

# Proof-of-concept de la formalisation LEAN 4 de la stabilité des familles exponentielles sous convolution gaussienne

Paper B — Contrats de fidélité, corpus adverse  
et dette mécanique résiduelle

Emmanuel Sérié

17 avril 2026

## Résumé

Ce papier documente un *proof-of-concept de la formalisation LEAN 4* du résultat principal de *Paper A* ([Paper A]) — la caractérisation des statistiques suffisantes  $\varphi$  pour lesquelles la famille exponentielle  $\{p_\theta \propto e^{-\theta^\top \varphi}\}$  est stable par convolution avec le noyau gaussien  $\gamma_\sigma$ , et sa réduction au cas quadratique via la reparamétrisation fermée  $\widetilde{M} = M(I_d + \sigma^2 M)^{-1}$ ,  $\widetilde{b} = (I_d + \sigma^2 M)^{-1}b$ . Le livrable est explicitement un *proof-of-concept* : la bibliothèque `lean/EFS` pinnée à `MATHLIB v4.29.0` déclare les contrats Q2 et Q3 et héberge un corpus adverse à vingt entrées, mais la chaîne d'outils `elan/lake/lean` n'est pas disponible sur le PATH de cet hôte, si bien qu'aucun `lake build` n'a encore fermé le *firewall LLM* : la vérification mécanique reste en chantier. Plus précisément, `EFS/Q2.lean` porte six `sorry` et `EFS/Q3.lean` en porte quatre (inventaire intégral dans `docs/lore/lean-unproved.md`) ; trois étapes d'amorçage restent à exécuter avant toute promotion : (i) installation globale de `elan` sur le PATH, (ii) `lake exe cache get` pour rapatrier les quelque 5 Go de `.olean` MATHLIB, (iii) élimination itérative des `sorry` en commençant par Q2. La bibliothèque `lean/EFS` déclare trois contrats : (Q2) la stabilité du cas quadratique avec formule fermée, (Q3) l'équivalence de Hamilton–Jacobi sous *stabilité régulière*, et un *corpus adverse* de vingt fichiers intentionnellement cassés que `lake env lean devra` rejeter une fois la chaîne d'outils amorcée. Les énoncés sont *pinnés* comme ancrages de fidélité (modulo `sorry`) ; chaque `sorry` est documenté comme *upstream-sketch* ou *narrative-only* dans `docs/lore/lean-unproved.md`. Nous discutons ce qui a été *facile* en LEAN 4 (l'algèbre matricielle SPD, la continuité  $C^1$  des chemins), ce qui a été *difficile* (l'absence d'un laplacien général et d'un théorème de chaleur en MATHLIB `v4.29.0`) et esquissons trois contributions MATHLIB candidates : `multivariateGaussian_conv_multivariateGaussian`, `laplacian` sur un espace euclidien, et `heat_equation_gaussian_convolution`. Le corpus adverse couvre dix modes de défaillance par question (inversion de quantificateurs, oubli de minimalité, glissement de signe Cole–Hopf, etc.) et constitue la contrepartie formelle des *circularités* diagnostiquées en *Paper A*.

## Table des matières

<b>1</b>	<b>Motivation — pourquoi formaliser ce problème</b>	<b>3</b>
1.1	Le risque de dérive narrative . . . . .	3
1.2	Ce que ce papier apporte . . . . .	3
1.3	Public visé et règles du jeu . . . . .	3
1.4	Statut de proof-of-concept et environnement cible . . . . .	4
<b>2</b>	<b>Énoncés formels : de la phrase naturelle au contrat LEAN 4</b>	<b>4</b>
2.1	Théorème Q2 — cas quadratique, formule fermée . . . . .	4

2.1.1	Énoncé informel . . . . .	4
2.1.2	Contrat LEAN 4 . . . . .	4
2.1.3	Ce que le contrat garantit . . . . .	5
2.2	Théorème Q3 — équivalence Hamilton–Jacobi sous stabilité régulière . . . . .	5
2.2.1	Énoncé informel (Feynman-reformulé) . . . . .	5
2.2.2	Contrat LEAN 4 . . . . .	5
2.2.3	Ce que le contrat garantit . . . . .	6
2.3	Le contrat structurel : pourquoi des <i>structures</i> plutôt que des propositions? . . . . .	6
<b>3</b>	<b>Architecture de la bibliothèque lean/EFS</b>	<b>7</b>
3.1	Arborescence . . . . .	7
3.2	Imports MATHLIB — les briques réellement utilisées . . . . .	7
3.2.1	Cas Q2 (module EFS/Q2.lean) . . . . .	7
3.2.2	Cas Q3 (module EFS/Q3.lean) . . . . .	7
3.3	Dépendances internes . . . . .	8
3.4	Régression guard : l’invariant 10 <code>sorry</code> + 20 <code>rejects</code> . . . . .	8
<b>4</b>	<b>Ce qui a été facile en LEAN 4 — et ce qui a été dur</b>	<b>8</b>
4.1	Facile : l’algèbre matricielle SPD . . . . .	8
4.2	Facile : la continuité et la différentiabilité des chemins . . . . .	8
4.3	Dur : le laplacien général sur $\mathbb{R}^d$ . . . . .	9
4.4	Dur : l’équation de la chaleur pour la convolution gaussienne . . . . .	9
4.5	Dur : l’équivalence Cole–Hopf dans le langage LEAN 4 . . . . .	9
4.6	Dur : unicité de la chaleur (Widder / Tychonoff) . . . . .	9
4.7	Récapitulatif : carte des obstacles . . . . .	10
<b>5</b>	<b>Esquisses amont pour MATHLIB</b>	<b>10</b>
5.1	<code>multivariateGaussian_conv_multivariateGaussian</code> . . . . .	10
5.2	<code>laplacian</code> sur un espace euclidien de dim finie . . . . .	10
5.3	<code>heat_equation_gaussian_convolution</code> . . . . .	11
5.4	Deux contributions de seconde priorité . . . . .	11
<b>6</b>	<b>Le corpus adverse — vingt fichiers à faire échouer</b>	<b>11</b>
6.1	Principe — et pourquoi il est différent d’une simple suite de tests . . . . .	11
6.2	Taxonomie Q2 (dix entrées) . . . . .	12
6.3	Taxonomie Q3 (dix entrées) . . . . .	12
6.4	Un exemple complet : <code>SilentReparam</code> . . . . .	13
6.5	Un deuxième exemple : <code>WrongSignHJ</code> . . . . .	13
<b>7</b>	<b><code>sorry</code> résiduels et chemin vers 0</b>	<b>13</b>
7.1	Inventaire des dix <code>sorry</code> . . . . .	13
7.2	L’escape hatch axiomatique . . . . .	13
7.3	Statut UNPROVABLE_IN_BUDGET . . . . .	14
<b>8</b>	<b>Discussion — preuve informelle et preuve formelle</b>	<b>14</b>
8.1	Ce que Paper A voit et que Paper B ne voit pas . . . . .	14
8.2	Ce que Paper B voit et que Paper A ne voit pas . . . . .	14
8.3	Rapport coût/bénéfice . . . . .	14
8.4	Leçon méthodologique — le <i>fidelity anchor</i> . . . . .	15
<b>9</b>	<b>Conclusion</b>	<b>15</b>

# 1 Motivation — pourquoi formaliser ce problème

## 1.1 Le risque de dérive narrative

Le problème posé par Étienne ([Étienne]) concerne trois questions Q1–Q3 articulées autour d’un seul objet, la reparamétrisation  $\theta \mapsto \tilde{\theta}$  induite par la convolution  $p_\theta * \gamma_\sigma$ . Dans l’articulation de Paper A, trois pièges se sont manifestés comme des *circularités* ou des *glissements d’hypothèses* qu’une rédaction informelle peut masquer :

1. **Q1a.** La proposition (†) — fermeture de la famille  $\{\varphi_i\}$  sous  $G_{ij}, H_i$  dans le tube  $\mathcal{A}$  — demande une direction *statique* (identité polynomiale en  $\theta$ ) et une direction *dynamique* (construction du chemin). Les deux sont souvent fusionnées dans la littérature, produisant une apparence de circularité ; voir `circular-q1a-stability-vs-closure.md`.
2. **Q3.** L’énoncé PDF d’Étienne — « stabilité Q1  $\Leftrightarrow$  chemin  $C^1$  vérifiant (6) » — *glisse silencieusement* trois hypothèses entre Q1 ponctuelle et Q3 régulière : minimalité (H2), régularité  $C^1$  de  $\sigma \mapsto \tilde{\theta}(\sigma)$  (H3), sous-gaussianité uniforme (H4). Voir `circular-q3-glissement.md`.
3. **Cas quadratique.** La statistique  $\varphi(x) = (\text{vec}(xx^\top), x)$  est *non minimale* ; la partie antisymétrique de  $T$  est un noyau non identifiable en  $\theta$ .

Chacun de ces pièges est un *hasard de fidélité* : une formulation légèrement affaiblie ressemble au théorème correct mais est en fait fausse ou vide. Un assistant de preuve comme LEAN 4 est le seul outil connu pour rendre ces glissements mécaniquement détectables : l’élaborateur refuse tout énoncé sous-déterminé, et l’ensemble `#print axioms` expose immédiatement tout recours à un axiome non prévu.

## 1.2 Ce que ce papier apporte

*Paper A* est une preuve mathématique ; ce papier est son *ancrage mécanique*. Nous fournissons :

1. (§2.1, §2.2) les *contrats* LEAN 4 des théorèmes Q2 et Q3, traduisant chaque objet mathématique ( $\Theta^*$ ,  $\tilde{M}$ , chemin admissible, minimalité) en un terme type-checkable ;
2. (§3) l’architecture de la bibliothèque `lean/EFS` — modules, imports MATHLIB, dépendances internes ;
3. (§4) une discussion franche de ce qui passe facilement à la moulinette de MATHLIB v4.29.0 et de ce qui exige encore un travail non trivial ;
4. (§5) trois *upstream sketches* (`multivariateGaussian_conv_multivariateGaussian`, `laplacian`, `heat_equation_gaussian_convolution`) formulés comme candidats de contribution à MATHLIB ;
5. (§6) un *corpus adverse* de 20 fichiers (10 par question) encodant chacun un mode de défaillance connu ;
6. (§8) une lecture du rapport entre preuve informelle et preuve formelle — ce que chacune voit que l’autre ne voit pas.

Les obligations résiduelles sont intégralement énumérées dans `docs/lore/lean-unproved.md` et tracées ci-dessous en §7.

## 1.3 Public visé et règles du jeu

Ce papier s’adresse à deux publics : (i) le mathématicien de Paper A qui souhaite comprendre ce que la formalisation garantit *au-delà* de la preuve écrite, (ii) le *probe engineer* qui attaquera les `sorry` résiduels. Nous ne reproduisons aucune preuve informelle de Paper A ; nous nous restreignons aux *choix de formalisation* et aux *distances Mathlib*.

## 1.4 Statut de proof-of-concept et environnement cible

Le livrable est un *proof-of-concept* : la fidélité statement-level est établie (les énoncés Q2, Q3 et les vingt entrées du corpus adverse sont figés comme contrats typés), mais la vérification mécanique par le noyau LEAN4 n'a pas encore été exercée sur cet hôte : la chaîne d'outils `elan/lake/lean` n'est pas présente sur le PATH par défaut, et aucun `lake build` n'a, à ce jour, fermé le *firewall LLM* défini par la constitution de la sous-flotte `formal`. L'environnement cible, décrit dans `lean/README.md`, est :

- `leanprover/lean4:v4.29.0` (pin `toolchain`) ;
- `mathlib @ leanprover-community/mathlib4 v4.29.0` (pin `lakefile.toml`) ;
- commandes attendues : `lake build` (devra exiter 0 avec dix warnings *declaration uses 'sorry'* — six pour Q2, quatre pour Q3), `./scripts/verify-corpus.sh` (devra exiter 0 avec 20 rejets sur 20).

**Trois étapes d'amorçage résiduelles.** Pour qu'un futur *probe-engineer* transforme ce proof-of-concept en preuve mécanique complète :

1. **Installation globale de elan.** Placer `$HOME/.elan/bin` sur le PATH de l'environnement d'exécution, ou installer `elan` globalement selon la procédure `lean/README.md`.
2. **Cache MATHLIB.** Exécuter `lake exe cache get` depuis `lean/EFS/` pour rapatrier les quelque 5 Go de `.olean` précompilés depuis Azure ; sans cette étape, `lake build` recompile MATHLIB intégralement (ordre de grandeur : heures sur ARM, jours sur matériel modeste).
3. **Élimination itérative des sorry.** Commencer par `EFS/Q2.lean` (six `sorry`, toutes algébriques, aucune ne dépend d'un laplacien upstream ; voir `docs/lore/lean-unproved.md`) puis attaquer `EFS/Q3.lean` (quatre `sorry`, dont la moitié *narrative-only* dépend des *upstream-sketches* `laplacian`, `heat_equation_gaussian_convolution` et `coleHopf_density_heat_iff`).

## 2 Énoncés formels : de la phrase naturelle au contrat LEAN 4

### 2.1 Théorème Q2 — cas quadratique, formule fermée

#### 2.1.1 Énoncé informel

Soit  $\varphi(x) = (\text{vec}(xx^\top), x) \in \mathbb{R}^{d^2+d}$ . Paramétrons :

$$\theta^\top \varphi(x) = \frac{1}{2} x^\top M x + b^\top x, \quad (M, b) \in \text{Sym}_d^{++} \times \mathbb{R}^d.$$

Pour tout  $\sigma^2 > 0$ , la convolution  $p_\theta * \gamma_\sigma$  est proportionnelle à  $\exp(-\tilde{\theta}^\top \varphi)$  avec

$$\tilde{M} = M (I_d + \sigma^2 M)^{-1}, \quad \tilde{b} = (I_d + \sigma^2 M)^{-1} b. \quad (\star)$$

De plus,  $\tilde{M} \in \text{Sym}_d^{++}$  et l'image de la reparamétrisation est le sous-cône strict  $\{\tilde{M} \prec \sigma^{-2} I_d\}$  (le lissage est irréversible).

#### 2.1.2 Contrat LEAN 4

Le théorème de Paper A est encodé dans le module `lean/EFS/EFS/Q2.lean` par un *fidelity anchor* :

```
-- Canonical parameter space of the quadratic exponential family. -/  
structure CanonicalParam (d : Nat) where  
  M : Matrix (Fin d) (Fin d) Real  
  b : Fin d -> Real
```

```

hM_sym : M.IsSymm
hM_posDef : M.PosDef

noncomputable def shift (p : CanonicalParam d) (s : Real) :
  Matrix (Fin d) (Fin d) Real := 1 + s • p.M
noncomputable def tildeM (p : CanonicalParam d) (s : Real) := p.M * (p.shift s)-1
noncomputable def tildeB (p : CanonicalParam d) (s : Real) := (p.shift s)-1 *v p.b

/-- Fidelity anchor for Q2 : a canonical reparametrisation exists. -/
theorem q2_stability (p : CanonicalParam d) {s : Real} (hs : 0 < s) :
  ∃(q : CanonicalParam d),
    q.M = p.tildeM s ∧ q.b = p.tildeB s ∧
    (multivariateGaussian (toEuclidean (- (p.M-1 *v p.b))) p.M-1).conv
      (multivariateGaussian 0 (s • (1 : Matrix (Fin d) (Fin d) Real)))
    = multivariateGaussian (toEuclidean (- (q.M-1 *v q.b))) q.M-1 := by
sorry

```

Cinq lemmes structurels précèdent `q2_stability` : `shift_posDef`, `woodbury_commutative`, `tildeM_posDef`, `tildeM_lt_inv_s`, `reparam_injective`. Deux témoins du flot de Riccati sont fournis en forme *existentielle triviale* (`riccati_precision`, `riccati_linear`) qui *ne dépendent pas de sorry* : ils affichent que  $\dot{M} = -M^2$  et  $\dot{b} = -Mb$  sont des témoins constructibles, ce qui est suffisant pour pinner la forme.

### 2.1.3 Ce que le contrat garantit

L'énoncé `q2_stability` est une *égalité de mesures* sur `EuclideanSpace Real (Fin d)`, pas une proportionnalité : c'est la forme la plus forte compatible avec la convention PDF d'Étienne (les deux membres sont des mesures de probabilité au même terme de normalisation gaussienne près). L'existence de  $(q.M, q.b) \in \Theta^*$  avec formule fermée ( $\star$ ) est *intégrée* dans l'énoncé, non déduite a posteriori. Tout affaiblissement (oubli de `PosDef`, formule fermée silencieusement différente, quantification  $\forall s \exists q$  remplacée par  $\exists q \forall s$ ) est attrapé par le corpus adverse Q2 (§6.2).

## 2.2 Théorème Q3 — équivalence Hamilton–Jacobi sous stabilité régulière

### 2.2.1 Énoncé informel (Feynman-reformulé)

*Paper A §6.* Sous : (H0)  $\varphi \in C^2$ ; (H1) stabilité ponctuelle Q1 en  $\theta$ ; (H2) minimalité de  $\{\varphi_1, \dots, \varphi_r, \mathbf{1}\}$ ; (H3) régularité  $C^1$  de  $\sigma \mapsto \hat{\theta}(\sigma)$ ; (H4) sous-gaussianité uniforme en  $t$ ; l'existence d'un chemin  $t \mapsto \theta_t$  vérifiant

$$\dot{\theta}_t^\top \varphi(x) + c(t) = -\frac{1}{2} \|(\nabla \varphi(x)) \theta_t\|^2 + \frac{1}{2} \theta_t^\top \Delta \varphi(x), \quad \forall x, \forall t > 0, \quad (6)$$

avec  $c(t) = \frac{d}{dt} \log Z_\theta$ , est équivalente à la *stabilité régulière* en  $\theta$ . Sans (H2)–(H3)–(H4) l'équivalence est fautive ; Paper A documente trois contre-exemples.

### 2.2.2 Contrat LEAN 4

Le module `lean/EFS/EFS/Q3.lean` ancre cet énoncé par une chaîne de structures (chacune est un type dépendant) :

```

/-- Hypothesis (H0) : the statistic is C^2. -/
structure Statistic (d r : Nat) where
  comp : Fin r -> Point d -> Real
  hC2 : ∀i, ContDiff Real (2 : WithTop Nat∞) (comp i)

/-- Hypothesis (H2) : minimality, i.e. linear independence of {φ1, ..., φr, 1}. -/
def Statistic.Minimal (φ : Statistic d r) : Prop :=
  LinearIndependent Real φ.withOne

```

```

/-- Hypothesis (H3) : C1 path, with initial condition  $\theta_0 = \theta$ . -/
structure C1Path (r : Nat) (θ : Fin r -> Real) where
  θ_t : Real -> Fin r -> Real
  init : θ_t 0 = θ
  hasDerivAt : ∀t > 0, ∀i, ∃v, HasDerivAt (fun s => θ_t s i) v t
  continuous_right : ∀i, Continuous (fun s => θ_t s i)

/-- Definition 2.3 : regular stability, bundling (H1)+(H3)+(H4). -/
structure RegularStable (φ : Statistic d r) (Z : LogPartition φ)
  (θ : Fin r -> Real) where
  pointwise : PointwiseStable φZ θ
  path : C1Path r θ
  consistent : ∀σ > 0, ConvolutionReproducesAt φZ θσ(path.θ_t (σ2/2))
  sub_gaussian : ∀T > 0, ∃C a, 0 < C ∧ 0 < a ∧ ∀t ∈ Set.Icc 0 T, ∀x,
    Real.exp (Z.logZ (path.θ_t t)) * density φZ (path.θ_t t) x
    ≤ C * Real.exp (a * ||x||2)

/-- Definition 2.4 : admissible path for (6). -/
structure AdmissiblePath (φ : Statistic d r) (Z : LogPartition φ)
  (θ : Fin r -> Real) where
  path : C1Path r θ
  thetaDot : Real -> Fin r -> Real
  c : Real -> Real
  c_continuous : Continuous c
  c_is_deriv_logZ : ∀t > 0, HasDerivAt (fun s => Z.logZ (path.θ_t s)) (c t) t
  thetaDot_is_deriv : ∀t > 0, ∀i, HasDerivAt (fun s => path.θ_t s i) (thetaDot t i) t
  hj_identity : ∀t > 0, ∀x, (∑ i, thetaDot t i * φ.comp i x) + c t
    = coleHopfRHS φ(path.θ_t t) x

/-- Fidelity anchor for Q3 (under minimality). -/
theorem q3_regular_stability_iff_admissible_path
  (φ : Statistic d r) (Z : LogPartition φ)
  (_hMin : φ.Minimal) (θ : Fin r -> Real) (_hθ : θ ∈ Z.dom) :
  Nonempty (RegularStable φZ θ) ↔ Nonempty (AdmissiblePath φZ θ) := by
  sorry

```

### 2.2.3 Ce que le contrat garantit

Les trois *sauts cachés* diagnostiqués en Paper A sont surfacés comme *champs de structure*, pas comme des hypothèses optionnelles :

- (H2) est un prédicat séparé `Statistic.Minimal`, donné en paramètre de `q3_regular_stability_iff_admissible_path` ; sans lui l'élaborateur refuse.
- (H3) est le champ `hasDerivAt` de `C1Path` ; l'oubli est attrapé par la pièce adverse `NonC1Path`.
- (H4) est le champ `sub_gaussian` de `RegularStable` ; l'oubli est attrapé par `MissingSubGaussian`.

Le membre droit de (6) est encapsulé dans `coleHopfRHS`, ce qui rend tout glissement de signe un échec de `rfl` (`WrongSignHJ`). L'initialisation  $\theta_0 = \theta$  est un champ `init` de `C1Path` : l'omettre casse `MissingInitialCondition`.

## 2.3 Le contrat structurel : pourquoi des *structures* plutôt que des propositions ?

Nous avons délibérément modélisé `RegularStable` et `AdmissiblePath` comme des *structures* (produits dépendants de types) plutôt que des conjonctions de propositions (`_^_`). Trois raisons :

1. **Détection mécanique des oublis.** Un `anonymousConstructor`  $\langle x_1, x_2 \rangle$  sans tous les champs provoque une erreur *ciblée* du type « `field sub_gaussian is missing in anonymous constructor of RegularStable` », qui nomme exactement l'hypothèse oubliée (H4 dans l'exemple).
2. **Documentation auto-portée.** Chaque champ porte un nom (`init`, `hj_identity`) qui correspond à une (et une seule) clause de Paper A.
3. **Composition via path partagé.** `RegularStable` et `AdmissiblePath` partagent le champ `path : C1Path r θ`. Le lien entre Q1 ponctuelle (via `ConvolutionReproducesAt`) et (6) (via `AdmissiblePath`) est *explicitement* tracé par ce partage, empêchant le glissement diagnostiqué dans `circular-q3-glissement.md`.

## 3 Architecture de la bibliothèque lean/EFS

### 3.1 Arborescence

Fichier	Contenu
<code>EFS.lean</code>	point d'entrée : imports Q2, Q3
<code>EFS/Basic.lean</code>	placeholder neutre
<code>EFS/Q2.lean</code>	Q2 : <code>CanonicalParam</code> , <code>tildeM</code> , <code>q2_stability</code>
<code>EFS/Q3.lean</code>	Q3 : <code>Statistic</code> , <code>C1Path</code> , <code>AdmissiblePath</code> , théorème
<code>EFS/Adversarial/Q2/*.lean</code>	10 pièces adverses pour Q2
<code>EFS/Adversarial/Q3/*.lean</code>	10 pièces adverses pour Q3
<code>corpus-manifest.toml</code>	manifeste + cibles <code>expect = "fail"</code>
<code>scripts/verify-corpus.sh</code>	vérificateur, sortie non-zéro $\Leftrightarrow$ régression
<code>InventoryQ2.md</code> , <code>InventoryQ3.md</code>	cartographie MATHLIB
<code>lakefile.toml</code> , <code>lean-toolchain</code>	pin LEAN 4 + MATHLIB

### 3.2 Imports MATHLIB — les briques réellement utilisées

#### 3.2.1 Cas Q2 (module EFS/Q2.lean)

- `Mathlib.LinearAlgebra.Matrix.PosDef` — `Matrix.PosDef`, `PosDef.add`, `PosDef.inv`, `posDef_diagonal_iff`. Ces primitives couvrent *toute* l'algèbre SPD nécessaire pour `shift_posDef`, `tildeM_posDef`.
- `Mathlib.LinearAlgebra.Matrix.NonsingularInverse` — `Matrix.mul_nonsing_inv`, `isUnit_iff_isUnit_det`.
- `Mathlib.LinearAlgebra.Matrix.Symmetric` — `Matrix.IsSymm`.
- `Mathlib.Probability.Distributions.Gaussian.Multivariate` — `multivariateGaussian`, `charFun_multivariateGaussian`.
- `Mathlib.MeasureTheory.Group.Convolution` — `Measure.conv`, `conv_comm`, `conv_assoc`.

#### 3.2.2 Cas Q3 (module EFS/Q3.lean)

- `Mathlib.Analysis.Calculus.ContDiff.Defs` — `ContDiff` pour exprimer (H0).
- `Mathlib.Analysis.Calculus.Deriv.Basic` — `HasDerivAt` pour (H3).
- `Mathlib.Analysis.InnerProductSpace.EuclideanDist` — structure de l'espace ambiant.
- `Mathlib.LinearAlgebra.LinearIndependent.Defs` — (H2) `LinearIndependent`.
- `Mathlib.MeasureTheory.Measure.MeasureSpace` — support des mesures pour le pairage avec Q2.

Notable par son *absence* (GAP) : aucun module `Mathlib.Analysis.Calculus.Laplacian` en v4.29.0, aucun `Mathlib.PDE.Heat`. Voir §5.

### 3.3 Dépendances internes

Source	Cible	Nature
<code>q2_stability</code>	<code>tildeM_posDef</code>	lemme d'image
<code>q2_stability</code>	<code>woodbury_commutative</code>	cœur algébrique
<code>q2_stability</code>	<code>multivariateGaussian_conv</code> (upstream gap)	identification mesure
<code>tildeM_posDef</code>	<code>shift_posDef</code>	positivité du décalage
<code>reparam_injective</code>	<code>woodbury_commutative</code>	identifiabilité
<code>q3_regular_stab_iff_*</code>	<code>coleHopf_density_heat_iff</code>	forward + reverse
<code>q3_regular_stab_iff_*</code>	<code>heat_equation_gaussian_convolution</code>	reverse (Tychonoff/Widder)
<code>q3_regular_stab_iff_*</code>	<code>Statistic.Minimal</code>	identifie $\theta_t$

Le graphe de dépendances est *acyclique* par construction (les structures sont *en dessous* des théorèmes qui les consomment), ce qui est la garantie formelle contre les circularités diagnostiquées en Paper A.

### 3.4 Régression guard : l'invariant 10 sorry + 20 rejects

**Fait.** Un build sain de la bibliothèque `lean/EFS` produit *exactement 10 warnings declaration uses 'sorry'* :

- Q2 : 6 warnings sur `shift_posDef`, `woodbury_commutative`, `tildeM_posDef`, `tildeM_lt_inv_s`, `reparam_injective`, `q2_stability`.
- Q3 : 4 warnings sur `coleHopf_density_heat_iff`, `regularStable_to_admissible`, `admissible_to_regularStable`, `q3_regular_stability_iff_admissible_path`.

En plus, `./scripts/verify-corpus.sh` doit produire *20 rejects sur 20*. Toute dérive (9 ou 11 warnings ; un reject qui passe silencieusement) est une régression.

## 4 Ce qui a été facile en LEAN 4 — et ce qui a été dur

### 4.1 Facile : l'algèbre matricielle SPD

Le cône SPD, l'inverse d'une matrice symétrique définie positive, la somme de deux matrices SPD, le changement de variable de la convolution gaussienne dans l'espace de Fourier : tout cela existe en `MATHLIB v4.29.0` sous une forme *directement consommable*. La structure `CanonicalParam` s'écrit en six lignes ; les lemmes `shift_posDef` et `tildeM_posDef` demandent essentiellement l'assemblage `PosDef.add + posDef_diagonal_iff` avec l'hypothèse  $\sigma^2 > 0$ .

**Remarque 4.1** (Pin SPD). La notation `MATHLIB` utilise `A.PosDef` (méthode) plutôt que `A ∈ SPD` (ensemble). Cela *simplifie* la formalisation : une structure `CanonicalParam` embarque `hM_posDef` comme *champ*, ce qui interdit par construction la création d'un `CanonicalParam` sans preuve SPD. La pièce adverse `MissingSPD` (§6.2) exploite exactement ce blindage structurel.

### 4.2 Facile : la continuité et la différentiabilité des chemins

`ContDiff`, `HasDerivAt`, `Continuous` : la chaîne de régularité nécessaire pour (H3) est directement disponible. La structure `C1Path` se contente de composer ces primitives avec la condition initiale `init :  $\theta_t 0 = \theta$`  ; aucune infrastructure PDE n'est sollicitée à ce niveau.

### 4.3 Dur : le laplacien général sur $\mathbb{R}^d$

MATHLIB v4.29.0 définit `iteratedFDeriv` (différentielles itérées multilinéaires) et `second_derivative_symmetric` (symétrie du Hessian en  $C^2$ ), mais *pas* d'opérateur laplacien  $\Delta f(x) = \sum_i \partial_i^2 f(x)$  comme scalaire explicite. La définition naturelle via une base orthonormale,

$$\Delta f(x) := \sum_{i=1}^d (D^2 f(x))(e_i, e_i),$$

doit être accompagnée :

- d'un *lemme d'invariance* `laplacian_orthonormalBasis_invariant` par changement de base orthonormale,
- de règles algébriques `laplacian_sum`, `laplacian_smul`,
- et de l'identité produit `laplacian_exp_neg_linear` :

$$\Delta(e^{-\theta^\top \varphi}) = e^{-\theta^\top \varphi} (\|(\nabla \varphi)\theta\|^2 - \theta^\top \Delta \varphi).$$

Dans l'état actuel, EFS/Q3.lean ligne 187 déclare

```
noncomputable def laplacian (_f : Point d -> Real) (_x : Point d) : Real := 0
```

- un *stub sémantiquement inerte* : le type compile, mais toute tentative de prouver une propriété non triviale du laplacien est vide. La pièce adverse `SilentHeatFactor` (§6.3) exploite cette inertie pour montrer que le facteur  $\frac{1}{2}$  de (6) *ne peut pas* être silencieusement avalé dans le stub.

### 4.4 Dur : l'équation de la chaleur pour la convolution gaussienne

Le lemme

$$\partial_t(p * \gamma_{\sqrt{2t}})(x) = \frac{1}{2} \Delta_x(p * \gamma_{\sqrt{2t}})(x), \quad \forall t > 0, \forall x \in \mathbb{R}^d, \quad (1)$$

est la solution fondamentale de Weierstrass, absente de MATHLIB v4.29.0 en forme ponctuelle. Les briques existent : `fourierIntegral_gaussian`, `Measure.conv`, la régularité  $C^\infty$  de la gaussienne ; mais la dérivation sous l'intégrale requiert `hasDerivAt_integral_of_dominated_loc_of_deriv_le` avec des bornes de domination écrites à la main.

Dans EFS/Q3.lean nous avons un stub `True` :

```
theorem heat_equation_gaussian_convolution (_p : Point d -> Real) : True := by
  trivial
```

qui pinne uniquement la *signature attendue* (documentée en commentaire). Le probe engineer devra soit prouver le lemme réel, soit l'axiomatiser explicitement dans un module EFS/HJ/Axioms.lean (voir §7).

### 4.5 Dur : l'équivalence Cole–Hopf dans le langage LEAN 4

`coleHopf_density_heat_iff` affirme que, pour  $q > 0$ ,  $\partial_t q = \frac{1}{2} \Delta q$  équivaut à  $\partial_t g = \frac{1}{2} \Delta g - \frac{1}{2} \|\nabla g\|^2$  pour  $g := -\log q$ . La preuve mathématique est élémentaire (chain rule + substitution) mais requiert, dans notre stub, *d'abord* le laplacien réel. Le plat de pâtes n'est pas logique mais combinatoire : sans `laplacian`, la *RHS* de l'équivalence a une syntaxe sans sémantique.

### 4.6 Dur : unicité de la chaleur (Widder / Tychonoff)

Le sens ( $\Leftarrow$ ) de Q3 reconstruit  $p_{\theta_t} = p_\theta * \gamma_\sigma$  à partir de l'équation (6), via Cole–Hopf puis unicité de la solution de l'équation de la chaleur. MATHLIB v4.29.0 n'a pas de théorème d'unicité parabolique en forme *Widder 1944* (positivité densitaire) ni *Tychonoff 1935* (sous-gaussianité). C'est la brique qui *matérialise* (H4) dans la preuve du sens réverse.

## 4.7 Récapitulatif : carte des obstacles

Objet	Statut MATHLIB v4.29.0	Action
$\text{Sym}_d^{++}$ , <code>PosDef.add</code> , <code>PosDef.inv</code>	disponible	consommer
<code>multivariateGaussian</code> , <code>charFun</code>	disponible	consommer
<code>ContDiff</code> , <code>HasDerivAt</code>	disponible	consommer
<code>gaussianReal_conv_gaussianReal</code>	disponible (1D)	généraliser
<code>multivariateGaussian_conv</code>	absent	upstream candidate
Laplacien général sur espace euclidien	absent	upstream candidate
Équation de la chaleur ponctuelle	absent	upstream candidate
Unicité parabolique Widder/Tychonoff	absent	upstream candidate

## 5 Esquisses amont pour MATHLIB

Nous détaillons les trois obligations les plus rentables à contribuer en *amont*. Chaque esquisse donne le nom proposé, la signature, un schéma de preuve, et les dépendances MATHLIB existantes. Ces trois items sont tracés dans `docs/lore/lean-unproved.md` comme `upstream-sketch`.

### 5.1 `multivariateGaussian_conv_multivariateGaussian`

**Cible.**

```
lemma multivariateGaussian_conv_multivariateGaussian
  {ι : Type*} [Fintype ι]
  (μ₁ μ₂ : EuclideanSpace Real ι) (S₁ S₂ : Matrix ιReal)
  (h₁ : S₁.PosSemidef) (h₂ : S₂.PosSemidef) :
  (multivariateGaussian μ₁ S₁).conv (multivariateGaussian μ₂ S₂)
  = multivariateGaussian (μ₁ + μ₂) (S₁ + S₂)
```

**Schéma.** Fonction caractéristique :  $\widehat{\gamma}_\mu^S(\xi) = \exp(i\mu^\top \xi - \frac{1}{2}\xi^\top S\xi)$ , puis unicité de la mesure par sa fonction caractéristique (`Measure.ext_iff_charFun` présent en v4.29.0). Le produit de deux fonctions caractéristiques gaussiennes est la fonction caractéristique de leur somme.

**Bénéfice en aval.** Tout le volet *identification mesure* du théorème Q2 se réduit à trois lignes : écrire  $p_\theta$  comme `multivariateGaussian` via complétion de carré, appliquer ce lemme, réidentifier  $\widehat{M}$  via Woodbury.

### 5.2 `laplacian sur un espace euclidien de dim finie`

**Cible.**

```
noncomputable def laplacian
  (E : Type) [NormedAddCommGroup E] [InnerProductSpace Real E]
  [FiniteDimensional Real E]
  (f : E -> Real) (x : E) : Real :=
  let b := stdOrthonormalBasis Real E
  Finset.univ.sum (fun i => iteratedFDeriv Real 2 f x (basisPair b i))

lemma laplacian_orthonormalBasis_invariant (f : E -> Real) (x : E)
  (b1 b2 : OrthonormalBasis ιReal E) :
  Finset.univ.sum (fun i => iteratedFDeriv Real 2 f x (pair b1 i))
  = Finset.univ.sum (fun i => iteratedFDeriv Real 2 f x (pair b2 i))

lemma laplacian_smul, laplacian_add, laplacian_const, laplacian_exp_neg_linear
```

**Schéma.** L'invariance découle de la symétrie du Hessien (`second_derivative_symmetric`) combinée avec la conservation de la trace :  $\sum_i D^2 f(x)(e_i, e_i) = \text{tr Hess } f(x)$ . La trace est invariante par conjugaison par une matrice orthogonale, ce qui est exactement le changement de base orthonormale.

**Identité produit.** Pour  $f = -\theta^\top \varphi$ ,  $g = e^f$  :  $\nabla g = g \nabla f$ ,  $D^2 g = g (\nabla f)(\nabla f)^\top + g D^2 f$ , donc  $\Delta g = g (\|\nabla f\|^2 + \Delta f)$ . Appliqué à  $f = -\theta^\top \varphi$  on obtient la formule requise pour Q3.

**Bénéfice en aval.** Tout EFS/Q3.lean cesse d'utiliser le stub `:= 0` et devient une vraie bibliothèque d'analyse harmonique.

### 5.3 heat\_equation\_gaussian\_convolution

**Cible.**

```
theorem heat_equation_gaussian_convolution
  (d : Nat) (p : EuclideanSpace Real (Fin d) -> Real)
  (hp : Integrable p) (hp_cont : Continuous p) :
  forall (t : Real), 0 < t -> forall (x : EuclideanSpace Real (Fin d)),
    HasDerivAt (fun s => (conv p (gaussianKernel (Real.sqrt (2*s)))) x
      ((1/2 : Real) * laplacian
        (fun y => (conv p (gaussianKernel (Real.sqrt (2*t)))) y) x)
      t
```

**Schéma.** Fourier :  $\widehat{p * \gamma_{\sqrt{2t}}}(\xi) = \hat{p}(\xi) e^{-t\|\xi\|^2}$ . Dériver en  $t$  :  $\partial_t = -\|\xi\|^2 \cdot = \frac{1}{2} \widehat{\Delta}(\cdot)$ . Inverser et dériver sous l'intégrale via `hasDerivAt_integral_of_dominated_loc_of_deriv_le`.

**Bénéfice en aval.** Les deux directions de `q3_regular_stability_iff_admissible_path` deviennent consommables :  $(\Rightarrow)$  dérive (6) par Cole–Hopf,  $(\Leftarrow)$  la reconstruit via Widder 1944 sous positivité.

### 5.4 Deux contributions de seconde priorité

- **Widder / Tychonoff parabolique.** Un module `Mathlib.MeasureTheory.PDE.HeatUniqueness` est plus ambitieux et n'est pas bloquant tant que l'escape hatch axiomatique de §7 tient.
- **Intégrale gaussienne multivariée explicite.**  $\int \exp(-\frac{1}{2}x^\top Ax + b^\top x) dx = (2\pi)^{d/2} (\det A)^{-1/2} \exp(\frac{1}{2}b^\top A^{-1}b)$ . Elementary mais absent en v4.29.0 ; consommé indirectement par `charFun_multivariateGaussian`.

## 6 Le corpus adverse — vingt fichiers à faire échouer

### 6.1 Principe — et pourquoi il est différent d'une simple suite de tests

Une suite de tests vérifie que ce qui *devrait compiler* compile. Un corpus adverse vérifie que ce qui *ne devrait pas* ne compile *surtout pas*. Chaque fichier `EFS/Adversarial/{Q2,Q3}/*.lean` est une reformulation intentionnellement affaiblie du théorème correspondant, choisie pour isoler un mode de défaillance : quantificateur inversé, hypothèse supprimée, formule fermée tacitement modifiée, glissement de signe.

Le vérificateur `./scripts/verify-corpus.sh lit corpus-manifest.toml`, compile chaque entrée avec `lake env lean`, et exige un code de retour *non nul*. Sortie 0 iff toutes les pièces sont rejetées.

## 6.2 Taxonomie Q2 (dix entrées)

#	Slug	Mode de défaillance
1	OffByOneShape	$M \in \mathbb{R}^{d \times d}$ et $b \in \mathbb{R}^{d+1}$ — les multiplications refusent l'unification.
2	MissingSPD	Le lemme <code>tildeM_posDef</code> est appliqué sans témoin <code>PosDef</code> — placeholders <code>?_</code> restent ouverts.
3	SilentReparam	La reparamétrisation est silencieusement remplacée par $\tilde{M} = M + \sigma^2 I_d$ (faux) ; <code>rfl</code> échoue.
4	QuantifierInversion	$\exists q, \forall s$ au lieu de $\forall s, \exists q$ — le placeholder dépend d'une variable non encore introduite.
5	SigmaZeroBoundary	L'hypothèse $\sigma^2 > 0$ est remplacée par $\sigma^2 \geq 0$ ; <code>shift 0</code> est singulière.
6	DegeneratePhi	$\varphi = 0$ — le Jacobien perd son rang ; le théorème devient <code>vacuous</code> .
7	AxiomSmuggling	<code>theorem</code> remplacé par <code>axiom</code> — <code>#print axioms</code> expose.
8	TypeUniverseMismatch	Matrice sur $\mathbb{N}$ au lieu de $\mathbb{R}$ — pas d'unification avec les API <code>MATHLIB</code> .
9	MissingMinimality	$T$ traité comme unique ; antisymétrie ignorée — contrat brisé avec §2.4 de Paper A.
10	WrongTychonoff	Invocation d'une compacité de produit inexistante sur $\text{Sym}_d$ .

Par conception, chaque entrée touche un objet *différent* : la forme matricielle, l'hypothèse SPD, la formule fermée elle-même, l'ordre des quantificateurs, le bord du domaine  $\sigma^2 = 0$ , la dégénérescence de  $\varphi$ , la distinction théorème vs axiome, l'univers de types, la redondance antisymétrique, et un recours incorrect à Tychonoff.

## 6.3 Taxonomie Q3 (dix entrées)

#	Slug	Hyp. cachée	Mode de défaillance
11	MissingMinimality	(H2)	<code><math>\varphi</math>.Minimal</code> invoqué par <code>trivial</code> (faux).
12	NonC1Path	(H3)	Anonymous constructor de <code>C1Path</code> sans champ <code>hasDerivAt</code> .
13	MissingSubGaussian	(H4)	Anonymous <code>RegularStable</code> sans <code>sub_gaussian</code> .
14	QuantifierInversion	$\forall x \forall t$	<code>?x</code> échappe à sa portée.
15	WrongSignHJ	Cole–Hopf	Signe inversé sur <code>coleHopfRHS</code> — <code>rfl</code> échoue.
16	TypeShapeMismatch	$\dim r$	<code>Fin r -&gt; Real</code> donné à <code>Statistic d (r+1)</code> .
17	AxiomSmuggling	(H2)	<code>decide</code> appliqué à <code>LinearIndependent</code> (non décidable).
18	SilentHeatFactor	$\sigma^2 = 2t$	Facteur $1 \cdot \Delta$ au lieu de $(1/2) \cdot \Delta$ .
19	MissingInitialCondition	$\theta_0 = \theta$	<code>init := rfl</code> échoue quand $\theta_t(0) = 0 \neq \theta$ .
20	CircularPointwiseToHJ	glissement	<code>assumption</code> ne construit pas <code>AdmissiblePath</code> depuis <code>PointwiseStable</code> seul.

**Remarque capitale.** Les entrées #11–13 encodent *exactement* les trois sauts cachés diagnostiqués par Feynman (Paper A §6.1). L'entrée #20 est leur *dual collectif* : elle tente de conclure (6) depuis la seule Q1 ponctuelle, sans aucune des trois hypothèses supplémentaires. Le rejet de ces quatre pièces par le vérificateur *est* la formalisation mécanique de `circular-q3-glisement.md`.

## 6.4 Un exemple complet : SilentReparam

Pour illustrer la mécanique, voici la pièce #3 en entier :

```
import EFS.Q2
open scoped Matrix
namespace EFS.Adversarial.Q2
example (d : Nat) (p : EFS.Q2.CanonicalParam d) (s : Real) :
  p.tildeM s = p.M + s • (1 : Matrix (Fin d) (Fin d) Real) := by
  rfl -- expected error: not definitionally equal
end EFS.Adversarial.Q2
```

Un relecteur humain, même attentif, peut laisser passer «  $\widetilde{M} = M + \sigma^2 I_d$  » dans une preuve : la formule *ressemble* à la bonne identité (elle est exacte pour  $d = 1$ ,  $M = 0$ , ou  $\sigma^2 = 0$ ). Mais l'élaborateur de LEAN 4 ne laisse pas passer :  $M(I_d + \sigma^2 M)^{-1} \neq M + \sigma^2 I_d$  par définition. Le corpus *mémore* ce test pour le prochain probe engineer.

## 6.5 Un deuxième exemple : WrongSignHJ

```
example (d r : Nat) (φ : Statistic d r) (θ : Fin r -> Real) (x : Point d) :
  coleHopfRHS φ θ x
  = (1/2 : Real) * jacobianSqNorm φ θ x
  - (1/2 : Real) * ∑ i, θ i * laplacian (φ.comp i) x := by
  rfl -- sign flip destroys rfl
```

Un signe inversé dans le membre droit de (6) transformerait l'équation de la chaleur en équation de la chaleur *rétrograde* — mal posée ! La pièce est courte, mais le diagnostic qu'elle rend est très précis : toute dérive du générateur Cole–Hopf  $\mathcal{L}f = \frac{1}{2}\|\nabla f\|^2 - \frac{1}{2}\Delta f$  en son opposé explose à la compilation.

# 7 sorry résiduels et chemin vers 0

## 7.1 Inventaire des dix sorry

Identifiant	Tag	Contenu (résumé)
shift_posDef	upstream-sketch	Somme de SPD est SPD
woodbury_commutative	upstream-sketch	$(M^{-1} + sI_d)^{-1} = M(I_d + sM)^{-1}$
tildeM_posDef	upstream-sketch	Image du cône SPD
tildeM_lt_inv_s	narrative-only	Borne $\widetilde{M} \prec s^{-1}I_d$
reparam_injective	upstream-sketch	Injectivité via minimalité
q2_stability	narrative-only	Identité mesure Q2
coleHopf_density_heat_iff	upstream-sketch	Cole–Hopf densité/chaleur
regularStable_to_admissible	narrative-only	( $\Rightarrow$ ) de Q3
admissible_to_regularStable	narrative-only	( $\Leftarrow$ ) de Q3
q3_regular_stability_iff_*	narrative-only	Équivalence Q3

## 7.2 L'escape hatch axiomatique

Si le probe engineer juge les trois items *upstream-sketch* hors budget, la *voie d'évasion* documentée dans docs/lore/lean-unproved.md est :

1. Créer un module EFS/HJ/Axioms.lean qui axiomatise laplacian, heat\_equation\_gaussian\_convolution et coleHopf\_density\_heat\_iff avec leur signature et leurs *règles algébriques* attendues.

2. Remplacer les `sorry` de `regularStable_to_admissible` et `admissible_to_regularStable` par de vraies preuves qui *consomment* ces axiomes.
3. Ajouter un test `#print axioms` dans la suite qui pinne la surface axiomatique. Toute dérive élargit observablement la surface.

Ce chemin *améliore strictement* la fidélité : chaque axiome est exposé dans la signature du noyau, jamais caché derrière `sorry`. C'est la forme *honnête* du compromis budget.

### 7.3 Statut UNPROVABLE\_IN\_BUDGET

Aucun des dix `sorry` n'est étiqueté UNPROVABLE\_IN\_BUDGET. Tous réduisent à du travail MATHLIB mécanique (§5) ou à un axiome proprement déclaré.

## 8 Discussion — preuve informelle et preuve formelle

### 8.1 Ce que Paper A voit et que Paper B ne voit pas

Paper A *démontre* : la fermeture algébrique ( $\dagger$ ), la borne de degré  $D \leq 2$ , la non-existence de  $\varphi$  polynomiaux stables non quadratiques, l'équivalence Hamilton–Jacobi sous stabilité régulière. Ces démonstrations exploitent :

- Le comptage de degrés dominants dans  $\|\nabla f\|^2 \in \mathcal{A}$  (§4.4 Paper A) : LEAN 4 n'a pas d'oracle de degré polynomial directement branché sur `iteratedFDeriv` ; le formaliser exigerait soit `MvPolynomial` + lien explicite avec les dérivées, soit un argument ad-hoc.
- L'argument Liouville généralisé (§7.1 Paper A) : requiert `Entire` + croissance polynomiale globale, domaine Mathlib en chantier.
- L'interprétation Wasserstein/JKO (§3.8 Paper A) : `Mathlib.MeasureTheory.ProbabilityMassFunction` et la distance de Wasserstein existent en forme élémentaire ; le flot de gradient JKO n'est pas disponible.

*Paper B ne les formalise pas* ; ils restent du ressort de Paper A.

### 8.2 Ce que Paper B voit et que Paper A ne voit pas

- **La redondance antisymétrique de  $\varphi = (\text{vec}(xx^\top), x)$ .** Paper A la mentionne (§2.4) ; Paper B la rend *visible dans le type* : `CanonicalParam` utilise  $M \in \text{Sym}_d^{++}$ , pas  $T \in \mathbb{R}^{d \times d}$ , ce qui ferme la redondance *par construction*.
- **Le glissement de quantificateurs.** La pièce `QuantifierInversion` (Q2 #4 & Q3 #14) détecte mécaniquement un ordre  $\exists q, \forall s$  que l'œil humain confond avec  $\forall s, \exists q$ .
- **La dépendance cachée au laplacien.** Sans le laplacien formel, toutes les identités de Q3 sont *syntactiquement là mais sémantiquement inertes* ; le probe engineer est forcé de matérialiser la dépendance avant de pouvoir conclure.
- **Les axiomes utilisés.** `#print axioms q2_stability` donne la liste exacte de ce sur quoi le théorème *repose*. Aucun énoncé *au kilomètre* de MATHLIB ne peut se glisser silencieusement.

### 8.3 Rapport coût/bénéfice

Pour Q2, la formalisation est un multiplicateur de fidélité à coût bas : les six lemmes `sorry` se résolvent en  $\sim 200$  lignes chacun.

Pour Q3, la formalisation est un multiplicateur de fidélité à coût intermédiaire : les quatre `sorry` se résolvent soit via les trois upstream sketches (§5,  $\sim 500$  lignes MATHLIB par item), soit via l'échappatoire axiomatique ( $\sim 100$  lignes, surface axiomatique pinnée).

Dans les deux cas, le corpus adverse est *amorti* : il vérifie que l'énoncé reste pinné quoi que fasse le probe engineer.

## 8.4 Leçon méthodologique — le *fidelity anchor*

La discipline que nous avons utilisée se résume à trois règles :

1. **Pin avant prouver.** Un énoncé type-checkable (même avec `sorry`) qui capture *exactement* la phrase mathématique est un *fidelity anchor*.
2. **Structures avant conjonctions.** `RegularStable` est une structure à quatre champs ; chaque oubli est nommément détecté par l'élaborateur.
3. **Corpus adverse avant suite de tests.** Les pièces qui doivent *échouer* ont une valeur diagnostique plus grande que celles qui doivent *réussir*.

Nous pensons que cette discipline est transférable à toute formalisation d'un théorème multi-hypothèses, et en particulier aux équivalences de type Hamilton–Jacobi où les hypothèses de régularité et d'uniformité sont susceptibles de glisser.

## 9 Conclusion

La bibliothèque `lean/EFS` (v0.1.0, pin `MATHLIB v4.29.0`) ancre les théorèmes Q2 et Q3 de Paper A en dix énoncés type-checkables et vingt pièces adverses qui doivent toutes être rejetées. Dans son état actuel, elle *pin* la forme du résultat : chaque hypothèse cachée est surfacée, chaque circularité est barrée par construction, chaque `sorry` est tracé dans `docs/lore/lean-unproved.md`. Le chemin vers 0 `sorry` passe par trois contributions `MATHLIB` identifiées (`multivariateGaussian_conv`, `laplacian`, `heat_equation_gaussian_convolution`), ou par un escape hatch axiomatique explicitement documenté.

*Paper A* donne la preuve mathématique ; *Paper B* la verrouille mécaniquement. L'un sans l'autre est incomplet : une preuve sans ancrage dérive narrativement, un ancrage sans preuve est sans contenu. Leur composition est la version la plus stable du résultat dont nous disposons aujourd'hui.

## Remerciements et provenance

La formalisation Q2 est due à `task-20260416-edb4` (`lean-engineer`), Q3 à `task-20260416-8478` (`lean-engineer` + `probe-engineer`). Les corpus adverses respectifs sont produits par les mêmes tâches (rôle `red-team-mathematician`). Paper A est une synthèse de `task-20260416-a7b4` à partir du panel `delib-20260416-102e` (`hawking`, `shannon`, `von-neumann`, `feynman`, `wheeler`). Le présent papier est le livrable de `task-20260416-238b`.

## Références

- [Paper A] E. Sérié. *Stabilité des familles exponentielles par convolution gaussienne : caractérisation quadratique, fermeture de Cole–Hopf, et équivalence de Hamilton–Jacobi*. Paper A, Noogram Labs, 2026. Source : `docs/paper/paper.pdf`.
- [Étienne] É. Boursier. *Stabilité des familles exponentielles par convolution gaussienne et équation de Hamilton–Jacobi*, note de problème, 17 avril 2026. Source : `docs/problem.pdf`.
- [Mathlib] `MATHLIB`, `leanprover-community`, version `v4.29.0` (2026). [https://leanprover-community.github.io/mathlib4\\_docs/](https://leanprover-community.github.io/mathlib4_docs/).
- [Hopf] E. Hopf. *The Partial Differential Equation  $u_t + uu_x = \mu u_{xx}$* . *Comm. Pure Appl. Math.*, **3** (1950), 201–230.
- [Cole] J. D. Cole. *On a Quasi-Linear Parabolic Equation Occurring in Aerodynamics*. *Quart. Appl. Math.*, **9** (1951), 225–236.

- [Tychonoff] A. N. Tychonoff. *Théorèmes d'unicité pour l'équation de la chaleur*. Mat. Sb., **42** (1935), 199–216.
- [Widder] D. V. Widder. *Positive Temperatures on an Infinite Rod*. Trans. Amer. Math. Soc., **55** (1944), 85–95.
- [BGL] D. Bakry, I. Gentil, M. Ledoux. *Analysis and Geometry of Markov Diffusion Operators*. Grundlehren der mathematischen Wissenschaften, vol. 348. Springer, 2014.
- [Brown] L. D. Brown. *Fundamentals of Statistical Exponential Families*. IMS Lecture Notes — Monograph Series, vol. 9, 1986.
- [Amari] S. Amari, H. Nagaoka. *Methods of Information Geometry*. AMS Translations of Mathematical Monographs, 2000.
- [Stam] A. J. Stam. *Some Inequalities Satisfied by the Quantities of Information of Fisher and Shannon*. Information and Control, **2** (1959), 101–112.
- [JKO] R. Jordan, D. Kinderlehrer, F. Otto. *The Variational Formulation of the Fokker–Planck Equation*. SIAM J. Math. Anal., **29** (1998), 1–17.