LEGION INTELLIGENCE

CASE STUDY

Accelerating Research Validation and Modeling with AI

(DEPARTMENT OF ENERGY NATIONAL LABORATORIES)

AI-Assisted Code Generation

Industry: Federal / Department of Energy

Use Case: accelerate the generation of implementation logic and research validation

Capability: AI-assisted code generation, explanation, commenting and analysis

Core Problem: excessive time required to model and analyze physics problems in real-world

Deployment: CUI/Sensitive Proprietary Data

ABOUT

U. S. Department of Energy National Laboratories scientific and technical teams need a faster way to turn early research findings into working MATLAB code to either confirm or counter their existing research biases. After identifying a relevant paper, hypothesis, or technical approach, researchers still need to interpret the underlying assumptions, define the right variables, and build code to test whether the idea holds up in realistic scenarios.

The goal is not simply to generate code faster; it is to reduce the time required to move from initial discovery to a working analytical model that researchers could inspect, modify, and validate.

CHALLENGE

Scientists were limited in how quickly they could model fundamental physics problems in real-world scenarios. This work is not just code writing. It requires understanding the underlying physics, selecting variables, translating assumptions into analysis logic, and checking whether results align with known behavior, published specifications, or real-world constraints.

The core bottlenecks were:

- **Limited throughput:** Deep analysis ran against only one assessment every few weeks.
- **Heavy manual synthesis** before code-based validation could begin.
- **Slow implementation logic:** Each coding project typically took a week.
- **Workflow disruption:** AI was needed inside existing tools to reduce learning burden and workflow inefficiencies.
- **Bias validation:** Assumptions needed checking against prior research.

95%

Reduction in development time

>6 hours

6.5+ hours saved per coding task

LEGION INTELLIGENCE

CASE STUDY

SOLUTION

Legion gives scientists governed models to generate code, validate findings, and compare research claims. With Legion, scientists can:

- **Reuse**: Generated “elastic” code that is easy for scientists to adapt and reuse across other modeling or analysis paths.
- **Validate**: Generate code that helps scientists test whether their findings are supported, contradicted, or incomplete.
- **Compare**: Evaluate publicly available information across assessments to identify differences, gaps, and claims requiring deeper review.
- **Generate**: Produce implementation logic and documented code to support technical analysis.
- **Explain**: AI-generated code comments and reasoning show why specific methods were selected and tune inputs.
- **Integrate**: Enable scientists to stay in their normal development environment through an API interface or work in the Legion platform.

By running governed models inside the scientist’s environment, Legion minimizes disruption while delivering the same AI-assisted analysis inside the tools scientists already use.

RESULTS

Leion reduced implementation time and created a path to scale topic-focused research validation.

- **95% faster implementation from concept**: Scientists can move from an initial program concept to generated implementation logic almost immediately, reducing process from one week to a half day.
- **6.5+ hours saved per coding task**: Legion reduced manual development, commenting, troubleshooting, and reasoning time.
- **Documented code generated**: The user received functional code with inline comments, rationale, and guidance on variables and assumptions.
- **Higher research throughput**: Instead of completing one deep validation cycle every couple of weeks, scientists can move toward evaluating multiple topics, claims, or technical assumptions per week.
- **Stronger claim validation**: Scientists can test claims found in proposals, journal articles, technical talks, or other research materials against known results, published specifications, and real-world constraints.
- **Lower friction**: Native platform access and IDE-based API integration allow users to keep working inside their preferred environment.

“It did not just write the code. It explained the reasoning, commented the implementation, and helped me understand why that approach made sense”

- Research Scientist, U.S. Department of Energy