

Coding Agents at CHPC

Brandon Claridge

Center for High Performance Computing
University of Utah

Overview



Explain

Define what coding agents are, how they differ from chat-only assistants, and explain utility for scientific computing.



Coding agent modules

Which coding agents are available at CHPC, how they are configured for the CHPC, and what is allowed.

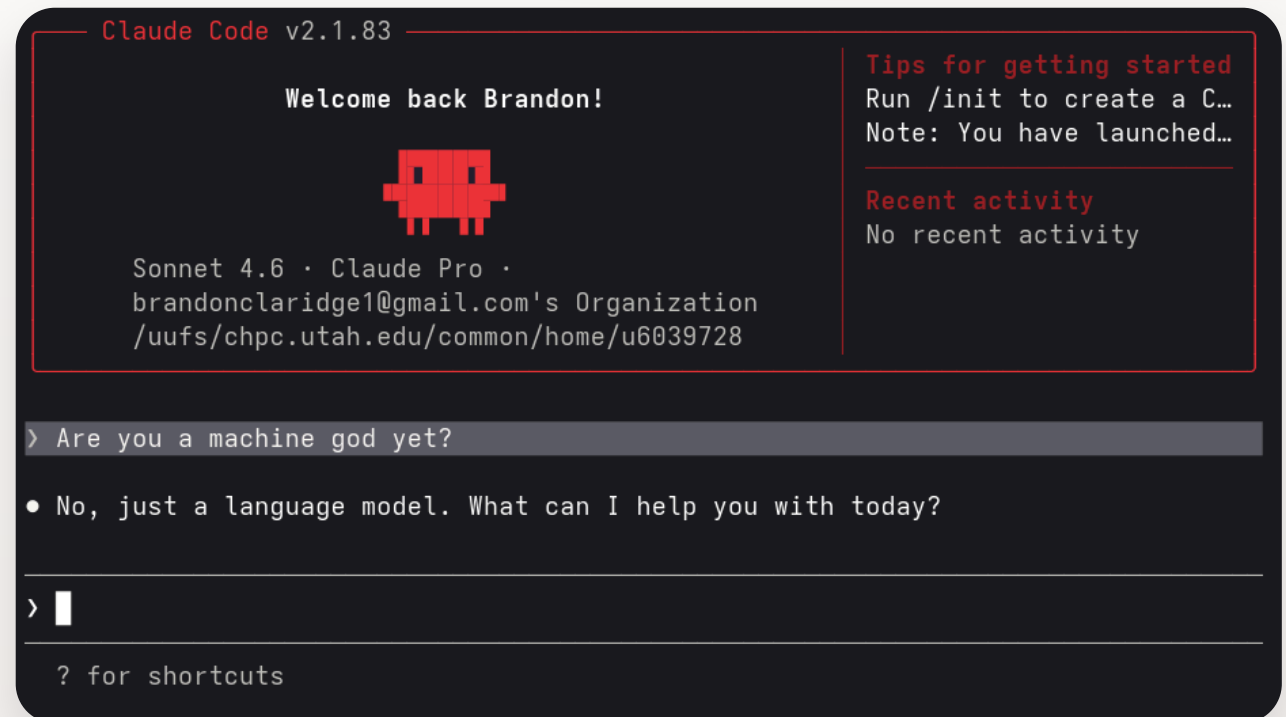


Tutorial and tips

Walk through a quick tutorial and share practical tips for getting started.

What are coding agents?

- An **LLM-based program** that can plan, write, inspect, explain, debug, and run code.
- Natural language prompts agent to edit files, execute shell commands, test output, docs, and feedback from the user.
- Compared with chat-only LLMs, agents are MUCH more powerful because they have direct access to your codebase/ information.



Harness vs. model

Harness

- Claude Code, Codex, OpenCode, Antigravity
- Runs as local process
- Reads and edits your files
- Runs shell commands
- Asks your permission to take actions
- Store, updates the conversation transcript

prompt + history +
tool output

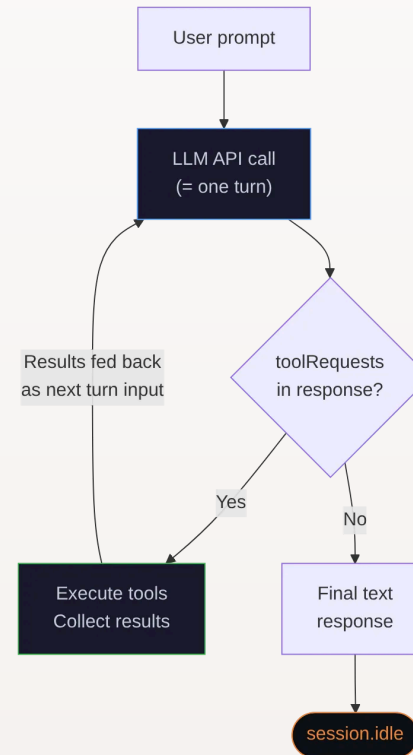


tool call (text)

LLM

- Opus 5.5, GPT-5.6, Qwen 3.8
- Runs on a model server (commercial provider or local server)
- Reads text, predicts text
- Decides which tool to call

Agentic loop



Source: GitHub Docs, “Agent loop” (GitHub Copilot SDK). docs.github.com/en/copilot/how-tos/copilot-sdk/features/agent-loop

Programming languages context

ABSTRACTION

AI & NATURAL LANGUAGE 2010S - NOW

YOU ARE HERE

"Sort my list of numbers."

SCRIPTING & MODERN 2000S

`sorted(numbers)`

OBJECT-ORIENTED 1980S - 90S

`Collections.sort(numbers);`

PROCEDURAL 1960S - 70S

`for (i=0; i<n-1; i++) for (j=0; j<n-1-i; j++) if (a[j]>a[j+1]) swap(a,j);`

ASSEMBLY 1950S

`CMP EAX, EBX    JLE done    XCHG EAX, EBX`

MACHINE CODE 1940S

`00111001 11011000 01111110 00000001 10010011`

Implications for scientific computing

Running HPC jobs

Generate test scripts, explain pipelines, spot syntax issues, and draft/execute SLURM batch scripts.

Software use

Lower the barrier to using complex tools such as workflow managers, containers, and schedulers.

Code adaptation

Translate examples from colleagues, local environments, other clusters, or other languages into local CHPC-friendly patterns.

Faster development

Reduce friction for scripting, documentation, and debugging around scientific workflows.

Running on HPC

- Routine **file/data wrangling** tasks, like reorganizing outputs, renaming batches of files, and reshaping directory structure.
- Troubleshoot **failed jobs**.
- **Monitor logs** from jobs and other processes.
- Ensure **SLURM jobs** run on the best partition/account/qos settings.
- Optimize **compute resource** use so jobs run efficiently without wasting allocation.
- Choose the right **software tools** for a given task.

Faster development

- Describe in detail your project or analysis and have the agent draft a script in a matter of **minutes instead of hours**.
- **Fix bugs** or quickly adjust analysis parameters.
- Generate comments and README file sections for a poorly documented script before **sharing with collaborators**.
- **Better understand** a project by having agent review codebase and asking questions.

Software use

- **No more limitations** with steep learning curves preventing software use!
- Describe your analysis pipeline in plain English and get a working **Snakemake or Nextflow skeleton** with proper SLURM executor configuration.
- Scale a script to **paralellize across multiple nodes** using MPI.

Code adaptation

- Convert a script from **one language to another**.
- Translate a local script into a **cluster-friendly version**.
- **Modernize legacy code** from your lab or collaborators.

Where are coding agents allowed?

ALLOWED

General Environment

Most coding agents are permitted in the GE for normal development, workflow assistance, and routine coding tasks.

NOT ALLOWED

Protected Environment

Coding agents are not currently permitted in the PE at this time. CHPC-hosted models should be available in the near future.

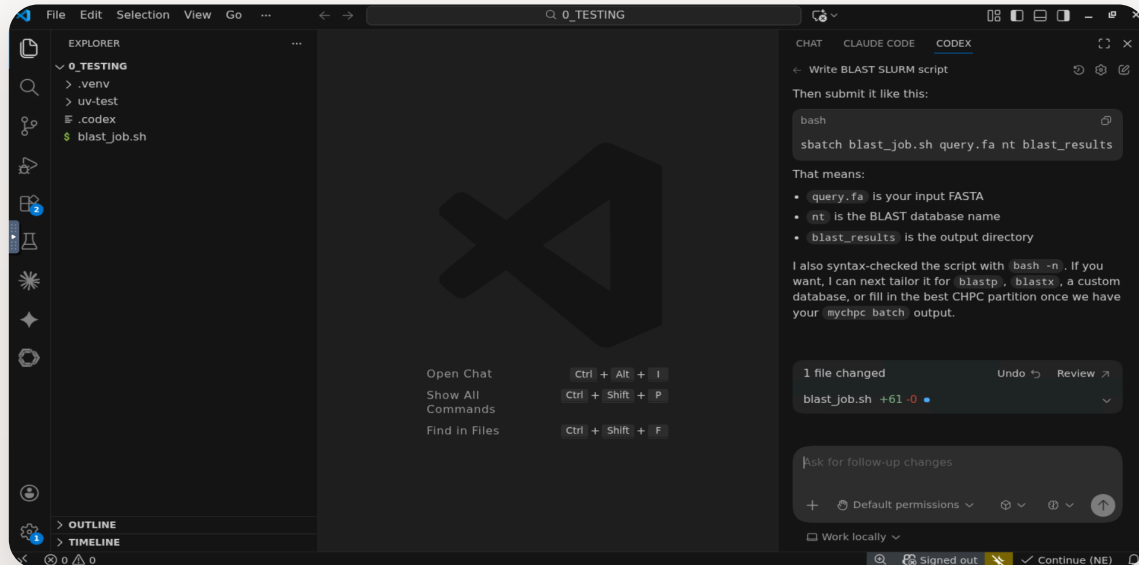
NOT ALLOWED

Citadel / Restricted Environment

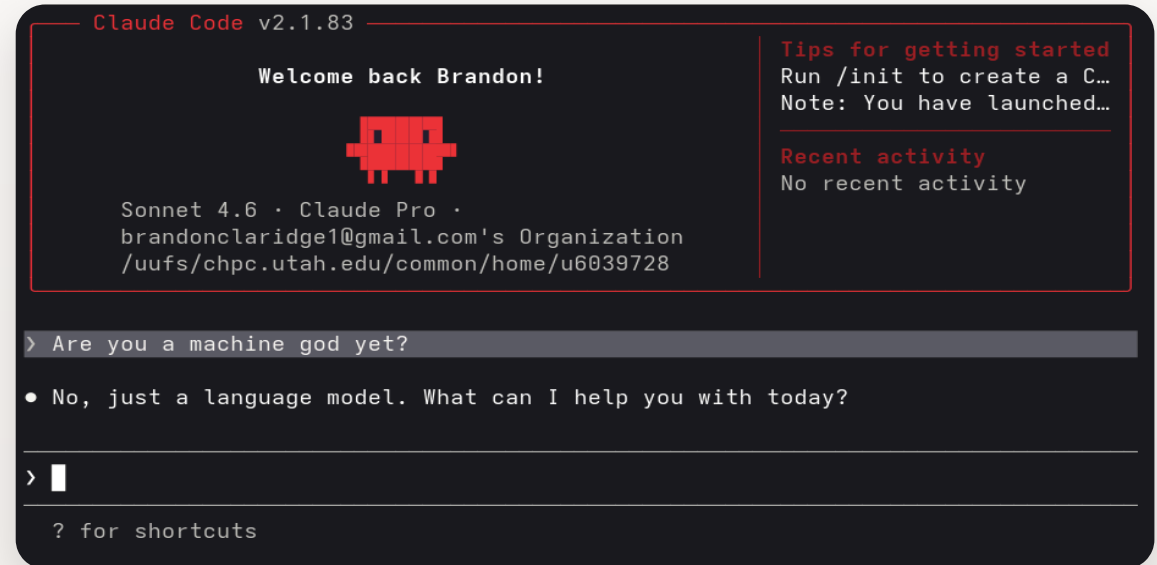
Coding agents are not permitted in Citadel or other restricted environments. CHPC-hosted models should be available in the future.

Coding agents at CHPC

VSCode/ IDE extensions



Terminal User Interfaces (TUIs)



Coding agents at CHPC

VSCoDe/ IDE extensions

- User-installed extensions in Open OnDemand VSCoDe interactive desktop.
- Best for inline assistance, smaller edits.
- Not natively aware of CHPC environment.

Terminal User Interfaces (TUIs)

- CHPC modules available (OpenCode, Codex, Claude Code).
- Useful for those who already work in shells, SSH sessions, and remote development environments.
- Provided with context on working on CHPC.

CHPC coding agent modules

- **Environment-aware system prompt** so the agent knows cluster norms, mounted filesystems, and scheduler expectations.
- **CHPC-specific skills** for Slurm, modules, software stacks, storage, and common scientific workflow patterns.
- **Execution guardrails** so agents can help with commands and jobs without silently doing unsafe things.
- **Documentation grounding** so answers reflect CHPC policy and local best practices rather than generic Linux advice.

TUI modules

Claude Code

```
(base) brandon_work@fedora:~/Documents/0_chpc_presentations/CHPC_AGENTIC_AI$ claude
Claude Code v2.1.92

Welcome back Brandon!

Sonnet 4.8 · Claude Pro
~/Documents/0_chpc_presentations/CHPC_AGENTIC_AI

> |

? for shortcuts                                /buddy
```

Tips for getting started
Run `/init` to create a `CLAUDE.md` file with instructions for Clau...

Recent activity
1w ago That's a little better. The text going verticale isn't ...
1w ago please separate the panels a bit and make the headers o...
/resume for more

Codex

```
🌟 Update available! 0.105.0 -> 0.121.0
Run npm install -g @openai/codex to update.

See full release notes:
https://github.com/openai/codex/releases/latest

> _ OpenAI Codex (v0.105.0)

model: gpt-5.4 medium /model to change
directory: ~/0_chpc_presentations/CHPC_AGENTIC_AI

Tip: Update Required - This version will no longer be supported starting May 8th. Please upgrade to the latest version
(https://github.com/openai/codex/releases/latest) using your preferred package manager.

> 🏃 /run /review on my current changes

gpt-5.4 medium · 100% left · ~/Documents/0_chpc_presentations/CHPC_AGENTIC_AI
```

OpenCode

```
opencode

Ask anything... "Fix a TODO in the codebase"
Build Qwen 3.5 72B CHPC Coder

tab agents ctrl+p commands
```

OpenCode and CHPC-hosted models

Model-agnostic interface

OpenCode is a TUI for coding agents that can connect to both CHPC-hosted and commercial models.

CHPC-hosted options

Current local choices include Qwen 3.8 27B and for users who want an on-cluster model path.

Open-source tradeoff

These models are strong for most day-to-day coding work, but they are less reliable on larger, more complex tasks.

Still maturing

The service is still under development, so expect occasional downtimes.

Codex with CHPC-hosted models

- The **codex_CHPC_local_model module** can also run the same CHPC-hosted **Qwen 3.8 27B** model used in OpenCode.
- Same Codex interface and workflow, but the model runs on the cluster instead of through an OpenAI subscription.
- A good fit if you like the Codex interface but have **run out of tokens** on your plan.
- Same tradeoffs as OpenCode: great for day-to-day work, less reliable on larger, complex tasks.

Codex/ OpenAI subscriptions

- Codex models can be run with the **Codex TUI** and in **OpenCode**.
- Access to Codex is provided by personal subscriptions, including **free and paid** plans, as well as **university subscriptions**.
- Authorization requires browser interactivity from the terminal (ssh -X forwarding, interactive desktop via Open Ondemand or FastX). Device code authorization does not work.

Claude - Antigravity

- **Claude** and **Gemini** commercial models (via subscriptions) are only available through their respective TUIs.
- Access is via **paid subscriptions**.

CHPC system prompt

- An `AGENTS.md` or `CLAUDE.md` file that is added first to the transcript when the TUI launches.
- Provides critical context about the **HPC computing environment**.
- Add your own **project-level** `AGENTS.md` or `CLAUDE.md` to directories.

Skills

- **Skill:** a reusable markdown file of instructions that the agent loads when a task calls for it.
- **Auto-invoked:** triggered by task context or explicitly called (/skills).
- **Consistency:** reduces variability between models
- **CHPC skills:** improve performance, enforce CHPC policy, efficient use of resources.

```
name: job-triage
description: Use when a SLURM job has failed, was cancelled, won't start, or produced unexpected output on CHPC. Diagnoses the root cause from job logs and sacct output - covers OOM kills, walltime exceeded, invalid partition/account, missing modules, shared library errors, NFS path issues, and OOD/RStudio/Jupyter session failures. Distinct from slurm-utilization-audit, which tunes jobs that ran but were inefficient.
---

# CHPC Job Triage

Use this skill to diagnose why a SLURM job failed, was cancelled, or won't start. Always gather evidence first, then work through the diagnostic tree. If the root cause cannot be determined, escalate to the helpdesk.

---

## Step 1: Gather Evidence

Ask for the job ID if it is missing. Do not start diagnosing until you have at least the `sacct` output or the user has pasted an error message.

Before interpreting missing or incomplete Slurm output, verify that you are querying the correct cluster. If `sacct` returns only the header, a job appears to be missing, or `squeue`/`scontrol` cannot find the job, retry with explicit cluster selection first.

```bash
Job state, exit code, and resource usage
sacct -j JOBID --format=JobID,JobName,State,ExitCode,Reason,Elapsed,ReqMem,MaxRSS,AllocCPUS,NodeList -X

If the job may be on a different cluster, query it explicitly
sacct -M CLUSTER -j JOBID --format=JobID,JobName,State,ExitCode,Reason,Elapsed,ReqMem,MaxRSS,AllocCPUS,NodeList -X
squeue -M CLUSTER -j JOBID
scontrol -M CLUSTER show job JOBID

Last 50 lines of job output (adjust filename as needed)
tail -50 slurm-JOBID.out

Full job details - especially useful for pending jobs
scontrol show job JOBID
```

---

## Step 2: Diagnostic Tree

### A. Job never started

#### "Invalid account or account/partition combination"
- Run `mychpc batch` to show all valid partition/account/QOS combinations for this user
- Copy `--partition`, `--account`, and `--qos` values exactly from `mychpc batch` output - do not guess or construct them
- Verify the allocation is active for the current quarter

#### "Node count specification invalid" or `Reason=ReqNodeNotAvail`
- Single-job node limits: ~32 nodes (notchpeak), 24 (kingspeak), 186 (lonepeak)
- Check chpc.utah.edu news page for scheduled maintenance
- Suggest a different partition or waiting out maintenance

### B. Job stuck in PENDING indefinitely
- If the job seems to be missing or not found, rule out a wrong-cluster query first by retrying the Step 1 commands with `--M CLUSTER`
- Check `scontrol show job JOBID` → look at the `Reason` field
```

Example skill: job-triage.md

CHPC skills



job-triage

Helps diagnose why a SLURM job failed, stalled, or produced unexpected output.



partition-selection

Guides users to the best partition, account, and QOS choices for a SLURM job.



slurm-utilization-audit

Reviews completed jobs for CPU or GPU efficiency and suggests concrete tuning changes.



software-environments

Helps load modules and build Python, conda, or container-based environments on CHPC.



storage-placement

Advises where inputs and outputs should live based on use case, size, I/O pattern, and retention.

Demonstration

- Modules and Codex authentication: available coding agent TUIs, and University subscription authorization
- Job troubleshooting: find and resolve bugs with an existing batch script
- Simple bioinformatics example: align `12S_rRNA` sequences for 8 species and infer an evolutionary tree

Tips

- **Plan mode** - discussion with agent to provide context, clarify goals, set parameters, etc.
 - Conserve tokens by planning with a chat-based LLM that outputs a markdown file.
- **Build/execute mode** - allows agent to edit code and call tools.
- `/effort` increasing effort practically means more tokens (~=words) generated for internal reasoning and chain of thought and more tool calls
- **Beware of context rot!!** - saturating model context with too much information will result in more hallucinations and other degradation.
- `/context` or `/status` - see context stats
- `/compact` - compacts the context to reduce context rot.

Tips

- `/connect` - connect to a commercial provider in OpenCode.
- `/model` - change model.
- `@<path to file/dir>` - specify files (with autocomplete).
- `! bash` - execute verbatim bash commands inside the TUI.
- `/init` - initialize project, create summary markdown file.
- Direct agents to produce NOTES.md summaries of the project when ending a session.
- `/resume` resume a conversation from the same directory.

Security

Inherited permissions

Coding agents inherit the basic Linux permissions of the user who launches them.

Destructive commands denied

CHPC modules deny some potentially destructive commands (e.g., `rm`).

Set your own permissions

Users can set additional permissions with `/permissions`.

No network or external MCPs

Network access and external MCP servers are denied due to the potential for prompt injection.

Where do coding agents fall short?

Scientific software is underrepresented

Model quality is often stronger in mainstream software development than in niche research stacks.

Context saturation hurts performance

Long sessions can accumulate stale assumptions, contradictory state, and missed details.

Execution does not equal understanding

An agent may produce runnable commands while still misunderstanding scientific intent.

Permissions matter

On shared systems, autonomy has to be constrained by policy, auditability, and resource controls.

Active development

What CHPC is actively building next for coding-agent workflows.

Protected Environment support

Likely release of an OpenCode module with CHPC-hosted models in the Protected Environment.

Stronger CHPC skills

Increasing capabilities of CHPC models with more skills and additional training on CHPC documentation.

Open-source model rollout

Continued deployment of leading open-source models.

More training coverage

More focus in training presentations on agentic coding, including fundamentals and practical use.

Questions & Discussion

How have you been using coding agents?

WINS

Where has it helped?

GAPS

Where has it fallen short?

QUESTIONS?