



How intelligent is Artificial Intelligence?

**A summer converting code to modern
languages and architectures using LLMs**

Mentor: Kate Thayer-Calder

AUGUST 6, 2025

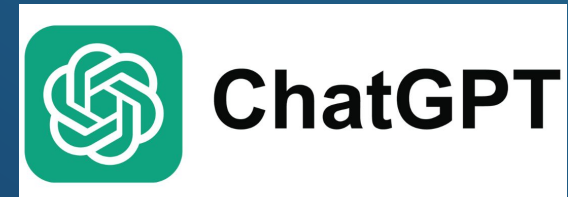
What was the goal of this project?

- Test the ability of a LLM to convert old code from shell to Python
- Inform software engineers at NCAR (and everyone else) of the possibilities that LLMs create for development
- Stay mindful of environmental impact



What are LLMs? How can they help write code?

- LLMs are large language models
- Understand and imitate language styles
- Generate text
- Different models are capable of
 - Answering questions during programming and debugging
 - In-line predictions
 - Agentic coding



What is the topography updating script?

- A script run when Community Earth System Model (CESM) restarts
 - Scripts can only run for so long at a time on Derecho
- Its purpose is to update topography when these restarts occur
- Transferring topography data between different files



How does Claude work?

- In-browser generative chatbot that responds to written prompts
- Can generate text in chat or as a document in various types (e.g., .txt, .py)

TS Pretend you are really good at converting code from one language to another. Walk me through how you would approach a conversion.

Here's my systematic approach to code conversion, developed through handling complex multi-language projects:

Phase 1: Deep Analysis & Mapping

First, I become a detective. I don't just read the code—I *understand* it. I trace through the execution flow, identify the core algorithms, and map out data dependencies. For Fortran-to-Python specifically, I'm looking for:

- **Computational kernels:** Where's the real work happening? These often become NumPy operations.
- **Data structures:** How is information organized? Fortran arrays become NumPy arrays, but the indexing philosophy changes.
- **Control patterns:** Fortran's structured approach vs Python's more flexible paradigms.

TS How could I test the accuracy of the conversion/compare results from the original shell script?

Here's a comprehensive testing strategy to validate the accuracy of the Python conversion against the original Bash script:

Testing and Validation Strategy for Python Conversion
Document

Testing and Validation
Strategy for Python
Conversion
Overview
This comprehensive testing
strategy ensures the Python
conversion accurately
replicates the original
Bash script's behavior.

Interacting with Claude

There are two good strategies I found and tested out to get Claude to really do what you want it to do

- Ask Claude to prompt you with questions

TS I want Claude to convert a codebase from Fortran to Python. Ask me a series of yes/no questions that would facilitate this process for Claude and clear up ambiguities.

- Give Claude a role to take on

TS Pretend you are a very experienced web developer who is extremely thorough and cognizant. The project you are working on (bin_to_cube) is extremely important and any errors could cause immense environmental pollution. Assuming this role, your job is to create detailed unittests for the provided Fortran code to test its functionality.

Working on the conversion

- Using the role-playing strategy, asked Claude to create a plan of a conversion strategy
 - Target python architecture, different error classes
- Asked Claude to follow this plan and carry out the conversion

postrun_CAM_to
po_regen.sh
502 lines

SH

Bash to Python
Conversion
Plan.txt
327 lines

TXT

TS Now, still pretending you are the skilled software engineer mentioned above (with expertise in converting scripts from one programming language to another; who knows how to take into account all the nuances of each language and make sure that no functionality or precision or components are lost during the conversion process), follow the plan attached very closely to convert the attached postrun_CAM_topo_regen.sh to python.

Working on the script

- Asked Claude to create a suite of unit tests to test the output of the conversion against the original
 - There was a lot of back and forth as I asked for revisions
 - In the end, the original test suite worked the best
 - With too much kneading of the output, it can go off the rails...
- Made two minor manual changes to script, then it ran
 - Changing one directory
 - Removing testing parameters



Testing Output

In the end, 33 out of 51 of the tests Claude created passed. Not bad!

```
===== test session starts =====
platform linux -- Python 3.12.8, pytest-8.4.1, pluggy-1.6.0 -- /glade/u/apps/opt/conda/envs/npl-2025a/bin/python
cachedir: .pytest_cache
rootdir: /glade/work/tsegal/test_topoupdater_data/dynamic_atm_topog
plugins: anyio-4.8.0
collected 51 items

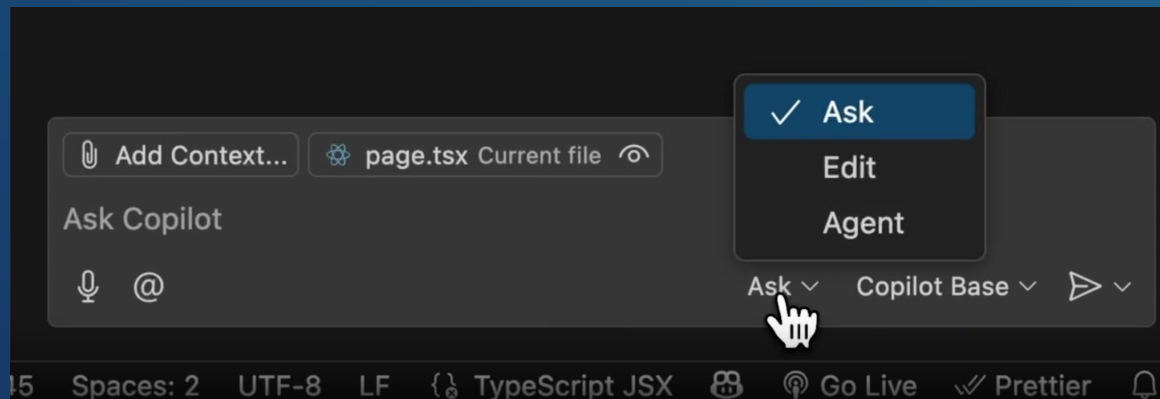
postrun_unit_test_suite.py::TestTopoConfig::test_config_initialization PASSED [ 1%]
postrun_unit_test_suite.py::TestTopoConfig::test_config_defaults PASSED [ 3%]
postrun_unit_test_suite.py::TestTopoConfig::test_config_path_conversion PASSED [ 5%]
postrun_unit_test_suite.py::TestFileManager::test_find_latest_restart_file PASSED [ 7%]
postrun_unit_test_suite.py::TestFileManager::test_find_latest_restart_file_no_files PASSED [ 9%]
postrun_unit_test_suite.py::TestFileManager::test_create_symlink_safe PASSED [ 11%]
postrun_unit_test_suite.py::TestFileManager::test_create_symlink_safe_overwrite PASSED [ 13%]
postrun_unit_test_suite.py::TestFileManager::test_check_netcdf_dimensions PASSED [ 15%]
postrun_unit_test_suite.py::TestFileManager::test_check_netcdf_dimensions_invalid_file PASSED [ 17%]
postrun_unit_test_suite.py::TestFileManager::test_setup_production_paths PASSED [ 19%]
postrun_unit_test_suite.py::TestFileManager::test_setup_test_paths PASSED [ 21%]
postrun_unit_test_suite.py::TestFileManager::test_validate_file_existence PASSED [ 23%]
postrun_unit_test_suite.py::TestFileManager::test_validate_file_existence_missing_file PASSED [ 25%]
postrun_unit_test_suite.py::TestProcessExecutor::test_run_command_success PASSED [ 27%]
postrun_unit_test_suite.py::TestProcessExecutor::test_run_command_failure PASSED [ 29%]
postrun_unit_test_suite.py::TestProcessExecutor::test_run_command_no_check PASSED [ 31%]
postrun_unit_test_suite.py::TestProcessExecutor::test_run_ncl_script PASSED [ 33%]
postrun_unit_test_suite.py::TestProcessExecutor::test_run_fortran_executable PASSED [ 35%]
postrun_unit_test_suite.py::TestGridConfigManager::test_detect_cism_5km_grid FAILED [ 37%]
postrun_unit_test_suite.py::TestGridConfigManager::test_detect_cism_4km_grid FAILED [ 39%]
postrun_unit_test_suite.py::TestGridConfigManager::test_detect_cism_incompatible_grid FAILED [ 41%]
postrun_unit_test_suite.py::TestGridConfigManager::test_detect_cam_fv1_grid FAILED [ 43%]
postrun_unit_test_suite.py::TestGridConfigManager::test_detect_cam_fv2_grid FAILED [ 45%]
postrun_unit_test_suite.py::TestRegridStage::test_regrid_stage_setup FAILED [ 47%]
postrun_unit_test_suite.py::TestRegridStage::test_regrid_stage_preprocess PASSED [ 49%]
postrun_unit_test_suite.py::TestBinToCubeStage::test_generate_namelist PASSED [ 50%]
postrun_unit_test_suite.py::TestCubeToTargetStage::test_base_namelist_generation_fv1 FAILED [ 52%]
postrun_unit_test_suite.py::TestCubeToTargetStage::test_base_namelist_generation_fv2 FAILED [ 54%]
postrun_unit_test_suite.py::TestCubeToTargetStage::test_modify_namelist_parameter PASSED [ 56%]
postrun_unit_test_suite.py::TestPostProcessStage::test_setup_output_dataset PASSED [ 58%]
postrun_unit_test_suite.py::TestPostProcessStage::test_get_cube_to_target_outputs_fv1 PASSED [ 60%]
postrun_unit_test_suite.py::TestPostProcessStage::test_get_cube_to_target_outputs_fv2 PASSED [ 62%]
postrun_unit_test_suite.py::TestTopographyProcessorIntegration::test_processor_initialization FAILED [ 64%]
postrun_unit_test_suite.py::TestTopographyProcessorIntegration::test_setup_and_validate_success FAILED [ 66%]
postrun_unit_test_suite.py::TestTopographyProcessorIntegration::test_setup_validation_failure FAILED [ 68%]
postrun_unit_test_suite.py::TestNetCDFComparison::test_identical_files PASSED [ 70%]
postrun_unit_test_suite.py::TestNetCDFComparison::test_different_dimensions PASSED [ 72%]
postrun_unit_test_suite.py::TestNetCDFComparison::test_different_variables PASSED [ 74%]
postrun_unit_test_suite.py::TestNetCDFComparison::test_numerical_differences FAILED [ 76%]
postrun_unit_test_suite.py::TestNetCDFComparison::test_within_tolerance FAILED [ 78%]
postrun_unit_test_suite.py::TestPerformance::test_file_operations_performance FAILED [ 80%]
postrun_unit_test_suite.py::TestPerformance::test_netcdf_comparison_performance PASSED [ 82%]
postrun_unit_test_suite.py::TestErrorHandling::test_validation_error_inheritance FAILED [ 84%]
postrun_unit_test_suite.py::TestErrorHandling::test_processing_stage_error_inheritance FAILED [ 86%]
postrun_unit_test_suite.py::TestErrorHandling::test_environment_error_inheritance FAILED [ 88%]
postrun_unit_test_suite.py::TestErrorHandling::test_file_manager_error_scenarios PASSED [ 90%]
postrun_unit_test_suite.py::TestErrorHandling::test_process_executor_error_scenarios FAILED [ 92%]
postrun_unit_test_suite.py::TestConfiguration::test_pytest_configuration PASSED [ 94%]
postrun_unit_test_suite.py::TestConfiguration::test_required_imports PASSED [ 96%]
postrun_unit_test_suite.py::TestBashComparison::test_mock_bash_comparison PASSED [ 98%]
postrun_unit_test_suite.py::TestBashComparison::test_acceptable_differences PASSED [100%]
```



Next steps

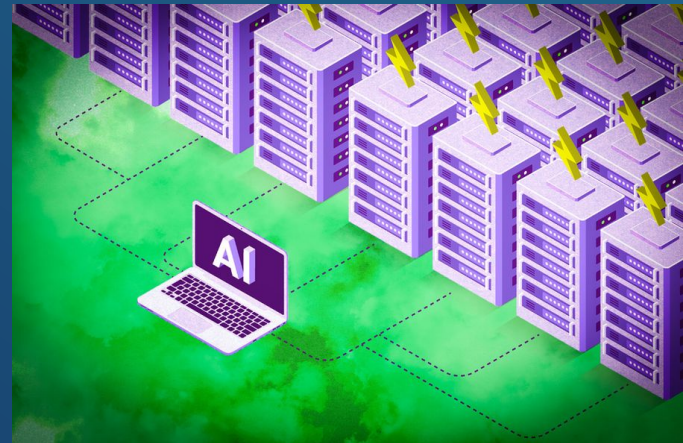
Since the summer is over, I think these would be good next steps to continue this project:

- Deeper review of the final script Claude produced and the output from testing
- Trying out other LLMs, like GitHub Copilot, which have different features



Notes on environmental impact

- Unsurprisingly, LLMs use a lot of energy (and water!)
- Manufacturing hardware, training, queries, cooling machinery down
- Hard to know exactly how much energy they use, but estimates show it is significant
- Benefits can outweigh the negatives, but we should always be aware and proactive.



References

Harvard University Information Technology. “Getting started with prompts for text-based Generative AI tools.” 30 August 2023,

<https://www.huit.harvard.edu/news/ai-prompts>

Visual Studio Code. “Essential AI prompts for developers.” *Youtube*, 17 July 2024, <https://www.youtube.com/watch?v=H3M95i4iS5c&t=168s>

Microsoft Developer. “GitHub Copilot for Skeptics Who Still Think AI is Overrated | BRK124.” *Youtube*, 22 May 2025,

<https://www.youtube.com/watch?v=anl4Hr-ag80&t=393s>



References

Shah, Simmone. “Some AI Prompts Can Cause 50 Times More CO2 Emissions Than Others.” Time, 2 July 2025,

<https://time.com/7295844/climate-emissions-impact-ai-prompts/>

Zhao, Celina. “How much energy does your AI prompt use? It depends.” Science News, 2 July 2025,

<https://www.sciencenews.org/article/ai-energy-carbon-emissions-chatgpt.>

Zewe, Adam. “Explained: Generative AI’s environmental impact.” MIT News, 17 January 2025,

<https://news.mit.edu/2025/explained-generative-ai-environmental-impact-0>





TAMARA SEGAL
CI COMPASS, INTERN

Thank you all for tuning in!

**Thank you to Kate and Brian from NCAR,
the folks at CI Compass, and Professor
Matthew Lamoureux from UConn, and the
NSF (through U.S. National Science
Foundation, Grant #2127548)
who made this summer possible!**

JULY 24, 2025