪ *B*) + *emeasure lborel C*
using *AB AC BC* **using** *AB AC AD BC BD CD* **by** (*intro emeasure-Un′*) (*auto simp*: *Int-Un-distrib2*)
also have *emeasure lborel* (*A* ∪ *B*) = *emeasure lborel A* + *emeasure lborel B*
using *AB* **using** *AB AC AD BC BD CD* **by** (*intro emeasure-Un′*) (*auto simp*: *Int-Un-distrib2*)
also have *emeasure lborel B* = *emeasure* (*distr lborel lborel* (λ(*x*,*y*). (−*x*, *y*))) *A*
(**is** - = *emeasure ?M* -) **unfolding** *B-def*
by (*subst emeasure-distr*) (*simp-all add*: *case-prod-unfold*)
also have *?M* = *lborel* **unfolding** *lborel-prod* [*symmetric*]
by (*subst pair-measure-distr* [*symmetric*]) (*simp-all add*: *sigma-finite-lborel lborel-distr-uminus*)
also have *emeasure lborel C* = *emeasure* (*distr lborel lborel* (λ(*x*,*y*). (*x*, −*y*))) *A*
(**is** - = *emeasure ?M* -) **unfolding** *C-def*
by (*subst emeasure-distr*) (*simp-all add*: *case-prod-unfold*)
also have *?M* = *lborel* **unfolding** *lborel-prod* [*symmetric*]
by (*subst pair-measure-distr* [*symmetric*]) (*simp-all add*: *sigma-finite-lborel lborel-distr-uminus*)
also have *emeasure lborel D* = *emeasure* (*distr lborel lborel* (λ(*x*,*y*). (−*x*, −*y*))) *A*
(**is** - = *emeasure ?M* -) **unfolding** *D-def*
by (*subst emeasure-distr*) (*simp-all add*: *case-prod-unfold*)
also have *?M* = *lborel* **unfolding** *lborel-prod* [*symmetric*]
by (*subst pair-measure-distr* [*symmetric*]) (*simp-all add*: *sigma-finite-lborel lborel-distr-uminus*)
finally have *emeasure lborel Buffon-set* =
of-nat (*Suc* (*Suc* (*Suc* (*Suc 0*)))) ∗ *emeasure lborel A*
unfolding *of-nat-Suc ring-distribs* **by** *simp*
also have *of-nat* (*Suc* (*Suc* (*Suc* (*Suc 0*)))) = (*4* :: *ennreal*) **by** *simp*
finally show *?thesis* **unfolding** *A-def* **.**
qed

It only remains now to compute the measure of *Buffon-set′*. We first reduce this problem to a relatively simple integral:

lemma *emeasure-buffon-set′*:
emeasure lborel Buffon-set′ =
ennreal (*integral* {*0*..*pi*} (λ*x*. *min* (*d* / *2*) (*sin x* ∗ *l* / *2*)))
(**is** *emeasure lborel ?A* = -)
proof −
have *emeasure lborel ?A* = *nn-integral lborel* (λ*x*. *indicator ?A x*)
by (*intro nn-integral-indicator* [*symmetric*]) *simp-all*
also have (*lborel* :: (*real* × *real*) *measure*) = *lborel* $\bigotimes_M$ *lborel*
by (*simp only*: *lborel-prod*)
also have *nn-integral* ... (*indicator ?A*) = (∫⁺φ. ∫⁺*x*. *indicator ?A* (*x*, φ) ∂*lborel* ∂*lborel*)
by (*subst lborel-pair.nn-integral-snd* [*symmetric*]) (*simp-all add*: *lborel-prod*

borel-prod)
  also have ... = ($\int^+\varphi$. $\int^+x$. *indicator* {*0..pi*} φ ∗ *indicator* {*max 0* (*d/2* − *sin* φ ∗ *l* / *2*) .. *d/2*} *x* ∂*lborel* ∂*lborel*)
    using *d l* **by** (*intro nn-integral-cong*) (*auto simp*: *indicator-def field-simps Buffon-set′-def*)
  also have ... = $\int^+$ φ. *indicator* {*0..pi*} φ ∗ *emeasure lborel* {*max 0* (*d* / *2* − *sin* φ ∗ *l* / *2*)..*d* / *2*} ∂*lborel*
    by (*subst nn-integral-cmult*) *simp-all*
  also have ... = $\int^+$ φ. *ennreal* (*indicator* {*0..pi*} φ ∗ *min* (*d* / *2*) (*sin* φ ∗ *l* / *2*)) ∂*lborel*
    (**is** - = *?I*) **using** *d l* **by** (*intro nn-integral-cong*) (*auto simp*: *indicator-def sin-ge-zero max-def min-def*)
  also have *integrable lborel* (λφ. (*d* / *2*) ∗ *indicator* {*0..pi*} φ) **by** *simp*
  hence *int*: *integrable lborel* (λφ. *indicator* {*0..pi*} φ ∗ *min* (*d* / *2*) (*sin* φ ∗ *l* / *2*))
    by (*rule Bochner-Integration.integrable-bound*)
       (*insert l d*, *auto intro*!: *AE-I2 simp*: *indicator-def min-def sin-ge-zero*)
  hence *?I* = *set-lebesgue-integral lborel* {*0..pi*} (λφ. *min* (*d* / *2*) (*sin* φ ∗ *l* / *2*))
    by (*subst nn-integral-eq-integral*, *assumption*)
     (*insert d l*, *auto intro*!: *AE-I2 simp*: *sin-ge-zero min-def indicator-def set-lebesgue-integral-def*)
  also have ... = *ennreal* (*integral* {*0..pi*} (λ*x*. *min* (*d* / *2*) (*sin x* ∗ *l* / *2*)))
    (**is** - = *ennreal ?I*) **using** *int* **by** (*subst set-borel-integral-eq-integral*) (*simp-all add*: *set-integrable-def*)
  finally show *?thesis* **by** (*simp add*: *lborel-prod*)
qed

We now have to distinguish two cases: The first and easier one is that where the length of the needle, l, is less than or equal to the strip width, d:

context
  assumes *l-le-d*: *l* ≤ *d*
begin

lemma *emeasure-buffon-set′-short*: *emeasure lborel Buffon-set′* = *ennreal l*
proof −
  have *emeasure lborel Buffon-set′* =
          ennreal (*integral* {*0..pi*} (λ*x*. *min* (*d* / *2*) (*sin x* ∗ *l* / *2*))) (**is** - = *ennreal ?I*)
    by (*rule emeasure-buffon-set′*)
  also have ∗: *sin* φ ∗ *l* ≤ *d* **if** φ ≥ *0* φ ≤ *pi* **for** φ
    using *mult-mono*[*OF l-le-d sin-le-one - sin-ge-zero*] *that d* **by** (*simp add*: *algebra-simps*)
  have *?I* = *integral* {*0..pi*} (λ*x*. (*l* / *2*) ∗ *sin x*)
    using *l d l-le-d*
    by (*intro integral-cong*) (*auto dest*: ∗ *simp*: *min-def sin-ge-zero*)
  also have ... = *l* / *2* ∗ *integral* {*0..pi*} *sin* **by** *simp*
  also have (*sin has-integral* (−*cos pi* − (− *cos 0*))) {*0..pi*}
    by (*intro fundamental-theorem-of-calculus*)
     (*auto intro*!: *derivative-eq-intros simp*: *has-real-derivative-iff-has-vector-derivative* [*symmetric*])

hence *integral* $\{0..pi\}$ *sin* $= -$*cos pi* $-$ ($-$*cos 0*)
by (*simp add*: *has-integral-iff*)
finally show *?thesis* **by** (*simp add*: *lborel-prod*)
qed

lemma *emeasure-buffon-set-short*: *emeasure lborel Buffon-set* = *4* $*$ *ennreal l*
by (*simp add*: *emeasure-buffon-set-conv-buffon-set′ emeasure-buffon-set′-short l-le-d*)

lemma *prob-short-aux*:
Buffon $\{(x, \varphi)$. *needle l x* $\varphi \cap \{-\ d\ /\ 2,\ d\ /\ 2\} \neq \{\}\}$ = *ennreal* (*2* $*$ *l* / (*d* $*$ *pi*))
unfolding *buffon-prob-aux emeasure-buffon-set-short* **using** *d l*
by (*simp flip*: *ennreal-mult ennreal-numeral add*: *divide-ennreal*)

lemma *prob-short*: $\mathcal{P}((x,\varphi)$ *in Buffon*. *needle l x* $\varphi \cap \{-d/2,\ d/2\} \neq \{\}) = 2 * l\ /\ (d * pi)$
using *prob-short-aux* **unfolding** *emeasure-eq-measure*
using *l d* **by** (*subst* (*asm*) *ennreal-inj*) *auto*

end

The other case where the needle is at least as long as the strip width is more complicated:

context
assumes *l-ge-d*: $l \geq d$
begin

lemma *emeasure-buffon-set′-long*:
shows $l * (1 - sqrt\ (1 - (d\ /\ l)^2)) + arccos\ (d\ /\ l) * d \geq 0$
and *emeasure lborel Buffon-set′* =
ennreal $(l * (1 - sqrt\ (1 - (d\ /\ l)^2)) + arccos\ (d\ /\ l) * d)$
proof $-$
define φ' **where** $\varphi' = arcsin\ (d\ /\ l)$
have φ'*-nonneg*: $\varphi' \geq 0$ **unfolding** φ'*-def* **using** *d l l-ge-d arcsin-le-mono*[*of 0 d/l*]
by (*simp add*: φ'*-def*)
have φ'*-le*: $\varphi' \leq pi\ /\ 2$ **unfolding** φ'*-def* **using** *arcsin-bounded*[*of d/l*] *d l l-ge-d*
by (*simp add*: *field-simps*)
have *ge-phi′*: $sin\ \varphi \geq d\ /\ l$ **if** $\varphi \geq \varphi'\ \varphi \leq pi\ /\ 2$ **for** φ
using *arcsin-le-iff*[*of d / l* φ] *d l-ge-d that* φ'*-nonneg* **by** (*auto simp*: φ'*-def field-simps*)
have *le-phi′*: $sin\ \varphi \leq d\ /\ l$ **if** $\varphi \leq \varphi'\ \varphi \geq 0$ **for** φ
using *le-arcsin-iff*[*of d / l* φ] *d l-ge-d that* φ'*-le* **by** (*auto simp*: φ'*-def field-simps*)
have $cos\ \varphi' = sqrt\ (1 - (d\ /\ l)\hat{\ }2)$
unfolding φ'*-def* **by** (*rule cos-arcsin*) (*insert d l l-ge-d*, *auto simp*: *field-simps*)

have $l * (1 - cos\ \varphi') + arccos\ (d\ /\ l) * d \geq 0$
using *l d l-ge-d*

by (*intro add-nonneg-nonneg mult-nonneg-nonneg arccos-lbound*) (*auto simp*: *field-simps*)
thus *l* ∗ (*1* − *sqrt* (*1* − (*d* / *l*)^2)) + *arccos* (*d* / *l*) ∗ *d* ≥ *0*
by (*simp add*: ‹*cos* φ′ = *sqrt* (*1* − (*d* / *l*)ˆ*2*)›)

let *?f* = (λ*x*. *min* (*d* / *2*) (*sin x* ∗ *l* / *2*))
have *emeasure lborel Buffon-set*′ = *ennreal* (*integral* {*0*..*pi*} *?f*) (**is** - = *ennreal* *?I*)
by (*rule emeasure-buffon-set*′)
also have *?I* = *integral* {*0*..*pi*/*2*} *?f* + *integral* {*pi*/*2*..*pi*} *?f*
by (*rule Henstock-Kurzweil-Integration.integral-combine* [*symmetric*]) (*auto intro*!: *integrable-continuous-real continuous-intros*)
also have *integral* {*pi*/*2*..*pi*} *?f* = *integral* {−*pi*/*2*..*0*} (*?f* ∘ (λφ. φ + *pi*))
by (*subst integral-shift*) (*auto intro*!: *continuous-intros*)
also have ... = *integral* {−(*pi*/*2*)..−*0*} (λ*x*. *min* (*d* / *2*) (*sin* (−*x*) ∗ *l* / *2*))
by (*simp add*: *o-def*)
also have ... = *integral* {*0*..*pi*/*2*} *?f* (**is** - = *?I*) **by** (*subst Henstock-Kurzweil-Integration.integral-reflect-real*) *simp-all*
also have ... + ... = *2* ∗ ... **by** *simp*
also have *?I* = *integral* {*0*..φ′} *?f* + *integral* {φ′..*pi*/*2*} *?f*
using *l d l-ge-d* φ′*-nonneg* φ′*-le*
by (*intro Henstock-Kurzweil-Integration.integral-combine* [*symmetric*]) (*auto intro*!: *integrable-continuous-real continuous-intros*)
also have *integral* {*0*..φ′} *?f* = *integral* {*0*..φ′} (λ*x*. *l* / *2* ∗ *sin x*)
using *l* **by** (*intro integral-cong*) (*auto simp*: *min-def field-simps dest*: *le-phi*′)
also have ((λ*x*. *l* / *2* ∗ *sin x*) *has-integral* (− (*l* / *2* ∗ *cos* φ′) − (− (*l* / *2* ∗ *cos* *0*)))) {*0*..φ′}
using φ′*-nonneg*
by (*intro fundamental-theorem-of-calculus*)
(*auto simp*: *has-real-derivative-iff-has-vector-derivative* [*symmetric*] *intro*!: *derivative-eq-intros*)
hence *integral* {*0*..φ′} (λ*x*. *l* / *2* ∗ *sin x*) = (*1* − *cos* φ′) ∗ *l* / *2*
by (*simp add*: *has-integral-iff algebra-simps*)
also have *integral* {φ′..*pi*/*2*} *?f* = *integral* {φ′..*pi*/*2*} (λ-. *d* / *2*)
using *l* **by** (*intro integral-cong*) (*auto simp*: *min-def field-simps dest*: *ge-phi*′)
also have ... = *arccos* (*d* / *l*) ∗ *d* / *2* **using** φ′*-le d l l-ge-d*
by (*subst arccos-arcsin-eq*) (*auto simp*: *field-simps* φ′*-def*)
also note ‹*cos* φ′ = *sqrt* (*1* − (*d* / *l*)ˆ*2*)›
also have *2* ∗ ((*1* − *sqrt* (*1* − (*d* / *l*)^2)) ∗ *l* / *2* + *arccos* (*d* / *l*) ∗ *d* / *2*) =
l ∗ (*1* − *sqrt* (*1* − (*d* / *l*)^2)) + *arccos* (*d* / *l*) ∗ *d*
using *d l* **by** (*simp add*: *field-simps*)
finally show *emeasure lborel Buffon-set*′ =
ennreal (*l* ∗ (*1* − *sqrt* (*1* − (*d* / *l*)^2)) + *arccos* (*d* / *l*) ∗ *d*) **.**
qed

lemma *emeasure-set-long*: *emeasure lborel Buffon-set* =
4 ∗ *ennreal* (*l* ∗ (*1* − *sqrt* (*1* − (*d* / *l*)^2)) + *arccos* (*d* / *l*) ∗ *d*)
by (*simp add*: *emeasure-buffon-set-conv-buffon-set*′ *emeasure-buffon-set*′*-long l-ge-d*)

lemma *prob-long-aux*:
  shows $2 \ / \ pi * ((l \ / \ d) - sqrt \ ((l \ / \ d)^2 - 1) + arccos \ (d \ / \ l)) \geq 0$
  and *Buffon* $\{(x, \varphi). \ needle \ l \ x \ \varphi \cap \{- \ d \ / \ 2, \ d \ / \ 2\} \neq \{\}\} =$
    $ennreal \ (2 \ / \ pi * ((l \ / \ d) - sqrt \ ((l \ / \ d)^2 - 1) + arccos \ (d \ / \ l)))$
  using *emeasure-buffon-set′-long*(*1*)
proof −
  have ∗: $l * sqrt \ ((l^2 - d^2) \ / \ l^2) + 0 \leq l + d * arccos \ (d \ / \ l)$
    using *d l-ge-d* **by** (*intro add-mono mult-nonneg-nonneg arccos-lbound*) (*auto simp*: *field-simps*)

  have $l \ / \ d \geq sqrt \ ((l \ / \ d)^2 - 1)$
    using *l d l-ge-d* **by** (*intro real-le-lsqrt*) (*auto simp*: *field-simps*)
  thus $2 \ / \ pi * ((l \ / \ d) - sqrt \ ((l \ / \ d)^2 - 1) + arccos \ (d \ / \ l)) \geq 0$
    using *d l l-ge-d*
    by (*intro mult-nonneg-nonneg add-nonneg-nonneg arccos-lbound*) (*auto simp*: *field-simps*)

  have *emeasure Buffon* $\{(x,\varphi). \ needle \ l \ x \ \varphi \cap \{-d/2, \ d/2\} \neq \{\}\} =$
    $ennreal \ (4 * (l - l * sqrt \ (1 - (d \ / \ l)^2) + arccos \ (d \ / \ l) * d)) \ / \ ennreal \ (2 * d * pi)$
    using *d l l-ge-d* ∗ **unfolding** *buffon-prob-aux emeasure-set-long ennreal-numeral* [*symmetric*]
    by (*subst ennreal-mult* [*symmetric*])
      (*auto intro*!: *add-nonneg-nonneg mult-nonneg-nonneg simp*: *field-simps*)
  also have $\ldots = ennreal \ ((4 * (l - l * sqrt \ (1 - (d \ / \ l)^2) + arccos \ (d \ / \ l) * d)) \ / \ (2 * d * pi))$
    using *d l* ∗ **by** (*subst divide-ennreal*) (*auto simp*: *field-simps*)
  also have $(4 * (l - l * sqrt \ (1 - (d \ / \ l)^2) + arccos \ (d \ / \ l) * d)) \ / \ (2 * d * pi) =$
    $2 \ / \ pi * (l \ / \ d - l \ / \ d * sqrt \ ((d \ / \ l)\hat{\ }2 * ((l \ / \ d)\hat{\ }2 - 1)) + arccos \ (d \ / \ l))$
    using *d l* **by** (*simp add*: *field-simps*)
  also have $l \ / \ d * sqrt \ ((d \ / \ l)\hat{\ }2 * ((l \ / \ d)\hat{\ }2 - 1)) = sqrt \ ((l \ / \ d) \ \hat{\ } \ 2 - 1)$
    using *d l l-ge-d* **unfolding** *real-sqrt-mult real-sqrt-abs* **by** *simp*
  finally show *emeasure Buffon* $\{(x,\varphi). \ needle \ l \ x \ \varphi \cap \{-d/2, \ d/2\} \neq \{\}\} =$
    $ennreal \ (2 \ / \ pi * ((l \ / \ d) - sqrt \ ((l \ / \ d)^2 - 1) + arccos \ (d \ / \ l)))$ **.**
qed

lemma *prob-long*:
  $\mathcal{P}((x,\varphi) \ in \ Buffon. \ needle \ l \ x \ \varphi \cap \{-d/2, \ d/2\} \neq \{\}) =$
    $2 \ / \ pi * ((l \ / \ d) - sqrt \ ((l \ / \ d)^2 - 1) + arccos \ (d \ / \ l))$
  using *prob-long-aux* **unfolding** *emeasure-eq-measure*
  by (*subst* (*asm*) *ennreal-inj*) *simp-all*

end

theorem *prob-eq*:
  defines $x \equiv l \ / \ d$
  shows $\mathcal{P}((x,\varphi) \ in \ Buffon. \ needle \ l \ x \ \varphi \cap \{-d/2, \ d/2\} \neq \{\}) =$

```
      (if l ≤ d then
         2 / pi * x
       else
         2 / pi * (x − sqrt (x² − 1) + arccos (1 / x)))
  using prob-short prob-long unfolding x-def by auto

end

end
```

References

[1] J. F. Ramaley. Buffon's Noodle Problem. *The American Mathematical Monthly*, 76(8):916–918, 1969.

[2] E. W. Weisstein. MathWorld – Buffon's Needle Problem. http://mathworld.wolfram.com/BuffonsNeedleProblem.html.