Objectives

- To explore the capabilities of AI-Based tools across different facets of scientific software development, encompassing areas like code generation, documentation support, code translation, and debugging.
- To identify the strength and limitations of using AI-Based tools for real-world context-based applications that can be easily integrated into existing frameworks of code.

Results

- ChatGPT was able to assist in low-level conversion of the example GOES-16 ABI spectral bands processing module. The chatbot was able to derive information about the input code on its own, understand at least in part what the code intends to do, and provide feedback, test cases, and improvements to the code if asked to do so. ChatGPT’s abilities include efficient python library usage and transition, code optimization, detailed annotations, and context-aware code segmentation.
- In evaluating AI-Based tools for scientific software development, distinct strengths were observed: Bard excelled in preserving code integrity during conversion, while ChatGPT showcased proficiency in code optimization and generation. For code documentation within VSCode, GitHub Copilot emerged as the superior tool due to its predictive capabilities, albeit lacking an interactive chatbot feature.
- These findings emphasize the potential of AI tools in streamlining and enhancing the software development process.