Formalising Mathematics Project 1

Alex Jin; CID:02418008

January 29, 2026

1 Overview

This project formalizes the standard ε–δ notions of continuity and uniform continuity for real-valued functions in Lean, and uses these definitions to analyze the function $f : \mathbb{R} \to \mathbb{R}$ given by $f(x) = x^2$. Although the arguments are familiar from first-year real analysis, translating them into Lean requires making every inequality choice and quantifier step explicit, which forces a more structured approach.

2 Mathematical Background

This section introduces the definitions and lemmas which are necessary to prove the main theorem.

Definition 1. ***(Continuity on a specific point)***

Let $f : \mathbb{R} \to \mathbb{R}$ and let $x_0 \in \mathbb{R}$. We say that f is continuous at x_0 *if*

$$\forall \varepsilon > 0,\ \exists \delta > 0,\ \forall x,\ |x - x_0| < \delta \to |f(x) - f(x_0)| < \varepsilon.$$

We say that f is continuous on $\mathbb{R}$ if it is continuous at every $x_0 \in \mathbb{R}$.

The crucial difference from point-wise continuity to uniform continuity is the choice of δ which depends only on ε in uniform scenario, not on the choice of point in S.

Definition 2. ***(Uniform continuity on a set)***

Let $S \subseteq \mathbb{R}$. We say that f is uniformly continuous *on S if*

$$\forall \varepsilon > 0,\ \exists \delta > 0,\ \forall x, y \in S,\ |x - y| < \delta \Rightarrow |f(x) - f(y)| < \varepsilon.$$

Before moving on to the main proofs, we first record a standard inequality that will be used repeatedly to bound expressions involving absolute values, namely the triangle inequality.

Lemma 1. ***(Triangle inequality).*** *For any $a, b \in \mathbb{R}$, $|a+b| \leq |a| + |b|$.*

With these definitions in place and the basic algebraic and inequality tools recorded, we can now prove the main results about $f(x) = x^2$: continuity on $\mathbb{R}$, failure of uniform continuity on $\mathbb{R}$, and uniform continuity on $[0, 1]$.

Theorem 1. ***(Continuity of $f(x) = x^2$ on $\mathbb{R}$).***

Fix $c \in \mathbb{R}$ and let $\varepsilon > 0$. Using $|x^2 - c^2| = |x - c|\,|x + c|$, if we assume $|x - c| < 1$ then $|x| \leq |x-c| + |c| < 1 + |c|$, so $|x + c| \leq |x| + |c| < 1 + 2|c|$. Hence $|x^2 - c^2| < |x - c|(1 + 2|c|)$. Take $\delta = \min\left(1, \frac{\varepsilon}{1+2|c|}\right)$. Then $|x - c| < \delta$ implies $|x^2 - c^2| < \varepsilon$, so x^2 is continuous at c, and therefore continuous on $\mathbb{R}$.

Theorem 2. ***($f(x) = x^2$ is not uniformly continuous on $\mathbb{R}$).***

We prove the negation of uniform continuity. Take $\varepsilon = 1$. Given any $\delta > 0$, set $x = 1/\delta$ and $y = 1/\delta + \delta/2$. Then $|x - y| = \delta/2 < \delta$, but

$$|x^2 - y^2| = |x - y|\,|x + y| = \frac{\delta}{2}\left(\frac{2}{\delta} + \frac{\delta}{2}\right) = 1 + \frac{\delta^2}{4} \geq 1.$$

Thus the $\varepsilon = 1$ condition fails for every δ, so x^2 is not uniformly continuous on $\mathbb{R}$.

Uniform continuity on $[0, 1]$ follows since $|x + y| \leq 2$ on $[0, 1]$, so $|x^2 - y^2| \leq 2|x - y|$; taking $\delta = \varepsilon/2$ completes the proof.

3 Code Implementation

Lean (via `mathlib`) already provides standard, well-developed notions of continuity and uniform continuity. However, since the goal of this project is to work strictly within the first-year real analysis viewpoint, we deliberately avoid relying on these library-level abstractions. Instead, we introduce our own definitions of continuity and uniform continuity using the explicit ε–δ notions on $\mathbb{R}$.

```
def my_continuous_at (f : ℝ → ℝ) (x₀ : ℝ) : Prop :=
  ∀ ε > 0, ∃ δ > 0, ∀ x, |x - x₀| < δ → |f x - f x₀| < ε
```

```
def my_uniformly_continuous (f : ℝ → ℝ) : Prop :=
  ∀ ε > 0, ∃ δ > 0, ∀ x y, |x - y| < δ → |f x - f y| < ε

def sq_fun (x : ℝ) : ℝ := x^2
```

The Lean definitions are straightforward translations of the ε–δ statements above. Before proving the main results, we establish a few preliminary lemmas used repeatedly later, including the algebraic factorisation for $x^2 - c^2$ and the triangle inequality. For the triangle inequality, I use the existing tactic `abs_add`, which states $|x+y| \leq |x| + |y|$ in the general setting of a `linear_ordered_add_comm_group`; in our case we simply specialise it to $\mathbb{R}$.

```
lemma abs_sq_fun (x y : ℝ) : |sq_fun x - sq_fun y| = |x - y| * |x + y| :=
↪ by
  dsimp [sq_fun]
  rw[← abs_mul]
  ring_nf

theorem my_tri_ineq (x y : ℝ) : |x + y| ≤ |x| + |y| := by
  exact abs_add x y
```

We now start the point-wise continuity proof. The overall structure follows the standard ε–δ strategy from the lecture repository (Section 02, Sheet 6). The first step is to introduce an auxiliary constant $c := 2|x_0| + 1$ and prove $c > 0$, so that later we can safely choose δ involving the quotient ε/c.

```
theorem sq_fun_continuous (x₀ : ℝ) : my_continuous_at sq_fun x₀ := by
  intro ε hε
  let c := 2 * |x₀| + 1
  have hc_pos : c > 0 := by
    simp [c]
    have h0 : 0 ≤ 2 * |x₀| := by
      exact mul_nonneg (by norm_num) (abs_nonneg x₀)
    exact add_pos_of_nonneg_of_pos h0 (by norm_num)
```

In informal mathematics we can simply choose $\delta := \min(1, \varepsilon/c)$ and proceed. But in Lean, after `use min 1 (ε / c)`, the goal becomes a conjunction (both $\delta > 0$ and the main implication), so we have to prove each part separately. The final simplification step is also more complicate than on paper, since we must explicitly guide Lean to rewrite and cancel terms in the inequality.

```
-- Choose δ = min(1, ε / c)
use min 1 (ε / c)

constructor
· apply lt_min
  · norm_num
  · exact div_pos hε hc_pos
· intro X hX
  rw[abs_sq_fun]

  have h_le_1 : |X - x₀| < 1 := by
    apply lt_of_lt_of_le hX
    exact min_le_left 1 (ε / c)

  have h_le_eps_c : |X - x₀| < ε / c := by
    apply lt_of_lt_of_le hX
    apply min_le_right 1 (ε / c)

  have h_upper_bound : |X + x₀| < c := by
    calc |X + x₀|
      _ = |X - x₀ + 2 * x₀| := by ring_nf
      _ ≤ |X - x₀| + |2 * x₀| := my_tri_ineq (X - x₀) (2 * x₀)
      _ < 1 + |2 * x₀| := by linarith
      _ = 1 + 2 * |x₀| := by rw[abs_mul, abs_two]
      _ = c := by ring

  calc |X - x₀| * |X + x₀|
    _ < (ε / c) * c := by
      apply mul_lt_mul'' h_le_eps_c h_upper_bound (abs_nonneg _)
       ↪  (abs_nonneg _)
    _ = ε := by simp[div_eq_mul_inv, mul_assoc, hc_pos.ne']
```

We now move on to the non-uniform continuity proof. There is one key difference: the goal is a negation, ¬`my_uniformly_continuous sq_fun`. Here we start with `unfold my_uniformly_continuous` so the goal is rewritten into its explicit quantified form. This is important because `push_neg` works on quantifiers and inequalities directly; it does not automatically expand our custom definition.

```
theorem sq_fun_not_uniform_on_r : ¬ (my_uniformly_continuous sq_fun) :=
↪  by
  unfold my_uniformly_continuous
  push_neg

  use 1
  constructor
  · norm_num
  · intro δ hδ
    use 1 / δ, 1 / δ + δ / 2
    constructor
    · rw [abs_sub_comm]
      ring_nf
      rw [abs_of_pos]
      · linarith
      · linarith
    · rw [abs_sq_fun]
      have h_diff : |1/δ - (1/δ + δ/2)| = δ/2 := by
        ring_nf
        rw [abs_mul, abs_of_pos hδ]
        norm_num
      have h_sum : |1/δ + (1/δ + δ/2)| = 2/δ + δ/2 := by
        ring_nf
        rw [abs_of_pos]
        apply add_pos
        · nlinarith
        · exact mul_pos (inv_pos.2 hδ) (by norm_num)

      rw [h_diff, h_sum]
      calc δ / 2 * (2 / δ + δ / 2)
        _ = (δ / 2) * (2 / δ) + (δ / 2) * (δ / 2) := by rw [mul_add]
        _ = 1 + δ^2 / 4 := by
          field_simp [hδ.ne']
          ring
        _ ≥ 1 := by
          apply le_add_of_nonneg_right
          apply div_nonneg
          · exact sq_nonneg δ
          · norm_num
```

Finally, we show uniform continuity of $f(x) = x^2$ on $[0, 1]$. Compared to the previous proofs, the key new step is that we must explicitly use the hypotheses hx and hy describing the bounds $0 \le x \le 1$ and $0 \le y \le 1$. These hypotheses are conjunctions, so we use `obtain ⟨x_lower, x_upper⟩ := hx` (and similarly for hy) to extract the individual inequalities.

```
theorem sq_fun_uniform_on_01 :
  ∀ ε > 0, ∃ δ > 0, ∀ x y,
  (0 ≤ x ∧ x ≤ 1) → (0 ≤ y ∧ y ≤ 1) → |x - y| < δ → |sq_fun x - sq_fun
  ↪  y| < ε := by
  intro ε hε
  use ε / 2
  constructor
  · linarith
  · intro x y hx hy h
    rw [abs_sq_fun]
    have h_bound_fun : |x + y| ≤ 2 := by
      obtain ⟨x_lower, x_upper⟩ := hx
      obtain ⟨y_lower, y_upper⟩ := hy
      apply le_trans (my_tri_ineq x y)
      rw [abs_of_nonneg x_lower]
      rw [abs_of_nonneg y_lower]
      linarith

    apply lt_of_le_of_lt (b := |x - y| * 2)
    · apply mul_le_mul_of_nonneg_left
      · exact h_bound_fun
      · exact abs_nonneg (x - y)

    exact (lt_div_iff₀ two_pos).mp h
```

4 Conclusion

Overall, this project showed me that a standard ε–δ proof becomes much more detailed in Lean. I proved that $f(x) = x^2$ is continuous on $\mathbb{R}$, not uniformly continuous on $\mathbb{R}$, and uniformly continuous on $[0, 1]$. The mathematical ideas are the same as in first-year analysis, but Lean forces me to write every step clearly: every rewrite, every inequality, and every side condition has to be justified.

The main difficulties were: (i) finding the right lemmas for inequalities, such as lemmas about positivity/nonnegativity and lemmas that let you multiply or add inequalities safely; (ii) writing `calc` blocks, because in maths I would often skip steps, but in Lean I have to show each intermediate equation or inequality and give Lean enough information to accept it; and (iii) dealing with absolute values, since they often create extra goals and conjunctions that I need to split and manage carefully. I also found that simple algebraic rewriting can take a surprising amount of time: to turn an expression into the form I wanted, I often had to try different tactics like `simp`, `ring_nf`, `linarith/nlinarith`, or manual rewriting before the proof would move forward. Finally, I am still not confident with lemma search tools: `exact?` often did not close the goal, and it suggested using `apply?`, but `apply?` usually returned a long list of options that were not helpful, so I relied a lot on reading documentation and trial-and-error.

Looking back, the biggest lesson is that planning matters. Proofs became much easier once I started splitting them into small helper lemmas (for example, lemmas for bounding $|x + c|$ or for rewriting $x^2 - c^2$), instead of trying to finish everything inside one long proof. If I did this project again, I would also improve my definition of uniform continuity by defining it on a set $S \subseteq \mathbb{R}$, so that the Lean statements match the usual mathematics more directly and proofs on intervals have a cleaner structure. I did not do this mainly because I am not familiar with Lean's set techniques yet, but learning them would make the development shorter, clearer, and easier to extend.