

Rapport de Stage

Mesures des paramètres d'oscillation des neutrinos avec l'expérience DUNE

Romain Faure, sous la supervision de Leïla Haegel
romain.faure@etu.univ-lyon1.fr

Résumé

Les neutrinos sont les particules élémentaires les moins connues. La détermination de certaines de leurs propriétés permettrait d'ouvrir la voie vers la compréhension de la matière à l'échelle subatomique et de la composition de l'Univers, mais aussi l'élaboration de nouvelles théories en physique des particules. Parmi elles, leur masse et la possibilité que les neutrinos et antineutrinos se comportent différemment sont encore inconnues. L'expérience DUNE permettra de mesurer la phase δ_{CP} , l'ordre des états propres de masses et les paramètres de la matrice de mélange en étudiant les oscillations des neutrinos. À l'aide de données simulées de l'expérience DUNE, on arrive à ce jour à obtenir des estimations précises des paramètres d'oscillations θ et Δm^2 par l'utilisation de deux méthodes statistiques différentes : l'algorithme de Metropolis-Hastings et l'algorithme Nested Sampling. Le premier, reposant sur les chaînes de Markov, est déjà utilisé par la collaboration DUNE. Le Nested Sampling quant à lui, est appliqué ici pour la première fois aux estimations des paramètres d'oscillations, dans l'objectif de proposer une alternative statistique aux futures analyses de données.

Table des Matières

1	Introduction	2
1.1	Éléments de physique du neutrino	2
1.2	Oscillations des neutrinos	2
1.3	Au-delà du Modèle Standard	3
1.4	DUNE	4
2	Méthodes statistiques d'échantillonnage de distribution	4
2.1	Aperçu de statistiques Bayésiennes	4
2.2	Chaîne de Markov et méthode de Monte-Carlo	5
2.3	Algorithme Nested Sampling	5
3	Exemples d'applications de l'algorithme de Metropolis-Hastings	6
3.1	Échantillonnage d'une gaussienne	6
3.2	Estimation des paramètres de plusieurs gaussiennes à partir de données	6
4	Estimation des paramètres θ et Δm^2 à partir de données simulées de DUNE	8
4.1	Contexte	8
4.2	Résolution par algorithme de Metropolis-Hastings	9
4.3	Résolution par algorithme Nested Sampling	11
5	Conclusion et Perspectives	11
	Appendice	11

Liste des Figures

1	Échantillonnage par algorithme de Metropolis-Hastings	6
2	Données du spectre d'énergie du spectromètre	7
3	Graphique de corrélation des moyennes des quatre gaussiennes	8
4	Comparaison des taux d'interactions à courant chargé ν_μ	8
5	Visualisation en histogramme des estimations des valeurs des paramètres θ et Δm^2	10
6	Histogramme en deux dimensions des estimations des paramètres θ et Δm^2	10
7	Traces de θ et Δm^2 pour la résolution par l'algorithme de Metropolis-Hastings.	10
8	Visualisations des paramètres θ et Δm^2 par algorithme Nested Sampling	11

1 Introduction

1.1 Éléments de physique du neutrino

Les neutrinos sont des particules élémentaires du modèle standard de la physique des particules. Ce sont des fermions de spin 1/2 électriquement neutres et disponibles en trois saveurs. Ils ont été postulés en 1930 par Wolfgang Pauli pour expliquer la non-conservation du moment cinétique ainsi que la nature continue du spectre de la désintégration β^- . La première détection expérimentale de neutrino eu lieu en 1956.

Les neutrinos sont présents abondamment dans l'Univers. Une grande partie fut produite dans la période juste après le Big Bang, et ils sont encore produits aujourd'hui par des étoiles, le Soleil par exemple. Ils peuvent également être produits sur Terre par des isotopes radioactifs.

Les deux manières dont les neutrinos peuvent interagir avec la matière sont soit en rebondissant sur une cible, (cédant ainsi de l'énergie et de l'impulsion à la cible), soit en se convertissant en lepton chargé (mais cela requiert de l'énergie, de l'impulsion et un transfert de charge électrique). On peut donc soit détecter la cible, soit le lepton chargé.

Dans le cadre du modèle standard, les neutrinos sont au nombre de 3 : le neutrino électronique ν_e , le neutrino muonique ν_μ et le neutrino tauique ν_τ , chacune étant associée à son lepton respectif. Ces 3 particules possèdent chacune leur antiparticule, à savoir $\bar{\nu}_e$, $\bar{\nu}_\mu$ et $\bar{\nu}_\tau$.

Une propriété propre aux neutrinos est qu'ils ne sont sensibles qu'aux interactions faibles. En 1956, la physicienne Chien-Shiung Wu montre que la conservation de la parité est violée par l'interaction faible, alors qu'elle est conservée par l'interaction forte et l'interaction électromagnétique. Plus particulièrement, l'interaction faible n'influe que sur les particules d'hélicité gauche. La conséquence de cette expérience est donc que les neutrinos du modèle standard n'existent que dans une seule hélicité, à savoir l'hélicité gauche (et par conséquent les antineutrinos sont d'hélicité droite) [1, 2].

Étant très difficilement détectable en raison de leur très faible section efficace, les neutrinos ont d'abord été considérés comme n'ayant pas de masse, car la détermination expérimentale par la désintégration β^- de celle-ci échoua. Aujourd'hui leur masse est considérée comme très faible (au moins 10^6 fois moins lourde qu'un électron, la particule la plus légère du Modèle Standard), mais non nulle.

1.2 Oscillations des neutrinos

La première expérience qui visait à détecter les neutrinos électroniques produits par le Soleil (Expérience Homestack, fin des années 60) ne voyait que 1/3 des quantités théoriques prédites [3]. Dans les années 80 - 90, d'autres expériences similaires firent face au même problème. Ce déficit de neutrinos électronique amena à l'élaboration de la théorie des oscillations de neutrinos, confirmée expérimentalement par les expériences du Sudbury Neutrino Observatory et de Super-Kamiokande [4].

Il existe, pour un neutrino massif, la possibilité que ses états propres de saveurs ν_α ne correspondent pas à ses états propres de masses i . Les neutrinos sont en effet, produits par l'interaction faible dans un état propre de saveur, mais voyagent dans leur état propre de masse. La rotation sous laquelle les neutrinos sont soumis pendant leur trajet peut donc les amener à être détectés dans un autre état propre de saveur. Un neutrino électronique ν_e peut alors, après un temps Δt , être détecté comme étant un neutrino muonique ν_μ ou tauique ν_τ [5]. La possibilité pour un neutrino de changer de saveur est prédit par la mécanique quantique, si et seulement si sa masse n'est rigoureusement pas nulle [6]. On considère alors un "mélange" d'états propres de masse pour chaque neutrino, tel que :

$$|\nu_\alpha(t=0)\rangle = \sum_{i=1,2,3} U_{\alpha,i}^* |\nu_i\rangle \quad (1)$$

où α désigne la saveur du neutrino (e, μ, τ) et U désigne la matrice PMNS. On a donc une correspondance entre l'état propre de saveur et l'état propre de masse par une matrice de mélange [7]. Cette matrice s'écrit :

$$U_{PMNS} \equiv U_{\alpha,i}^* \equiv \begin{pmatrix} U_{e2} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix} \quad (2)$$

Cette matrice peut aussi se mettre sous la forme d'une fonction de 3 angles ($\theta_{12}, \theta_{13}, \theta_{23}$) :

$$U_{PMNS} = \underbrace{\begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{solaire}} \underbrace{\begin{pmatrix} c_{13} & 0 & s_{13}e^{-i\delta} \\ 0 & 0 & 0 \\ -s_{13}e^{-i\delta} & 0 & c_{13} \end{pmatrix}}_{\text{accélérateur + réacteur}} \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix}}_{\text{atmosphérique}} \quad (3)$$

où $c_{12} \equiv \cos \theta_{12}$ et $s_{12} \equiv \sin \theta_{12}$. Le terme *solaire* correspond aux angles θ_{ij} ayant un effet dominant sur les neutrinos produits par le Soleil, le terme *accélérateur + réacteur* ceux ayant un effet dominant sur les neutrinos produits dans les accélérateurs ou des réacteurs, et le terme *atmosphérique* ceux ayant un effet dominant sur les neutrinos émergeant de la collision de particules cosmiques [8].

En plus de la parité, les neutrinos violent la symétrie charge-parité (CP). Cette symétrie est très importante, car elle indique une différence entre particules et antiparticules. L'équation de probabilité d'oscillation (4) comporte une phase δ dont le terme change de signe pour ν ou $\bar{\nu}$.

$$\begin{aligned}
P(\nu_\mu^{(-)} \rightarrow \nu_e^{(-)}) = & s_{23}^2 s_{13}^2 \left(\frac{\Delta_{13}}{A \pm \Delta_{13}} \right)^2 \sin^2 \frac{|A \pm \Delta_{13}| + L}{2} \\
& + c_{13}^2 \sin^2 2\theta_{12} \left(\frac{\Delta_{12}}{2} \right)^2 \sin^2 \frac{AL}{2} \\
& + J \cos \delta \left(\frac{\Delta_{12}}{A} \right) \left(\frac{\Delta_{13}}{|A \pm \Delta_{13}|} \right) \cos \frac{\Delta_{13}L}{2} \sin \frac{AL}{2} \sin \frac{|A \pm \Delta_{12}| + L}{2} \\
& \mp J \sin \delta \left(\frac{\Delta_{12}}{A} \right) \left(\frac{\Delta_{13}}{|A \pm \Delta_{13}|} \right) \sin \frac{\Delta_{13}L}{2} \sin \frac{AL}{2} \sin \frac{|A \pm \Delta_{12}| + L}{2}
\end{aligned} \tag{4}$$

où :

\pm et \mp diffère selon ν ou $\bar{\nu}$

$$\Delta_{ij} = \frac{\Delta m_{ij}^2}{2E} \tag{5}$$

$$A = \sqrt{2} G_F N_e$$

$$J = c_{13}^2 \sin^2 2\theta_{12} \sin^2 2\theta_{13} \sin^2 2\theta_{23}$$

On sait que les angles θ_{12} et θ_{23} sont grands, et que θ_{13} est faible. Une violation CP observable requiert d'avoir les 3 angles de mélanges différents de 0, la détermination de θ_{13} est donc crucial pour mesurer δ_{CP} .

Le terme Δm_{ij}^2 , est très important, car dans le cas de 3 saveurs, la mesure de Δm_{32}^2 permettrait de déterminer laquelle des deux hiérarchies des états propres de masse possibles (normale ou inversée) est la vraie, ce qui aiderait dans la caractérisation des neutrinos.

Dans le cas de 2 saveurs, la probabilité pour un neutrino de passer, par exemple, de ν_e à ν_μ , est donnée par :

$$P_{\nu_e \rightarrow \nu_\mu}(l, E) = \sin^2(\theta) \sin^2\left(\frac{\Delta m_{ij}^2 l}{4E}\right) \tag{6}$$

où l est la distance, E est l'énergie, Δm_{ij}^2 est le terme de différence de masse au carré, θ est l'angle de mélange de la matrice de rotation :

$$\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \tag{7}$$

1.3 Au-delà du Modèle Standard

Les neutrinos ayant une masse, le Modèle Standard doit être amélioré pour la prendre en compte. Deux solutions s'opposent alors : celle de Dirac et celle de Majorana.

Dans la solution de Dirac, les neutrinos sont considérés comme des fermions ordinaires possédant, eux aussi, une hélicité droite. Le neutrino droit serait alors insensible aux interactions fondamentales, sauf gravitationnelle s'il est massif. Ce neutrino est appelé neutrino stérile. Il est donc indétectable, sauf par interaction gravitationnelle ou sauf si les neutrinos dits "actifs" (ν_e, ν_μ, ν_τ) oscillent dans ce neutrino stérile.

La solution de Majorana, quant à elle, montre qu'il est possible de construire un neutrino massif uniquement à partir de particules d'hélicité gauche si et seulement si les neutrinos sont identiques à leur antiparticule. Expérimentalement, cela se traduirait par la possibilité pour certains noyaux de se désintégrer selon la double désintégration β sans émission de neutrino. C'est donc une signature expérimentale qu'il serait possible de détecter, même si ce processus est très rare. On peut alors écrire la matrice d'oscillation globale des neutrinos :

$$U_{\nu,osc} = \underbrace{\begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{solaire}} \underbrace{\begin{pmatrix} c_{13} & 0 & s_{13}e^{-i\delta} \\ 0 & 0 & 0 \\ -s_{13}e^{-i\delta} & 0 & c_{13} \end{pmatrix}}_{\text{accélérateur + réacteur}} \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix}}_{\text{atmosphérique}} \cdot P \tag{8}$$

Si les neutrinos sont des particules de Dirac, alors :

$$P = \mathbb{1}_3 \tag{9}$$

Dans le cas où les neutrinos sont des particules de Majorana, alors :

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{i\alpha_{21}/2} & 0 \\ 0 & 0 & e^{i\alpha_{31}/2} \end{pmatrix} \quad (10)$$

1.4 DUNE

DUNE (Deep Underground Neutrino Experiment) est une expérience située aux USA qui se focalise sur la détection du changement de saveur par oscillations des neutrinos et des antineutrinos [9]. Le faisceau de neutrinos muoniques ν_μ crée au Fermilab traversera 1300 km en 4 ms afin d'être détecté par 4 détecteurs au Sanford Underground Research Facility et d'observer la nouvelle saveur des neutrinos. Ces détecteurs sont enfouis sous terre afin d'éviter les perturbations dues aux muons et aux neutrinos qui proviennent de l'atmosphère et des rayons cosmiques.

Les détecteurs sont remplis d'argon liquide à 88 K. De ce fait, quand un neutrino rentre dans un détecteur, il peut interagir avec un noyau d'argon pour créer son lepton associé et des protons / neutrons. Ces particules chargées iront ioniser l'argon, créant des électrons récoltés par trois couches de circuits de cuivre pour reconstruire la trace des parties issues de l'interaction grâce à l'application d'un champ électrique. Connaissant la trajectoire et le dépôt en énergie des électrons de dérive, on peut alors remonter à la saveur du neutrino incident, mais aussi à son impulsion et son énergie, et ainsi avoir des informations sur son oscillation au cours du trajet.

DUNE fournira une mesure expérimentale de la violation de la symétrie CP, un ingrédient des scénarios de disparition de l'antimatière. La détermination des paramètres d'oscillation permettrait également de connaître les états propres initiaux de masse des neutrinos, ainsi que leur véritable hiérarchie. L'expérience DUNE est donc une expérience cruciale pour le développement de la physique au-delà du Modèle Standard de la physique des particules.

Parmi les autres objectifs de DUNE, on retrouve la détection d'explosion de supernovæ, ce qui permettrait de mieux comprendre la formation d'étoiles à neutrons ou possiblement de trou noir. DUNE pourrait également observer la désintégration du proton, car même si son temps de vie est très long, sa désintégration n'est pas impossible dans certaines modèles de nouvelle physique, et pourrait arriver à tout moment.

2 Méthodes statistiques d'échantillonnage de distribution

2.1 Aperçu de statistiques Bayésiennes

Pour chercher à estimer les paramètres de la matrice PMNS, on doit utiliser une approche statistique sur les données du détecteur. La collaboration DUNE utilise des statistiques Bayésiennes, qui reposent sur le théorème de Bayes :

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)} \quad (11)$$

où :

- $P(H|D)$ est la probabilité de l'hypothèse H connaissant les données D, c'est la probabilité *postérieure* sur H
- $P(D|H)$ est la probabilité d'observer les données D selon l'hypothèse H, appelée *vraisemblance* \mathcal{L}
- $P(H)$ est la probabilité de l'hypothèse H, c'est la probabilité dite *a priori*
- $P(D)$ est la probabilité des données D indépendamment de l'hypothèse H, aussi appelée *l'évidence*

La probabilité postérieure est donc la probabilité à priori pondérée par la vraisemblance de l'observation des données, normalisée par la probabilité d'observer les données. Le théorème de Bayes permet donc de déterminer la probabilité qu'un événement arrive en se basant sur l'événement qui s'est réalisé précédemment, en particulier quand ces deux événements sont interdépendants. Il est donc possible à partir de ce théorème de calculer la probabilité d'un événement en se basant sur les informations connues et sur les données provenant de nouvelles observations.

Ce théorème s'applique très bien à la physique du neutrino [10]. Si, par exemple, on cherche à déterminer δ_{CP} , qu'on ne connaît pas. Notre probabilité *a priori* du paramètre est donc plate. En calculant la vraisemblance de l'observation des neutrinos ν_e et $\bar{\nu}_e$ selon certaines valeurs hypothétiques de δ_{CP} , on peut alors mesurer la probabilité postérieure pour chaque hypothèse et ainsi estimer la valeur de δ_{CP} correspondant à la probabilité postérieure la plus élevée.

Ce théorème actualise donc la probabilité postérieure en variant les paramètres du système à chaque itération du processus. Cependant, en physique du neutrino, les paramètres à varier sont d'un nombre important. Faire varier ces paramètres devient alors vite coûteux en temps de compilation au vu du nombre de points à évaluer. Un processus alternatif peut alors être utilisé pour parcourir l'espace des paramètres.

2.2 Chaîne de Markov et méthode de Monte-Carlo

Une chaîne de Markov est un processus statistique qui possède la propriété suivante : l'information utile pour la prédiction d'un événement futur est entièrement contenue dans l'événement précédent, et ne dépend pas des événements antérieurs. Elle représente un outil statistique puissant permettant de prévoir les changements futurs à partir de l'observation du présent.

Une méthode de Monte-Carlo est un algorithme ayant pour but de calculer une valeur numérique par le biais de procédés aléatoires [11]. Cette méthode permet d'échantillonner l'espace des paramètres d'oscillations en utilisant un processus de marche semi-aléatoire.

En combinant ces deux méthodes, on peut alors créer des algorithmes de Monte-Carlo où l'échantillonnage des paramètres se base sur le parcours de chaînes de Markov. Ces méthodes sont alors appelées Markov Chain Monte-Carlo (MCMC).

Dans le cadre des statistiques Bayésiennes, l'algorithme de Metropolis-Hastings est une implémentation générique d'une méthode MCMC. Ici, le tirage aléatoire est proportionnel à la distribution recherchée, et les pas sont tirés de la distribution postérieure. Cette propriété est assurée par l'équation d'équilibre détaillée :

$$G(x_i)T(x_{i+1}|x_i) = G(x_{i+1})T(x_i|x_{i+1}) \quad (12)$$

où $G(x_i)$ est la distribution cible et $T(x_{i+1}|x_i) = J(x_{i+1}|x_i)A(x_{i+1},x_i)$ est la probabilité de transition, avec $J(x_{i+1}|x_i)$ la fonction de saut et $A(x_{i+1},x_i) = \min(1, r)$ la fonction d'acceptance.

Ceci permet de garantir deux propriétés importantes : l'apériodicité et l'ergodicité. Ainsi, l'algorithme n'oscille pas entre les mêmes valeurs et converge vers la distribution stationnaire, qui s'avère être la distribution à échantillonner [10, 12].

On va se placer dans le cas simple où l'on cherche à échantillonner une distribution Gaussienne $G(x)$. L'algorithme est le suivant [10] :

- Première itération : $i = 1$. On commence avec une valeur aléatoire de x_i . Cette valeur est forcément acceptée.
- On propose une nouvelle itération $i + 1$ en utilisant la fonction $J(x_i + 1|x_i)$.
- On calcule le rapport de Metropolis Hastings, définit ainsi :

$$r = \frac{G(x_{i+1})J(x_i|x_{i+1})}{G(x_i)J(x_{i+1}|x_i)} \quad (13)$$

On applique ensuite la fonction d'acceptance $A(x_{i+1},x_i) = \min(1, r)$:

- Si $r < 1$: On tire aléatoire un nombre $u \in U(0, 1)$
 - * Si $r \geq u$: l'itération $i + 1$ est acceptée
 - * Si $r < u$: l'itération est refusée, on compte alors une seconde fois l'itération i

–

- On répète ce processus un nombre suffisant de fois pour que l'échantillonnage soit suffisant.

2.3 Algorithme Nested Sampling

L'algorithme Nested Sampling est une autre approche d'implémentation de statistiques Bayésiennes [13]. Il permet, tout comme l'algorithme de Metropolis-Hastings, d'échantillonner une distribution postérieure, mais contrairement à celui-ci, il ne repose pas sur une chaîne de Markov. Il approxime la distribution postérieure par une somme de rectangle, et l'intégration de ces rectangles donnent la probabilité postérieure.

Cet algorithme est aussi utilisé pour calculer le terme $P(D)$ dans l'équation (11), terme très difficile à calculer sans passer par une méthode d'échantillonnage. Ce terme est négligeable dans l'algorithme de Metropolis-Hastings, mais il devient essentiel lors de la comparaison de différents modèles physiques. Cet algorithme est uniquement basé sur la fonction de vraisemblance, et s'exécute ainsi :

- On jette un nombre de points dans notre espace a priori.
- Le point ayant la vraisemblance la plus faible est "tué" et on rejette un nouveau point dans l'espace des paramètres. Ce point doit posséder une vraisemblance plus haute que celle du point "tué".
- À chaque itération, on calcule le volume de l'espace des paramètres grâce à la fonction bêta :

$$B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} dt \quad (14)$$

- Le processus s'arrête une fois que l'espace des paramètres atteint un volume très proche de 0.

On peut alors calculer l'évidence Z en utilisant la formule :

$$Z \simeq \sum_i \Delta V_i \times \mathcal{L}_i \quad (15)$$

où ΔV_i est le volume du point "tué" i . En calculant le poids de chaque point "tué" selon l'équation :

$$w_i = \Delta V_i \times \mathcal{L}_i \quad (16)$$

On peut alors remonter à la probabilité postérieure :

$$P(H|D) \sim \frac{\Delta V_i \times \mathcal{L}_i}{Z} \quad (17)$$

Le Nested Sampling possède plusieurs avantages. Premièrement, il peut traiter des problèmes avec des fonctions de vraisemblances très irrégulières et non continues. Il est par exemple possible d'échantillonner une probabilité postérieure en "boîte d'œuf" [14]. De plus, l'algorithme commence par échantillonner vaguement dans les régions de basse vraisemblance, et échantillonne densément dans les régions de hautes vraisemblances, ce qui se traduit par une bonne convergence et bonne estimation des paramètres recherchés, ce qui le rend plus efficaces que la plupart des méthodes MCMC [15].

Il est cependant sujet à plusieurs désavantages, par exemple la possibilité de ne pas échantillonner une très petite région X de haute vraisemblance, ce qui biaiserait le calcul de Z . De même, l'optimisation de la méthode de détermination du prochain point peut devenir problématique pour de grandes dimensions, et ainsi diminuer l'efficacité de l'algorithme.

3 Exemples d'applications de l'algorithme de Metropolis-Hastings

3.1 Échantillonnage d'une gaussienne

Pour se familiariser avec l'algorithme de Metropolis-Hastings et en comprendre son fonctionnement, l'échantillonnage d'une distribution gaussienne de paramètres $\mu = 0$ et $\sigma = 1$ fut pris comme premier exemple. Cet exercice permet également de montrer que les itérations de la chaîne de Markov vont suivre la distribution cible. On cherche alors à coder cet algorithme en Python sous la forme d'un Jupyter notebook. On dispose d'une distribution cible (Figure 1).a que l'on souhaite échantillonner. Il est alors naturel de définir la fonction de saut comme une gaussienne également. C'est cette fonction qui va choisir la prochaine itération. Dans cet exemple, on fixe la valeur de l'écart-type dans la fonction de saut. Après 100000 itérations, on obtient l'échantillonnage figurant en Figure 1.b au bout d'une dizaine de secondes d'exécution.

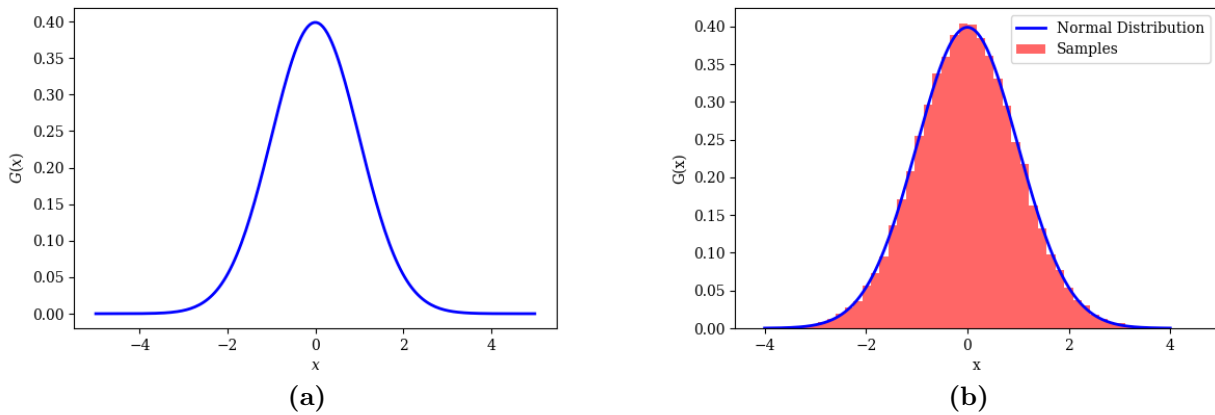


Figure 1: (a) Distribution cible ayant pour paramètres $\mu = 0$ et $\sigma = 1$. (b) Échantillonnage par algorithme de Metropolis-Hastings en comparaison avec la distribution cible

La moyenne et l'écart type de la distribution cible peuvent ainsi facilement être trouvés à partir des données de l'échantillonnage. On trouve alors $\mu = 0,015$ et $\sigma = 1,005$. Ce simple exemple rend compte de la précision de l'algorithme de Metropolis-Hastings, ainsi que de sa rapidité d'exécution de quelques secondes.

3.2 Estimation des paramètres de plusieurs gaussiennes à partir de données

On s'intéresse maintenant à un cas plus concret où l'on dispose uniquement de données et d'indications du constructeur. Dans cet exemple, on cherche à déterminer les écarts-types et moyennes de différentes gaussiennes à partir de données d'un spectromètre pour retrouver les différentes sources composants ce spectromètre [16, 17].

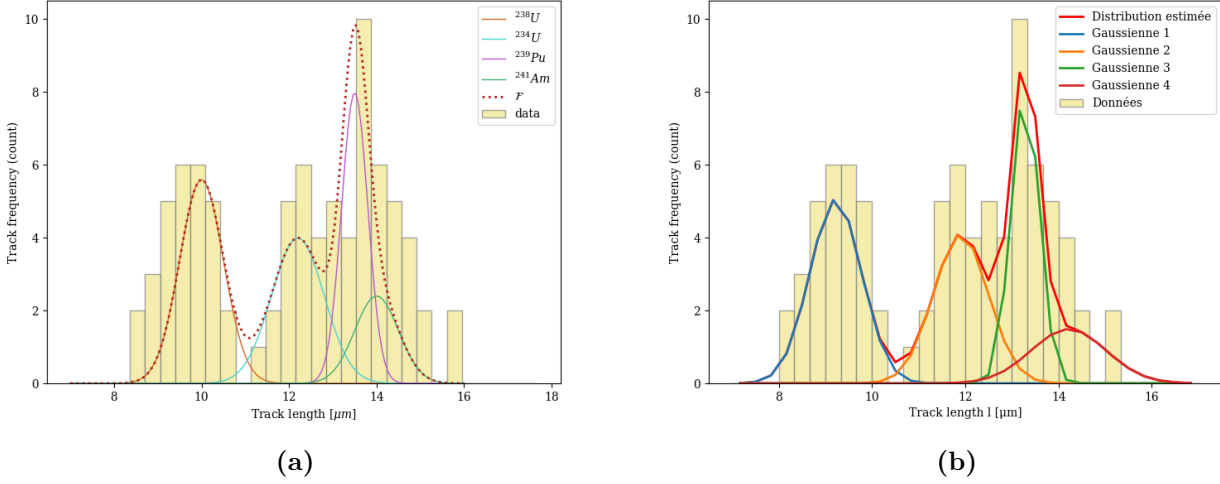


Figure 2: (a) Visualisation des quatre gaussiennes correspondant aux quatre sources, ainsi que la distribution totale cible [16, 17]. (b) Distribution finale obtenue par l'algorithme de Metropolis-Hastings pour 100000 itérations

Pour procéder à la détermination des paramètres souhaitée, il est tout d'abord nécessaire de définir la fonction de vraisemblance (dans le cas d'un histogramme de données) [18] :

$$-\ln(\mathcal{L}_i) = \sum_{j=1}^{N \text{ bins}} N_{i,j}^H(\mu_i, \sigma_i) - N_{i,j}^D + N_{i,j}^D \ln \left(\frac{N_{i,j}^D}{N_{i,j}^H(\mu_i, \sigma_i)} \right) \quad (18)$$

avec $N_{i,j}^H$ le nombre d'événements selon la densité de probabilité totale supposée, et $N_{i,j}^D$ le nombre d'événements selon les données de l'histogramme.

Le maximum de vraisemblance nous va nous indiquer quelles seront les valeurs les plus probables pour l'estimation des paramètres. En calculant la vraisemblance à chaque itération, on peut alors réécrire le rapport de Metropolis-Hastings en fonction de la vraisemblance :

$$r = \frac{\mathcal{L}_{i+1}}{\mathcal{L}_i} * \frac{J(x_i|x_{i+1})}{J(x_{i+1}|x_i)} \quad (19)$$

Il nous suffit donc de connaître, à chaque itération, la valeur de la vraisemblance ainsi que la fonction de saut pour en déduire la distribution cible grâce à l'algorithme de Metropolis-Hastings. Précédemment, la fonction de saut était une gaussienne avec la taille de pas fixée. Ici, il nous faut utiliser une loi normale multidimensionnelle afin de prendre en compte les huit paramètres du problème. Pour ceci, il nous faut définir la matrice de covariance du problème. C'est cette matrice qui va déterminer les sauts effectués par la fonction de saut. Cette fonction s'écrit donc ainsi :

$$J(x_{i+1}|x_i) = \{ \vec{\mu}; \varsigma(\vec{\mu}) \} \quad (20)$$

où $\vec{\mu}$ est notre espace des paramètres, et $\varsigma(\vec{\mu})$ est la matrice de covariance associée à nos paramètres. Cette matrice a dû être réglée manuellement pour chaque paramètre afin d'identifier la valeur de pas la plus optimale à la convergence de la chaîne de Markov.

Concernant la convergence, l'algorithme de Metropolis-Hastings n'est pas immédiat, il prend plusieurs itérations avant d'identifier la bonne gamme de valeur dans laquelle échantillonner. Pour ne pas fausser les résultats finaux, on doit donc implémenter une fonction burn-in qui va supprimer les paramètres échantillonnés biaisant les estimations. Ce burn-in est déterminé en regardant la trace de la chaîne de Markov.

En partant de paramètres initiaux définis par l'utilisateur, et pour un nombre d'itérations de 100000, on obtient la distribution cible finale (Figure 2b). L'algorithme va mettre à jour à chaque itération l'espace de paramètres, si bien qu'à la fin, les paramètres de chaque gaussienne peuvent être estimés en regardant le maximum de vraisemblance.

Remarquons que la précision du paramètre varie selon la gaussienne échantillonnée. Le peu de donnée fournie ne permet pas d'échantillonner très précisément les gaussiennes, l'algorithme va alors converger vers une plage de donnée, puis osciller dans celle-ci jusqu'à la fin des itérations. En regardant sur la Figure 2, on aperçoit que trois des quatre gaussiennes se chevauchent, ce qui rend difficile le processus d'échantillonnage dans cette région. Cette corrélation se visualise très bien dans la Figure 3 où l'on observe la corrélation entre μ_2/μ_3 et μ_3/μ_4 représenté par la diagonale prononcée dans leur histogramme 2D respectif. Au contraire, μ_1 n'est corrélée avec aucune autre moyenne, et ses histogrammes 2D ne présentent pas de diagonale, c'est également le cas pour μ_2/μ_4 .

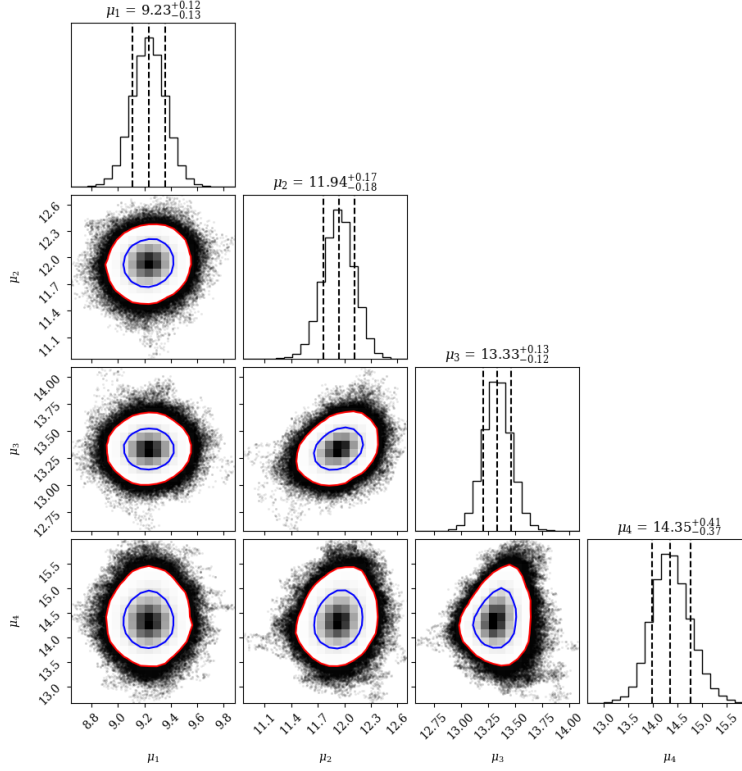


Figure 3: Graphique de corrélation des moyennes des quatre gaussiennes présentes dans le spectre de données, avec la valeur estimée du paramètre par l’algorithme de Metropolis-Hastings dans un intervalle de confiance de 68%. En orange est tracé le countour à 68% et en bleu celui à 95%. Les zones les plus foncées représentent le maximum de vraisemblance et donc la probabilité d’estimation la plus grande

4 Estimation des paramètres θ et Δm^2 à partir de données simulées de DUNE

4.1 Contexte

On cherche désormais à déterminer les paramètres d’oscillations θ et Δm^2 . On souhaite estimer ces paramètres à partir d’histogrammes de données simulées de l’expérience DUNE (Figure 4) [19]. Pour cela, on se place dans un modèle simplifié avec uniquement deux saveurs de neutrinos.

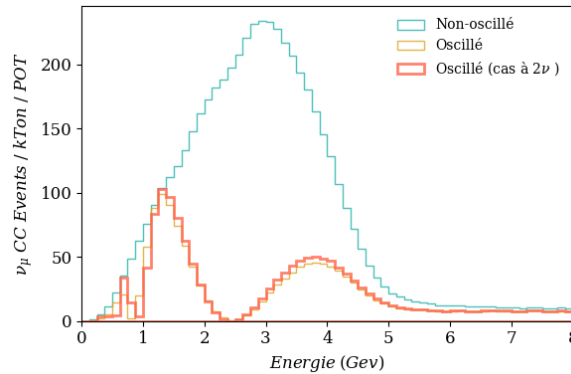


Figure 4: Comparaison des taux d’interactions à courant chargé ν_μ dans le détecteur lointain de DUNE, pour 1 an d’exploitation

La probabilité pour un neutrino d’osciller est alors :

$$P_{\nu_\mu \rightarrow \nu_e} = \sin^2(2\theta) \sin^2\left(\frac{\Delta m^2 L}{4E}\right) \quad (21)$$

avec :

$$\sin^2\left(\frac{\Delta m^2 L}{4E}\right) \simeq \sin^2\left(1.27 \frac{\Delta m^2 [eV^2] L[km]}{E[GeV]}\right) \quad (22)$$

L'équation sur la distribution cible s'écrit alors :

$$P(\theta, \Delta m^2 | \vec{D}) = P(\vec{D} | \theta, \Delta m^2) P(\theta, \Delta m^2) \quad (23)$$

où :

- $P(\theta, \Delta m^2 | \vec{D})$ est la probabilité postérieure
- $P(\vec{D} | \theta, \Delta m^2)$ est la vraisemblance
- $P(\theta, \Delta m^2)$ est la probabilité à priori sur les paramètres θ et Δm^2 choisie respectivement comme $P(\theta) = \Pi(-\pi/2, \pi/2)$ et $P(\Delta m^2) = \Pi(0, 0.01)$

Le processus d'estimation des paramètres est alors le suivant :

- On propose une valeur aléatoire contenue dans la probabilité à priori : $\{\theta_i, \Delta m_i^2\}$
- On calcule la probabilité d'oscillation pour ces paramètres $\{\theta_i, \Delta m_i^2\}$ à l'énergie E_j : $P_{\nu_\mu \rightarrow \nu_e}(\theta_i, \Delta m_i^2, E_j)$
- On applique la probabilité d'oscillation à l'histogramme non-oscillé (coube bleue sur la Figure 4) : $N_{osc(i)}(E_j) = N_{unosc}(E_j) \times P_{\nu_\mu \rightarrow \nu_e}(\theta_i, \Delta m_i^2, E_j)$
- On calcule la vraisemblance à chaque itération entre l'histogramme généré et celui extrait des données.

Le reste de l'estimation dépend de l'algorithme utilisé.

4.2 Résolution par algorithme de Metropolis-Hastings

Une première façon de résoudre ce problème consiste à utiliser une nouvelle fois l'algorithme de Metropolis-Hastings. On utilise encore une fois la méthode de la vraisemblance en calculant le ratio de Metropolis-Hastings (19), où la fonction de saut choisie est également une loi normale multidimensionnelle définie comme :

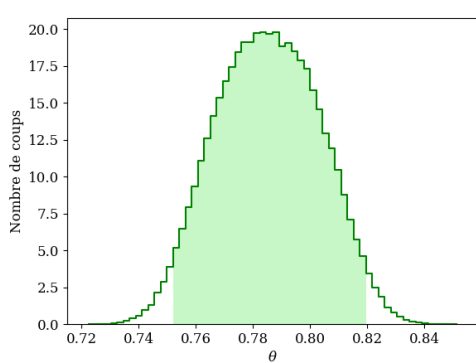
$$J(\theta_i, \Delta m_i^2 | \theta_{i-1}, \Delta m_{i-1}^2) = \mathcal{N}(\{\theta_{i-1}, \Delta m_{i-1}^2\}, \Sigma_{\theta, \Delta m^2}) \quad (24)$$

où $\Sigma = \text{diag}(\sigma_\theta, \sigma_{\Delta m^2})$ représente les valeurs de saut de chaque paramètre. Ces valeurs influencent grandement sur la convergence de la chaîne de Markov, et doivent donc être choisies heuristiquement. Le procédé de l'algorithme est le même que celui de la section 3.2. Les valeurs théoriques de θ et Δm^2 sont $\theta = \pi/4 \simeq 0.7853$ et $\Delta m^2 = 0.022$. Pour un million d'itérations, et après vingt minutes d'exécution, les résultats obtenus par l'algorithme de Metropolis-Hastings sont $\theta = (0.788 \pm 0.018)$ (soit $\sin^2(\theta) = 0.499 \pm 0.018$) et $\Delta m^2 = (2,199 \pm 0.009) \times 10^{-3} \text{ eV}^2$. En comparaison avec les valeurs fournies par le Particle Data Group [20] (à un σ), qui sont de $\sin^2(\theta) = (0.558^{+0.015}_{-0.021})$ et $\Delta m^2 = (2.455 \pm 0.028) \times 10^{-3} \text{ eV}^2$, les valeurs trouvées sont similaires, et donc DUNE pourra, en 1 an d'exploitation, vite trouver cette précision et ensuite l'améliorer. Cependant, aucune erreur systématique n'est prise en compte dans l'algorithme, telle que les incertitudes dues au flux, au modèle d'interaction ou au détecteur, ce qui donne des valeurs artificiellement précises pour nos estimations.

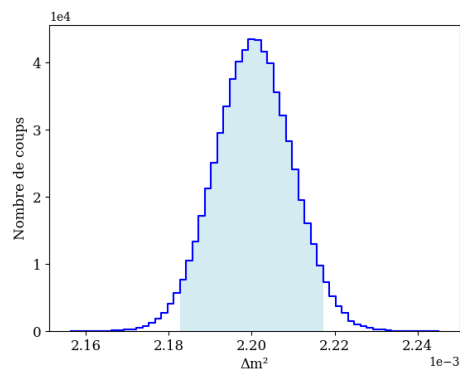
Ces valeurs sont obtenues en prenant le mode de l'histogramme représentant la probabilité postérieure, comme montré sur la Figure 5. La Figure 6 montre la visualisation de ces 2 paramètres par un histogramme en deux dimensions en fonction de la vraisemblance. Pour obtenir ces paramètres, la matrice Σ des sauts a été fixée ainsi :

$$\Sigma = \begin{pmatrix} 10^{-3} & 0 \\ 0 & 10^{-10} \end{pmatrix} \quad (25)$$

Ces paramètres peuvent tout à fait être ajustés plus précisément pour optimiser sur la convergence de la chaîne de Markov, leurs déterminations ici n'ayant été faites qu'heuristiquement.



(a) θ , avec un intervalle crédible de $[0.750, 0.820]$



(b) Δm^2 , avec un intervalle crédible de $[2.18, 2.21] 10^{-3} eV^2$

Figure 5: Visualisation en histogramme des estimations des valeurs des paramètres θ et Δm^2 . Les aires des histogrammes sont normalisées à 1. L'aire coloriée sur chacune des figures correspond à un intervalle crédible de 95%, indiqué dans la légende de chaque sous-plot.

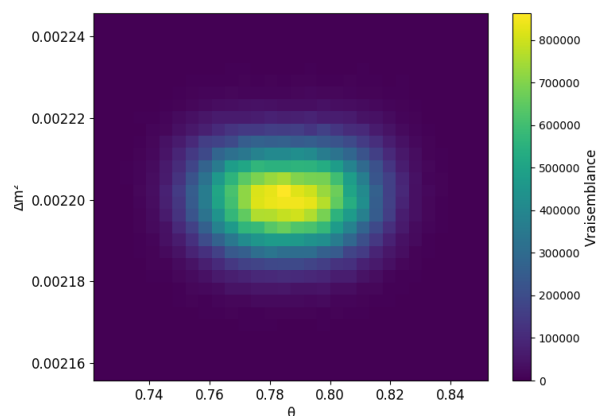
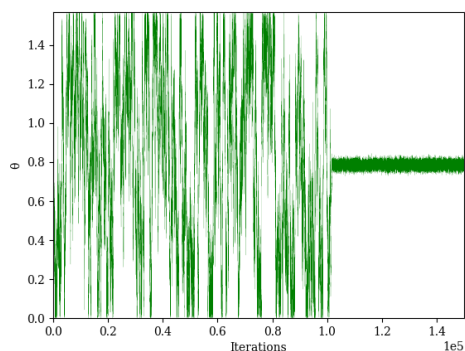
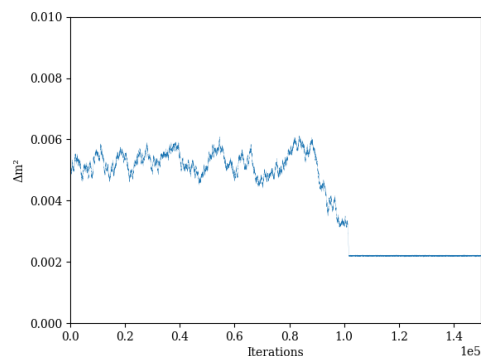


Figure 6: Histogramme en deux dimensions des estimations des paramètres θ et Δm^2 . Les points les plus jaunes correspondent aux maximums de vraisemblance, et donc aux estimations les plus probables des paramètres.

Une façon d'observer la convergence de la chaîne de Markov est de tracer graphiquement les valeurs de θ et Δm^2 pour chaque itération. Ces traces permettent de plus de rendre compte de l'efficacité de l'algorithme ainsi que de la nécessité d'instaurer un burn-in, comme montré en Figure 7.



(a) θ



(b) Δm^2

Figure 7: Traces de θ et Δm^2 pour la résolution par l'algorithme de Metropolis-Hastings. Par souci de visualisation, seules les 150000 premières itérations sont représentées afin de montrer la convergence de la chaîne de Markov.

4.3 Résolution par algorithme Nested Sampling

L'algorithme Nested Sampling constitue une alternative pour estimer les paramètres θ et Δm^2 qui est n'a jusqu'alors jamais été expérimentée. L'implémentation de cet algorithme à ce problème permet de comparer les deux méthodes de résolution, également sur leur temps de compilation et sur la distribution des points échantillonnés. Cependant, à la différence de l'algorithme de Metropolis-Hastings, le nombre d'itérations est une variable moins importante. En effet, le Nested Sampling repose avant tout sur le nombre de points jetés dans l'espace des paramètres, et sur la méthode de sélection du prochain point, que sur le nombre d'itérations global. C'est en particulier cette dernière qui, une fois optimisée, permet de rendre compte de la puissance de l'algorithme. De plus, l'algorithme possède une limite liée au volume restant, contrairement à l'algorithme de Metropolis-Hastings. L'implémentation de ce code a pu être réalisé grâce à l'exemple commenté par Johannes Buchner [21].

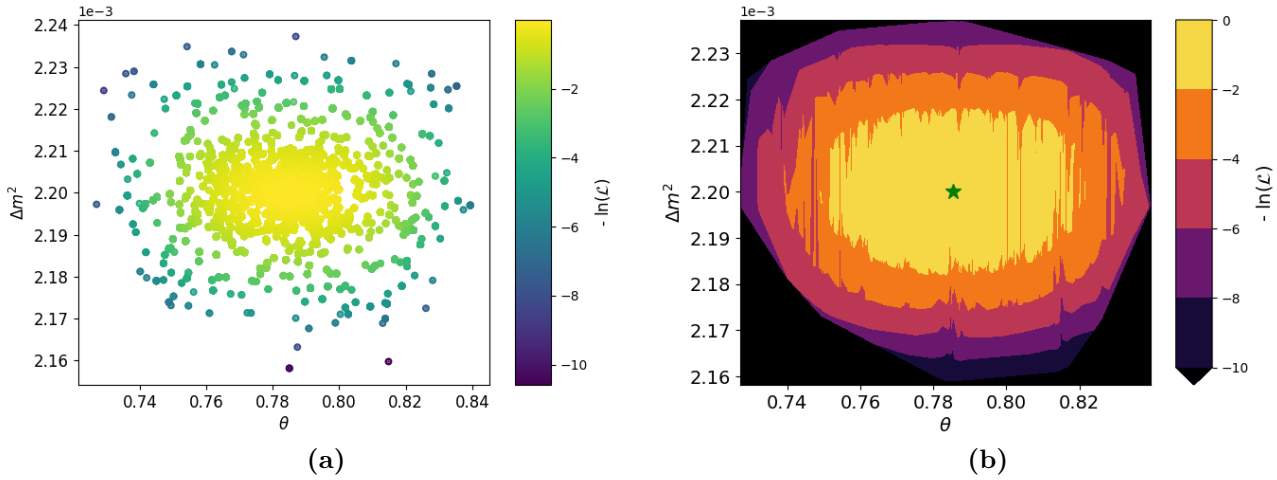


Figure 8: Visualisation des estimations des paramètres θ et Δm^2 (a). Les points les plus jaunes correspondent au minimum de la log-vraisemblance négative et donc aux valeurs d'estimations les plus probables. (b) montre l'interpolation de la fonction de log-vraisemblance négative ($-\ln \mathcal{L}$). Les régions les plus jaunes correspondent à un minimum de vraisemblance, donc aux estimations les plus probables. La valeur théorique des paramètres d'oscillation θ et Δm^2 est représentée par une étoile verte

Les résultats ont été obtenus avec un nombre de points jetés de 200, et un nombre d'itérations de 20000. Le temps d'exécution est de 30 minutes. Ce temps est plus long que l'algorithme de Metropolis-Hastings et est dû au nombre important de points jetés à chaque itération. On détermine la valeur la plus probable en prenant la valeur avec la plus petite vraisemblance. Il n'est pas rigoureux de représenter les estimations de θ et Δm^2 par un histogramme en deux dimensions comme fait précédemment en Figure 6, car avec le Nested Sampling, on ne dispose pas de bins pour tracer des histogrammes. On représente donc en Figure 8.a l'estimation des paramètres par une distribution de points dans l'espace des paramètres dont le volume a été réduit par l'algorithme. Connaissant cette distribution, on peut alors interpoler la vraisemblance dans l'espace des paramètres, comme représenté en Figure 8.b.

5 Conclusion et Perspectives

Les estimations effectuées durant ce stage se sont révélées être en accord avec les données théoriques fournies par le Particle Data Group [18]. Nos estimations se sont retrouvées être artificiellement similaires. En effet, les données utilisées ne sont soumises à aucune erreur systématique : le flux est supposé être parfaitement connu, l'efficacité de détection est aussi supposée parfaite. L'expérience DUNE prendra en compte beaucoup plus de paramètres liés au flux, au modèle d'interaction ou au détecteur en lui-même. Tout cela conduit à un grand nombre de paramètres pour l'estimation. Cela requiert l'utilisation de processus statistiques efficaces à large dimensions, comme les chaînes de Markov. Cependant, d'autres types d'algorithme, comme le Nested Sampling, permettent également d'estimer précisément ces paramètres, et pourrait fournir une alternative qui n'a encore jamais été utilisée dans le cadre de l'expérience DUNE.

Appendice

Tous les codes créés dans le cadre de ce stage peuvent être retrouvés sur la page Github référencée [22].

Références utilisées

- [1] Introduction to neutrinos. <https://shorturl.at/eHqTf>.
- [2] Le neutrino, du rifi dans les symétries - nathalie besson - youtube. <https://t.ly/HQebb>, 2024.
- [3] Expérience Homestake. <https://shorturl.at/Uwd4a>, 2024.
- [4] Problème des neutrinos solaires. <https://shorturl.at/5uryY>, 2024.
- [5] C. Giganti, S. Lavignac, and M. Zito. Neutrino oscillations: The rise of the pmns paradigm. *Progress in Particle and Nuclear Physics*, 98:1–54, January 2018.
- [6] Olivia Meredith Bitter. An introduction to neutrino oscillation: A quantum mechanical perspective. 2021.
- [7] Pontecorvo–Maki–Nakagawa–Sakata matrix - wikipédia. <https://shorturl.at/R8jdy>, 2024.
- [8] Haegel, Leïla. *Measurement of neutrino oscillation parameters using neutrino and antineutrino data of the T2K experiment*. PhD thesis, 2017.
- [9] The science of the Deep Underground Neutrino Experiment (dune), howpublished = <https://t.ly/2klzq> .
- [10] Leïla Haegel. Bayesian inference for neutrino physics. <https://t.ly/I-2CZ>, 2023.
- [11] Florian Kuzler. Programming in c++ under Linux.
- [12] Gersende Fort. Méthodes de Monte Carlo par chaînes de Markov. <https://t.ly/M2GwQ>.
- [13] John Skilling. Nested sampling for general bayesian computation. 2006.
- [14] Mcmc interactive gallery. <https://chi-feng.github.io/mcmc-demo/app.html>.
- [15] Lecture 13: Nested sampling for bayesian inference. <https://rb.gy/cxb43z>, 2018.
- [16] J.A. Bartz, C.J. Zeissler, V.V. Fomenko, and M.S. Akselrod. An imaging spectrometer based on high resolution microscopy of fluorescent aluminum oxide crystal detectors. *Radiation Measurements*, 56:273–276, 2013. Proceedings of the 8th International Conference on Luminescent Detectors and Transformers of Ionizing Radiation (LUMDETR 2012).
- [17] <https://t.ly/68erk>.
- [18] P. A. Zyla et al. Review of Particle Physics. *PTEP*, 2020(8):662, 2020.
- [19] DUNE Collaboration et al. Experiment simulation configurations approximating DUNE tdr, 2021.
- [20] lxxx005-sumtab.dvi. <https://rb.gy/e9o1zi>, 2024.
- [21] Practical inference for researchers in the physical sciences course. <https://rb.gy/121mdp>.
- [22] Codes créés pendant le stage. <https://github.com/RomaiinF/Stage-IP2I-DUNE>, 2024.