

Using AI Agents **Effectively** for Computational Science

*Personal experience using AI to assist research
in March and April, 2026: date matters as AI evolves quickly*

Xiangxiong Zhang, Department of Mathematics, Purdue University

CoE ThinkAI Initiative Workshop, Purdue University, June 9 2026

Plan of the talk

1. Examples of how I used AI in Mar/April 2026 for my own research projects.
2. My research expertise is focused on scientific computing and numerical analysis.
3. Hopefully it is useful for many of the audience.

Personal Timeline: I was in denial mode until Jan 2026

How my use of AI tools has evolved

- ▶ I tried Claude Code in Jan, 2026 and I went from Denial to Acceptance. The following is taken from slides by Javier Gomez-Serrano (Brown University)

The Mathematician's Stages of Grief (2022–2026)

- 2022 **Denial:** “AI will never work in mathematics.”
- 2023 **Anger:** “It is just a stochastic parrot with a GPU!”
- 2024 **Bargaining:** “Maybe it can just write my grant applications?”
- 2025 **Depression:** “It just formalized a proof I’ve been stuck on for months.”
- 2026 **Acceptance?**

Acceptance: 50% Loaded...

(Estimated time remaining: 5 years)

I don't dare to make predictions beyond 12 months anymore.

Focus of this talk is NOW. Not past or future. No hype or denial.

- ▶ Tech Details are not that important since AI tools are changing exponentially.
- ▶ Focus on what AI can do for us **right now**, rather than hyping it up or denying it.
- ▶ Programming in 1964 was punching cards. Beyond 2026, it could be natural language.



ibm 029 (1964)



Any questions are welcome but I'm not an expert of AI

This talk is NOT about Automatic Research.

Which AI tool is better: start with one then try more later.

How to use AI: you will get some answers during the talk.

Quick questions/remarks are welcome any time during the talk.

Save long questions/comments for the Q & A at the end.

Chatbot (a talker) V.S. AI Agent (a talker and also a doer)

Understanding the distinction

Chatbot

let $z=x+iy$ be a complex number, is $\text{Arg}(z)=\arctan(y/x)$ correct?

Short answer: **not exactly** — it's incomplete.

If

$$z = x + iy \neq 0,$$

then the argument of z satisfies

$$\tan(\text{Arg}(z)) = \frac{y}{x},$$

but

$$\text{Arg}(z) \neq \arctan\left(\frac{y}{x}\right)$$

in general.

Why not?

The function



AI Agent

```
in which case, please supply explicit cell edges to pcolormesh.
... +58 lines (ctrl+o to expand)
└ (timeout 10m)

The optimizer is now moving (E drops from 1743 → 393), but it's stuck far from E* = 33.80. The issue is the normalization constant in log_ref is wrong. Let me derive it carefully:

For Beta(5,5) on (-1,1): the integral is ∫_{-1}^1 (-1)^1 (1-z^2)^4 dz = 512·B(5,5) = 512·(4!)^2/9! = 512/630 = 256/315 ≈ 0.813. So the correct constant is C_1 = 315/(256·L), not 35/(2·L). The plan's derivation was wrong (it confused 9! with 8!).

With the wrong C_1 being ~14× too large, F_R = β/4N ∫ λ(z_i)/exp(ld) is scaled by ~14×, wildly distorting the gradient.

• Update(gpe3d_groundstate_pwgf.py)
└ Added 4 lines, removed 3 lines

def log_ref z:
    """log lambda l(z). Beta(5,5)^3 on (-L,L)^3. z: (N,3)
133 - Normalization: int_{-L}^L (1-z^2/L^2)^4 dz = 2L/35, so C_1 = 35/(2L).
134 - log C_3 = 3*log(35/(2L))."""
135 - log_C3 = 3.0 * torch.log(torch.tensor(35.0 / (2.0 * L), dtype=z.dtype))
136 + Normalization: int_{-L}^L (1-z^2/L^2)^4 dz = 2L * 256/315
137 + (since int_{-1}^1 (1-t^2)^4 dt = 512*B(5,5) = 512*(4!)^2/9! = 256/315).
138 + C_1 = 315/(256*L), so log C_3 = 3*log(315/(256*L))."""
139 + log_C3 = 3.0 * torch.log(torch.tensor(315.0 / (256.0 * L), dtype=z.dtype))
140 + return float(log_C3 - 1) # = 4
141 + return torch.exp(log_C3 * torch.tensor(1.0 - torch.tensor(0.0)**2 / (**2 + 1e-12)

• Update(gpe3d_groundstate_pwgf.py)
└ Added 4 lines, removed 4 lines

# Beta(5,5): C_1 = 35/(2L), exponent 4
492 - fac1 = (35.0 / (2.0 * L)) * (1.0 - (z1_flat / L)**2) ** (ALPHA_REF - 1)
493 - fac2 = (35.0 / (2.0 * L)) * (1.0 - (z2_flat / L)**2) ** (ALPHA_REF - 1)
494 - fac3 = (35.0 / (2.0 * L)) * (1.0 - (z3_flat / L)**2) ** (ALPHA_REF - 1)
495 + # Beta(5,5): C_1 = 315/(256*L), exponent 4
496 + fac1 = (315.0 / (256.0 * L)) * (1.0 - (z1_flat / L)**2) ** (ALPHA_REF - 1)
497 + fac2 = (315.0 / (256.0 * L)) * (1.0 - (z2_flat / L)**2) ** (ALPHA_REF - 1)
498 + fac3 = (315.0 / (256.0 * L)) * (1.0 - (z3_flat / L)**2) ** (ALPHA_REF - 1)
499 + return torch.exp(log_C3 * torch.tensor(1.0 - torch.tensor(0.0)**2 / (**2 + 1e-12)

• Bash(time /Users/zhan1966/.julia/conda/3/aarch64/bin/python3 gpe3d_groundstate_pwgf.py 2>&1)
└ Running_ (16s · timeout 10m)
ctrl+b to run in background
```

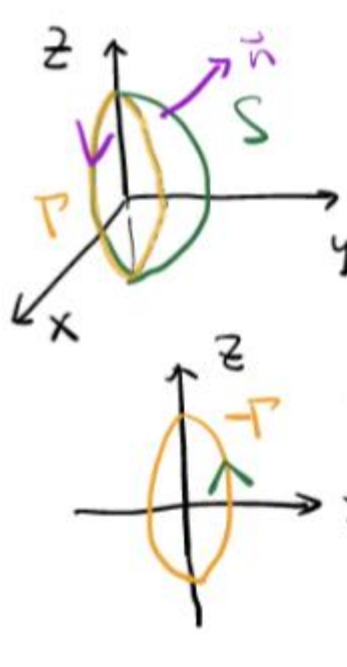
► Key takeaway: chatbot is your advisor, agent is your advisor & co-worker reading/editing your files and papers, running/debugging codes, etc.

AI chatbot can fail even for Calculus

For this vector calculus problem, the correct sol is 4π , and ChatGPT (Plus, \$20 per month) gave -4π , because it cannot figure out correct orientation of Gamma, even after I pointed its mistake. But Grok and Claude both get it right. Gemini and DeepSeek made the same mistake

S is the right half of $x^2 + \frac{y^2}{4} + \frac{z^2}{4} = 1$ (i.e., the half represented by $y \geq 0$). Find the flux $\nabla \times \vec{F}$ through S with normal vector pointing to the right, where $\vec{F} = \langle z, e^{y^2}, -x \rangle$.

Correct Solution:



$$\iint_S \nabla \times \vec{F} \cdot \vec{n} \, dA \quad (\text{Stokes Thm})$$

$$= \oint_{\Gamma} \vec{F} \cdot d\vec{r}$$

$$\Gamma: x^2 + \frac{z^2}{4} = 1$$

$$\vec{r}(t) = \cos t \hat{i} + 0 \hat{j} + 2 \sin t \hat{k}$$

$$0 \leq t \leq 2\pi$$

$$\vec{F} = z \hat{i} + e^{y^2} \hat{j} + (-x) \hat{k}$$

$$d\vec{r} = -\sin t \hat{i} + 0 \hat{j} + 2 \cos t \hat{k}$$

$$\oint_{\Gamma} \vec{F} \cdot d\vec{r} = - \oint_{\Gamma} \vec{F} \cdot d\vec{r}$$

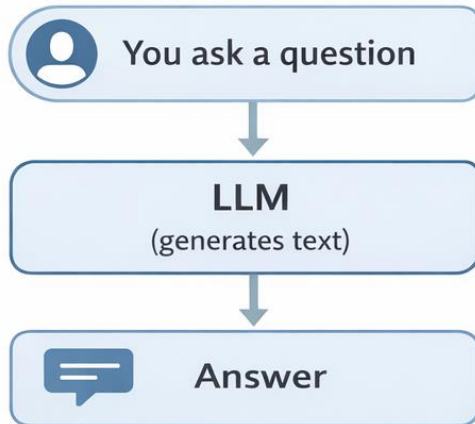
$$= - \int_0^{2\pi} (-2 \sin^2 t - 2 \cos^2 t) dt$$

$$= 4\pi$$

Chatbot (ChatGPT or Claude) v.s. AI Agent (Codex or Claude Code)

You can debate with a chatbot on whether $1+1=3$
AI Agent can write a code then run it to show you $1+1=2$

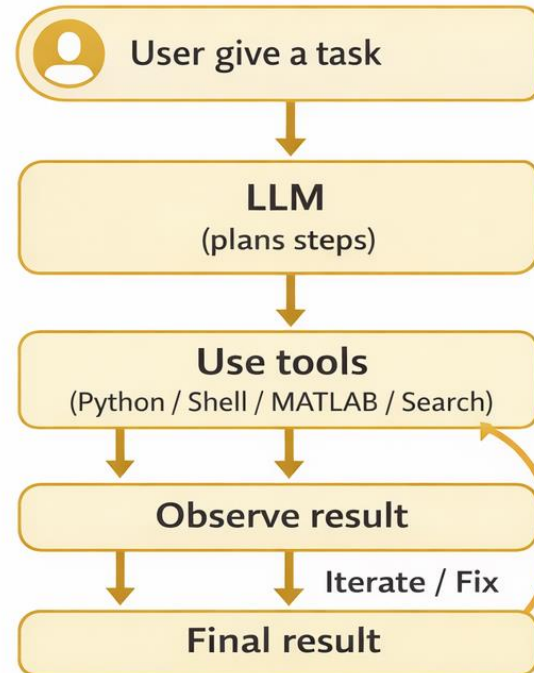
CHATBOT



Chatbot = text in → text out

No actions, no tools, no environment interaction.

AI AGENT



Agent = plan → act → observe → iterate



AUTOMATIC ITERATION

If the code has errors or the output isn't right:

1. LLM reads the error message
2. Edits the code to fix it
3. Runs it again on your computer
4. Repeats until success

No manual copy-paste needed!

A chatbot talks. An agent acts.

Free AI agent tool by OpenAI: Codex App on Mac

The screenshot displays the OpenAI Codex application on a Mac. The top menu bar includes 'Codex', 'File', 'Edit', 'View', 'Window', and 'Help'. The sidebar on the left contains options for 'New thread', 'Automations', 'Skills', and 'Threads'. The 'code' folder is selected, showing a list of threads: 'Locate current folder' (with a +59 -1 change and 1h ago) and 'Review Python code in ./code' (1w ago). The main workspace shows a chat thread with the following content:

Locate current folder code ...

The metric G is very much ill-conditioned, let's try the pseudo-inverse computed via SVD as the preconditioner in the minimal residue method for inverting G ?

Worked for 4m 44s >

Implemented and tested.

What I changed:

- Added a new natural-gradient solver option in `gpe2d_pwgf_blockpcg.py`:
 - `--natgrad-solver minres_svd`
 - Uses MINRES with an SVD pseudo-inverse preconditioner for damped G .

Run used:

- `K=400, N=3000, H=10, N_ODE=10`
- Solver: `minres_svd`

Results:

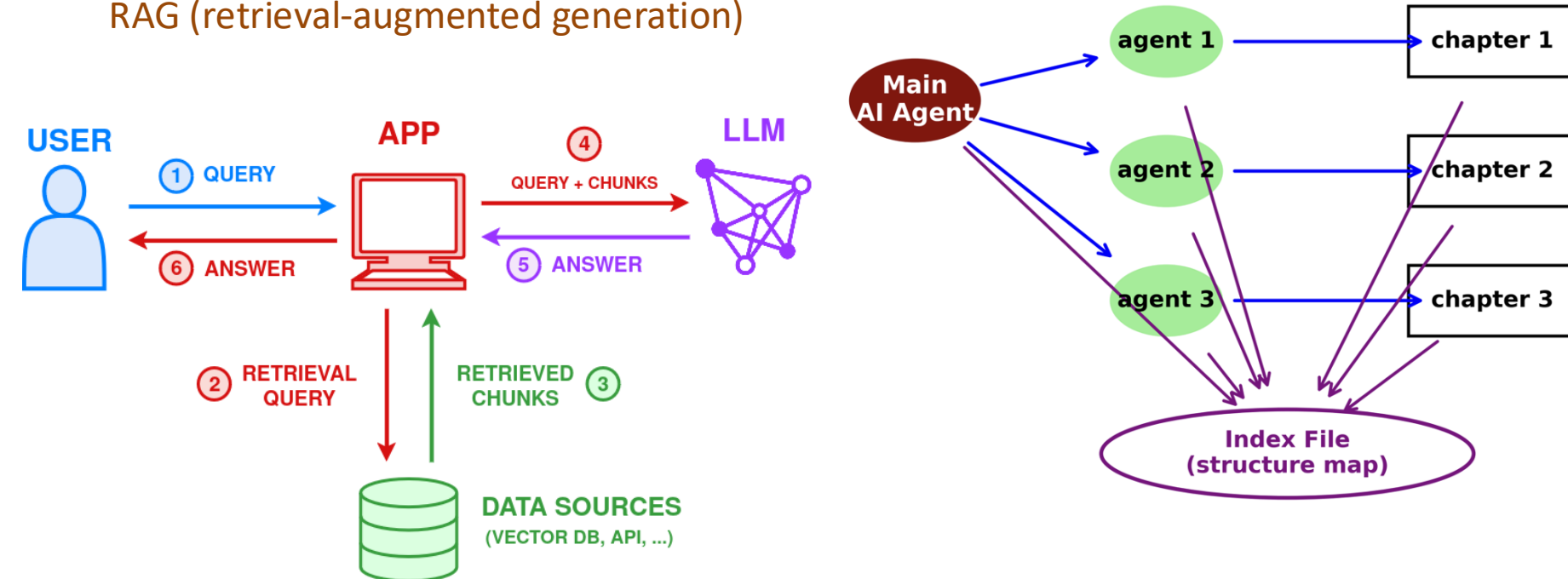
- PWGF best energy: `0.223776`
- FD warm-start energy on `n=200` : `0.2388311850`
- Runtime: `163.2s`

First Example: Proofreading a Grad Textbook in 1 Hour

AI agent proof read the final version of my grad level textbook (300+ pages).

1. Claude Code first divided the big PDF into a dozen small files by chapters/sections.
2. An index file was created so that it knew the connections of different files.
3. Claude Code launched different sub-agents for different chapters.
4. In this example, those smaller PDF files are **data sources** in RAG.

RAG (retrieval-augmented generation)



Errata — Ubiquitous Laplacian

Errata Report

Ubiquitous Laplacian: An Introduction to Numerical Methods with Applications in Data Science

Rongjie Lai and Xiangxiang Zhang

Publisher's Final Proof Review — Coverage: all 308 pages

#	Page / Location	Issue	Fix
1	p. 3, Figure 1.1, subcaptions (b) (c)	Subject-verb disagreement in both captions.	(b) are
2	p. 27, §2.5.5	B is called "positive definite" but was identified as singular two sentences earlier.	Re "re"
3	p. 37, §2.10, 9-point Laplacian stencil	$U_{i+1,j+1}$ appears twice; $U_{i-1,j+1}$ is missing.	Re with
4	p. 48, 68–69, §§3.2.1, 3.6 (≥ 3 occurrences)	"Cauchy-Schwartz" — non-standard spelling.	Re "Ca"

How to Learn the Details of AI Tools? Ask ChatGPT

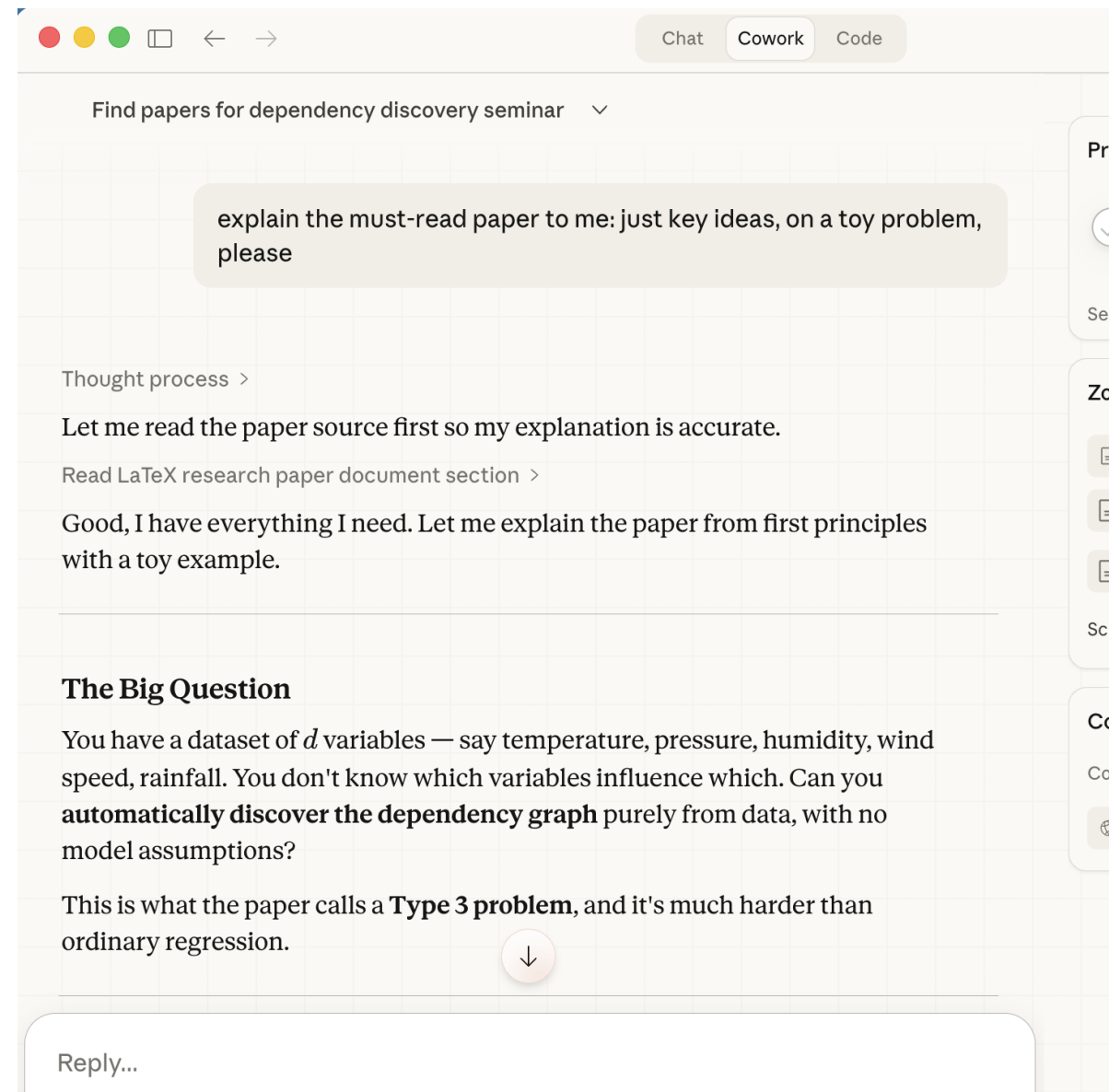
- No need to remember details of how to use specific AI tools.
- Details in this talk are probably outdated anyway as AI evolves.
- When using AI tools, we can ask ChatGPT how to use them.
- Previously I asked ChatGPT to write prompts for various AI tools. But Claude Code has evolved so that even this can be skipped.
- Remember that ChatGPT can be wrong, or even worse: they can lie.

Research/Learning Example: Seminar Preparation

1. I ask AI to find me papers to read.
2. AI reads it then explains the paper to me.
3. I ask AI to do a toy problem for me.
4. I explore ideas and new directions.
5. AI writes a latex doc (my cheat sheet).
6. I'll have a better discussion with speaker.

If I'm already kind of familiar with relevant materials, and if I have time, I do this before I go to a talk.

If not, and if I find the talk interesting enough, I can do this after/during the talk.



The screenshot shows a chat window with a search bar containing "Find papers for dependency discovery seminar". The chat history includes a user message: "explain the must-read paper to me: just key ideas, on a toy problem, please". The AI response includes a "Thought process" section: "Let me read the paper source first so my explanation is accurate." followed by "Read LaTeX research paper document section". The main response text is: "Good, I have everything I need. Let me explain the paper from first principles with a toy example." Below this is a section titled "The Big Question" with the text: "You have a dataset of d variables — say temperature, pressure, humidity, wind speed, rainfall. You don't know which variables influence which. Can you automatically discover the dependency graph purely from data, with no model assumptions?" and "This is what the paper calls a **Type 3 problem**, and it's much harder than ordinary regression." A "Reply..." input field is visible at the bottom.

Research Example: I used AI to write a paper in March

Concrete examples from my own workflow

- ▶ Day 0 (Sunday): AI explained the paper and wrote toy codes.
- ▶ Day 1 (Monday): I talked to the CCAM seminar speaker for 30 minutes.
- ▶ Day 2 (Tuesday): 10pm, I asked whether my idea can work. AI said yes and coded it for me.
- ▶ Day 3 (Wednesday): by 8pm, my idea was already fully tested on 3D benchmark tests.
- ▶ Day 4 (Thursday): by 10pm, full complete draft was already written.
 - Math/Code is correct? I have to know how to check.
 - Does not read like AI wrote it? (Guess how I did it)
 - No hallucinations in the references? (Guess how I did it)

95%~99% positive answers, **if using AI properly.**

It took me few weeks to pick it up.

Tools for research: Claude Code+Codex+ChatGPT+Server

Black session: Claude Code writing idea/paper

Green session: Claude Code writing Python code, submitting it to GPU server to run.

Blue one: I monitor the GPU server to see if the job goes wrong.

Light color one: Codex audits the code Claude wrote.

Ask ChatGPT for prompts if needed.

The image displays four terminal windows illustrating a research workflow:

- Top Left (Black):** Claude Code writing a paper. The terminal shows a section titled "Direct solver for $-\Delta + V$ with a separable potential" and a bash command to compile a LaTeX document: `Bash(pdflatex -interaction=nonstopmode paper.tex > /dev/null 2>&1 && pdflatex -interaction=nonstopmode paper.tex 2>&1 | tail -3)`. It also shows the output of the compilation and a note about the GitHub URL placeholder.
- Top Right (Green):** Claude Code writing Python code and submitting it to a GPU server. The terminal shows the code for `inverse_iter_paper.py` and the submission command: `python inverse_iter_paper.py --q_order 10 --n_cell 60 --precision fp 64`.
- Bottom Left (Light Yellow):** Codex auditing the code. The terminal shows the `OpenAI Codex` interface with the model `gpt-5.4 high` and the directory `/`. It also shows the tip: "New Try the Codex App with 2x rate limits until April 2nd. Run 'codex app' or visit <https://chatgpt.com/codex?app-landing-page=true>".
- Bottom Right (Blue):** Monitoring the GPU server. The terminal shows the output of the `ssh` command, displaying a list of processes running on the server, including `python` and `ssh`. The output shows the server's IP address, user, and various system metrics.

What it looked like during those 48 hours: not exactly Vibe Research

- I spent 15-20 minutes to write down clearly what I want AI to implement.
- AI can finish the implementation, in minutes or hours (weeks or months of work for me).
- It may or may not be satisfying. If not, ask it to improve the results.



* Churned for 39s

```
> Now let's go back to work on paper.tex, which is a serious research paper about applying 1D TV Lasso as a non-oscillatory limiter for solving conservation laws. Can you first switch Section 2 and Section 3? New section 2 should be about introducing the TV denoising problem as a non-oscillatory to a numerical scheme for conservation laws, and we can use spectral method as an example (current section 3), though in principle this limiter can be applied to many other schemes such as finite difference or cell averages of DG polynomials. Then new Section 3: in the beginning, mention that the idea of using TV denoising for limiter is certainly not new at all, but the main difficulty is always computational cost, so this section will explain to the PDE audience about popular 1D TV models and the classical solvers (current Section 2, and appendix, you can merge the current appendix into this new Section 3), then introduce Differential Inclusion method (you can merge the current Section 4 into it here) and show readers its performance (merge/move all non-PDE related numerics here).
```

• Read 1 file (ctrl+o to expand)

• This is a major restructuring. Let me first understand the full layout, then execute the reorganization step by step.

Current structure:

Research Example: I still need to polish it, but 95%~99% was done

1. Co-author and I spent one hours discussing it, another few hours to polish it. Afterwards it is ready for submission.
2. Both co-author and I are happy about this paper.
3. Both of use still feel necessary to polish it, but it is light polishing.
4. AI save both of us lots of time, because neither of us would write this paper without help from AI.
5. Lucky on this paper, because AI failed for many other ideas/projects.

A (small) paper in 2 days is possible because I am an expert of relevant math subject and I know how to use AI properly (otherwise it's waste of time)

- Critical condition 1: it is still within my domain of expertise.
- Critical condition 2: I know current AI's capabilities and limitations.
- Critical condition 3: I know the literature well (I had read literature carefully).
- Critical condition 4: I have plenty of numerical experience (I wrote many codes).

I can use AI well for this particular research topic because I did my homework.

Spending enough time on studying literature should be first step of research, which is often skipped/ignored by students or even junior researchers.

- **AI is not perfect: I had to feed it my old code and my previous papers.**
- **AI accelerated implementation and validation of my idea.**
- **AI did help a bit on math:** neither I nor co-author had time/energy to derive the math formulae, though we both are capable. Everyone is busy nowadays.

Research Example: a reading/learning seminar slides

- I want to read a few hard but exiting new papers with students
- AI helped me build a very nice slides for it (**guess how long this took me to build?**)
- <https://www.math.purdue.edu/~zhan1966/teaching/574/2026/Bolte/slides/index.html>

Convergence of Optimizers for Nonconvex Problems in Machine Learning

Slides for an informal reading seminar at Purdue University, Spring 2026. Email the author, [Xiangxiong Zhang](#), for time and location. Grounded in Bolte & Pauwels (NeurIPS 2020); Bolte, Le, Pauwels & Silveti-Falls (NeurIPS 2021); Bolte, Pauwels & Vaiter (NeurIPS 2022); and KL framework papers (2007–2014).

Intro 10 11 12 13 14 | Preliminaries p1 p2 p3 | Review r1 r2 r3 | Overview o1 o2 o3 o4 | Part I 1 2 3
4 5 6 7 8 | Part II 9 10 11 12 13 14 | Part III 15 16 17 18 19 20 21 22 23 | Part IV 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 Refs

Introduction

Slide 10: What This Seminar Is About Introduction

Four themes of this seminar, each requiring new mathematical tools:

- 1. Smooth
- 2. Convex nonsmooth
- 3. Nonconvex nonsmooth
- 4. Automatic differentiation

Slide 20 — Proof Roadmap for Theorem 2.9

Proof roadmap (5 steps):

1. **Monotone values.** H1 implies $f(x^k)$ is nonincreasing. Combined with H3, $f(x^k) \rightarrow f(\bar{x})$ and $f(x^k) \geq f(\bar{x})$ for all k .
2. **Enter the KL regime.** For k sufficiently large, $x^k \in U$ and $f(\bar{x}) < f(x^k) < f(\bar{x}) + \eta$, so the KL inequality applies: $\varphi'(f(x^k) - f(\bar{x})) \cdot \text{dist}(0, \partial f(x^k)) \geq 1$.
3. **Telescoping inequality.** Apply H2 (bounding $\text{dist}(0, \partial f(x^{k+1}))$) and the concavity of φ to derive (schematically; see exact constants):
$$\|x^{k+1} - x^k\| \leq C[\varphi(f(x^k) - f(\bar{x})) - \varphi(f(x^{k+1}) - f(\bar{x}))] + C'\|x^k - x^{k-1}\|.$$
The right side telescopes when summed.
4. **Finite length.** Summing the telescoping inequality yields $\sum_k \|x^{k+1} - x^k\| < +\infty$. This means (x^k) is a Cauchy sequence (complete), so it converges.
5. **Criticality.** H3 plus the closedness of $\text{graph}(\partial f)$ ensures the limit \bar{x} satisfies $0 \in \partial f(\bar{x})$.



Where KL is used: Exclusively in Step 2–3. The KL inequality converts the “decrease in $\varphi \circ (f - f(\bar{x}))$ ” into a bound on step sizes, enabling the telescoping argument.

Attach, Bolte & Svaiter, Math. Programming 137, 2013, Lemma 2.6 (pp. 100–103).

More Example: I tried to do coding homework in a CS course

- Then I tried to Claude Code to do all the homework in Stanford CS 336 (guess how long?)

CS336 Schedule Coursework

 NLP CS336: Language Modeling from Scratch 

Stanford / Spring 2025

This is the archived website for the Spring 2025 offering of CS336.
The latest offering is [here](#).

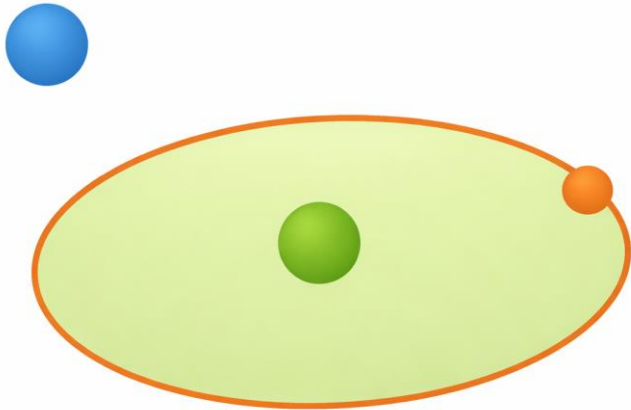
Coursework

Assignments

- **Assignment 1: Basics** [[leaderboard](#)]
 - Implement all of the components (tokenizer, model architecture, optimizer) necessary to train a standard Transformer language model.
 - Train a minimal language model.
- **Assignment 2: Systems** [[leaderboard](#)]
 - Profile and benchmark the model and layers from Assignment 1 using advanced tools, optimize Attention with your own Triton implementation of FlashAttention2.
 - Build a memory-efficient, distributed version of the Assignment 1 model training code.
- **Assignment 3: Scaling**
 - Understand the function of each component of the Transformer.
 - Query a training API to fit a scaling law to project model scaling.
- **Assignment 4: Data** [[leaderboard](#)]
 - Convert raw Common Crawl dumps into usable pretraining data.
 - Perform filtering and deduplication to improve model performance.

3 examples: inside/outside of my knowledge domain

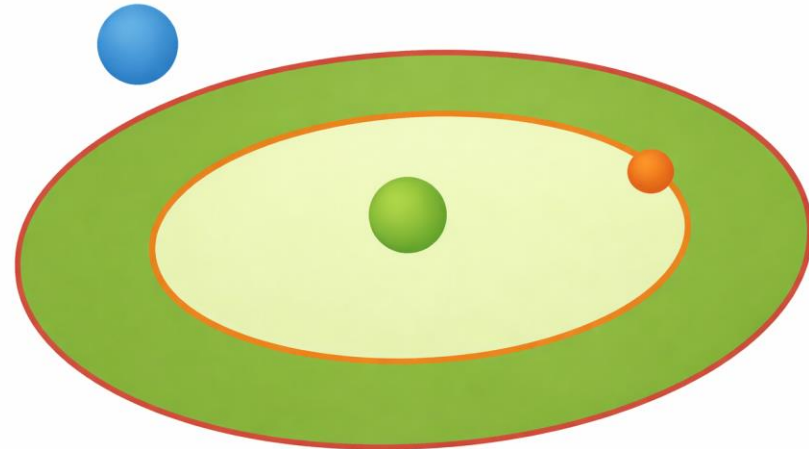
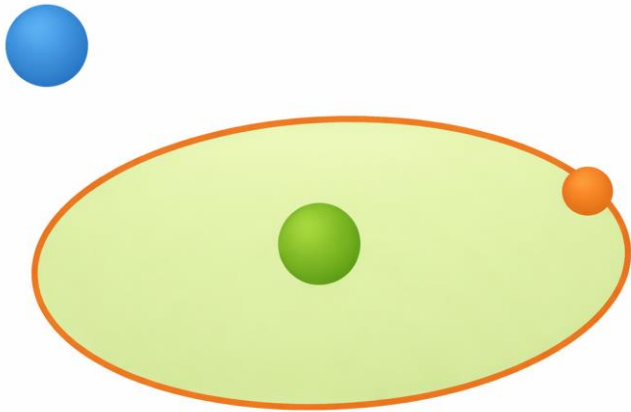
1. Write a paper inside domain of knowledge: 2 days.
2. Prepare slides for learning new papers: 3 weeks.
3. Do Stanford CS homework: infinity, I stopped at HW#1. The code was running for too long because AI did not use optimal algorithms





3 examples: inside/outside of my knowledge domain

AI boosts productivity-> larger knowledge domain ->use AI on more tasks effectively

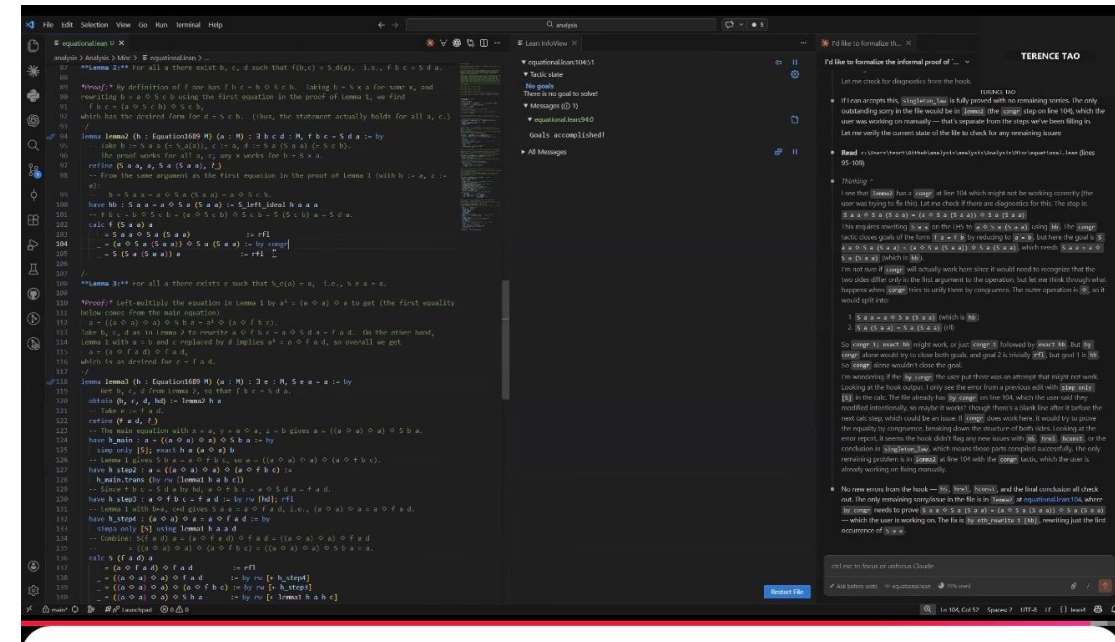


Limitations: AI Makes Mistakes

- For grad level applied math, Claude Code Opus 4.6 can struggle.
- It made the same mistake as my PhD students did.
- I had to teach it multiple times like how I have taught students.
- **Mistakes cannot be excuses of not using it.**
- Professors/Textbooks/Wikipedia/Papers can be wrong.

Problem-Solving Strategy: First Principle

- Terry Tao Posted YouTube on Mar 7, 2026 using **Lean+Claude Code** for formalizing a proof.
- His first attempt: one prompt, 45 minutes, failed.
- His second attempts: multiple steps, 25 minutes, worked.



Formalizing a proof in Lean using Claude Code

@TerenceTao27 68K views 2d ago ...more

Subscribe 3K Save

What is Changed in the AI Era?

1970s: no email, no internet, no LaTeX,
no personal computer, no smartphone, no Google.
Revising a draft meant retyping entire pages.
Sharing results overseas meant waiting weeks for mail.



1980s

Typewriters,
physical libraries,
postal mail

1990s

Email, early web,
LaTeX, Personal Computer

2010s

Smartphones, Google,
GPU/cloud computing

2020s

AI chatbots & agents,
automated workflows

Tools change, but the fundamental workflow never changes:

Idea → Implementation → Validation

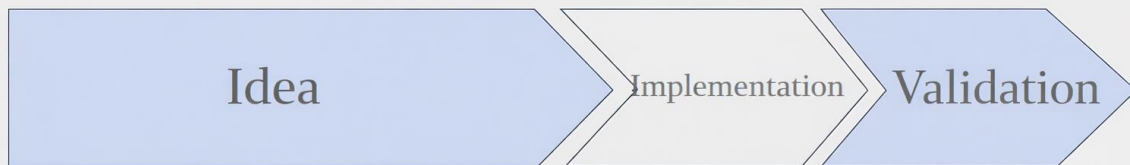
For students: learning is more important in AI era

- The faster **implementation** is, the more important **ideas** become.
- **Better learning** (classroom/book/seminar/conference) => **Better ideas**.
- **In AI era, learning is more important.**
- **Never AI everything: I tried to AI everything in Feb, I became dumber.**
- **But do consider using AI to assist learning.**

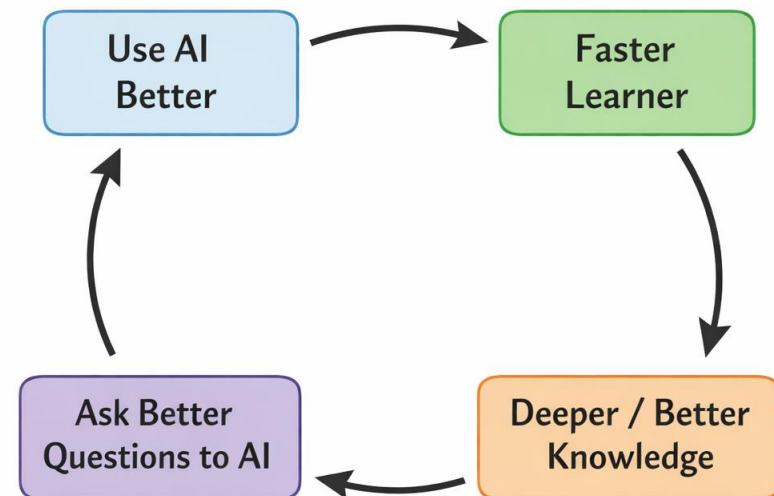
Without AI:



With AI:



Positive AI Learning Feedback Loop



Worked Learning Example for Statistics Students

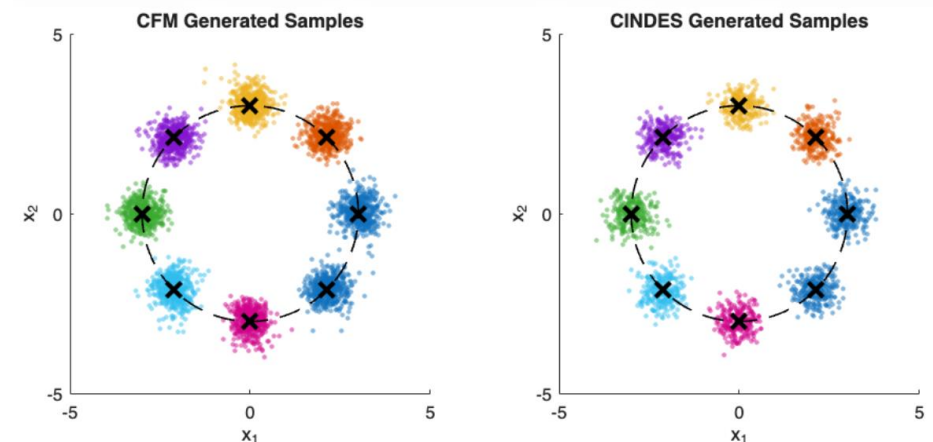
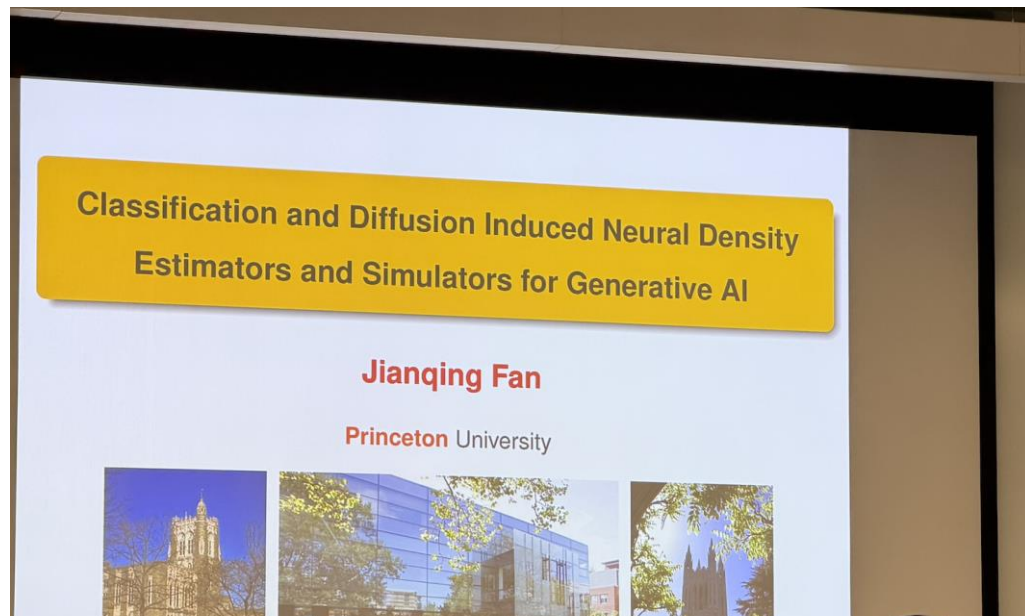
I'll always be a student when going to statistics talks.

CoS Frederick L. Hovde Distinguished Lecture on March 11, 2026

Step 1: I downloaded the latex source of the paper from arXiv

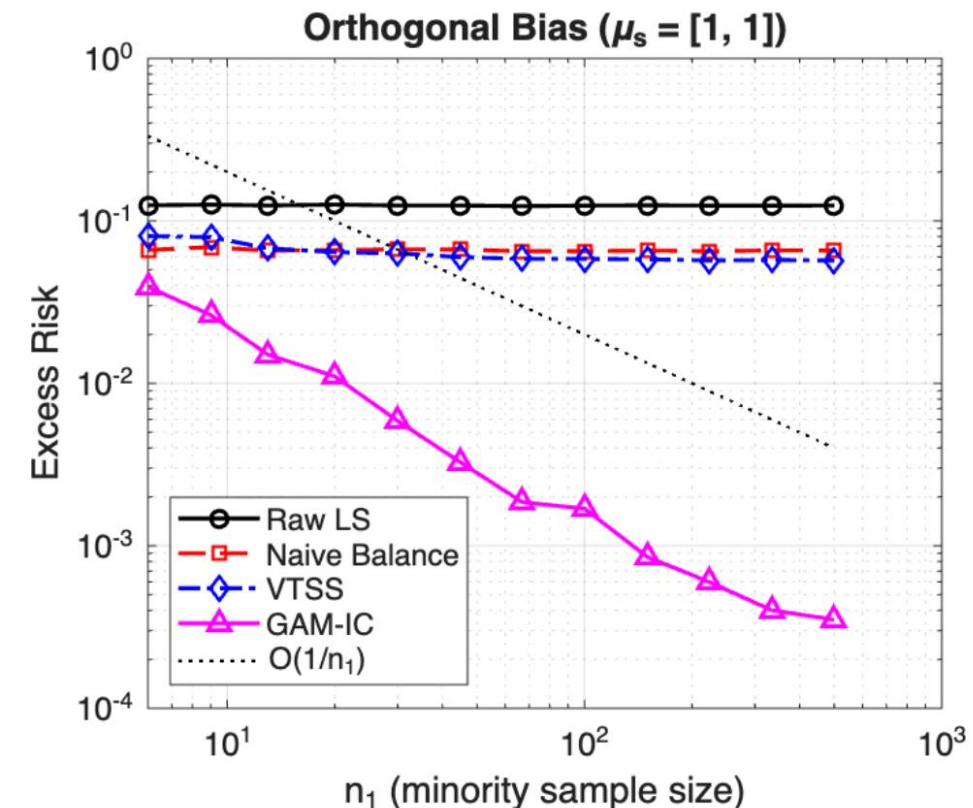
Step 2: I asked AI to read/explain it, then write code to reproduce it.

Still requires hard work to go for full understanding, but already much easier.



Worked Research Example for Statistics Faculty

Prof. Xiao Wang (Purdue Stat) attended a talk on generative model in conference. He tried an idea on my claude code: maybe that model can be easily improved. After one hour, it looks working, but requires careful checking.



How to build a mentoring Agent for research

All the worked examples clearly show survivor bias. Many of my ideas did NOT work when I used AI to explore.

1. What if some idea works, but not worthy for me to wrap up as a paper?
2. I ask Claude Code to create a new CLAUDE.md (a text file, default setup file to be loaded every time Claude Code is opened), to pass to students.
3. This is what I want to do: I want students to finish those small ideas, either as undergrad projects or as grad training projects.
4. I send students a folder with reference papers, CLAUDE.md, codes and latex files of the working small ideas.
5. Students open Claude Code in that folder: their AI know all my AI knows.
6. **Win-win: I don't need to explain basics to students, and students get much better explanations from AI about basics.**

OpenClaw? Fully Autonomous? Too much RISKS

- OpenClaw is too open to be safe for me.
- I tried some **fully autonomous mode for a specific task** in Claude Code.
 1. I was simulating Schrodinger equation on GPU servers (DeltaAI at UIUC).
 2. It took me a day to interact with Claude Code to finish 4D computation.
 3. It took a day because AI made many mistakes.
 4. But once we (me and AI) finish 4D, AI should know how to do 6D.
 5. I let Claude Code to do 6D fully by itself while I went to sleep.
 6. I expected it to take 7~8 hours to finish, but it took only 1 hour.
- **The risks I took: it could use up my GPU hours on server, or my token limits; it may also cause trouble to my laptop. I was prepared that it might burn 300 GPU hours, but I was lucky that it used only 15 hours in the auto mode.**

AI submitted 25 jobs to two GPU servers and finished the job in 1 hour

- I tried some **fully autonomous mode** for a specific task in Claude Code.
- Results are quite satisfying for me: 1 hour in total, costing 15 GPU hour on DeltaAI (GH200). Majda (A100) is my departmental testing server.

Total time: ~1 hour

Job submissions:

Server	Jobs	Purpose
Majda	2	Ground state ($\delta=0.1, 0.01$)
Majda	2	qHOP M=1,3 debug
Majda	2	qHOP M=5,7 debug
Majda	2	Magnus-2 M=1,3 debug
Majda	2	Magnus-2 M=5,7 debug
DeltaAI	2	Ground state n=39 (OOM, failed)

DeltaAI	2	Ground state n=29
DeltaAI	2	qHOP M=1,3 complex128 (OOM, failed)
DeltaAI	1	qHOP M=1 complex128 retry (cancelled)
DeltaAI	2	qHOP M=1,3 complex64
DeltaAI	2	qHOP M=5,7 complex64
DeltaAI	2	Magnus-2 M=1,3 complex64
DeltaAI	2	Magnus-2 M=5,7 complex64
Total	25	(5 failed/cancelled, 20 successful)

Example: Scraping Data Behind a Paywall

The image shows a browser window with the URL `mathjobs.org/jobs?list-230----20`. The page displays a list of job applications with columns for Received, LastUpdated, Degree, Year, and Degree_Institution. A Claude AI interface is overlaid on the right, showing a task to save applicant data and a plan to automate the process.

Browser Address Bar: `mathjobs.org/jobs?list-230----20`

Page Header: MATHJOBS.ORG

Page Title: List [download, xls, ratings, Zip || PDFs, map] [doc]

Navigation: K L M N O P Q R S T U V W X Y Z + All [GOLOMB POPPIN : Active Inactive]

Table:

Received	LastUpdated	Degree	Year	Degree_Institution
2025/11/27	2025/11/27	PhD Mathematics	2026/05 exp	University of South Carolina
2025/11/07	2025/12/19	PhD	2026	University of Sheffield School of Mathematics and Physical Sciences
2025/12/05	2025/12/17	PhD	2021/11	Université de Bordeaux Institut de mathématiques de Bordeaux
2025/11/28	2026/02/09	PhD	2022/08	Ohio State University
			2026/05	University of Maryl

Claude Interface:

Sonnet 4.6

save in a folder /Desktop/applicant, just the current 20 applicants

Claude's plan

Allow actions on these sites

- mathjobs.org

Approach to follow

- For each of the 20 applicants, navigate to their individual page
- Find the cover letter PDF link on their page
- Download the cover letter PDF and save it to /Desktop/applicant/ named after

Approve plan

Make changes

Claude will only use the sites listed. You'll be asked before accessing anything else.

Example: Monitoring Junk Mail

I have heard of at least two people who missed academic job interview/offer emails this year, because they didn't check their Junk box.

The screenshot shows the Outlook web interface. The left sidebar displays the 'Junk Email' folder with 6385 items. The main content area shows a 'Notes' section with a 'Loading your notes...' message. On the right, a 'Claude' sidebar displays a plan titled 'Claude's plan' with the following steps:

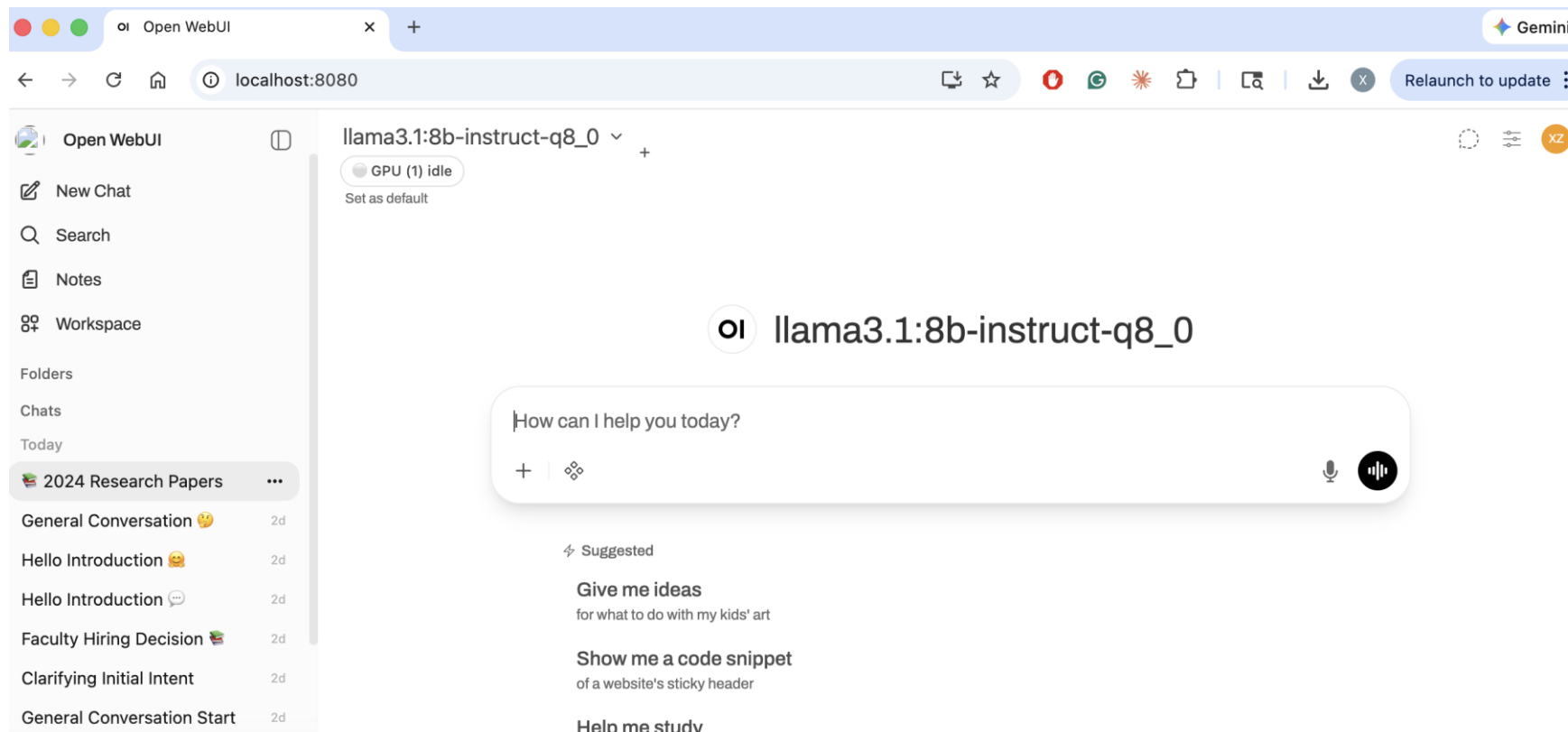
- 1. Navigate to the Junk Email folder in Outlook
- 2. Filter or scan for unread emails received this week
- 3. Review each unread junk email to identify any that may be important
- 4. Report back with a summary of findings

Buttons for 'Approve plan' and 'Make changes' are visible at the bottom of the plan.

Deploying Local LLMs for Privacy

If you have a Mac Studio with 64G~128G memory, then using a local LLM like Qwen3.5 32b in Ollama will be great for many tasks like

1. Turning your PDF paper into a slides in Beamer latex
2. Reading application files, writing recommendation letters, etc.



Case Study: how Prof. Ainsworth (Brown Univ.) used AI for coding

- Francis Wayland Professor of Applied Mathematics, Brown University
- Editor in Chief of SIAM Journal on Numerical Analysis
- Decades of experience in serious numerical analysis and scientific computing

Mark used a local LLM Qwen on a Mac for coding for research, asked ChatGPT for prompts. On April 11, 2026, Mark commented on using AI in Numerical Analysis Day conference panel.

1. “Code I did in the past weeks are more than what I did in the past decade”
 2. “Don’t do openclaw and sit there drinking lemonade, vibe coding”
 3. “prompts are important”
- No way a local LLM can be better than frontier models, unless with excellent prompts.
 - Asking advisor (ChatGPT/Claude/Gemini) for prompts is not enough.
 - (I think) Mark’s critical prompt is his decades of experience on computational math.
 - For students: very likely you won’t use AI as well as Mark and I do in the beginning due to limited knowledge; but keep using AI to learn faster and learn more, you will be better and better later. Goal of a PhD is to surpass advisors anyway.

No need to stress out: you can pick it up quickly

Me scared by AI tools in January 2026



Me after trying to play with AI.



A talk by Michael Brenner on May 9, 2026

He used Gemini 2.5 in Dec 2025.

I'm fairly confident that we can easily reproduce probably at least 80% of it if we use Claude Code in June 2026.

Building a Science Assistant

Michael P. Brenner
Google Research & Harvard University

A Snapshot of Problems

1. Data Analysis of Single Cell RNA sequencing Predict Future
2. **Epidemiological prediction (Center for Disease Control)**
3. Neural Activity Prediction (Modeling and Data Analysis)
4. Geospatial Analysis
5. Numerical methods for applied mathematics
6. Time series prediction
7. Solve nonlinear PDE
8. Solve complex singular boundary value problem
9. Stochastic processes--models of bacterial cell growth
10. Hydrodynamic stability
11. **Cosmology Integral Asymptotics** Theoretical Physics
12. Crop Yield Prediction
13. **Solar Cell Design** Optimization

SIAM Journal Policy: you can be banned if using AI improperly

Improper use of AI in publications can be as serious as plagiarism.



6. Egregious misrepresentations, including those due to use of AI in the writing or analysis, will lead to an investigation and may have consequences including rejection of the submission, limits on future submissions to SIAM publications, notification of the authors' institutions, and addendum to or retraction of an already published article.

AI is just a tool, and no tool can be responsible for us

This was what Opus 4.7 gave me on April 19, 2026.

Opus 4.7 was Anthropic best publicly available model in April.

pointwise nonlinear product, so that
in $|k| \leq 2N/3 < N/2$ and no high
back into the retained low-wavenum

Just like Tesla Full Self-Driving: no tool can be responsible for us

- Tesla FSD (Supervised) is reliable for 99% driving conditions.
- Due to 1% it cannot handle, we have to supervise it all the time.
- It does all the heavy work for road trip.
- Bad idea for student drivers using only FSD.
- Is \$99 per month for FSD worth it?
- No, if using it ONLY for easy/short trips.
- Yes, if using it for heavy road trips.
- Same for \$100 per month for AI tools.



How I have changed from Denial to Acceptance

- Denial: no need to learn or do research because AI can do it.
- Acceptance: Learning is more important than ever. And it is the best time to do research.
- Denial: AI cannot even code up an efficient 3D Poisson solver in my paper unless I tell it.
- Acceptance: I first teach AI, then I get AI tools for serious scientific computing work.
- Denial: I'm way behind on using AI. I need to pay \$20/\$100 to use it.
- Acceptance: it's never too late and I'm not wrong: I started using it on Jan 27, 2026. I gain far more productivity than a few hundred dollars.

For students: avoid abusing AI, do NOT deny AI

Rule #1: You still need to do all the **HOMWORK**.

- ❖ **Do learn coding.** Example: efficient matlab code can be done by EITHER vectorize loops OR wrap up C++ code. If you don't know such knowledge, AI can cheat on you.
- ❖ **Do learn math.** Example: if AI generated some unknown Sobolev inequality, then it is a good opportunity to learn inequalities. If correct, you learn something new. If wrong, you learn something new then correct AI, and your AI is now better than other people's AI.
- ❖ **The goal is to learn, to have progress.** If you used AI to generate something you can NOT understand or easily verify, then you have wasted time.

Rule #2: with or without AI, it is the same for how to do research.

- ❖ **First principle:** solve an easier/smaller problem if you can't solve the original problem.
- ❖ **Second principle:** understand *why* something works, not just *that* it works.

Rule #3: **better tools imply higher expectations on research.** If we can push a button to generate a paper, then that kind of paper is no longer publishable.

NBC News on AI (May 7, 2026)

Yes, AI helps many things.
But humans are still critical.
Because we want to solve harder problems.

Misconception: human V.S. AI.

Reality: whoever using AI well can get a job.

Question: who can use AI well?

My 2 cents: still the students who learned well



Description

Georgia Tech get three hours to build an app using Claude AI

Concluding Remarks

- Myself: I will no longer write code/paper without AI, just like I no longer drive long distance without using FSD on Tesla.
- I ask students to use AI but also learn well. My son wants to drive Tesla when he becomes 16 but he understands the necessity to learn driving a conventional car without FSD.
- My productivity during implementation stage is definitely BOOSTED.
- The verification stage is MORE demanding than before: higher expectation, more to check.
- Most research communities will see exponential growth in publications, if not already.
- Larger volume/quantity does not necessarily imply negativity, but we need to re-think what we do as researchers. The academia research expectations must evolve accordingly.

Take-Home Message: AI Agent Tools. CLI is better for research tasks.

A comparison of the major AI ecosystems



Google

 Chatbot

Gemini

Web / Mobile Chat

 Desktop App

Gemini App

Workspace Extensions

 Terminal CLI

Antigravity CLI

Google DeepMind Agent



Anthropic

 Chatbot

Claude

Web / Mobile Chat

 Desktop App

Claude Mac App

With Computer Use

 Terminal CLI

Claude Code

Terminal Developer Agent



OpenAI

 Chatbot

ChatGPT

Web / Mobile Chat

 Desktop App

Codex Mac App

Advanced Voice & Desktop

 Terminal CLI

Codex

OpenAI Codex Agent

Take-Home Message: What To Do With AI Tools

Embrace AI tools properly — for all three pillars of academia



Learning

For Students

- ▶ Core skills — math, coding, are more important than ever.
- ▶ AI helps us learn faster.
- ▶ If you can't verify or understand AI's output, you've wasted your time.



Teaching

For Faculty

- ▶ Faculty still play an irreplaceable role as mentors.
- ▶ Course content need evolution — **an open challenge for all of us.**
- ▶ Teaching materials can be better prepared with AI assistance.



Research

For Everyone

- ▶ Productivity in is definitely boosted.
- ▶ Better tools imply higher expectations from peers, society, and ourselves.
- ▶ The verification stage is quite demanding, as it should be.