Going LLM Forward with RAS-Commander

```
$ pip install
ras-commander
```

Automate • Analyze • Visualize

CT Association of
Floodplain Managers
2025 Conference
November 19, 2025

CLB
ENGINEERING
CORPORATION

LLM Forward Approach

AI FIRST ENGINEERING, REDEFINED

LICENSED PROFESSIONALS IN RESPONSIBLE CHARGE

DELIVERING DETERMINISTIC, CODE-BASED SOLUTIONS

LLMS ARE PUT FORWARD WHERE THEY PROVIDE THE MOST IMPACT

REVISITING INDUSTRY'S HARDEST PROBLEMS FROM FIRST PRINCIPLES

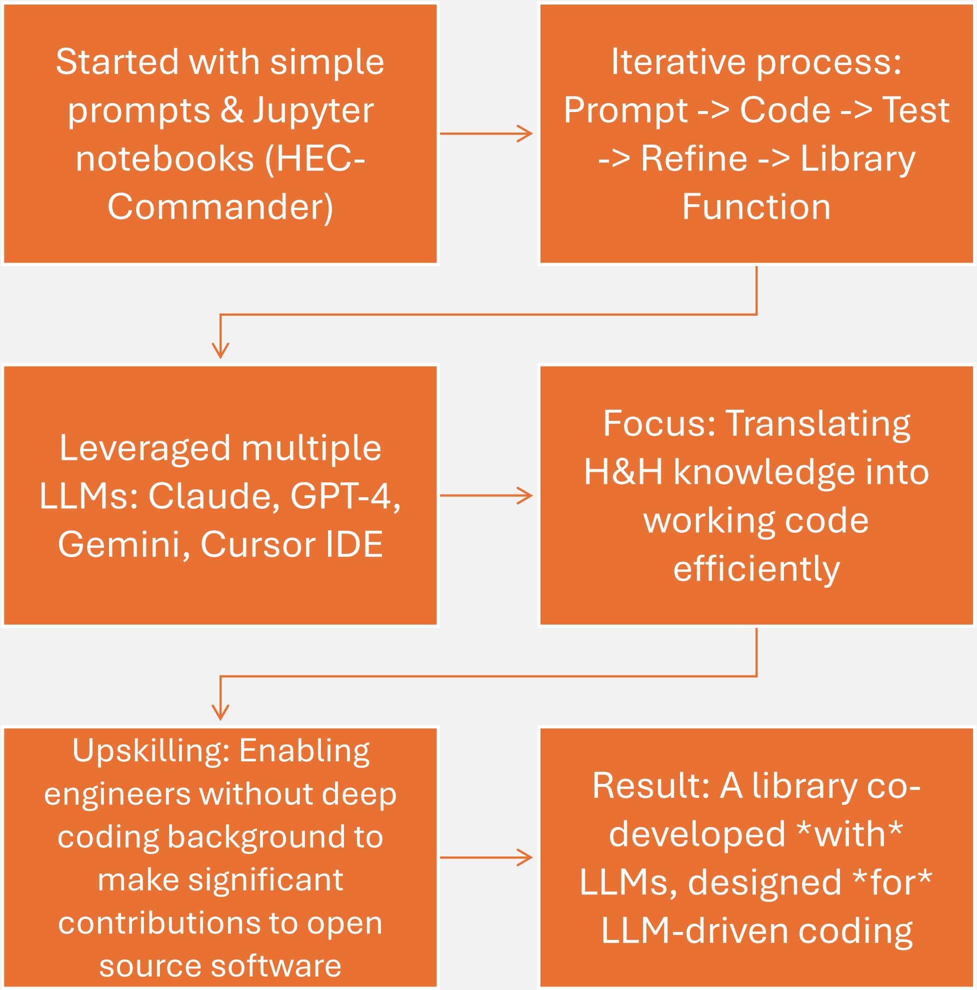
DOMAIN EXPERTISE ACCELERATED BY LANGUAGE MODELS

Vision Statement: To provide a reasonable framework for AI adoption in engineering practice with a concise and accurate slogan

CLB
ENGINEERING
CORPORATION

Read More

My Journey with LLM-Driven Code Development



Started with simple prompts & Jupyter notebooks (HEC-Commander)

Iterative process: Prompt -> Code -> Test -> Refine -> Library Function

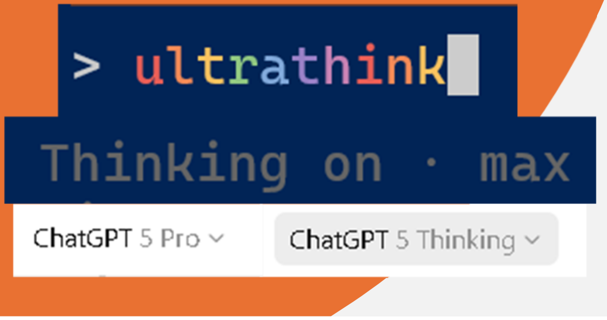
Leveraged multiple LLMs: Claude, GPT-4, Gemini, Cursor IDE

Focus: Translating H&H knowledge into working code efficiently

Upskilling: Enabling engineers without deep coding background to make significant contributions to open source software

Result: A library co-developed *with* LLMs, designed *for* LLM-driven coding

Reasoning Models: Another Step-Change Improvement



```
> ultrathink
Thinking on · max
```

- “Reasoning” or Test Time Compute created another step-change improvement in Large Language Models for scientific and math-based questions.

- LLM’s generate token-by-token, and generally can’t change previous tokens, once generates.

- Reasoning allowed previous tokens to be revised inside of the “thinking” response, reducing false descents – where an incorrect binary answer leads to the model going down a false path. Reasoning depends on the ability to do self-corrections and parallel branch predictions.

- For a human analogue, non-reasoning models find solutions based on first intuition vs reasoning models find solutions by applying orders of magnitude more self-reflection, internal scoring and logical analysis, up to a maximum limit of effort.

Larger input and output context windows introduced with reasoning models enable one-shot scripting and complex library development

```
$ pip install
ras-commander
```

Workflows Supported:

- Geometry, Plan and Unsteady Data Extraction
- Mannings N Sensitivity and Calibration
- Infiltration Sensitivity and Calibration
- RASMapper Floodplain Mapping Automation
- Fluvial/Pluvial Delineation
- Project Benefit and Adverse Impact Analysis

RAS Commander Library

Released March 2025

In a rapid 4-month development, a full HEC-RAS Automation Library was developed for the HEC-RAS 6.x series, with:

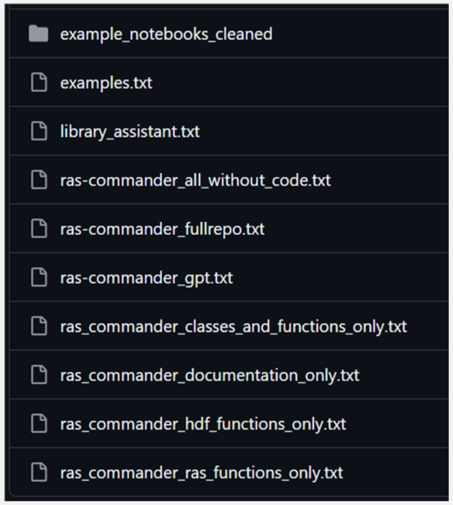
- Windows-native python automation via command line execution
- Flexible runtime options
- Project Management via Dataframes
- Multi-Project Capable
- Unified Function Calling by Plan Number
- HDF Data Connectors

This LLM Native Library allows development of scripts and notebooks quickly and easily, shifting the burden of development to applied creativity, iteration and verification.

Developed in ~4 Months and Presented at ASFPM 2025

RAS Commander’s LLM-Centered Design Choices

LLM-ready Knowledge Bases for web-based chat assistance

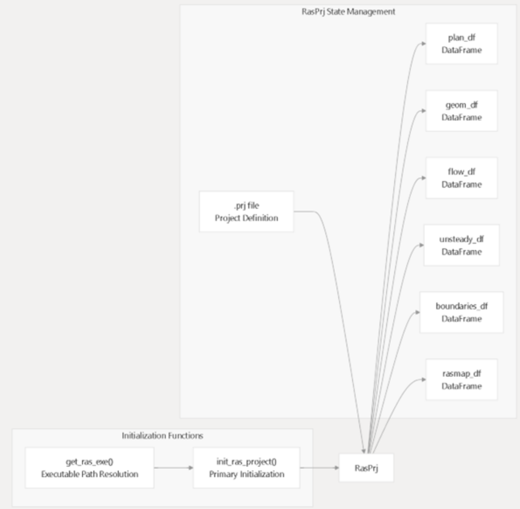


- example_notebooks_cleaned
- examples.txt
- library_assistant.txt
- ras-commander_all_without_code.txt
- ras-commander_fullrepo.txt
- ras-commander_gpt.txt
- ras_commander_classes_and_functions_only.txt
- ras_commander_documentation_only.txt
- ras_commander_hdf_functions_only.txt
- ras_commander_ras_functions_only.txt

Well-Documented Design Patterns and Organizational Structure

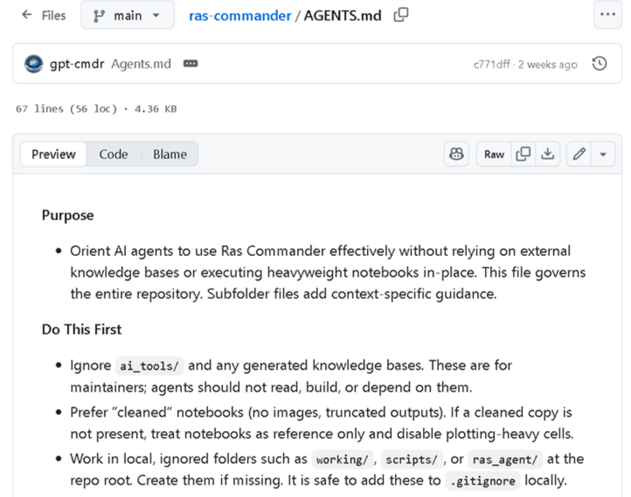
- Aligns Coding Efforts
- Reduces Hallucinations and Errors
- Blueprint for future development

LLM Powered Repository Indexing by DeepWiki



Open source code enables public tools like Deepwiki to index, creatively visualize, interpret and report repository code structures and design patterns for future LLM training, supercharging future abilities.

Agent-Ready with LLM Instructions Files for Agentic CLI Tools



ras-commander / AGENTS.md

gpt-cmdr Agents.md c771dff · 2 weeks ago

67 lines (56 loc) · 4.36 KB

Purpose

- Orient AI agents to use Ras Commander effectively without relying on external knowledge bases or executing heavyweight notebooks in-place. This file governs the entire repository. Subfolder files add context-specific guidance.

Do This First

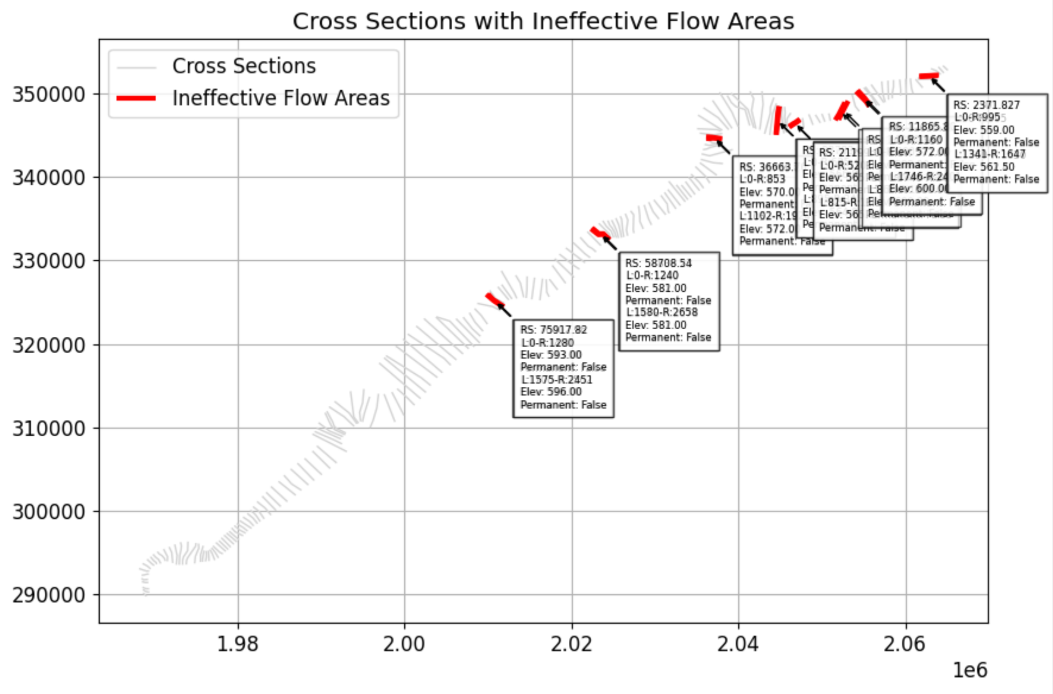
- Ignore `ai_tools/` and any generated knowledge bases. These are for maintainers; agents should not read, build, or depend on them.
- Prefer ”cleaned” notebooks (no images, truncated outputs). If a cleaned copy is not present, treat notebooks as reference only and disable plotting-heavy cells.
- Work in local, ignored folders such as `working/`, `scripts/`, or `ras_agent/` at the repo root. Create them if missing. It is safe to add these to `.gitignore` locally.

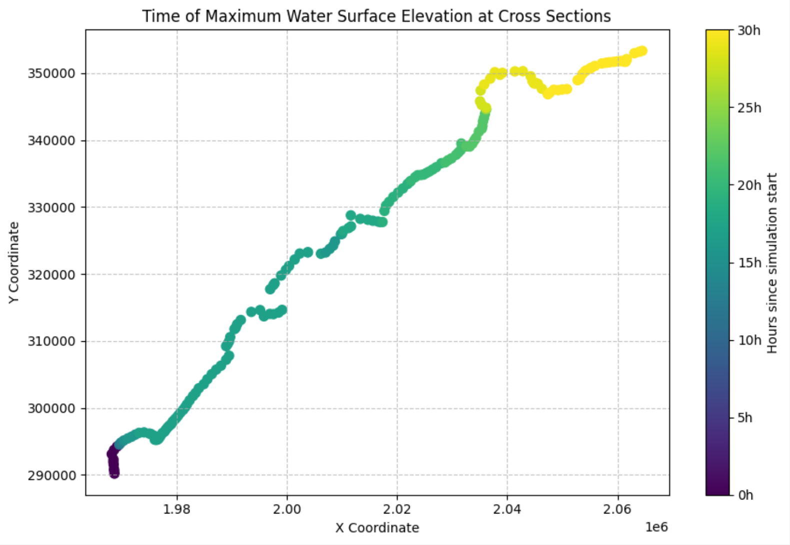
Knowledge Bases and Agent Documentation Facilitate:

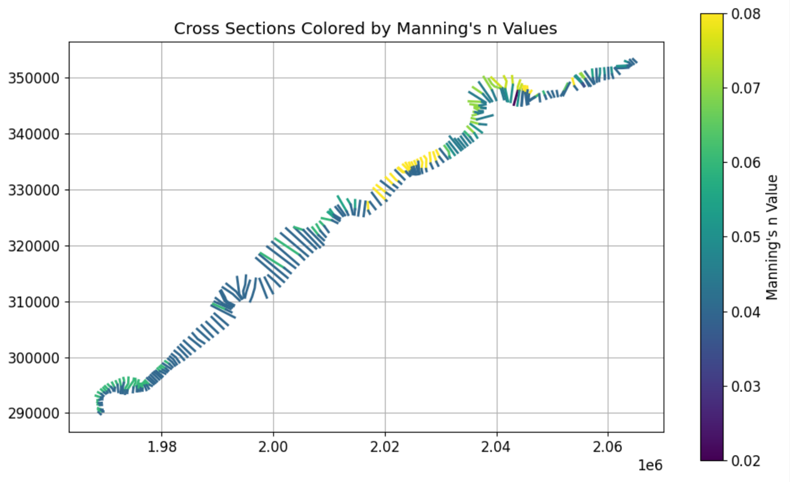
- Flexible use within any web-based chatbot
- Broad Integrations and Targeted Development
- Improved Cursor codebase navigation
- Instructions for popular CLI tools
- Flexible usage of the library with your chosen LLM, architectures and frameworks by building the repository to be LLM coding agent driven by design

Accessing 1D HDF Data

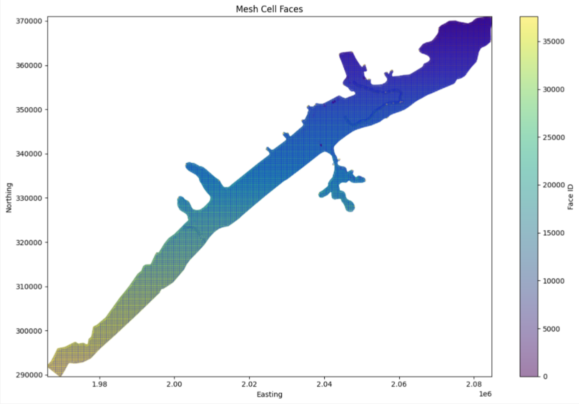
Programmatic access for hard-to-reach data like ineffective flow area permanence, or creative report visualizations

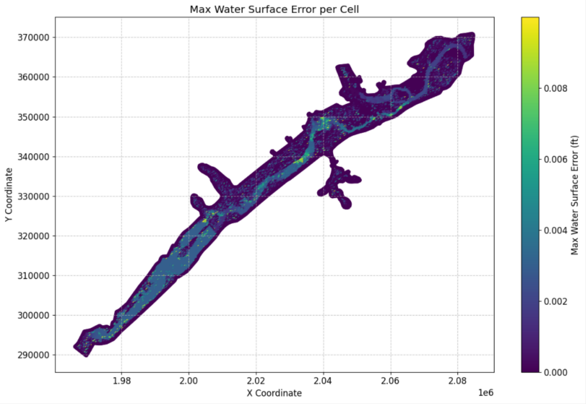


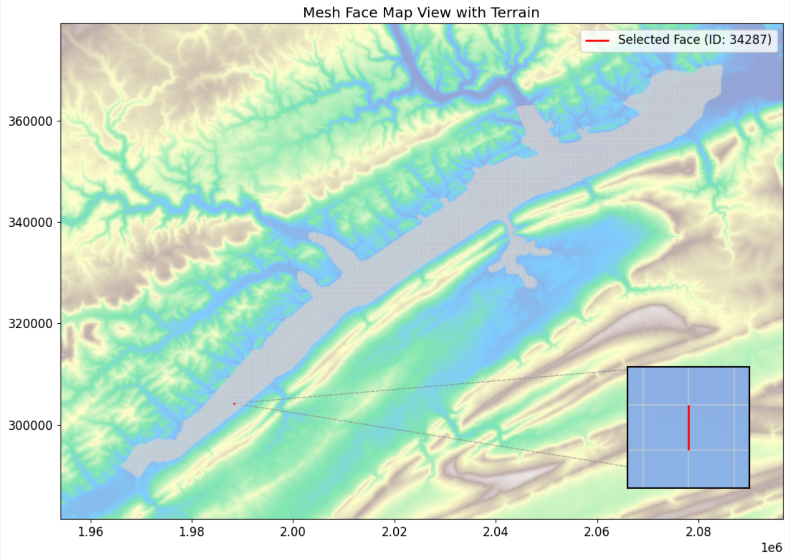


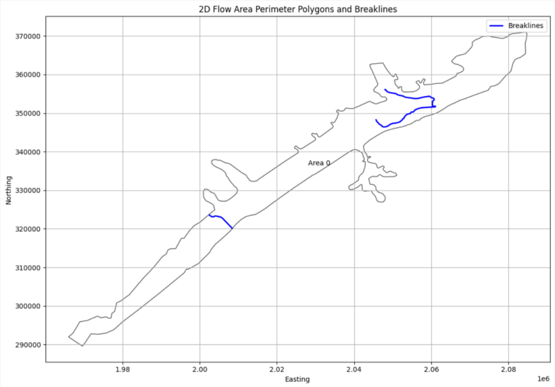


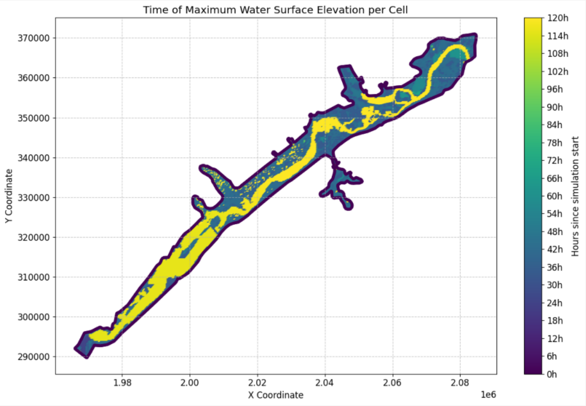
Accessing 2D HDF Data

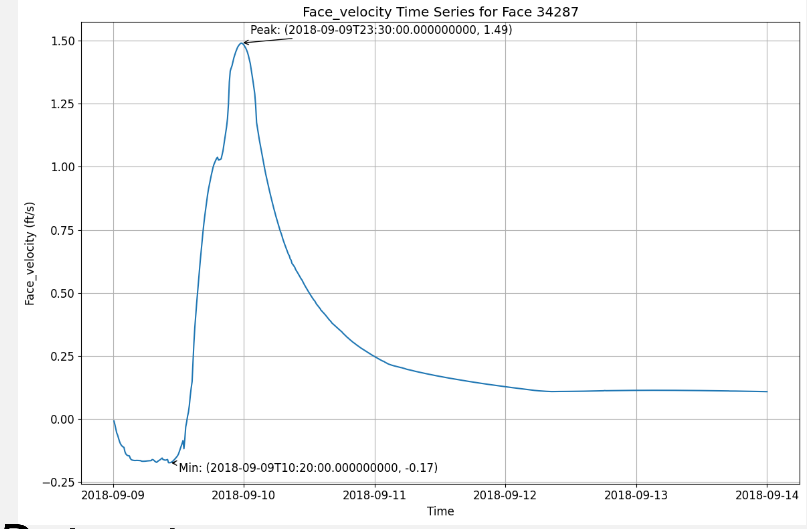






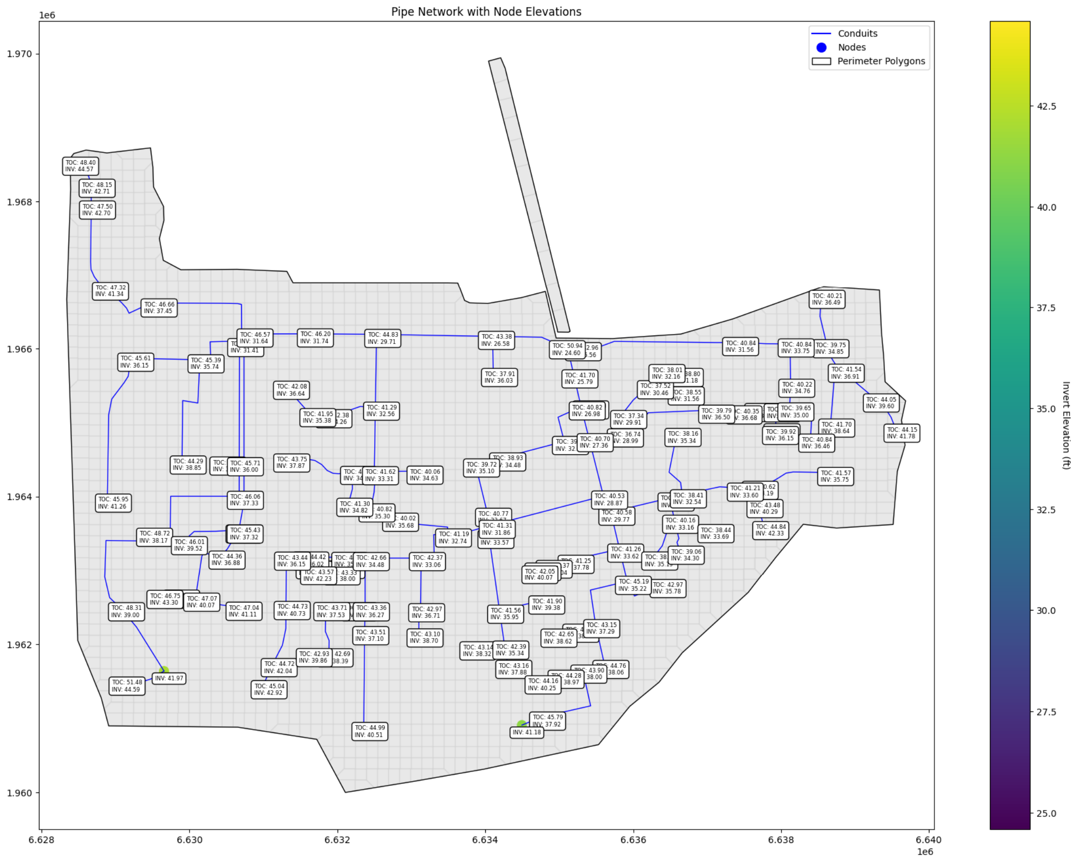


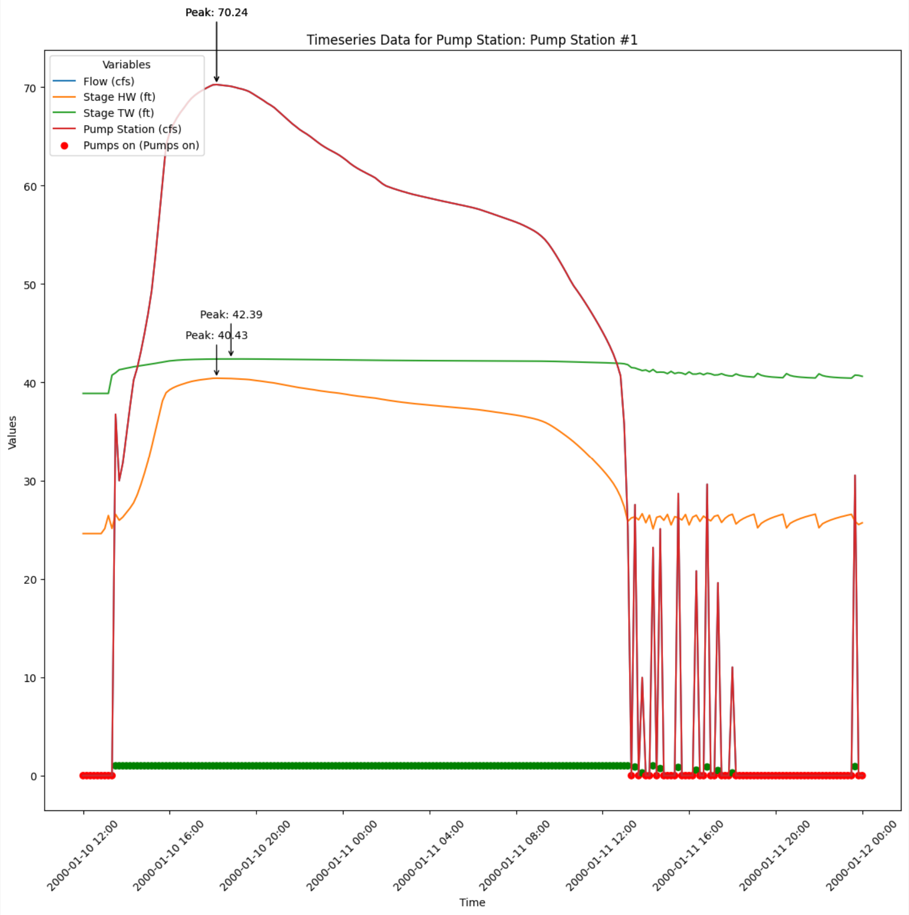




HDF Data now available via Geodataframes or Xarray Datasets

Pipes & Pumps HDF Data Extraction

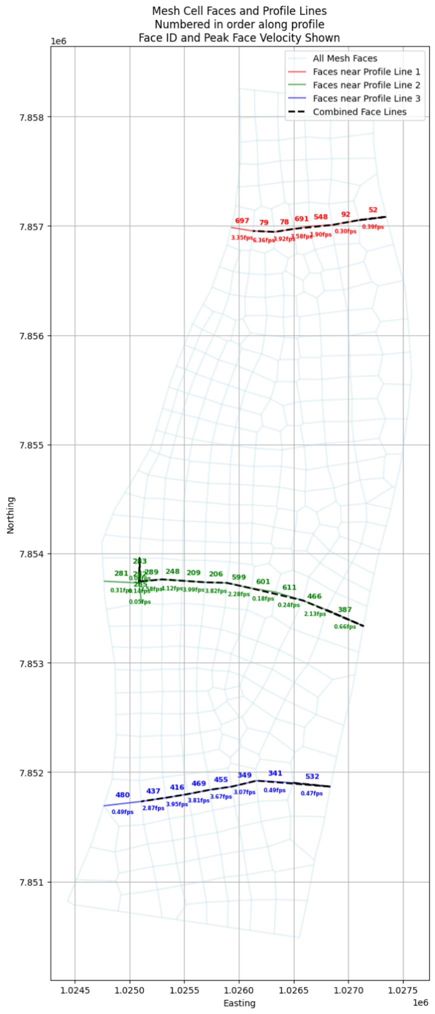




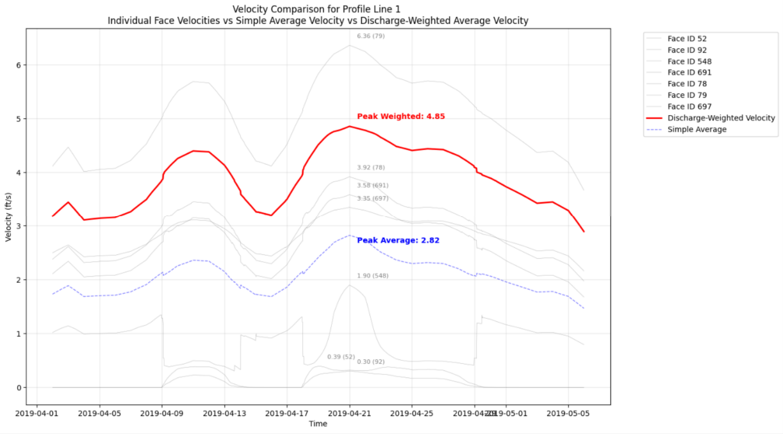
2D Detail Face Data Extraction

Example notebook showing process for extracting mesh faces along a profile line to calculate a discharge-weighted average velocity.

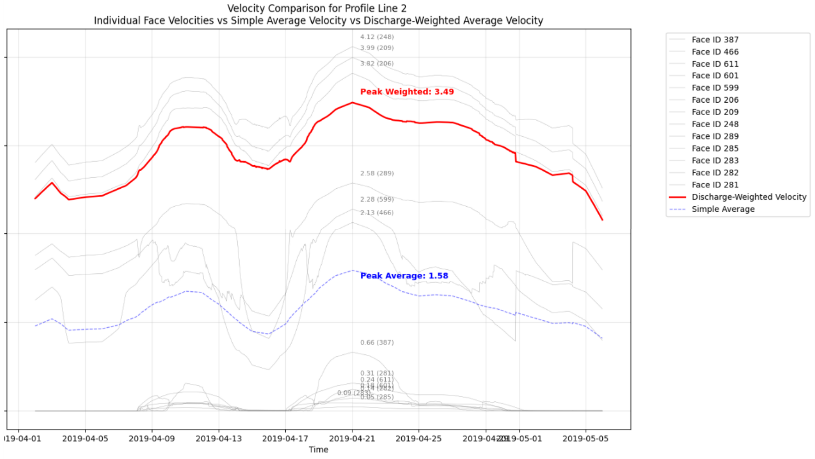
- Define custom profile lines (e.g., from GeoJSON).
- Identify mesh faces intersecting or near the profile line.
- Filter faces based on orientation relative to flow (optional).
- Calculate metrics like Discharge-Weighted Velocity along the profile:
- V_weighted = Sum(|Q_face| * |V_face|) / Sum(|Q_face|) across selected faces.



Mesh Cell Faces and Profile Lines
Numbered in order along profile
Face ID and Peak Face Velocity Shown

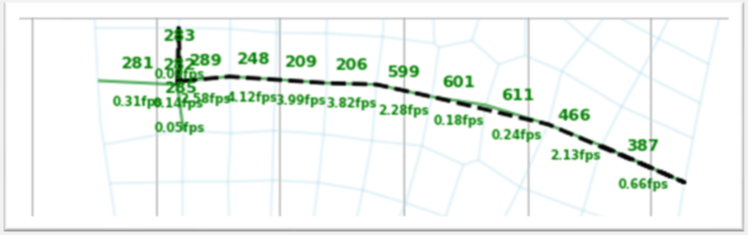


Velocity Comparison for Profile Line 1
Individual Face Velocities vs Simple Average Velocity vs Discharge-Weighted Average Velocity

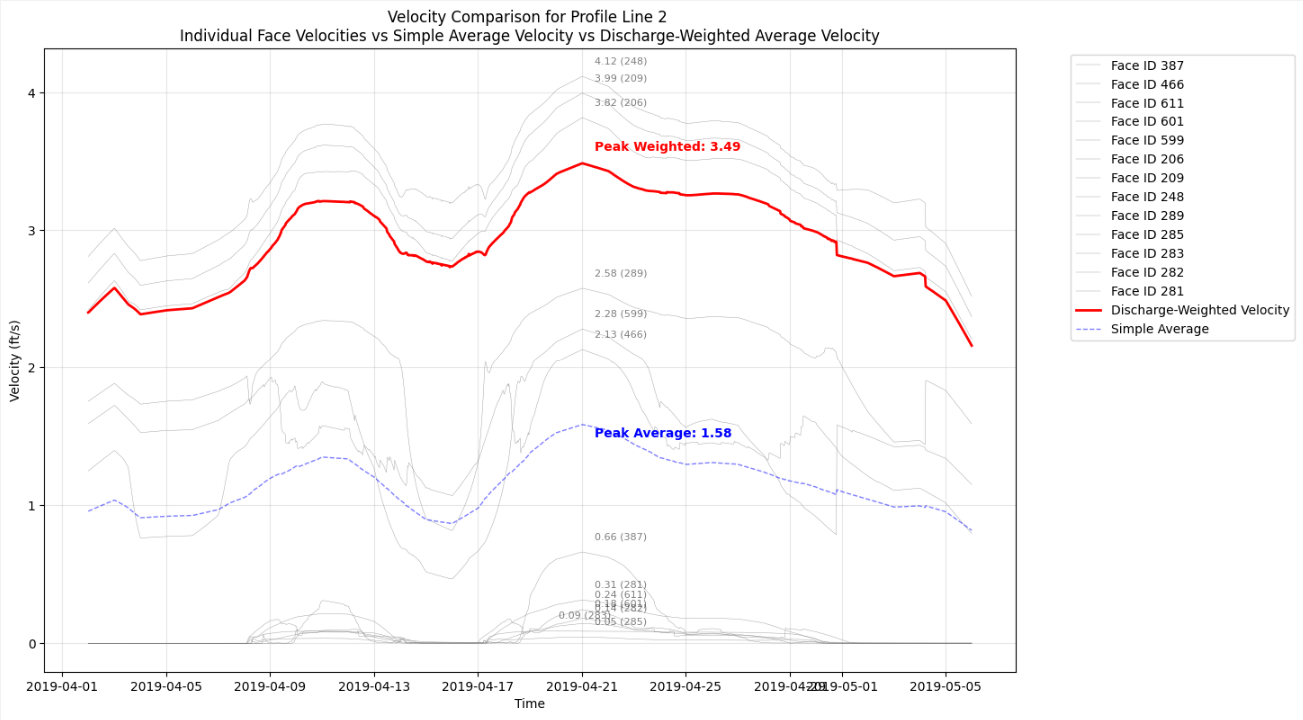


Velocity Comparison for Profile Line 2
Individual Face Velocities vs Simple Average Velocity vs Discharge-Weighted Average Velocity

Calculating Discharge Weighted Average Velocity



The example notebook includes the calculation of Discharge-Weighted Average Velocity and other variables that would otherwise only be available via Reference Lines

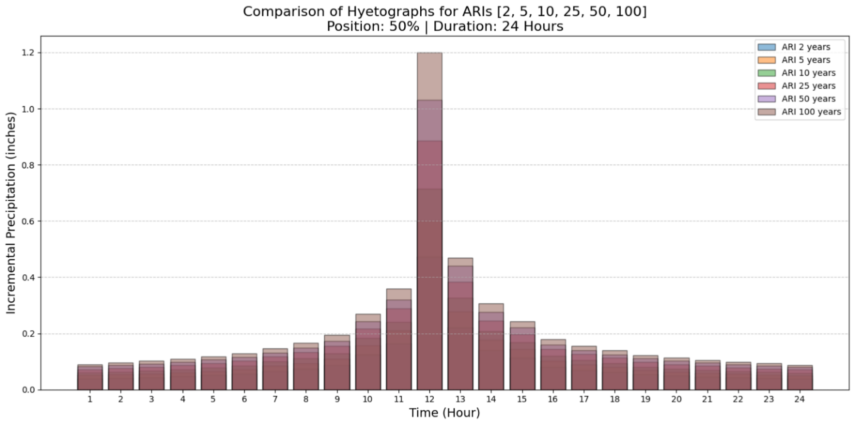


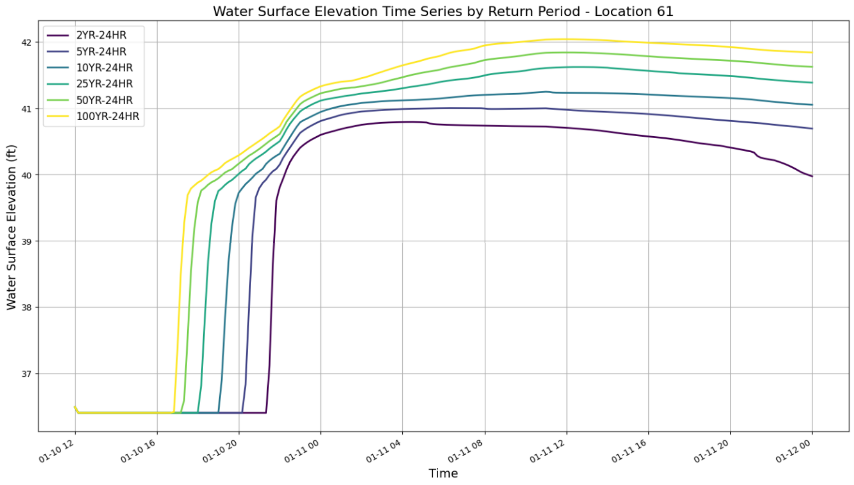
Automated Atlas 14 AEP Events for Rain on Grid

Example Notebook Demonstrating this workflow:

- Read NOAA Atlas 14 precipitation data (CSV).
- Generate balanced storm hyetographs (Alternating Block Method).
- Programmatically create new .u* files with hyetographs.
- Clone base plan (.p*) for each AEP (2yr, 10yr, 100yr, etc.).
- Link new unsteady files to new plans.
- Execute all AEP plans in parallel (compute_parallel).
- Extract and compare results (e.g., max WSE vs. AEP).

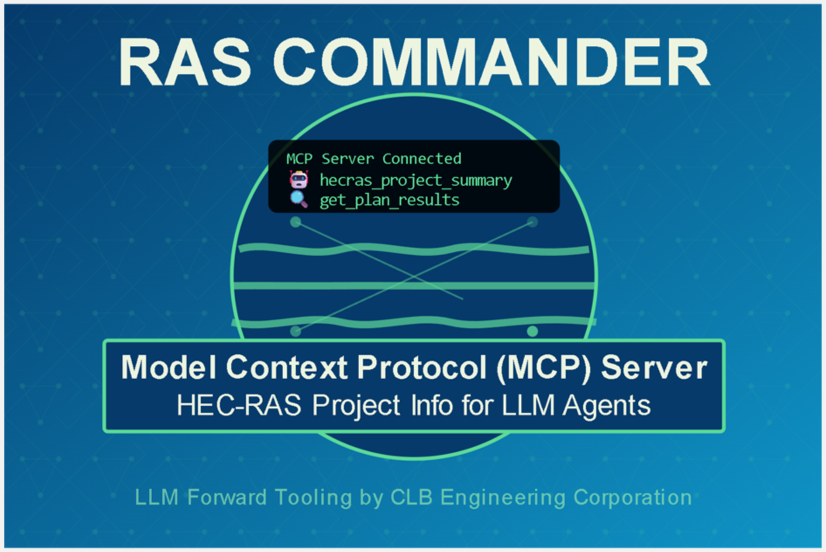
Extend to include upper and lower confidence bounds, additional durations + more with a simple request to your favorite LLM!



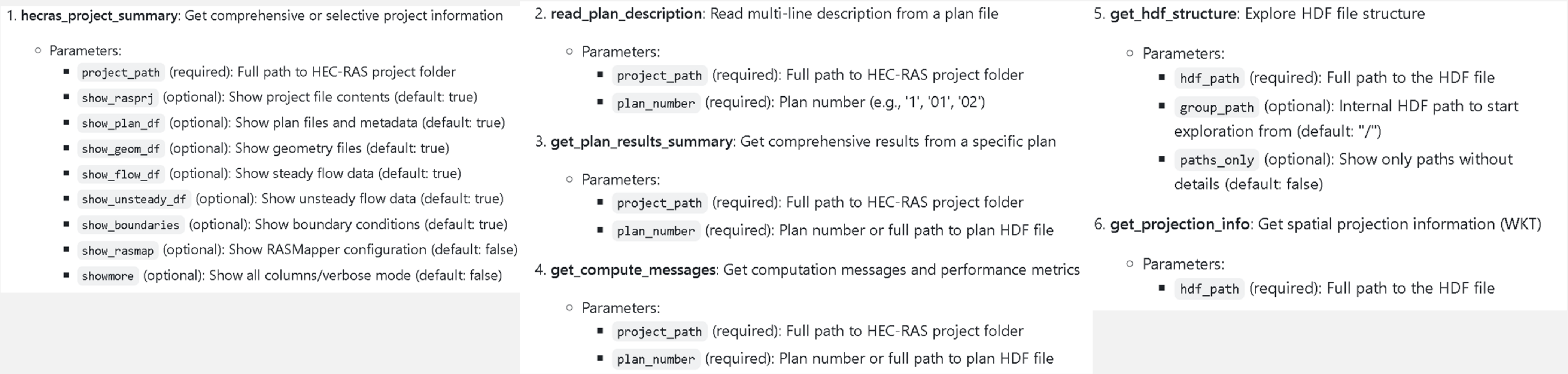


HEC-RAS MCP Server

- Adds LLM Tool Calls for retrieving all of the standard RAS-Commander Dataframes
- Uses “uvx” for executing as a self-contained tool within Claude Desktop or ChatGPT
- Great for basic QAQC and querying unfamiliar models
- Can retrieve compute messages for debugging


RAS COMMANDER
MCP Server Connected
hecras_project_summary
get_plan_results
Model Context Protocol (MCP) Server
HEC-RAS Project Info for LLM Agents
LLM Forward Tooling by CLB Engineering Corporation

Available Tool Calls:

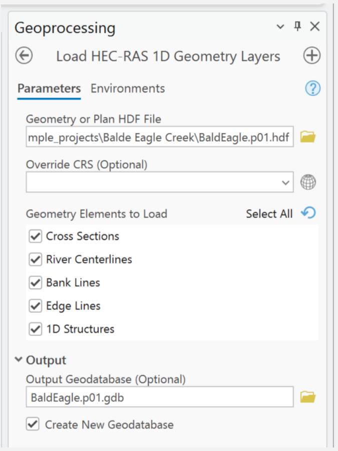


1. **hecras_project_summary**: Get comprehensive or selective project information
   - Parameters:
     - `project_path` (required): Full path to HEC-RAS project folder
     - `show_rasprj` (optional): Show project file contents (default: true)
     - `show_plan_df` (optional): Show plan files and metadata (default: true)
     - `show_geom_df` (optional): Show geometry files (default: true)
     - `show_flow_df` (optional): Show steady flow data (default: true)
     - `show_unsteady_df` (optional): Show unsteady flow data (default: true)
     - `show_boundaries` (optional): Show boundary conditions (default: true)
     - `show_rasmap` (optional): Show RASMapper configuration (default: false)
     - `showmore` (optional): Show all columns/verbose mode (default: false)
2. **read_plan_description**: Read multi-line description from a plan file
   - Parameters:
     - `project_path` (required): Full path to HEC-RAS project folder
     - `plan_number` (required): Plan number (e.g., '1', '01', '02')
3. **get_plan_results_summary**: Get comprehensive results from a specific plan
   - Parameters:
     - `project_path` (required): Full path to HEC-RAS project folder
     - `plan_number` (required): Plan number or full path to plan HDF file
4. **get_compute_messages**: Get computation messages and performance metrics
   - Parameters:
     - `project_path` (required): Full path to HEC-RAS project folder
     - `plan_number` (required): Plan number or full path to plan HDF file
5. **get_hdf_structure**: Explore HDF file structure
   - Parameters:
     - `hdf_path` (required): Full path to the HDF file
     - `group_path` (optional): Internal HDF path to start exploration from (default: "/")
     - `paths_only` (optional): Show only paths without details (default: false)
6. **get_projection_info**: Get spatial projection information (WKT)
   - Parameters:
     - `hdf_path` (required): Full path to the HDF file

RAS Commander Arc Hydro Toolboxes

- Rapid Toolbox Development
- Released at ESRI’s 2025 User Conference
- Download today with Arc Hydro Tools
- Version 6.x Only
- Feedback welcome

RAS COMMANDER
Toolboxes:
▸ RAS Commander Tools
ARC HYDRO TOOLS
HEC-RAS 6.x Data for ArcGIS Pro
In Partnership with CLB Engineering Corporation


Geoprocessing

Load HEC-RAS 1D Geometry Layers

Parameters | Environments

Geometry or Plan HDF File: mple_projects\Balde Eagle Creek\BaldEagle.p01.hdf

Override CRS (Optional):

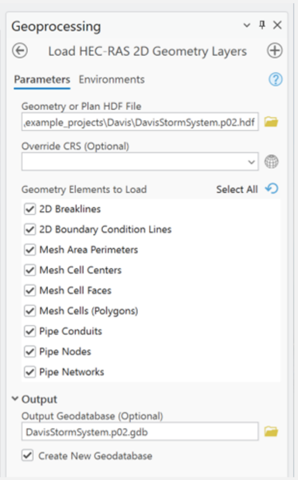
Geometry Elements to Load — Select All

- ☑ Cross Sections
- ☑ River Centerlines
- ☑ Bank Lines
- ☑ Edge Lines
- ☑ 1D Structures

Output

Output Geodatabase (Optional): BaldEagle.p01.gdb

☑ Create New Geodatabase


Geoprocessing

Load HEC-RAS 2D Geometry Layers

Parameters | Environments

Geometry or Plan HDF File: ,example_projects\Davis\DavisStormSystem.p02.hdf

Override CRS (Optional):

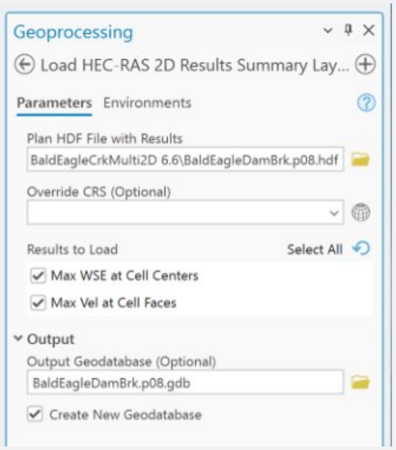
Geometry Elements to Load — Select All

- ☑ 2D Breaklines
- ☑ 2D Boundary Condition Lines
- ☑ Mesh Area Perimeters
- ☑ Mesh Cell Centers
- ☑ Mesh Cell Faces
- ☑ Mesh Cells (Polygons)
- ☑ Pipe Conduits
- ☑ Pipe Nodes
- ☑ Pipe Networks

Output

Output Geodatabase (Optional): DavisStormSystem.p02.gdb

☑ Create New Geodatabase


Geoprocessing

Load HEC-RAS 2D Results Summary Lay...

Parameters | Environments

Plan HDF File with Results: BaldEagleCrkMulti2D 6.6\BaldEagleDamBrk.p08.hdf

Override CRS (Optional):

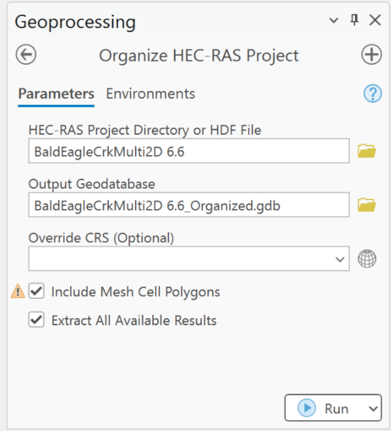
Results to Load — Select All

- ☑ Max WSE at Cell Centers
- ☑ Max Vel at Cell Faces

Output

Output Geodatabase (Optional): BaldEagleDamBrk.p08.gdb

☑ Create New Geodatabase


Geoprocessing

Organize HEC-RAS Project

Parameters | Environments

HEC-RAS Project Directory or HDF File: BaldEagleCrkMulti2D 6.6

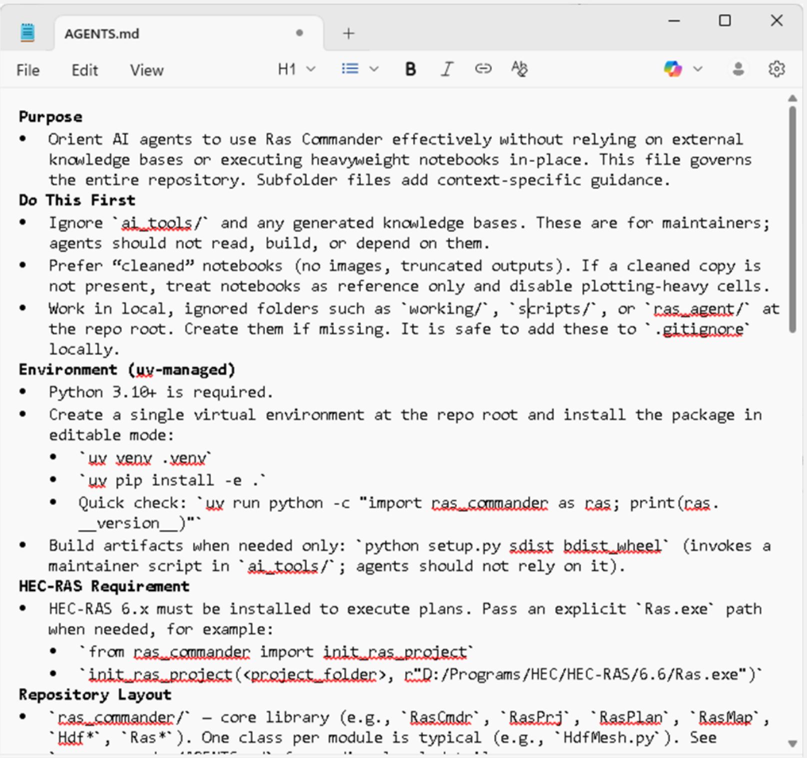
Output Geodatabase: BaldEagleCrkMulti2D 6.6_Organized.gdb

Override CRS (Optional):

⚠ ☑ Include Mesh Cell Polygons

☑ Extract All Available Results

Enabling Agentic Workflows



Purpose

- Orient AI agents to use Ras Commander effectively without relying on external knowledge bases or executing heavyweight notebooks in-place. This file governs the entire repository. Subfolder files add context-specific guidance.

Do This First

- Ignore `ai_tools/` and any generated knowledge bases. These are for maintainers; agents should not read, build, or depend on them.
- Prefer “cleaned” notebooks (no images, truncated outputs). If a cleaned copy is not present, treat notebooks as reference only and disable plotting-heavy cells.
- Work in local, ignored folders such as `working/`, `scripts/`, or `ras_agent/` at the repo root. Create them if missing. It is safe to add these to `.gitignore` locally.

Environment (uv-managed)

- Python 3.10+ is required.
- Create a single virtual environment at the repo root and install the package in editable mode:
  - `uv venv .venv`
  - `uv pip install -e .`
  - Quick check: `uv run python -c "import ras_commander as ras; print(ras.__version__)"`
- Build artifacts when needed only: `python setup.py sdist bdist_wheel` (invokes a maintainer script in `ai_tools/`; agents should not rely on it).

HEC-RAS Requirement

- HEC-RAS 6.x must be installed to execute plans. Pass an explicit `Ras.exe` path when needed, for example:
  - `from ras_commander import init_ras_project`
  - `init_ras_project(<project_folder>, r"D:/Programs/HEC/HEC-RAS/6.6/Ras.exe")`

Repository Layout

- `ras_commander/` – core library (e.g., `RasCmdr`, `RasPrj`, `RasPlan`, `RasMap`, `Hdf*`, `Ras*`). One class per module is typical (e.g., `HdfMesh.py`). See

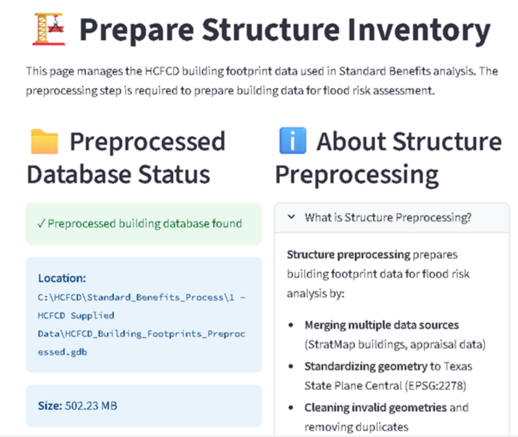
RAS Commander has been upgraded with an Agents.md, which contain information for LLM agents using the repository.

This allows flexible use of the RAS Commander logic to generate helpful tools to assist with your modeling workload.

Just open Claude Code/Codex or your favorite CLI tool in the ras-commander directory to give your LLM tools to work with HEC-RAS.

Coming Soon:

Automating Standard Benefits Frameworks and FEMA BCA Inputs from HEC-RAS Projects



Prepare Structure Inventory

This page manages the HCFCD building footprint data used in Standard Benefits analysis. The preprocessing step is required to prepare building data for flood risk assessment.

Preprocessed Database Status

✓ Preprocessed building database found

Location:
`C:\HCFCD\Standard_Benefits_Process\1 - HCFCD Supplied Data\HCFCD_Building_Footprints_Preprocessed.gdb`

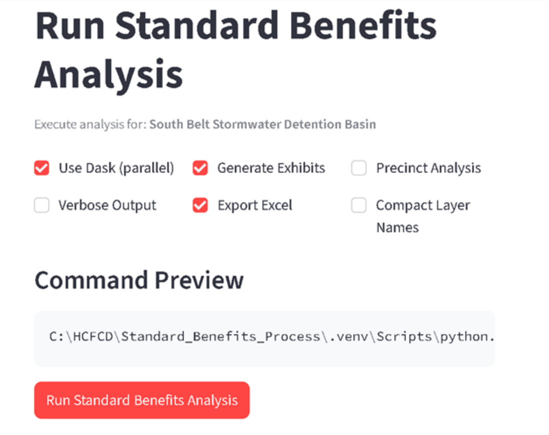
Size: 502.23 MB

About Structure Preprocessing

What is Structure Preprocessing?

Structure preprocessing prepares building footprint data for flood risk analysis by:

- **Merging multiple data sources** (StratMap buildings, appraisal data)
- **Standardizing geometry** to Texas State Plane Central (EPSG:2278)
- **Cleaning invalid geometries** and removing duplicates



Run Standard Benefits Analysis

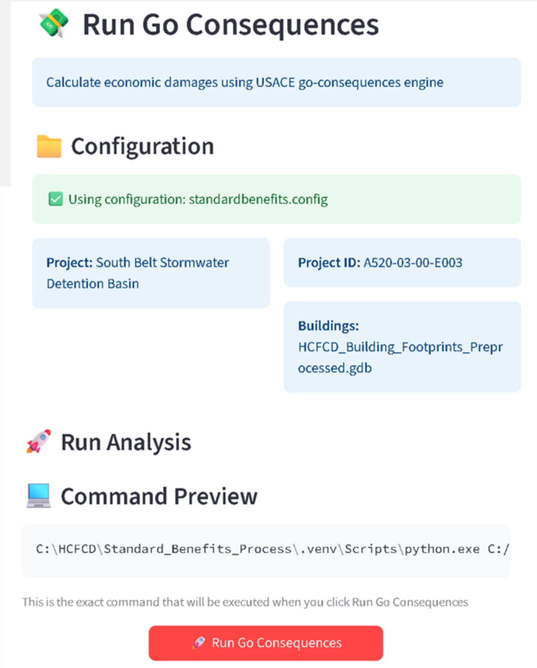
Execute analysis for: **South Belt Stormwater Detention Basin**

☑ Use Dask (parallel) ☑ Generate Exhibits ☐ Precinct Analysis

☐ Verbose Output ☑ Export Excel ☐ Compact Layer Names

Command Preview

```
C:\HCFCD\Standard_Benefits_Process\.venv\Scripts\python.
```



Run Go Consequences

Calculate economic damages using USACE go-consequences engine

Configuration

☑ Using configuration: standardbenefits.config

Project: South Belt Stormwater Detention Basin

Project ID: A520-03-00-E003

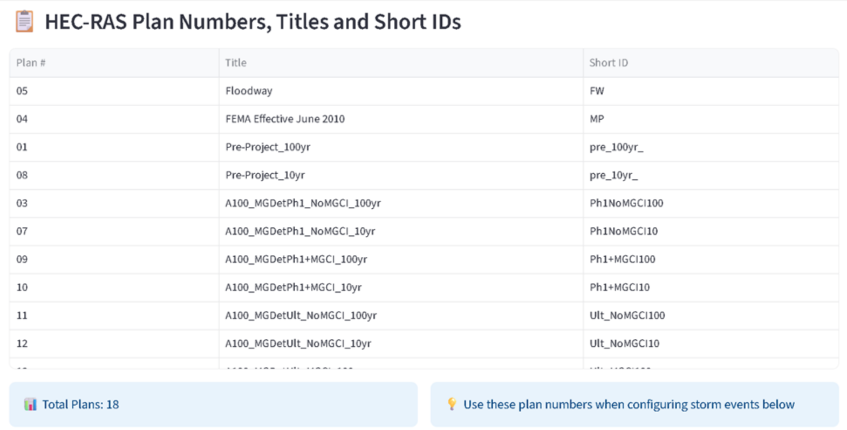
Buildings: HCFCD_Building_Footprints_Preprocessed.gdb

Run Analysis

Command Preview

```
C:\HCFCD\Standard_Benefits_Process\.venv\Scripts\python.exe C:/
```

This is the exact command that will be executed when you click Run Go Consequences

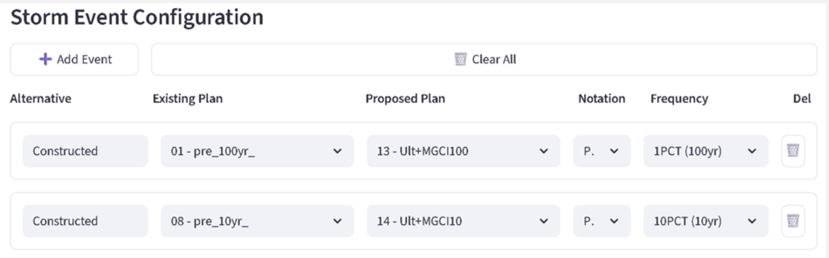


HEC-RAS Plan Numbers, Titles and Short IDs

Plan #	Title	Short ID
05	Floodway	FW
04	FEMA Effective June 2010	MP
01	Pre-Project_100yr	pre_100yr_
08	Pre-Project_10yr	pre_10yr_
03	A100_MGDetPh1_NoMGCI_100yr	Ph1NoMGCI100
07	A100_MGDetPh1_NoMGCI_10yr	Ph1NoMGCI10
09	A100_MGDetPh1+MGCI_100yr	Ph1+MGCI100
10	A100_MGDetPh1+MGCI_10yr	Ph1+MGCI10
11	A100_MGDetUlt_NoMGCI_100yr	Ult_NoMGCI100
12	A100_MGDetUlt_NoMGCI_10yr	Ult_NoMGCI10

Total Plans: 18

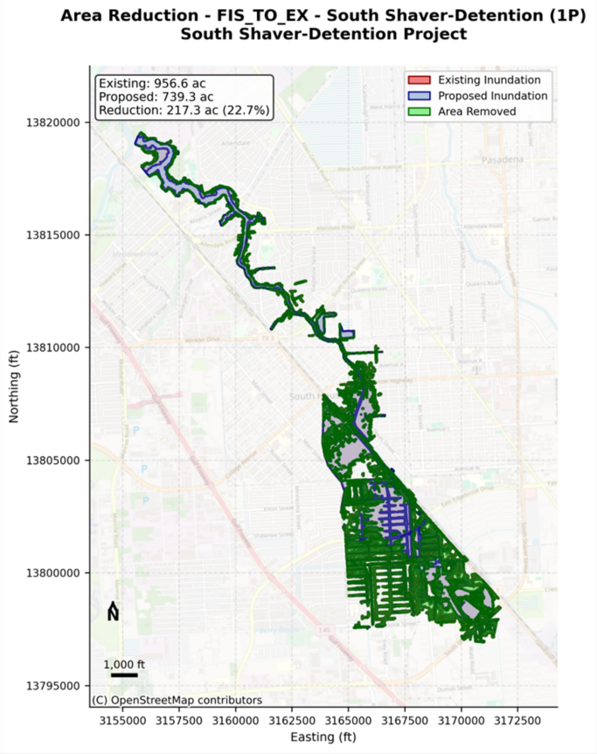
Use these plan numbers when configuring storm events below

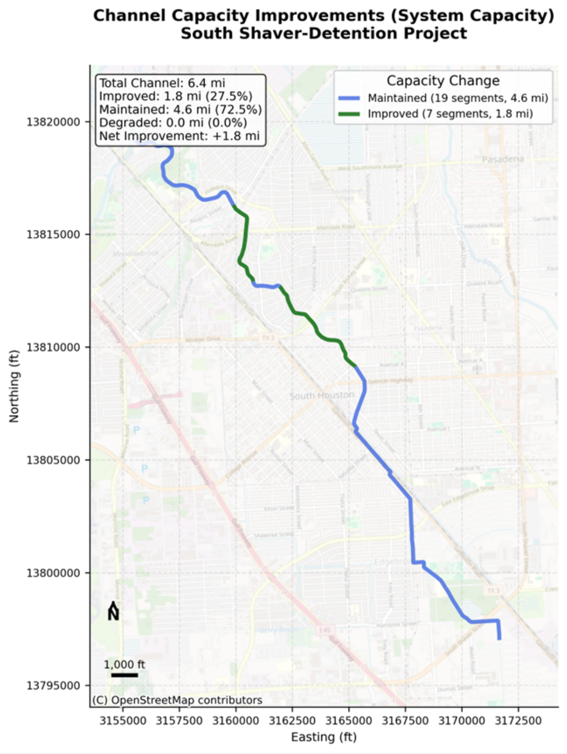


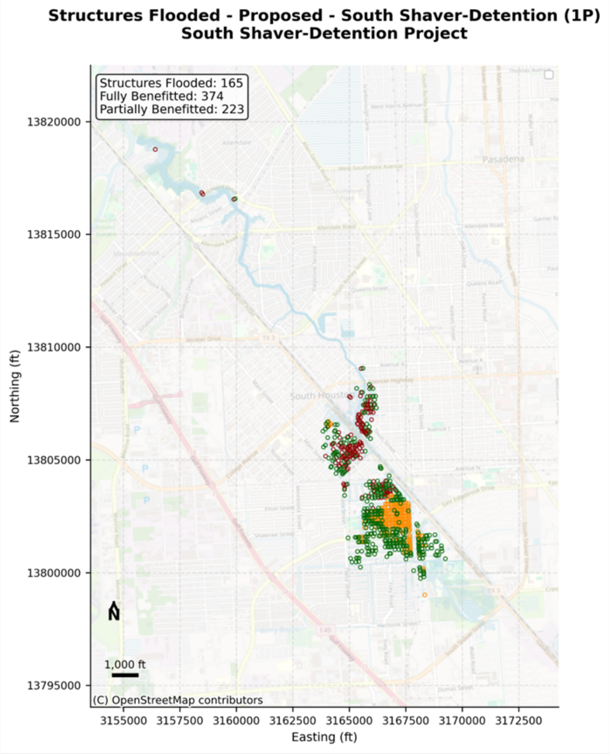
Storm Event Configuration

Alternative	Existing Plan	Proposed Plan	Notation	Frequency	Del
Constructed	01 - pre_100yr_	13 - Ult+MGCI100	P.	1PCT (100yr)	
Constructed	08 - pre_10yr_	14 - Ult+MGCI10	P.	10PCT (10yr)	

Showing Benefits for Area of Inundation, Channel Capacity and Roadway


Area Reduction - FIS_TO_EX - South Shaver-Detention (1P)
South Shaver-Detention Project


Channel Capacity Improvements (System Capacity)
South Shaver-Detention Project


Structures Flooded - Proposed - South Shaver-Detention (1P)
South Shaver-Detention Project

Coming Soon:

Dam Break Simulation Dashboard

In Partnership with Tennessee Technical University

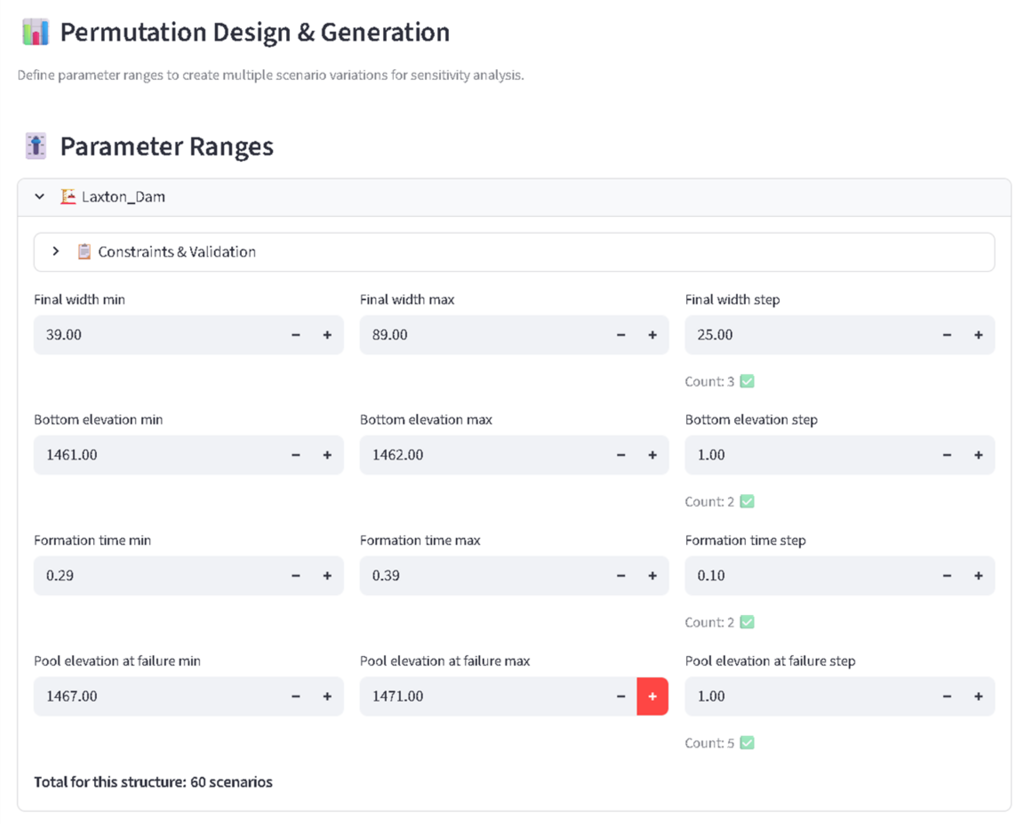
Objective: Conducting dam breach simulations using randomly sampled values for:

- Pool Elevation at Failure
- Breach Bottom Elevation
- Breach Formation Time
- Final Bottom Width

In Partnership with Tennessee Tech Department of Civil & Environmental Engineering's Center for Management, Utilization and Protection of Water Resources

Permutation Design and Generation

- Linear Distribution to start
- Currently under development and testing
- Focused on development of a statistical framework for *Sequential* Dam Breach failures, where cross-permutations need to be sampled.



📊 Permutation Design & Generation

Define parameter ranges to create multiple scenario variations for sensitivity analysis.

Parameter Ranges

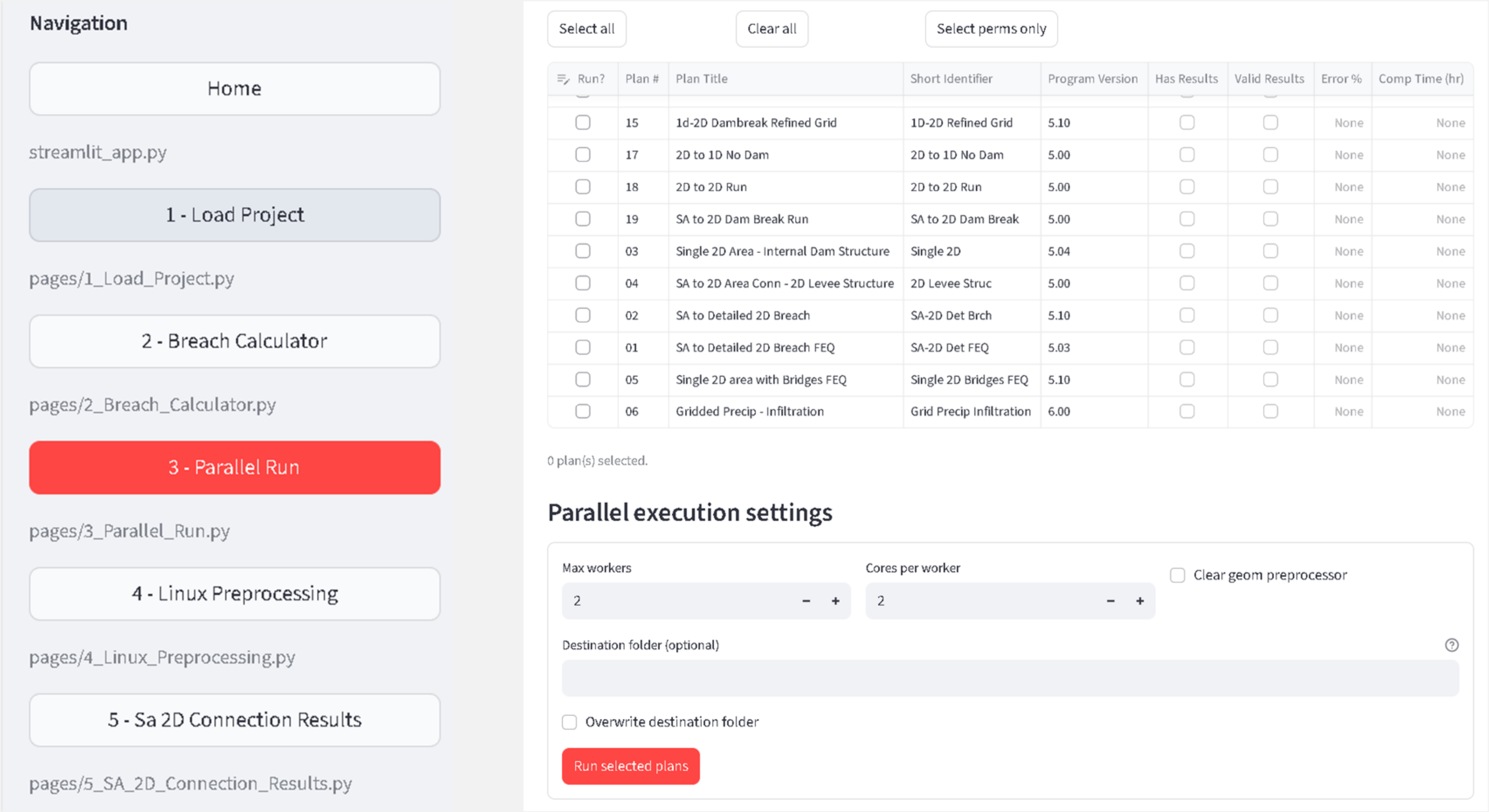
Laxton_Dam

Constraints & Validation

Final width min 39.00	Final width max 89.00	Final width step 25.00 Count: 3 ✓
Bottom elevation min 1461.00	Bottom elevation max 1462.00	Bottom elevation step 1.00 Count: 2 ✓
Formation time min 0.29	Formation time max 0.39	Formation time step 0.10 Count: 2 ✓
Pool elevation at failure min 1467.00	Pool elevation at failure max 1471.00	Pool elevation at failure step 1.00 Count: 5 ✓

Total for this structure: 60 scenarios

Dashboard Extras: Parallel Execution



Run?	Plan #	Plan Title	Short Identifier	Program Version	Has Results	Valid Results	Error %	Comp Time (hr)
□	15	1d-2D Dambreak Refined Grid	1D-2D Refined Grid	5.10	□	□	None	None
□	17	2D to 1D No Dam	2D to 1D No Dam	5.00	□	□	None	None
□	18	2D to 2D Run	2D to 2D Run	5.00	□	□	None	None
□	19	SA to 2D Dam Break Run	SA to 2D Dam Break	5.00	□	□	None	None
□	03	Single 2D Area - Internal Dam Structure	Single 2D	5.04	□	□	None	None
□	04	SA to 2D Area Conn - 2D Levee Structure	2D Levee Struc	5.00	□	□	None	None
□	02	SA to Detailed 2D Breach	SA-2D Det Brch	5.10	□	□	None	None
□	01	SA to Detailed 2D Breach FEQ	SA-2D Det FEQ	5.03	□	□	None	None
□	05	Single 2D area with Bridges FEQ	Single 2D Bridges FEQ	5.10	□	□	None	None
□	06	Gridded Precip - Infiltration	Grid Precip Infiltration	6.00	□	□	None	None

0 plan(s) selected.

Parallel execution settings

Max workers: 2

Cores per worker: 2

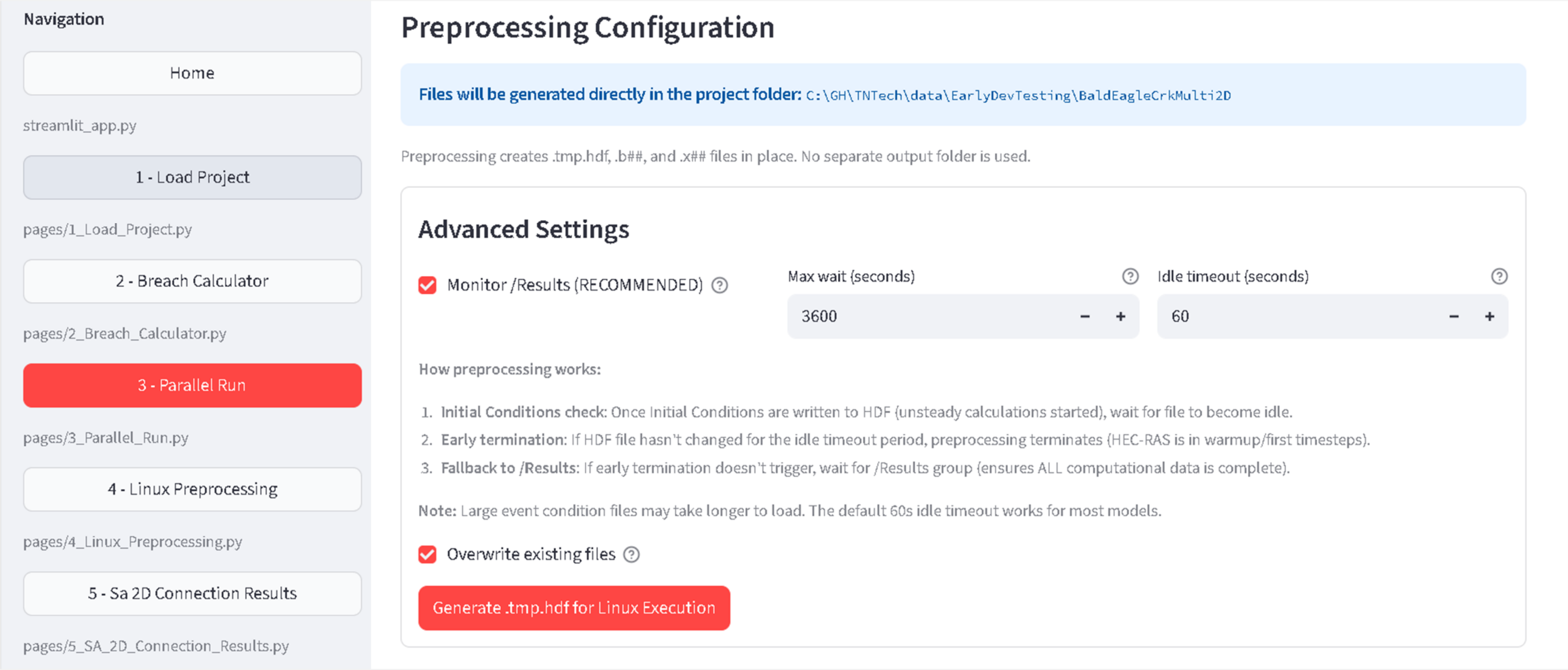
□ Clear geom preprocessor

Destination folder (optional):

□ Overwrite destination folder

For models with short runtimes, this is adequate for running small batches

Linux Run Preprocessing for HEC-RAS



Preprocessing Configuration

Files will be generated directly in the project folder: `C:\GH\TNTech\data\EarlyDevTesting\BaldEagleCrkMulti2D`

Preprocessing creates .tmp.hdf, .b##, and .x## files in place. No separate output folder is used.

Advanced Settings

☑ Monitor /Results (RECOMMENDED)

Max wait (seconds): 3600

Idle timeout (seconds): 60

How preprocessing works:

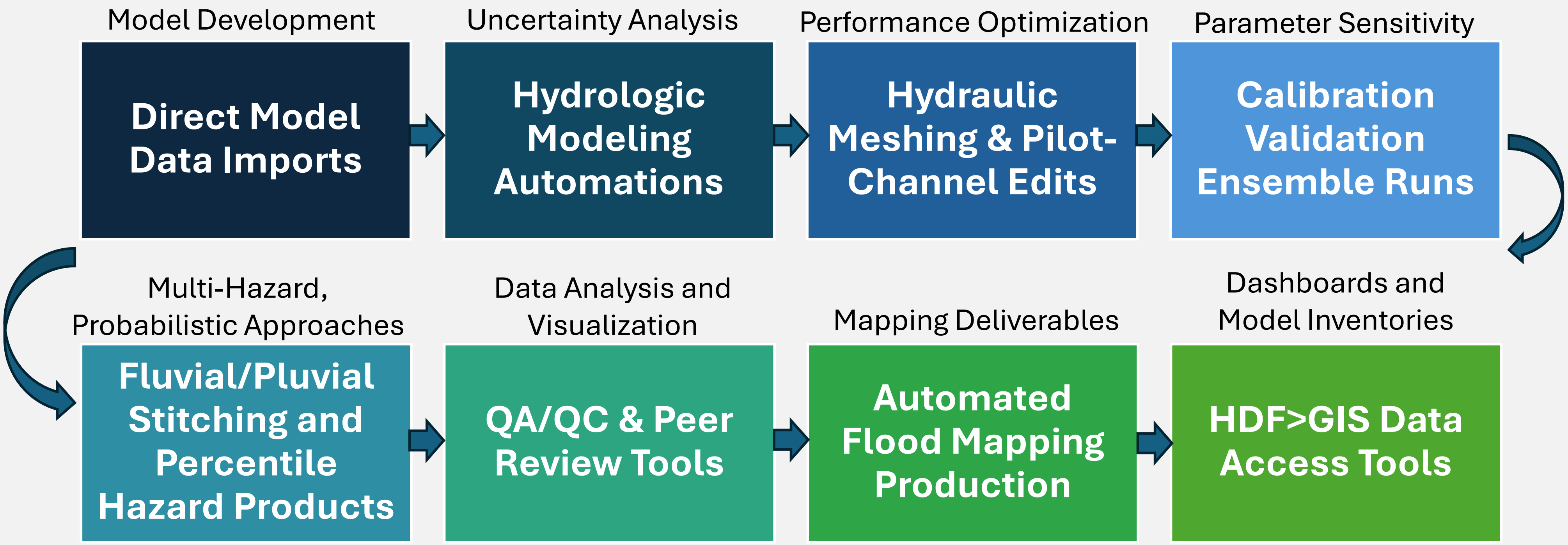
1. **Initial Conditions check:** Once Initial Conditions are written to HDF (unsteady calculations started), wait for file to become idle.
2. **Early termination:** If HDF file hasn't changed for the idle timeout period, preprocessing terminates (HEC-RAS is in warmup/first timesteps).
3. **Fallback to /Results:** If early termination doesn't trigger, wait for /Results group (ensures ALL computational data is complete).

Note: Large event condition files may take longer to load. The default 60s idle timeout works for most models.

☑ Overwrite existing files

Essential for Cloud Execution on TITAN High Performance Computing System

RAS Commander Integration Opportunities at Every Phase of Your Modeling and Mapping Efforts



Model Development

Direct Model Data Imports

Uncertainty Analysis

Hydrologic Modeling Automations

Performance Optimization

Hydraulic Meshing & Pilot-Channel Edits

Parameter Sensitivity

Calibration Validation Ensemble Runs

Multi-Hazard, Probabilistic Approaches

Fluvial/Pluvial Stitching and Percentile Hazard Products

Data Analysis and Visualization

QA/QC & Peer Review Tools

Mapping Deliverables

Automated Flood Mapping Production

Dashboards and Model Inventories

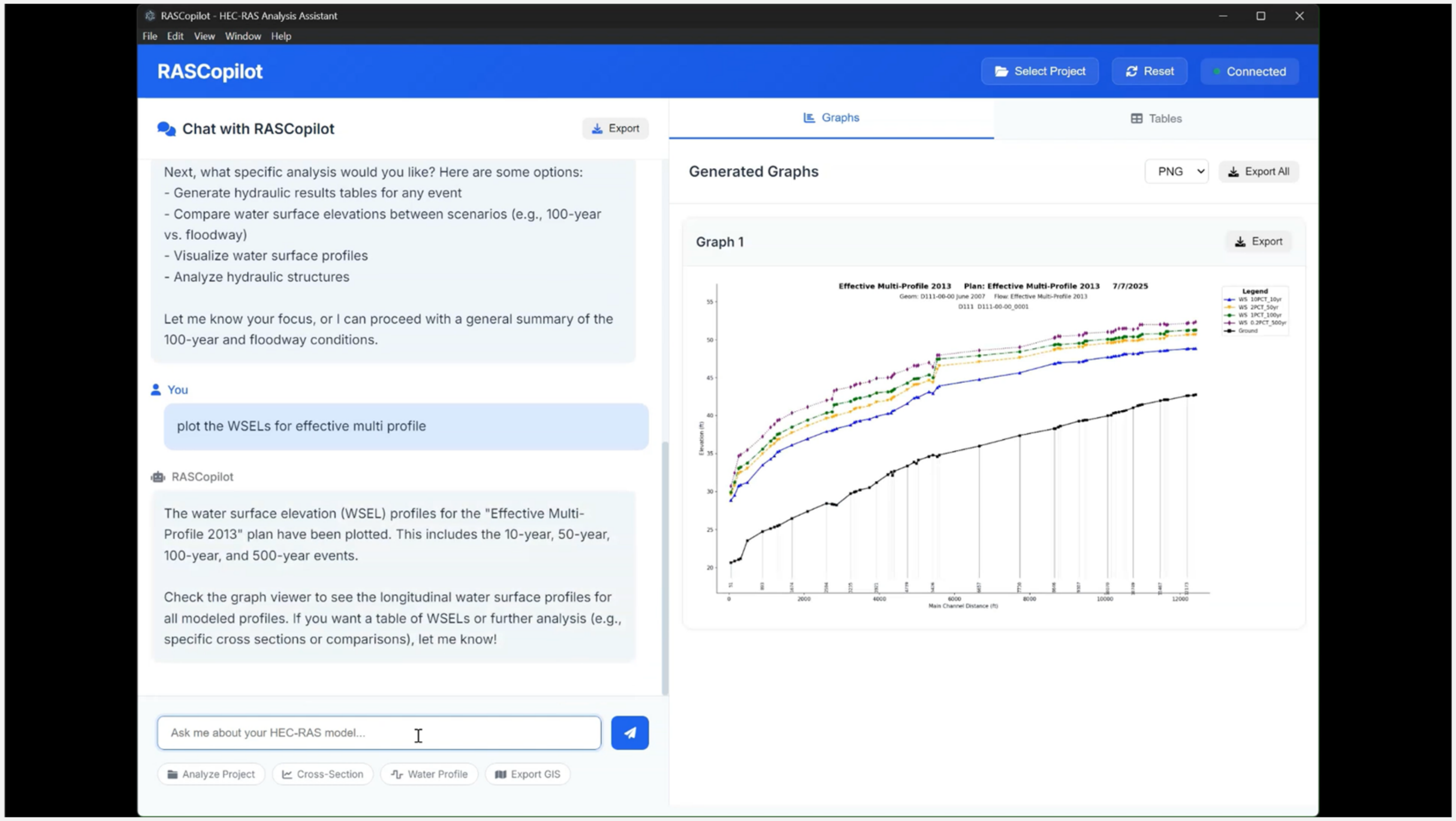
HDF>GIS Data Access Tools

Adding value to your innovative H&H projects and proposals

LLM Forward in Action: RAS Copilot

Ajith Sunarraj
Author, RASCopilot.com

Independent project to build an LLM-powered chat interface that leverages the hecrascontroller library using a streamlit interface with plotting and mapping capabilities



Next, what specific analysis would you like? Here are some options:
- Generate hydraulic results tables for any event
- Compare water surface elevations between scenarios (e.g., 100-year vs. floodway)
- Visualize water surface profiles
- Analyze hydraulic structures

Let me know your focus, or I can proceed with a general summary of the 100-year and floodway conditions.

plot the WSELs for effective multi profile

The water surface elevation (WSEL) profiles for the "Effective Multi-Profile 2013" plan have been plotted. This includes the 10-year, 50-year, 100-year, and 500-year events.

Check the graph viewer to see the longitudinal water surface profiles for all modeled profiles. If you want a table of WSELs or further analysis (e.g., specific cross sections or comparisons), let me know!

Ask me about your HEC-RAS model...

Links and Audience Questions

My Bio:	EngineeringWithLLMs.info
CLB Engineering:	CLBEngineering.com
Github Repositories:	
HEC-Commander	gpt-cmdr/HEC-Commander
RAS Commander	gpt-cmdr/RAS-Commander
Arc Hydro Tools	gpt-cmdr/RAS-Commander-Hydro
HEC-RAS MCP	gpt-cmdr/RAS-Commander-mcp
Linkedin:	William (Bill) Katzenmeyer, P.E., C.F.M.

CLB ENGINEERING CORPORATION