Ptolemy's Theorem

Lukas Bulwahn

March 17, 2025

Abstract

This entry provides an analytic proof to Ptolemy's Theorem using polar form transformation and trigonometric identities. In this formalization, we use ideas from John Harrison's HOL Light formalization [1] and the proof sketch on the Wikipedia entry of Ptolemy's Theorem [3]. This theorem is the 95th theorem of the Top 100 Theorems list [2].

Contents

1 Ptolemy's Theorem

theory *Ptolemys-Theorem*
imports
HOL−Analysis.Multivariate-Analysis
begin

1.1 Preliminaries

1.1.1 Additions to Rat theory

hide-const (**open**) *normalize*

1.1.2 Additions to Transcendental theory

Lemmas about *arcsin* and *arccos* commonly involve to show that their argument is in the domain of those partial functions, i.e., the argument y is between $-\ 1$ and 1. As the argumentation for $-\ 1 \leq y$ and $y \leq 1$ is often very similar, we prefer to prove $|y| \leq 1$ to the two goals above.

The lemma for rewriting the term *cos* (*arccos* *y*) is already provided in the Isabelle distribution with name *cos-arccos-abs*. Here, we further provide the analogue on *arcsin* for rewriting *sin* (*arcsin* *y*).

```
lemma sin-arcsin-abs: |y| ≤ 1 ⟹ sin (arcsin y) = y
  by (simp add: abs-le-iff)
```

The further lemmas are the required variants from existing lemmas *arccos-lbound* and *arccos-ubound*.

```
lemma arccos-lbound-abs [simp]:
  |y| ≤ 1 ⟹ 0 ≤ arccos y
by (simp add: arccos-lbound)

lemma arccos-ubound-abs [simp]:
  |y| ≤ 1 ⟹ arccos y ≤ pi
by (simp add: arccos-ubound)
```

As we choose angles to be between 0 between $2 * pi$, we need some lemmas to reason about the sign of *sin x* for angles *x*.

```
lemma sin-ge-zero-iff:
  assumes 0 ≤ x x < 2 * pi
  shows 0 ≤ sin x ⟷ x ≤ pi
proof
  assume 0 ≤ sin x
  show x ≤ pi
  proof (rule ccontr)
    assume ¬ x ≤ pi
    from this ‹x < 2 * pi› have sin x < 0
      using sin-lt-zero by auto
    from this ‹0 ≤ sin x› show False by auto
  qed
next
  assume x ≤ pi
  from this ‹0 ≤ x› show 0 ≤ sin x by (simp add: sin-ge-zero)
qed

lemma sin-less-zero-iff:
  assumes 0 ≤ x x < 2 * pi
  shows sin x < 0 ⟷ pi < x
using assms sin-ge-zero-iff by fastforce
```

1.1.3 Addition to Finite-Cartesian-Product theory

Here follow generally useful additions and specialised equations for two-dimensional real-valued vectors.

lemma *axis-nth-eq-0* [*simp*]:
  assumes $i \neq j$
  shows *axis i x* \$ $j = 0$
using *assms* **unfolding** *axis-def* **by** *simp*

lemma *norm-axis*:
  fixes x :: *real*
  shows *norm* (*axis i x*) = *abs x*
by (*simp add*: *norm-eq-sqrt-inner inner-axis-axis*)

lemma *norm-eq-on-real-2-vec*:
  fixes x :: *real* ^ *2*
  shows *norm x* = *sqrt* ((*x* \$ *1*) ^ *2* + (*x* \$ *2*) ^ *2*)
by (*simp add*: *norm-eq-sqrt-inner inner-vec-def UNIV-2 power2-eq-square*)

lemma *dist-eq-on-real-2-vec*:
  fixes *a b* :: *real* ^ *2*
  shows *dist a b* = *sqrt* ((*a* \$ *1* − *b* \$ *1*) ^ *2* + (*a* \$ *2* − *b* \$ *2*) ^ *2*)
unfolding *dist-norm norm-eq-on-real-2-vec* **by** *simp*

1.2 Polar Form of Two-Dimensional Real-Valued Vectors

1.2.1 Definitions to Transfer to Polar Form and Back

definition *of-radiant* :: *real* ⇒ *real* ^ *2*
where
  of-radiant ω = *axis 1* (*cos* ω) + *axis 2* (*sin* ω)

definition *normalize* :: *real* ^ *2* ⇒ *real* ^ *2*
where
  normalize p = (*if p* = *0 then axis 1 1 else* (*1* / *norm p*) $*_R$ *p*)

definition *radiant-of* :: *real* ^ *2* ⇒ *real*
where
  radiant-of p = (*THE* ω. $0 \leq \omega \wedge \omega < 2 * pi$ ∧ *of-radiant* ω = *normalize p*)

The vector *of-radiant* ω is the vector with length *1* and angle ω to the first axis. We normalize vectors to length *1* keeping their orientation with the normalize function. Conversely, *radiant-of p* is the angle of vector *p* to the first axis, where we choose *radiant-of* to return angles between *0* and *2* ∗ *pi*, following the usual high-school convention. With these definitions, we can express the main result *norm p* $*_R$ *of-radiant* (*radiant-of p*) = *p*. Note that the main result holds for any definition of *radiant-of 0*. So, we choose to define *normalize 0* and *radiant-of 0*, such that *radiant-of 0* = *0*.

1.2.2 Lemmas on *of-radiant*

lemma *nth-of-radiant-1* [*simp*]:
  of-radiant ω $ *1* = *cos* ω
unfolding *of-radiant-def* **by** *simp*

lemma *nth-of-radiant-2* [*simp*]:
  of-radiant ω $ *2* = *sin* ω
unfolding *of-radiant-def* **by** *simp*

lemma *norm-of-radiant*:
  norm (*of-radiant* ω) = *1*
unfolding *of-radiant-def norm-eq-on-real-2-vec* **by** *simp*

lemma *of-radiant-plus-2pi*:
  of-radiant (ω + *2* ∗ *pi*) = *of-radiant* ω
unfolding *of-radiant-def* **by** *simp*

lemma *of-radiant-minus-2pi*:
  of-radiant (ω − *2* ∗ *pi*) = *of-radiant* ω
proof −
  have *of-radiant* (ω − *2* ∗ *pi*) = *of-radiant* (ω − *2* ∗ *pi* + *2* ∗ *pi*)
    by (*simp only*: *of-radiant-plus-2pi*)
  also have ... = *of-radiant* ω **by** *simp*
  finally show *?thesis* **.**
qed

1.2.3 Lemmas on *normalize*

lemma *normalize-eq*:
  norm p $*_R$ *normalize p* = *p*
unfolding *normalize-def* **by** *simp*

lemma *norm-normalize*:
  norm (*normalize p*) = *1*
unfolding *normalize-def* **by** (*auto simp add*: *norm-axis*)

lemma *nth-normalize* [*simp*]:
  |*normalize p* $ *i*| ≤ *1*
using *norm-normalize component-le-norm-cart* **by** *metis*

lemma *normalize-square*:
  (*normalize p* $ *1*)^2 + (*normalize p* $ *2*)^2 = *1*
using *dot-square-norm*[*of normalize p*]
by (*simp add*: *inner-vec-def UNIV-2 power2-eq-square norm-normalize*)

lemma *nth-normalize-ge-zero-iff*:
  0 ≤ *normalize p* $ *i* ⟷ *0* ≤ *p* $ *i*
proof
  assume *0* ≤ *normalize p* $ *i*

from *this* **show** *0* ≤ *p* $ *i*
unfolding *normalize-def* **by** (*auto split*: *if-split-asm simp add*: *zero-le-divide-iff*)
next
assume *0* ≤ *p* $ *i*
have *0* ≤ *axis 1* (*1* :: *real*) $ *i*
using *exhaust-2*[*of i*] **by** *auto*
from *this* ‹*0* ≤ *p* $ *i*› **show** *0* ≤ *normalize p* $ *i*
unfolding *normalize-def* **by** *auto*
qed

lemma *nth-normalize-less-zero-iff*:
normalize p $ *i* < *0* ⟷ *p* $ *i* < *0*
using *nth-normalize-ge-zero-iff leD leI* **by** *metis*

lemma *normalize-boundary-iff*:
|*normalize p* $ *1*| = *1* ⟷ *p* $ *2* = *0*
proof
assume |*normalize p* $ *1*| = *1*
from *this* **have** *1*: (*p* $ *1*) ^ *2* = *norm p* ^ *2*
unfolding *normalize-def* **by** (*auto split*: *if-split-asm simp add*: *power2-eq-iff*)
moreover have (*p* $ *1*) ^ *2* + (*p* $ *2*) ^ *2* = *norm p* ^ *2*
using *norm-eq-on-real-2-vec* **by** *auto*
ultimately show *p* $ *2* = *0* **by** *simp*
next
assume *p* $ *2* = *0*
from *this* **have** |*p* $ *1*| = *norm p*
by (*auto simp add*: *norm-eq-on-real-2-vec*)
from *this* **show** |*normalize p* $ *1*| = *1*
unfolding *normalize-def* **by** *simp*
qed

lemma *between-normalize-if-distant-from-0*:
assumes *norm p* ≥ *1*
shows *between* (*0*, *p*) (*normalize p*)
using *assms* **by** (*auto simp add*: *between-mem-segment closed-segment-def normalize-def*)

lemma *between-normalize-if-near-0*:
assumes *norm p* ≤ *1*
shows *between* (*0*, *normalize p*) *p*
proof −
have *0* ≤ *norm p* **by** *simp*
from *assms* **have** *p* = (*norm p* / *norm p*) $*_R$ *p* ∧ *0* ≤ *norm p* ∧ *norm p* ≤ *1* **by** *auto*
from *this* **have** ∃ *u*. *p* = (*u* / *norm p*) $*_R$ *p* ∧ *0* ≤ *u* ∧ *u* ≤ *1* **by** *blast*
from *this* **show** *?thesis*
by (*auto simp add*: *between-mem-segment closed-segment-def normalize-def*)
qed

1.2.4 Lemmas on *radiant-of*

```
lemma radiant-of:
  0 ≤ radiant-of p ∧ radiant-of p < 2 ∗ pi ∧ of-radiant (radiant-of p) = normalize
p
proof −
  let ?a = if 0 ≤ p $ 2 then arccos (normalize p $ 1) else pi + arccos (− (normalize
p $ 1))
  have 0 ≤ ?a ∧ ?a < 2 ∗ pi ∧ of-radiant ?a = normalize p
  proof −
    have 0 ≤ ?a by auto
    moreover have ?a < 2 ∗ pi
    proof cases
      assume 0 ≤ p $ 2
      from this have ?a ≤ pi by simp
      from this show ?thesis
        using pi-gt-zero by linarith
    next
      assume ¬ 0 ≤ p $ 2
      have arccos (− normalize p $ 1) < pi
      proof −
        have |normalize p $ 1| ≠ 1
          using ‹¬ 0 ≤ p $ 2› by (simp only: normalize-boundary-iff)
        from this have arccos (− normalize p $ 1) ≠ pi
          unfolding arccos-minus-1[symmetric] by (subst arccos-eq-iff) auto
        moreover have arccos (− normalize p $ 1) ≤ pi by simp
        ultimately show arccos (− normalize p $ 1) < pi by linarith
      qed
      from this ‹¬ 0 ≤ p $ 2› show ?thesis by simp
    qed
    moreover have of-radiant ?a = normalize p
    proof −
      have of-radiant ?a $ i = normalize p $ i for i
      proof −
        have of-radiant ?a $ 1 = normalize p $ 1
          unfolding of-radiant-def by (simp add: cos-arccos-abs)
        moreover have of-radiant ?a $ 2 = normalize p $ 2
        proof cases
          assume 0 ≤ p $ 2
          have sin (arccos (normalize p $ 1)) = sqrt (1 − (normalize p $ 1) ^ 2)
            by (simp add: sin-arccos-abs)
          also have . . . = normalize p $ 2
          proof −
            have 1 − (normalize p $ 1)² = (normalize p $ 2)²
              using normalize-square[of p] by auto
         from this ‹0 ≤ p $ 2› show ?thesis by (simp add: nth-normalize-ge-zero-iff)
          qed
          finally show ?thesis
            using ‹0 ≤ p $ 2› unfolding of-radiant-def by auto
        next
```

```
        assume ¬ 0 ≤ p $ 2
        have − sin (arccos (− normalize p $ 1)) = − sqrt (1 − (normalize p $ 1)^2)
          by (simp add: sin-arccos-abs)
        also have . . . = normalize p $ 2
        proof −
          have 1 − (normalize p $ 1)^2 = (normalize p $ 2)^2
            using normalize-square[of p] by auto
          from this ‹¬ 0 ≤ p $ 2› show ?thesis
            using nth-normalize-ge-zero-iff by fastforce
        qed
        finally show ?thesis
          using ‹¬ 0 ≤ p $ 2› unfolding of-radiant-def by auto
      qed
      ultimately show ?thesis by (metis exhaust-2[of i])
    qed
    from this show ?thesis by (simp add: vec-eq-iff)
  qed
  ultimately show ?thesis by blast
qed
moreover {
  fix ω
  assume 0 ≤ ω ∧ ω < 2 ∗ pi ∧ of-radiant ω = normalize p
  from this have 0 ≤ ω ω < 2 ∗ pi normalize p = of-radiant ω by auto
  from this have cos ω = normalize p $ 1 sin ω = normalize p $ 2 by auto
  have ω = ?a
  proof cases
    assume 0 ≤ p $ 2
    from this have ω ≤ pi
      using ‹0 ≤ ω› ‹ω < 2 ∗ pi› ‹sin ω = normalize p $ 2›
      by (simp add: sin-ge-zero-iff[symmetric] nth-normalize-ge-zero-iff)
    from ‹0 ≤ ω› this have ω = arccos (cos ω) by (simp add: arccos-cos)
    from ‹cos ω = normalize p $ 1› this have ω = arccos (normalize p $ 1)
      by (simp add: arccos-eq-iff)
    from this show ω = ?a using ‹0 ≤ p $ 2› by auto
  next
    assume ¬ 0 ≤ p $ 2
    from this have ω > pi
      using ‹0 ≤ ω› ‹ω < 2 ∗ pi› ‹sin ω = normalize p $ 2›
      by (simp add: sin-less-zero-iff[symmetric] nth-normalize-less-zero-iff)
    from this ‹ω < 2 ∗ pi› have ω − pi = arccos (cos (ω − pi))
      by (auto simp only: arccos-cos)
    from this ‹cos ω = normalize p $ 1› have ω − pi = arccos (− normalize p $ 1) by simp
    from this have ω = pi + arccos (− normalize p $ 1) by simp
    from this show ω = ?a using ‹¬ 0 ≤ p $ 2› by auto
  qed
}
ultimately show ?thesis
```

unfolding *radiant-of-def* **by** (*rule theI*)
qed

lemma *radiant-of-bounds* [*simp*]:
$0 \leq$ *radiant-of p radiant-of p* $< 2 * pi$
using *radiant-of* **by** *auto*

lemma *radiant-of-weak-ubound* [*simp*]:
radiant-of p $\leq 2 * pi$
using *radiant-of-bounds*(*2*)[*of p*] **by** *linarith*

1.2.5 Main Equations for Transforming to Polar Form

lemma *polar-form-eq*:
norm p $*_R$ *of-radiant* (*radiant-of p*) = *p*
using *radiant-of normalize-eq* **by** *simp*

lemma *relative-polar-form-eq*:
$Q +$ *dist P Q* $*_R$ *of-radiant* (*radiant-of* $(P - Q)$) $= P$
proof −
have *norm* $(P - Q)$ $*_R$ *of-radiant* (*radiant-of* $(P - Q)$) $= P - Q$
unfolding *polar-form-eq* **..**
moreover have *dist P Q* = *norm* $(P - Q)$ **by** (*simp add*: *dist-norm*)
ultimately show *?thesis* **by** (*metis add.commute diff-add-cancel*)
qed

1.3 Ptolemy's Theorem

lemma *dist-circle-segment*:
assumes $0 \leq$ *radius* $0 \leq \alpha$ $\alpha \leq \beta$ $\beta \leq 2 * pi$
shows *dist* (*center* + *radius* $*_R$ *of-radiant* α) (*center* + *radius* $*_R$ *of-radiant* β) $= 2 *$ *radius* $*$ *sin* $((\beta - \alpha) / 2)$
(**is** *?lhs* = *?rhs*)
proof −
have *trigonometry*: $(\cos \alpha - \cos \beta)^2 + (\sin \alpha - \sin \beta)^2 = (2 * \sin ((\beta - \alpha) / 2))^2$
proof −
have *sin-diff-minus*: $\sin ((\alpha - \beta) / 2) = - \sin ((\beta - \alpha) / 2)$
by (*simp only*: *sin-minus*[*symmetric*] *minus-divide-left minus-diff-eq*)
have $(\cos \alpha - \cos \beta)^2 + (\sin \alpha - \sin \beta)^2 =$
$(2 * \sin ((\alpha + \beta) / 2) * \sin ((\beta - \alpha) / 2))^2 + (2 * \sin ((\alpha - \beta) / 2) * \cos ((\alpha + \beta) / 2))^2$
by (*simp only*: *cos-diff-cos sin-diff-sin*)
also have $\ldots = (2 * \sin ((\beta - \alpha) / 2))^2 * ((\sin ((\alpha + \beta) / 2))^2 + (\cos ((\alpha + \beta) / 2))^2)$
unfolding *sin-diff-minus* **by** *algebra*
also have $\ldots = (2 * \sin ((\beta - \alpha) / 2))^2$ **by** *simp*
finally show *?thesis* **.**
qed
from *assms* **have** $0 \leq \sin ((\beta - \alpha) / 2)$ **by** (*simp add*: *sin-ge-zero*)

have *?lhs* = *sqrt* (*radius*^{2} ∗ ((*cos* α − *cos* β)^2 + (*sin* α − *sin* β)^2))
unfolding *dist-eq-on-real-2-vec* **by** *simp algebra*
also have . . . = *sqrt* (*radius*^{2} ∗ (*2* ∗ *sin* ((β − α) / *2*))^2) **by** (*simp add*: *trigonometry*)
also have . . . = *?rhs*
using ‹*0* ≤ *radius*› ‹*0* ≤ *sin* ((β − α) / *2*)› **by** (*simp add*: *real-sqrt-mult*)
finally show *?thesis* **.**
qed

theorem *ptolemy-trigonometric*:
fixes ω_1 ω_2 ω_3 :: *real*
shows *sin* (ω_1 + ω_2) ∗ *sin* (ω_2 + ω_3) = *sin* ω_1 ∗ *sin* ω_3 + *sin* ω_2 ∗ *sin* (ω_1 + ω_2 + ω_3)
proof −
have *sin* (ω_1 + ω_2) ∗ *sin* (ω_2 + ω_3) = ((*sin* ω_2)^2 + (*cos* ω_2)^2) ∗ *sin* ω_1 ∗ *sin* ω_3 + *sin* ω_2 ∗ *sin* (ω_1 + ω_2 + ω_3)
by (*simp only*: *sin-add cos-add*) *algebra*
also have . . . = *sin* ω_1 ∗ *sin* ω_3 + *sin* ω_2 ∗ *sin* (ω_1 + ω_2 + ω_3) **by** *simp*
finally show *?thesis* **.**
qed

theorem *ptolemy*:
fixes *A B C D center* :: *real* ^ *2*
assumes *dist center A* = *radius* **and** *dist center B* = *radius*
assumes *dist center C* = *radius* **and** *dist center D* = *radius*
assumes *ordering-of-points*:
radiant-of (*A* − *center*) ≤ *radiant-of* (*B* − *center*)
radiant-of (*B* − *center*) ≤ *radiant-of* (*C* − *center*)
radiant-of (*C* − *center*) ≤ *radiant-of* (*D* − *center*)
shows *dist A C* ∗ *dist B D* = *dist A B* ∗ *dist C D* + *dist A D* ∗ *dist B C*
proof −
from ‹*dist center A* = *radius*› **have** *0* ≤ *radius* **by** *auto*
define α β γ δ
where α = *radiant-of* (*A* − *center*) **and** β = *radiant-of* (*B* − *center*)
and γ = *radiant-of* (*C* − *center*) **and** δ = *radiant-of* (*D* − *center*)
from *ordering-of-points* **have** *angle-basics*:
α ≤ β β ≤ γ γ ≤ δ
0 ≤ α α ≤ *2* ∗ *pi* *0* ≤ β β ≤ *2* ∗ *pi*
0 ≤ γ γ ≤ *2* ∗ *pi* *0* ≤ δ δ ≤ *2* ∗ *pi*
unfolding *α-def β-def γ-def δ-def* **by** *auto*
from *assms*(*1*−*4*) **have**
A = *center* + *radius* $*_R$ *of-radiant* α *B* = *center* + *radius* $*_R$ *of-radiant* β
C = *center* + *radius* $*_R$ *of-radiant* γ *D* = *center* + *radius* $*_R$ *of-radiant* δ
unfolding *α-def β-def γ-def δ-def*
using *relative-polar-form-eq dist-commute* **by** *metis*+

from *this* **have** *dist-eqs*:
dist A C = *2* ∗ *radius* ∗ *sin* ((γ − α) / *2*)
dist B D = *2* ∗ *radius* ∗ *sin* ((δ − β) / *2*)

$dist\ A\ B = 2 * radius * sin\ ((\beta - \alpha)\ /\ 2)$
$dist\ C\ D = 2 * radius * sin\ ((\delta - \gamma)\ /\ 2)$
$dist\ A\ D = 2 * radius * sin\ ((\delta - \alpha)\ /\ 2)$
$dist\ B\ C = 2 * radius * sin\ ((\gamma - \beta)\ /\ 2)$
using *angle-basics* ‹$radius \geq 0$› *dist-circle-segment* **by** (*auto*)

have $dist\ A\ C * dist\ B\ D = 4 * radius\ \hat{}\ 2 * sin\ ((\gamma - \alpha)\ /\ 2) * sin\ ((\delta - \beta)\ /\ 2)$
unfolding *dist-eqs* **by** (*simp add*: *power2-eq-square*)
also have ... $= 4 * radius\ \hat{}\ 2 * (sin\ ((\beta - \alpha)\ /\ 2) * sin\ ((\delta - \gamma)\ /\ 2) + sin\ ((\gamma - \beta)\ /\ 2) * sin\ ((\delta - \alpha)\ /\ 2))$
proof −
define $\omega_1\ \omega_2\ \omega_3$ **where** $\omega_1 = (\beta - \alpha)\ /\ 2$ **and** $\omega_2 = (\gamma - \beta)\ /\ 2$ **and** $\omega_3 = (\delta - \gamma)\ /\ 2$
have $(\gamma - \alpha)\ /\ 2 = \omega_1 + \omega_2$ **and** $(\delta - \beta)\ /\ 2 = \omega_2 + \omega_3$ **and** $(\delta - \alpha)\ /\ 2 = \omega_1 + \omega_2 + \omega_3$
unfolding ω_1*-def* ω_2*-def* ω_3*-def* **by** (*auto simp add*: *field-simps*)
have $sin\ ((\gamma - \alpha)\ /\ 2) * sin\ ((\delta - \beta)\ /\ 2) = sin\ (\omega_1 + \omega_2) * sin\ (\omega_2 + \omega_3)$
using ‹$(\gamma - \alpha)\ /\ 2 = \omega_1 + \omega_2$› ‹$(\delta - \beta)\ /\ 2 = \omega_2 + \omega_3$› **by** (*simp only*:)
also have ... $= sin\ \omega_1 * sin\ \omega_3 + sin\ \omega_2 * sin\ (\omega_1 + \omega_2 + \omega_3)$
by (*rule ptolemy-trigonometric*)
also have ... $= (sin\ ((\beta - \alpha)\ /\ 2) * sin\ ((\delta - \gamma)\ /\ 2) + sin\ ((\gamma - \beta)\ /\ 2) * sin\ ((\delta - \alpha)\ /\ 2))$
using ω_1*-def* ω_2*-def* ω_3*-def* ‹$(\delta - \alpha)\ /\ 2 = \omega_1 + \omega_2 + \omega_3$› **by** (*simp only*:)
finally show *?thesis* **by** *simp*
qed
also have ... $= dist\ A\ B * dist\ C\ D + dist\ A\ D * dist\ B\ C$
unfolding *dist-eqs* **by** (*simp add*: *distrib-left power2-eq-square*)
finally show *?thesis* **.**
qed

end

References

[1] J. Harrison. Ptolemy's theorem. https://github.com/jrh13/hol-light/blob/master/100/ptolemy.ml.

[2] F. Wiedijk. Formalizing 100 theorems. http://www.cs.ru.nl/~freek/100/.

[3] Wikipedia. Ptolemy's theorem — wikipedia, the free encyclopedia, 2016. https://en.wikipedia.org/w/index.php?title=Ptolemy%27s_theorem&oldid=727017817 [Online; accessed 6-August-2016].