==========

SYNTHÈSE INTERNE DU PROJET

Dans la vraie vie, la tomographie quantique vise à comparer l'état idéal que le scientifique essaye de fabriquer, avec l'état réel. Les mesures permettent de donner les mesures moyennes. Mais cela ne nous permet pas de recalculer directement les variables d'état (theta et phi) par le calcul (la relation que x = cos(theta)sin(phi) pour deux raisons : On utilise la méthode du MLE pour reconstruire l'état à partir des valeurs des mesures pour deux raisons : -le bruit statistique fait que les valeurs moyennes de X,Y et Z ne sont plus physiques, c'est à dire X_chap^2 + Y_chap^2 + Z_chap^2 !=1. Le MLE prend ça en compte, sinon les thêtas et phi seraiebnt incohérents.

- le relation n'étant pas linéaire (car sinusoïdale), on ne peut pas passer d'une valeur moyenne par transformation linéaire. La tomograaphie quantique à base de ML est un axe de recherche actuel, car ça permet de passer outre ces problématiques, comme le MLE.

◇ 1. Objectif global Notre but est de comparer deux méthodes de reconstruction d'état pour un qubit : (a) la tomographie classique (MLE), (b) la tomographie assistée par Machine Learning, afin d'étudier : • la précision, • la robustesse au bruit, • la dépendance au nombre de shots, • le coût computationnel.

Nous travaillons d'abord sur le cas 1 qubit, avec possibilité d'étendre à plusieurs qubits plus tard (où le MLE devient très coûteux).

◇ 2. Notions cruciales : état idéal, état réel, mesure bruitée Dans notre simulation, il faut distinguer trois objets :

(1) ÉTAT IDÉAL :

- l'état pur que nous voulons préparer.
- défini par les paramètres (θ, φ) sur la sphère de Bloch.
- vecteur de Bloch normé : ||(X,Y,Z)|| = 1.

(2) ÉTAT RÉEL :

- l'état effectivement produit AVANT LES MESURES.
- il peut être : → identique à l'état idéal (si aucun bruit physique même si il y a le bruit statistique (car l'état reste pure)), → ou contracté par un bruit physique simulé (donc mixte).
- c'est cet état que les méthodes de tomographie cherchent à reconstruire.

(3) MESURES BRUITÉES :

- generated via tirages binomiaux à partir des probabilités de l'état réel.
- bruit purement statistique : il ne modifie PAS l'état réel.

Ainsi : • le bruit physique transforme l'état réel, • le bruit statistique transforme uniquement les mesures, • le MLE/ML tente de retrouver l'état réel à partir de ces mesures bruitées.

◇ 3. Ce que signifie "état réel" dans notre projet

Cas A — PAS de bruit physique (seulement bruit statistique) : → état réel = état idéal (pur) → les labels ML = (X_ideal, Y_ideal, Z_ideal)

Cas B — Bruit physique simulation par shrink anisotrope : → état réel = vecteur shrinké (X_real, Y_real, Z_real) → il reste dans la sphère de Bloch (physique) → les labels ML = (X_real, Y_real, Z_real)

Dans les deux cas : • l'état réel est parfaitement déterminé dans la simulation, même si le shrink est aléatoire (une fois les paramètres tirés, l'état est défini). • les données de mesure ne sont jamais les labels : ce sont les features bruitées.

◇ 4. Comment nous simulons le bruit physique (decoherence)

Nous utilisons un modèle "jouet" mais cohérent :

- avec probabilité = decoherence_level, l'état est bruité.
- la contraction (shrink) appliquée sur (X,Y,Z) est : X_real = factor_X * X_ideal Y_real = factor_Y * Y_ideal Z_real = factor_Z * Z_ideal
- où les factors sont générés à partir de : base_factor = 1 - strength (avec strength ≤ decoherence_level) anisotropy = tirage uniforme [0.5, 1.5] factors = clip(base_factor * anisotropy, 0, 1)
- Résultat : l'état réel est une contraction anisotrope dans la sphère de Bloch.

Ce bruit n'est pas un canal CPTP standard, mais :

- il reste toujours PHYSIQUE,
- il produit des états mixtes cohérents (respectant $x^2+y^2+z^2 \leq 1$),
- il permet de tester la robustesse du ML,
- il est contrôlable par le paramètre decoherence_level.

◇ 5. Pourquoi on peut toujours récupérer la valeur réelle

Parce que dans une simulation :

- l'état réel est construit mathématiquement et sans ambiguïté.
- même si les coefficients de shrink sont aléatoires, ils sont tirés UNE FOIS par échantillon → état déterministe.
- l'état réel est donc EXACTEMENT connu pour chaque donnée du dataset.

Le bruit statistique n'altère jamais l'état réel : il ne touche que les mesures. On peut donc utiliser (X_real, Y_real, Z_real) comme labels "parfaits".

◇ 6. Ce que le modèle ML apprend réellement

Le ML apprend : • à reconstruire l'état réel (pur ou mixte), • à partir des données de mesure bruitées, • exactement comme le ferait un estimateur statistique, • mais potentiellement plus vite ou avec moins de shots que le MLE.

Nous comparerons donc :

- l'erreur du MLE ≈ distance entre ρ_MLE et ρ_real,

- l’erreur du ML ≈ distance entre ρ_ML_pred et ρ_real.

◇ 7. Workflow final pour le dataset

Pour chaque échantillon :

1. Générer un état idéal (θ,φ).
2. Appliquer (ou pas) le bruit physique (shrink anisotrope) → état réel.
3. Calculer les probabilités théoriques des mesures X/Y/Z.
4. Tirer n_shots mesures → valeurs +1/-1 → bruit statistique.
5. Extraire les features : (⟨X⟩_mesuré, ⟨Y⟩_mesuré, ⟨Z⟩_mesuré)
6. Définir les labels = composantes de l’état réel : (X_real, Y_real, Z_real)

◇ 8. En résumé

- L’état idéal = on part de ça pour contruire le dataset, mais il ne faut surtout pas l'utiliser directement pour entrainer le ML
- L’état réel = ce qui existe physiquement après bruit (ou égal à l’idéal si aucun bruit physique).
- Le ML doit apprendre l’état réel, pas l’idéal, pas le MLE.
- Le shrink anisotrope est un bruit physique jouet mais valide.
- Le bruit statistique ne change pas l’état, seulement les mesures.
- Les labels = état réel déterministe (pur ou mixte).
- Nous comparons MLE vs ML pour reconstruire cet état réel.

README — LAB0 & LAB1 : Dataset Exploration & Classification (SVC)

Ce document décrit les deux premiers Labs du projet :
Lab0 (exploration du dataset) et **Lab1 (classification via SVC)**.
Ils servent à vérifier la validité physique des données, calibrer le pipeline, et établir des benchmarks avant la tomographie machine learning.

🎯 Objectifs généraux des Labs 0 et 1

Ces deux Labs permettent de :

- **valider la génération du dataset**,
- **explorer les propriétés des données** (plots, statistiques, bruit),
- **comprendre le comportement des mesures X/Y/Z**,
- **tester les kernels SVC comme premier benchmark ML**,
- **vérifier que la chaîne expérimentale simulée fonctionne**,
- **préparer les Labs de régression**, qui réaliseront la vraie tomographie ML.

Ces Labs **ne reconstruisent pas encore l’état**.
Ils servent de base de calibration, exactement comme dans un laboratoire.

LAB 0 — Exploration du Dataset

Objectifs du Lab 0

1. Vérifier que la fonction `generate_dataset` respecte les exigences physiques :
    - état réel correctement défini,
    - bruit statistique bien appliqué,
    - shrink anisotrope gardant l'état physique,
    - labels cohérents avec l'état réel.
2. Comprendre visuellement la structure des données :
    - distributions,
    - effet du bruit statistique,
    - effet du shrink anisotrope,
    - structure dans l'espace de Bloch.
3. Fournir des **plots explicatifs** utiles pour le rapport.
4. S'assurer que le dataset est **physiquement sain** avant d'entraîner les modèles.

LAB 1 — Classification (SVC & kernels)

Clarification essentielle : ce que la classification ne peut pas faire

La classification ne reconstruit pas l'état. Elle ne trouve pas (θ, φ). Elle n'est pas une méthode de tomographie. Elle ne produit pas la matrice de densité.

Elle sert exclusivement à reconnaitre une classe parmi un ensemble d'états discrets.

Alors à quoi sert la classification ?

1. Benchmark simple pour tester les kernels

SVC permet de comparer :

kernel linéaire

RBF

polynomial

QSVM (si intégré)

2. Vérifier la qualité du dataset

Si la classification échoue → problème dans :

la simulation des mesures,

le bruit,

les features,

les labels.

3. Étudier la robustesse au bruit

On peut mesurer l'impact :

du nombre de shots,

du shrink,

du niveau de décohérence.

Résumé

Le Lab0 valide la physique du dataset (état réel, shrink, bruit statistique). Le Lab1 teste les kernels SVC sur une tâche simple de classification.

La classification n'est PAS une forme de tomographie. Elle ne sert pas à trouver (θ, φ) ni à reconstruire la matrice de densité. Elle sert à :

- valider le dataset,
- comparer les kernels classiques et quantiques,
- étudier l'effet du bruit,
- préparer les Labs de régression.

Les Labs suivants (Lab2 et Lab3) traiteront la vraie tomographie : → reconstruction du vecteur de Bloch réel avec ML, → comparaison directe ML vs MLE.