An Introduction to Formal Real Analysis, Rutgers University, Fall 2025, Math 311H

Lecture 6: Advanced Proof Techniques and the Squeeze Theorem

Prof. Alex Kontorovich

This text is automatically generated by LLM from “Real Analysis, The Game”, Lecture 6

1 Big Boss: The Sum of Convergent Sequences

One of the most fundamental ideas in analysis is that 'nice operations preserve convergence.' If two sequences each converge, then their sum also converges, and converges to the sum of their limits.

This might seem obvious at first – after all, if $a(n)$ is getting close to L and $b(n)$ is getting close to M, shouldn't $a(n)+b(n)$ get close to $L+M$? While the intuition is correct, making this rigorous requires some clever maneuvering with our epsilon-N definition.

Here's the key insight: if an engineer demands that our combined output be within ε of the target $L + M$, we can't just demand that each factory independently meet the full tolerance ε. Instead, we need to be clever about how we allocate our 'tolerance budget.'

Think of it this way: if the first factory can guarantee its output is within $\varepsilon/2$ of L, and the second factory can guarantee its output is within $\varepsilon/2$ of M, then by the triangle inequality, their sum will be within ε of $L + M$. This is the heart of the proof!

1.1 The Mathematical Setup

Suppose we have:

- A sequence $a : \mathbb{N} \to \mathbb{R}$ that converges to L
- A sequence $b : \mathbb{N} \to \mathbb{R}$ that converges to M
- A sequence $c : \mathbb{N} \to \mathbb{R}$ with the property that $c(n) = a(n) + b(n)$ for all n

We want to prove that c converges to $L + M$.

1.2 Strategic Approach

1. Start by unfolding the definitions of convergence in the goal and hypotheses
2. Given any $\varepsilon > 0$, use the convergence of a to get an N_a that works for $\varepsilon/2$
3. Similarly, use the convergence of b to get an N_b that works for $\varepsilon/2$
4. Take $N = N_a + N_b$ (ensuring both conditions are satisfied)
5. Use the triangle inequality to combine the two half-tolerances

1.3 Lean Solution

```
Statement SumLim (a b c : ℕ → ℝ) (L M : ℝ)
    (ha : SeqLim a L) (hb : SeqLim b M) (hc : ∀ n, c n =
        a n + b n) :
    SeqLim c (L + M) := by
  change ∀ ε > 0, ∃ N : ℕ, ∀ n ≥ N, |c n - (L + M)| < ε
  intro ε hε
  unfold SeqLim at ha
  change ∀ ε₁ > 0, ∃ Na : ℕ, ∀ n ≥ Na, |a n - L| < ε₁ at
      ha
  change ∀ ε₂ > 0, ∃ Nb : ℕ, ∀ n ≥ Nb, |b n - M| < ε₂ at
      hb
  specialize ha (ε / 2)
```

```
  specialize hb (ε / 2)
  have eps_on_2_pos : 0 < ε / 2 := by linarith [hε]
  specialize ha eps_on_2_pos
  specialize hb eps_on_2_pos
  choose Na hNa using ha
  choose Nb hNb using hb
  use Na + Nb
  intro n hn
  specialize hc n
  rewrite [hc]
  have thing : a n + b n - (L + M) = (a n - L) + (b n -
     M) := by ring_nf
  rewrite [thing]
  specialize hNa n
  specialize hNb n
  have ineq_a : Na ≤ n := by bound
  have ineq_b : Nb ≤ n := by bound
  specialize hNa ineq_a
  specialize hNb ineq_b
  have ineq : |a n - L + (b n - M)| ≤ |a n - L| + |(b n
     - M)| := by apply abs_add
  bound
```

1.4 Natural Language Proof

Theorem: If two sequences of real numbers converge to their respective limits, then the sequence formed by adding corresponding terms also converges, and its limit is the sum of the original limits.

Proof: Suppose sequences $a(n)$ and $b(n)$ converge to L and M respectively, and we want to show that $c(n) = a(n) + b(n)$ converges to $L + M$.

By definition, we need to show that for any tolerance $\varepsilon > 0$, we can find a point N such that for all $n \geq N$, we have $|c(n) - (L + M)| < \varepsilon$.

Since $a(n)$ converges to L, we can find N_1 such that $|a(n) - L| < \varepsilon/2$ for all $n \geq N_1$. Since $b(n)$ converges to M, we can find N_2 such that $|b(n) - M| < \varepsilon/2$ for all $n \geq N_2$.

Let $N = N_1 + N_2$. Then for any $n \geq N$:

$$|c(n) - (L + M)| = |(a(n) + b(n)) - (L + M)| = |(a(n) - L) + (b(n) - M)|$$

By the triangle inequality, this is at most:

$$|a(n) - L| + |b(n) - M| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$$

Therefore, $c(n)$ converges to $L + M$. **QED**

2 Split Ands: Breaking Down Complex Goals

Mathematical proofs often require us to establish multiple related facts simultaneously. When your goal involves proving multiple statements connected by "and" ($\wedge$), the `split_ands` tactic becomes invaluable.

Think of `split_ands` as a way to break down a complex manufacturing specification into individual quality checks. Instead of trying to verify that a product meets three different standards all at once, we can tackle each standard separately and systematically.

2.1 New Tools

The `split_ands` tactic breaks apart "and" goals into individual pieces. If your goal is $h_1 \wedge h_2 \wedge h_3$, then calling `split_ands` will break that into three separate goals: first h_1, then h_2, and finally h_3.

2.2 Example

```
Statement (x y : ℝ) (hx : x = 2) (hy : y = 3) :
    x = 2 ∧ y = 3 := by
split_ands
apply hx
apply hy
```

The `split_ands` tactic might seem simple, but it's incredibly powerful for organizing complex proofs. Many important mathematical theorems have conclusions that are conjunctions—statements of the form "A and B and C". Being able to break these down systematically makes proofs much more manageable and readable.

3 Left and Right: Making Choices in Mathematics

After mastering `split_ands` to handle situations where we need to prove multiple things simultaneously, we now turn to a fundamentally different scenario: proving that at least one of several possibilities is true. This is the world of "or" statements ($\vee$).

While `split_ands` was about being comprehensive—proving every part of a conjunction—proving an "or" statement (disjunction) is about making a strategic choice. When faced with proving "P or Q," you don't need to prove both P and Q. You just need to prove one of them!

3.1 New Tools

When your goal is to prove an "or" statement, $P \vee Q$, you can do that by proving either P or Q:

- If you want to prove P, then say `left`, and the goal will turn into P
- If you want to prove Q, then say `right`, and the goal will turn into Q

3.2 Example

```
Statement (x y : ℝ) (hx : x = 2) (hy : y = 3) :
    x = 3 ∨ y = 3 := by
right
apply hy
```

"Or" statements are everywhere in mathematics. Existence proofs often have this flavor, and being able to choose the right branch can dramatically simplify your proof strategy.

4 Dot Notation: Accessing Parts of Complex Information

Often in mathematics, you'll be given a hypothesis that contains multiple pieces of information bundled together. For instance, you might know that "$x = 2$ AND $y = 3$" but only need the fact that "$y = 3$" for your current goal. Lean provides an elegant shorthand: dot notation.

4.1 Dot Notation Rules

When you have a hypothesis `h : P ∧Q`, you can access:

- The first part with `h.1` (which gives you P)
- The second part with `h.2` (which gives you Q)

For longer conjunctions like $P \land Q \land R$, note that `h.1` gives P, but `h.2` gives $Q \land R$ due to hidden parentheses: $P \land Q \land R = P \land (Q \land R)$. To get Q alone, use `h.2.1`, and for R, use `h.2.2`.

4.2 Example

```
Statement (x y : ℝ) (h : x = 2 ∧ y = 3) :
    y = 3 := by
apply h.2
```

Dot notation is essential for maintaining clarity in complex proofs where multiple conditions or properties are bundled together.

5 Cases': Handling All Possibilities

When you have a hypothesis like `h : P ∨Q`, you know that either P is true or Q is true, but you don't know which one. To proceed with your proof, you need to consider both possibilities systematically. The `cases'` tactic does exactly this.

5.1 The Cases' Tactic

When you have a hypothesis `h : P ∨Q`, you can say `cases' h with h1 h2`. This creates two separate goals:

- In the first goal, you get a new hypothesis `h1 : P`
- In the second goal, you get a new hypothesis `h2 : Q`

You must solve both goals to complete your proof, ensuring you've covered all logical possibilities.

5.2 Example

```
Statement (x y : ℝ) (h : x = 2 ∨ y = 3) :
    (x - 2) * (y - 3) = 0 := by
cases' h with h1 h2
rewrite [h1]
ring_nf
rewrite [h2]
ring_nf
```

5.3 The Complete And/Or Toolkit

You now have the complete And/Or matrix:

	∧	∨
Goal	`split_ands`	`left/right`
Hypothesis	`h.1, h.2`	`cases'`

Case analysis is everywhere in mathematics, from proving that every integer is either even or odd, to showing that continuous functions on closed intervals achieve their extrema.

6 AbsLt: Working with Absolute Values in Convergence

Working with absolute values is fundamental in real analysis, especially in the context of sequence convergence. Sometimes we need to extract more specific information from absolute value conditions, such as directional bounds.

6.1 The abs_lt Theorem

The `abs_lt` theorem states that $|x| < y$ if and only if $-y < x \wedge x < y$. This allows you to convert between absolute value inequalities and conjunctions of regular inequalities, making them easier to work with in proofs.

The key insight is that $|x| < y$ captures the idea that x is within distance y of zero, which means x lies in the interval $(-y, y)$.

6.2 Example Application

```
Statement (a : ℕ → ℝ) (L : ℝ) (ha : SeqLim a L) :
  ∃ N, ∀ n ≥ N, a n ≥ L - 1 := by
specialize ha 1 (by bound)
choose N hN using ha
use N
intro n hn
specialize hN n hn
rewrite [abs_lt] at hN
have : -1 < a n - L := by apply hN.1
bound
```

This proof demonstrates extracting a lower bound from a convergence condition. We showed that any convergent sequence is eventually bounded below (relative to its limit), which is a building block for many major theorems.

7 Big Boss: Squeeze Theorem

The Squeeze Theorem (also known as the Sandwich Theorem or Pinching Theorem) beautifully captures the intuitive idea that if you trap a sequence between two other sequences that both converge to the same limit, then the trapped sequence must also converge to that limit.

7.1 The Intuitive Picture

Imagine three runners on a track. Runner A and Runner C are both heading to the same finish line L, and Runner B is always between them. No matter how A and C weave back and forth, as long as they both end up at L and B stays between them, B must also end up at L. There's simply nowhere else for B to go!

7.2 The Mathematical Statement

Squeeze Theorem: If $a, c : \mathbb{N} \to \mathbb{R}$ both converge to L, and b is another sequence squeezed between a and c (i.e., $a(n) \leq b(n) \leq c(n)$ for all n), then b also converges to L.

7.3 Lean Proof

```
Statement SqueezeThm (a b c : ℕ → ℝ) (L : ℝ) (aToL :
   SeqLim a L)
(cToL : SeqLim c L) (aLeb : ∀ n, a n ≤ b n) (bLec : ∀ n,
    b n ≤ c n) :
  SeqLim b L := by
intro ε hε
specialize aToL ε hε
specialize cToL ε hε
choose Na hNa using aToL
choose Nc hNc using cToL
use Na + Nc
intro n hn
have hna : Na ≤ n := by bound
have hnc : Nc ≤ n := by bound
specialize hNa n hna
specialize hNc n hnc
```

```
rewrite [abs_lt] at hNa
rewrite [abs_lt] at hNc
rewrite [abs_lt]
split_ands
specialize aLeb n
bound
specialize bLec n
bound
```

7.4 Natural Language Proof

Proof: Given any $\varepsilon > 0$, we need to show that $|b(n) - L| < \varepsilon$ for sufficiently large n.

Since $a(n) \to L$, there exists N_a such that $|a(n) - L| < \varepsilon$ for all $n \geq N_a$. Since $c(n) \to L$, there exists N_c such that $|c(n) - L| < \varepsilon$ for all $n \geq N_c$.

Let $N = N_a + N_c$. For any $n \geq N$, we have:

$$L - \varepsilon < a(n) \leq b(n) \leq c(n) < L + \varepsilon \tag{1}$$

Therefore, $L - \varepsilon < b(n) < L + \varepsilon$, which means $|b(n) - L| < \varepsilon$. Hence $b(n) \to L$. **QED**

7.5 Applications

The Squeeze Theorem is a workhorse of mathematical analysis, used to prove challenging convergence results by reducing them to easier problems. Examples include:

- $\sin(1/n) \to 0$ (squeezed between $-1/n$ and $1/n$)
- Recursive sequences where exact formulas are intractable
- Sequences defined by complex geometric or probabilistic processes

The theorem demonstrates the power of combining multiple techniques: epsilon-N arguments, absolute value manipulation, logical decomposition, and inequality reasoning. This synthesis of tools is what makes advanced mathematical proof possible.

Thm (Sum Lim):

Assumptions: $a_n \to L$ (ha)

$b_n \to M$ (hb),

$\forall n$, $c_n = a_n + b_n$ (hc).

Goal: $c_n \to L+M$.

Pf: Let ε be given & assume $\varepsilon > 0$.

[Problem: where is N supposed to come from? Idea: get a_n within $\varepsilon/2$ of L & b_n within $\varepsilon/2$ of M. Then $c_n = a_n + b_n$ will be with ε of $L+M$]

Of course $\varepsilon/2 > 0$ (hε2), so ha $(\varepsilon/2)$ hε2 says: $\exists N_1, \forall n \geq N_1, |a_n - L| < \varepsilon/2$. Similarly, hb $(\varepsilon/2)$ hε2 says: $\exists N_2, \ldots$.

Choose N_1 hN_1 & choose N_2 hN_2.

use $N =$ ~~max N_1 N_2~~ $N_1 + N_2$.

Current goal: $\forall n \geq N_1 + N_2$, $|c_n - (L+M)| < \varepsilon$.

let n be given & assume $n \geq N_1 + N_2$.

have $f_1 : N_1 \leq n$:= by bound

have $f_2 : N_2 \leq n$:= by bound.

Specialize hN_1 n f_1 ($hN_1 : |a_n - L| < \varepsilon/2$).

Specialize hN_2 n f_2 ($hN_2 : |b_n - M| < \varepsilon/2$).

Specialize hc n ($hc : c_n = a_n + b_n$)

rewrite [hc]. (Goal: $|(a_n + b_n) - (L+M)| < \varepsilon$.).

have $f_3 : (a_n + b_n) - (L+M) = (a_n - L) + (b_n - M)$

:= by ring.

rewrite [f_3].

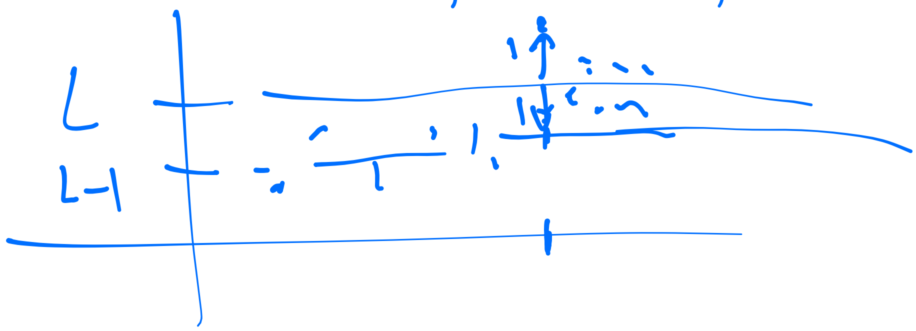
have $f_4 : |(a_n - L) + (b_n - M)| \leq |a_n - L| + |b_n - M|$:=

by apply abs_add

linarith [f_4, hN_1, hN_2].

Theorem: Assume $a : \mathbb{N} \to \mathbb{R}$ is given as is $L : \mathbb{R}$. Moreover, assume (ha) that $a_n \to L$.

Goal: $\exists N, \forall n \geq N, a_n \geq L - 1$.



Idea: set $\varepsilon = 1$.

New tool: abs_lt : $|x| < y \iff x < y \land -y < x$

have fo : 0<1 := by norm_num

Continuing the proof: Specialize ha 1 fo.

Then ha: $\exists N, \forall n \geq N, |a\ n - L| < 1$.

choose N hN using ha. (hN: $\forall n \geq N, |a_n - L| < 1$).

Use N.

New goal: $\forall n \geq N, a\ n \geq L - 1$.

Let n be given & assume that (hn) $n \geq N$.

Specialize hN n hn.

Now hN: $|a_n - L| < 1$.

rewrite [abs_lt] at hN.

New hN: $a_n - L < 1 \wedge -1 < a_n - L$.

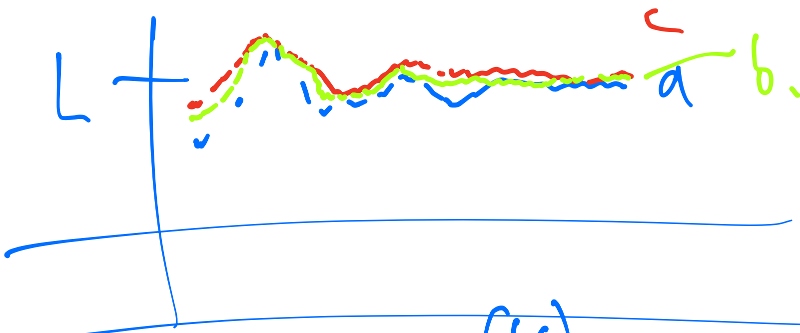
By theorem, Goal: $a_n \geq L - 1$.

linarith [hN.2].

(Idea: Could have chosen $\varepsilon = 1/2$ (≤ 1).).

Thm (Squeeze Thm): Given $a_n \leq b_n \leq c_n$ (aLeb) (bLec),

(aToL) $a_n \to L$, (cToL) $c_n \to L$. Goal: $b_n \to L$.



Pf: Let $\varepsilon > 0$ (hε) be given.

New goal: $\exists N, \forall n \geq N, |b_n - L| < \varepsilon$.

Specialize aToL ε hε. (aToL: $\exists N, \forall n \geq N, |a_n - L| < \varepsilon$)

Specialize cToL ε hε. (cToL : ∃ N_2, ∀ n ≥ N_2, $|c_n - L| < ε$)

Choose N_1 hN_1 using aToL

Choose N_2 hN_2 using cToL

Use ~~max~~ $N_1 + N_2$. (= N).

Intro n hn.

Current Goal State:
a b c : ℕ → ℝ,
L : ℝ, n N_1 N_2 : ℕ
ε : ℝ
hN_1 : ∀ n ≥ N_1, $|a_n - L| < ε$.
hN_2 : ∀ n ≥ N_2, $|c_n - L| < ε$.
hε : 0 < ε.
aLeb : ∀ n $a_n ≤ b_n$
bLec : ∀ n $b_n ≤ c_n$.
hn : n ≥ $N_1 + N_2$.
Goal $|b_n - L| < ε$

---

have f_1 : $N_1 ≤ n$:= by lomd
have f_2 : $N_2 ≤ n$:= ----

Specialize hN_1 n f_1
- - - - - 2 n f_2 ..   "⊢ goal".

rewrite [abs_lt] at hN_1 hN_2 ⊢.

New hN_1 : $-ε < a_n - L ∧ a_n - L < ε$.

- - - 2 : $-ε < c_n - L ∧ c_n - L < ε$.

New Goal : $-ε < b_n - L ∧ b_n - L < ε$,

Specialize aLeb n : $a_n ≤ b_n$
- - - bLec n, : $b_n ≤ c_n$.

split_ands.
subgoal 1 : linarith [hN_1.1, aLeb].

Subgoal z_i " [$4N_{2,2}$, bLeC)