Fundamentals of Unconstrained Optimization

Dustin Bryant

February 6, 2026

Abstract

As formal methods gain traction in machine learning and numerical analysis, the community needs computer-checked proofs of core optimization results. Existing Isabelle libraries still lack a foundational framework for unconstrained optimization. We close this gap with a comprehensive Isabelle/HOL development that formalizes:

(1) minimizers, strict and isolated local minimizers;

(2) first- and second-order optimality conditions for scalar functions $f : \mathbb{R} \to \mathbb{R}$;

(3) first-order optimality conditions for vector functions $g : \mathbb{R}^n \to \mathbb{R}$; and

(4) a worked example showing that the continuous function

$$h(x) = \begin{cases} x^4\big(\cos(1/x) + 2\big), & x \neq 0, \\ 0, & x = 0 \end{cases}$$

has a *strict* but *non-isolated* local minimizer at $x = 0$.

The new session `Unconstrained_Optimization` provides sound, reusable foundations for future proof-checking tools and mechanized research in optimization, analysis, and algorithmic correctness.

Contents

1 Auxiliary Facts

theory *Auxilary-Facts*
imports
Sigmoid-Universal-Approximation.Limits-Higher-Order-Derivatives
begin

1.1 Differentiation Lemmas

lemma *has-derivative-imp*:
fixes f :: *real* ⇒ *real*
assumes (f *has-derivative* f') (*at* x)
shows f *differentiable* (*at* x) ∧ *deriv* f x = f' *1*
⟨*proof*⟩

lemma *DERIV-inverse-func*:
assumes $x \neq 0$
shows *DERIV* (λ*w*. *1* / *w*) x :> −*1* / x^*2*
⟨*proof*⟩

lemma *power-rule*:
fixes z :: *real* **and** n :: *nat*
shows *deriv* (λ*x*. x ^ n) z = (*if* n = *0* *then* *0* *else* *real* n ∗ z ^ (n − *1*))
⟨*proof*⟩

1.1.1 Transfer Lemmas

lemma *has-derivative-transfer-on-ball*:
fixes f g :: *real* ⇒ *real*
assumes *eps-gt0*: *0* < ε
assumes *eq-on-ball*: ∀ y. y ∈ *ball* x ε ⟶ f y = g y
assumes *f-has-deriv*: (f *has-derivative* D) (*at* x)
shows (g *has-derivative* D) (*at* x)

⟨*proof*⟩

corollary *field-differentiable-transfer-on-ball*:
fixes $f\ g$ :: *real* ⇒ *real*
assumes $0 < \varepsilon$
assumes *eq-on-ball*: $\forall y.\ y \in ball\ x\ \varepsilon \longrightarrow f\ y = g\ y$
assumes *f-diff*: *f field-differentiable at x*
shows *g field-differentiable at x*
⟨*proof*⟩

1.2 Trigonometric Contraction

lemma *cos-contractive*:
fixes $x\ y$ :: *real*
shows $|cos\ x - cos\ y| \leq |x - y|$
⟨*proof*⟩

lemma *sin-contractive*:
fixes $x\ y$ :: *real*
shows $|sin\ x - sin\ y| \leq |x - y|$
⟨*proof*⟩

1.3 Algebraic Factorizations

lemma *biquadrate-diff-biquadrate-factored*:
fixes $x\ y$::*real*
shows $y\hat{}4 - x\hat{}4 = (y - x) * (y\hat{}3 + y\hat{}2 * x + y * x\hat{}2 + x\hat{}3)$
⟨*proof*⟩

1.4 Specific Trigonometric Values

lemma *sin-5pi-div-4*: $sin\ (5 * pi\ /\ 4) = -\ (sqrt\ 2\ /\ 2)$
⟨*proof*⟩

lemma *cos-5pi-div-4*: $cos\ (5 * pi\ /\ 4) = -\ (sqrt\ 2\ /\ 2)$
⟨*proof*⟩

1.5 Local Sign Preservation of Continuous Functions

1.5.1 Local Positivity

lemma *cont-at-pos-imp-loc-pos*:
fixes g :: *real* ⇒ *real* **and** x :: *real*
assumes *continuous* (*at x*) *g* **and** $g\ x > 0$
shows $\exists\delta > 0.\ \forall y.\ |y - x| < \delta \longrightarrow g\ y > 0$
⟨*proof*⟩

lemma *cont-at-pos-imp-loc-pos'*:
fixes g :: *real* ⇒ *real* **and** x :: *real*
assumes *continuous* (*at x*) *g* **and** $g\ x > 0$

shows $\exists\,\Delta > 0.\ \forall\,\delta.\ 0 < \delta \wedge \delta \leq \Delta \longrightarrow (\forall\,y.\ |y - x| < \delta \longrightarrow g\ y > 0)$
⟨*proof*⟩

1.5.2 Local Negativity

lemma *cont-at-neg-imp-loc-neg*:
fixes g :: *real* ⇒ *real* **and** x :: *real*
assumes *continuous* (*at* x) g **and** $g\ x < 0$
shows $\exists\,\delta > 0.\ \forall\,y.\ |y - x| < \delta \longrightarrow g\ y < 0$
⟨*proof*⟩

lemma *cont-at-neg-imp-loc-neg'*:
fixes g :: *real* ⇒ *real* **and** x :: *real*
assumes *continuous* (*at* x) g **and** $g\ x < 0$
shows $\exists\,\Delta > 0.\ \forall\,\delta.\ 0 < \delta \wedge \delta \leq \Delta \longrightarrow (\forall\,y.\ |y - x| < \delta \longrightarrow g\ y < 0)$
⟨*proof*⟩

end

2 Minimizers in Topological and Metric Spaces

theory *Minimizers-Definition*
imports *Auxilary-Facts*
begin

2.1 Abstract Topological Definitions

definition *global-minimizer* :: (*'a::topological-space* ⇒ *real*) ⇒ *'a* ⇒ *bool* **where**
global-minimizer f x-star ⟷ (∀ *x*. *f x-star* ≤ *f x*)

definition *local-minimizer-on* :: (*'a::topological-space* ⇒ *real*) ⇒ *'a* ⇒ *'a set* ⇒ *bool* **where**
local-minimizer-on f x-star U ⟷ (*open U* ∧ *x-star* ∈ *U* ∧ (∀ *x* ∈ *U*. *f x-star* ≤ *f x*))

definition *local-minimizer* :: (*'a::topological-space* ⇒ *real*) ⇒ *'a* ⇒ *bool* **where**
local-minimizer f x-star ⟷ (∃ *U*. *open U* ∧ *x-star* ∈ *U* ∧ (∀ *x* ∈ *U*. *f x-star* ≤ *f x*))

definition *isolated-local-minimizer-on* :: (*'a::topological-space* ⇒ *real*) ⇒ *'a* ⇒ *'a set* ⇒ *bool* **where**
isolated-local-minimizer-on f x-star U ⟷
(*local-minimizer-on f x-star U* ∧ ({*x* ∈ *U*. *local-minimizer f x*} = {*x-star*}))

definition *isolated-local-minimizer* :: (*'a::topological-space* ⇒ *real*) ⇒ *'a* ⇒ *bool* **where**
isolated-local-minimizer f x-star ⟷
(∃ *U*. *local-minimizer-on f x-star U* ∧ ({*x* ∈ *U*. *local-minimizer f x*} = {*x-star*}))

definition *strict-local-minimizer-on* :: ($'a$::*topological-space* ⇒ *real*) ⇒ $'a$ ⇒ $'a$ *set* ⇒ *bool* **where**
strict-local-minimizer-on f x-star U ⟷
(*open U* ∧ *x-star* ∈ *U* ∧ (∀ *x* ∈ *U* − {*x-star*}. *f x-star* < *f x*))

definition *strict-local-minimizer* :: ($'a$::*topological-space* ⇒ *real*) ⇒ $'a$ ⇒ *bool* **where**
strict-local-minimizer f x-star ⟷ (∃ *U*. *strict-local-minimizer-on f x-star U*)

2.2 Metric Space Reformulations

lemma *local-minimizer-on-def2*:
fixes *f* :: $'a$::*metric-space* ⇒ *real*
assumes *local-minimizer f x-star*
shows ∃ *N* > *0*. ∀ *x* ∈ *ball x-star N*. *f x-star* ≤ *f x*
⟨*proof*⟩

lemma *local-minimizer-def2*:
fixes *f* :: $'a$::*metric-space* ⇒ *real*
assumes *local-minimizer f x-star*
shows ∃ *N* > *0*. ∀ *x*. *dist x x-star* < *N* ⟶ *f x-star* ≤ *f x*
⟨*proof*⟩

lemma *isolated-local-minimizer-on-def2*:
fixes *f* :: $'a$::*metric-space* ⇒ *real*
assumes *isolated-local-minimizer-on f x-star U*
shows ∃ *N* > *0*. ∀ *x* ∈ *ball x-star N*. (*local-minimizer f x* ⟶ *x* = *x-star*)
⟨*proof*⟩

lemma *isolated-local-minimizer-def2*:
fixes *f* :: $'a$::*metric-space* ⇒ *real*
assumes *isolated-local-minimizer f x-star*
shows ∃ *N* > *0*. ∀ *x* ∈ *ball x-star N*. (*local-minimizer f x* ⟶ *x* = *x-star*)
⟨*proof*⟩

lemma *strict-local-minimizer-on-def2*:
fixes *f* :: $'a$::*metric-space* ⇒ *real*
assumes *strict-local-minimizer-on f x-star U*
shows ∃ *N* > *0*. ∀ *x* ∈ *ball x-star N* − {*x-star*}. *f x-star* < *f x*
⟨*proof*⟩

lemma *strict-local-minimizer-def2*:
fixes *f* :: $'a$::*metric-space* ⇒ *real*
assumes *strict-local-minimizer f x-star*
shows ∃ *N* > *0*. ∀ *x* ∈ *ball x-star N* − {*x-star*}. *f x-star* < *f x*
⟨*proof*⟩

lemma *local-minimizer-neighborhood*:
fixes *f* :: *real* ⇒ *real*

assumes *loc-min*: *local-minimizer f x-min*
shows $\exists\,\delta > 0.\ \forall\, h.\ |h| < \delta \longrightarrow f\ (x\text{-}min + h) \geq f\ x\text{-}min$
⟨*proof*⟩

lemma *local-minimizer-from-neighborhood*:
fixes *f* :: *real* ⇒ *real* **and** *x-min* :: *real*
assumes $\exists\,\delta > 0.\ \forall\, x.\ |x - x\text{-}min| < \delta \longrightarrow f\ x\text{-}min \leq f\ x$
shows *local-minimizer f x-min*
⟨*proof*⟩

end

3 Minimizer Implications

theory *First-Order-Conditions*
imports *Minimizers-Definition*
begin

notation *norm* (‖-‖)

3.1 Implications for a Given Minimizer Type

lemma *strict-local-minimizer-imp-local-minimizer*:
assumes *strict-local-minimizer f x-star*
shows *local-minimizer f x-star*
⟨*proof*⟩

lemma *isolated-local-minimizer-imp-strict*:
assumes *isolated-local-minimizer f x-star*
shows *strict-local-minimizer f x-star*
⟨*proof*⟩

3.2 Characterization of Non-Isolated Minimizers

lemma *not-isolated-minimizer-def*:
assumes *local-minimizer f x-star*
shows $(\exists\, x\text{-}seq{::}nat \Rightarrow real.\ (\forall\, n.\ local\text{-}minimizer\ f\ (x\text{-}seq\ n) \wedge x\text{-}seq\ n \neq x\text{-}star)$
$\wedge\ ((x\text{-}seq \longrightarrow x\text{-}star)\ at\text{-}top)) = (\neg\ isolated\text{-}local\text{-}minimizer\ f\ x\text{-}star)$
⟨*proof*⟩

3.3 First-Order Condition

theorem *Fermat's-theorem-on-stationary-points*:
fixes *f* :: *real* ⇒ *real*
assumes $(f\ has\text{-}derivative\ f')\ (at\ x\text{-}min)$
assumes *local-minimizer f x-min*
shows $(deriv\ f)\ x\text{-}min = 0$
⟨*proof*⟩

definition *stand-basis-vector* :: $'n$::*finite* ⇒ *real*^'*n* — the i-th standard basis vector
where *stand-basis-vector i* = (χ *j*. *if j* = *i then 1 else 0*)

lemma *stand-basis-vector-index*[*simp*]: (*stand-basis-vector i*) $ *j* = (*if j* = *i then* (*1*::*real*) *else 0*)
⟨*proof*⟩

lemma *stand-basis-vector-nonzero*[*simp*]: *stand-basis-vector i* ≠ *0*
⟨*proof*⟩

lemma *norm-stand-basis-vector*[*simp*]: *norm* (*stand-basis-vector i*) = *1*
⟨*proof*⟩

lemma *inner-stand-basis-vector*[*simp*]: *inner* (*stand-basis-vector i*) (*stand-basis-vector j*) = (*if i* = *j then 1 else 0*)
⟨*proof*⟩

lemma *Basis-characterisation*:
stand-basis-vector i ∈ (*Basis* :: (*real*^'*n*) *set*) **and**
∀ *b*∈(*Basis*::(*real*^'*n*)*set*). ∃ *i*. *b* = *stand-basis-vector i*
⟨*proof*⟩

lemma *stand-basis-expansion*:
fixes *x* :: *real*^'*n*
shows *x* = (∑ *j*∈*UNIV*. (*x* $ *j*) $*_R$ *stand-basis-vector j*)
⟨*proof*⟩

lemma *has-derivative-affine*:
fixes *a v* :: '*a*::*real-normed-vector*
shows ((λ*t*. *a* + *t* $*_R$ *v*) *has-derivative* (λ*h*. *h* $*_R$ *v*)) (*at x*)
⟨*proof*⟩

theorem *Fermat's-theorem-on-stationary-points-mult*:
fixes *f* :: *real* ^ '*n* ⇒ *real*
assumes *der-f*: (*f has-derivative f'*) (*at x-min*)
assumes *min-f*: *local-minimizer f x-min*
shows *GDERIV f x-min* :> *0*
⟨*proof*⟩

end

4 Second-Order Conditions

theory *Second-Derivative-Test*
imports *First-Order-Conditions*
begin

4.1 Necessary Condition

lemma *snd-derivative-nonneg-at-local-min-necessary*:
  fixes *f* :: *real* ⇒ *real*
  assumes *C2-cont-diff-at-xmin*: *C-k-on 2 f* (*U* :: *real set*)
  assumes *min-in-U*: (*x-min* :: *real*) ∈ *U*
  assumes *loc-min*: *local-minimizer f x-min*
  shows *deriv* (*deriv f*) *x-min* ≥ *0*
⟨*proof*⟩

4.2 Sufficient Condition

lemma *second-derivative-test*:
  fixes *f* :: *real* ⇒ *real* **and** *a* :: *real* **and** *b* :: *real* **and** *x-min* :: *real*
  assumes *valid-interval*: *a* < *b*
  assumes *twice-continuously-differentiable*: *C-k-on 2 f* {*a* <..< *b*}
  assumes *min-exists*: *x-min* ∈ {*a* <..< *b*}
  assumes *fst-deriv-req*: (*deriv f*) *x-min* = *0*
  assumes *snd-deriv-req*: *deriv* (*deriv f*) *x-min* > *0*
  shows *loc-min*: *local-minimizer f x-min*
⟨*proof*⟩

end

5 Pathological Example: Non-Isolated Strict Local Minima

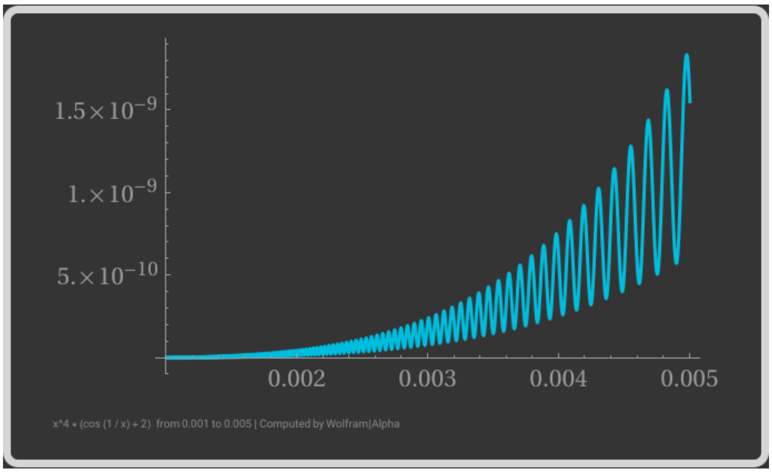
theory *Cont-Nonisolated-Strict-Local-Minimizer-Exists*
  imports *Second-Derivative-Test HOL−Library.Quadratic-Discriminant*
begin

Idea of the example. We construct a continuous function

$$f(x) = \begin{cases} x^4\big(\cos(1/x) + 2\big), & x \neq 0, \\ 0, & x = 0 \end{cases}$$

whose oscillations *speed up* as $x \to 0$ because of the $\cos(1/x)$ term. Multiplying by x^4 makes the function and its first derivative vanish at the origin, ensuring that $x = 0$ is a strict local minimizer, while the shifted cosine creates infinitely many additional strict local minimizers that accumulate at 0. Hence the minimizer at 0 is *strict* but *not isolated*.

theorem *Exists-Continuous-Func-with-non-isolated-strict-local-minimizer*:
  ∃ *f*::*real* ⇒ *real*. *continuous-on* ℝ *f* ∧
    (∃ *x-star*. *strict-local-minimizer f x-star* ∧ ¬ *isolated-local-minimizer f x-star*)
⟨*proof*⟩



end
theory *Unconstrained-Optimization*
  imports *Auxilary-Facts*
    Minimizers-Definition
    First-Order-Conditions
    Second-Derivative-Test
    Cont-Nonisolated-Strict-Local-Minimizer-Exists
begin

end

References

[1] T. Nipkow, M. Wenzel, and L. C. Paulson. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 2002.

[2] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.