AI workflows for AthenaK and beyond

Code Development | Environment Debugging | Job Monitoring

Princeton gravity Initiative

Hengrui Zhu

Princeton Gravity Initiative

AthenaK Summer School @ Penn State

Three sectors of use (for me)

1. Code Development

- TDE
- Dynamical-spacetime radiation
- IrisK

2. Environment Setup/Debug

- Delta/NCSA patching

3. Job Monitoring

- BBH accretion excision tests

Sector 01

Code Development

with increasing complexity

Princeton gravity Initiative

01

TDE

1. pre-existing formulation
2. existing Z4c solver

→

02

Dynamical Spacetime Boltzmann Radiation

1. no pre-existing production ready formulation
2. HARM-like solver can be reused

→

03

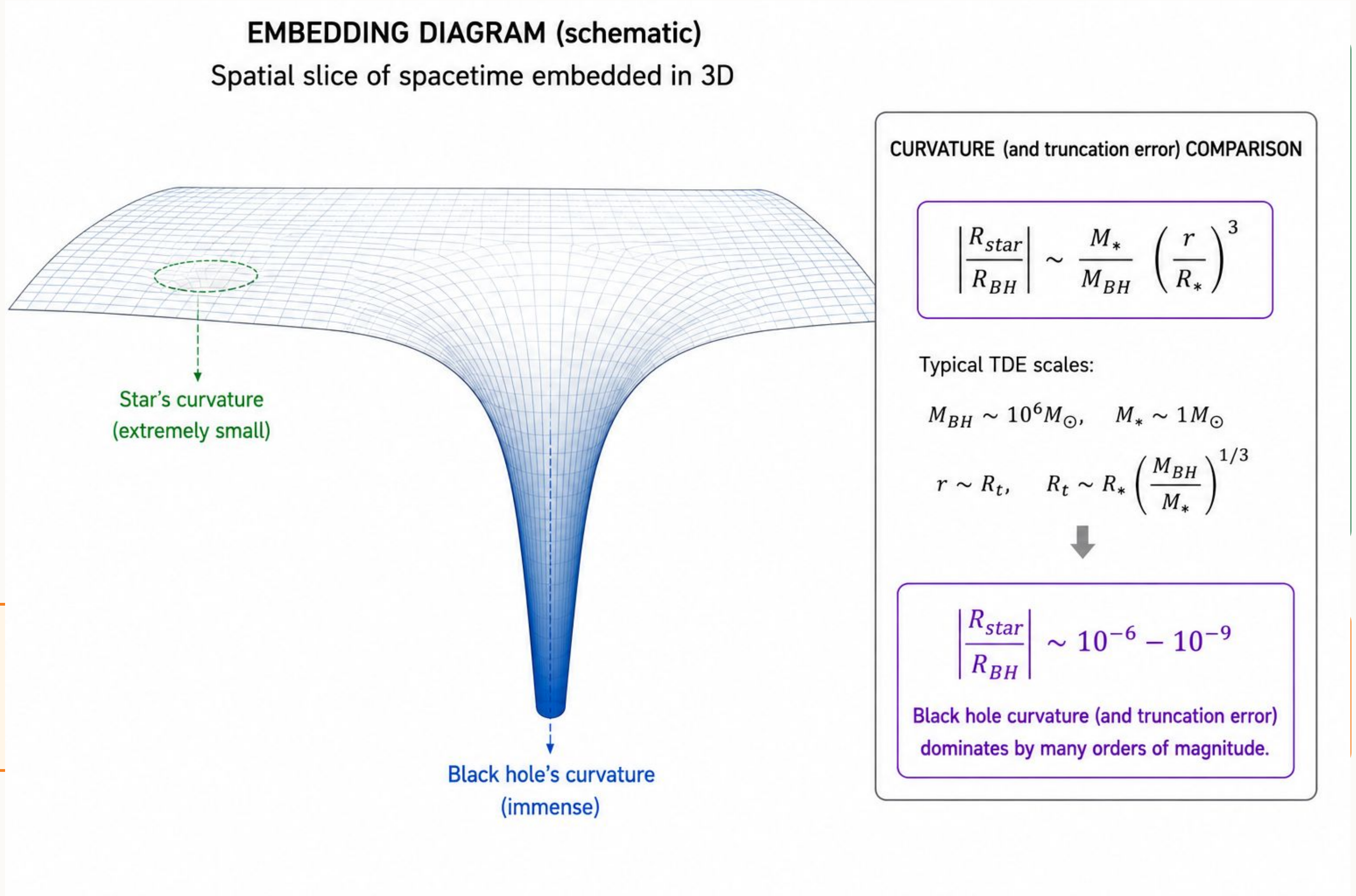
IrisK - Pseudo-Spectral Black Hole Evolution

1. ultra-long horizon task (>100k lines of code already!)
2. existing algorithm but very complicated
3. Coding from scratch

TDE: residual Z4c evolution on a stationary background

THE DIFFICULTY

Truncation error in the background dominates over the self-gravity of the secondary for Extreme Mass Ratio systems.



Background-error-subtraction lineage: East & Pretorius 2013; TDE application: East 2014; BSSN/Z4c context: Shibata & Nakamura 1995, Baumgarte & Shapiro 1999.

CASE STUDY

TDE: residual Z4c evolution on a stationary background

THE DIFFICULTY

Truncation error in the background dominates over the self-gravity of the secondary for Extreme Mass Ratio systems.

THE SOLUTION

- Factor out the background truncation error! (East & Pretorius, generalized harmonic)
- Lam, Shibata, Kiuchi scheme: Subtract at BSSN equation level.

Z4C GENERALIZATION

- Generalization to Z4c is mathematically trivial.
- Implementation is tedious, requiring ~1 month of human time.

$g = g_{bg} + g_{residual}$ **evolve the perturbation without forgetting that the sum is the physical field**

Background-error-subtraction lineage: East & Pretorius 2013; TDE application: East 2014; BSSN/Z4c context: Shibata & Nakamura 1995, Baumgarte & Shapiro 1999.

Princeton gravity Initiative

TDE: Human Input (through prompt)

WELL-DEFINED TESTS

Clear passing criteria used to establish baseline stability:

- Convergence & strong stability Minkowski evolution
- Linear gravitational wave propagation
- TOV and boosted TOV evolution with AMR

OTHER HUMAN INPUT

Additional context, guidelines, and resource updates:

- Equations and paper references
- Task list insertion and re-ordering
- allocate new DvceArray5D for residual field and full field

TDE: Human Input (through prompt)

Princeton gravity Initiative

WELL-DEFINED TESTS

Clear passing criteria used to establish baseline stability:

- Convergence & strong stability Minkowski evolution
- Linear gravitational wave propagation
- TOV and boosted TOV evolution with AMR

OTHER HUMAN INPUT

Additional context, guidelines, and resource updates:

- Equations and paper references
- Task list insertion and re-ordering
- allocate new DvceArray5D for residual field and full field

DO NOT WRITE YOUR OWN PROMPT FROM SCRATCH!!! ITERATE IDEA WITH AI AND LET IT BUILD A PROMPT FOR YOU

The core idea

Princeton gravity Initiative

AI is most useful for Test-Driven Development

1. Human Design

- **Physics Formulation**
  Defining scientific intent
- **Pre-designed Tests**
  Test-Driven Development (TDD) criteria

2. AI Inner Loop (TDD)

Closed, Idealized, Reproducible Sandbox

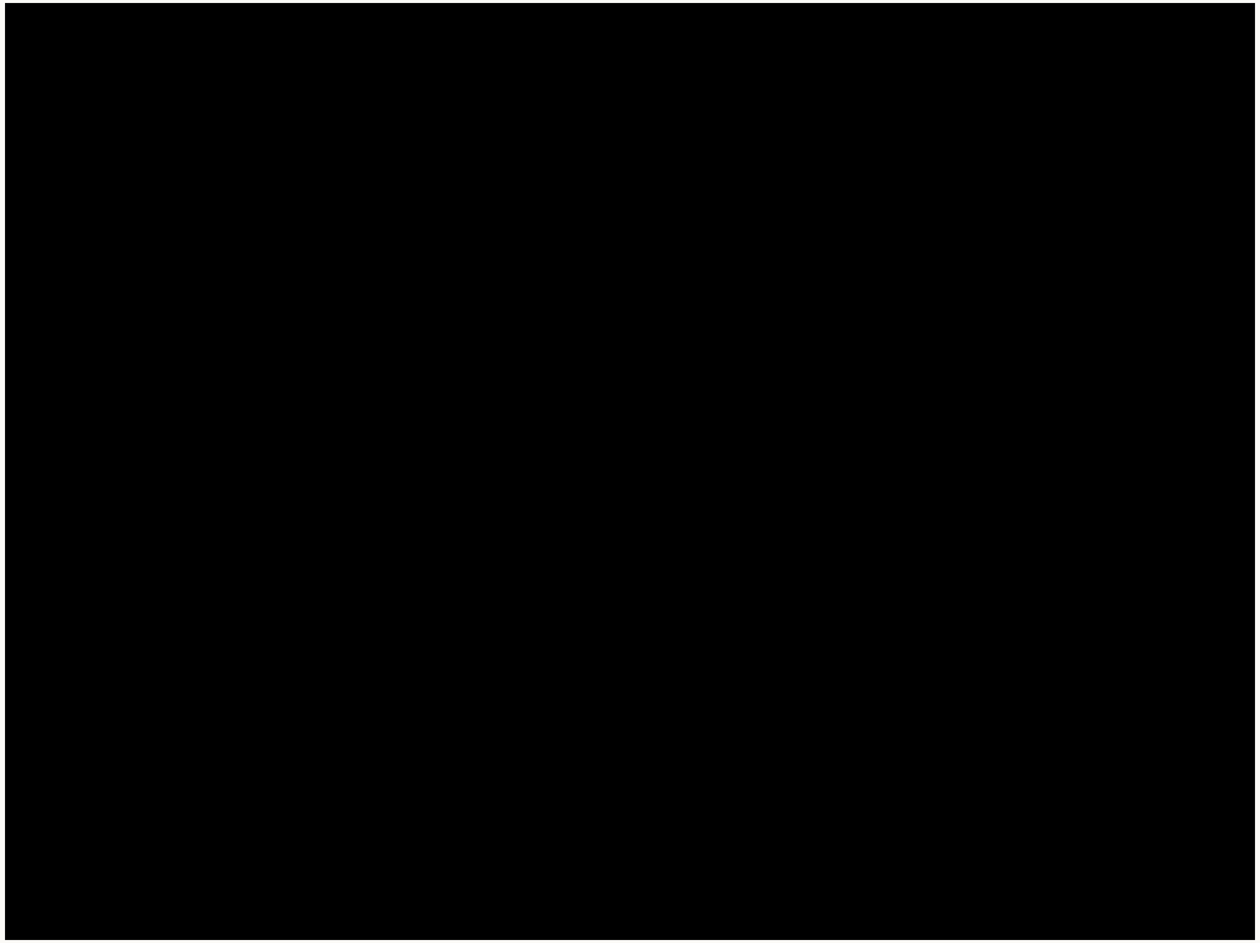
Agentic Implementation → Compile & Run Tests → Results Verification → Agentic Implementation

3. Human Judgment

- **Final Approval**
  Assessing acceptable risk
- **Physical Alignment**
  Ensuring physics correctness

The human contribution shifts upward: from typing every line to defining correctness, tests, and acceptable risk.

TDE: after 3 hours of iteration with codex …



Radiation transport on dynamical spacetime

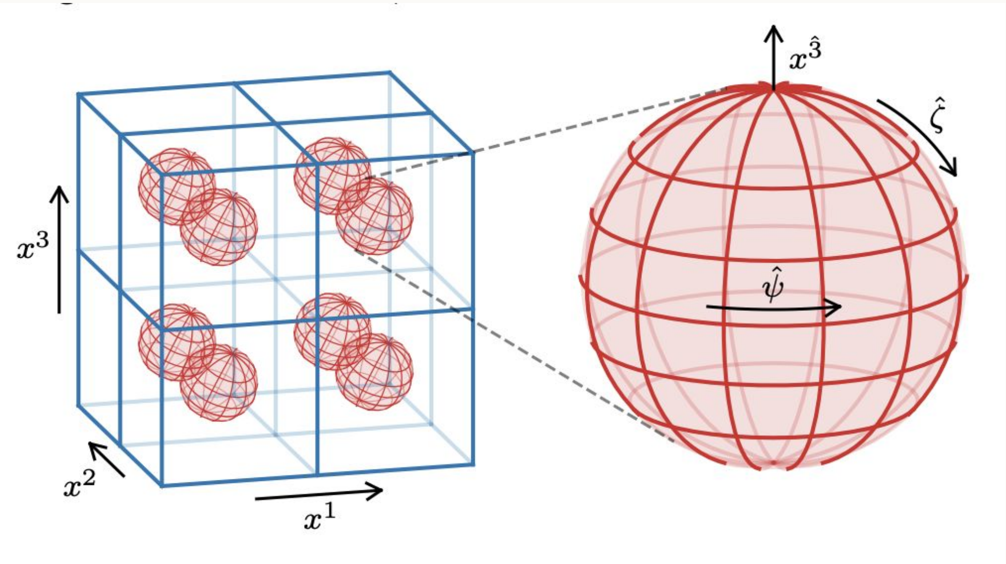
This was more substantial: not building from scratch, but no stable pre-existing production scheme for the target problem.

PRE-EXISTING PIECES

- HARM-style GRMHD solver context
- matter coupling ready to port from HARM path
- finite-solid-angle transport infrastructure already in AthenaK

DIFFICULTY

- time-dependent tetrad frame
- HARM → Valencia reformulation
- New tests required



Context references: Gammie, McKinney & Tóth 2003 for HARM; Banyuls et al. 1997 for Valencia formulation.

Princeton gravity Initiative

The bottleneck: a smooth time-dependent tetrad

The finite-solid-angle method needs local orthonormal frames, but the spacetime is evolving.

NAÏVE DANGER

- Gram–Schmidt-like orthonormalization can introduce ordering choices and non-smooth behavior.
- Hidden branch choices are risky near strong fields, coordinate features, or GPU execution.
- A stable run can still encode the wrong transport operator.

TARGET PROPERTY

- algebraic dependence on metric and derivatives
- smooth wherever the 3+1 metric is regular
- no iterative orthonormalization step
- clear identities for tests: $e^a\_\mu\ e^b\_\nu\ g\hat{}\{\mu\nu\} = \eta^{ab}$

AI iteration helped identify a Cholesky-inspired construction for the tetrad: linear algebra as a route to a smoother physics implementation.

CASE STUDY

Pre-Existing Static Spacetime Tests

TEST SUITE

01. Beam Crossing

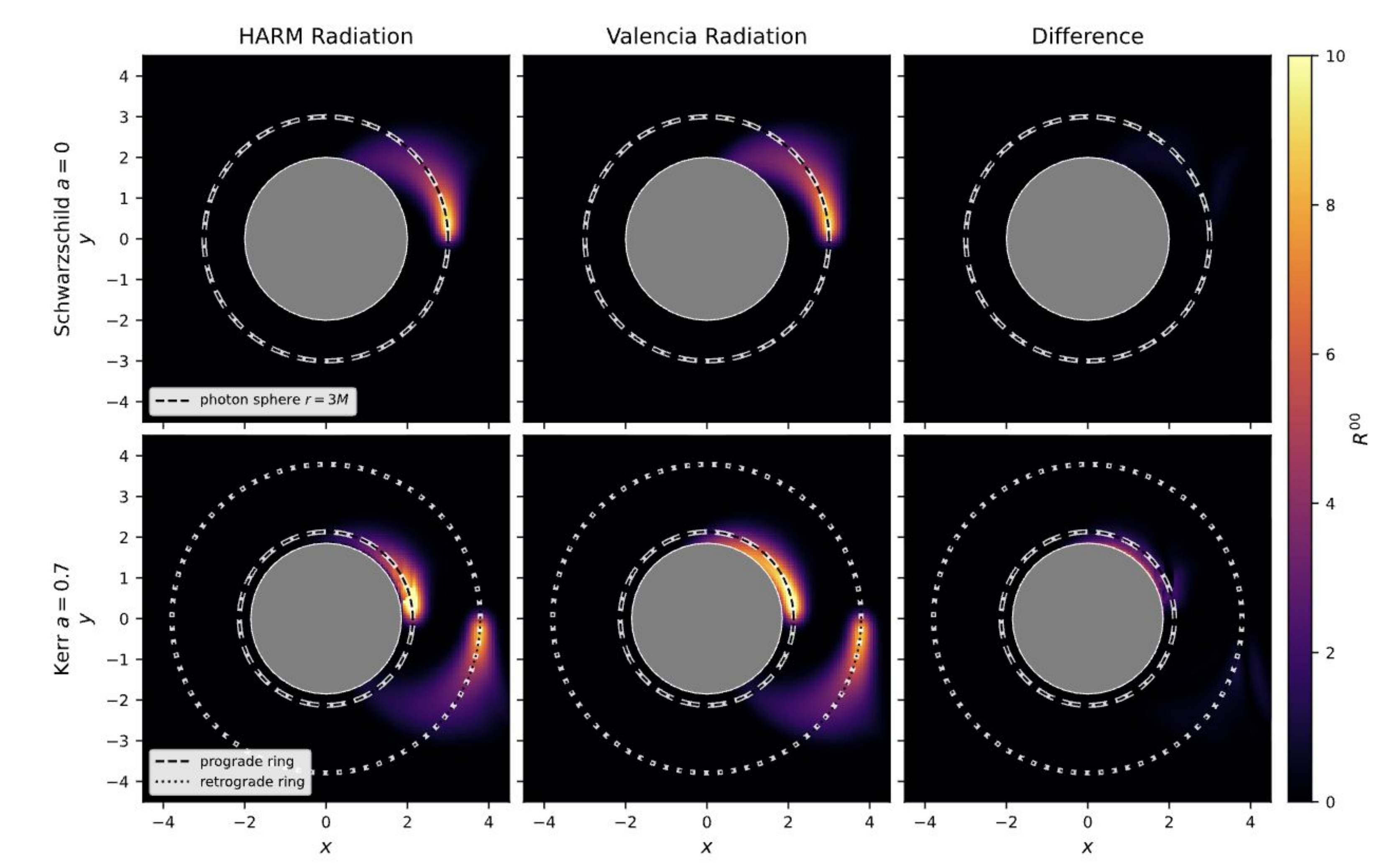
Verifies propagation consistency across overlapping paths.

02. Matter Coupling

Ensures correct energy-momentum coupling dynamics.

03. Kerr Beam Tests

Validates beam behaviors in stable, curved spacetime metric.



Design New Tests for Dynamical Spacetime

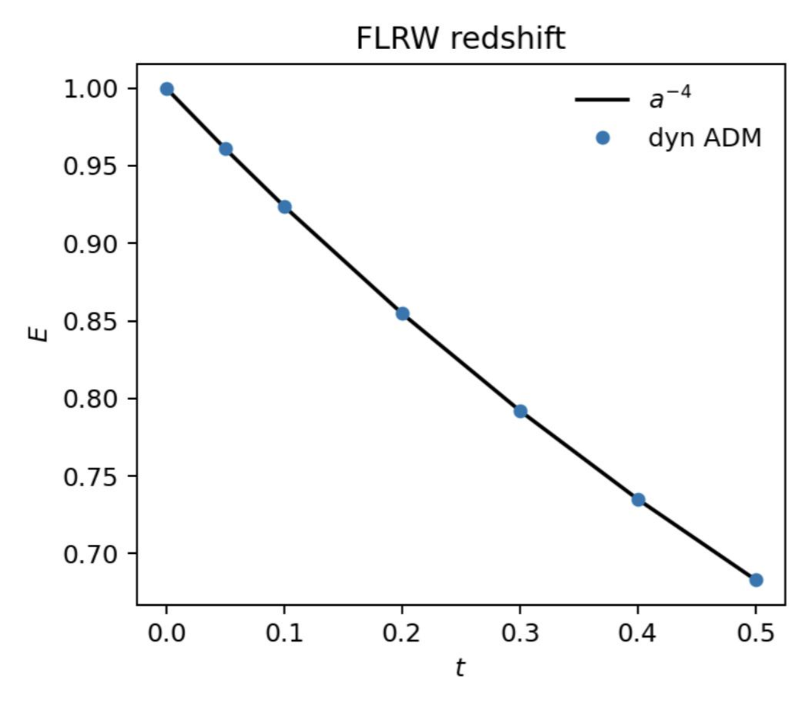
TEST SUITE

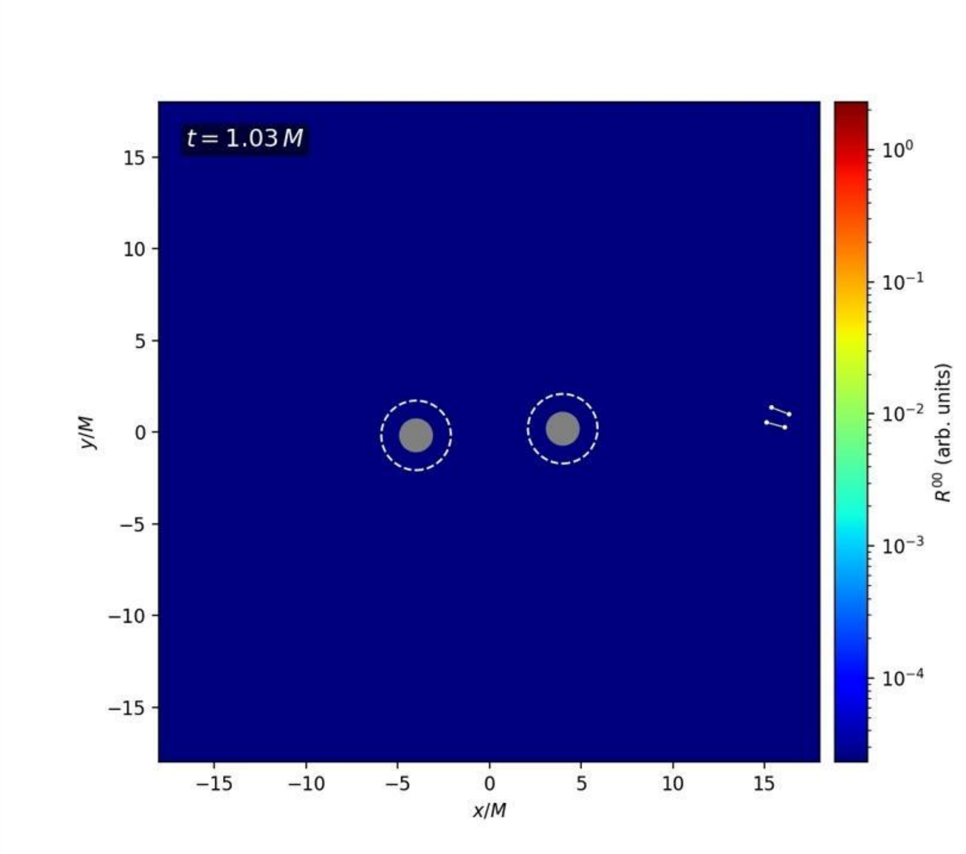
01. Redshift in FLRW

analytically known solution, though isotropic

02. Binary Black Hole Beam

Verifies radiation propagate along null geodesics on time-dependent background, up-to numerical diffusion





The core idea

1. Joint Formulation

• AI Brainstorming

Leverages inter-disciplinary knowledge for physics formulation & test suites.

• Human Cross-Check

Crucial to verify schemes & correct AI-generated physical errors.

2. TDD Loop

Formulation results drive test implementation & code generation.

The core idea

1. Joint Formulation

- **AI Brainstorming**

  Leverages inter-disciplinary knowledge for physics formulation & test suites

- **Human Cross-Check**

  Crucial to verify schemes & correct AI-generated physical errors

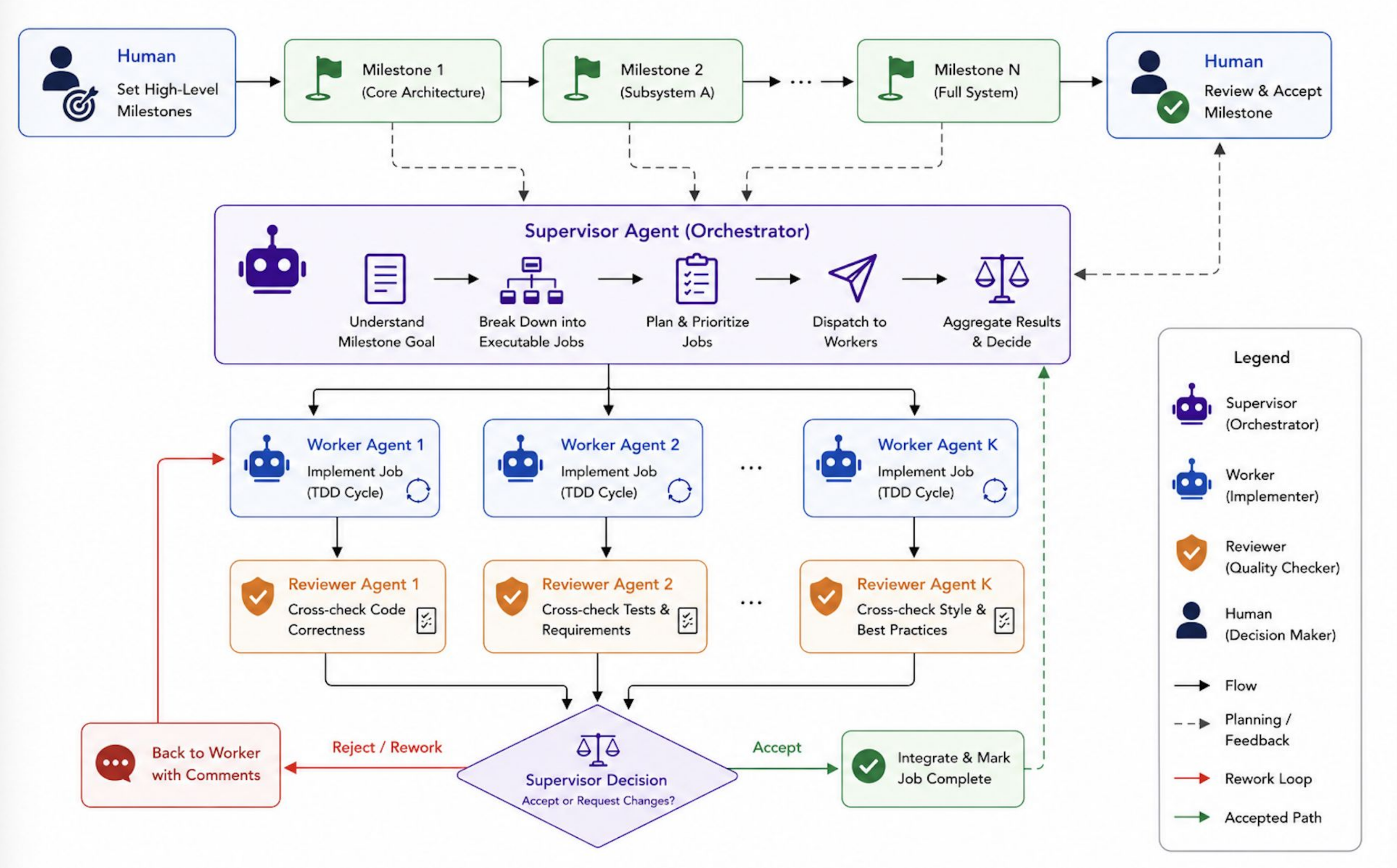
2. TDD Loop

Formulation results drive test implementation & code

AI WILL HALLUCINATE!!! TRIPLE CHECK FORMULATION AND TEST DESIGN BEFORE PASSING TO TDD LOOP

CASE STUDY

IrisK: Mitigating Drift in Long-Horizon Code Gen



Open-source project here:

https://github.com/HengruiZhu99/ai-supervisor-worker-workflow

02

Environment setup and debugging

AI for compilers, modules, machine-specific patches, and portability

Princeton gravity Initiative

AI-Assisted HPC Build with codex-cli

1. CLI Install & Auth

Install the codex-cli binary directly onto your cluster. No Node.js or npm required.

```
# Standalone bootstrap
curl -fsSL https://chatgpt.com/codex/install.sh
| sh

# Authenticate agent
codex login
```

2. Point Agent to Build Docs

Instruct the agent by providing the target repository, documentation context, and configurations.

AthenaK Repository:
https://github.com/IAS-Astrophysics/athenak

Machine Specific Notes:
https://ias-astrophysics.github.io/athenak-docs/...

Cluster Guide (e.g., Perlmutter):
https://docs.nersc.gov/systems/perlmutter/...

3. Auto-Build & Env Cache

Loads modules, compiles problem generator (pgen), and caches the environment.

Prompt Request:
"Configure for CPU or GPU, compile the chosen pgen, and save environment to ~/athenak.env"

```
# Restore env instantly later
source ~/athenak.env
```

USER PROMPT INPUT

"I would like to clone and build athenak here. Here is the repo's url: https://github.com/IAS-Astrophysics/athenak, and this is the instruction page for running jobs on the cluster: https://docs.alcf.anl.gov/polaris/running-jobs/. I would like to configure and compile the code for **gpu (A100) with MPI***. And here's some instruction from athenak to build on different machines: https://ias-astrophysics.github.io/athenak-docs/notes_specific_machines.html. I would like to build the* **fluids/gr_torus pgen***. And please set up the correct environment for* **multi-node, multi-gpu runs***. Please give me also an environment file so that I can set it up later myself just by running source ~/athenak.env"* **Do not use more than 8 threads while building on the login node.** Write me a job script for submitting to the QUEUE_NAME with N_nodes and N_GPU per node with INPUTFILE.athinput

03

Running and Monitoring jobs

AI as an operations assistant for expensive simulations

Princeton gravity Initiative

Job monitoring: stability tuning/termination on failure

Example usage: excision and stability tuning for accretion flow onto a binary black hole.

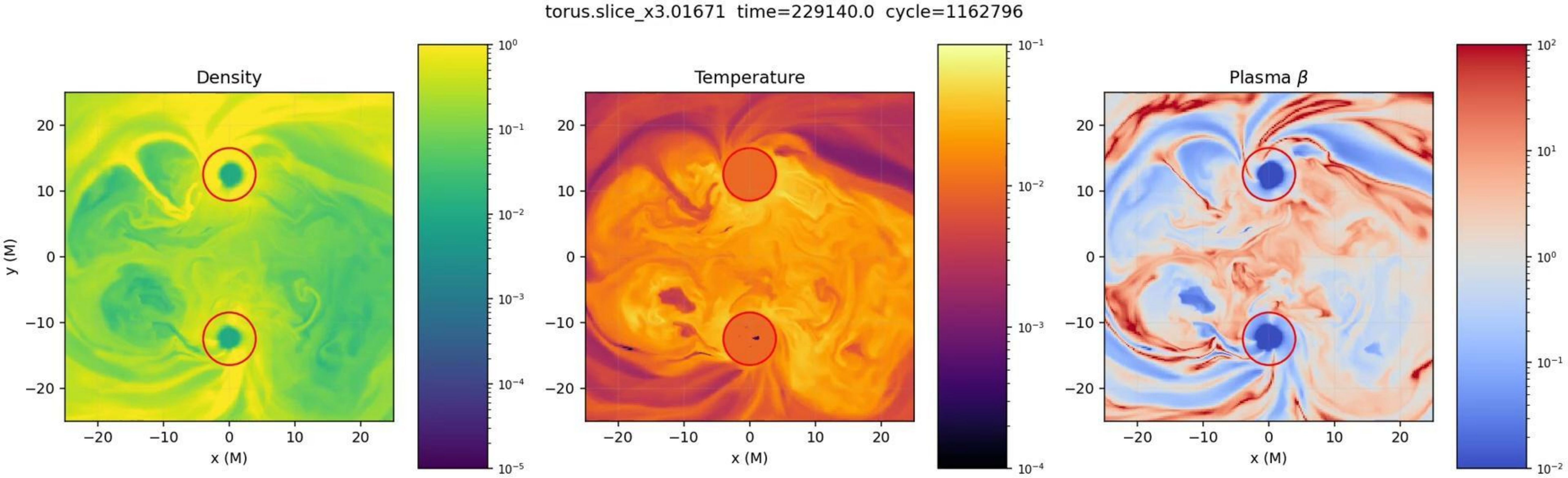
AI CAN MONITOR

horizon / excision margin
constraint norms and gauge quantities
mass, energy, angular-momentum drift
density, pressure, radiation positivity
crash location, timestep, and rank/node patterns

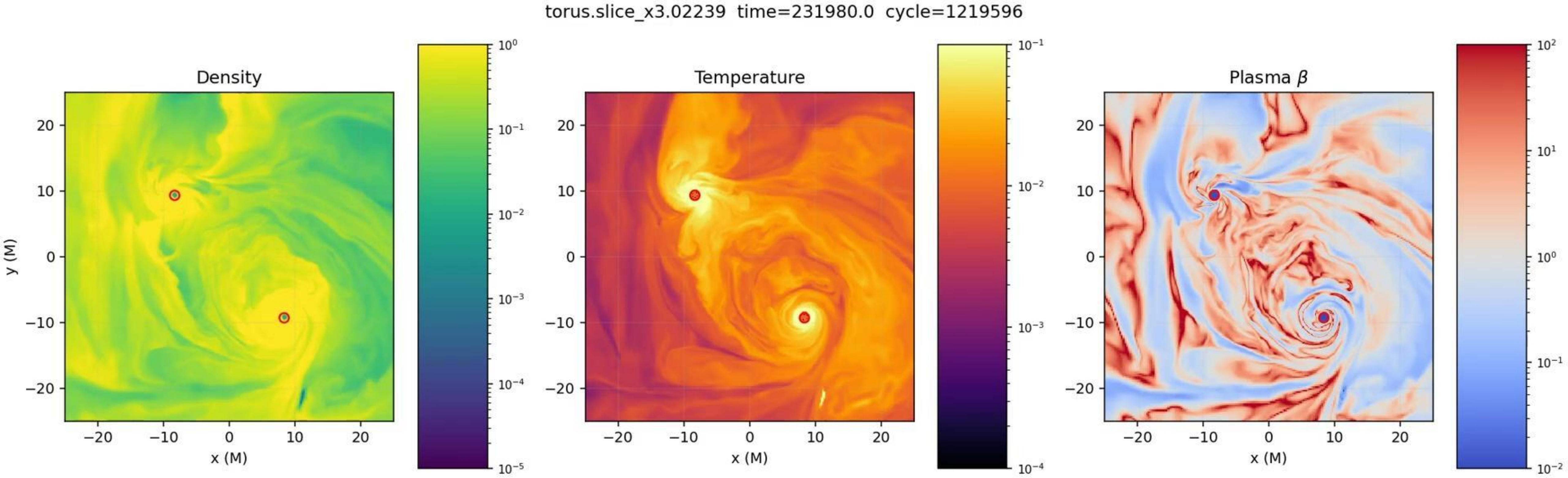
HUMAN DECIDES

which parameter changes preserve the intended physical problem
whether stability is numerical robustness or unphysical damping
when to fallback, rerun, or abandon a parameter region
what diagnostic evidence is sufficient

Job monitoring: stability tuning/termination on failure



Job monitoring: stability tuning/termination on failure



Thank You!

Contact Info: hz0693@princeton.edu