



COLORADO SCHOOL OF
MINES

The postdoc in my terminal

AGENTIC AI ACROSS A REAL RESEARCH WORKFLOW

Valerio Mascolino | October 6, 2026



Before we start: two agents get to work



Agent 1: review a legacy script ~5 min

"This is a legacy heat-sink fin script my students have used for years. Review it like a careful senior engineer... fix the bugs one commit at a time... add tests."

comes back with:

the bugs, explained

one commit per fix

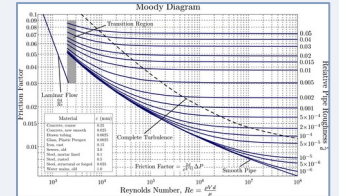
tests + REVIEW.md

```
$ cd ~/service/me-ai-committee/ai-demo/fin-review
$ claude --bg --permission-mode auto \
  -n fin-review "$(cat ../prompts/A.txt)"
```



Agent 2: from a textbook chart to code ~12 min

"Digitize the Moody chart, implement Colebrook-White, Haaland and Swamee-Jain, plan and run your own validation, commit, and write a web report."



comes back with:

digitized data

error metrics

a web report

```
$ cd ~/service/me-ai-committee/ai-demo/moody
$ claude --bg --permission-mode auto \
  -n moody "$(cat ../prompts/B.txt)"
```

They work unattended while I talk. **We check on both in about 15 minutes.**

Four weeks with an AI research partner

 **26**

days

Sep 7 to Oct 2, 2026

 **3**

projects

a paper, an engine, a study

 **66**

sessions

~112 h of active agent work

 **834**

prompts

median length: 90 characters

 **10,579**

actions

files read, code run, edits

 **~233**

commits

186 co-authored by Claude

 **15k+**

lines of code

new, tested Python and C++

 **11**

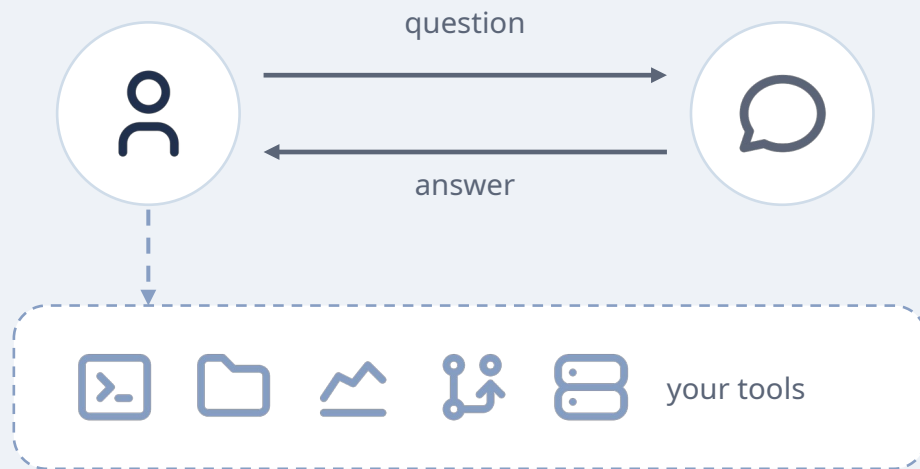
web reports

private, shareable pages

A typical prompt from me: 90 characters. **What came back: everything in this talk.**

A chatbot answers. An agent gets it done.

Chatbot

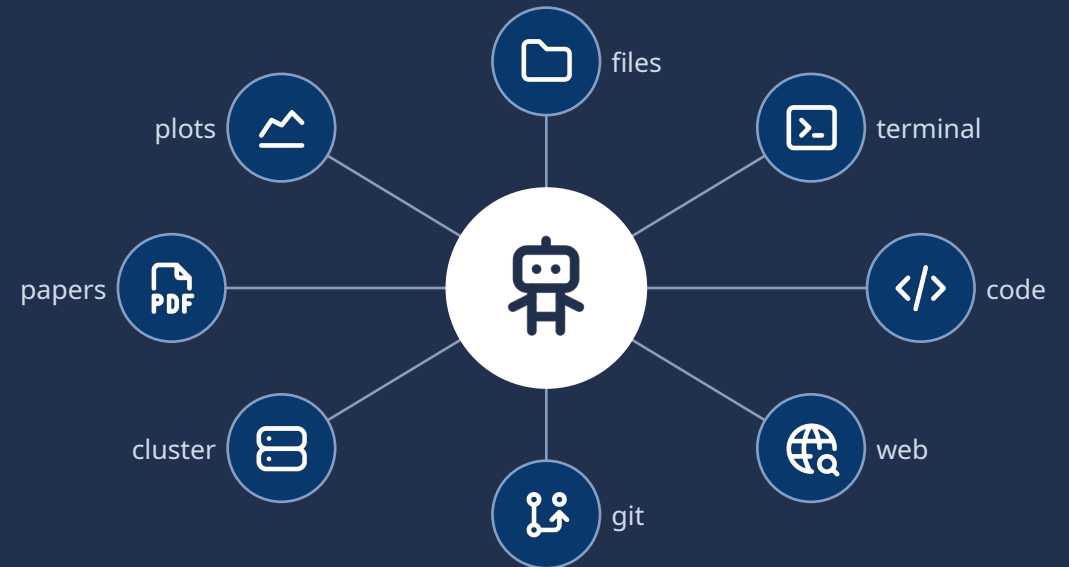


One question, one answer.

You copy, paste, run, debug, and repeat.

vs.

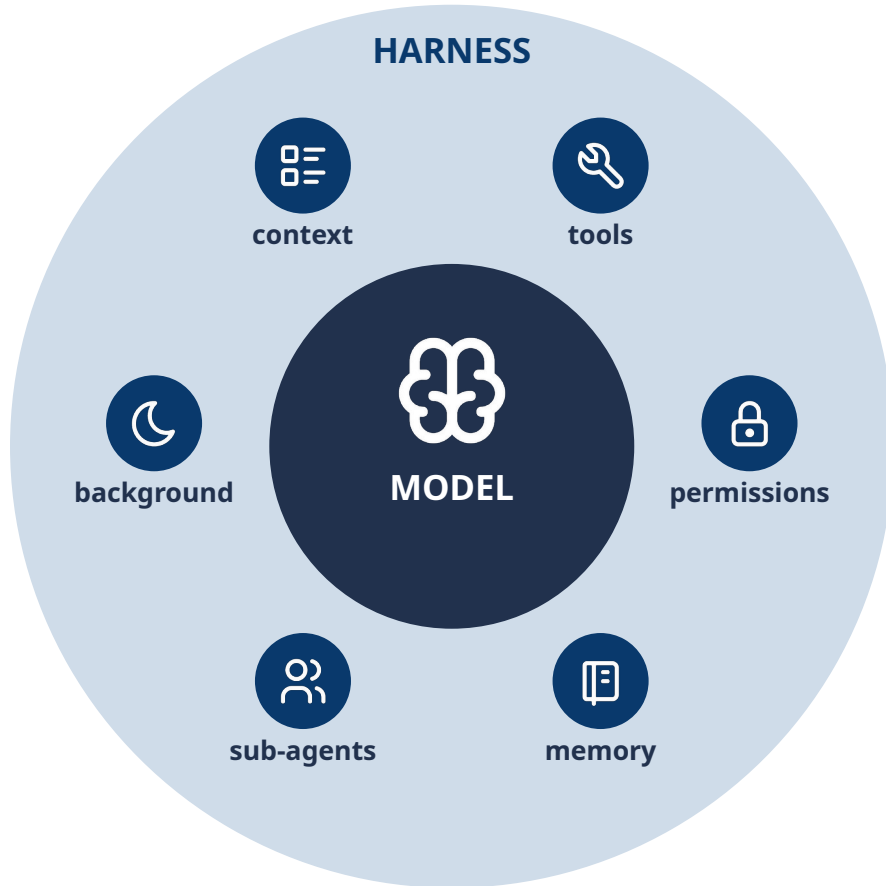
Agent



One goal, hundreds of actions.

It reads, runs, checks, and reports back.

The model thinks; the harness acts



Model

The neural network that reasons and writes. Claude, GPT, Gemini.



Harness

The software that gives it hands: runs code, edits files, asks permission, remembers, spawns helpers. Claude Code, Codex, Antigravity.

Model = a brilliant new postdoc.

Harness = their desk, computer, lab keys and notebook.

One tool, many working styles



How it clicked for me

Prof. Jeff Ackerman shared an example: building a “second brain” after Andrej Karpathy's work, where the agent keeps and organizes your notes.

I didn't adopt it right away.

But it showed me what an agent could do with my files, my tools and my time. It led to the three projects in this talk.



Second brain

it keeps and links your notes and references



Pair programmer

side by side in your editor, step by step



Delegate and walk away

background jobs overnight, between classes

my style



Research manager

it runs helper agents and reviewers for you

my style



Writing partner

drafts, critiques, keeps LaTeX in sync

my style



Teaching assistant

polls, reports, examples for students

How I talk to it: a real kickoff prompt

1 My prompt Oct 1, 15:53, ~330 words

GOAL

"two sensitivity analyses for js1 core 239: start and end of the rod insertion and extraction, and the neutron lifetime"

OPEN QUESTION

"not constant rod speed. Sigmoid maybe? Think about other alternatives and we can discuss"

CONSTRAINTS

"use the C++ engine; alter it if needed and propagate the changes to the python package"

COMPUTE

"slurm, 32 cpus and 64 GB, in parallel; save results in a convenient format"

DELIVERABLE

"a well-documented jupyter notebook that generates the figures"

TALK FIRST

"Try to understand this prompt well and we can chat about it before I leave to work in the background."

2 It plans, then asks 2 min later

- read the engine code first
- flagged a flaw: time step tied to ℓ_{eff} mixes two effects
- 8 rod-speed profiles, with a pick
- cluster sizing and a folder layout

Q1 Move single times, or shift whole movements?

Single times

Q2 Is the +10 s baseline shift OK?

Yes, it is ok

Q3 Are the time and lifetime ranges right?

Yes, they are

Q4 Which speed profiles should I run?

Try all the profiles. This is a big uncertainty.

Q5 Inverse kinetics? Measured data to overlay?

I want it. I have data, but ignore it for now.

Q6 Accept the runtime risk at 32 CPUs?

I accept the runtime risk.

3 I answer, it acts

45 words

...then it acts: engine extended, tests, a PR, 97 cluster runs, report in 2 h

It plans, executes, checks, and polishes

Chatbot

instruction → output. Stops there.

Agent

one request → a loop until the work is done (sensitivity study, Oct 1, 15:53 to 17:50)

 PLAN

Read my request and my 2024 paper, asked me 6 questions, and caught a flaw in my study design before running anything.

 EXECUTE

Added variable rod motion to the C++ engine, ran the tests, opened a pull request, launched 97 cluster runs.

 CHECK

Noticed it was recording the wrong detector signal. Cancelled its own runs, fixed the engine, re-ran 74 tests, relaunched.

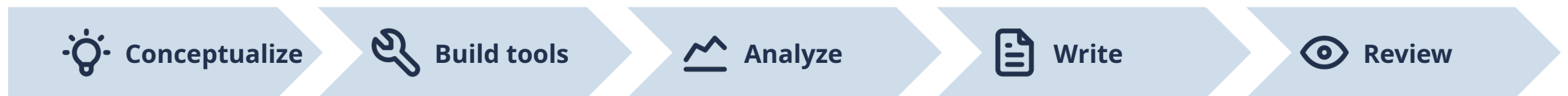
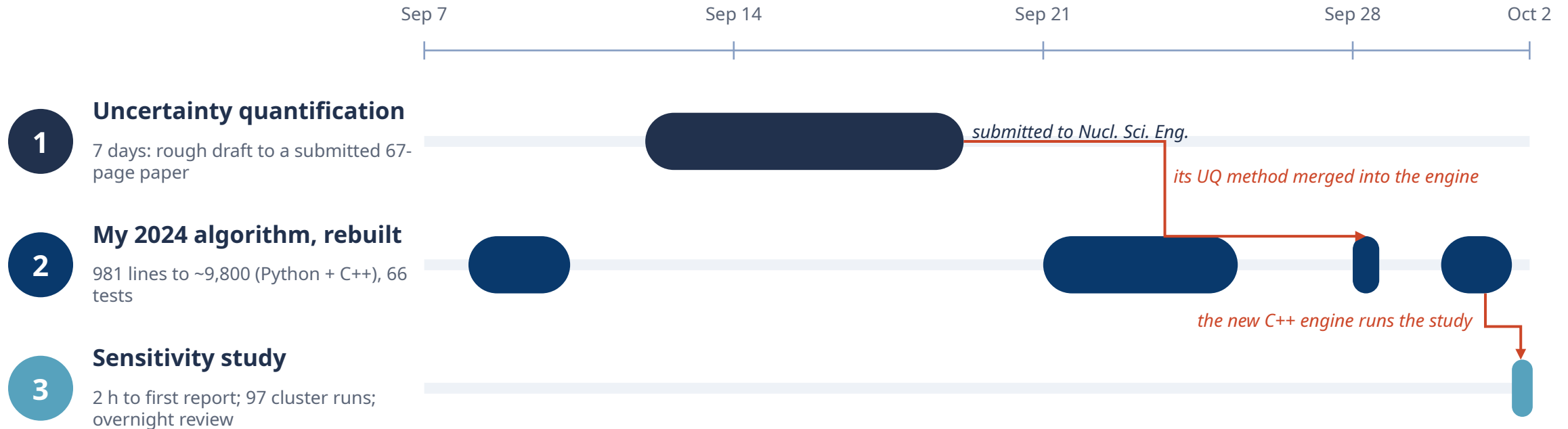
 POLISH

Notebook, 25 figures, a 5,400-word report and a shareable web page, 2 hours after my prompt.

iterate until its own checks say the work is done

What I typed: one 330-word request, then a 45-word answer to its questions.

Three projects, one continuous thread



Every project went through every stage, with the same assistant and one git history.

What I asked, what it did

1 Uncertainty quantification

I ASKED

-  Review my analysis, give feedback
-  Help fix the math and rerun all the cases
-  Carry it into the paper, and keep iterating



IT DID

-  found real bugs in my old analysis
-  turned notebooks into a tested package
-  compared methods, flagged pitfalls
-  looped code, runs and LaTeX together

a submitted paper in 7 days

2 My 2024 algorithm, rebuilt

I ASKED

-  Implement the method from my papers
-  Validate it against my published results
-  Make it fast: write a C++ engine



IT DID

-  turned paper equations into code
-  digitized published figures
-  built a C++ engine, ~20× faster
-  tests, pull requests, web reports

a validated, reusable engine

3 Sensitivity study

I ASKED

-  Perturb the inputs: what really matters?
-  Run it in parallel on the cluster
-  Give me a notebook, figures and a report

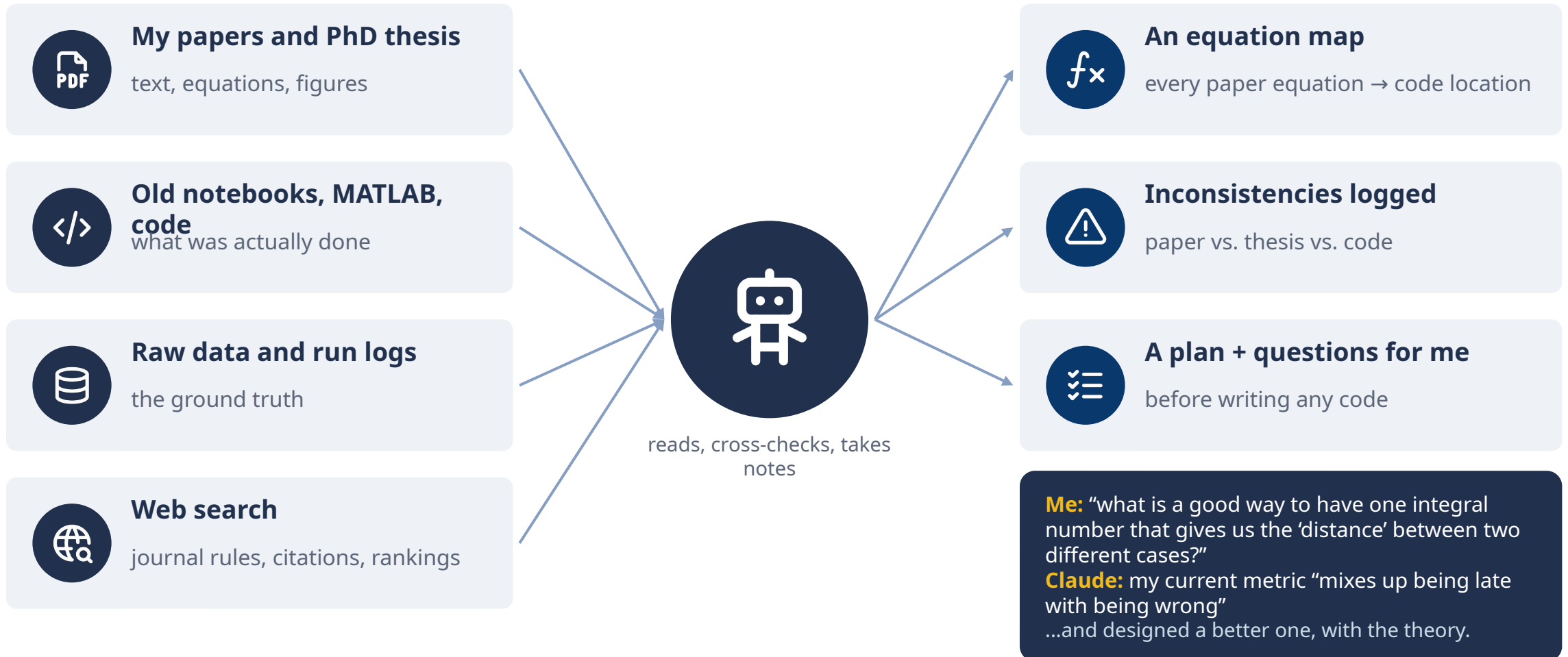


IT DID

-  planned it and asked me questions
-  extended the engine it needed
-  ran 97 cases and checked itself
-  set reviewer agents loose overnight

first full report in 2 hours

It reads first, then helps you think



Equations to code, figures to data

D REACTOR KINETICS

$$S_{tot,i} = \frac{1}{k_{eff}} \sum_{j=1}^{N_{GR}} a_{tot,i,j} S_{tot,j} \quad (33)$$

where $S_{tot,i}$ is the sum of $S_{p,i}$ and $S_{d,i}$, the total number of fission neutrons (both prompt and delayed) generated in cell i by a fission neutron born in cell j . It is trivial to deduce that the total TFM coefficient $a_{tot,i,j}$ must be equal to

$$a_{tot,i,j} = (a_{pp,i,j} + a_{dp,i,j}) \frac{S_{p,j}}{S_{p,j} + S_{d,j}} + (a_{pd,i,j} + a_{dd,i,j}) \frac{S_{d,j}}{S_{p,j} + S_{d,j}} \quad (34)$$

Equation (34) is physically intuitive: the total TFM is obtained by weighing the production of neutrons (both prompt and delayed) by the fraction of the respective source (prompt or delayed) with respect to the total source. Finally, the total fission neutron importance S_i^* , by transposing the coefficients and solving the following equation using the power-iteration technique, is

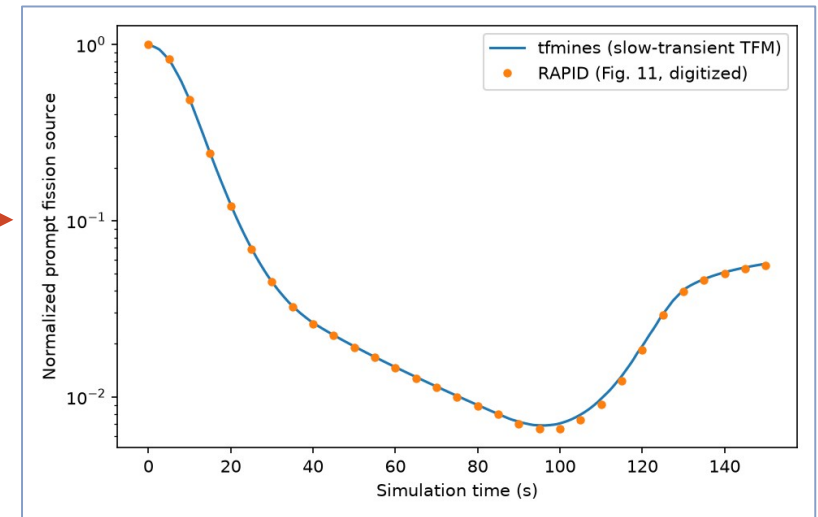
$$S_i^* = \frac{1}{k_{eff}} \sum_{j=1}^{N_{GR}} a_{tot,i,j} S_j^* \quad (35)$$

where S_i is the neutron importance to the fission neutrons born in cell i .

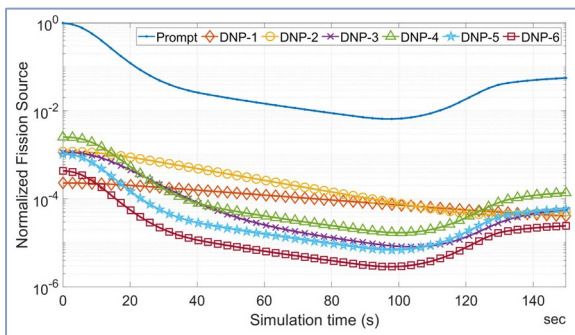
III.B. Calculation of Kinetics Parameters

my 2024 paper: the equations

```
def solve_fast_transient(...):
    """Fast-transient TFM
    solution: full time
    convolution of the TFM
    (Eqs. 38-40)."""
```



new code (line) vs. digitized published result (dots)



a published figure (no data file)

 **Digitized**
colour masking of the image +
axis calibration, with its own
error estimate



Eq. 37 taken literally gives nonsense; it
derived the consistent form



Found a mislabeled entry in Table II of my
own 2024 paper

It checks its work, then calls in reviewers

"Comparing against the old notebook would validate an existing error against itself."

Claude, Sep 12, before testing its new code

So it re-derived every derivative and checked each one numerically.

Result: two bugs in my original derivation, fixed in code and paper.



Caught:

unseeded random estimator

hidden \iffalse block

broken equation reference

physically impossible rod setup

It keeps working while you sleep or teach



Afternoon

15:53 to 17:50

ME

330-word request; 45 words of answers

CLAUDE

plans, extends the engine, tests, opens a PR, launches 97 runs, writes report v1



Evening

18:07 to 21:35

ME

7 short steering messages

CLAUDE

32 more runs, theory sections, cleanup, a new metric: report v7



Night

23:15 to 06:52

ME

"I will be going to bed..."

CLAUDE

4 reviewer agents audit the experimental data; a critical error caught



Morning

until 10:18

ME

open questions waiting for me in a file

CLAUDE

resumes, puts the project under git, pushes to GitHub

```
$ claude --bg "run the study..."
```

start it, close the laptop, go teach



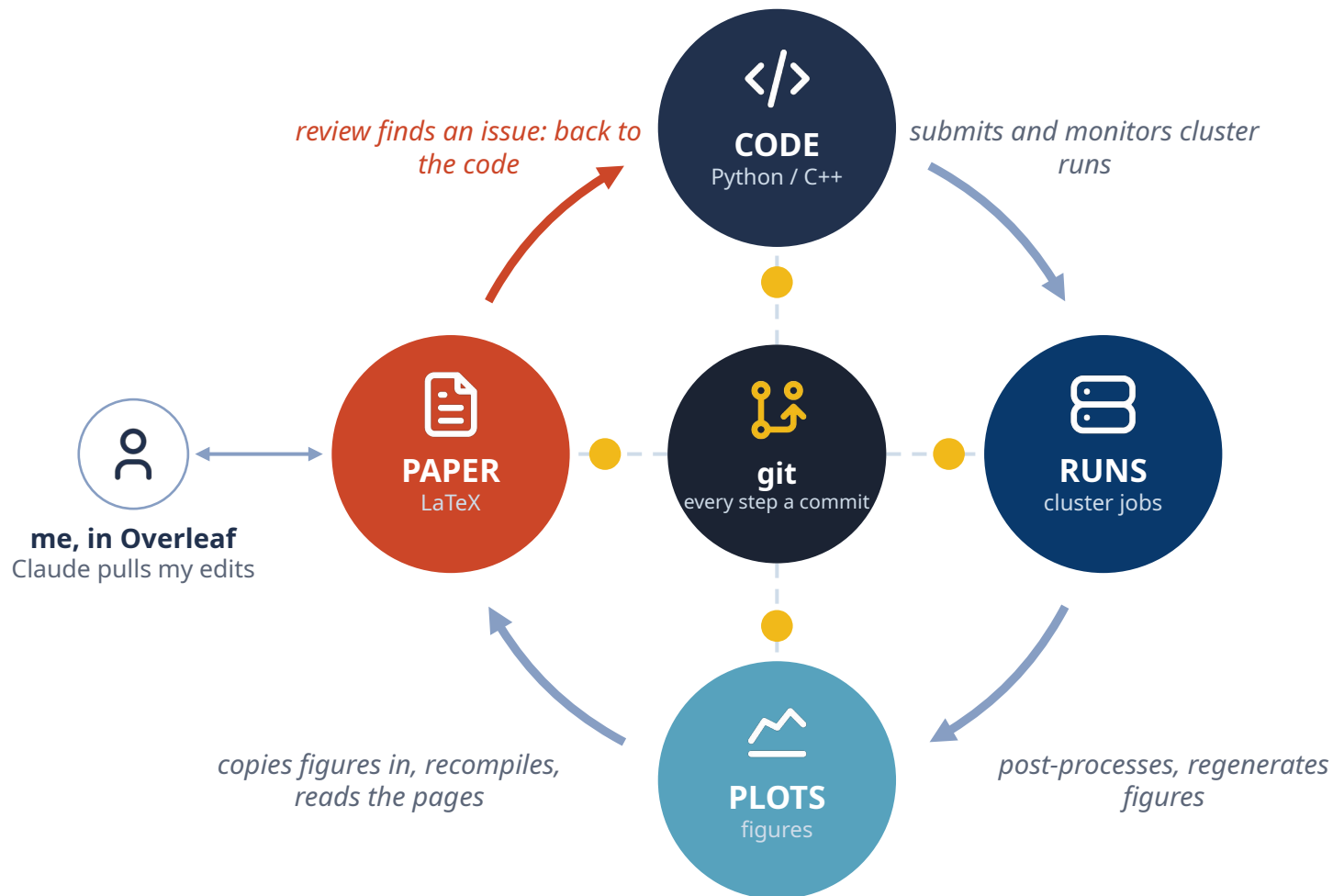
Me: "Any news?"

Claude: "batch 249/300, ~3 h to completion"



Another night: a full reactor model built from a report's figures (27 commits)

Code, runs, plots, paper: one loop in git



One evening around the loop

Sep 12, real commits

20:26	RUNS	cluster job resized (queue said 2027)
21:14	CODE	Jacobian bug fixed
21:56	PAPER	derivation corrected, 42 min later
23:51	PLOTS	new diagnostic plot added

...and the loop keeps turning:

Sep 13	PAPER	I fix a justification in Overleaf
Sep 16	CODE	estimator fixed, random seed added

46 + 81 code + paper commits in 7 days

Writing for itself, for you, for journals



For itself: Markdown



CLAUDE.md + memory

"never use em-dashes"



handoff.md

resume tomorrow, no re-explaining



DECISION_LOG.md

28 decisions, with reasons

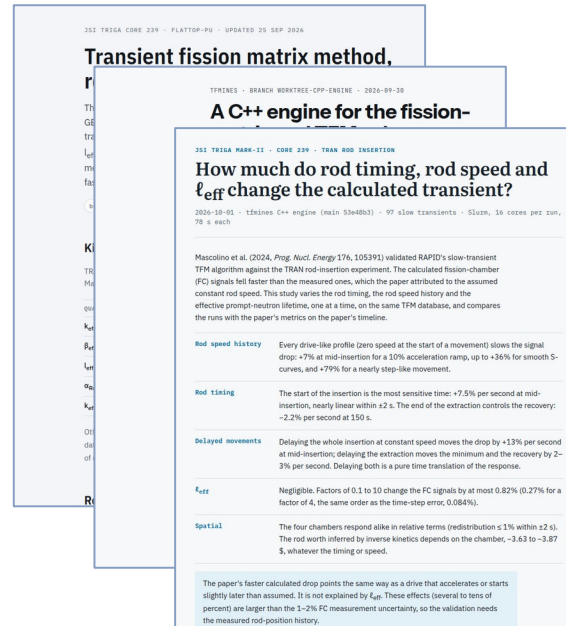


README.md

58 KB of documentation



For you: web pages



11 private, shareable reports
figures, equations, tables; versioned



For the world: papers, docs



67-page journal manuscript

LaTeX, synced with Overleaf



13-page class-poll report

PDF, from one prompt



Emails to collaborators

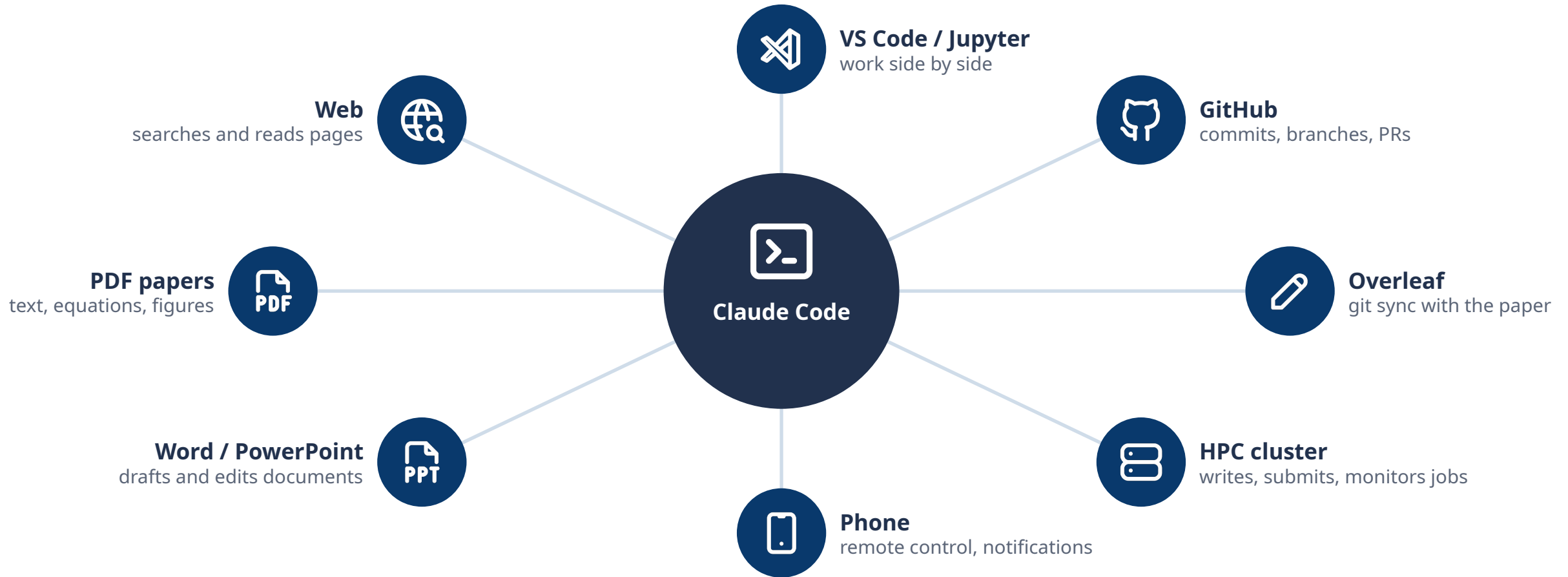
e.g. a summary of 18 commits



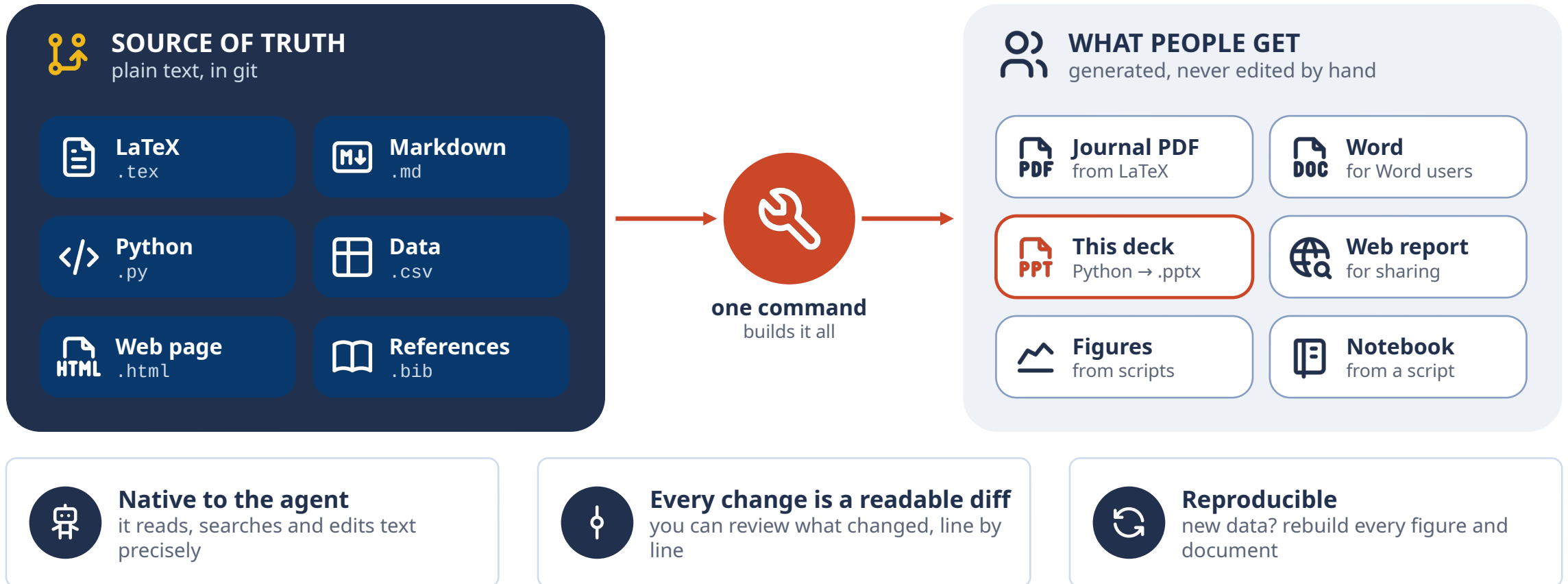
This deck

PowerPoint, on the Mines template

It plugs into the tools you already use



Text in, documents out



Caveats: PDF is fine as input · Word users get an export with tracked changes · big data gets checksums, not git

No more blank page

Daunting starts it removes:

first concept

first algorithm

data extraction

conceptual analysis



"development of a C++ engine for tfmines's number crunching..."



5,200-line parallel C++ engine committed 33 min after I answered its design questions



"what is a good way to have one integral number that gives us the "distance" between two different cases?"



A new metric, and why the old one "mixes up being late with being wrong"



"Resume work based on the handoff you jenerated yesterday."



Picks up exactly where it stopped, from its own notes



"pull the Overleaf. Make sure that my changes make sense."



Reads my edits and flags what doesn't hold up



"Any news?"



"batch 249/300, ~3 h to completion"

real prompts, typos included

A brilliant, overconfident postdoc



It makes mistakes

Wrong physics more than once; I caught it. Review it like a student's work.



It can be a people-pleaser

Changed its journal advice three times as I pushed back. Ask for evidence.



Ask how it knows

It once reported a result it had not re-run. Make it show the output.



Mind data and IP

Check sponsor and export-control rules before it reads anything. Never paste passwords.



Disclose

Journals require it. Our NSE paper names every AI tool used.



Limits and cost

Usage caps are real (mine hit one at 23:41). Lab plans and API exist.

You stay the PI: you ask the question, judge the physics, and sign the paper.

LIVE DEMO

The background of the slide is a scenic photograph of a city at sunset. In the foreground, a large, ornate building with a prominent golden dome and classical columns is visible. The city extends into the middle ground with various buildings and trees showing autumn foliage. In the background, a large, flat-topped mesa or butte rises against a sky filled with soft, orange and yellow clouds from the setting sun.

1 Launched in the background when I started talking:

- review a legacy heat-sink script
- digitize a Moody chart, then code and validate three friction-factor equations

2 Live: one more model

watch it plan, code, validate, and commit

How to start on Monday

1



Get access

Claude for Science: free research-lab accounts for PIs. Codex and Antigravity are alternatives.

2



Install and open a project

```
$ curl -fsSL https://claude.ai/  
install.sh | bash  
$ cd my-project  
$ claude
```

3



Pick one annoying task

A figure to digitize, a script to review, a draft to critique, a cluster study to set up.



Put the project under git first



Ask it to ask you questions



Make it check its own work



Keep a CLAUDE.md of your rules

This deck was made the same way



What I gave it

- a 19-bullet instructions.txt
- 8 answers to its questions
- a pasted email
- short follow-ups: "I don't like the title", "add a loop slide"



What it did

- mined my session logs and git history
- wrote ~1,100 lines of Python that draw every slide
- rendered, inspected and fixed each slide
- ran the demo end to end, three times



What came out

- this 24-slide deck, on the Mines template, with speaker notes
- a demo kit with backups and an answer key
- the handout and the setup guide (web pages)

About 35 minutes to the first version. Every infographic is code, so each revision is one sentence away.



COLORADO SCHOOL OF
MINES

Questions?

→ [MINES.EDU](https://mines.edu)

Valerio Mascolino
Assistant Professor
Mechanical Engineering
Colorado School of Mines