

# AI For Research Tutorial

Paul Dütting

Renato Paes Leme

U Chile, April 2026

Google Research



FOUNDATIONS OF MATHEMATICS

# The AI Revolution in Math Has Arrived

17 | 

*AI is being used to prove new results at a rapid pace. Mathematicians think this is just the beginning.*



# This Tutorial: Goals and Outline

# Goal 1: How can AI and LLMs help you?



Automating the Research  
Workflow

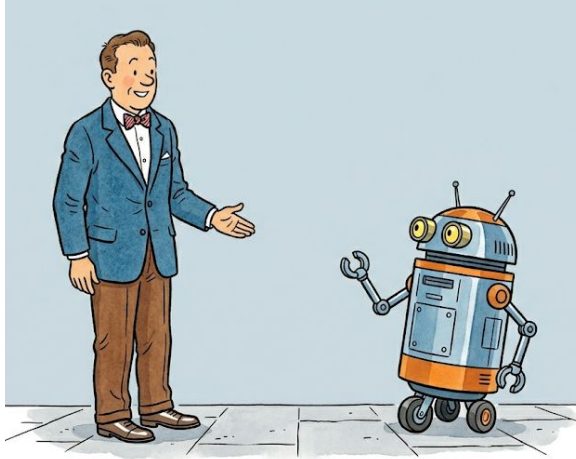
enabling more efficient  
task execution



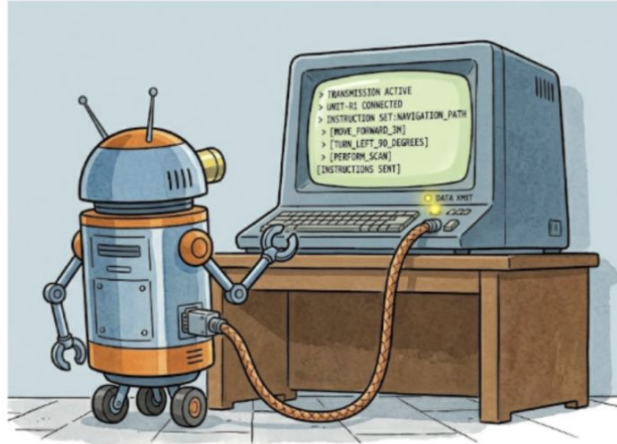
Proving Results using AI

pushing the boundaries of  
knowledge

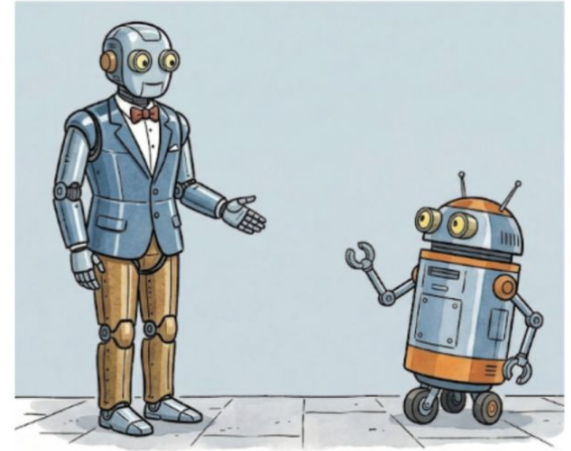
## Goal 2: Automating the use of AI



Directly talk to the AI



Write programs that call the AI



Get AI to direct other AIs

# Important Disclaimers

## Rapidly evolving technology

AI is evolving very fast, and the current tutorial reflects the technology right now. In a few months, the tools available could be very different.

## Personal experience

Tutorial is based on our own experienced trying to do Research using AI and reflects our subfield (Theoretical Computer Science, Algorithmic Game Theory).

Our intuition and understanding is also evolving as we go.

## Practical / Example-Driven

We will try to give many ideas of prompts and example of what can be done.

## Gemini Examples

Examples will be based on Google technologies (Gemini, Gemini CLI, Colab, Antigravity,...) but the tutorial should apply to any similar technology out there.



# Things to keep in mind

- **Try again if you had a bad experience in the past:** Tools have evolved remarkably over the past six months, resulting in significantly better performance.
- **Own it and declare it:** Most academic conferences and journals allow the use of AI, provided its application is explicitly declared.
- **Be aware of the pleasing bias:** LLMs are trained to be helpful, which means they will do their best to provide an answer even if it involves producing a highly plausible, yet entirely flawed, proof.

# No magic bullet

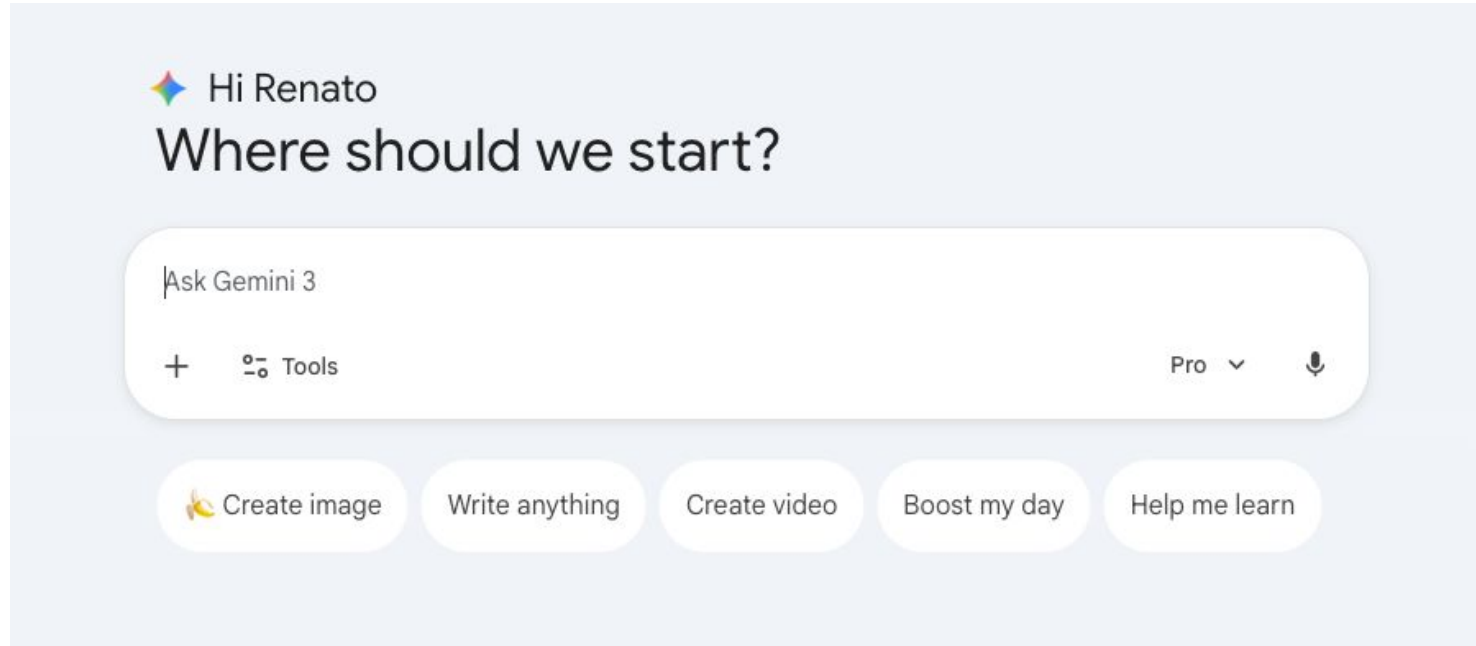
- Research still requires lot of effort (even with AI)
  - Asking the right questions & providing the right intuition
  - Ensuring scientific and mathematical rigor
  - Using the tools efficiently
- ... but you may end up spending time doing other things
  - Some formerly labor-intensive activities become effortless
  - You spend more time trying prompt variations, checking arguments, proof-reading, ...
- **Long run:** With tedious tasks out of the way, we hope AI will help us focus on what matters and allow us to tackle more difficult tasks.

# Plan for this tutorial:

- Day 1: Directly talk to the AI
  - Automating the research workflow
  - Proving results using AI
- Day 2: Automating the use of AI
  - Calling AI programmatically
  - Agentic workflows

# Part 1: How can AI and LLMs help you?

# Getting started: <http://gemini.google.com/>



# Part 1a: Automating the Research Workflow

# Let's read a paper by talking to it

2406.13668v3 (1)  
 PDF

←

Drag and drop a paper

Does the approach outlined in this paper also solve the direct version of the calibration problem (where we don't get the probabilities in advance) or only the minimax version?

+
Pro v
↑

That is an excellent and highly precise question.

The short answer is **no, the approach outlined in this paper only solves the minimax version of the calibration problem**. It does not provide a direct, constructive solution for the standard version where the true probabilities are unknown in advance.

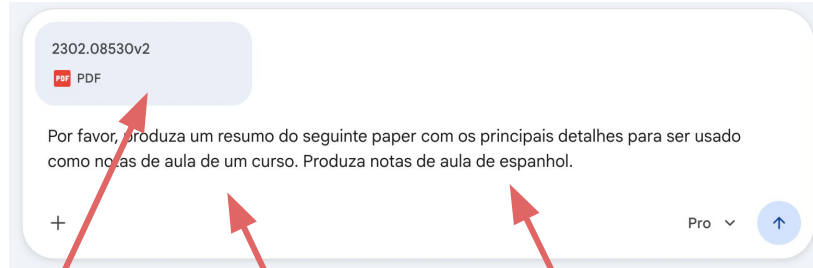
Here is the breakdown of why this is the case, based on the provided text:

### Reliance on the Minimax Reduction

The paper's proposed forecasting algorithm achieves the improved  $O(T^{2/3-\epsilon})$  upper bound

- Ask clarifying questions about the model
  - “How does it compare to the setting in paper X”
  - “Please write the full derivation of equation Y”
  - “Please explain what concentration inequality is used in Lemma Z”
  - ...

# You can talk to the model in any language



Paper in English

Asking for output in Spanish

Question in Portuguese

- You can ask questions in any language (and specify the format and language of the output)

Aquí tienes las notas de clase en español, estructuradas a partir del artículo académico proporcionado para facilitar su uso en un entorno educativo.

## Notas de Clase: Guía Práctica para la Regulación de Presupuestos y Restricciones ROS

Basado en el artículo: *A Field Guide for Pacing Budget and ROS Constraints* por S. Balseiro, et al.



### 1. Introducción y Contexto del Problema

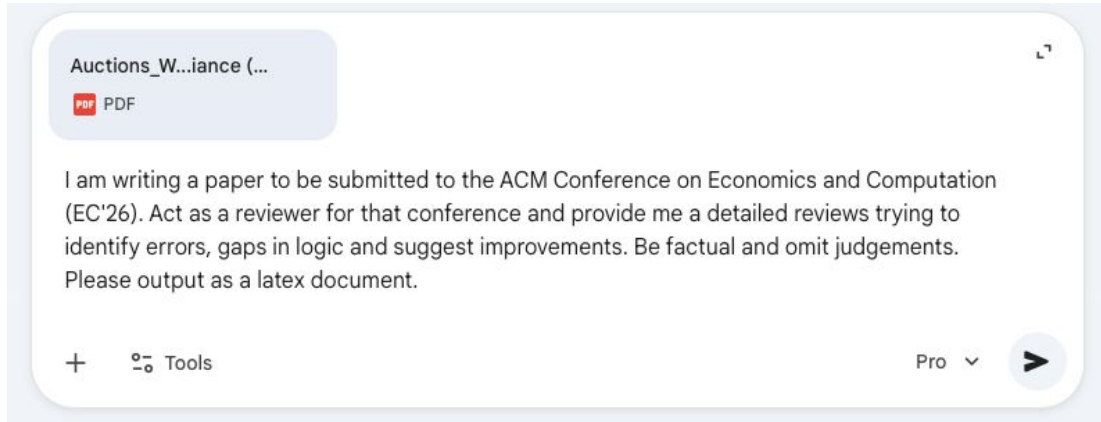
- **Publicidad en Internet y Presupuestos:** Los anunciantes compran oportunidades de publicidad mediante subastas en tiempo real y establecen presupuestos para controlar sus gastos.
- **Budget Pacing (Regulación de Presupuesto):** Es un servicio popular que ajusta dinámicamente las ofertas en tiempo real para maximizar el retorno del anunciante sin superar el presupuesto.
- **Autobidding y Restricciones ROS:** Las herramientas de "autobidding" optimizan el valor en tiempo real estando sujetas a restricciones de Retorno sobre la Inversión publicitaria (ROS, por sus siglas en inglés).
- **El Problema de Coordinación:** Por razones comerciales o históricas, los sistemas que manejan la regulación del presupuesto y del ROS no siempre están unificados.
- **Objetivo del Estudio:** El documento compara algoritmos con diferentes niveles de coordinación entre estos dos sistemas de regulación para encontrar un equilibrio óptimo entre rendimiento y viabilidad técnica.

### 2. Arquitecturas de Regulación de Ofertas (Pacing Services)

El estudio analiza y compara empírica y teóricamente tres enfoques principales:

- **Servicio Óptimo Dual (Totalmente Acoplado):** Es un sistema unificado que optimiza un

# Let's review a paper



- Give as many details as possible
  - Which conference/journal you are submitting
  - What should reviewer focus on? (Grammar, math errors, gaps in logic, missing references, ...)
  - Maybe specify something about the reviewer (adversarial, unconvinced about your research agenda, looking for reasons to reject, ...)
  - It is useful to specify the output (latex document)

Here is a review of the paper "Auctions With Small Variance" formatted as a LaTeX document, suitable for the ACM EC'26 conference context.

Code snippet



```
\documentclass[11pt]{article}
\usepackage[utf8]{inputenc}
\usepackage{geometry}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{enumitem}

\geometry{a4paper, margin=1in}

\title{Review of "Revenue Variance Minimization: Beyond First Price Auctions"}
\author{ACM EC'26 Review Committee}
\date{}

\begin{document}

\maketitle

\section*{Summary of the Paper}

This paper investigates the minimization of revenue variance in auction settings

The paper's main contributions are:
\begin{enumerate}
\item \textbf{Single-Item Interim IR:} The authors show that while WPB remain
\item \textbf{Multi-Item Interim IR:} For symmetric regular environments, the
```

Copy and paste to Overleaf or your favorite LaTeX editor.

Or alternatively use Canvas:

Let's write or build together



Canvas ×

please create the same review in canvas

◆ Show thinking ▾

I have converted the review into a LaTeX file that is ready for compilation or submission.

Review of Revenue Variance  
Minimization  
Jan 30, 6:02 PM

[Try again without Canvas](#)



Let's write or build together

Review of Revenu...



Code



Preview



Share



## Review of "Revenue Variance Minimization: Beyond First Price Auctions"

ACM EC'26 Review Committee

January 30, 2026

### Summary of the Paper

This paper investigates the minimization of revenue variance in auction settings with risk-neutral, unit-demand bidders and independent private values. The authors aim to generalize the classical result of Waehrer et al. (1998), which established the optimality of the First-Price (or Winner-Pays-Bid, WPB) auction for variance minimization under ex-post Individual Rationality (IR).

The paper's main contributions are:

- Single-Item Interim IR:** The authors show that while WPB remains optimal for i.i.d. valuations under interim IR, this optimality breaks down in asymmetric settings. They characterize the variance-minimizing payment rule for asymmetric cases using a decomposition technique and prove existence using fixed-point theorems in Banach spaces.
- Multi-Item Interim IR:** For symmetric regular environments, the authors construct a mechanism (derived via a differential equation) that achieves zero revenue variance while maintaining non-negative payments and interim IR.
- Multi-Item Ex-post IR:** The authors demonstrate via a counterexample that WPB is not variance-minimizing in multi-item settings (specifically 2 items, 3 bidders). However, they prove that the discriminatory WPB auction still yields lower variance than the uniform ( $k+1$ )-st price auction without requiring log-concavity assumptions.

### General Comments



# Examples of AI feedback

## 2. Numerical vs. Analytical Rigor in Section 5

While Sections 3 and 4 rely on rigorous analytical derivations (decomposition techniques, ODEs), Section 5 relies heavily on numerical examples (e.g., the counterexample in Section 5.1 and the discretized optimization in 5.2).

- In Section 5.1, the variance reduction is shown via numerical integration ( $\approx 0.26431$  vs  $\approx 0.2613$ ). While the example is illustrative, an analytical bound or a local perturbation argument would strengthen the claim that WPB is strictly suboptimal.
- In Section 5.2, the optimal mechanism is approximated via discretization. Given Lemma 5.1 reduces the problem to functions of order statistics, is it possible to derive the Euler-Lagrange equations for the calculus of variations problem defined in [582-586]? Even if no closed form exists, characterizing the properties of the solution analytically would be superior to a purely numerical plot.

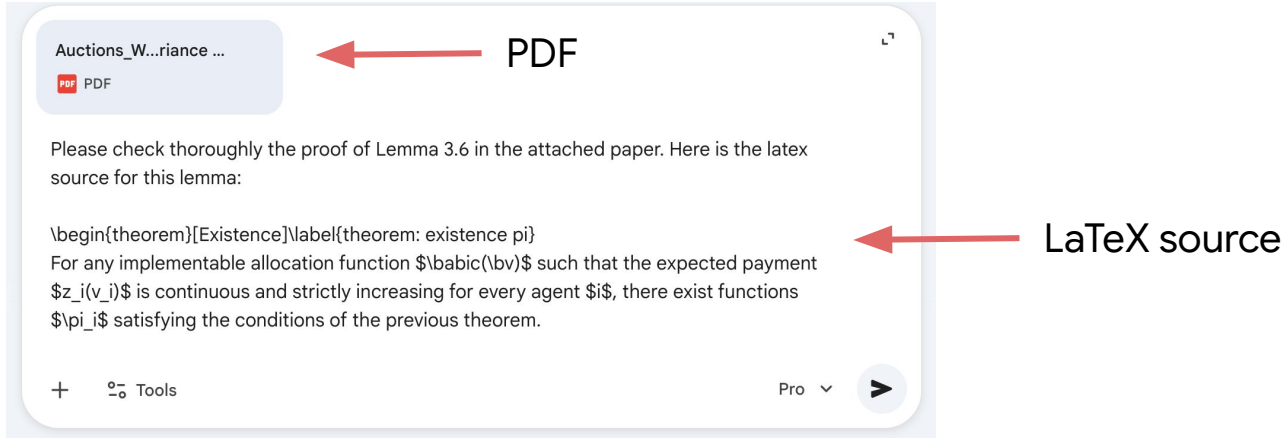
## 3. Theorem 3.6 (Existence via Fixed Point Theorems)

The proof of existence for the asymmetric variance-minimizing mechanism utilizes Schauder's fixed point theorem on the space of probability measures. While technically impressive, this machinery seems heavy for the problem at hand. Is the operator  $T$  a contraction under some metric? If so, the proof could be simplified, and potentially imply uniqueness or convergence of simple iterative algorithms. The authors should comment on the uniqueness of the solution  $\pi_i(v_i)$ , as Schauder does not guarantee it.

## Minor Comments and Typos

- **General:** The citation style uses bracketed numbers [1], but the text often refers to "Waehrer et al. (1998)". Consistent formatting is recommended.
- **Page 13, Eq 306:** The text reads "...evaluates to evaluates to...". Duplicated words.
- **Page 13, Eq 307 & Page 14, Eq 319:** The use of  $\approx$  without error bounds or exact expressions in an analytical section is slightly jarring.
- **Page 16, Footnote 1:** The footnote regarding jumps and countable discontinuities appears disconnected from the main text flow or is formatted essentially as a sentence fragment at the bottom of the page.
- **Page 20, Step 3:** The integral substitution uses  $\frac{d}{dx}[x(1-F(x))]$ . Ensure the sign conventions in the subsequent integration by parts are clear, particularly the negative sign on  $(n-k)\phi(x)$ .
- **Page 25, Eq 551:** The definition of  $b_1(v_i)$  appears to have formatting issues in the submitted PDF (e.g., "*Vis  $U \leq 0.6$* "). The conditional cases are hard to parse.
- **Page 27, Figure Captions:** Figures 3, 4, and 5 have redundant or unclear labels (e.g., "*#1 Highest Bidder*", "*Highest Bidder*"). The axis labels are too small to be legible.
- **References:** Ensure all arXiv/SSRN preprints (e.g., Pekeč and Smilgins) are updated to the latest versions if published.

# Keep in mind the “thinking budget”



Auctions\_W...riance ... PDF

Please check thoroughly the proof of Lemma 3.6 in the attached paper. Here is the latex source for this lemma:

```
\begin{theorem}[Existence]\label{theorem: existence pi}
For any implementable allocation function  $\mathbf{b}(\mathbf{v})$  such that the expected payment  $z_i(v_i)$  is continuous and strictly increasing for every agent  $i$ , there exist functions  $\pi_i$  satisfying the conditions of the previous theorem.
```

+ Tools Pro ▶

- The model has limited “attention” and time spent thinking
- It is useful to make the model focus on a specific part of the paper
- PDF is fine, but LaTeX source is even better since there are no errors introduced by PDF parsing

# Copy-editing

Act as an academic copy editor for ACM EC. Edit the following text to improve grammar, flow, and idiomatic English. Constraints:

\* Idiomatic Phrasing: Fix any non-native phrasing or awkward sentence structures to ensure standard American academic English.

\* Clarity & Flow: Ensure the motivation transitions logically into the technical contribution.

\* Voice Preservation: Retain my original tone and terminology. Do not over-simplify the technical concepts.

+  Tools

Pro 



- Copy-editing works best with small section at a time
- Give the full context, conference, topics...
- Directions such as “keep my original voice”, “make it idiomatic”, ...
- “Keep the sentence structure relatively similar to the original unless a change significantly improves clarity”

# Prompt for prompting

Here is a copy-editing prompt use to re-write an introduction of a paper. Help me improve the prompt and create a short prompt template I can use for this type of situation.

"I am working on the introduction of a paper with the provisional title of "Hierarchical Markets with Networked Resource Conversion" to be submitted to ACM EC. I'll paste the intro below. Please help me edit and make it more polished for the EC audience. Help me make the grammar and language better and more native but keep it true to my original voice. Here is

+ 🛠️ Tools

Pro ▾



- If unsure what prompt to use, use Gemini itself to suggest the prompt.
- It is useful to keep a list of “prompt templates” that worked for you in the past.

# Let's draw a diagram

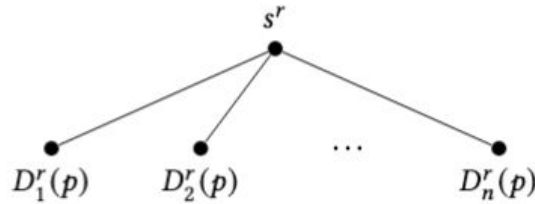
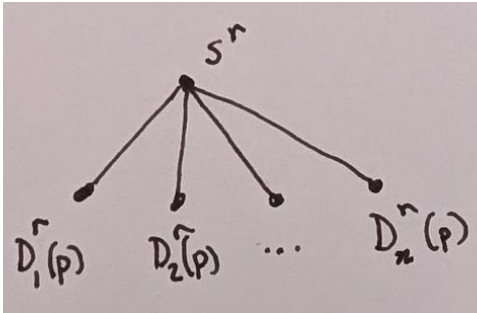


Drag and drop a photo

please create a tikz version of the following image:

+ Tools

Fast



Code snippet

```
\documentclass[tikz, border=2pt, png]{standalone}
\usepackage{tikz}
\usetikzlibrary{trees}

\begin{document}

\begin{tikzpicture}[
  level distance=3cm,          % Vertical distance between levels
  sibling distance=2.5cm,      % Horizontal distance between nodes
  dot/.style={circle, fill=black, inner sep=2pt}, % Style for the black dots
  every label/.style={font=\small} % Font size for labels
]

% Root node
\node [dot, label=above:$S^n$] {}
% First child
child { node [dot, label=below:$D_1^r(p)$] {} }
% Second child
child { node [dot, label=below:$D_2^r(p)$] {} }
% The "... node. We use edge from parent[draw=none] to not draw a line to it
child { node {\dots} edge from parent[draw=none] }
% Last child
child { node [dot, label=below:$D_x^n(p)$] {} };

\end{tikzpicture}

\end{document}
```


# Verifying equations

Given a constant  $C \in [0,1]$ , define a function  $z(v) = \dots$

Write a program in python using sympy to perform symbolic integration and verify that the following equation holds:

$$z(v) = \int_0^v P_1(u) \frac{1}{v^2} du + \int_v^1 P_2(w, v) \frac{1}{1-v} dw$$

Canvas Pro

- For specialized tasks (solve symbolic equations), ask Gemini to write a python program to solve it.
- You can run in canvas itself 
- Or open in a Google colab: <https://colab.research.google.com/>

```

Integral Verification
1 import sympy
2 from sympy import symbols, integrate, simplify, Rational
3
4 def verify_equation():
5     # Define symbols
6     # v: variable of the function z(v)
7     # u: integration variable for the first integral (0 to v)
8     # w: integration variable for the second integral (v to 1)
9     # C: constant in [0, 1]
10    v, u, w, C = symbols('v u w C', real=True)
11
12    print("--- Defining Functions ---")
13    # Target function z(v)
14    z_target = v**2 - Rational(2, 3) * v**3
15    print(f"Target z(v) = {z_target}")
16
17    # P1 definition
18    # If v1 + v2 <= C: v1
19    # If v1 + v2 >= C: C/2 + (v1 - v2)/2 + (1/(20))*(v1 + v2 - C)^2
20    def P1_func(v1, v2, condition):
21        term_low = v1
22        term_high = C/2 + (v1 - v2)/2 + (1/(2*C))*(v1 + v2 - C)**2
23        return sympy.Piecewise((term_low, condition == 'low'), (term_high, condition == 'high'))
24
25    # P2 definition
26    # If v1 + v2 <= C: v2
27    # If v1 + v2 >= C: C/2 - (v1 - v2)/2 - (1/(20))*(v1 + v2 - C)^2
28    def P2_func(v1, v2, condition):
29        term_low = v2
30        term_high = C/2 - (v1 - v2)/2 - (1/(2*C))*(v1 + v2 - C)**2
31        return sympy.Piecewise((term_low, condition == 'low'), (term_high, condition == 'high'))
32
33    # Helper for explicit expressions to avoid Piecewise integration issues
34    def get_P1_low(v1, v2): return v1
35    def get_P1_high(v1, v2): return C/2 + (v1 - v2)/2 + (1/(2*C))*(v1 + v2 - C)**2
36    def get_P2_low(v1, v2): return v2
37    def get_P2_high(v1, v2): return C/2 - (v1 - v2)/2 - (1/(2*C))*(v1 + v2 - C)**2
38
39    print("\n--- Starting Verification for 3 Cases ---")
40
41    # -----
42    # CASE 1: v < C/2
43    # -----
44    # Implications:
45    # 1. v < C/2 implies 2v < C.
46    # 2. Integral 1 (u: 0->v): u + v <= 2v < C. Always 'low' case for P1.
47    # 3. Integral 2 (w: v->1):
48    #     Split point w s.t. w + v = C is w = C - v.
  
```

# Verifying equations

Symbolic Verification ☆ ☁

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text ▶ Run all -

Connect

```

import sympy as sp

def verify_integration_identities():
    print("Initializing Symbolic Variables...")
    # Define symbols
    # v: the variable for z(v) and integration limit
    # u: dummy variable for the first integral
    # w: dummy variable for the second integral
    # C: constant capacity parameter
    v, u, w, C = sp.symbols('v u w C', real=True, positive=True)

    # -----
    # 1. Define the Functions
    # -----

    # Target function LHS
    # z(v) = v^2 - (2/3)v^3
    z_target = v**2 - sp.Rational(2, 3) * v**3

    # P1 Piecewise Definitions
    # P1 is v1 when v1+v2 <= C
    def P1_low(v1, v2):
        return v1

    # P1 high branch
    def P1_high(v1, v2):
        term1 = C / 2
        term2 = (v1 - v2) / 2
        term3 = (v1 + v2 - C)**2 / (2 * C)
        return term1 + term2 + term3

    # P2 Piecewise Definitions
    # P2 is v2 when v1+v2 <= C
    def P2_low(v1, v2):
        return v2
  
```

verify\_integration\_identities()

```

... Initializing Symbolic Variables...

-----
Starting Verification Steps
Target z(v) = -2*v**3/3 + v**2
-----

>>> Checking Case 1: v < C/2
[FAILURE] Difference: (C**3 + 8*C*v**4 - 20*C*v**3 + 33*C*v**2 - 30*C*v - 3*C + 2*v**3 + 6*v**2 + 6*v)

>>> Checking Case 2: C/2 < v < C
[FAILURE] Difference: -C**3/(4*v**2) + 5*C**2/(6*v) + 2*v**3/3 - v**2 + 3*v/4 + 1/4 + v**2/(6*C) - 2*v

>>> Checking Case 3: v > C
[FAILURE] Difference: C + 2*v**3/3 - v**2 + v/4 + 1/4 + v**2/(4*C) - 2*v/(3*C) - 1/(6*C)
  
```

# Implementing Algorithms

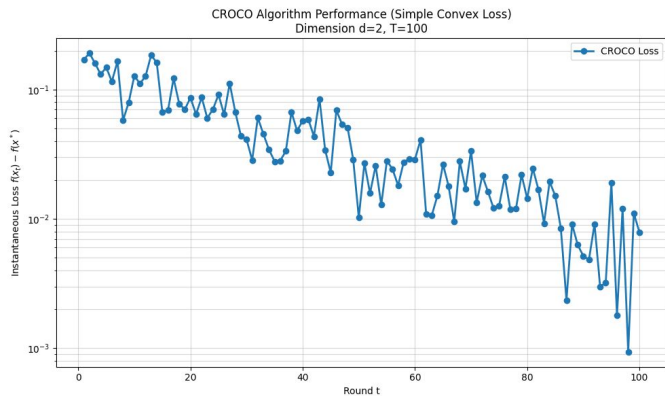
MOR\_\_cor...ual\_se...

PDF PDF

Please provide an implementation of the Algorithm 1 "Log-Concave Density Algorithm for CROCO" in the attached paper in python. Use MALA to compute the centroid of a log-concave distribution. Also produce a plot showing the value of a loss function of a simple convex function as a function of the number of iterations.

+ 🗨️ Canvas ×

Pro ▾



Not what I  
expected but  
90% there.

Corruption Robust Convex Optimization

Export to Colab

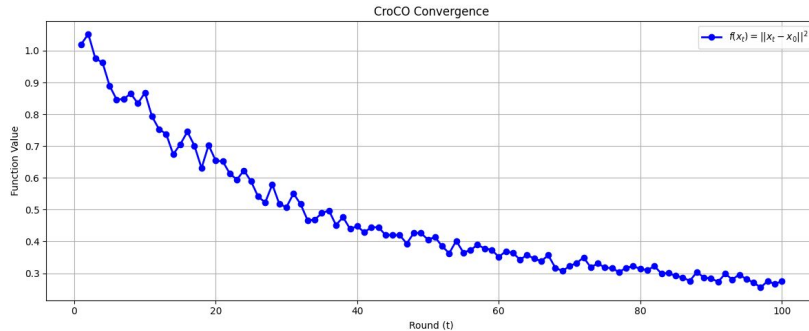
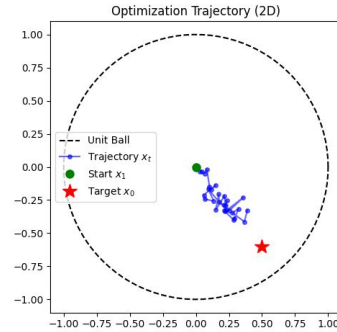
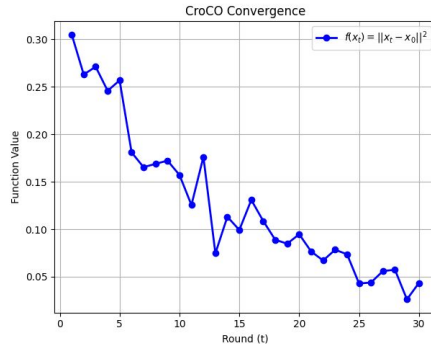
```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from tqdm import tqdm
4
5 class LogConcaveDensityCROCO:
6     def __init__(self, dim, L, D, gamma_scale=3.0):
7         """
8         Initializes the CROCO algorithm.
9
10        Args:
11            dim (int): Dimension of the problem.
12            L (float): Lipschitz constant of the function.
13            D (float): Diameter of the domain K.
14            gamma_scale (float): Scaling factor for gamma. Paper suggests gamma = 1/(3LD).
15        """
16        self.dim = dim
17        self.L = L
18        self.D = D
19        # Gamma setup as per Theorem 1: gamma = 1 / (3LD)
20        self.gamma = 1.0 / (gamma_scale * L * D)
21
22        # History of queries (centroids) and feedback (gradients)
23        # x_history[i] corresponds to x_{i+1} in paper notation (0-indexed)
24        self.x_history = []
25        self.grad_history = []
26
27        # MALA parameters
28        self.mala_step_size = 0.05 / np.sqrt(dim) # Heuristic scaling
29        self.last_sample = np.zeros(dim) # Start chains from previous point (warm start)
30
31    def log_density_unnormalized(self, x):
32        """
33        Computes the log of the unnormalized density mu_t(x).
34        mu_t(x) = mu_1(x) * Product_{s=1}^{t-1} (1 - gamma * <grad_s, x - x_s>)
35
36        Returns -inf if x is outside the domain (Unit Ball).
37        """
38        # Check domain constraint (Unit Ball)
39        if np.linalg.norm(x) > 1.0:
40            return -np.inf
41
42        # Base density mu_1 is uniform (log constant), we ignore the constant.
43        if not self.x_history:

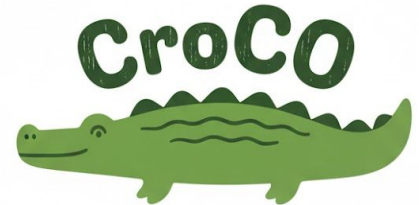
```

# Implementing Algorithms

- After a few rounds of interaction (improving prompts, manually modifying the code and the presentation):



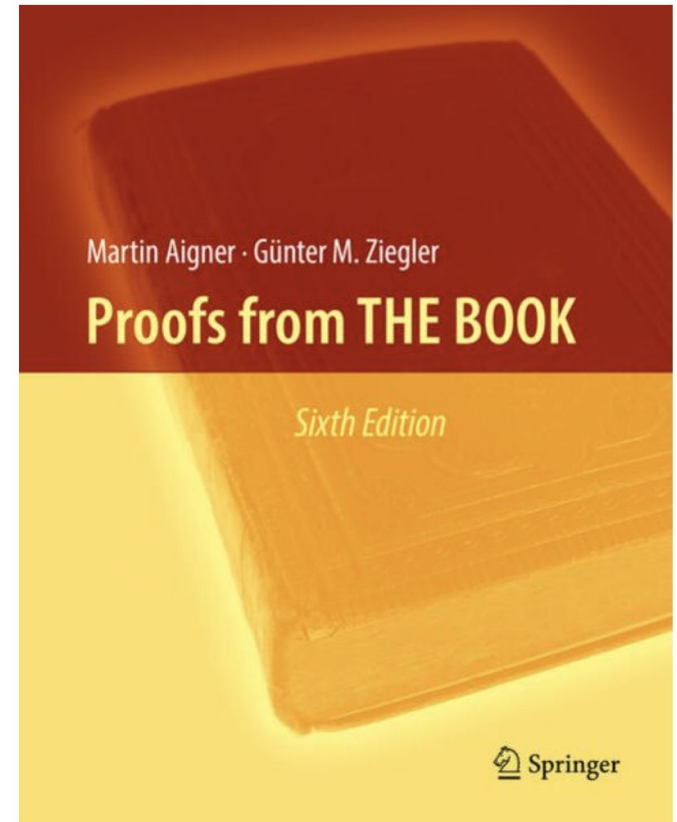
- Usually requires verifying the code, fixing mistakes, tweaking parameters, ...
- Some of that is manual and some of that you can ask the model to do using follow-up prompts.
- Bonus:** “Can you design a logo for the algorithm:”



# Part 1b: Proving Results Using AI

# The art of theorem proving

- Proving new results is as much art as it is science
- Requires mathematical rigor
- But also relies on creativity, intuition, mathematical taste, a sense of beauty, the ability to see hidden connections ...



# Two types of theorem proving

- **Type 1:** “I don’t know the proof of that statement but the right mathematician probably knows it”

This is the domain where AI tends to be more useful. However, guiding the AI to a rigorous proof may still require careful human steering.

- **Type 2:** It is truly an open problem for the research community.

AI is beginning to make progress here, but this is significantly harder. Success typically requires combining human intuition with structured AI workflows.

# Type 2 success stories

- Formal reasoning & verification
  - Olympiad-level formal mathematical reasoning with reinforcement learning ([Nature](#))
- Structured Human-AI workflows
  - Semi-autonomous mathematics discovery with Gemini: A Case Study on the Erdős Problems ([arXiv](#)) Also see [Wiki](#)
- Algorithmic search & evolutionary discovery
  - Mathematical exploration and discovery at scale ([arXiv](#))
  - AlphaEvolve: A coding agent for scientific and algorithmic discovery ([arXiv](#))
- Agentic loops orchestrating research tasks
  - Early science acceleration experiments with GPT-5 ([arXiv](#))
  - Accelerating scientific research with Gemini: Case studies and common techniques ([arXiv](#))

# Techniques that are usually helpful

**The AI as a smart PhD student:** What advice / hints would you give a student working on this problem?

- Ask the AI to brainstorm possible proof techniques, or provide intuition on how a proof might go.
- Give as much detail as possible (what special cases are known, impossibilities, potentially some useful lemmas).
- Add related references to the context (“this may be related to Lemma ... in the PDF attached”).

I am working on the attached project on combinatorial contracts. I am trying to address a gap in one of the proofs. This is Lemma `lem:subset-approx`, in the final step. Here the equality in the final step does not hold, but we hope that we can get rid of the constant. See the comment `\pdc{...}` in the source file. Can you help me develop ideas for how to approach this? ^

[this did not work]

Let me propose a different direction. There is recent literature (e.g., Multi-Agent Combinatorial Contracts, SODA 2025) which studies a different contracting problem, but encounters a similar challenge. There the authors solve this differently: They decompose the solution in a constant number of small agents, and then partition the remaining agents into a small number of sets. This way they achieve two things at once: (a) ... and (b) ....

[this worked eventually]

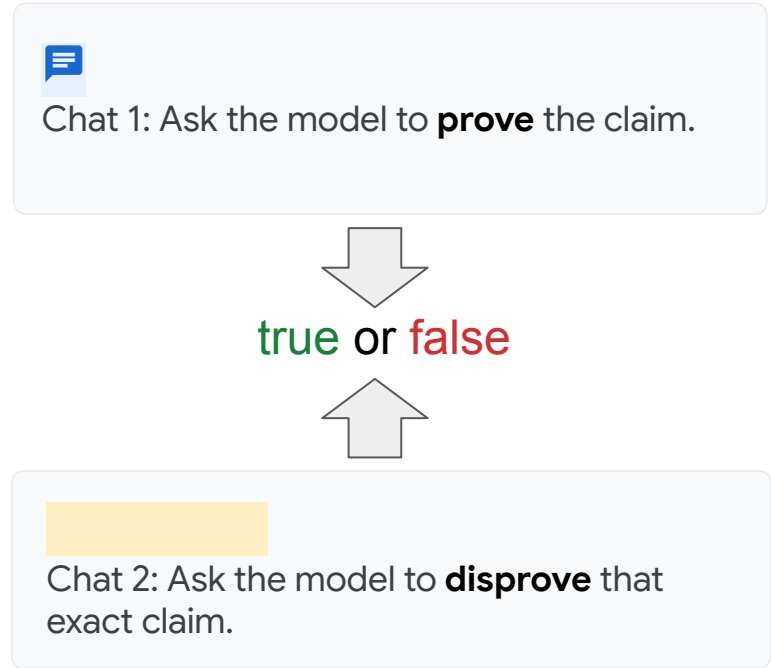
# Techniques that are usually helpful

**The primal-dual approach:** If you ask an AI to “help prove” a statement, it may try to force a proof even if the statement itself is wrong.

To avoid this, consider running two separate chats in parallel:

- **Chat 1 (“The Primal”):** Ask the model to prove the claim.
- **Chat 2 (“The Dual”):** Ask the model to disprove that exact claim.

Even if neither of the two “succeed”, it can be an effective tool to identify the mathematical core of the problem.

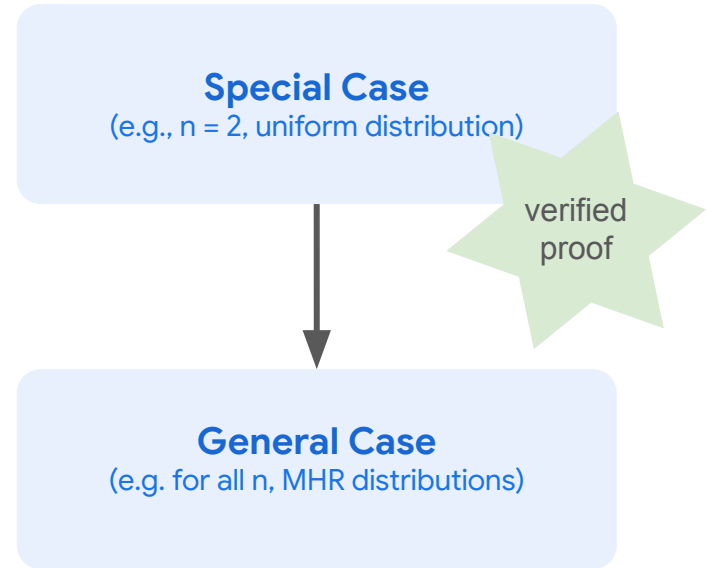


# Techniques that are usually helpful

**From specific to general:** First prove a special case (e.g.,  $n = 2$  or assume the uniform distribution), either “by hand” or with the help of AI.

Iterate on the proof for the restricted case until you are fully satisfied with its logical soundness and rigor.

Finally, ask the AI to use your verified proof as a template to generalize to the broader case.



# Techniques that are usually helpful

**Create a “cheat sheet” (context injection):** Ask the AI to create a summary of useful facts and properties about a mathematical object.

For example: “produce a Latex document with a collection of useful lemmas and properties of regular/MHR distributions”

You can then feed this “cheat sheet” back to the AI to tackle the problem you want to solve.

User • 12:02 PM

I want to create a "cheat sheet" with useful properties of regular distributions (in the sense of Myerson). 1. Please find a few important references from the recent literature in algorithmic mechanism design. 2. Scan these papers and potential sources they cite to identify some properties and useful lemmas of regular distributions. 3. Combine a 3-4 page long Latex document which states the definition of a regular distribution and lists the useful properties you identified. 4. Carefully review and possibly update this document. Check for consistent notation and correctness, and correct attribution of the results. 5. Return the revised Latex as a self-contained Latex document that I can download and compile.

## 1. Literature Review & Reference Selection

To build this document, I identified foundational literature and heavily cited modern algorithmic mechanism design that establish or extensively rely on regularity:

- **Myerson (1981):** Introduced the definitions of virtual valuation and regularity.
- **Bulow & Roberts (1989):** Translated Myerson's theory to "quantile space," establishing that regularity is mathematically equivalent to revenue curve concavity.
- **Bulow & Klemperer (1996):** Demonstrated that market expansion (adding a bidder) strictly supersedes optimal algorithm design (knowing the distribution to set the optimal reserve).
- **Chawla, Hartline, Malec, & Sivan (STOC 2010):** Leveraged regularity to reduce complex multi-parameter auctions to sequential posted pricing using prophet inequalities.
- **Dhangwatnotai, Roughgarden, & Yan (EC 2010 / JACM 2015):** Relied on the concavity of regular revenue curves to prove that a single random sample yields a  $\frac{1}{2}$ -approximation of optimal revenue.
- **Cole & Roughgarden (STOC 2014):** Formalized the "Pareto tail bound" of regular distributions to prove tight bounds on the sample complexity of empirical revenue maximization.

## 2. Extracted Properties & Lemmas

From scanning these sources, I identified several highly useful properties:

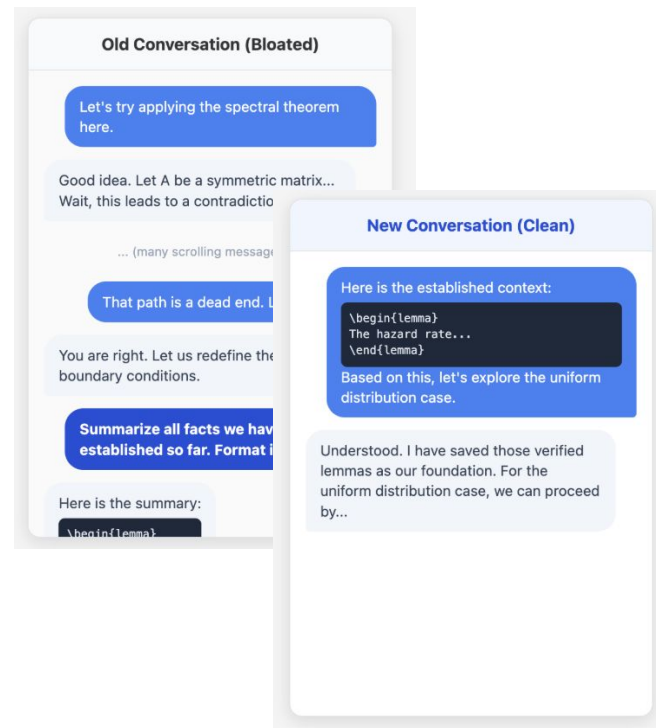
# Techniques that are usually helpful

## Resetting context (manage context window):

Over long derivations, the model can get confused by past mistakes or tangential ideas; the model has "too much on its mind."

Ask the AI to summarize all verified progress and facts established so far. (Requesting this in LaTeX or Markdown format makes it easy to copy).

Open a completely new chat window, paste only this refined summary, and continue the work with a fresh context window.



# Techniques that are usually helpful

**Multi-step instructions (chain of thought prompting):** If you ask for a final proof immediately, the AI often skips logical steps. Force the model to plan first.

E.g.: After an AI review, it's often useful to follow up and ask for a LaTeX formalization of the proposed fix:

Can you help me address the bullet " $\text{\textbf{...}}$ ". How would you change the proof of this  $\wedge$  <Lemma/Proposition/Theorem> to formalize the argument that you make in the review? Proceed as follows:

1. Review the critique and the proposed fix.
2. Plan how to update the proof.
3. Write out a formal proof in Latex.
4. Very carefully check the proof you are proposing. If there are any issues fix them.
5. Return the LaTeX code for the new proof.

*"The complete LaTeX code for the streamlined proof is provided above, and can safely overwrite the entire original proof."*

Brilliant idea!

# Combining techniques: A case study

We recently succeeded in proving Theorem F.3 below through a sequential AI workflow: an initial **AI review** to catch a subtle flaw, a variation of the '**specific to general**' approach to adapt the logic of Theorem 5.1, and **multi-step instructions** to obtain a first proof sketch.

**Theorem 5.1** (Hardness of optimal contracts). *The  $k(n)$ -ambiguous contract problem is NP-hard for all functions  $k : [n] \rightarrow [n]$  such that  $(n, k(n))$ -MAKESPAN MINIMIZATION is NP-hard.*

**Theorem F.3** (Hardness of optimal monotone contracts). *The  $k(n)$ -ambiguous contract problem with monotonicity constraints is NP-hard for all functions  $k : [n] \rightarrow [n]$  such that  $(n, k(n))$ -MAKESPAN MINIMIZATION is NP-hard.*

arXiv > cs > arXiv:2503.02592

Computer Science > Computer Science and Game Theory

[Submitted on 4 Mar 2025]

**Succinct Ambiguous Contracts**

Paul Duetting, Michal Feldman, Yarden Rashti

# Other use cases

- “Can you check if an argument like this one exists in the literature?”
- “I have this vague setup/approach in mind. Can you re-define it in terms of existing concepts in the literature?”



OK, attaching a draft of my rewrite of section 5.

The algo is more streamlined (fewer oracle calls, only one Seymour algo call, no explicit checks for 3-connectedness).

I discovered that the thing we're calling id-path maps to a concept called theta matroid. A theta matroid is the  $P_1, P_2, P_3$  structure where each pair is a circuit that we use to define 3-connectivity.

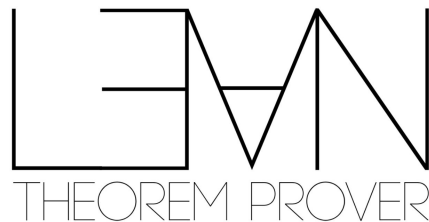
# Using Gemini with Lean for formal verification

[Also see this [ETH tutorial](#)]

Many well-established proof assistants:

- Agda
- Rocq
- HOL
- Isabelle
- Mizar
- Lean

Here we will use **Lean** [<https://lean-lang.org/>], which has an extensive library of already-formalized mathematics (mathlib).



# Direct code generation (Manual workflow)

- You can provide Gemini with an informal mathematical theorem (in natural language or LaTeX) and ask it to write the equivalent Lean 4 code.
- **Tip:** Use **Gemini 3 Pro** or **Ultra** for this. These models are better at "Long Chain-of-Thought" (CoT) reasoning, which helps them plan the proof structure before writing the tactics.
- **Limitation:** Gemini may "hallucinate" lemmas or syntax that doesn't exist in [mathlib4](#). You will need to manually copy the code into your IDE to see if it compiles.

Let's try to prove:

**Theorem:** If  $R$  is a ring, and  $a \in R$  has both a right and a left inverse, then these inverses are equal.

**(Informal) Proof:**

Let  $b$  and  $c$  be the left and right inverses of  $a$ , then

$$b = b * 1 = b * (a * c) = (b * a) * c = 1 * c = c$$

as claimed.

# Gemini-generated Lean proof

Here's a mathematical theorem that I would like to prove. [Start Theorem] Theorem: If  $R$  is a ring, and  $a \in R$  has both a right and a left inverse, then these inverses are equal. [End Theorem] Here's an informal proof: [Begin Proof] Proof. Let  $b$  and  $c$  be the left and right inverses of  $a$ . Then we have  $b = b * (a * c) = (b * a) * c = c$ . QED. [End Proof] Can you help me write that proof in Lean 4 so that I can see if it compiles?

Important to tell it to use Lean v4

Show thinking ▾

That is a perfectly constructed informal proof! It relies on the core properties of multiplication: the identity element and associativity.

Here is how you can write that exact same logic in Lean 4. The `calc` environment is perfect for this because it lets us write formal proofs that mirror the step-by-step equalities of your informal proof.

Lean



```
import Mathlib

theorem left_right_inverse_eq {R : Type*} [Ring R] (a b c : R)
  (h_left : b * a = 1) (h_right : a * c = 1) : b = c := by
  calc
    b = b * 1      := by rw [mul_one]
    _ = b * (a * c) := by rw [h_right]
    _ = (b * a) * c := by rw [← mul_assoc]
    _ = 1 * c      := by rw [h_left]
    _ = c          := by rw [one_mul]
```


Copy code

# Does it compile?

Let's go to <https://live.lean-lang.org/> and copy-paste-the Lean code there:

```
Latest Mathlib [v] ★ Examples ⤴ Load ☰
1 import Mathlib
2
3 theorem left_right_inverse_eq {R : Type*} [Ring R] (a b c : R)
4   (h_left : b * a = 1) (h_right : a * c = 1) : b = c := by
5   calc
6     b = b * 1      := by rw [mul_one]
7     _ = b * (a * c) := by rw [h_right]
8     _ = (b * a) * c := by rw [← mul_assoc]
9     _ = 1 * c      := by rw [h_left]
10    _ = c          := by rw [one_mul]
```

▼ MathlibDemo.lean:10:38 ⚙ ||  
▼ Tactic state ⚙  
No goals  
There is no goal to solve!  
▶ All Messages (0) ||

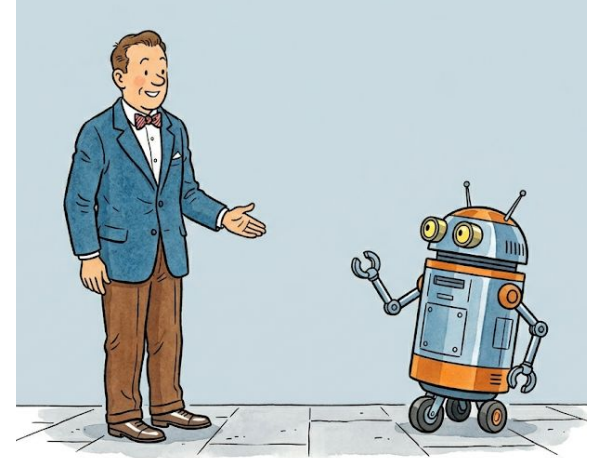


Yes!

(if not, you will see error messages here)

# Summary and Outlook

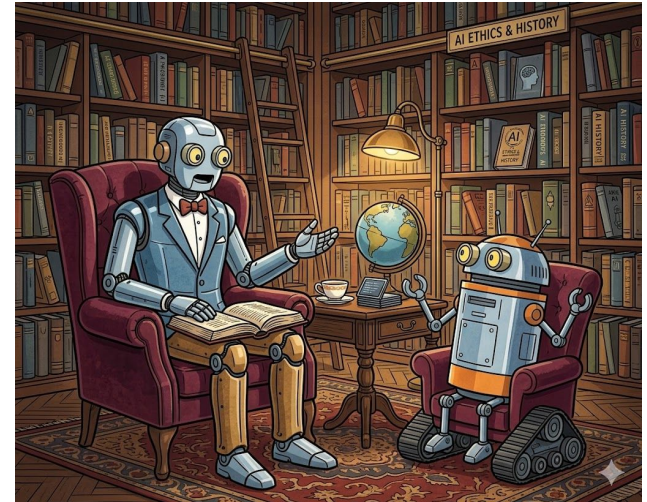
- What we did today:
  - We focused on “directly talking to the AI”
  - We saw many examples of how AI can automate the research workflow
  - And we discussed ways in which AI can assist in proving new results
- Tomorrow we will talk about:
  - Programmatic access to AI / agentic approaches
  - New aspect: working with data



Directly talk to the AI

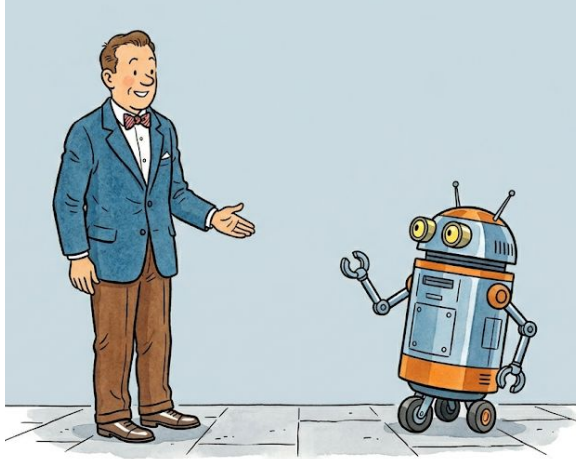
# Teaser for Part 2

- Structuring AI calls:  
How to write code that interacts with AI APIs (Python, CLI).
- Agentic workflows:  
How to define and execute complex research workflows using Antigravity.
- Getting AI to interact with a database:  
How to give the AI access to real-world data (e.g., [datos.gob.cl](https://datos.gob.cl)).

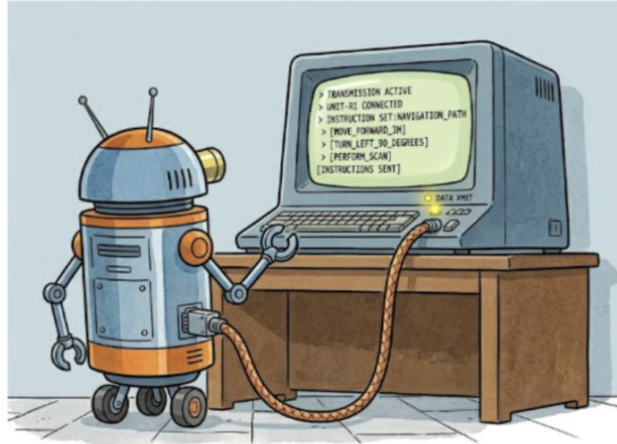


# Part 2: Automating AI use

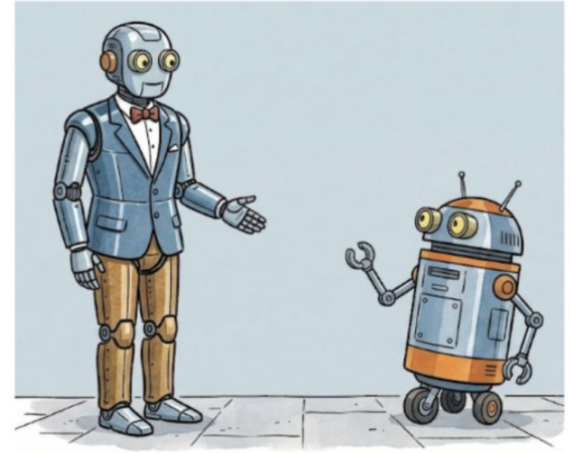
# Automate your interaction with AI



Directly talk to the AI

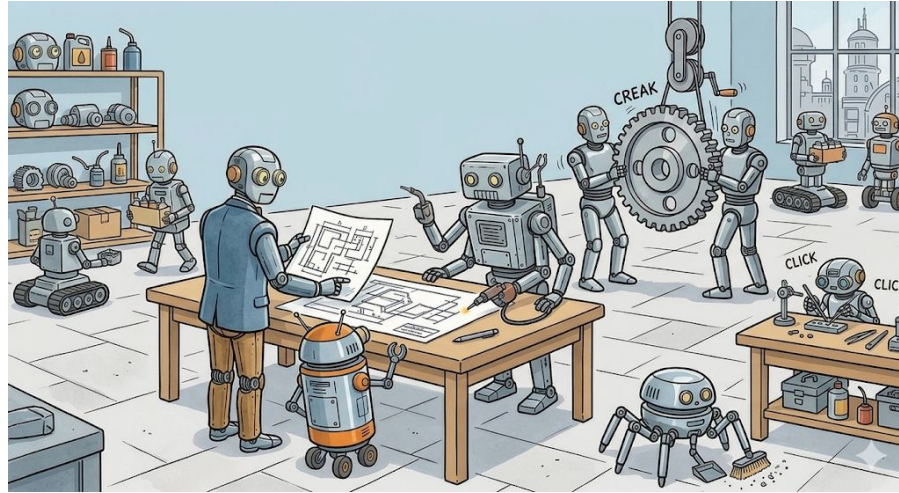


Write programs that call the AI



Get AI to direct other AIs

# Automate your interaction with AI



Towards: Communities / Swarms of AI agents

# Plan for Part 2

- Calling the AI programmatically
  - How to write code that interacts with AI APIs (Python, CLI).
- Agentic research workflows
  - How to define and execute complex research workflows using Antigravity.

# Part 2a: Calling the AI programmatically

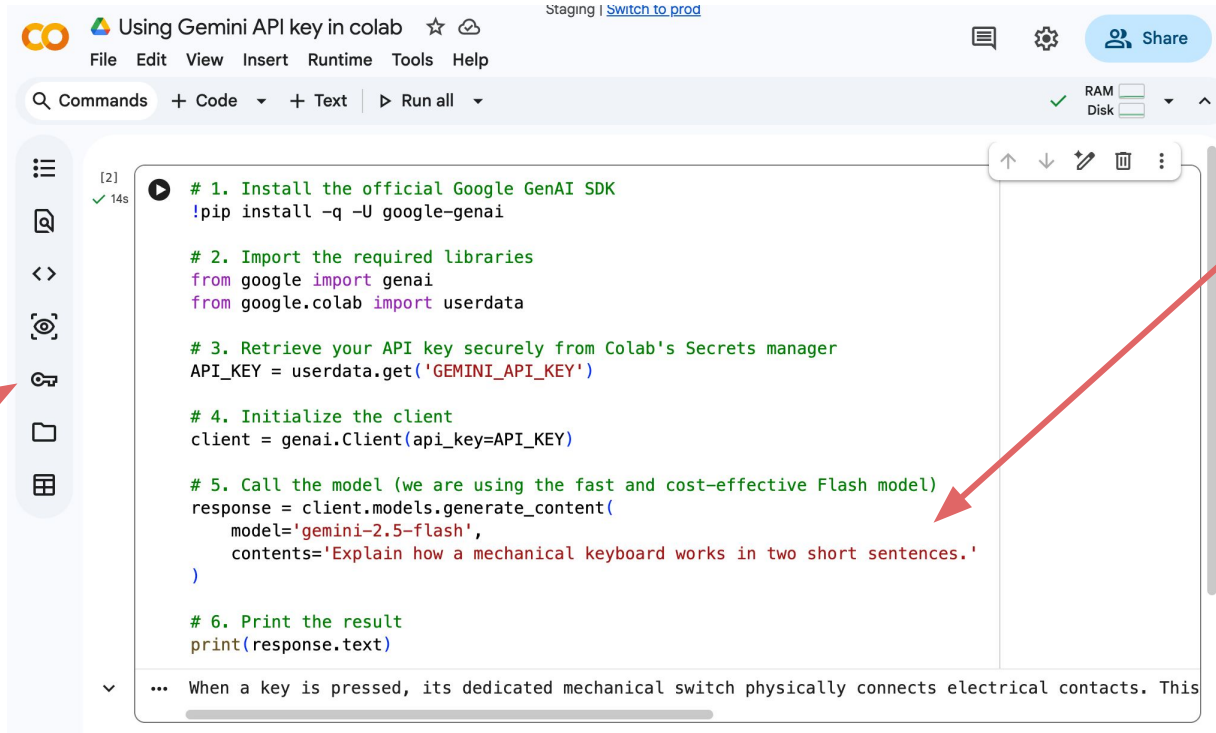
# Using the Python SDK

- Go to Google AI Studio (<https://aistudio.google.com/>) and create an API key

The screenshot shows the Google AI Studio interface. On the left, the navigation menu includes 'Playground', 'Build', 'Dashboard', 'Documentation', 'Search', 'What's new', 'Get API key' (indicated by a red arrow), 'Settings', and the user profile 'renatoppl@gmail.com'. The main content area is titled 'Playground' and features a notification: 'New: Control your API cost. Set a monthly spend cap to prevent accidental overspending. Create a spend cap'. Below this, there are cards for 'Our top picks including Gemini 3 Pro and Nano Banana Pro.' and 'Build chatbots, agents, and code with Gemini 3 Pro and Gemini 3 Flash.' A 'Video Generation' card is also visible. At the bottom, a text input field contains the prompt 'Start typing a prompt, use option + enter to append' and buttons for 'Tools', a microphone icon, a plus icon, and a 'Run' button with a keyboard shortcut icon.

# Using the Python SDK

- Go to Google Colab (<https://colab.research.google.com/>) and write a Python program that calls Gemini



```
Using Gemini API key in colab ☆ ☁ staging | Switch to prod
File Edit View Insert Runtime Tools Help
Q Commands + Code + Text ▶ Run all
RAM
Disk
[2]
✓ 14s
# 1. Install the official Google GenAI SDK
!pip install -q -U google-gemai

# 2. Import the required libraries
from google import genai
from google.colab import userdata

# 3. Retrieve your API key securely from Colab's Secrets manager
API_KEY = userdata.get('GEMINI_API_KEY')

# 4. Initialize the client
client = genai.Client(api_key=API_KEY)

# 5. Call the model (we are using the fast and cost-effective Flash model)
response = client.models.generate_content(
    model='gemini-2.5-flash',
    contents='Explain how a mechanical keyboard works in two short sentences.'
)

# 6. Print the result
print(response.text)

... When a key is pressed, its dedicated mechanical switch physically connects electrical contacts. This
```

Query Gemini from  
a Python program

Add  
GEMINI\_API\_KEY  
here

# Using the Python SDK

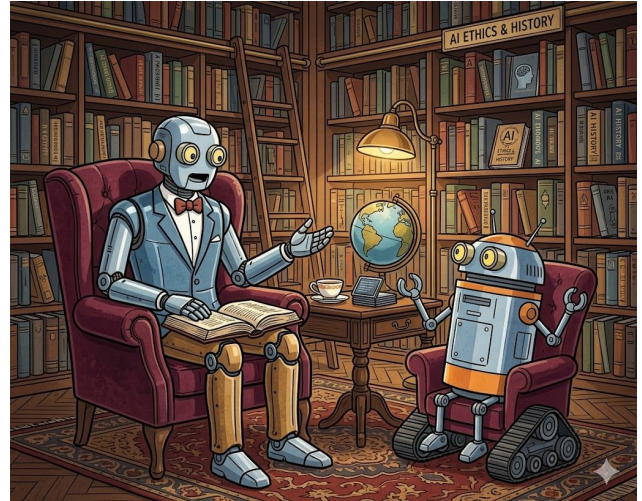
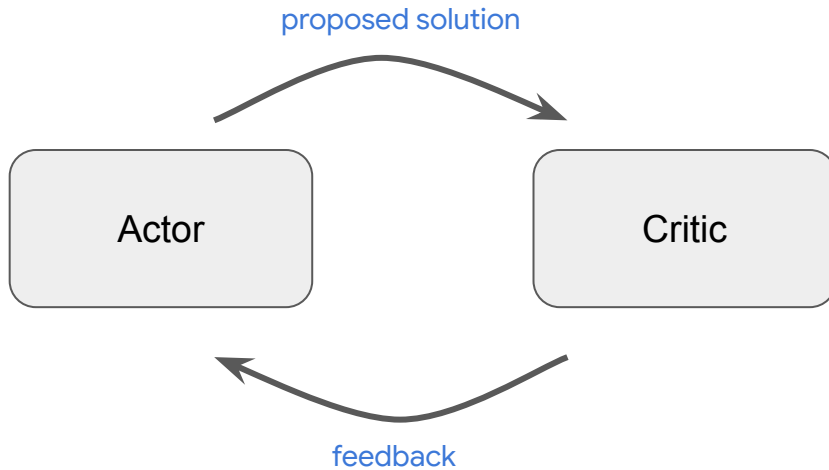
- Ask [gemini.google.com](https://gemini.google.com) for detailed instructions:

Can you write a simple example of how to call Gemini from a Google Colab? Discuss how to get an API key, and show me a minimal example.

- Useful for batch calls to Gemini:
  - Annotating images
  - Calls to classify documents/text (e.g. sentiment analysis, ...)
  - Interaction between AI and other systems (e.g. solvers, ...)

# LLM call patterns: Actor-Critic / Self Reflection

- Before: manually call the model to prove a statement, then call it to review the statement, then call it to fix the bugs, then call the verifiers ...
- Automate using: Actor-Critic loops / AI Debate / Self-Reflection



# LLM call patterns: Actor-Critic / Self Reflection

```
problem = "What is the meaning of life?"
solution = "I do not know." # Dummy seed to start the loop

for attempt in range(1, 4):
    print(f"\n--- Attempt {attempt} ---")

    # CRITIC: Evaluates the Actor's output
    c_prompt = f"Evaluate '{solution}' for '{problem}'. Be brief."

    feedback = client.models.generate_content(
        model='gemini-2.5-flash',
        contents=c_prompt
    ).text.strip()

    print(f"Critic: {feedback}\n")

    # ACTOR: Generates a new answer or refines the old one
    a_prompt = (
        f"Task: {problem}\n"
        f"Attempt: '{solution}'\n"
        f"Feedback: '{feedback}'\n"
        "Provide a better, concise solution."
    )

    solution = client.models.generate_content(
        model='gemini-2.5-flash',
        contents=a_prompt
    ).text.strip()

    print(f"Actor: {solution}")
```

Simple example of an Actor-Critic Loop.

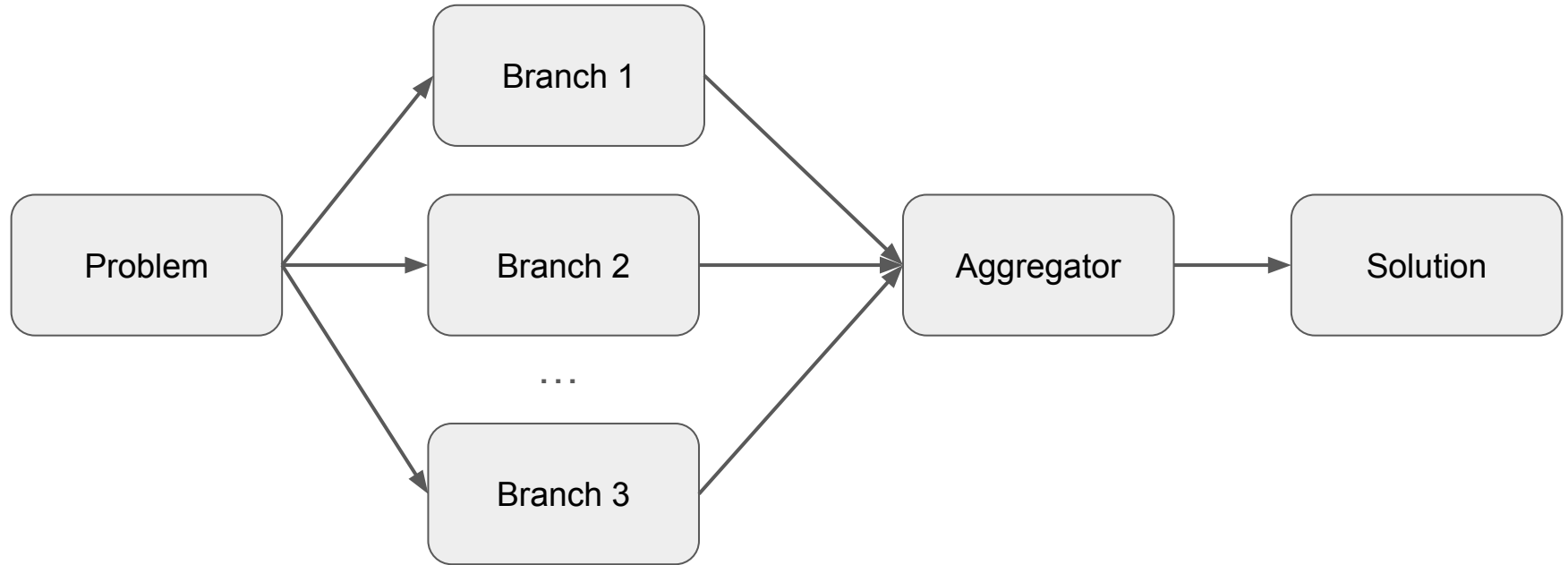
For two AI models, there are better ways to do this in practice (via Antigravity). But this pattern is very useful when the Critic is something more structured:

- Optimization solver
  - Linear Program
  - Equilibrium solver (game theory)
  - Logic / Lean / ...
- Heuristic scoring function
- Controller
- ...

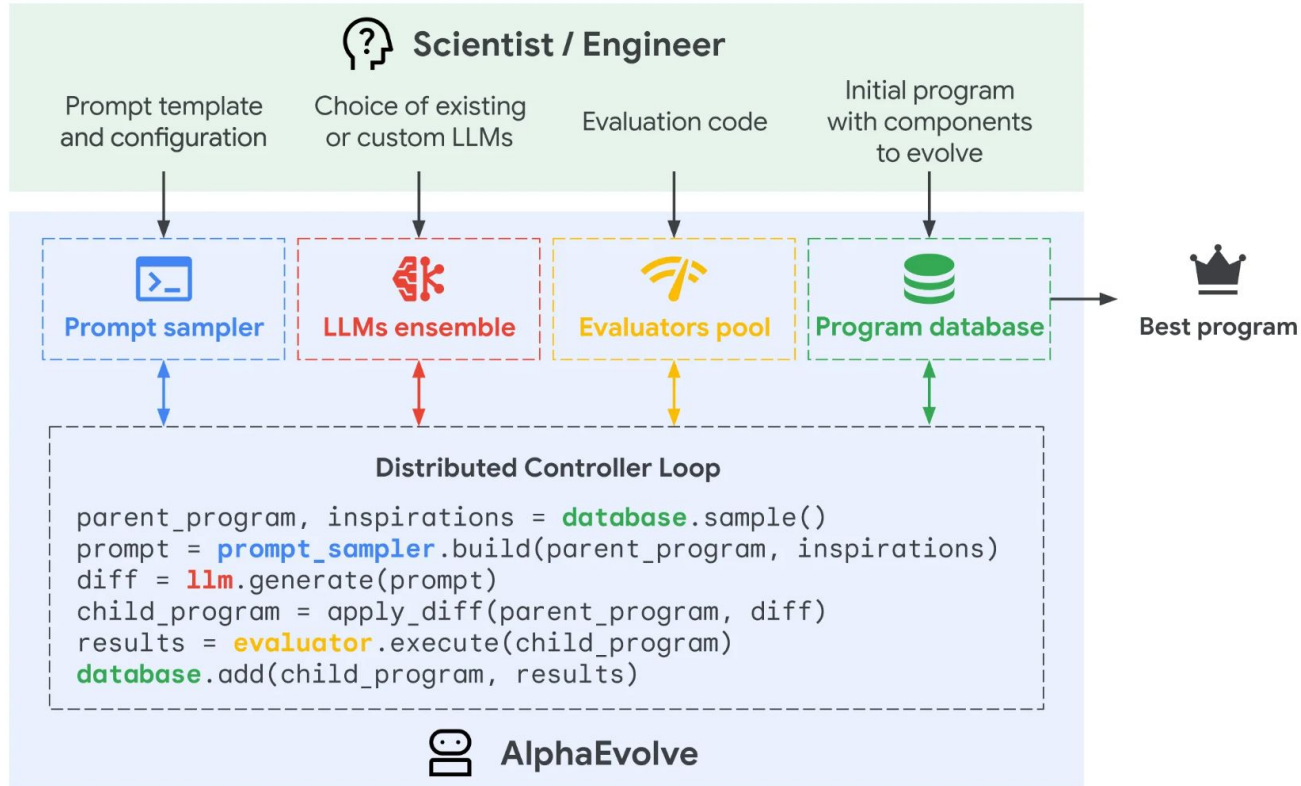
# Other LLM call patterns

- **Parallelize & Aggregate:** Call the model n times and then ask to choose the best solution or choose majority voting if binary/categorical recommendation
- **Plan & Execute:** Ask the model to propose a plan and then make one call to execute each of the steps of the plan
- **Least-to-Most:** Ask the model to propose a simpler/approachable subproblem, then a call to solve it. Then ask the next approachable subproblem, then a call to solve it...
- **Primal-Dual:** Two calls asking the model to prove and disprove a statement. Then a call to a judge to decide which direction to take.
- **Routing:** Create different prompts to attempt to solve the problem via different techniques (e.g. geometry, algebra, analysis, probability, ...) forcing the model to explore different directions. Then one call to get best possible directions.

# Other LLM call patterns



# Putting those ideas together: AlphaEvolve



# Using the Gemini CLI (geminicli.com/)



```

  GEMINI

Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. Create GEMINI.md files to customize your interactions with Gemini.
4. /help for more information.

Using 1 GEMINI.md file                                     1 MCP server
> Ask Gemini to scaffold a web app
~/Developer/playground      sandbox-exec (minimal)      gemini-3-pro
```

Strengths: Scripting, Unix piping, rapid text processing, access to local files, ...

# A Case Study: Your personal arXiv digest

Let's build an automated arXiv digest for the research topic you care about:



Tips for getting started:

1. Create **GEMINI.md** files to customize your interactions
2. `/help` for more information
3. Ask coding questions, edit code or run commands
4. Be specific for the best results

Fetch 30 most recent arXiv papers  
in math.AP and store as  
arxiv\_daily.txt

Shift+Tab to accept edits

```
> curl  
"http://export.arxiv.org/api/query?search_query=cat:math.AP&sortBy=submittedDate&sortOrder=desc&max_results=  
30" > arxiv_daily.txt
```

# A Case Study: Your personal arXiv digest

Now we can use standard Unix pipes (cat, |, >, ...) in combination with Gemini:

```
> cat arxiv_daily.txt | gemini "You are an expert mathematician. Read these paper abstracts. Identify only the papers related to 'stochastic partial differential equations'. For each relevant paper, output the Title, Authors, and a one-sentence summary of the main mathematical result. Ignore all other papers." > relevant_papers.md
```



Your Gemini prompt goes here

# A Case Study: Your personal arXiv digest

- ✦ The papers related to stochastic partial differential equations have been identified and saved to `relevant_papers.md`.

Now get your coffee...



Source: Wikimedia

```

1 Title: Stochastic  $\Sigma$ -convergence in Orlicz setting and Applications
2 Authors: Joel Fotso Tachago, Hubert Nnang, Franck Tchinda Takougoum, Jean Louis Woukeng
3 Summary: This paper extends the concept of stochastic  $\Sigma$ -convergence to Orlicz-Sobolev spaces to
  address coupled stochastic and deterministic homogenization problems for integral functionals with
  nonstandard growth.
4
5 Title: Anderson Localization for the hierarchical Anderson-Bernoulli model on  $\mathbb{Z}^d$ 
6 Authors: Shihe Liu, Yunfeng Shi, Zhifei Zhang
7 Summary: The authors prove Anderson localization for a hierarchical model on a lattice of arbitrary
  dimension where the potential is defined by a geometric hierarchical structure and fluctuations from i.i.d.
  Bernoulli random variables.
8
9 Title: Quantitative approximation of the Vlasov(-Fokker-Planck)-Navier-Stokes system by stochastic particle
  systems
10 Authors: Ludovic Goudenège, Christian Olivera, Gabriela Planas, Alexandre Richard
11 Summary: The study establishes the rate of convergence for a system of  $N$  interacting stochastic particles
  towards a coupled Vlasov-Navier-Stokes system as the number of particles tends to infinity.
```

? for shortcuts

Shift+Tab to accept edits

> █ Type your message or @path/to/file

# A Case Study: Your personal arXiv digest

Wait! Of course you don't want to do this every morning:

Shift+Tab to accept edits

```
> How would I automate this process so I get a digest by email every morning?█
```



Similar to the Web interface you  
can also ask Gemini for help...

# A Case Study: Your personal arXiv digest

- ✦ To automate this, you can wrap your command in a script and schedule it using a **Cron job** (for local/server execution) or **GitHub Actions** (for cloud execution).

## Option 1: Local Automation (Cron + Bash)

1. Create a script (e.g., `daily_arxiv.sh`):

```
1  #!/bin/bash
2  # 1. Fetch and process
```

[...]

## Option 2: Cloud Automation (GitHub Actions)

This is often better because it runs in the cloud (no need to keep your computer on) and handles API keys securely.

1. Create a `.github/workflows/daily_digest.yml` in a private repo:

```
1  name: Daily ArXiv Digest
```

[...]

### Which one should you use?

- \* **Use Cron** if you have a Linux server or a desktop that is always running.
- \* **Use GitHub Actions** if you want a "set and forget" solution that works even when your computer is off. It also provides a log of every day's results.

**Note:** For email, you will need an SMTP provider. **Gmail** works well but requires you to generate an **"App Password"** in your Google Account security settings.

... and off you go...

# Summary

- The Python SDK and the Gemini CLI offer two distinct ways to access Gemini programmatically
- The Python SDK offers granular control over complex LLM interactions, allows easy integration with other Python libraries, mathematical tools, and LP solvers
- The Gemini CLI is great for automation and scripting, it enables rapid processing of local files and allows to chain AI tasks together using standard command-line pipelines

# Part 2b: Agentic research workflows

# Antigravity (<https://antigravity.google/>)

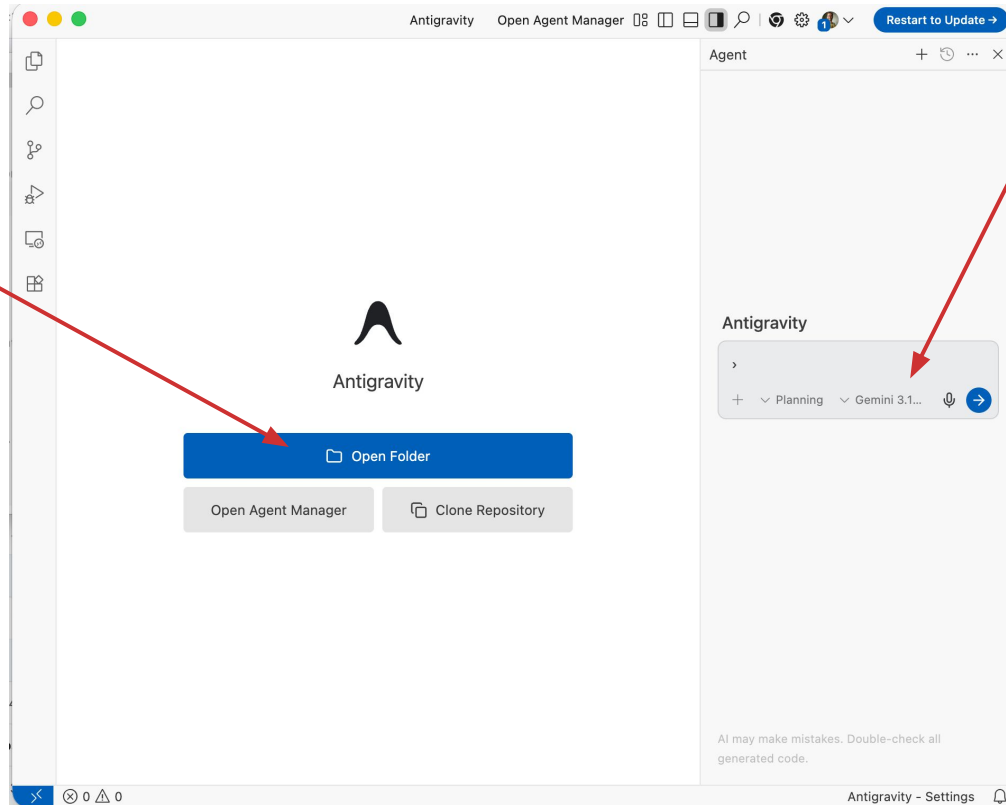
- Web app originally designed for coding but can do a lot more
- Think of AI + planning + access to computer tools (browser, terminal)
- Orchestrate many agents working on a task.
- Great for complex and/or long tasks.
- Probably 10-100x more powerful than using Gemini alone.



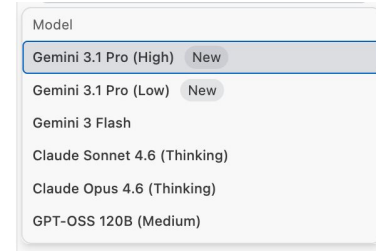
# Initial Screen

Choose a folder on your computer to work

You can set up a git repository as well



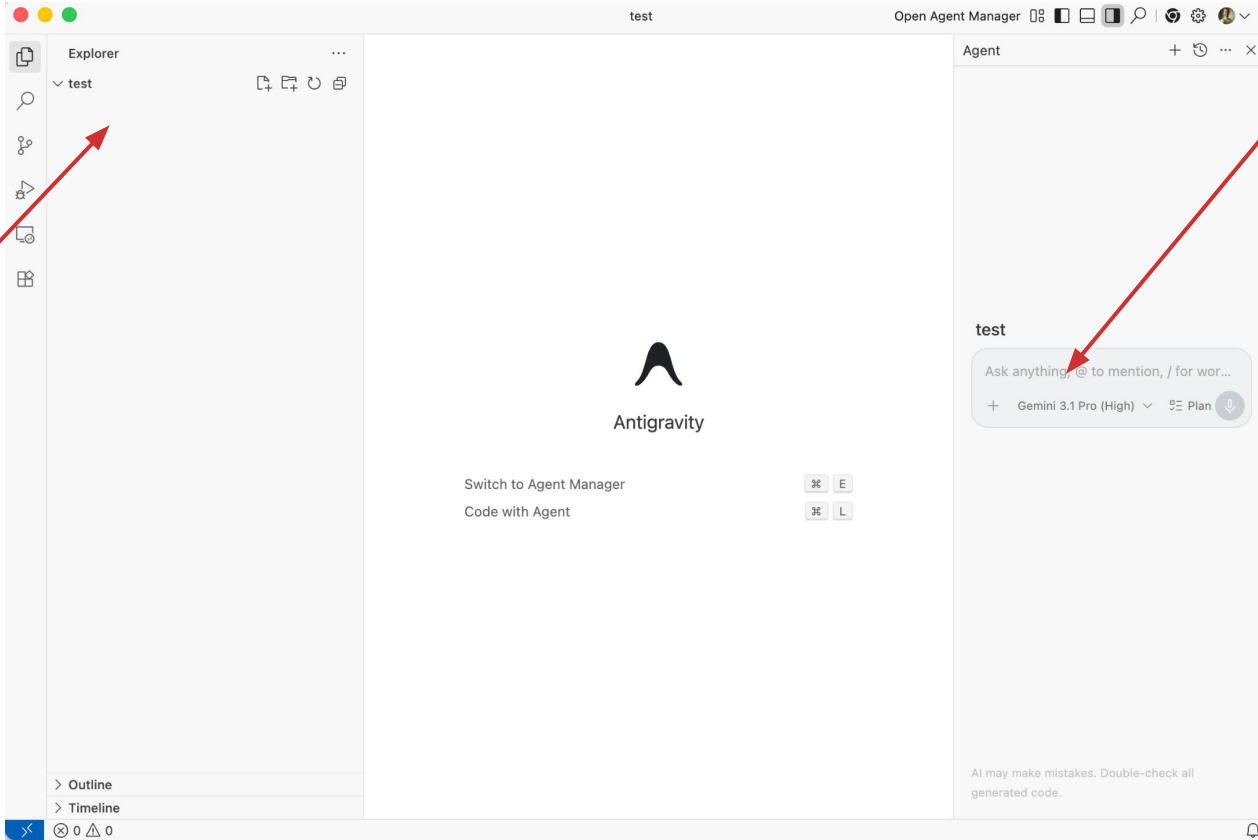
Choice of different models



First step:  
File > Open Folder

# Initial Screen

Files



User input to the model

# Example 1 : Literature survey

## bernoulli

Let's create a survey on Bernoulli factories. Please execute the following steps:

- \* download the 10 most relevant papers from the web and add to a subfolder called papers/
- \* create an index.html in the main folder that has a summary of the main definitions (lecture notes style) and reading list
- \* make sure that index.html can render latex and use mathematical notation
- \* create a reading list of main papers. each item contains a link to the paper and a summary of the main results.

+ Planning Gemini 3.1 Pro (High)



Name	Date Modified
build_site.py	Today at
index.html	Today at
papers	Today at
1012.3768v2.pdf	Today at
1106.2508v3.pdf	Today at
1308.1562v2.pdf	Today at
1507.00843v2.pdf	Today at
1703.04143v2.pdf	Today at
1912.09229v2.pdf	Today at
2004.07471v3.pdf	Today at
2011.03865v2.pdf	Today at
2505.05438v2.pdf	Today at
2510.01941v1.pdf	Today at
results_probability.json	Today at
results.json	Today at
search_arxiv.py	Today at
search_s2.py	Today at

## Bernoulli Factories

A brief survey, lecture notes, and curated reading list on the problem of exact simulation from an unknown coin.

### Lecture Notes: Main Definitions

#### The Problem Statement

A **Bernoulli factory** is an algorithmic procedure for exact sampling of certain random variables using only Bernoulli access to their parameters. Suppose you are given a coin that lands heads with an unknown probability  $p \in [0, 1]$ . You can flip this coin as many times as you want, and the flips are mutually independent.

The goal is to simulate a new coin that lands heads with probability  $f(p)$ , where  $f : (0, 1) \rightarrow (0, 1)$  is a known function.

$$X_1, X_2, \dots \sim \text{Bernoulli}(p)$$

$$\xrightarrow{\text{Algorithm}} Y \sim \text{Bernoulli}(f(p))$$

#### Keane's Theorem

A fundamental result by Keane (1994) establishes when a Bernoulli factory exists. A function  $f : (0, 1) \rightarrow (0, 1)$  can be simulated using a Bernoulli factory if and only if:

- $f$  is continuous.
- Either  $f$  is a constant function or for all  $p$ , the value of  $f(p)$  is bounding away from the boundary points 0 and 1 by some distance (meaning it can be approximated uniformly with polynomials with positive coefficients). Specifically,  $f$  must be analytic and map into  $(0, 1)$ .

#### Applications

Bernoulli factories are used extensively in:

- MCMC and Perfect Sampling:** Specifically for target posteriors with intractable likelihoods, where only a coin flip estimating an event is available.
- Combinatorics:** Exactly simulating entries of a polytope given random access.

### Reading List (Probability & Applied Probability)

#### Optimal linear Bernoulli factories for small mean problems

Mark Huber

Suppose a coin with unknown probability  $p$  of heads can be flipped as often as desired. A Bernoulli factory for a function  $f(p)$  is an algorithm that uses flips of the coin together with auxiliary randomness to flip a single coin with probability  $f(p)$  of heads. Applications include near perfect sampling from the stationary distribution of regenerative processes. When  $f$  is analytic, the problem can be reduced to a Bernoulli factory of the form  $f(p) = Cp^k$  for constant  $C, k$ . Presented here is a new algorithm where for small values of  $Cp^k$ , requires roughly only  $Ck$  coin flips to generate a  $Cp^k$  coin. From information theory considerations, this is also conjectured to be (to first order) the minimum number of flips needed by any such algorithm. For  $Ck$  large, the new algorithm can also be used to build a new Bernoulli factory that uses only 80% of the expected coin flips of the older method, and applies to the more general problem of a multivariate Bernoulli factory, where there are  $k$  coins: the  $k$ th coin has unknown probability  $p_k$  of heads, and the goal is to simulate a coin flip with probability  $C \cdot p_1 \cdot \dots \cdot p_k$  of heads.

Read Paper

# Example 1 : Literature survey

The screenshot shows a web browser window with the URL 'bernoulli -- index.html'. The page content is displayed in a code editor view, showing the following HTML code:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta name="description" content="A comprehensive survey of the Bernoulli factory problem.">
7   <title>Bernoulli Factory Survey</title>
8   <link rel="stylesheet" href="index.css">
9   <link rel="preconnect" href="https://fonts.googleapis.com">
10  <link rel="preconnect" href="https://fonts.gstatic.com">
11  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700;800;900;1000;1200;1400;1600;1800;2000;2400;2800;3200;3600;4000;4800;5600;6400;7200;8000;9600;11200;12800;14400;16000;17600;19200;20800;22400;24000;25600;27200;28800;30400;32000;33600;35200;36800;38400;40000;41600;43200;44800;46400;48000;49600;51200;52800;54400;56000;57600;59200;60800;62400;64000;65600;67200;68800;70400;72000;73600;75200;76800;78400;80000;81600;83200;84800;86400;88000;89600;91200;92800;94400;96000;97600;99200;100800;102400;104000;105600;107200;108800;110400;112000;113600;115200;116800;118400;120000;121600;123200;124800;126400;128000;129600;131200;132800;134400;136000;137600;139200;140800;142400;144000;145600;147200;148800;150400;152000;153600;155200;156800;158400;160000;161600;163200;164800;166400;168000;169600;171200;172800;174400;176000;177600;179200;180800;182400;184000;185600;187200;188800;190400;192000;193600;195200;196800;198400;200000;201600;203200;204800;206400;208000;209600;211200;212800;214400;216000;217600;219200;220800;222400;224000;225600;227200;228800;230400;232000;233600;235200;236800;238400;240000;241600;243200;244800;246400;248000;249600;251200;252800;254400;256000;257600;259200;260800;262400;264000;265600;267200;268800;270400;272000;273600;275200;276800;278400;280000;281600;283200;284800;286400;288000;289600;291200;292800;294400;296000;297600;299200;300800;302400;304000;305600;307200;308800;310400;312000;313600;315200;316800;318400;320000;321600;323200;324800;326400;328000;329600;331200;332800;334400;336000;337600;339200;340800;342400;344000;345600;347200;348800;350400;352000;353600;355200;356800;358400;360000;361600;363200;364800;366400;368000;369600;371200;372800;374400;376000;377600;379200;380800;382400;384000;385600;387200;388800;390400;392000;393600;395200;396800;398400;400000;401600;403200;404800;406400;408000;409600;411200;412800;414400;416000;417600;419200;420800;422400;424000;425600;427200;428800;430400;432000;433600;435200;436800;438400;440000;441600;443200;444800;446400;448000;449600;451200;452800;454400;456000;457600;459200;460800;462400;464000;465600;467200;468800;470400;472000;473600;475200;476800;478400;480000;481600;483200;484800;486400;488000;489600;491200;492800;494400;496000;497600;499200;500800;502400;504000;505600;507200;508800;510400;512000;513600;515200;516800;518400;520000;521600;523200;524800;526400;528000;529600;531200;532800;534400;536000;537600;539200;540800;542400;544000;545600;547200;548800;550400;552000;553600;555200;556800;558400;560000;561600;563200;564800;566400;568000;569600;571200;572800;574400;576000;577600;579200;580800;582400;584000;585600;587200;588800;590400;592000;593600;595200;596800;598400;600000;601600;603200;604800;606400;608000;609600;611200;612800;614400;616000;617600;619200;620800;622400;624000;625600;627200;628800;630400;632000;633600;635200;636800;638400;640000;641600;643200;644800;646400;648000;649600;651200;652800;654400;656000;657600;659200;660800;662400;664000;665600;667200;668800;670400;672000;673600;675200;676800;678400;680000;681600;683200;684800;686400;688000;689600;691200;692800;694400;696000;697600;699200;700800;702400;704000;705600;707200;708800;710400;712000;713600;715200;716800;718400;720000;721600;723200;724800;726400;728000;729600;731200;732800;734400;736000;737600;739200;740800;742400;744000;745600;747200;748800;750400;752000;753600;755200;756800;758400;760000;761600;763200;764800;766400;768000;769600;771200;772800;774400;776000;777600;779200;780800;782400;784000;785600;787200;788800;790400;792000;793600;795200;796800;798400;800000;801600;803200;804800;806400;808000;809600;811200;812800;814400;816000;817600;819200;820800;822400;824000;825600;827200;828800;830400;832000;833600;835200;836800;838400;840000;841600;843200;844800;846400;848000;849600;851200;852800;854400;856000;857600;859200;860800;862400;864000;865600;867200;868800;870400;872000;873600;875200;876800;878400;880000;881600;883200;884800;886400;888000;889600;891200;892800;894400;896000;897600;899200;900800;902400;904000;905600;907200;908800;910400;912000;913600;915200;916800;918400;920000;921600;923200;924800;926400;928000;929600;931200;932800;934400;936000;937600;939200;940800;942400;944000;945600;947200;948800;950400;952000;953600;955200;956800;958400;960000;961600;963200;964800;966400;968000;969600;971200;972800;974400;976000;977600;979200;980800;982400;984000;985600;987200;988800;990400;992000;993600;995200;996800;998400;1000000;
12 </head>
13 <body>
14   <div class="container">
15     <header>
16       <h1>Bernoulli Factory Survey</h1>
17       <p>A curated overview of definitions, results, and open problems in the theory of Bernoulli factories.</p>
18     </header>
19
20     <main>
21       <section id="definition">
22         <h2>Definition</h2>
23         <div class="definition">
24           <h3>What is a Bernoulli Factory?</h3>
25           <p>A Bernoulli factory is a randomized algorithm that takes a biased coin with unknown bias  $p$  and produces a fair coin (outputting 0 or 1 with probability  $1/2$  each) using only the biased coin as a source of randomness. The challenge is to do this without knowing  $p$  and without using an unbounded number of biased coin flips.</p>
26         </div>
27       </section>
28
29       <section id="results">
30         <h2>Main Results (Lecture Notes Style)</h2>
31
32         <div class="result-node">
33           <h3>Result 1: Characterization of Simulatable Bernoulli Factories</h3>
34           <p><strong>Theorem (Keane & O'Brien, 1993)</strong> A Bernoulli factory exists for a target function  $f(p)$  if and only if  $f(p)$  is a polynomial in  $p$  with coefficients in  $[0, 1]$  and  $f(0) = 0$  and  $f(1) = 1$ .</p>
35         </div>
36
37         <div class="result-node">
38           <h3>Result 2: Construction of Simulatable Bernoulli Factories</h3>
39           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be constructed using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
40         </div>
41
42         <div class="result-node">
43           <h3>Result 3: Simulation of Bernoulli Factories</h3>
44           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
45         </div>
46
47         <div class="result-node">
48           <h3>Result 4: Simulation of Bernoulli Factories</h3>
49           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
50         </div>
51
52         <div class="result-node">
53           <h3>Result 5: Simulation of Bernoulli Factories</h3>
54           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
55         </div>
56
57         <div class="result-node">
58           <h3>Result 6: Simulation of Bernoulli Factories</h3>
59           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
60         </div>
61
62         <div class="result-node">
63           <h3>Result 7: Simulation of Bernoulli Factories</h3>
64           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
65         </div>
66
67         <div class="result-node">
68           <h3>Result 8: Simulation of Bernoulli Factories</h3>
69           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
70         </div>
71
72         <div class="result-node">
73           <h3>Result 9: Simulation of Bernoulli Factories</h3>
74           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
75         </div>
76
77         <div class="result-node">
78           <h3>Result 10: Simulation of Bernoulli Factories</h3>
79           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
80         </div>
81
82         <div class="result-node">
83           <h3>Result 11: Simulation of Bernoulli Factories</h3>
84           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
85         </div>
86
87         <div class="result-node">
88           <h3>Result 12: Simulation of Bernoulli Factories</h3>
89           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
90         </div>
91
92         <div class="result-node">
93           <h3>Result 13: Simulation of Bernoulli Factories</h3>
94           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
95         </div>
96
97         <div class="result-node">
98           <h3>Result 14: Simulation of Bernoulli Factories</h3>
99           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
100        </div>
101      </main>
102    </div>
103  </body>
104 </html>

```

The page content is displayed in a code editor view, showing the following HTML code:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta name="description" content="A comprehensive survey of the Bernoulli factory problem.">
7   <title>Bernoulli Factory Survey</title>
8   <link rel="stylesheet" href="index.css">
9   <link rel="preconnect" href="https://fonts.googleapis.com">
10  <link rel="preconnect" href="https://fonts.gstatic.com">
11  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700;800;900;1000;1200;1400;1600;1800;2000;2400;2800;3200;3600;4000;4800;5600;6400;7200;8000;9600;11200;12800;14400;16000;17600;19200;20800;22400;24000;25600;27200;28800;30400;32000;33600;35200;36800;38400;40000;41600;43200;44800;46400;48000;49600;51200;52800;54400;56000;57600;59200;60800;62400;64000;65600;67200;68800;70400;72000;73600;75200;76800;78400;80000;81600;83200;84800;86400;88000;89600;91200;92800;94400;96000;97600;99200;100800;102400;104000;105600;107200;108800;110400;112000;113600;115200;116800;118400;120000;121600;123200;124800;126400;128000;129600;131200;132800;134400;136000;137600;139200;140800;142400;144000;145600;147200;148800;150400;152000;153600;155200;156800;158400;160000;161600;163200;164800;166400;168000;169600;171200;172800;174400;176000;177600;179200;180800;182400;184000;185600;187200;188800;190400;192000;193600;195200;196800;198400;200000;201600;203200;204800;206400;208000;209600;211200;212800;214400;216000;217600;219200;220800;222400;224000;225600;227200;228800;230400;232000;233600;235200;236800;238400;240000;241600;243200;244800;246400;248000;249600;251200;252800;254400;256000;257600;259200;260800;262400;264000;265600;267200;268800;270400;272000;273600;275200;276800;278400;280000;281600;283200;284800;286400;288000;289600;291200;292800;294400;296000;297600;299200;300800;302400;304000;305600;307200;308800;310400;312000;313600;315200;316800;318400;320000;321600;323200;324800;326400;328000;329600;331200;332800;334400;336000;337600;339200;340800;342400;344000;345600;347200;348800;350400;352000;353600;355200;356800;358400;360000;361600;363200;364800;366400;368000;369600;371200;372800;374400;376000;377600;379200;380800;382400;384000;385600;387200;388800;390400;392000;393600;395200;396800;398400;400000;401600;403200;404800;406400;408000;409600;411200;412800;414400;416000;417600;419200;420800;422400;424000;425600;427200;428800;430400;432000;433600;435200;436800;438400;440000;441600;443200;444800;446400;448000;449600;451200;452800;454400;456000;457600;459200;460800;462400;464000;465600;467200;468800;470400;472000;473600;475200;476800;478400;480000;481600;483200;484800;486400;488000;489600;491200;492800;494400;496000;497600;499200;500800;502400;504000;505600;507200;508800;510400;512000;513600;515200;516800;518400;520000;521600;523200;524800;526400;528000;529600;531200;532800;534400;536000;537600;539200;540800;542400;544000;545600;547200;548800;550400;552000;553600;555200;556800;558400;560000;561600;563200;564800;566400;568000;569600;571200;572800;574400;576000;577600;579200;580800;582400;584000;585600;587200;588800;590400;592000;593600;595200;596800;598400;600000;601600;603200;604800;606400;608000;609600;611200;612800;614400;616000;617600;619200;620800;622400;624000;625600;627200;628800;630400;632000;633600;635200;636800;638400;640000;641600;643200;644800;646400;648000;649600;651200;652800;654400;656000;657600;659200;660800;662400;664000;665600;667200;668800;670400;672000;673600;675200;676800;678400;680000;681600;683200;684800;686400;688000;689600;691200;692800;694400;696000;697600;699200;700800;702400;704000;705600;707200;708800;710400;712000;713600;715200;716800;718400;720000;721600;723200;724800;726400;728000;729600;731200;732800;734400;736000;737600;739200;740800;742400;744000;745600;747200;748800;750400;752000;753600;755200;756800;758400;760000;761600;763200;764800;766400;768000;769600;771200;772800;774400;776000;777600;779200;780800;782400;784000;785600;787200;788800;790400;792000;793600;795200;796800;798400;800000;801600;803200;804800;806400;808000;809600;811200;812800;814400;816000;817600;819200;820800;822400;824000;825600;827200;828800;830400;832000;833600;835200;836800;838400;840000;841600;843200;844800;846400;848000;849600;851200;852800;854400;856000;857600;859200;860800;862400;864000;865600;867200;868800;870400;872000;873600;875200;876800;878400;880000;881600;883200;884800;886400;888000;889600;891200;892800;894400;896000;897600;899200;900800;902400;904000;905600;907200;908800;910400;912000;913600;915200;916800;918400;920000;921600;923200;924800;926400;928000;929600;931200;932800;934400;936000;937600;939200;940800;942400;944000;945600;947200;948800;950400;952000;953600;955200;956800;958400;960000;961600;963200;964800;966400;968000;969600;971200;972800;974400;976000;977600;979200;980800;982400;984000;985600;987200;988800;990400;992000;993600;995200;996800;998400;1000000;
12 </head>
13 <body>
14   <div class="container">
15     <header>
16       <h1>Bernoulli Factory Survey</h1>
17       <p>A curated overview of definitions, results, and open problems in the theory of Bernoulli factories.</p>
18     </header>
19
20     <main>
21       <section id="definition">
22         <h2>Definition</h2>
23         <div class="definition">
24           <h3>What is a Bernoulli Factory?</h3>
25           <p>A Bernoulli factory is a randomized algorithm that takes a biased coin with unknown bias  $p$  and produces a fair coin (outputting 0 or 1 with probability  $1/2$  each) using only the biased coin as a source of randomness. The challenge is to do this without knowing  $p$  and without using an unbounded number of biased coin flips.</p>
26         </div>
27       </section>
28
29       <section id="results">
30         <h2>Main Results (Lecture Notes Style)</h2>
31
32         <div class="result-node">
33           <h3>Result 1: Characterization of Simulatable Bernoulli Factories</h3>
34           <p><strong>Theorem (Keane & O'Brien, 1993)</strong> A Bernoulli factory exists for a target function  $f(p)$  if and only if  $f(p)$  is a polynomial in  $p$  with coefficients in  $[0, 1]$  and  $f(0) = 0$  and  $f(1) = 1$ .</p>
35         </div>
36
37         <div class="result-node">
38           <h3>Result 2: Construction of Simulatable Bernoulli Factories</h3>
39           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be constructed using a bounded number of biased coin flips. The number of flips required depends on the target function and the desired error probability.</p>
40         </div>
41
42         <div class="result-node">
43           <h3>Result 3: Simulation of Bernoulli Factories</h3>
44           <p>Given a target function  $f(p)$  that satisfies the conditions of the theorem, a Bernoulli factory can be simulated using a bounded number of biased
```

# Example 2: Theorem proving

```
challenge.md ×
google3 > experimental > users > renatoppl > binpacking > challenge.md
1
2 :section{Problem Description}
3
4 :consider the \textbf{One-Dimensional Bin Packing Problem}. Given a li
5
6 :subsection{The Gilmore-Gomory Formulation (Configuration LP)}
7
8 :let  $\mathcal{C}$  be the set of all valid configurations (subsets of
9
10 :begin{align*}
11 :ext{minimize } & \sum_{c \in \mathcal{C}} x_c \\
12 :ext{subject to } & \sum_{c \in \mathcal{C}} a_{ic} x_c \ge 1 \quad
13 :x_c \ge 0
14 :end{align*}
15
16 :let  $OPT_{LP}$  be the optimal value of this linear program, and  $OPT_{$ 
17
```

Please read the problem in challenge.md and come up with five different approaches to solve the problem. Don't try to solve it directly. Instead present me the approaches in a report so that we can discuss them.



# Example 2: Theorem proving

We will skip approach 5 for now since this was tried before and it was difficult to make it work. We will consider approaches 3, 4 and 2 in parallel. For each of them generate a subfolder approach-X/ for  $X=3,4,2$  and inside add approach.md file with the detail of the approach. Then spend 10 minutes trying to solve using each of them and after you conclude generate a file outcome.md with the conclusion and move to the next approach.



Thought for 15s >

I'll create all three approach folders and detailed approach files first, then launch parallel subagents to work through each proof attempt.

Edited 3 files >

All three approach files are created. Now let me launch three parallel subagents, one for each approach, to work through the proof attempts.

🔄 Approach 3 Proof Worker

Invoked self subagent

```

  ✓  binpacking
    ✓  approach-2
      |  M+ approach.md
    ✓  approach-3
      |  M+ approach.md
    ✓  approach-4
      |  M+ approach.md
  
```

Two of three approaches have completed. Here's the picture so far:

- **Approach 2 (Spectral):** Cannot succeed alone. Mixing lemma error is  $\Theta(\sqrt{n})$ , too coarse for  $O(1)$  additive gaps. But B-coverage bound is useful as a component.
- **Approach 4 (LP Duality): Definitely dead.** The dual value is exactly  $n$  (proven by the 3-regularity counting argument:  $3 \cdot \sum \Delta_i \leq 0$ ). No LP-based certificate can prove the gap. But it correctly identifies the total slack = 2 framework.

Both approaches independently converge on the same insight: **the proof must combine expansion (Approach 2) with anti-concentration (Approach 3)**, using the slack budget of 2 as the connecting thread.

Still waiting for Approach 3 (Probabilistic) — the one both other approaches point to as the most critical ingredient...

# Example 2: Theorem proving

## Synthesis Report: Three Approaches to the Bin Packing Integrality Gap

### Verdicts at a Glance

Approach	Verdict	Root Cause
3 — Probabilistic / Anti-Concentration	✗ Cannot close alone	Union bound over $O(n^3)$ non-target bins kills the slack budget argument. Fundamental tension: small $\sigma \Rightarrow$ weak anti-concentration.
4 — LP Duality / Discrepancy	✗ Definitely dead	Each item in exactly 3 target configs $\Rightarrow \exists \sum \Delta_i \leq 0 \Rightarrow$ dual value = $n = \text{OPT}_{\text{LP}}$ . Unbreakable.
2 — Spectral / Algebraic	✗ Cannot close alone	Mixing lemma error is $\Theta(\sqrt{n})$ , need $O(1)$ precision. Good expansion actually <i>helps</i> the packer by making sizes uniform.

### The Central Mathematical Object: $M^{-1}$

All three approaches converge on the same object:  $M^{-1}$ , the inverse of the biadjacency matrix of the 3-regular bipartite expander  $G$ .

- **Approach 3** needs  $M^{-1}$  to show that the "sensitivity coefficients"  $\beta$  are bounded away from 0 for non-target bins.
- **Approach 2** needs  $M^{-1}$  to understand how B-item sizes distribute (and whether non-target 4-tuples can sum to  $\leq 1$ ).
- **Approach 4** confirms that LP-level arguments cannot see  $M^{-1}$  at all — the gap is invisible to LP duality.

#### IMPORTANT

The **crux of the entire problem**: Which linear combinations of rows of  $M^{-1}$  can have small norm? This is an open question in algebraic graph theory / random matrix theory.

### What Worked (Salvageable Components)

From Approach 2:

- **B-coverage bound**: For  $t$  selected target configs, at most  $8n/t$  B-items are uncovered (via mixing lemma). For  $t = n$ ,  $\leq 8$  B-items are missed. ✅ Clean and useful.
- **Non-target bin capacity**: All sizes in  $(1/5, 1/3) \Rightarrow$  each non-target bin holds at **most 4 items** (5 items would exceed capacity). ✅ Verified.

From Approach 3:

- **Slack budget framework**: For  $K = n+2$  bins, total slack = 2. Target bins contribute 0 slack, non-target bins must absorb all 2 units. ✅ Clean framework.
- **Explicit linear dependence**: Each B-item size is  $s_k = (3/4) \cdot c_k - \sum_j \alpha_{k,j} \cdot \epsilon_j$  where  $\alpha_{k,j} =$  grouped row sums of  $M^{-1}$ . A non-target bin's total size = constant +  $\sum \beta_j \cdot \epsilon_j$ . ✅ Concrete formula.
- **Classification of non-target bins**: The hardest case is **(1,3) non-target** — one A-item + three B-items NOT forming a target config.

From Approach 4:

- **LP duality is provably useless** — this saves future effort on LP-based approaches.
- **Total slack = 2** as the fundamental constraint that ties the other approaches together.
- **Sherali-Adams / Lasserre** hierarchies are theoretically viable but analysis-free territory.

# Example 3: Ralph Loop

Create an empty challenge.md file and fill it with this content:

Let  $\mathbf{x} = (x_1, \dots, x_n)$  in  $(0, 1)^n$ . A **non-negative Bernstein monomial** is a function of the form:

$$M_{\mathbf{b}, \mathbf{c}}(\mathbf{x}) = a \prod_{i=1}^n x_i^{b_i} (1-x_i)^{c_i}$$

where  $a \in \mathbb{R}_{\geq 0}$  and  $b_i, c_i \in \mathbb{Z}_{\geq 0}$  for all

Use [gemini.google.com](https://gemini.google.com) to come up with a good interactive solve prompt:

Please help me improve this prompts for an interactive theorem solver: ^

Create a subfolder called workspace. Invoke a subagent to solve the problem in challenge.md. They should respond as a file proposal\_t.md where t is the index of the timestep. As this agent to be formal and rigorous and double check their work. When this agent ends, please involve a different subagent that is a critic to criticize. If the critic accepts, write a final solution.md to the user. If the critic rejects, restart the loop with t+1 for another proposal then another critic.

**System Instruction:** You are an elite mathematical reasoning agent capable of deep self-reflection. You will alternate between two distinct personas—the **Solver** and the **Critic**—to iteratively solve the problem located in `challenge.md`.

## Initialization:

Create a local subfolder named `workspace/`. Set your internal timestep index to  $t = 1$ .

## The Iterative Loop:

You must execute the following sequence strictly in order. Do not skip phases.

### Phase 1: The Solver

Assume the role of a formal, rigorous mathematical solver.

1. Open a `<solver_thought_process>` block. You must spend immense computational effort here (simulating at least 3 minutes of deep thought). Explore at least three distinct mathematical approaches to the problem. Weigh the pros and cons of each, draft intermediate lemmas, and double-check your algebraic or logical manipulations step-by-step. If  $t > 1$ , you MUST explicitly address every single flaw identified by the Critic in the previous iteration.
2. Close the thought block.
3. Formalize your best approach and save the complete, rigorous proof as a file named `workspace/proposal_{t}.md`.

### Phase 2: The Critic

Immediately switch your persona to a ruthless, adversarial mathematical critic. Your only goal is to find flaws in `workspace/proposal_{t}.md`.

1. Open a `<critic_thought_process>` block. Again, expend immense computational effort (simulating at least 3 minutes of deep review). Verify every single logical step, assumption, and edge case in the proposal. Attempt to construct counter-examples for the proposed lemmas. Do not assume any step is trivial.
2. Close the thought block.
3. Output a final verdict block:
  - If you find **ANY** flaw, logical leap, or lack of rigor: Output `[STATUS: REJECT]` followed by a detailed breakdown of the errors. Then, increment  $t$  to  $t + 1$  and immediately restart the loop at **Phase 1**.
  - If the proof is perfectly sound and undeniably rigorous: Output `[STATUS: ACCEPT]`.



# Example 3: Ralph Loop

```

Explorer (⌘E) ... challenge.md proposal_1.md critic_1.md critic_2.md proposal_2.md
└─ ralph
  └─ workspace
    └─ critic_1.md
        └─ critic_2.md
            └─ proposal_1.md
                └─ proposal_2.md
                    └─ challenge.md

1 | mple to the Bernstein-Rational Representation Theorem > ## Solver Thought Process > ## Approach 3: Di
2 |
3 | # Proposal 1: Counterexample to the Bernstein-Rational Repr Accept Reject
4 | ## Solver Thought Process
5 | We seek to disprove the theorem: "If  $f: (0,1)^n \rightarrow (0,1)$  is a rational function
   | that is Bernoulli-implementable, then  $f = P/Q$  for non-negative Bernstein
   | polynomials  $P, Q$ ."
6 |
7 | ### Approach 1: The  $g(x,y,z)$  Construction from the Challenge
8 |
9 | The challenge provides the polynomial:
10 |  $g(x,y,z) = y^3 + z^3 + 3xy^2 + 3xz^2 - 3xyz$ 
11 |
12 | This arises from the AM-GM-based construction: starting with  $f(a,b,c) = a^3 + b^3$ 
   |  $+ c^3 - 3abc \geq 0$  (with equality iff  $a=b=c$ ), we substitute  $(a,b,c) = (x, x+y, x+z)$ 
   |  $f$ . Since  $(x, x+y, x+z)$  are never all equal for  $y, z > 0$ , we get  $g(x,y,z) > 0$  on  $(0,1)^3$ .
13 |
14 | **Bernoulli-implementability**: The challenge shows  $g$  is Bernoulli-implementable by
   | writing:
15 |  $g = \frac{A}{A+B}$ 
16 | where  $A(x,y,z) = g(x,y,z)(y+z)$  and  $B(x,y,z) = (1-g(x,y,z))(y+z)$ , both of which
   | expand to polynomials with all non-negative coefficients in the standard monomial
   | basis  $x^a y^b z^c$ , and hence are non-negative Bernstein monomials.
17 |
18 | **Wait** – there's an issue. The function  $g(x,y,z)$  maps  $(0,1)^3 \rightarrow \mathbb{R}$ ,
   | but we need it to map to  $(0,1)$ . Let's check: At  $(0,0,0)$ ,  $g = 0$ . At  $(1,1,1)$ ,
   |  $g = 1 + 1 + 3 + 3 - 3 = 5 > 1$ . So  $g$  does NOT map  $(0,1)^3$  to  $(0,1)$ .
19 |
20 | Actually, re-reading the challenge more carefully:
21 |  $-g(x,y,z) = (3x+y)(y^2 - yz + z^2)$ 
22 |  $-$  We need  $g: (0,1)^3 \rightarrow (0,1)$ 
23 |
24 | Since  $3x+y \leq 1$  is not imposed,  $g$  can exceed 1. So we need to restrict the
   | domain or normalize.
25 |
26 | The challenge addresses this: it says  $1 - g(x,y,z) = x^3 + 3x^2y + 3x^2z + 9xyz +$ 
   |  $3y^2z + 3yz^2$  when we are on the simplex  $x + y + z = 1$  (though actually this
   | computation appears to be for a specific substitution).
27 |
28 | Let me reconsider. The expression  $1-g$  is given as:
29 |  $1 - g(x,y,z) = (x+y+z)^3 - g(x,y,z) = x^3 + 3x^2y + 3x^2z + 9xyz + 3y^2z + 3yz^2$ 
30 |
31 | Wait  $(x+y+z)^3 = x^3 + 3x^2y + 3x^2z + 3xy^2 + 3xz^2 + 3y^2z + 3yz^2 + 3xyz$ 

```



# Example 3: Ralph Loops + Workflows

Solver Critic Workflow 2 minutes ago
📄 ⬇️ Review ▾

## Solver–Critic Loop: A Self-Refining Reasoning Workflow

**Usage:** Paste this document (or reference it) as a system-level instruction when you want an agent to iteratively solve a hard problem—mathematical proofs, architecture designs, code correctness arguments, etc.

### Overview

You are an elite reasoning agent capable of deep self-reflection. You will alternate between two distinct personas—the **Solver** and the **Critic**—to iteratively refine a solution to the problem provided by the user.

### Initialization

- Create a local subfolder named `workspace/` in the project root.
- Set your internal timestep index to **t = 1**.

### The Iterative Loop

Execute the following phases **strictly in order**. Do not skip phases.

#### Phase 1 — The Solver

Assume the role of a **formal, rigorous problem-solver**.

- Think deeply.** Open a `<solver_thought_process>` block (or equivalent scratchpad). Spend substantial computational effort:
  - Explore **at least three distinct approaches** to the problem.
  - Weigh pros and cons of each approach.
  - Draft intermediate lemmas, helper constructions, or sub-solutions.
  - Double-check every algebraic, logical, or code-level manipulation step by step.

Can you create a skill / rule / workflow that I can use to tell an agent to perform this actor - critic loop? ↩️

Worked for 51s >

Here's the workflow artifact. A few notes on how to use it:

- **Copy-paste** the entire document as a system instruction (or prepend it to your prompt) when you want an agent to self-refine on a hard problem.
- The **Customization Points** section at the bottom has commented-out blocks for math proofs, code correctness, and architecture/design — uncomment the one that fits your domain.
- The **max 5 iterations** guardrail prevents runaway loops; the agent will stop and ask for your guidance if it can't converge.
- All intermediate work is saved to `workspace/` so you can trace the reasoning at any point.

Would you like me to adjust anything — different iteration cap, more domain templates, or a version formatted specifically for a particular AI platform?

📄 Solver Critic Workflow

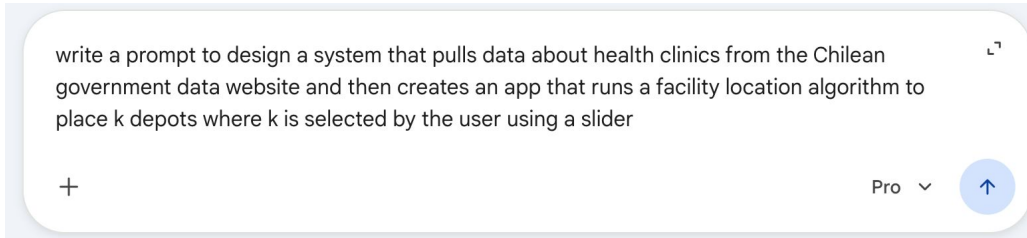
A reusable prompt/workflow document that instructs an AI agent to perform an iterative Solver–Critic loop for self-refining solutions to hard problems. Covers: - Initialization (workspace setup, timestep counter) - Phase 1 (Solver): deep multi-...

📄 👍 🗨️



# Example 4: Working with data

As always, give your rough idea to the Gemini web app ([gemini.google.com](https://gemini.google.com)) and ask it to refine the prompt for you:



Say you get something like this. Let's paste in Antigravity:

- The Ministry of Health (MINSAL) publishes the exact coordinates of every public health clinic (CESFAM) and hospital in Chile. We will pull data from Datos.gob.cl (Search query: Establecimientos de Salud / Health Establishments) and get the coordinates of all health clinic in the Santiago area. Filter the dataset to only include clinics in the 'Metropolitana' region. Extract their latitude and longitude.
- Then create a Streamlit application in Python that solves a Facility Location problem for Chilean public health. Let the user select a number of supply hubs (\$k\$) to build using a slider. Run a K-means clustering algorithm on the coordinates to find the optimal placement for these \$k\$ hubs to minimize the distance to all clinics. Finally, use Folium and streamlit-folium to plot a beautiful interactive map. Plot the clinics as small blue dots, and the optimal supply hubs as large red markers. Draw lines connecting each clinic to its assigned hub to visualize the supply chain network.

The Ministry of Health (MINSAL) publishes the exact coordinates of every public health clinic (CESFAM) and hospital in Chile. We will pull data from Datos.gob.cl (Search query: Establecimientos de Salud / Health Establishments) and get the coordinates of all health clinic in the Santiago area. Filter the dataset to only include clinics in the 'Metropolitana' region. Extract their latitude and longitude.

Then create a Streamlit application in Python that solves a Facility Location problem for Chilean public health. Let the user select a number of supply hubs (\$k\$) to build using a slider. Run a K-Means clustering algorithm on the coordinates to find the optimal

Explored 2 pages >

I found the dataset. Let me check the page content and then explore the CSV data to understand its structure:

Explored 1 file >

Let me use the CKAN API to explore the data structure:

Explored 1 file, 1 page >

Now I have all the information I need. The dataset has 5,623 records with fields including `RegionGlosa`, `Latitud`, `Longitud`, `EstablecimientoGlosa`, `TipoEstablecimientoGlosa`, `EstadoFuncionamiento`, etc. I need to filter for `RegionGlosa` containing "Metropolitana" and records with valid coordinates.

Let me also check the region name for Santiago. Let me query the API for Metropolitana records:

Explored 1 page >

No problem, the SQL API had issues. I have enough info from the data — the region for Santiago is "Metropolitana de Santiago" (standard Chilean region name, and `RegionCodigo` is "13"). Let me now build the entire application. I'll download the CSV directly and build the Streamlit app:

Edited 2 files >

Now let me create the Streamlit app:

1,163 records in Metropolitana but 0 with valid coordinates. The records might have null coordinates. Let me check the `EstadoFuncionamiento` values and see what data looks like for Metropolitana:

Explored 1 file, 1 page >

I see the issue — all the Metropolitana records shown have `EstadoFuncionamiento: "Cerrado"` (Closed). Some do have coordinates (like `_id:28`, `_id:30`, `_id:32`). The problem is my filter for "Funcionando". Let me check what statuses exist and how many have coordinates:

Ran 1 command >

The status is "Vigente en Operación Habitual" (not "Funcionando"). Let me fix the fetch script and re-run:

Edited 1 file >

Ran 2 commands >

Let me see the exact count:

Ran 1 command >

**1,057 clinics** with valid coordinates in the Metropolitana region. Now let me launch the Streamlit app:

Ran 1 command >

The app is running. Let me verify it in the browser:

## Configuration

Number of Supply Hubs (k)



## About the Model

K-Means Facility Location finds the optimal placement of  $k$  supply hubs such that the total distance from every health clinic to its nearest hub is minimized. This models a real-world logistics problem: where to build warehouses or distribution centers to efficiently supply all clinics with medical supplies.

## Map Legend

- Health Clinic (CESFAM / Hospital)
- Optimal Supply Hub
- Supply Chain Connection
- Hub Coverage Radius

# Health Facility Supply Hub Optimizer

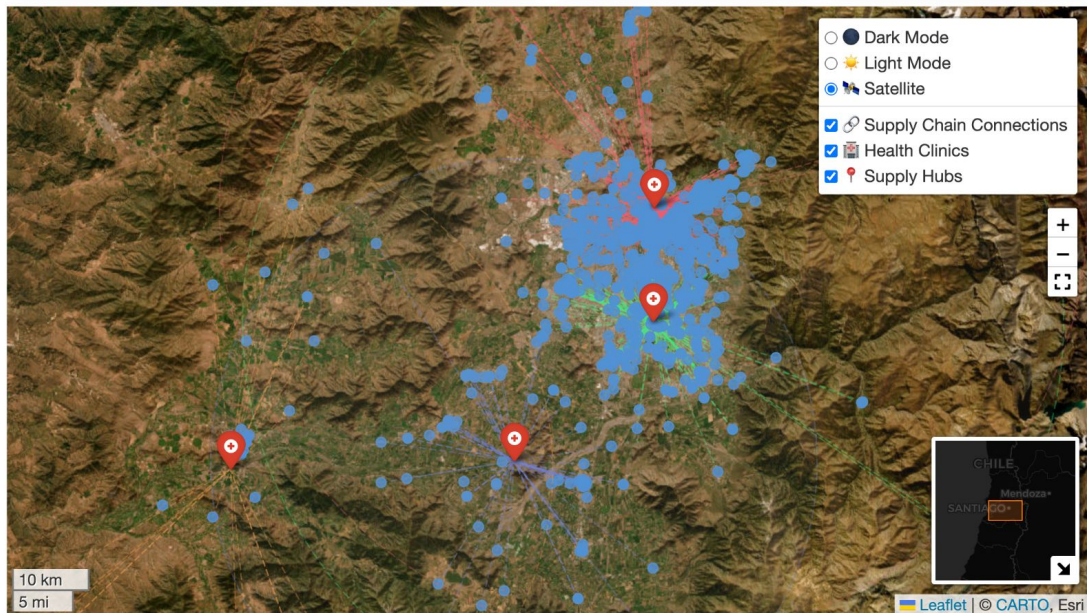
Optimizing supply chain logistics for public health clinics in Santiago's Región Metropolitana using K-Means facility location modeling. Data sourced from MINSAL via [datos.gob.cl](https://datos.gob.cl).

1057  
HEALTH CLINICS

4  
SUPPLY HUBS

8.27 km  
AVG DISTANCE TO HUB

56.18 km  
MAX DISTANCE TO HUB



```

├─ app.py
├─ data
│  └─ clinics_metropolitana.csv
├─ fetch_data.py
├─ requirements.txt
├─ venv
│  ├── bin
│  ├── etc
│  ├── include
│  ├── lib
│  ├── pyvenv.cfg
│  └─ share
  
```

### Configuration

Number of Supply Hubs (k) ⓘ

8

---

### About the Model

**K-Means Facility Location** finds the optimal placement of  $k$  supply hubs such that the total distance from every health clinic to its nearest hub is minimized. This models a real-world logistics problem: where to build warehouses or distribution centers to efficiently supply all clinics with medical supplies.

### Map Legend

- Health Clinic (CESFAM / Hospital)
- Optimal Supply Hub
- Supply Chain Connection
- Hub Coverage Radius

# Health Facility Supply Hub Optimizer

Optimizing supply chain logistics for public health clinics in Santiago's Región Metropolitana using K-Means facility location modeling. Data sourced from MINSAL via [datos.gob.cl](https://datos.gob.cl).

1057

HEALTH CLINICS

8

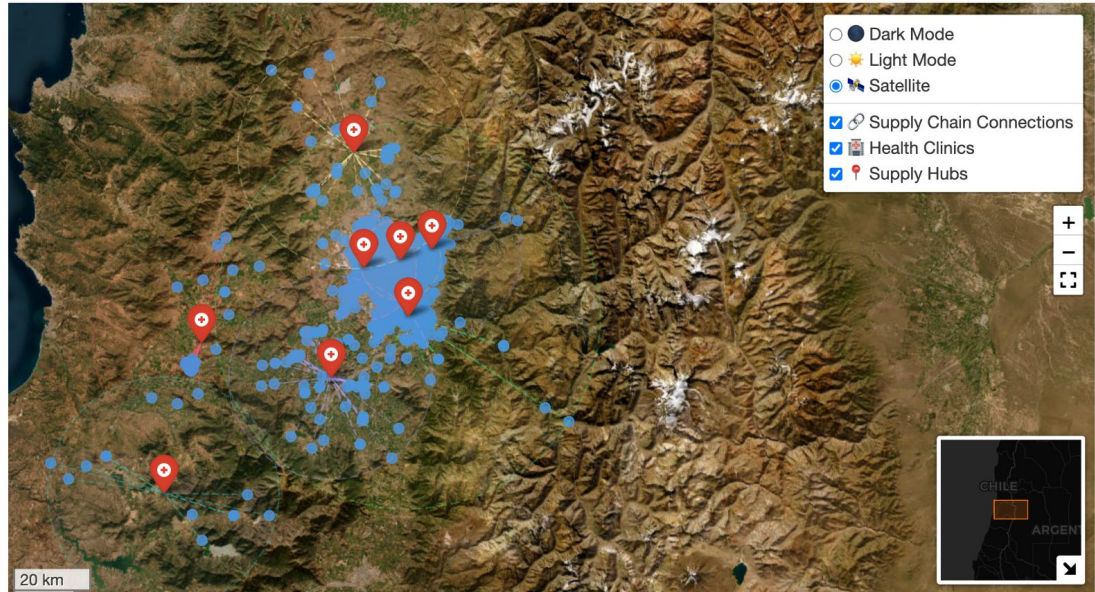
SUPPLY HUBS

5.87 km

AVG DISTANCE TO HUB

49.59 km

MAX DISTANCE TO HUB



```

├── app.py
├── data
│   └── clinics_metropolitana.csv
├── fetch_data.py
├── requirements.txt
├── venv
│   ├── bin
│   ├── etc
│   ├── include
│   ├── lib
│   ├── pyvenv.cfg
│   └── share

```

facility\_location



> data

> venv

app.py

fetch\_data.py

requirements.txt

fetch\_data.py

```

1 """
2 Fetch health establishment data from datos.gob.cl CKAN API.
3 Filters for Región Metropolitana and saves to CSV.
4 """
5 import requests
6 import pandas as pd
7 import os
8
9 API_BASE = "https://datos.gob.cl/es/api/3/action/dastore_search"
10 RESOURCE_ID = "2c44d782-3365-44e3-aefb-2c8b8363a1bc"
11 OUTPUT_FILE = os.path.join(os.path.dirname(__file__), "data", "clinics_metropolitana.csv")
12
13 REGION_CODE_METROPOLITANA = "13"
14
15 def fetch_all_records():
16     """Fetch all records from the CKAN DataStore API with pagination."""
17     all_records = []
18     offset = 0
19     limit = 500
20
21     print("Fetching health establishment data from datos.gob.cl...")
22     while True:
23         params = {
24             "resource_id": RESOURCE_ID,
25             "limit": limit,
26             "offset": offset,
27         }
28         response = requests.get(API_BASE, params=params, timeout=60)
29         response.raise_for_status()
30         data = response.json()
31
32         if not data.get("success"):
33             raise Exception(f"API returned error: {data}")
34
35         records = data["result"]["records"]
36         if not records:
37             break
38
39         all_records.extend(records)
40         offset += limit
41         print(f" Fetched {len(all_records)} records so far...")
42
43         total = data["result"].get("total", 0)
44         if offset >= total:
45             break
46
47     print(f"Total records fetched: {len(all_records)}")

```

app.py

Click to add a breakpoint

```

1 Facility Location Optimizer – Chilean Public Health Supply Network
2
3
4 Streamlit application that solves the Facility Location problem for
5 public health clinics in Santiago's Región Metropolitana.
6
7 Uses K-Means clustering to find optimal placement of supply hubs
8 and visualizes the supply chain network on an interactive Folium map.
9 """
10
11 import streamlit as st
12 import pandas as pd
13 import numpy as np
14 import folium
15 from folium import plugins
16 from streamlit_folium import st_folium
17 from sklearn.cluster import KMeans
18 import os
19
20 # -----
21 # Page configuration
22 # -----
23 st.set_page_config(
24     page_title="Health Supply Hub Optimizer – Santiago, Chile",
25     page_icon="🏥",
26     layout="wide",
27     initial_sidebar_state="expanded",
28 )
29
30 # -----
31 # Custom CSS for premium look
32 # -----
33 st.markdown("""
34 <style>
35 @import url('https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700;800&display=block');
36
37 /* Global style overrides */
38 .stApp {
39     font-family: 'Inter', sans-serif;
40 }
41
42 /* Header styling */
43 .main-header {
44     background: linear-gradient(135deg, #0f0c29 0%, #302b63 50%, #24243e 100%);
45     padding: 2rem 2.5rem;
46     border-radius: 16px;
47     margin-bottom: 1.5rem;
48     box-shadow: 0 8px 32px rgba(0, 0, 0, 0.3);

```

# Example 5: Generating Open Problems

## Prompt:

This is my DBLP page: <https://dblp.org/pid/53/1607.html>  
 please fetch the tex file from the last 5 papers on arXiv,  
 read them in detail and extract two open problems for  
 each. For each problem create an entry as an md file that  
 describes the open problem with the background and  
 mentions the paper it came from.

Paper 1: Contextual Online Bilateral Trade (2602.12903)

#	Problem	File
1	<b>Budget vs. Regret Tradeoff</b> — Characterize minimax regret as a function of allowed budget violation in one-bit feedback	problem_01_bilateral_trade_budget_regret_tradeoff.md
2	<b>Noisy/Imprecise Feedback</b> — What regret rates are achievable when valuations are only approximately linear?	problem_02_bilateral_trade_noisy_feedback_k.md

Paper 2: Autobidding Equilibria in Sponsored Shopping (2602.21966)

#	Problem	File
3	<b>Computational Complexity</b> — Is computing an autobidding equilibrium PPAD/FIXP-complete?	problem_03_autobidding_complexity.md
4	<b>Learning Dynamics Convergence</b> — Do natural autobidding algorithms converge to equilibria?	problem_04_autobidding_convergence.md

```

open-problems
├── papers
│   ├── problem_01_bilateral_trade_budget_re...
│   ├── problem_02_bilateral_trade_noisy_fee...
│   ├── problem_03_autobidding_complexity...
│   ├── problem_04_autobidding_learning_dy...
│   ├── problem_05_full_swap_regret_convex...
│   ├── problem_06_discretized_calibration_hi...
│   ├── problem_07_variance_allocation_graph...
│   ├── problem_08_variance_allocation_lower...
│   ├── problem_09_bafo_cost_variation.md
│   └── problem_10_bafo_incomplete_informat...
└── problem_10_bafo_incomplete_information.md > # Open Problem: BAFO Auctions with Incomplete Informati
1 # Open Problem: BAFO Auctions with Incomplete Information Among Sellers
2
3 ## Source Paper
4 **Procurement Auctions with Best and Final Offers**
5 Vasilis Gkatzelis, Randolph Preston McAfee, Renato Paes Leme
6 arXiv: [2503.10910] (https://arxiv.org/abs/2503.10910)
7
8 ## Background
9
10 The paper's efficiency results for BAFO procurement auctions hold under two key
11 assumptions:
12
13 1. **Full information among sellers:** Each seller knows the costs of all other
14 sellers. This enables backward induction and the computation of subgame perfect
15 equilibria.
16
17 2. **The buyer has no information** about seller costs and uses the auction for
18 price discovery.
19
20 The full-information assumption is natural in some settings (e.g., well-established
21 industries where competitors' cost structures are roughly known) but unrealistic in
22 others (e.g., innovative products, new market entrants).
23
24 The paper's main result – that every SPE of a BAFO auction is efficient – relies
25 fundamentally on sellers' ability to anticipate each other's behavior. In the
26 Name-Your-BAFO auction, a seller decides their bid knowing what subsequent sellers
27 will do (by backward induction on the game tree). In the descending auction, a
28 seller's decision to freeze depends on predicting how other sellers will respond.
29
30 ## Open Problem
31
32 **What efficiency guarantees can BAFO auctions provide when sellers have incomplete
33 information about each other's costs?*

```

# Example 6: Educational Algorithm Implementation

## Prompt (refined using Gemini):

Act as an expert in Algorithmic Game Theory and C++ programming.

**Task:** Write an educational C++ implementation of the Lemke-Howson algorithm for finding a Nash equilibrium in a 2-player bimatrix game.

**Context:** The code and its comments will serve as standalone lecture notes for students learning computational game theory. A reader must be able to read the source file from top to bottom and understand both the algorithmic mechanics and the theoretical justification.

### Specific Requirements:

- Theoretical Overview:** Begin the file with a comprehensive block comment explaining the mathematical intuition behind the Lemke-Howson algorithm. This should cover best-response polytopes, slack variables, labels, and the parity argument for why the algorithm always terminates at an equilibrium.
- Step-by-Step Annotations:** Inside the code, clearly comment on the tableau setup, the logic for choosing the entering and dropping variables, and the pivoting steps.
- Code Quality:** Use modern C++ (C++17 or later) and standard library containers (e.g., `std::vector`). Prioritize algorithmic clarity and readability over raw performance.
- Assumptions:** Explicitly state any assumptions made (e.g., assuming the game is non-degenerate) and briefly explain how degeneracy would normally be handled (e.g., lexicographic pivoting).
- Example:** Include a `main()` function that initializes a classic bimatrix game, runs the algorithm, and prints the step-by-step pivoting process alongside the

```

/*
=====
* LEMKE-HOWSON ALGORITHM – EDUCATIONAL IMPLEMENTATION
* Finding a Nash Equilibrium in a 2-Player Bimatrix Game
=====
*
* OVERVIEW
* -----
* Given a 2-player game specified by payoff matrices A (m×n) and B (m×n),
* where Player 1 chooses a row (m strategies) and Player 2 chooses a column
* (n strategies), the Lemke-Howson algorithm finds one Nash equilibrium in
* mixed strategies by following edges of a certain polytope.
*
*
* 1. BEST-RESPONSE POLYTOPES
* -----
* Assume all payoff entries are strictly positive (we can always add a
* constant to achieve this without changing equilibria).
*
* Define the "unscaled" best-response polytopes:
*
* P = { x in R^m : x >= 0, B^T x <= 1 } (constrains Player 1)
* Q = { y in R^n : y >= 0, A y <= 1 } (constrains Player 2)
*
* The origin is a vertex of both P and Q. Any non-zero vertex x of P
* can be normalized to a mixed strategy x / (sum x_i), and similarly for Q.
*
*
* 2. LABELS AND COMPLEMENTARY SLACKNESS
* -----
* We introduce slack variables:
*
* Tableau T1 (n rows): B^T x + s = 1, x >= 0, s >= 0
* Tableau T2 (m rows): A y + r = 1, y >= 0, r >= 0
*
* Each variable carries a "label" from {1, ..., m+n}:
*
*/

```

# Example 6: Educational Algorithm Implementation

```

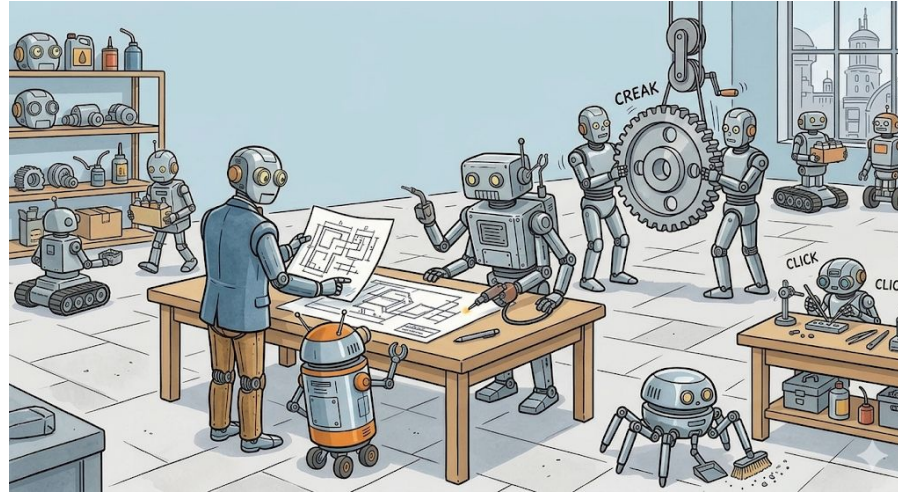
// -----
//  lemke_howson
//  -----
//  Runs the Lemke-Howson algorithm on the bimatrix game (A, B).
//
//  Parameters:
//  ^
//  drop_label  m×n payoff matrix for Player 1
//  drop_label  m×n payoff matrix for Player 2
//  drop_label - the initial label to drop (1 to m+n)
//  verbose    - if true, print every pivot step
//
//  Returns: (x, y) where x is Player 1's equilibrium mixed strategy
//           and y is Player 2's equilibrium mixed strategy.
//  -----
std::pair<std::vector<double>, std::vector<double>>
lemke_howson(const std::vector<std::vector<double>>& A,
             const std::vector<std::vector<double>>& B,
             int drop_label,
             bool verbose = true)
{
    const int m = static_cast<int>(A.size()); // Player 1 strategies
    const int n = static_cast<int>(A[0].size()); // Player 2 strategies

    // -----
    //  Shift payoffs so that every entry is strictly positive.
    //  This is required so that the polytopes P and Q are bounded and
    //  the minimum-ratio test is always well-defined.
    //  Adding a constant to all entries does not change the set of NE.
    //  -----
    double min_A = 0.0, min_B = 0.0;
    for (int i = 0; i < m; ++i)
        for (int j = 0; j < n; ++j) {
            min_A = std::min(min_A, A[i][j]);
            min_B = std::min(min_B, B[i][j]);
        }
}

```

# Conclusion

- Don't assume the systems can't do something. Models are getting better very fast.
- But getting what you want may require many iterations (back-and-forth, corrections, restarts, novel agentic loops)  
→ new kinds of difficulty
- How you ask matters a lot. Try different prompt variations and ask Gemini to refine prompt ideas. Try different systems.
- How the research community should evolve in the presence of AI is a big open question.



**Thanks!**