RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

PROGRAMME DE RECHERCHE
NUMÉRIQUE POUR L'EXASCALE

HPC-AI convergence

AI-in-HPC, HPC-for-AI, AI-as-Oracle

J Bobin, J. Gusak, C. Prud'homme, J.P. Vilotte

February 3, 2026

PEPR NumPEx

Why This Workshop? Success Criteria

Context and Stakes

- **NumPEx post-exascale:** HPC–AI convergence is inevitable, but pathways are multiple
- **Audience:** computer scientists, mathematicians, decision-makers, ...
- **Horizon:** 10 years, milestones **2028** and **2030**

Success Criteria (end of workshop)

- A **shared vision** (key messages) for CSA positioning
- A rough **Pareto curve HPC/AI** (?)
- A **roadmap** with concrete outcomes for 2028 and 2030
- **Sandboxes** projects to test ideas and build expertise
- Open **Guidelines** for our community and beyond

The 3 Topics of HPC–AI Convergence

T1: AI in HPC
(hybrid simulation + AI components)

T2: AI as Oracle
(agentic AI, dev, perf, tuning)

T3: HPC for AI
(training/inference at scale, co-design)

Each topic is interrogated by:

- Cross-cutting requirements (trust, energy, sovereignty, skills, benchmarks)
- Flagship use-case: "DT operations + AI advisor" (stress-tests all 3 topics)

Cross-Cutting Requirements

Apply systematically to each topic:

- **Trust & certification:** V&V, UQ, reproducibility, traceability. What do we certify?
- **Energy & resource constraints:** J/solution vs J/insight, CO_2/experiment. Frugal post-exascale.
- **Software sustainability:** AI vs HPC pace, qualification, portability of stacks.
- **Sovereignty:** build vs buy, lock-in, critical dependencies, public policies.
- **Roles & skills:** HPC software eng, scientific MLOps, data steward, performance engineer, HPC/AI domain application developers.
- **Benchmarking & reporting:** credible metrics, eval suites (prompts/tests for agentic HPC tools).
- **New technologies:** Quantum, Neuromorphic arch, Photonic in transit Computing

Flagship Use-Case: DT Operations + AI Advisor

Why this use-case as a filter?

- **Operational Digital Twin:** real-time or near-real-time simulations (latency-critical)
- **AI advisor:** suggests parameters, detects anomalies, optimizes online
- **Stress-tests all 3 topics simultaneously**

T1: AI in HPC

- Fast surrogates
- ML closures
- Guarantees, UQ
- Out-of-domain detection

T2: AI as Oracle

- Orchestration
- Observability
- Governance, audit
- Trace exploitation

T3: HPC for AI

- Model updates
- Training at scale
- Data generation
- Drift monitoring

DT + AI Advisor: Operationalization Requirements

This flagship use-case has critical requirements:

- **Lifecycle management:** versioning (model+data), drift monitoring, retraining triggers, rollback
- **Auditability & trust:** logs, provenance, responsibility, explainability
- **Degraded modes:** safe fallback when AI is uncertain, default behaviors
- **SLA/KPIs:** time-to-insight, energy-to-insight, confidence levels

These requirements actually apply beyond DT:

- Online steering and optimization
- Continuous data assimilation
- Automated decision workflows
- Any production AI-in-HPC system

Framing Questions — Post-Exascale HPC-AI Future

In 10 years, what will computing look like?

- Supercomputers: fully AI-optimized, hybrid HPC-AI, or domain-specific?
- Hardware: training-optimized (GPUs) vs low-latency hybrid (fast interconnect, synchronization)?
- Bottleneck: interconnect bandwidth, data locality, or memory capacity?

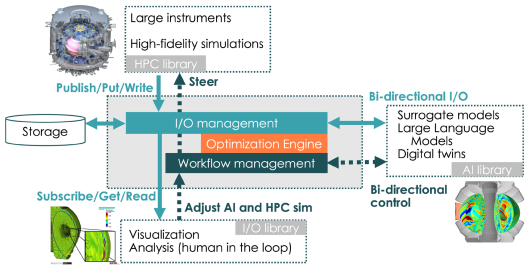
Key trade-offs and constraints:

- Energy: J/solution vs J/insight vs CO_2/experiment?
- Orchestration: AI-assisted or fully automated workflows? Human-in-loop or not?
- Sustainability: Can HPC stacks keep pace with AI framework evolution?
- Objective: Time-to-solution or time-to-insight?

T1: AI in HPC — Hybrid Simulation (Post-Exascale)

Post-exascale drivers:

- Surrogates & neural operators
- ML closures (when physics is too expensive)
- Parameter inversion at scale
- Multi-fidelity orchestration in optimization, UQ or inverse problems



Discussion questions:

- Which AI uses become standard: surrogates, closures, online steering, mesh generation, autotuning?
- Which domains will refuse AI-in-loop (regulation, safety, reproducibility)?
- Certification: What to certify—code, model, pipeline, or final decision?
- UQ: Where does uncertainty come from—physics, numerics, data, AI? How to combine?
- From interpretable (physics-driven) to black-box (foundation models): validation?

T1: AI-in-HPC

- AI/HPC hybridisation is a challenge, is building a converged software stack possible ?
- AI and HPC software have different lifecyles, AI-based frameworks are rapidly evolving while the HPC software /application code evolution have longer lifecycles. Is is possible to build a sustainable AI/HPC software stack ?
- AI-based frameworks are mainly developed by hyperscalers ? What is a "sovereign" AI/HPC software stack ?

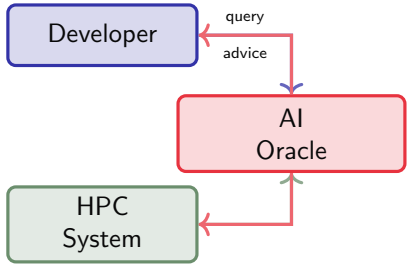
KPIs for T1 ?

- Speedup vs accuracy (e.g., 10× faster with <5% error)
- UQ coverage / confidence level (uncertainty quantification quality)
- Energy-to-solution ratio (J/simulation with AI vs baseline)

T2: AI as Oracle — Agentic Development (Post-Exascale)

Post-exascale needs:

- Code generation for extreme heterogeneity
- Auto-tuning for complex architectures
- AI-driven debugging at scale
- Workflow orchestration & automation



Discussion questions:

- Software Heritage as training data: opportunity for France/Europe?
- Can AI accelerate dev cycles (porting, translation, optimization)?
- Profiling oracles: AI-driven analysis to identify bottlenecks?
- Trace-based learning: Collect structured traces to optimize partitioning, scheduling?
- Trust & benchmarking: How to audit/validate AI-generated HPC code?

T2: AI-as-Oracle

What's the future of code development at the AI-dominated era ?

- **Trust & certification:** AI will likely take a large part of the inner loop of software development (code generation/translation/porting - orchestration/optimisation/automation), where the cursor should be put between AI-assisted to AI-automated developement ? How to verify (consistency) and validate (domain-specific performance) AI-generated codes ?
- **Data and qualification:** HPC codes are increasingly complex (e.g. heterogenous hardware), can AI be trusted for code generation at all levels of the software stack ? How to ensure the the qualification of AI-generated/orchestrated HPC codes ?

T2: AI-as-Oracle

What's the future of code development at the AI-dominated era ?

- **Skills :** what skills future computer scientists should have ? Promp-engineers only ?
- **Energy & resource constraints:** AI has a significant energy cost. How to optimize the energy cost of AI for software development ?
- **Data:** how to ensure data availability (e.g. from simulations to traces/log) and trust, which are central for validation and robustness ?

KPIs for T2 ?

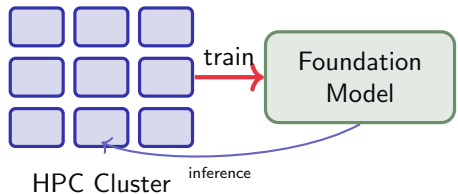
- Developer productivity gain (time saved, bugs prevented, tasks automated)
- Code quality metrics (correctness rate, performance regression rate)
- Human review overhead (% of AI output requiring manual validation)

T3: HPC for AI — Training & Inference (Post-Exascale)

Not just FLOPs: qualified, reproducible, sustainable pipelines for science/DT.

Post-exascale challenges:

- Foundation models for science (PB-scale data)
- Distributed training at extreme scale
- Real-time inference for DT/steering
- Co-design: CPU+GPU+NPU+emerging arch



Discussion questions:

- Co-design: Algorithms + architectures for transformers, foundation models?
- Scalability: Memory limits, heterogeneous arch, slow interconnects, low-precision?
- Distributed learning: Train on inherently distributed data (sim traces)?
- Sovereignty: Build-vs-buy for European AI-HPC stack?
- Data: Access, open models vs open data, PB-scale science datasets?

T3: HPC for AI

- **Trust / qualification:** what do we certify (code/model/pipeline/decision) for training & inference? What tests + what drift monitoring?
- **Energy & frugality:** which KPI is binding (J/insight, CO_2/experiment) and what trade-offs are acceptable (precision, batch size, comms)?
- **Sustainability of stacks:** how do we reconcile fast AI framework cycles with long HPC qualification/portability cycles?
- **Sovereignty / build-vs-buy:** where is vendor lock-in acceptable vs unacceptable (frameworks, kernels, compilers, runtimes)?
- **Skills / roles:** who operates this (scientific MLOps, perf engineer, data steward) and what is the minimum operating model?
- **Benchmarking & reporting:** shared benchmarks + periodic reports + prompt/eval suite for HPC agentic tools (regression gating).

KPIs for T3 ?

- Time-to-insight (not only throughput)
- Energy-to-insight (J/insight)
- Trust metric (QoI error + uncertainty coverage / generalization boundary)
- *Engineering diagnostics:* GPU efficiency (MFU), scaling efficiency.