

Mach Cutoff Simulation - One-Page App Summary

Evidence source: repository files only (README and mach_cutoff modules).

What It Is

Mach Cutoff Simulation is an experimental Python package for simulating supersonic sonic-boom ray propagation with time-varying HRRR atmosphere data. It evaluates whether emitted shock rays intersect the ground or cut off along a waypoint-defined route.

Who It Is For

Primary user/persona: Not found in repo. Closest evidence suggests research and engineering users running sonic-boom propagation experiments.

What It Does

- Loads waypoint JSON (lat/lon/alt/time) and interpolates the flight path over time.
- Retrieves HRRR pressure-level snapshots for emission times, with local caching and S3/HTTP retrieval paths.
- Computes sound speed, projected wind component, and effective sound speed from atmospheric state.
- Builds a point-mass supersonic aircraft model and emits cone-distributed shock directions per emission time.
- Ray-traces each emission with adaptive RK4 integration and checks terrain/ground intersections.
- Writes summary and hit artifacts, then renders Matplotlib, Plotly, and PyVista outputs.

How It Works (Architecture)

- `mach_cutoff/cli.py` parses args and loads JSON config via `mach_cutoff/config.py`.
- `MachCutoffSimulator` in `mach_cutoff/simulation/engine.py` orchestrates the run loop.
- `HRRRDatasetManager` and `HRRRInterpolator` fetch and sample atmospheric snapshots.
- `build_acoustic_grid_field` builds refractive-index fields; `integrate_ray` propagates rays.
- `SimulationResult` serializes JSON/NPZ and feeds visualization backends.

Data flow: waypoint JSON + config -> simulator -> HRRR snapshots/interpolation -> acoustic field -> ray integration -> SimulationResult -> JSON/NPZ + visual outputs.

How To Run (Minimal)

```
python3 -m venv .venv && source .venv/bin/activate
python3 -m pip install --upgrade pip && python3 -m pip install -e '.[all]'
mach-cutoff --waypoints examples/waypoints_example.json --config examples/config_example.json
--output-dir outputs
```

Expected outputs: `outputs/simulation_summary.json`, `outputs/simulation_hits.npz`, and enabled backend artifacts.