DOCUMENT 2 : STATE OF THE ART

AIM²P — Cartographie du domaine et état de l'art

Phase : Phase 0 (cadrage scientifique) · Projet : AIM²P Initiative · Version : 0.2

> **État des connaissances au 17 novembre 2026**
>
> Cette version remplace la cartographie préliminaire de la version 0.1, dont la référence centrale (DeepSeek-Prover-V1.5, 2024) est devenue un repère historique. Les affirmations sont datées et sourcées. Les chiffres annoncés par un fournisseur sans reproduction indépendante sont signalés comme tels. Les scores de systèmes différents ne sont pas directement comparables : les budgets de calcul vont d'un essai agentique unique à plusieurs milliers d'échantillons.

1. PANORAMA : SIX FAMILLES DE TRAVAUX

L'intersection entre intelligence artificielle, calcul symbolique et preuve formelle s'organise aujourd'hui en six familles. Les travaux les plus proches d'AIM²P sont les systèmes spécialisés à trois composants et les prouveurs Lean appliqués à la physique.

Famille	Projets et acteurs clés	Forces majeures	Limites structurelles
LLM de raisonnement	DeepSeek-R1, OpenAI o1/o3, Gemini Deep Think, Llemma	Intuition mathématique, brouillons de preuve rapides ; niveau or à l'IMO 2025 en langue naturelle (Gemini Deep Think).	Aucune garantie formelle ; la notation par réponse finale ne certifie aucune étape intermédiaire.
Assistants de preuve	Lean 4 (Mathlib), Rocq (ex-Coq), Isabelle/HOL	Vérification par un petit noyau de confiance ; bibliothèques étendues (Mathlib : 8 718 fichiers en novembre 2026).	Le noyau vérifie l'énoncé formel, pas sa fidélité au problème ; les benchmarks Lean contiennent eux-mêmes des énoncés défectueux ; formalisation manuelle lente.
Prouveurs agentiques LLM + Lean	Seed-Prover 1.5, Hilbert, Goedel-Architect, AxProverBase, Numina-Lean-Agent, Ax-Prover, Aleph ; prouveurs à révision Goedel-Prover-V2, OProver	miniF2F saturé (≥ 99 %) ; PutnamBench de 88,8 % (modèle à poids ouverts) à 99,4 % (annonce non reproduite) ; réparation guidée par les messages d'erreur de Lean.	Budgets incomparables ; systèmes de tête fermés et coûteux ; preuves à auditer au niveau du noyau ; évaluation dominée par les concours.
Solveurs symboliques	SymPy, SageMath, Wolfram Mathematica, Singular	Calcul algébrique exact, simplification et dérivation symboliques.	Peuvent rendre des résultats faux mais bien formés ; pas de certificat vérifiable hors de l'algèbre polynomiale ; ne structurent pas une preuve.

Famille	Projets et acteurs clés	Forces majeures	Limites structurelles
Systèmes spécialisés à trois composants	LIPS, MechGeo, NSPI, Aristotle	LLM, calcul formel et Lean dans une même chaîne ; résultats certifiés par Lean dans leur domaine.	Un domaine chacun (inégalités, géométrie, sommes de carrés) ; le calcul formel y est un moteur de certificats, pas un critique général ; sa fiabilité n'est pas mesurée.
Architectures plus générales à trois outils	Moxia, SPL	LLM, calcul formel et Lean orchestrés au-delà d'un seul domaine.	Le calcul formel y résout le problème (Moxia) ou est appelé par une couche d'orchestration (SPL) ; il ne critique pas le raisonnement du LLM et ne pilote pas de boucle d'auto-correction évaluée.
IA pour la physique	IAIFI (MIT/Harvard), SciML, PINN ; depuis octobre 2025, prouveurs et bancs Lean : Ax-Prover, Lean4PHYS, PhysProver, FormalPhysics, AxQM	Modélisation numérique ; premières évaluations de prouveurs LLM + Lean en physique.	Côté numérique, pas de vérification formelle ; côté formel, bancs récents et petits (134 à 1 019 tâches), centrés sur l'information quantique et la physique de premier cycle.

2. PROUVEURS LEAN ET USAGE DU RETOUR DU COMPILATEUR

2.1 Des benchmarks de concours saturés

En 2024, DeepSeek-Prover-V1.5 atteignait 63,5 % sur miniF2F-test et 25,3 % sur ProofNet, grâce à l'apprentissage par renforcement sur le retour de Lean (RLPAF) et à RMaxTS, une variante de MCTS à récompense intrinsèque (arXiv:2408.08152). Deux ans plus tard, miniF2F est saturé. Hilbert (Apple, UCSD) atteint 99,2 % et résout 462/660 problèmes de PutnamBench (arXiv:2509.22819). Goedel-Architect obtient 99,2 % en pass@1 avec un modèle à poids ouverts, puis 88,8 % (597/672) sur PutnamBench avec amorçage en langue naturelle (arXiv:2606.06468). Seed-Prover 1.5 résout 581/672 problèmes de PutnamBench et 11 des 12 problèmes du Putnam 2025 (arXiv:2512.17260). Numina-Lean-Agent, agent généraliste bâti sur Claude Code et Claude Opus 4.5, en résout 12/12 en Lean et a servi à formaliser le théorème de Brascamp–Lieb (arXiv:2601.14027).

Au sommet, Aleph annonce 668/672 (99,4 %) (Logical Intelligence) et un pipeline fondé sur GPT-5.5 annonce 670/672 (message de l'auteur) : ces deux annonces n'ont pas été reproduites indépendamment et ne publient pas de coût par problème. AlphaProof est désormais publié dans *Nature* (novembre 2025) mais reste fermé (Google Research). La médaille d'or de Gemini Deep Think à l'IMO 2025 a été obtenue en langue naturelle, sans Lean : ce n'est pas de la vérification formelle (DeepMind).

2.2 Quatre manières d'utiliser le retour de Lean

Mode	Principe	Exemples
Pas à pas	Acceptation ou rejet de chaque tactique, nouvel état des buts	ReProver, BFS-Prover
Récompense scalaire	Le verdict de Lean sert de récompense à l'apprentissage par renforcement ou à la recherche arborescente	DeepSeek-Prover-V1.5, AlphaProof

Mode	Principe	Exemples
Révision par messages d'erreur	Le texte d'erreur est réinjecté pour réviser la preuve	Goedel-Prover-V2, OProver, APOLLO, Hilbert
Boucle agentique	Un LLM généraliste appelle des dizaines de fois les outils de diagnostic, d'état des buts, de recherche et d'édition via LSP ou MCP	Ax-Prover, AxProverBase, Numina-Lean-Agent, Leanstral, Aleph

En 2026, la boucle agentique combinée à une décomposition en lemmes ou en plan de preuve définit la frontière. L'agent minimal AxProverBase (un proposeur ReAct, un compilateur qui renvoie les erreurs, un relecteur qui vérifie que l'énoncé n'a pas été modifié, jusqu'à 50 itérations) atteint 54,7 % en pass@1 sur PutnamBench pour environ 12,6 $ par échantillon, avec un code ouvert (arXiv:2602.24273). L'écart entre systèmes ouverts et fermés s'est fortement réduit, mais les budgets vont d'un essai agentique de 50 itérations à pass@8192. La recherche arborescente de type MCTS n'est plus le cœur de l'état de l'art : c'est une option parmi d'autres, et Reward-Oracle MCTS montre qu'utiliser le compilateur comme simple récompense scalaire expose au piratage de récompense (arXiv:2608.28639).

2.3 Ce que l'on sait de la dynamique de réparation

Retirer le retour du compilateur fait chuter les performances, par exemple de 93,3 à 88,4 sur miniF2F pour OProver-32B, mais plus de tours n'aide pas toujours (11,3 % sur PutnamBench avec 8 tours contre 10,7 % avec 16) (arXiv:2605.17283). Goedel-Prover-V2, étendu à 5 révisions, atteint 92,7 % en pass@32 sur miniF2F contre 92,2 % en pass@8192 sans correction (arXiv:2508.03613). Sur 1 990 théorèmes LeanDojo, 84 à 96 % du gain de réparation arrive au premier tour, seules 6,2 % des tactiques rejetées sont réparées, et les incompatibilités de types, les échecs de classes de types et les délais dépassés ne se réparent presque jamais (arXiv:2610.26942). Enfin, à coût égal, l'auto-réparation n'aide que modestement, et seulement pour les modèles forts (arXiv:2306.09896). À notre connaissance (novembre 2026), aucun article ne publie de distribution du nombre de tours par lemme pour un système à décomposition, ni de courbe de réparation sur des énoncés de physique.

3. VALIDITÉ DES PREUVES : BENCHMARKS DÉFECTUEUX ET FIDÉLITÉ DES ÉNONCÉS

Un audit publié en juin 2026 sur cinq benchmarks Lean (miniF2F, ProofNet, FormalMATH, CombiBench, ProverBench ; 13 variantes, environ 10 318 problèmes) relève 4 833 signalements et 398 défauts certifiés par une preuve : hypothèses manquantes, traductions incomplètes, domaines ou types erronés, hypothèses vacues, énoncés sources faux. Les LLM utilisés comme juges n'y détectent les défauts sémantiques qu'avec une précision de 0,24 à 0,30 (arXiv:2606.29493). ProofNet# a dû corriger 118 des 371 énoncés des portages Lean 4 de ProofNet, soit 31,8 % (arXiv:2406.07222), et Aleph a produit des contre-exemples à 15 énoncés de PutnamBench, environ 2 % (Logical Intelligence).

Côté prouveurs, des preuves de DeepSeek-Prover-V2-7B compilent et passent le filtre `sorry` tout en reposant sur `sorryAx`, ce qui impose un audit au niveau du noyau et des axiomes (arXiv:2608.28639) ; avant Lean 4.20.0, un bogue de `apply?` faisait passer pour réussies des preuves sans déclaration vérifiée par le noyau (arXiv:2606.29493). L'autoformalisation progresse sans être fidèle : en moyenne sur six benchmarks, MathForm-8B produit 88,06 % d'énoncés qui compilent mais 72,37 % d'énoncés fidèles, et seulement 63 % sur FATE-H et 37 % sur FATE-X (arXiv:2608.14221).

Conséquence. Le noyau de Lean ne certifie que l'énoncé formel, pas sa fidélité au problème. Une affirmation du type « 100 % vérifié par le noyau » doit être remplacée par une chaîne d'audit explicite : énoncé gelé, liste

blanche d'axiomes, comparaison mécanique avec l'énoncé de référence, relecture humaine.

4. CALCUL FORMEL ET LEAN : DES PONTS ÉTABLIS, AVEC OU SANS LLM

4.1 LLM et calcul formel

Le couplage LLM–Python/SymPy est mature. ToRA, premier modèle ouvert à dépasser 50 % sur MATH, entrelace raisonnement en langue naturelle et appels à des bibliothèques de calcul et à des solveurs symboliques (arXiv:2309.17452), et Qwen2.5-Math propose un mode de raisonnement intégrant des outils (arXiv:2409.12122). L'idée de l'interpréteur comme critique remonte notamment à CRITIC et à l'auto-vérification par code de GPT-4 (résumés non revérifiés pour cette version). O-Forge couple un LLM de pointe et Mathematica pour des inégalités asymptotiques de niveau recherche, sans assistant de preuve (arXiv:2510.12350). Ces travaux traitent les plantages et les désaccords, jamais un résultat faux mais bien formé.

4.2 Calcul formel et Lean, sans LLM

Le schéma « le calcul formel calcule, Lean vérifie » est établi : Mathematica comme oracle non fiable vérifié par Lean (arXiv:2101.07758) ; tactiques où SageMath ou SymPy calculent et Lean vérifie restes, bases de Gröbner et appartenance à un idéal (arXiv:2604.13514) ; LeanSage, dont le README rappelle qu'« un succès de l'oracle n'est pas une preuve Lean » (Reservoir) ; LeanLink côté Wolfram (Wolfram). La tactique `polyrith` de Mathlib est en pratique hors service depuis la fermeture de son serveur Sage externe, saturé selon les mainteneurs peut-être par des agents IA (Zulip Lean) : un système comme AIM²P doit exécuter son calcul formel en local.

4.3 Les systèmes spécialisés à trois composants

Plusieurs systèmes associent déjà un LLM, un système de calcul formel et Lean, chacun dans un domaine précis :

- **LIPS** (ICLR 2025) : GPT-4o génère des tactiques de réécriture, les tactiques de sommes de carrés et de tangente sont implémentées à partir de SymPy, et Lean 4 vérifie les preuves finales ; 76,3 % de 161 inégalités olympiques. C'est le plus proche d'AIM²P par la pile logicielle (arXiv:2502.13834).
- **MechGeo** (2026) : certificats algébriques produits par Singular ou SymPy, chaque preuve et chaque contre-exemple vérifiés par le noyau de Lean, réparation des énoncés à partir des contre-exemples ; 29 des 43 énoncés historiques de géométrie de l'IMO prouvés et les 14 autres formellement réfutés. C'est le plus proche par le mécanisme (arXiv:2608.02295).
- **NSPI** (ICML 2026) : le LLM devine une décomposition en sommes de carrés approchée, le calcul symbolique la rend exacte, Lean certifie ; seul le résumé a été lu (arXiv:2605.15445).
- **Aristotle** (Harmonic, 2025) : LLM, exécution de code, moteur algébrique hors Lean et Lean ; niveau or à l'IMO 2025 en preuves formelles, selon ses auteurs (arXiv:2510.01346).
- **Xia et al.** (2026) : agent LLM, calcul formel et solveurs pour une étude de cas en combinatoire, dont la borne est vérifiée en Lean 4 (arXiv:2603.08322).

Deux architectures de 2026 dépassent un seul domaine, sans faire du calcul formel un critique du raisonnement du LLM :

- **Moxia** (2026) : le LLM se borne à reformuler l'énoncé dans un schéma étroit ; un pipeline déterministe de calcul formel résout le problème ou s'abstient, et lorsque la dérivation entre dans un fragment de Lean 4 et Mathlib, elle devient un théorème vérifié par le noyau. Selon l'auteur, 95,6 % de réponses correctes sur MATH sans réponse fausse affirmée, avec publication enregistrement par enregistrement. C'est le précédent

généraliste le plus proche d'AIM²P, mais le calcul formel y remplace le raisonnement du LLM au lieu de le critiquer, et il n'y a pas de boucle d'auto-correction (arXiv:2606.00671).

- **SPL** (2026) : un langage déclaratif qui compose appels au LLM et outils symboliques (SymPy, SageMath, Lean) dans une même spécification ; son évaluation porte sur la correction vérifiée des réponses (82 à 93 % selon les modèles) (arXiv:2607.07727).

Dans le même esprit, Peppy (2026) structure des preuves de convergence d'algorithmes d'optimisation pour qu'elles soient vérifiables par SymPy plutôt que par Lean, et referme plusieurs problèmes ouverts du domaine, dont des conjectures sur l'algorithme FGM de Nesterov (arXiv:2609.35762).

À l'inverse, les grands agents Lean généralistes (Ax-Prover, Numina-Lean-Agent, AIProver) ne listent aucun outil de calcul formel (arXiv:2610.05367 pour AIProver). Dans aucun des systèmes ci-dessus le calcul formel n'est un critique généraliste des étapes informelles du LLM dont la fiabilité serait elle-même mesurée. Une synthèse de 2026 signée par 41 auteurs, dont T. Tao, place d'ailleurs « l'intégration fiable d'outils » parmi les priorités du domaine (arXiv:2607.07779).

4.4 Le calcul formel n'est pas infaillible

Mathematica a calculé faux, et de façon incohérente, des déterminants de matrices entières, produisant de faux contre-exemples repérés par recoupement avec Maple (arXiv:1312.3270) ; le traqueur de SymPy maintient une étiquette `simplify` avec des problèmes de correction ouverts (GitHub SymPy). À notre connaissance (novembre 2026), aucune étude empirique systématique du taux d'erreur de SymPy n'a été publiée. En physique, l'équipe de TPBench constate que l'auto-vérification par SymPy « a à peine amélioré les résultats » (arXiv:2502.15815) ; un agent SymPy ultérieur transforme 83 % des étapes de dérivation en scripts avec 88 % de correction « perçue par l'agent », ce qui mesure le comportement de l'agent et non la vérité (arXiv:2506.20729) ; HELIOS, quand `simplify` dépasse son délai, saute la vérification et se fie à la dérivation du LLM (arXiv:2607.24051).

5. PHYSIQUE FORMELLE ET PROUVEURS LEAN POUR LA PHYSIQUE

5.1 Physlib : une bibliothèque en forte croissance

Physlib (leanprover-community) est issue du renommage HepLean → PhysLean, puis de la fusion avec Lean-QuantumInfo. Le 17 novembre 2026, elle comptait environ 1 635 fichiers et 451 000 lignes sur 20 domaines : mécanique lagrangienne et hamiltonienne (équations de Hamilton, crochet de Poisson), mécanique quantique avec opérateurs non bornés, commutateurs et inégalité d'incertitude, unités et dimensions, relativité, cosmologie, matière condensée, QFT et information quantique (Physlib). Environ 245 000 de ces lignes sont dans PhyslibAlpha, niveau à relecture allégée, et le cœur relu en compte environ 166 000. Le module `Units` définit les dimensions comme un groupe commutatif et des grandeurs typées `WithDim`. Physlib a sept mainteneurs, recommande de citer l'article HepLean (arXiv:2405.08863) et embarque des répertoires de compétences pour agents IA, ce qui en fait la cible amont naturelle d'une contribution AIM²P.

5.2 Prouveurs et bancs d'essai LLM + Lean en physique

Depuis octobre 2025, une grappe de travaux est apparue. Ax-Prover introduit QuantumTheorems (134 problèmes de mécanique quantique), où il atteint 96 % contre 40 % pour Claude Sonnet seul (arXiv:2510.12787). Lean4PHYS et son banc LeanPhysBench (200 énoncés) donnent 35 % à Claude-Sonnet-4 contre 16 % au meilleur prouveur mathématique spécialisé, signe d'un transfert limité depuis les mathématiques (arXiv:2510.26094). PhysProver gagne 2,4 % par apprentissage par renforcement sur des données PhysLean (arXiv:2601.15737). FormalPhysics

propose 200 problèmes universitaires et caractérise la « dérive sémantique » (arXiv:2604.23002). AxQM fournit 1 019 tâches vérifiées par le noyau, tirées de Nielsen & Chuang (arXiv:2609.05157).

5.3 Formalisations de niveau recherche et erreurs trouvées

Des formalisations de niveau recherche assistées par agents existent : le théorème fondamental des états produits de matrices (TNLean, arXiv:2607.07857), la sécurité composable de BB84 à taille finie, dont le pipeline a ensuite resserré la pénalité de taille finie (arXiv:2610.17286), la solution de Seiberg–Witten dérivée d'axiomes physiques nommés, Lean indiquant de quels postulats dépend chaque résultat (arXiv:2607.06379), un théorème central limite pour la fonction de partition du modèle SYK à haute température, démontré avec l'aide d'une IA puis formalisé en Lean 4 (arXiv:2610.18440), et une classification de spectres de charges de modèles SU(5), formalisée dans Physlib comme infrastructure réutilisable (arXiv:2603.28406). L'analyse dimensionnelle a aussi été formalisée en Lean 4 : dimensions en groupe abélien, grandeurs physiques typées, théorème Pi de Buckingham (arXiv:2509.13142). La formalisation trouve aussi des erreurs : en 2026, Tooby-Smith a montré en Lean qu'une condition de stabilité du modèle à deux doublets de Higgs, présentée comme nécessaire et suffisante dans un article de 2006 très cité, n'est pas suffisante (arXiv:2603.08139) ; plus tôt, Isabelle Marries Dirac avait aidé à découvrir une faille dans une preuve publiée sur le dilemme du prisonnier quantique (arXiv:2110.01320). Tous ces travaux désignent la fidélité de l'énoncé, et non la recherche de preuve, comme le goulot principal ; TNLean le dit ainsi : le principal obstacle est de « faire respecter l'intention mathématique ».

5.4 Le cas de la complexité de Krylov

Le domaine est mûr et dispose de revues de référence (arXiv:2405.09628 ; arXiv:2507.06286 ; arXiv:2503.10753). L'hypothèse de croissance universelle de Parker et al. (arXiv:1812.08657) lie croissance linéaire des coefficients de Lanczos et chaos, mais la borne sur l'exposant de Lyapunov n'est démontrée qu'à température infinie, et la croissance exponentielle de K(t) n'est pas à elle seule un diagnostic de chaos (arXiv:2104.09514 ; arXiv:2203.03534). Côté formel, une recherche directe sur les dépôts au 17 novembre 2026 ne trouve aucun fichier contenant « Lanczos », « tridiagonal », « operator growth » ou « Liouvillian » dans Mathlib (la seule occurrence de « Krylov » y concerne le théorème de Krylov–Bogolyubov, sans rapport), ni « Krylov », « Lanczos », « operator growth », « Liouvillian » ou « holograph » dans Physlib ; aucun algorithme de Lanczos vérifié n'a été trouvé en Coq ou en Isabelle. Les travaux adjacents sont un algorithme de Lanczos en algèbre exacte sans assistant de preuve (arXiv:2403.06391) et un dépôt Lean isolé, mono-auteur, sans licence ouverte et non relu, consacré à la dynamique de Krylov en dimension finie (lanzarini-krylov-lean).

6. BENCHMARKS INFORMELS DE PHYSIQUE

TPBench donnait 15 % à o3-mini au niveau recherche début 2025 et documentait des erreurs de signe, des facteurs i perdus, des développements de Taylor hors de leur rayon de convergence et des erreurs de convention (arXiv:2502.15815). CritPt donnait 5,7 % à GPT-5 dans l'article (arXiv:2509.26574), mais son classement v2 affiche 70 à 83 % pour les modèles de fin 2026, avec des réglages d'outils non précisés qui rendent la comparaison incertaine (Hugging Face). Une recorrection experte de septembre 2026 trouve que, sur 250 rejets audités dans six benchmarks de physique, 57,2 % étaient des erreurs du benchmark, 38,0 % du correcteur et seulement 4,8 % du modèle ; une fois ces erreurs réparées, les modèles atteignent 87 à 95 % sur quatre de ces benchmarks (arXiv:2609.13009).

L'argument le plus solide n'est donc plus « les LLM ne savent pas faire de physique », mais que **la notation par réponse finale ne certifie aucune étape intermédiaire**. La vérification pas à pas devient la mesure qui garde du pouvoir discriminant.

> **Limites de cette revue**
>
> État au 17 novembre 2026 : des préprints pertinents paraissent chaque semaine, et une recherche arXiv finale sur « SymPy + Lean », « Krylov + Lean » et « physics + Lean 4 » doit précéder toute publication. Le Zulip Lean et le Lean Reservoir n'ont pas été fouillés directement, l'Archive of Formal Proofs n'a pas été interrogée par mot-clé, et Seed-Prover 1.5, AxiomProver, Gauss et Leanstral n'ont pas été examinés pour un éventuel outil de calcul formel. La matrice comparative complète (LLM, Lean 4, SymPy, Rocq, Isabelle) reste l'objet de la phase P1.

RÉFÉRENCES

Prouveurs Lean et boucles de réparation

- Xin et al., *DeepSeek-Prover-V1.5: Harnessing Proof Assistant Feedback for Reinforcement Learning and Monte-Carlo Tree Search*, ICLR 2025. arXiv:2408.08152
- Goedel-Prover-V2 (Princeton), auto-correction par messages d'erreur, 2025. arXiv:2508.03613
- Hilbert (Apple, UCSD), 2025. arXiv:2509.22819
- Seed-Prover 1.5 (ByteDance Seed), 2025. arXiv:2512.17260
- Numina-Lean-Agent (Project Numina), 2026. arXiv:2601.14027
- AxProverBase (Axiomatic AI), agent minimal à code ouvert, 2026. arXiv:2602.24273
- OProver, raffinement multi-tours sur le retour brut de Lean, 2026. arXiv:2605.17283
- Goedel-Architect (Princeton), décomposition en plan de preuve, 2026. arXiv:2606.06468
- Reward-Oracle MCTS, compilateur comme récompense scalaire, 2026. arXiv:2608.28639
- *Fix, Not Just Resample*, réparabilité par type d'erreur, 2026. arXiv:2610.26942
- AlphaProof (Google DeepMind), *Nature*, 2025. Google Research
- Aleph (Logical Intelligence), annonce PutnamBench non reproduite, 2026. Billet

Validité des preuves et méthodologie d'évaluation

- *Faults in Our Formal Benchmarking*, 2026. arXiv:2606.29493
- ProofNet#, correction des énoncés Lean 4 de ProofNet, 2024. arXiv:2406.07222
- MathForm, autoformalisation et fidélité des énoncés, 2026. arXiv:2608.14221
- Olausson et al., auto-réparation évaluée à coût égal, 2023. arXiv:2306.09896
- Bowyer et al., statistiques d'évaluation sur petits benchmarks, 2025. arXiv:2503.01747

Calcul formel, Lean et LLM

- Li et al., *Proving Olympiad Inequalities by Synergizing LLMs and Symbolic Reasoning* (LIPS), ICLR 2025. arXiv:2502.13834
- Bruno, *Moxia: A Trust-First Neuro-Symbolic Execution Architecture for Self-Explaining Mathematical Reasoning*, 2026. arXiv:2606.00671
- Gong, *SPL: Orchestrating Workflows with Declarative Deterministic-Probabilistic Composition*, 2026. arXiv:2607.07727

- Suh et al., *Peppy: An AI-Assisted Workflow for Tight Convergence Analysis of Optimization Algorithms*, 2026. arXiv:2609.35762
- Shen et al., *Autoformalizing and Proving Euclidean Geometry in Lean 4* (MechGeo), 2026. arXiv:2608.02295
- Zuo et al., NSPI, preuve d'inégalités polynomiales par certificats de sommes de carrés, ICML 2026. arXiv:2605.15445
- Aristotle (Harmonic), 2025. arXiv:2510.01346
- Xia et al., collaboration neuro-symbolique agentique en combinatoire, 2026. arXiv:2603.08322
- AIProver, 2026. arXiv:2610.05367
- Shen et al., *Automated Tactics for Polynomial Reasoning in Lean 4*, 2026. arXiv:2604.13514
- Lewis & Wu, pont Lean–Mathematica, 2021. arXiv:2101.07758
- LeanSage et LeanLink : Reservoir ; Wolfram
- ToRA, ICLR 2024. arXiv:2309.17452 ; Qwen2.5-Math, 2024. arXiv:2409.12122
- CRITIC, 2023. arXiv:2305.11738 ; auto-vérification par code de GPT-4, 2023. arXiv:2308.07921 (résumés non revérifiés)
- Khaitan & Ganesh, O-Forge, 2025. arXiv:2510.12350
- Durán, Pérez & Varona, erreurs de calcul formel dans Mathematica, *Notices AMS*, 2014. arXiv:1312.3270
- *From Solvers to Research: LLM-Driven Formal Mathematics at the Research Frontier* (41 auteurs), 2026. arXiv:2607.07779

Physique formelle et évaluation en physique

- Tooby-Smith, *HepLean: Digitalising high energy physics*, 2024. arXiv:2405.08863 ; Physlib : GitHub
- Formalisation du théorème de Wick en Lean 4 (PhysLean), 2025. arXiv:2505.07939
- Tooby-Smith, erreur du modèle à deux doublets de Higgs trouvée par formalisation, 2026. arXiv:2603.08139
- Ax-Prover, 2025. arXiv:2510.12787 ; Lean4PHYS, 2025. arXiv:2510.26094 ; PhysProver, 2026. arXiv:2601.15737
- FormalPhysics, 2026. arXiv:2604.23002 ; AxQM, 2026. arXiv:2609.05157
- TNLean, 2026. arXiv:2607.07857 ; BB84, 2026. arXiv:2610.17286 ; Seiberg–Witten, 2026. arXiv:2607.06379
- Kamijima et al., *A Central Limit Theorem for High-Temperature SYK Model: A case study on mathematical research using AI and Lean*, 2026. arXiv:2610.18440 ; Krippendorf & Tooby-Smith, *Physics as Code: From Scans to Theorems with ITP APIs in SU(5) Model Building*, 2026. arXiv:2603.28406 ; Bobbin et al., *Formalizing dimensional analysis using the Lean theorem prover*, 2025. arXiv:2509.13142
- TPBench, 2025. arXiv:2502.15815 ; CritPt, 2025. arXiv:2509.26574 ; recorrection experte, 2026. arXiv:2609.13009

Complexité de Krylov

- Parker et al., hypothèse de croissance universelle des opérateurs, 2019. arXiv:1812.08657
- Caputa, Magán & Patramanis, régimes exacts de la complexité de Krylov, 2021. arXiv:2109.03824
- Nandy et al., revue de référence, *Physics Reports*, 2025. arXiv:2405.09628 ; Rabinovici et al., revue, 2025. arXiv:2507.06286 ; Baiguera et al., revue, *Physics Reports*, 2026. arXiv:2503.10753
- Croissance exponentielle de K(t) hors chaos : théories conformes, y compris libres. arXiv:2104.09514 ; modèle intégrable LMG. arXiv:2203.03534

- Sasaki, Lanczos en algèbre exacte, 2024. arXiv:2403.06391 ; dépôt lanzarini-krylov-lean (non relu)

HISTORIQUE DES VERSIONS

- **v0.1 :** cartographie préliminaire de la phase 0.
- **v0.2 (17 novembre 2026) :** réécriture complète après l'audit bibliographique de clôture de P0 : tableau de synthèse corrigé (assistants de preuve, solveurs symboliques, IA pour la physique) et complété (prouveurs agentiques LLM + Lean, systèmes spécialisés à trois composants) ; sections nouvelles sur l'usage du retour de Lean, la validité des preuves, les ponts calcul formel–Lean, la physique formelle et les benchmarks informels de physique ; références mises à jour. Intègre la recherche arXiv de vérification finale du 17 novembre 2026 (notamment Moxia et SPL).