

An opinionated take on coding agents in computational science

Eoin Murray

Agenda

1. **Coding agents** — why now?
2. **Demolab v1** — ready for trial.
3. **Demolab v2** — autoresearch and lab scale.

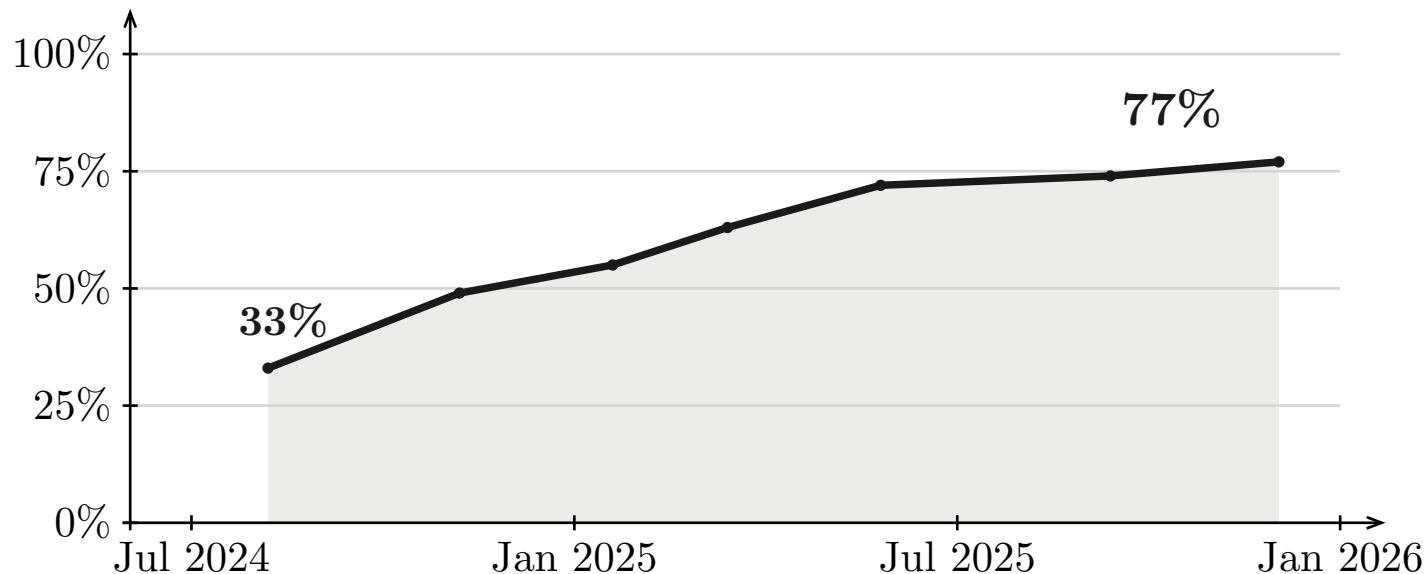
PART ONE

Coding agents

An observation

- **Software engineers** use coding agents like crazy.
- **Scientists**, far less so.

Coding agents are now useful



Share of real, human-validated GitHub issues an agent resolves end-to-end — SWE-bench Verified (swebench.com). Software bugs, not science.

autocomplete (2021) → chat (2023) → **agents that do real work (2025)**

What a coding agent is (demo)

- **Does the work**, not just autocomplete: reads, runs, edits, verifies.
- **Can do anything your terminal can do.**
- Many exist, commercial and open — no vendor lock-in.
- Only recently good enough — still uneven, still supervised.

What an agent can do

- “Port this script to Python; check outputs match.”
- “Read this paper and code up a simulation.”
- “Simulate an LIF neuron.”
- “Draft the methods section from the run log.”
- “Pull yesterday’s runs from the database and plot them.”
- “Run the sweep overnight; write up what you find.”

Strengths of coding agents

- **Do your work faster** — the same tasks, less time.
- **Clear the backlog** — the things you never had time for.
- **Document as they go** — the work writes itself down.
- **Read the literature** — turn a paper straight into runnable code.
- **Work much better with larger contexts** — whole repos, not snippets.

Weaknesses of coding agents

- **Confidently wrong** — fake APIs, numbers, citations.
- **Comprehension debt** — you ship code you don't understand.
- **Weak judgement** — won't kill a bad experiment.
- **Brainrot** — can't agent-code and read deeply the same day.

Powerful tool, needs supervision.

The price of coding agents

Agent	Entry	Heavy use/mo
GitHub Copilot	\$10/mo	\$39–100/mo
Cursor	\$20/mo	\$60–200/mo
Windsurf	\$20/mo	\$200/mo
Claude Code	\$20/mo	\$100–200/mo
Codex (ChatGPT)	\$20/mo	usage tail
Antigravity	\$20/mo	\$100–200/mo
DeepSeek (API)	API key	\$15–90/mo
Local Llama (8B)	\$0	not agent-grade

Bundled agents: seat + usage meter. DeepSeek: pay-per-token. Local Llama: free, assist-tier only.

PART TWO

Demolab v1 — rails for the agent

What Demolab is

- **Demolab writes code, runs programs, and documents results.**
- **A lab notebook for computational science** — the science lives as code.
- **One loop** — run → data → write-up → publish.
- **One build** — the whole repo becomes a website and PDFs.
- **Numbers come from the run** — never retyped, so the page can't drift.
- **Reproducible by default** — provenance stamped on every result.

Demolab is not a replacement for your thinking.

You review *everything*.

Why Demolab

- **Results drift** — retyped numbers, stale figures. Here they come from the run.
- **Reproducibility rots** — a year on, nothing builds. One command rebuilds it all.
- **Code and paper split up** — the experiment and the write-up live in one repo.
- **Agents need rails** — operate in plain language, keep the science yours.

Principles of Demolab

1. **Tools compute, experiments analyse** — clean split.
2. **One experiment** = a runner + a write-up.
3. **Talk through files**, never imports.
4. **Raw data is disposable** — the record is committed.
5. **Bring any stack.**
6. **Provenance built in.**
7. **Documentation is everything.**

The shape of a Demolab repo (demo: pinglab)

<code>tools/</code>	the science – models & solvers
<code>experiments/</code>	the runners – <code>expNNN.py</code> per experiment
<code>writings/</code>	the writeups – one <code>.typ</code> per entry
<code>temp/</code>	the raw run data – gitignored, big
<code>artifacts/</code>	the distilled record – <code>data/</code> + <code>pdfs/</code> (committed)
<code>demolab.yaml</code>	branding + collections config
<code>.demolab/</code> + the	demolab-cli package – the engine, never edited
by hand	

Typst: a modern LaTeX

LaTeX

- Slow, multi-pass compiles.
- Cryptic error messages.
- PDF only.
- Macro soup and package hell.

Typst

- **Hot-reloading.**
- **Readable** errors with line numbers.
- One source → **web and PDF.**
- **Imports files**, no hardcoded numbers.

The same beautiful math: selectable MathML on the web, typeset in the PDF.

Guides

```
my-coding-agent ► HELP█
```

```
GUIDES # always on – the conventions
```

```
RULES HOUSESTYLE SLIDES STRUCTURE GLOSSARY SUPPORT
```

Type **HELP** — Demolab walks you through its own documentation.

Runbooks

RUNBOOKS # on demand — run it, step by step

GETTING-STARTED TOUR LINT DOCTOR RED-TEAM STEELMAN

NEXT MIGRATE-CODE MIGRATE-STACK FROM-JUPYTER FROM-PAPER

EMBED-DOCS GROUND-CLAIMS UPDATE

Agent programs — read papers, convert code, migrate the stack,
update.

Driven by a handful of commands

<code>uvx demolab-cli init</code>	scaffold a lab
<code>uv run python experiments/exp000.py</code>	run one experiment
<code>demolab dev</code>	live-reload the site
<code>demolab build</code>	website + PDFs
<code>demolab test</code>	run tests

Or just operate manually — no agent required.

Use any stack you want

- **Default: Python** — `uv`, `.py` runners, `pytest`.
- **Files, not imports** — tools write `numbers.json`; runners never import tools.
- **Switch by asking** — migrate the stack to **MATLAB**; the agent runs a runbook.
- **MATLAB · R · Julia · Octave** — Typst and the contract stay put.
- **Python optional** — plotting only, or drop it entirely.

GETTING STARTED

demolab.eoinmurray.info

SUPPORT

github.com/eoinmurray/demolab

Open a GitHub issue and I'll fix it quickly.

Need help? Just type **HELP**.

PART THREE

Demolab v2 — the future of Demolab

Autoresearch

- **Give the agent a goal and compute.**
- **It runs the loop** — goal → code → compute → read the data → document → iterate.
- **Every step is recorded** — a committed experiment and write-up per attempt.
- **Runs safely in a pull request** — review the work before it merges.
- **The rails already exist** — provenance, reproducibility, and a published trail come free.

Status: implemented, but untested.

Lab scale

- **v1 is one person** — a single researcher's lab notebook.
- **Next: a lab on a GitHub org** — many users, one shared flow.
- **It compounds** — questions and answers, code review, shared code across the lab.