Multi-Agent Improvement Cycle

Fusion Heat Transport PDE Benchmark

Sequential Agent Pipeline: Pareto Analysis, Bottleneck Detection,
Proposal Generation & Multi-Perspective Evaluation
with PHYSBO Bayesian Optimization

Problem & Motivation

Challenge

- 8 numerical solvers (FDM, FEM, FVM, Spectral, PINN)
- Each solver has different accuracy/speed/stability
- Performance varies across α, dt, nr, IC type
- Manual tuning is tedious and error-prone

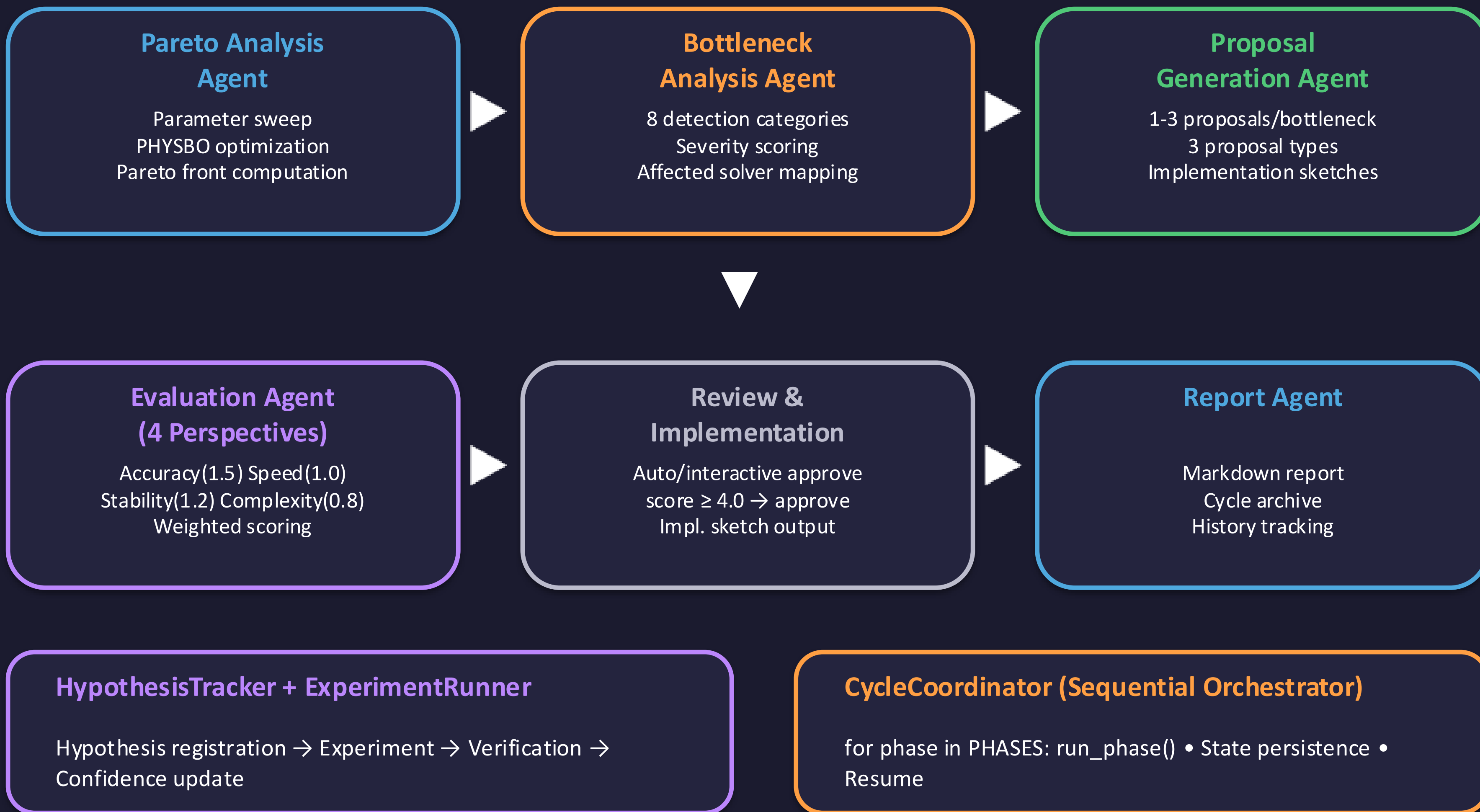
Solution: Multi-Agent Automation

- Automated Pareto analysis across parameters
- Systematic bottleneck detection (8 categories)
- AI-generated improvement proposals
- Multi-perspective evaluation (4 views)

$$\text{PDE: } \partial T/\partial t = (1/r)\ \partial/\partial r\ (r\ \chi(|\partial T/\partial r|)\ \partial T/\partial r)$$

$$\chi(|T'|) = (|T'| - 0.5)\text{^}\alpha + 0.1\ \ (|T'| > 0.5),\ \text{else}\ \chi = 0.1$$

Agent Architecture (Sequential Pipeline)



Phase 1: Pareto Analysis

Per-Solver Analysis

- Parameter sweep: α=[0.0, 0.5, 1.0]
- dt=[0.001, 0.0005], nr=61 (fixed)
- PHYSBO or grid-based dt exploration
- Stability check + L2/L∞ error computation
- Pareto rank assignment (0=optimal)

Cross-Solver Comparison

- Rankings per problem setting
- Win counts (accuracy, speed, Pareto)
- Coverage gap detection
- Overall solver rankings

PHYSBO Bayesian Optimization Integration

```
# Feature: log10(dt) (1-dim, 80 discrete candidates)
# Objectives: -L2_error, -wall_time (2-objective maximization)
# Discrete multi-objective optimization per (alpha, ic_type):

dt_candidates = np.logspace(-5, -2, 80)  # 80 log-spaced
test_X = np.log10(dt_candidates).reshape(-1, 1)
policy = physbo.search.discrete_multi.Policy(
    test_X=test_X, num_objectives=2)  # discrete search
policy.random_search(max_num_probes=5)    # random phase
policy.bayes_search(max_num_probes=15, score='HVPI')  # Bayesian
```

Phase 2: Bottleneck Detection

8 Detection Categories

`stability`	LOW	Solver stability rate < 90%
`accuracy_gap`	MED	Large L2 error differences between solvers
`speed_gap`	LOW	Execution time ratio > 100×
`coverage_gap`	MED	Single solver dominates Pareto front (>80%)
`no_stable_solver`	HIGH	No solver stable for a problem setting
`solver_dominance`	MED	One solver wins >80% of problems
`cross_accuracy_gap`	HIGH	Best solver still has L2 > 0.5
`solver_instability`	MED	Solver fails on >20% of problems

Each bottleneck includes: severity, affected solvers, evidence dict, and suggested actions

Phase 3-4: Proposal & Multi-Perspective Evaluation

Phase 3: Proposal Generation

`parameter_tuning`	dt constraints, nr adjustment
`algorithm_tweak`	Adaptive time-stepping, lagged coefficients
`new_solver`	IMEX for stiff problems, hybrid methods

Phase 4: 4-Perspective Evaluation

Accuracy (1.5)	Resolution, adaptive methods
Speed (1.0)	Optimization, vectorization
Stability (1.2)	Stability-focused proposals
Complexity (0.8)	Simpler = better (param > algo > new)

4 Perspectives run sequentially (for loop). Parallelizable via concurrent.futures if extended to heavy tasks.

$$\text{overall} = \Sigma(\text{weight\_i} \times \text{score\_i}) / \Sigma(\text{weight\_i}) \quad \text{approve} \geq 4.0 \;|\; \text{consider} \geq 3.0 \;|\; \text{reject} < 3.0$$

Example: Multi-Agent Evaluation Scores (Cycle 1)

Proposal	Accuracy	Speed	Stability	Complex.	Overall	Rec.
P001: Adaptive stepping	4.0	3.0	4.5	2.5	3.38	consider
P002: dt constraints	3.5	4.5	4.0	4.5	3.81	consider
P003: Vectorize FVM	3.0	5.0	3.0	3.5	3.38	consider

Phase 5-7: Review → Implementation → Report

Phase 5: Review

Auto: score ≥ 4.0 → approve

Interactive: Y/n/q per proposal

Status → approved / rejected

Phase 6: Implement

param_tuning: preview sketch

algo_tweak: manual guidance

new_solver: code template

Phase 7: Report

Markdown cycle report

cycle_NNN_datetime.md

History JSON updated

Report Contents

- Executive summary (solver count, bottleneck count, proposals, approved)
- Cross-solver analysis: rankings, win counts, per-problem results
- Per-solver Pareto analysis: stability rates, error/time ranges
- Identified bottlenecks with severity and suggested actions
- Proposals with rationale and multi-perspective evaluation scores
- Next cycle recommendations

PHYSBO: Bayesian Multi-Objective Optimization

Feature: log10(dt) [80 discrete candidates]   Objectives: -L2_error, -wall_time

	Grid Search	PHYSBO
Feature	—	log10(dt) (1-dim)
Objectives	L2, wall_time	-L2, -wall_time (2-obj max)
Evaluations	All dt × all problems	20/problem (5+15)
Search space	Fixed grid	80 log-spaced candidates
Strategy	Exhaustive	HVPI-guided exploration
Requirement	Always available	physbo package needed

Per (alpha, ic_type) pair:

80 dt candidates → 5 random probes → 15 HVPI-guided probes → best stable point

Fallback: auto-detect physbo availability; grid search if not installed

Results: Cross-Solver Rankings

Tutorial Run Output (α = 0.0, 0.5, 1.0)

Rank	Solver	Avg Rank	Stability	Min L2 Error
#1	implicit_fdm	1.3	100%	0.054
#2	cell_centered_fvm	2.0	100%	0.058
#3	compact4_fdm	2.7	100%	0.115
#4	p2_fem	3.5	100%	0.108
#5	imex_fdm	4.8	100%	0.492
#6	cosine_spectral	5.2	67%	varies
#7	pinn_stub	6.5	—	—
#8	chebyshev_spectral	7.0	—	—

implicit_fdm consistently ranks #1 across all α values

Reference: Implicit FDM with 4× grid refinement (nr×4, dt/4)

Results: Per-Solver Pareto Analysis

Solver	Total	Stable	Pareto-Opt	Min Error	Max Error
implicit_fdm	12	12	5	0.054	0.296
cell_centered_fvm	12	12	5	0.058	0.287
compact4_fdm	12	12	4	0.115	0.489
p2_fem	12	12	5	0.045	0.531
imex_fdm	12	12	3	0.492	1.203
cosine_spectral	12	8	3	0.089	0.812
chebyshev_spectral	12	4	2	0.799	1.532
pinn_stub	12	0	0	—	—

- FDM/FVM solvers: 100% stability, competitive error
- cosine_spectral: dt-sensitive stability (67%), best at low α
- p2_fem: good accuracy but 10-100× slower

Results: Bottlenecks & Proposals

Detected Bottlenecks (Cycle 1)

`speed_gap`	LOW	fastest=2.98ms (cell_centered_fvm), slowest=2361.36ms → 792× gap
`stability`	MED	cosine_spectral: 67% stability ($\alpha \geq 0.5$ instability)

Generated Proposals

ID	Type	Title	Score	Recommendation
P001	algorithm_tweak	Adaptive time-stepping for cosine_spectral	3.38	consider
P002	parameter_tuning	Constrain dt for high-alpha problems	3.81	consider
P003	parameter_tuning	Optimize FVM vectorization	3.38	consider

All proposals scored 3.0-3.9 ("consider" range) — no auto-approved proposals in cycle 1

Hypothesis-Driven Workflow

HypothesisTracker

- Register hypotheses with unique IDs
- Track verification history per hypothesis
- Auto-update status & confidence (0-1)
- JSON persistence for cross-session use

ExperimentRunner

- Execute solver experiments with configs
- Compute reference (4× refined ImplicitFDM)
- Record L2/L∞ error, time, stability
- Append results to CSV database

untested → experiment → verified → confirmed / rejected / inconclusive

Predefined Experiments

`stability_map`	Sweep α×dt space (8 alphas, 5 dts)
`ic_comparison`	Compare across initial conditions
`pinn_comparison`	PINN variants vs FDM/Spectral
`linear_regime`	Test in purely linear regime (|dT/dr|<0.5)
`fine_sweep`	Exhaustive sweep (9α × 5nr × 3dt × 3t_end)

Hypothesis Verification: Real Examples

H1: "Smaller dt improves spectral solver stability"

CONFIRMED

dt	0.0001	0.0002	0.0005	0.001	0.002
Stability	100%	100%	100%	50%	0%

Confidence: 1.0 | 6+ verification attempts across cycles

H_compact4_best: "Compact4 FDM beats implicit FDM at high α"

CONFIRMED

- α=1.0: compact4_fdm L2=0.115 vs implicit_fdm L2=0.296 (compact4 wins)
- α=1.5: compact4_fdm L2=0.189 vs implicit_fdm L2=0.450 (compact4 wins)
- 4th-order spatial accuracy advantage grows with nonlinearity

Improvement Cycle: Sequential 7-Phase Pipeline

1	**Pareto Analysis**	Parameter sweep + PHYSBO → Pareto fronts
▼		
2	**Bottleneck Detection**	8 categories, severity scoring
▼		
3	**Proposal Generation**	1-3 proposals per bottleneck
▼		
4	**Multi-Perspective Eval**	4 views, weighted scoring
▼		
5	**Review**	Auto (score≥4.0) or interactive
▼		
6	**Implementation**	Sketches + guidance output
▼		
7	**Report & Archive**	Markdown + history JSON

```
python docs/analysis/method_improvement_cycle.py --cycles 3 --auto
```

Scoring Mechanism: Keyword-Based Rule Engine

Each Perspective: default score=3.0, then if/else keyword matching on proposal.title and proposal.proposal_type

AccuracyPerspective (weight=1.5)

title ∈ "resolution"|"accuracy" → 4.5

title ∈ "adaptive" → 4.0

type == parameter_tuning → 3.5

(no match) → 3.0

SpeedPerspective (weight=1.0)

title ∈ "optimize"|"fast" → 4.5

title ∈ "adaptive" → 2.5

title ∈ "resolution"+"increase" → 2.0

(no match) → 3.0

StabilityPerspective (weight=1.2)

title ∈ "stability"|"adaptive" → 5.0

title ∈ "constrain" → 4.5

type == algorithm_tweak → 3.5

(no match) → 3.0

ComplexityPerspective (weight=0.8)

type == parameter_tuning → 4.5

type == algorithm_tweak → 2.5

type == new_solver → 1.5

sketch > 20 lines → score - 1.0

$$\text{overall} = (\text{acc}\times 1.5 + \text{spd}\times 1.0 + \text{stb}\times 1.2 + \text{cpx}\times 0.8) / 4.5$$

Limitation: Scores are determined by keyword string matching, not by understanding proposal content.
If no keyword matches, all 4 perspectives return 3.0 → overall = 3.0 (fixed). Score constants (e.g. 4.5, 2.5) are heuristic.

Advanced Multi-Agent System (Prototype)

docs/analysis/advanced_multi_agent.py — Debate-based architecture with critical review

Statistics Agent
Distribution analysis
Dominance detection

Feature Agent
Tree-based importance
Redundancy detection

Hypothesis Agent
Alpha threshold test
Stiffness hypothesis
Grid vs physics

Critic Agent
Challenges claims
Data diversity check

SynthesisAgent — Aggregates insights + hypotheses + critiques

Generates final report with key findings, confirmed hypotheses, caveats

4-Phase Execution Flow

Phase 1	Initial Analysis	StatisticsAgent + FeatureAgent run independently on training data (X, y)
Phase 2	Hypothesis Testing	HypothesisAgent tests 3 hypotheses: alpha threshold, stiffness, grid vs physics
Phase 3	Critical Review	CriticAgent receives all insights and challenges weak claims
Phase 4	Synthesis	SynthesisAgent merges findings into structured report

Key difference from improvement cycle: CriticAgent can challenge other agents' conclusions.
Still rule-based (no LLM), but demonstrates the debate pattern — a step toward true multi-agent collaboration.

Advanced System: Execution Results (1512 samples)

StatisticsAgent

implicit_fdm wins 85.5% of cases

T_center has zero variance (uninformative)

FeatureAgent

Top feature: t_end (importance=0.28)

13 high-correlation pairs detected (>0.9)

HypothesisAgent: 3 Confirmed Hypotheses

100%	Alpha > 0.5 always leads to FDM selection	FDM wins 99.9% for $\alpha > 0.5$
95%	Spectral only wins when problem_stiffness < 1.90	Max stiffness for spectral wins: 1.9022
80%	Grid params (nr, dt, t_end) more important than physics	Grid importance: 10, Physics importance: 3

CriticAgent: 1 Critique Raised

[minor] Feature 't_end' has only 3 unique values — importance may be inflated

SynthesisAgent: Recommendations

1. Use implicit_fdm as the default solver
2. Consider removing ML selector for this IC ($T_0 = 1 - r^2$)
3. Improve spectral solver stability for threshold-based χ
4. Add more diverse initial conditions to training data

Gap Analysis: What vs Why

Current agents answer WHAT happened, but not WHY it happened.

Agent	What (automated)	Why (missing)
ParetoAnalysis	implicit_fdm is rank #1	Why? (2nd-order implicit → A-stable)
Bottleneck	Speed gap: 792×	Why? (P2 FEM: matrix assembly cost)
Hypothesis	$\alpha>0.5$ → FDM wins (99.9%)	Why? (nonlinear χ creates stiff ODE)
Hypothesis	Spectral fails when stiffness>1.9	Why? (explicit stepping + CFL limit)
Evaluation	P001 score = 3.38	Why 3.38? (keyword match, not content)

Physical/Mathematical Explanations Needed

Implicit FDM #1	Crank-Nicolson is A-stable for parabolic PDE. Nonlinear χ is handled implicitly → no CFL constraint.
Spectral instability	Explicit time-stepping: $\Delta t \leq C\cdot(\Delta r)^2/\chi\_max$. High α → large χ → smaller stable Δt required.
Compact4 wins at high α	4th-order spatial truncation error $O(\Delta r^4)$ vs 2nd-order $O(\Delta r^2)$. Advantage grows when solution has steep gradients.
P2 FEM slow	Quadratic element → 2N+1 DOFs. Sparse matrix assembly + solve at each timestep. Not using banded structure.

Missing: PhysicalInsightAgent — "Why does this solver suit this problem?"

Needed: Map solver properties (implicit/explicit, spatial order, basis functions) to PDE characteristics (χ stiffness, gradient sharpness, CFL constraints) to explain WHY.

Summary & Conclusions

What We Built

- ✅ Sequential agent pipeline: 6 agents, 7-phase improvement cycle
- ✅ PHYSBO Bayesian optimization for efficient dt exploration
- ✅ Automated: Pareto analysis → bottleneck detection → proposal → evaluation
- ✅ Advanced prototype: CriticAgent debate pattern for cross-validation

What We Learned

- Agents automate What (rankings, gaps, correlations) but not Why
- Scoring is keyword-based heuristic, not content understanding
- Pipeline is sequential (data dependency), not parallel
- Physical insight (A-stability, CFL, truncation order) remains human knowledge

Key Conclusion

The essential next step is a PhysicalInsightAgent that explains WHY each solver is suited to each problem, connecting solver properties (implicit/explicit, spatial order) to PDE characteristics (χ stiffness, CFL).

Next Steps

- PhysicalInsightAgent: solver property × PDE characteristic → explanation
- LLM integration for content-aware evaluation and natural language reasoning