Non-reversible and generative sampling algorithms. Application to the inference of cosmological parameters

Vincent Souveton



Thèse pour obtenir le grade de
Docteur en Mathématiques appliquées
de l'Université Clermont Auvergne.

Non-reversible and generative sampling algorithms. Application to the inference of cosmological parameters.

Préparée au sein du Laboratoire de Mathématiques Blaise Pascal (UMR 6620) et de l'École Doctorale des Sciences Fondamentales de l'Université Clermont Auvergne.

Présentée et soutenue publiquement par
Vincent SOUVETON

le mercredi 25 septembre 2024 devant un jury composé de

M. Emmanuel Gangler	LPC, CNRS, Université Clermont Auvergne	Président du jury
M. Karim Benabed	IAP, CNAP, Sorbonne Université	Rapporteur
M. Christian Robert	CEREMADE, Université Paris-Dauphine	Rapporteur
Mme Marylou Gabrié	CMAP, École Polytechnique	Examinatrice
M. Arnaud Guillin	LMBP, Université Clermont Auvergne	Directeur de thèse
M. Guilhem Lavaux	IAP, CNRS, Sorbonne Université	Directeur de thèse
Mme Manon Michel	LMBP, CNRS, Université Clermont Auvergne	Directrice de thèse

À Elsa,
pour tout.

Remerciements

Tu devrais mettre des citations pas très intéressantes.

Julian Le Clainche

Si ta thèse est très bien écrite, tu peux viser un prix Nobel de Littérature.

Émilien Zabeth

Elle est virevoltante, cette route.

Elsa Boissy

Ce manuscrit ne raconte au final qu'un bout presque insignifiant de ces années de thèse et du chemin parcouru pour en arriver là. Si tout ne fut pas toujours facile, je mesure l'immense chance que j'ai d'avoir pu accomplir ce travail en étant rémunéré pour chercher des réponses à de si jolies questions. Tout cela n'aurait évidemment pas été possible si l'époque, le lieu de ma naissance, ma condition sociale et, plus généralement, les aléas de la vie, m'avaient été moins favorables. Il faut sans cesse le rappeler : la méritocratie n'existe pas. Je suis néanmoins fier d'apporter une minuscule pierre à l'édifice gigantesque de la connaissance humaine et espère pouvoir suivre cette route encore longtemps. Si vous lisez ces lignes, alors sachez qu'elles sont en grande partie le fruit de rencontres et de discussions avec de nombreuses personnes que je tiens ici à remercier.

Merci à Arnaud Guillin, Guilhem Lavaux et Manon Michel qui ont encadré ce projet et m'ont accordé leur confiance dès mon stage de recherche en Master. Le sujet élégant et ambitieux qu'elle et ils ont proposé m'a permis de trouver ma voie en tant que scientifique. Arnaud, Manon, merci de m'avoir donné l'opportunité de faire autant de voyages et de rencontres, merci aussi de m'avoir initié aux méthodes d'échantillonnage statistique avec rigueur. Guilhem, merci pour ton expertise en code, ta patience et ton incroyable disponibilité malgré les quelques 400 kilomètres qui séparent nos labos. Ta gentillesse, ton rire communicatif et l'étendue de tes connaissances scientifiques sont un modèle pour moi. Merci également à Karim Benabed et Christian Robert pour avoir accepté de rapporter cette thèse et pour leurs retours constructifs. Enfin, merci à Marylou Gabrié et Emmanuel Gangler, dont j'ai eu la chance d'assister à de jolis exposés, d'avoir accepté de faire partie du jury.

During this journey, I had the opportunity to meet and work with remarkable people from the Aquila consortium. Thank you so much Jens Jasche for welcoming me in Stockholm. I enjoyed discussing science and non-science topics with you. It was an honor that you shared some of your brilliant and original thoughts with me during *fika* or in front of a white board and I hope one day I will be as wonderful as you in the role of a supervisor. I appreciated collaborating with Ewoud Wempe, a genius

coder and a very kind person. Thank you Adam Andrews and Eleni Tsaprazi for your kindness and advice.

À Clermont-Ferrand, j'ai eu la chance d'être très bien accueilli depuis mon arrivée en Master. Je pense en particulier à tous les enseignant·e·s que j'ai croisé·e·s et qui m'ont accompagné dans des années Covid pas évidentes. Je remercie notamment Hacène Djellout qui m'a toujours soutenu, depuis mes choix d'orientation en Master jusqu'à mes récentes recherches de post-doc, pour sa bienveillance et sa bonne humeur. Richard Griffon a été un excellent parrain au cours de ces trois années ; je le remercie pour son écoute et son aide dans les moments délicats. Je remercie beaucoup Cédric Barrel et Damien Ferney, qui m'ont impressionné par leur expertise technique. Cédric, merci d'avoir sauvé mon ordinateur à de (trop) nombreuses reprises. Damien, merci pour tes explications nombreuses au cours des trois dernières années ainsi que pour le temps consacré à l'organisation de la soutenance. Merci aussi à Yanick Heurteaux, Thierry Lambre et Laurent Serlet pour leur implication dans la vie de l'école doctorale. Je remercie enfin le personnel administratif pour son aide bienvenue dans les différentes démarches : Valérie Sourlier, Camille Guillot, Laurence Schmitt, Sylvie Chassagne et Émilie Habouzit, j'aurais été perdu à de nombreuses reprises sans vous, alors merci. Laurence, je te souhaite une heureuse continuation.

J'aimerais adresser un mot à tou·te·s les étudiant·e·s que j'ai eu·e·s sous ma responsabilité, à la fois en tant que tuteur et chargé de TD à l'UCA. La transmission, l'enseignement, sont des aspects essentiels de la vie des chercheur·e·s et ne devraient jamais être négligés. J'ai apprécié travailler avec l'ensemble des groupes à qui j'ai eu l'honneur d'enseigner. Beaucoup d'étudiant·e·s m'ont marqué par leur curiosité et leur vivacité d'esprit. Je voudrais en particulier remercier le groupe de MP2/DLMP de l'année 2023-2024 et souhaite à tous les profs du monde de faire cours à des personnes aussi brillantes et motivées.

Au LMBP, mais aussi en dehors, j'ai eu l'honneur de forger de mémorables souvenirs en la compagnie de Hawa Farah Aden, Martin Azón, Alexandre Desmoulins (bon courage pour rester dans le bureau, mais je ne m'inquiète pas, tu sembles t'intégrer à la perfection), Clément Legrand, Martin Metodiev (merci pour la soutenance blanche !), Thi Hoà Nguyen (où est mon cadeau ?!) et Florian Tilliet. À Léo Hahn Lecler, je souhaite une belle continuation en Suisse où il fera un excellent mathématicien. Merci Rémi Boutin pour les soirées foot, les pauses café et les discussions sur de si nombreux sujets. L'énergie dont tu fais preuve malgré ton âge avancé est une source d'inspiration pour moi. Sue Claret, Tristan Guyon, je vous remercie d'avoir été des soutiens sans faille au cours de cette aventure. J'espère que le travail que nous avons mené sur les violences subies par les jeunes chercheur·e·s porteront leurs fruits. Sue, je garde Michelangelo avec moi et tâcherai de revenir t'embêter à Clermont aussi souvent que possible - si tu ne boudes pas. Et Tristan, mon "Maître Carlo", puisque le destin semble faire en sorte que l'on se suive, alors j'ai hâte d'être invité dans ton nouveau palace à Orsay et d'y partager quelques bières en ta compagnie ! Enfin, il y a mes deux plus anciens co-bureaux, Julien La Clanche et Émilien Zabeth. Je n'ose imaginer combien ce fut pour vous un honneur de me côtoyer au quotidien. Émilien, avant ton incarcération prochaine, je veux te remercier pour tous les fous rires, les discussions philosophico-politiques de haut niveau (sur les traîtres notamment) et les moments partagés en-dehors du labo. Au moment où j'écris ce paragraphe, tu viens encore de ruiner une potentielle conversation sérieuse avec une remarque incongrue. Je pourrais écrire un livre entier avec toutes tes réflexions que j'ai pris grand soin de noter, bien qu'il me soit impossible de les oublier. Ta pensée originale et ton approche singulière de l'existence ont été, plus ou moins sérieusement, une vraie source d'inspiration. Julian, tu as souvent été moqueur et désagréable quand j'ai fait en sorte de toujours rester élogieux et bienveillant à ton égard. Alors je continuerai : merci pour toutes tes suggestions culturelles, tes cadeaux bien sentis et tes connaissances encyclopédiques (clairement du "mansplaining") alliées à ta mémoire effrayante. J'ai adoré nos discussions, des plus drôles (théorie des bulles) aux plus sérieuses (Eutopia, Damasio et ton tableau préféré, que j'ai soigneusement enregistré !), où j'ai toujours admiré ton esprit de synthèse, ton empathie et ta grande pédagogie. À tous les deux, je vous souhaite une belle année en tant

qu'ATER et espère de tout cœur vous retrouver rapidement pour partager de nouveaux grands moments.

Il est émouvant de constater que, malgré les années qui s'écoulent, on continue d'éprouver une constante affection pour certaines personnes. Bertille, Justine, Louïse, Mélanie et Thomas, cela fait si longtemps que je vous connais... et c'est pourtant à chaque fois un véritable bonheur de vous retrouver. Bertille et Louïse, j'espère que nous nous verrons plus souvent dans la capitale, désormais ! Justine, je te souhaite tout le bonheur du monde avec Léopold et même si vous vous en allez en terres lyonnaises, je suis sûr que nous nous reverrons vite. Thomas, Mélanie, c'était une immense joie de vous revoir cet été et j'espère qu'il y aura plein d'autres moments comme ceux-là, quelque part entre la France et le Canada.

Un des plus grands miracles de mon existence est d'être né au sein d'une famille aimante qui m'a toujours soutenu dans mes projets. À mes deux parents et à ma petite sœur, je ne pourrai jamais assez vous remercier d'avoir rendu ma vie si douce et agréable. L'évidence et la force de votre amour me donnent aujourd'hui la possibilité de vivre un rêve, celui de défendre ma thèse devant des personnes qui comptent pour moi. Papa, Maman, Romane, vous ne comprendrez sûrement pas grand chose à ce manuscrit ni à l'exposé qui s'en suit mais qu'importe, vous vous êtes toujours montré·e·s curieus·e·s, patient·e·s et à l'écoute dans les moments délicats comme dans les moments de joie, et c'est tout ce qui compte. Aussi loin que je puisse m'en souvenir, j'ai toujours passé de chouettes moments entre Cros, la Tour et le Puy. Alors merci à Serge et Tata Ouette pour m'accueillir chez vous depuis de si nombreuses années, merci à ma mamie, aux tontons et aux tatas, aux cousins et aux cousines, ainsi qu'à mon parrain et à ma marraine, pour la tendresse dont vous faites preuve à mon égard. C'est toujours un bonheur de vous revoir. Enfin, il me faut remercier la famille Boissy qui a su m'accueillir avec beaucoup de générosité. Merci encore pour les merveilleux repas (sans fromage !) à Veyre, j'espère qu'il y en aura encore plein d'autres.

Je conclus ces remerciements en omettant probablement de nombreuses personnes qui ont joué un rôle crucial dans ma vie. Mais je n'oublierai certainement pas celle avec qui je partage tout et qui a su me porter mieux que quiconque jusqu'à cet instant. Elsa, il y a quelque chose de vertigineux à se dire qu'il y a très longtemps, un grand "boum" a rendu possible notre rencontre, l'inscrivant de manière éternelle et nécessaire au sein d'une histoire cosmique bien plus vaste. Ton humour, ton intelligence, ta présence rassurante, nos souvenirs (du marché de Cham' jusqu'à nous deux, perdu·e·s dans une montagne du Vercors) et nos projets annulent toute l'absurdité de mon existence. Je te remercie. Et je t'aime.

Abstract

This PhD thesis is dedicated to the development and analysis of sampling algorithms with applications to the inference of cosmological parameters. We first review the mathematical tools as well as the cosmological framework. Then, we introduce two algorithms. The first one is called PDMC-BORG. It is a non-reversible Markov Chain Monte Carlo sampler used for performing large-scale structure inference. It relies on the BORG framework developed by the Aquila consortium to infer the primordial density field from astronomical data. We detail the main features of the algorithm, explain how to tune it and show that its performance are similar to that of a baseline Hamiltonian Monte Carlo sampler. Then, we introduce a fixed-kinetic energy variant of Neural Hamiltonian Flows, a type of generative model that uses symplectic Hamiltonian transformations to map a base distribution on any target. Our modification allows to enhance interpretability of the model while reducing its numerical complexity. We test its performance in image generation and explain how to use Neural Hamiltonian Flows and its variants in the context of Bayesian inference, illustrating the method on the inference of two cosmological parameters from supernovae observations.

Contents

General Introduction

To effectively contain a civilization's development and disarm it across such a long span of time, there is only one way: kill its science.

Liu Cixin, *The Three-Body Problem*

The understanding and modeling of complex physical systems have greatly benefited from two important developments: the design of sampling algorithms as a way to explore efficiently high-dimensional spaces, along with the rise of computing resources. In this thesis, we are interested in probabilistic sampling algorithms: given a target probability distribution that may be analytically intractable, a sampling algorithm is an automatic procedure implemented on a computer that is able to generate samples that follow the target distribution. Such distribution may have a scientific interest. For instance, it can describe the distribution of heights within a population, the distribution of galaxies in a certain volume, or it can model images of a certain object. The main challenge is to design automatic procedures that can adapt to the high-dimensionality and to the high complexity of the distributions of interest.

One important family of sampling algorithms is called Markov Chain Monte Carlo methods (Robert and Casella, 2004). It consists in designing a chain of correlated samples that moves through parameters space and converges in distribution to the right target. The first implementation on a computer was made in the middle of the 20th century (Metropolis et al., 1953) for simulating the equation of states of physical systems. The key idea was that exact dynamics is not necessary to simulate the system. Instead, it is sufficient to design a Markov chain which converges to the correct distribution. A major step forward was achieved with Hybrid or Hamiltonian Monte Carlo (Duane et al., 1987). Originally used for quantum chromodynamics, this algorithm uses Hamiltonian dynamics to explore an augmented phase-space in a consistent way. Within the statistics community, these techniques took some time to land-off. Hastings generalized Metropolis algorithm to the non-symmetrical case (Hastings, 1970). A special case of the Metropolis-Hastings algorithm, namely the Gibbs sampler, was described in Geman and Geman (1984). It was in 1990 that Gelfand and Smith noticed that the framework was amenable to the context of Bayesian inference (Gelfand and Smith, 1990). Convergence of the Markov chain to the target distribution requires that the Markov transition kernel K, giving the probability to go from one state to another, leaves the target measure π invariant, i.e. $\pi K = \pi$. This can be achieved by a constraining sufficient condition called detailed-balance which reads, for all pair of states (x_1, x_2), $\pi(dx_1)K(x_1, dx_2) = \pi(dx_2)K(x_2, dx_1)$. This is the case for Metropolis-Hastings or Hamiltonian Monte Carlo. However, it introduces artificial symmetry in the system and leads to reversible types of samplers with a rejection step, which can move both forward and backward in parameters space. This can be a real shortcoming for efficient exploration. To overcome such issues, non-reversible samplers were developed by the Physics community. These algorithms called Event-Chain Monte Carlo (Bernard et al., 2009; Michel et al., 2014) build on the lifting framework developed in Chen et al. (1999) and Diaconis et al. (2000), which consist in adding an auxiliary variable, as done in the HMC. However, by replacing samples rejection with re-sampling of the auxiliary variable, these schemes are rejection-free

and they do not rely on detailed-balance anymore. They inspired the development of a more general framework for designing non-reversible samplers (Bierkens et al., 2016; Alexandre Bouchard-Côté and Doucet, 2018; Michel et al., 2020) based on Piecewise Deterministic Markov Processes (Davis, 1984).

Another major line of research are generative Machine Learning models. They use artificial neural networks to learn complex relations from a massive amount of data in order to be able to generate original samples from the target distribution. The first models of neural networks, date back to the middle of the 20th century (Mcculloch and Pitts, 1943; Rosenblatt, 1958). At the same epoch, Alan Turing wondered if machines could think (Turing, 1950), opening the era of artificial intelligence. At this time, the field was impeded by the lack of computing power. In the 1980s, new progress were made along with the development of computing resources. The field also benefited from theoretical improvements such as the introduction of the backpropagation algorithm for training neural networks (Rumelhart et al., 1986) and universal approximation results were stated (Cybenko, 1989). In the 2010s, generative models were massively developed and multiple architectures were introduced (Goodfellow et al., 2014; Tabak and Vanden-Eijnden, 2010; Sohl-Dickstein et al., 2015; Vaswani et al., 2017). The recent success of large language models (OpenAI et al., 2023) clearly illustrate how the combination of the deep learning paradigm with powerful numerical resources will be able to shape new horizons within the next decade. While showing impressive results, neural netowrk-based architectures still suffer from interpretability issues. Indeed, neural networks often behave as black-boxes whose decisions are hard to interpret, which can be a limitation for disciplines such as medical science. These concerns gave birth to the field of Explainable Artificial Intelligence (Samek and Müller, 2019). Other solutions come from the incorporation of prior knowledge inside the architecture to solve unsupervised tasks while respecting physical laws (Raissi et al., 2019), something called *Physics-Inspired Neural Networks*.

When designing new algorithms, one naturally seeks challenging problems on which they can be tested. Conversely, scientific problems can be a source of inspiration for designing performing algorithms. The beginning of the 21st century is shaped by the immense amount of data to process. This is for instance the case of cosmology, the branch of Physics which studies the universe as a whole object. The new generation of space telescopes and observatories will lead to an exponentially increasing amount of data for astronomers and cosmologists (Amiaux et al., 2012; Gardner et al., 2023). This sudden rise in data obliges cosmologists to develop new statistical tools and use powerful computing device to process them. Also, they have to deal with complex non-linear large-scale structures that are difficult to describe mathematically. Knowing these facts, it is then no surprise to combine state-of-the-art sampling algorithms with cosmology. This is the purpose of the Aquila consortium[1], which now gathers about 30 cosmologists and statisticians and develop state-of-the art data analysis tools for understanding the large-scale structure of the universe. Characterizing the large-scale structure is a problem of the utmost importance in cosmology, as it would give precious information about the universe through the precise determination of the cosmological parameters leading its evolution. However, this is a challenging task for multiple reasons. First, any observer has only access to a partial vision of the universe, due to its expansion and to the finite velocity of light. Second, several observational and selection bias render almost impossible the task of making a precise cartography of the sky (York et al., 2000). Finally, as the universe is made for the major part of dark matter and dark energy that cannot be observed directly (Bertone et al., 2005), it results in a fundamental limitation to capture the whole cosmos structure through direct observation.

According to the Lambda-CDM model and observations, the universe was born about 13.8 billion years ago (Planck Collaboration et al., 2020). After a short phase of inflation, primordial nucleosynthesis gave birth to the first subatomic particles. As it continued to expand, the hot dense plasma started to cool down, allowing stable atomic structures to form, a period called recombination. As a consequence, the 380,000 year-old young universe became cold and non-compact enough, along with the right stable

[1] https://www.aquila-consortium.org/

atomic structures, for electromagnetic signals to propagate. This original glow now bathes the whole cosmos and is known as the Cosmic Microwave Background (CMB) (Alpher and Herman, 1948; Penzias and Wilson, 1965). While being very homogeneous, the CMB presents very small temperature - and thus density - anisotropies. It is believed that the amplification of the latter under gravitational force led to the inhomogeneous filamentary structure that we observe at the very large scale of the Universe. Indeed, at scales of order gigaparsec, astronomical observations show that matter is concentrated along large filaments, separated by gigantic voids, something called the cosmic web (Bond et al., 1996). At the intersections of filaments, so at the hottest and densest regions of the universe, lie galaxy clusters. If the present structure seems out of reach, due to its distribution complexity, this is not the case for that of the young universe. Theory (Linde, 2008), confirmed by strong empirical evidence (Komatsu et al., 2011; Planck Collaboration et al., 2020), suggests that the 380,000-year old universe can be described by much simpler statistics. Hence the idea to work instead on the probabilistic characterization of the young universe, constrained by astronomical data from the present universe (Jasche and Wandelt, 2013; Jasche and Lavaux, 2019). From a Bayesian perspective, this is a well-posed problem: it deals with building probable beginnings for the universe according to present astronomical observations, and make them evolve through a deterministic forward-model of gravitation. This point of view also allows for overcoming a major difficulty: the story of the universe has only happened once and it is pointless to hope for the observations of other cosmological histories: hence the necessity to simulate artificial plausible universes on a computer.

Such task requires multiple ingredient: **1)** an accurate, fast and differentiable forward-model of gravitation, **2)** efficient sampling algorithms for exploring high-dimensional parameters spaces and **3)** robust statistical tools as well as a deep understanding of Cosmology for evaluating the quality of samples. The Aquila consortium has been developing an algorithmic machinery called BORG (for Bayesian Origin Reconstruction from Galaxies) that incorporate all of these elements (Jasche and Wandelt, 2013; Jasche and Lavaux, 2019). In this work, we focused on the implementation of new samplers and their performance evaluation. The idea is to provide cosmologists powerful, robust and, if possible, interpretable numerical samplers for better characterizing the large-scale structure of the universe. We provide two main contributions. The first project consists in implementing a non-reversible Monte Carlo sampler as an alternative to a more classical algorithm already used in the task of inferring the initial conditions of the universe. In the second project, we are interested in using interpretable generative models based on Hamiltonian flows. In particular, we adapted a formalism in order to use them in the context of Bayesian inference and infer cosmological parameters from supernovae observation. Each of these methods presents advantages and drawbacks. If MCMC methods, and especially their non-reversible version, are both robust and exact in terms of convergence, each sample comes with a fixed computational cost that cannot be reduced. The Normalizing Flows models that we improved, on the other hand, are numerically efficient and highly interpretable. However, they do not come with exact convergence and they are not suited for inferring parameters in very high dimension. This PhD project thus clearly illustrates a crucial point in numerical methods: there is, in the end, no shortcut. Given a sampling task, it is about finding the most suitable method at fixed computational cost and sampling quality.

How to read the manuscript. The manuscript is organized as follows: Chapter 1 is dedicated to present some important ideas in sampling in high-dimensional spaces. We detail the mathematical framework and mention the main challenges. Then, we introduce two types of methods that will be at the heart of the manuscript: Markov Chain Monte Carlo algorithms and generative models. Chapter 2 details important concepts from Cosmology, in particular the standard model of cosmology, a classical forward-model of gravitation and the Bayesian inference problem that is discussed in the following. We also introduce the BORG framework and explain how the current state-of-the-art Hamiltonian Monte Carlo sampler works. The last two chapters detail the main contributions of this thesis. Chapter 3

deals with the presentation of a non-reversible Monte Carlo sampler for inferring the initial conditions of the universe and its comparison with a baseline reversible HMC algorithm. Finally, Chapter 4 is about our work on interpretable flow-based models, in particular their use in statistical inference of cosmological parameters from supernovae observations. My hope is that the resulting document is self-contained enough so that it can serve as a modest reference for sampling algorithms in Cosmology from a mathematical perspective.

Chapter 1

Sampling in high-dimensional spaces

> *Let an ultraintelligent machine be defined as a machine that can far surpass all the intellectual activities of any man however clever. Since the design of machines is one of these intellectual activities, an ultraintelligent machine could design even better machines; there would then unquestionably be an "intelligence explosion", and the intelligence of man would be left far behind. Thus the first ultraintelligent machine is the last invention that man need ever make, provided that the machine is docile enough to tell us how to keep it under control.*
>
> Nick Bostrom, *Superintelligence*

The task of inferring the initial conditions of the universe requires exploring very high-dimensional parameters spaces. Such performance is only accessible through numerically efficient algorithms. Also, assessing convergence in this setting is highly challenging. Thus, rigorous metrics need to be employed. This chapter begins with a few elements of probability theory and introduces the concept of Bayesian statistical inference. It also explains the main challenges when performing high-dimensional inference. Then, it reviews Markov Chain Monte Carlo approaches with emphasis on recent non-reversible samplers. Finally, it introduces generative models with focus on flow-based approach as a robust way to learn complex distributions from data.

1.1 Elements of Probability and Statistics

1.1.1 Mathematical framework

Probability spaces

Probability is the branch of Mathematics which aims at formalizing the familiar concept of randomness (Durrett, 1996; Candelpergher, 2013). It is based on a rich framework called Measure theory (Pages and Briane, 2018). One may define a random experiment as an experiment whose outcome is not predictable with absolute certainty. Let us call Ω the set whose elements are the possible outcomes of such an experiment. In this manuscript, one needs to think about a d-dimensional space $\Omega = \mathbb{R}^d$. Ω is often referred to as the **sample space** and it is equipped with a σ-algebra $\mathcal{E}$, for instance the Borel algebra generated by the open sets of Ω, or the Lebesgue-measurable sets. An element of $\mathcal{E}$ is called an **event**. Finally, one says that $\mathbb{P} : \mathcal{E} \to [0, 1]$ is a probability measure on the measurable space $(\Omega, \mathcal{E})$ if

- $\mathbb{P}(\Omega) = 1$;
- $\mathbb{P}\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(E_i)$, for all countable family of disjoint sets $E_1,\ E_2, ... \in \mathcal{E}$.

The resulting triplet $(\Omega, \mathcal{E}, \mathbb{P})$ is called a **probability space**.

As a toy example, one may think of a classical coin toss independently repeated 4 consecutive times.

- The set of possible outcomes is then $\Omega = \{(o_1, o_2, o_3, o_4) \in \{\text{"heads"}, \text{"tail"}\}^4\}$.
- The set of events is $\mathcal{E} = \mathcal{P}(\Omega)$, i.e. the set of all subsets of Ω.
- A probability measure $\mathbb{P} : \mathcal{E} \to [0, 1]$ can be defined as $\mathbb{P}(E) = \dfrac{\text{card}(E)}{\text{card}(\Omega)} = \dfrac{\text{card}(E)}{16}$.

Random variables

Using the language of probability spaces allows us to properly define random quantities called **random variables**. Formally, a (multivariate) random variable is a measurable function $X : \Omega \to V$, where $(V, \mathcal{V})$ is a measurable space. It might seem paradoxal at first to define a random quantity as a *fixed* object such as a function. However, this idea will allow us to use the language of Analysis to properly manipulate random variables (Candelpergher, 2013). Randomness comes from the fact that the events themselves happen at random. In this manuscript, we will consider $(V, \mathcal{V}) = (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$.

Following our 3 coin-tosses example, the function H counting the number of "heads" within a sequence $\omega \in \Omega$ is a random variable.

Since, by definition, X is measurable, then its reciprocal image is also measurable. This allows us to define another probability measure called the **probability distribution** of X:

$$\mathbb{P}_X(B) = \mathbb{P}(X^{-1}(B)), \quad \forall B \in V$$

The latter quantity is often denoted, by abuse of notations, $\mathbb{P}(X \in B)$. An important example is that of continuous random variables defined on $\mathbb{R}^d$. In this case, the probability measure can be described through the notion of **probability density function** (pdf). A random vector X on $\mathbb{R}^d$ is said to admit a pdf f_X if for any measurable subset $B \in \mathcal{B}(\mathbb{R}^d)$, one has:

$$\mathbb{P}_X(B) = \int_B f_X(x)dx.$$

The above integral must be understood in terms of Lebesgue integration. In our framework, all probability distributions will admit a density with respect to the Lebesgue measure. Table 1.1 below lists a few examples that will be encountered throughout the manuscript.

Important mathematical quantities related to random variables are the expectation and variance. The first describes the mean value of the random variables and the second describes its average deviation to the mean. They are respectively defined as follows, for a continuous random variable X which admits a density f_X with respect to the Lebesgue measure:

$$\mathbb{E}[X] := \int x f_X(x)dx, \quad \mathbb{V}[X] := \mathbb{E}[X^2] - \mathbb{E}[X]^2.$$

In this work, we will be interested in the convergence of a random variables sequence $(X_n)_{n\geq 0}$ to a certain *limit*. There are actually multiple notions of convergence, linked to one another according to classical mathematical results. Here are a few of them (Candelpergher, 2013).

Convergence in distribution. We say that the sequence $(X_n)_{n\geq 0}$ converges in distribution (or law) to X if for all borelian A

$$\lim_{n\to\infty} \mathbb{P}(X_n \in A) = \mathbb{P}(X \in A)$$

Name	Parameters	Density function	Comments
Univariate Normal	$\mu, \sigma \in \mathbb{R}$	$f_{\mathrm{UN}}(x) = \frac{\exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}}$	When $\mu = 0$ and $\sigma = 1$, this is referred to as a *standard* Normal distribution.
Multivariate Normal	$\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{C} \in \mathcal{M}_d(\mathbb{R})$	$f_{\mathrm{MN}}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{C}) = \frac{\exp\left(\frac{-1}{2}\sum_k (\mathbf{x}-\boldsymbol{\mu})\boldsymbol{C}^{-1}(\mathbf{x}-\boldsymbol{\mu})^T\right)}{\sqrt{2\pi\lvert\boldsymbol{C}\rvert^2}}$	$\boldsymbol{C}$ is called the covariance matrix. It is symmetrical positive and definite.
Multivariate Gaussian mixture	$\forall m \in \{1, ..., M\}$, $w_m \in [0, 1]$, $\boldsymbol{\mu}_m \in \mathbb{R}^d$, $\boldsymbol{C}_m \in \mathcal{M}_d(\mathbb{R})$	$f_{\mathrm{GM}}(\mathbf{x}) = \sum_{m=1}^{M} w_m f_{\mathrm{MN}}(\mathbf{x}; \boldsymbol{\mu}_m, \boldsymbol{C}_m)$	M is the number of contributing modes. Depending on their expansion and relative distance, the number of modes within the final distribution may be smaller. Each mode has a weight w_m so that $\sum_m w_m = 1$.
Uniform	$a, b \in \mathbb{R}$	$f_{\mathrm{U}}(x) = \frac{\mathbf{1}_{[a,b]}(x)}{b-a}$	The derivatives at the bounds of the interval are not continuous.
Soft-uniform	$\alpha, \beta \in \mathbb{R}$	$f_{\mathrm{SU}}(x) = \frac{s(\alpha(x+\beta))s(\alpha(x-\beta))}{\int s(\alpha(x+\beta))s(\alpha(x-\beta))dx}$	$\mathcal{C}^\infty$ version of the uniform distribution. s is the sigmoid function $s(x) = \frac{1}{1+\exp(-x)}$.

Table 1.1: A few probability laws that will be encountered throughout the manuscript.

Convergence in probability. The sequence $(X_n)_{n\geq 0}$ converges in probability to X if for all $\varepsilon > 0$

$$\lim_{n\to\infty} \mathbb{P}(|X_n - X| > \varepsilon) = 0$$

Almost sure convergence. The sequence $(X_n)_{n\geq 0}$ is said to converge almost surely to X if

$$\mathbb{P}\left(\lim_{n\to\infty} X_n = X\right) = 1$$

Convergence in L^p. Let $p \in [1, +\infty[$ and let us suppose that $\mathbb{E}(|X_n|^p)$ and $\mathbb{E}(|X|^p)$ exist. We say that $(X_n)_{n\geq 0}$ converges in L^p to X if

$$\lim_{n\to\infty} \mathbb{E}(|X_n - X|^p) = 0$$

It can be proven that almost sure convergence implies convergence in probability, which implies convergence in distribution. L^p convergence implies convergence in probability. Convergence in distribution is thus the weakest form of convergence. Regarding convergence of random variables, we have two important results about the convergence of the empirical mean to the mean of the probability distribution. Back to our coin toss experiment, these mathematical results formalize the idea that for an unbiased coin, the frequency of 'heads' tends to 1/2 as the number of tosses tends to infinity.

- More precisely, if $(X_n)_{n\geq 1}$ is a sequence of iid random variables such that $\mathbb{E}[X_1] = \mu < +\infty$, then

$$\frac{1}{n}\sum_{k=1}^{n} X_k \to \mu \text{ in probability.}$$

  This is the **weak law of large numbers**.

- Under the exact same hypotheses,

$$\frac{1}{n}\sum_{k=1}^{n} X_k \to \mu \text{ almost surely.}$$

  This is the **strong law of large numbers**.

The last important result that we cite deals with the convergence in distribution of the sample mean to a Normal distribution. More precisely, if $(X_n)_{n\geq 1}$ is a sequence of iid random variables with $X_1 \sim \pi$. Let us suppose that $\mu := \mathbb{E}[X_1] < +\infty$ and $\sigma^2 := \mathbb{V}[X_1] < +\infty$. Let us introduce the usual sample mean $\hat{\mu}_n := \frac{1}{n}\sum_{k=1}^{n} X_k$. In this case,

$$\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma} \to \mathcal{N}(0,1) \text{ in distribution.}$$

The so-called **Central Limit Theorem** is at the basis of Statistics. It is used to build confidence intervals when estimating a parameter. It also tells us how the error is decreasing as the inverse of the square root of sample size.

Note that the above results can be easily extended to the case of multivariate distributions and/or if we replace the X_k random variables by $g(X_k)$, where g is a measurable function.

The sampling problem

Given a probability measure with density function π, which is assumed here to be continuous and smooth, the purpose of sampling is to generate a sequence of samples $x_1, x_2, ...$ that are distributed according to the target measure. Note that we may consider cases where the object π is partially or entirely unknown. Being distributed according to the target measure implicitly means convergence in law to such measure. In practice, it is broadly defined as follows. Let us consider X a random variable following the target measure. One may require that for any test function g and for a (very) large number N of samples,

$$\mathbb{E}[g(X)] \approx \frac{1}{N}\sum_{k=1}^{N} g(x_k).$$

The approximation sign means that the convergence is not necessarily ensured. Some methods guarantee exact convergence in the limit of an infinite number of samples. In practice, though, the number of outputs is limited and the sum above remains an approximation. Also, some methods do not have exact convergence properties and thus one has to deal with biases in the computation of observables.

Let us now introduce some classical sampling techniques. The most famous and basic one is called *inversion sampling* (Casella and Berger, 2001). Given a one-dimensional real random variable X following a certain probability distribution whose cumulative distribution function F_X is known, this method consists in noticing that random variable $Y = F_X^{-1}(U)$, where U follows a uniform distribution

$\mathcal{U}[0,1]$, has the same distribution as X since for all $y \in \mathbb{R}$:

$$\begin{aligned} F_Y(y) &= \mathbb{P}(Y \leq y) \\ &= \mathbb{P}(F_X^{-1}(U) \leq y) \\ &= \mathbb{P}(U \leq F_X(y)) \\ &= F_X(y) \end{aligned}$$

Note that the method can be easily generalized to higher dimensions. However, it requires having access to an analytic expression of the inverse function F_X^{-1}. In most practical cases, such quantity is intractable since the target distribution might be unnormalized or even completely unknown.

Another class of sampling techniques relies on coordinates transformations. For instance, the Box-Muller procedure consists in generating a pair of independent random variables from two uniform random variables (Box and Muller, 1958). Indeed, one can show that if $U_1, U_2 \sim \mathcal{U}[0,1]$, then the two random variables

$$\begin{aligned} Z_1 &= \sqrt{-2\ln(U_1)}\cos(2\pi U_2) \\ Z_2 &= \sqrt{-2\ln(U_2)}\sin(2\pi U_2) \end{aligned}$$

are independent and they follow a standard Normal distribution. This can be seen as a mere mapping between cartesian and polar coordinates.

1.1.2 Bayes' theorem and statistical inference

Probability provides tools for analyzing random processes, i.e. the behavior of a probabilistic model knowing its parameters. The theory is based on the analytical tools that have just been introduced. However, in this thesis, we are rather interested in determining the values of parameters from observed data. This distinction is the key difference between Probability and Statistics (Robert, 2001). Before going any further, one needs to define a coherent way of building a knowledge from data. First of all, let us notice that the probabilistic tools previously discussed do not say anything about how to interpret the concept of probability. When examining the probability $\mathbb{P}(E)$ that an event E occurs, two main viewpoints may be considered:

- the **frequentist** approach interprets the quantity $\mathbb{P}(E)$ as the limit frequency that E occurs when the experiment is independently conducted for an infinite number of times, each time in the same experimental settings;
- the **Bayesian** perspective interprets it as the belief that E occurs, before any experiment is done, assuming perfect information is available.

These two school of thoughts lead to two different ways of doing Statistics. In particular, they differ in the questions that they may ask. A frequentist will ask questions of the form: "Assuming my hypothesis is true, what is the probability for some data to realize?". A Bayesian, on the other hand, will be interested in questions of the form: "Knowing some data, what is the probability that my hypothesis is true?".

The frequentist approach deals with hypotheses and see how well collected data fit this set of hypotheses. If they don't, then the set of hypotheses is rejected. If they do, then they are not rejected - but not accepted either. Such method is based on Karl Popper's interpretation of knowledge that everything is about refutability (Popper, 1934). A scientific question is an assumption that can be contradicted by empirical data. Note that this reduces the scope of science to questions that fall into this definition. Also, one can never conclude about the validity of a theory.

On the other hand, when trying to estimate the true value of a parameter θ, a Bayesian will model it as a probability distribution conditioned on observed data $\mathcal{D}$. This quantity can be expressed using Bayes' formula:

$$\mathbb{P}(\theta|\mathcal{D}) = \frac{\mathbb{P}(\theta) \times \mathbb{P}(\mathcal{D}|\theta)}{\mathbb{P}(\mathcal{D})} \tag{1.1}$$

A Bayesian probabilistic model is thus made of two probabilistic ingredients: a prior $\mathbb{P}(\theta)$ that summarizes expert knowledge and/or historical results and/or personal believes, and a likelihood $\mathbb{P}(\mathcal{D}|\theta)$ that tells us how to incorporate new data to refine the knowledge. The resulting quantity $\mathbb{P}(\theta|\mathcal{D})$ called the posterior distribution captures our probabilistic knowledge about the parameter of interest, as a combination of our a priori and data that have been observed. The choice of prior distribution as well as the likelihood design are two major aspects of Bayesian inference that will be discussed later when introducing the models at stake. But briefly, classical methods for designing a Bayesian model include Jefrreys' priors (Jeffreys, 1946) or conjugate priors (Minton et al., 1961).

This Bayesian framework is appealing for multiple reasons, some of them being explicited in Robert (2001). For instance, if one admits that probabilistic models can represent phenomenons of interest, then it should be acceptable to model the parameters governing them with a probabilistic description, too. It also provides a consistent way to develop a subjective knowledge conditioned on data with a unique inference system. Finally, it can be used for decision-making and thus solving problems from a practical point of view, since the long-term convergence properties on which frequentist statistics are built are, in practice, never met. One could also argue that they are consistent with the frequentist approach: for instance, under mild conditions, it can be shown that the posterior distribution converges to the Dirac distribution centered on the true value of the parameter of interest (Ghosal et al., 1995).

1.1.3 Dealing with the curse of dimensionality

The core problem of the manuscript is the following: given a target distribution π, which will often be a posterior distribution, one wants to design a procedure for generating a desired number N of independent samples $x_1, ..., x_N \sim \pi$. This problem is difficult for multiple reasons:

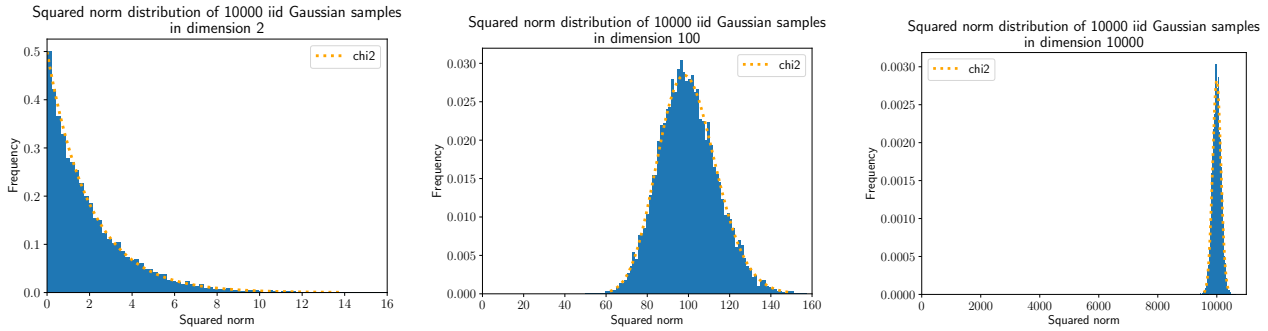
- the target distribution may be partially or totally intractable;
- the samples may live in a very high-dimensional space;
- the target distribution itself may be analytically complicated, i.e. multimodal or not smooth enough.

Hence the necessity to use automatized methods, implemented on a powerful computer, for overcoming theses difficulties.

Indeed, many distributions of interest live in very-high dimensional spaces which may cause difficulties for performing efficient sampling. The difficulty of scaling as dimension increases is referred to as the *curse of dimensionality*.

Numerical computing issues. First, one needs powerful computers to represent, store and manipulate many million or billion-dimensional vectors with a high level of precision. The inherent floating representation of real numbers in a programming language must be checked carefully in order to avoid numerical issues (Muller et al., 2018).

Structure of probability density functions in high-dimension. Also, high-dimensionality is a synonym of sparsity. This is due to a measure-concentration phenomenon (Ledoux, 2001). To see that, imagine sampling from a uniform distribution within the $[0, a]^d$ hypercube (with $0 < a < 1$) lying inside the bigger $[0, 1]^d$ hypercube. Such probability is equal to a^d. As dimension d increases, this



(a) Dimension 2 (b) Dimension 100 (c) Dimension 10,000

Figure 1.1: For high-dimensional standard Normal distributions, most of the mass lies at a distance approximately equal to $\sqrt{d}$ from the mode that does not belong to the highest probability mass region, called the typical set.

probability tends to 0, indicating that most of the region of interest will get insignificant compared to the whole space and hard to target. The algorithms must be suited to the specific geometry of the targets living in theses spaces. Another classical example is that of a standard Normal distribution. Understanding its structure is crucial as it is at the basis of many convergence results. In 1D or 2D, one may get a fairly good representation of such distribution, which looks like a bell curve or surface. However, the more dimension increases, the more the shape of a Gaussian gets thinner and thinner. This is a direct consequence of a classical results from statistics: let us consider a sequence $X_1, ..., X_n$ of random variables following a Normal distribution $\mathcal{N}(0, 1)$. Then, random variable $Z = (X_1^2 + ... + X_d^2)$ converges in distribution to a Chi-squared with d degrees of freedom. The latter has mean d and variance $\sigma^2 = \sqrt{d}$. Now, let us consider random vectors following a $\mathcal{N}(0, I_d)$, with $d >> 1$. Such vectors have coordinates d independent univariate standard Gaussian variables and one may thus apply the previous stated result to their norm. As a consequence, most of their probability mass will lie on a very thin shell located at a distance $\sqrt{d}$ from the mode. By *thin* we mean that since a high-dimensional Chi-squared is close to a Gaussian distribution, more than 99.7% of the probability mass associated to Z will be located in a band centered on the sphere with diameter $6\sigma \ll \sqrt{d}$. This region of interest is called the *typical set* of the distribution. The phenomenon is illustrated in Figure 1.1.

Consequences. These issues actually give us some clue about the approach that must be taken to tackle the problem and what we should expect from an efficient sampling algorithm in high-dimension. In terms of exploration, one should design algorithms that travel through parameter spaces and spend most of their time where it matters, i.e. in the highest probability mass regions. In a learning perspective, one should manage to get as much data as possible so that the examples are an accurate representation of the complex target distributions along each dimension but also deal with relevant architecture to process them efficiently (Zhu et al., 2016): this is a central aspect when it comes to generative modeling. These complementary approaches are both explored in the manuscript through examples drawn from cosmology.

1.2 Markov Chain Monte Carlo methods

Given a target distribution π, often known up to a normalizing constant, the idea of Monte Carlo methods is to use random samples to compute expectations of the form

$$\mu = \mathbb{E}_\pi [f] = \int \pi(s) f(s) ds.$$

If one is able to draw samples $x_1, ..., x_n$ from independent and identically distributed random variables $X_1, ..., X_n \sim \pi$, then the (strong) Law of Large Numbers ensures that $\hat{\mu}_n = \frac{1}{n} \sum_k X_k \xrightarrow{a.s} \mu$. Then, one

may treat the quantity

$$\hat{\mu}_n(x_1, ..., x_n) = \frac{1}{n} \sum_{k=1}^{n} f(x_k)$$

as an approximation to the true value of μ. The main difficulty here, as evoked in the previous Subsection, is to generate samples from the correct distribution. In high dimension, integration is hard precisely because when one has no access to analytical expressions, computing such quantities requires having access to many samples representing the target (Betancourt, 2018). The idea of MCMC methods is to design efficient numerical procedures for exploring the typical set of the target distribution, with long-term convergence guarantees. Such methods, however, introduce correlations within the output samples that need be evaluated for a full performance diagnosis.

1.2.1 Basics of Markov Chains

An important tool for high-dimensional sampling is the theory of Markov Chains (Norris, 1998). Further in the manuscript, we will indeed seek to design a Markov chain that admits the target distribution as its unique invariant distribution and generate iteratively as many samples as we want. We describe here the mathematical framework for analyzing this kind of objects. All random variables take values in a state space S. A stochastic process is a sequence of random variables $(X_i)_{i \in \mathcal{I}}$ indexed by some set $\mathcal{I}$ which can be either discrete or continuous. Informally, a Markov Chain is a stochastic process whose future state only depends on the present state, not on the past. In the context of this thesis, one has to account for the fact that **(i)** the state space is continuous and **(ii)** the indices family $\mathcal{I}$ may be non-countable.

We first present the discrete-time and discrete state space framework for building intuitions and then we move forward to more challenging settings.

Discrete time and discrete state space Markov Chains

Let us suppose that S is countable. Think for example to $S = \mathbb{Z}$. Also, the chain is indexed by the natural integers $\mathcal{I} = \mathbb{N}$. The fact that future state only depends on the current one is called the *Markov property* and it is written as:

$$\mathbb{P}(X_{n+1}|X_n, ..., X_0) = \mathbb{P}(X_{n+1}|X_n), \quad \forall n \in \mathbb{N}.$$

The chain is fully described by its initial distribution π_0 and the transition probability $K(x, y)$ from state x to y, that we suppose is independent from n. The convergence to a target distribution π is ensured if the following is verified:

1. the target is a **fixed point** for the transition kernel K, that is: $\pi K = \pi$;
2. the Markov chain is **irreducible**, that is one can always reach state y from state x, no matter x and y;
3. the Markov chain is **aperiodic**, meaning that for every state $x, y \in S$, there exists $n_{x,y} \in \mathbb{N}$ such that the probability to reach y from x in n steps is strictly positive as soon as $n \geq n_{x,y}$.

Under these conditions, the Markov Chain converges to the target distribution no matter the initial state of the chain (Levin et al., 2006), which reads:

$$\lim_{n \to \infty} \mathbb{P}(X_n = x|X_0 = x_0) = \pi(x), \quad \forall x_0, x \in S.$$

In particular, we have ergodic results. For instance:

$$\lim_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n} \mathbb{P}(X_k = x|X_0 = x_0) = \pi(x), \quad \forall x, x_0 \in S.$$

Discrete time and continuous state space Markov Chains

In this thesis, we have $S = \mathbb{R}^d$ which is not countable. Let us begin by exploring the case where the process is indexed by positive integers i.e. $\mathcal{I} = \mathbb{N}$. This will be useful to discuss some classical MCMC methods such as Metropolis-Hastings or Hamiltonian Monte Carlo. In this context, a Markov Chain is a sequence of random variables $(X_n)_{n\in\mathbf{N}}$. If we suppose once again that the transition rule from one state to another does not depend on the time - we call such a chain *homogeneous* -, the chain is entirely described by its initial distribution $X_0 \sim \pi_0$ and a Markov kernel K which, informally, describes the transition probability from one state to another. More specifically, given two measurable spaces $(X, \mathcal{X})$ and $(Y, \mathcal{Y})$, a Markov kernel is a mapping $K : \mathcal{Y} \times X \longrightarrow [0, 1]$ which satisfies the following properties:

1. for every $\mathcal{Y}_* \in \mathcal{Y}$, $x \mapsto K(\mathcal{Y}_*, x)$ is $\mathcal{X}$-measurable;
2. for every $x \in \mathcal{X}$, $\mathcal{Y}_* \mapsto K(x, \mathcal{Y}_*)$ is a probability measure on $(Y, \mathcal{Y})$.

In our case, for all open interval $B \subset \mathbb{R}^d$ and point $x \in \mathbb{R}^d$, the Markov kernel of the chain simply means that:

$$\mathbb{P}(X_{n+1} \in B | X_n = x) = K(x, B).$$

To apply convergence results, one needs to introduce the concept of Harris chain. These are special types of Makov processes that eventually return to a certain region of the state space an infinite number of times. A Markov chain is a Harris chain if there exists $A \subset S$, $\varepsilon > 0$ and a probability measure ρ such that $\rho(S) = 1$ such that:

1. no matter the starting point $x \in \Omega$, the chain will eventually return to A in a finite number of states with probability 1;
2. if $x \in A$, and $B \subset S$ is measurable, then $K(x, B) \geq \varepsilon\rho(C)$.

A Harris chain is additionally said to be **recurrent** if the first returning time to A is finite with probability 1, not matter the initial distribution.

Convergence of the process to a unique distribution π is now ensured if the following conditions are met:

1. the target is a **fixed point** for the transition kernel K, that is: $\pi K = \pi$;
2. the Markov chain is **aperiodic**, meaning in this context that $\mathbb{P}(X_n \in A | X_0 \in A) > 0$, regardless of the initial distribution.
3. the Markov chain is **recurrent Harris**.

Continuous time and continuous state space Markov Chains

This framework is the one used for describing stochastic processes (Doob, 1942) such as Piecewise Deterministic Markov Processes (PDMPs) that will be introduced later (Davis, 1984). Let us consider the probability space $(\Omega, \mathcal{E}, \mathbb{P})$. In this case, we are dealing with stochastic processes indexed by continuous time $\{X_t\}_{t\in\mathbb{R}_+}$, where the random variables take values in a measurable state space $(S, \mathcal{F})$. We assume that we have a collection of σ-algebras $\{\mathcal{E}_t\}_{\in\mathbb{R}_+}$ such that for all time $t > 0$, random variable X_t is measurable with respect to $\mathcal{E}_t$, with the additional property that for all $0 < t_1 < t_2$, one has $\mathcal{E}_{t_1} \subset \mathcal{E}_{t_2} \subset \mathcal{B}(\mathbb{R}^d)$. In this case, the random process is Markovian if for all times $s, t > 0$ and event $F \in \mathcal{F}$, one has:

$$\mathbb{P}(X_{s+t} \in F | \mathcal{E}_s) = \mathbb{P}(X_{s+t} \in F | X_s).$$

There is an equivalent formulation in terms of condition expectations: for all times $s, t > 0$ and test function g, one has:

$$\mathbb{E}[g(X_{s+t})|\mathcal{E}_s] = \mathbb{E}[g(X_{s+t})|X_s].$$

While being a bit technical, such condition is simply the mathematical translation of the fact that the future state at time $t+s$ only depends on the present one at time t, not on the previous states at times $0 < t' < t$.

Note that this framework generalizes the previous case where time is discrete. For the latter, one may use the above definitions by equipping the time space with a Borel σ-algebra and the counting measure.

Central Limit Theorem for correlated samples

By definition of the transition probability, each consecutive samples in a Markov Chain, i.e. points $x_1, ..., x_n \in \mathbb{R}^n$, seen as realizations from random variables $X_1, ..., X_n$, may be correlated. In this case, known asymptotic results such as the Laws of Large numbers or the central limit theorems do not apply directly since they are all based on the assumption that the random variables are independent and identically distributed. We investigate here asymptotic convergence results for correlated samples.

Crucially, the Law of Large Number still holds under the same assumptions. That is for all function g with finite first moment under π, one has:

$$\lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} g(X_k) = \mathbb{E}[g(X_1)].$$

However, there is a Markov Chain version of the the Central Limit Theorem that takes into account the correlated samples. More precisely, let us call $X_1, ..., X_n$ random elements from a Markov chain (representing the samples in this case) with unique stationary distribution π. Suppose that $X_1 \sim \pi$, so the chain is initialized at the stationary distribution. Now, let g be an observable function, such that $\mu = \mathbb{E}[g(X_1)] < +\infty$ and $\mathbb{V}[g(X_1)] < +\infty$. Let us introduce the usual sample mean $\hat{\mu}_n := \frac{1}{n}\sum_{k=1}^{n} g(X_k)$. In this case (Geyer, 2011),

$$\frac{\sqrt{n}(\hat{\mu}_n - \mu)}{\sigma_{\mathrm{MC}}} \to \mathcal{N}(0,1) \text{ in distribution.}$$

In the above result, we have introduced the Monte Carlo variance which equals:

$$\begin{aligned}\sigma^2_{\mathrm{MC}} &:= \mathbb{V}[g(X_1)] + 2\sum_{k=1}^{\infty} \mathrm{Cov}(g(X_1), g(X_{k+1})) \\ &= \mathbb{V}[g(X_1)]\left[1 + 2\sum_{k=1}^{\infty} \frac{\mathrm{Cov}(g(X_1), g(X_{k+1}))}{\mathbb{V}[g(X_1)]}\right]\end{aligned}$$

For $1 \leq k \leq +\infty$, the summand $\frac{\mathrm{Cov}(g(X_1), g(X_{k+1}))}{\mathbb{V}[g(X_1)]}$ is called the autocorrelation at lag k. The sum is thus a measure of the autocorrelation time between the samples. In practice, one defines the integrated autocorrelation time as

$$\tau_{\mathrm{int}} = \frac{1}{2} + \sum_{k=1}^{\infty} \frac{\mathrm{Cov}(g(X_1), g(X_{k+1}))}{\mathbb{V}[g(X_1)]}.$$

It allows us to compute the number of effective samples N_{eff}, i.e. the number of independent samples that have been generated during the process. For a sequence of N correlated samples generated by a Markov Chain, the relation reads as follows:

$$N_{\mathrm{eff}} = \frac{N}{2\tau_{\mathrm{int}}}.$$

1.2.2 Reversible MCMC algorithms

The idea of a MCMC algorithm is to explore the parameters space via a Markov chain and to generate a *chain* of samples $\{x_0, x_1 ..., x_n\}$ whose density will be equal to that of the target distribution π. The samples will then be used to compute expectations with respect to the target. In order to build such a chain, one needs to respect the ingredients defined previously in order to get existence and uniqueness of the invariant distribution, the one with density π.

A straightforward way to build samplers

For the Markov chain to converge to the target distribution, it is necessary that the Markov kernel K is a fixed-point for the target distribution π. It implies that the probability flow entering each state to be equal to the probability flow leaving that same state, a condition called *global-balance*. Mathematically, it can be written as follows:

$$\pi = \pi K \tag{1.2}$$

A sufficient condition for equation 1.2 to be verified is that the Markov chain satisfies an even stronger condition, imposing that the flow between each pair of states (x_1, x_2) offsets itself. This condition is called *detailed-balance* and it leads to the Markov chain being reversible, by adding a completely artificial local symmetry constraint. Mathematically, it can be written as:

$$\pi(dx_1)K(x_1, dx_2) = \pi(dx_2)K(x_2, dx_1) \tag{1.3}$$

Such behavior has a physical interpretation: on the microscopic scale, it is conventionally assumed that processes are indeed reversible, although the corresponding emerging macroscopic phenomenon is not necessarily so. As such, starting from a random point in the parameters space, the Markov chain will get closer and closer to the high density regions of the target. Then, it will generate states ditributed according to the target distribution. Building an efficient MCMC algorithm thus depends to a large extent on a good choice for the transition kernel $\mathcal{T}$. A historical and classical choice is given by the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970), detailed below.

The Metropolis-Hastings algorithm

Let x_0 be a starting point in the parameters space. Let $Q(x'|x)$ be a probability distribution for proposing a candidate point x' from the current point x. For example, one may think of Q as a Normal distribution with mean the current point x and fixed chosen variance $\sigma^2 > 0$. At each step, one draws a candidate x'. The latter is accepted with probability

$$p_{\text{acc}}(x, x') = \min\left(1, \frac{\pi(x')}{\pi(x)} \times \frac{Q(x|x')}{Q(x'|x)}\right)$$

If not, the old point x is repeated and the process goes on for a fixed pre-determined number of steps. One can easily check that this method obeys the invariance property from above called *detailed-balance*.

Let us note that such algorithm is particularly well-suited to the context of Bayesian inference, in which the target distribution is not analytically known but can be written as $\tilde{\pi} \times \mathcal{Z}^{-1}$, the quantity $\tilde{\pi}$ being the product of the prior and the likelihood terms. The ratio $\frac{\pi(x')}{\pi(x)}$ is thus equal to $\frac{\tilde{\pi}(x')}{\tilde{\pi}(x)}$. Finally, in the case where Q is symmetric, i.e. $Q(x|x') = Q(x'|x)$ for all x, x', then the procedure is called the Metropolis algorithm and it corresponds to the very first formulation of such an algorithm in 1953 in Metropolis et al. (1953). The generalization to the non-symmetric case was made by Hastings in 1970 (Hastings, 1970). This version called Metropolis-Hastings is written under the form of a pseudo-code in Algorithm 1.

The main limitation is that if the choice of Q is poorly made and not suited to the target, then the exploration induced by the algorithm will be inefficient. In particular, for high-dimensional targets, it

may suffer from a diffusive behavior provoking very slow exploration of the typical set of the distribution in parameters space. Indeed, as the region of high probability masses are often spherical, the proposals will often fall outside of the target region, leading to a slow acceptance rate. The resulting chain is thus doomed to spend a lot of time on the same point in parameters space. On the other hand, it is a remarkable algorithm because it only requires being able to evaluate the unnormalized function $\tilde{\pi}$ making it suitable to a very large class of target distributions.

Algorithm 1 Building a list of samples with a Metropolis-Hastings algorithm and random starting point.

Require: I, $\tilde{\pi}$, Q
Draw $\mathbf{x}^1 \sim \mathcal{N}(0, I_d)$
$X \leftarrow [\mathbf{x}^1]$
for $i \in \{1, ..., I\}$ **do**
    Draw $x' \sim Q(x'|x^i)$ ▷ Propose new state
    $p_A \leftarrow \min\left(1, \frac{\pi(x')}{\pi(x)} \times \frac{Q(x|x')}{Q(x'|x)}\right)$ ▷ Accept-reject step
    Draw $u \sim \mathcal{U}[0, 1]$
    if $u \leq p_A$ **then**
        $\mathbf{x}^{i+1} \leftarrow \mathbf{x}'$ ▷ Add new state to the chain
    else
        $\mathbf{x}^{i+1} \leftarrow \mathbf{x}^i$ ▷ Replicate previous state in the chain
    end if
    Add $\mathbf{x}^{i+1}$ to list X
end for
Return X

Hybrid or Hamiltonian Monte Carlo

Let us suppose we want to generate samples from a probability distribution with density π lying in a d-dimensional space. One way to see the problem is to consider a particle moving stochastically along the landscape shaped by the distribution. In average, the particle tends to lie in the highest probability zones of the distribution. These zones correspond to minima of its negative logarithm. To be efficient at simulating this behaviour, one may think of Physics (Duane et al., 1987). The Hamiltonian Monte Carlo algorithm for sampling aims at simulating a Hamiltonian dynamics (Landau and Lifshitz, 1982) to explore the parameters space in a consistent manner. Let us start by a few recaps on Hamiltonian dynamics.

Hamiltonian mechanics. The Hamiltonian formalism plays a great role in classical Mechanics. It provides a way to derive the equations of motions from a simple first-order system. Let us consider a system of N particles with same mass in a Galilean reference frame. In classical Mechanics, a system is fully described by its coordinates $(\mathbf{q}, \mathbf{p})$ in phase-space, with $\mathbf{q} = (q_1, ..., q_N) \in \mathbb{R}^{dN}$ and $\mathbf{p} = (p_1, ..., p_N) \in \mathbb{R}^{dN}$ the list of particles positions and momenta, respectively. From that description, it is possible to define a scalar quantity called a Hamiltonian (Landau and Lifshitz, 1982) that can be seen as the total energy of the system. It is written as the sum of a potential energy term V, solely depending on the generalized positions $\mathbf{q}$, and a kinetic energy term K, solely depending on the momenta $\mathbf{p}$:

$$H(\mathbf{q}, \mathbf{p}) = V(\mathbf{q}) + K(\mathbf{p}) \tag{1.4}$$

The system evolves in phase-space following Hamilton's equations that link the first-order time-

derivatives of the coordinates with the first-order partial derivatives of the Hamiltonian:

$$\frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \ \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}} \tag{1.5}$$

Denoting $\mathbf{z} = (\mathbf{q}, \mathbf{p})$ the points in phase-space, the system can be re-written as:

$$\frac{d\mathbf{z}}{dt} = J\nabla H(\mathbf{z}) \tag{1.6}$$

with $J := \begin{pmatrix} 0 & I_N \\ -I_N & 0 \end{pmatrix}$.

Starting from position $(\mathbf{q}_0, \mathbf{p}_0)$ in phase-space, System 1.5 defines a unique flow $\mathcal{T}_H^t(\mathbf{q}_0, \mathbf{p}_0)$ linking the initial position to that of any time $t > 0$, under smoothness conditions on H:

$$(\mathbf{q}(t), \mathbf{p}(t)) = \mathcal{T}_H^t(\mathbf{q}_0, \mathbf{p}_0)$$

Note that by reversibility of Hamilton's equations, the mapping $\mathcal{T}_H^t$ is also reversible by simply changing the sign of time in System 1.5.

Proof. Let us define the mapping $\mathcal{T}_H^{-t}$ induced by the system:

$$\frac{d\mathbf{q}}{dt} = -\frac{\partial H}{\partial \mathbf{p}}, \ \frac{d\mathbf{p}}{dt} = \frac{\partial H}{\partial \mathbf{q}} \tag{1.7}$$

Integrating Equation 1.7 yields

$$\mathcal{T}_H^{-t}(\mathbf{z}_t) = \mathbf{z}_t + \int_0^t -J\nabla H(\mathbf{z})dt$$

Now integrating Equation 1.6 gives

$$\begin{aligned} \mathbf{z}_t = \mathbf{z}_0 + \int_0^t J\nabla H(\mathbf{z})dt &\iff \mathbf{z}_0 = \mathbf{z}_t - \int_0^t J\nabla H(\mathbf{z})dt \\ &\iff \mathbf{z}_0 = \mathbf{z}_t + \int_0^t -J\nabla H(\mathbf{z})dt \\ &\iff \mathbf{z}_0 = \mathcal{T}_H^{-t}(\mathbf{z}_t) \end{aligned}$$

which proves that $\mathcal{T}_H^{-t}$ is the inverse mapping of $\mathcal{T}_H^t$. □

Another major property is that each elementary volume $d\mathbf{q}\ d\mathbf{p}$ remains constant along the trajectory. In other words, Hamiltonian mappings are volume-preserving in phase-space. This results is known as Liouville's theorem. In fact, they have an even more general property: they are symplectic. Calling B_t the Jacobian matrix of the transformation, one has:

$$B_t^T J B_t = J \tag{1.8}$$

This immediately gives volume preservation since $\det\left(B_t^T J B_t\right) = \det J \implies |\det B_t| = 1$. Out of completeness, let us prove Equation 1.8.

Proof. (Neal, 2012) Let us call Z' the Jacobian matrix of the flow induced by Equation 1.6. Differenting Equation 1.6 with respect to $\mathbf{z}$ yields

$$\frac{d}{dt}Z' = J\nabla^2 H$$

Now, differentiating $P = Z'^T J Z'$ with respect to time gives

$$\begin{aligned}\frac{d}{dt}P &= \left(\frac{d}{dt}Z'\right)^T JZ' + Z'^T J\left(\frac{d}{dt}Z'\right)\\ &= Z'^T\nabla^2 H J^T JZ' + Z'^T J^2\nabla^2 HZ'\\ &= Z'T\nabla^2 Z' - Z'^T\nabla^2 HZ'\\ &= 0.\end{aligned}$$

Hence the functional $P(t)$ is constant with respect to time. In particular, this implies that for all $t \geq 0$,

$$Z'(t)^T JZ'(t) = P(t) = P(0) = J,$$

which is the desired result. □

Numerically, the continuous solution of System 1.5 can be approached by a symplectic, invertible and stable integrator as a Leapfrog scheme 1.9, which keeps its volume-preservation property even when $\delta t \nrightarrow 0$ (Neal, 2012):

$$\left\{\begin{array}{rcl}\mathbf{p}_{n+\frac{1}{2}} &=& \mathbf{p}_n - \nabla V(\mathbf{q}_n)\times\frac{\delta t}{2},\\ \mathbf{q}_{n+1} &=& \mathbf{q}_n + \nabla K(\mathbf{p}_{n+\frac{1}{2}})\times\delta t,\\ \mathbf{p}_{n+1} &=& \mathbf{p}_{n+\frac{1}{2}} - \nabla V(\mathbf{q}_{n+1})\times\frac{\delta t}{2}.\end{array}\right. \tag{1.9}$$

MCMC with Hamiltonian mechanics. As mentioned before, any efficient MCMC algorithm should be suited to the exploration of the typical set. Hamiltonian Monte Carlo (Duane et al., 1987) exploits the geometry of the problem to propose moves that fit the region with high probability mass. To do so, the idea is to interpret the quantity $V(\mathbf{x}) = -\ln(\pi(\mathbf{x}))$ as a potential for a single particle motion. So here, $N = 1$ and $\mathbf{x} \in \mathbb{R}^d$. But having access to the gradient is not enough. If the particle finds itself is in a free fall situation, it will eventually crash into a mode of the distribution and be trapped forever. So one needs instead to build a vector field that is aligned with the typical set. Having a velocity, the particle is now able to explore the region of interest - the orbit around the mode - without falling inside the latter. This is made possible by extending the position space into a phase space by adding a momentum $\mathbf{p} \in \mathbb{R}^d$. The particle now possesses a kinetic energy $K(\mathbf{p})$ which depends on its momentum $\mathbf{p}$ (the product mass × speed). The total energy of the particle is called the Hamiltonian. It is defined as:

$$H(\mathbf{x}, \mathbf{p}) = V(\mathbf{x}) + K(\mathbf{p}). \tag{1.10}$$

In our case, the $\mathbf{x}$ vectors correspond to realizations of the interest variable. The kinetic energy is often set to a positive quadratic form as:

$$K(\mathbf{p}) = \frac{1}{2}\sum_{i,j=1}^{d} p_i \mathcal{M}^{-1}_{ij} p_j, \tag{1.11}$$

where $\mathcal{M}$ is a symmetric definite positive matrix. The tuning of the latter plays a key role in the overall efficiency. The optimal choice corresponds to the inverse of the covariance matrix of the target distribution (Neal, 2012). The Hamiltonian describing the dynamics of the system in the extended phase-space is then:

$$H(\mathbf{x}, \mathbf{p}) = -\ln\pi(\mathbf{x}) + \frac{1}{2}\sum_{i,j=1}^{d} p_i \mathcal{M}^{-1}_{ij} p_j \tag{1.12}$$

Exponentiating $-H(\mathbf{x}, \mathbf{p})$ yields:

$$\exp(-H(\mathbf{x}, \mathbf{p})) = \pi(\mathbf{x})\exp\left(-\frac{1}{2}\sum_{i,j=1}^{d} p_i \mathcal{M}^{-1}_{ij} p_j\right) \tag{1.13}$$

This quantity is the product of a Gaussian distribution of $\mathbf{p}$ and the target distribution. This latter can be obtained by marginalizing over the momenta. In summary, the HMC algorithm (Duane et al., 1987; Neal, 2012) simulates a trajectory along the probability distribution landscape following Hamiltonian dynamics. The use of a non-exact integrator requires the use of a Metropolis accept-reject step to account for numerical errors. This leads to the following pseudo-code 2:

Algorithm 2 Building a list of samples with a HMC and random starting point.

Require: $I,\ \varepsilon,\ L,\ H,\ \nabla V, \nabla K$

Draw $\mathbf{x}^1 \sim \mathcal{N}(0, I_d)$
Draw $\mathbf{p}^1 \sim \mathcal{N}(0, \mathcal{M})$
$X \leftarrow [\mathbf{x}^1]$
for $i \in \{1, ..., I\}$ **do**
    for $j \in \{1, ..., L\}$ **do** ▷ Leapfrog steps for proposing new state
        $\mathbf{p}^{j+1} \leftarrow \mathbf{p}^j - \frac{\varepsilon}{2}\nabla V(\mathbf{x}^j)$
        $\mathbf{x}^{j+1} \leftarrow \mathbf{x}^j + 2\varepsilon\mathbf{p}^{j+1}$
        $\mathbf{p}^{j+1} \leftarrow \mathbf{p}^{j+1} - \frac{\varepsilon}{2}\nabla V(\mathbf{x}^{j+1})$
    end for
    $p_A \leftarrow \min\{1, \exp(-(H(\mathbf{x}^{L+1}, \mathbf{p}^{L+1}) - H(\mathbf{x}^1, \mathbf{p}^1)))\}$ ▷ Accept-reject step
    Draw $u \sim \mathcal{U}[0, 1]$
    if $u \leq p_A$ **then**
        Add $\mathbf{x}^{L+1}$ to list X ▷ Add new state to the chain
        $\mathbf{x}^1 \leftarrow \mathbf{x}^{L+1}$
    else
        Add $\mathbf{x}^1$ to list X ▷ Replicate previous state in the chain
    end if
    Draw $\mathbf{p}^1 \sim \mathcal{N}(0, \mathcal{M})$
end for
Return X

1.2.3 Non-reversible Monte Carlo algorithms

Towards non-reversibility

We have seen in Subsection 1.2.2, how the use of the detailed-balance condition is a straightforward recipe for constructing samplers. From a sampling point of view, such condition can be a limitation. By imposing local and artificial symmetry constraints, it can lead to a slow, diffusive exploration of the parameter space. This is because the trajectory can go back and forth in between states. For high-dimensional distributions and/or those with many modes, this shortcoming can become a real curse, preventing sufficient exploration of the energy landscape associated with the target in reasonable times. This effect is particularly noticeable on anisotropic distributions in high-dimensional spaces, see for example Michel et al. (2020).

However, this phenomenon is not inevitable: it is possible to design algorithms that exploit this *'no going back'* strategy. First, one needs to remember that the detailed-balance condition is not mandatory for building Monte Carlo samplers, which need only meet the more general *global-balance* condition $\pi K = \pi$. From a historic perspective, explicit design of non-reversible Monte Carlo samplers thrived with the advent of the *lifting* framework (Chen et al., 1999; Diaconis et al., 2000). The solution consists in augmenting the parameter-space with an additional lifting variable, as in done for Hamiltonian Monte Carlo (Duane et al., 1987). The convergence conditions devised in Subsection 1.2.1 now apply to the target product measure $\rho = \pi \otimes \nu$ where π is the target distribution in parameters space and ν is the distribution of the extra-parameter. This additional parameter allows to get rid of the accept-reject

step of classical MCMC algorithms. Rejections, which often happen when the proposed state is unlikely or, in some cases, impossible, are now replaced with resampling of the lifting variable. For instance, if the latter is the direction of the trajectory, a rejection would now be replaced by a direction change that would bring the process closer to the highest probability mass zone of the distribution. This led to the development of a non-reversible version of the Metropolis-Hastings algorithm (Gustafson, 1998; Turitsyn et al., 2011) that is presented in Algorithm 3. Note that an additional refreshment of the lifting variable may be introduced to ensure ergodicity. These methods do break detailed balance. However, they still rely on a local symmetry constraint called the *skew-detailed balance* condition (Turitsyn et al., 2011).

More recently, within the Physics community, an even more general framework getting rid of the (skew) detailed-balance condition was developed. It consists in drawing ballistic moves in parameters space with direction changes happening at random times, forming a so-called *event-chain* (Bernard et al., 2009; Michel et al., 2014). The procedure is summarized in Algorithm 4. They were formally derived as the infinitesimal time-limit of lifted chains. Event-Chain Monte Carlo (ECMC) methods have shown impressive results regarding the simulation of hard-sphere systems were later characterized by the Mathematics community as a certain class of stochastic processes, as shall be seen in the next Subsection.

Since then, new classes of algorithms based on the sole *global-balance* condition have been developed. First coming from Physics, they were re-discovered by the Mathematics community through the use of analytical tools from Markovian stochastic processes (Bierkens et al., 2016; Alexandre Bouchard-Côté and Doucet, 2018; Michel et al., 2020). Since they dispense with the need to introduce artificial symmetries, these algorithms lose their reversibility property and are much more efficient in their exploration.

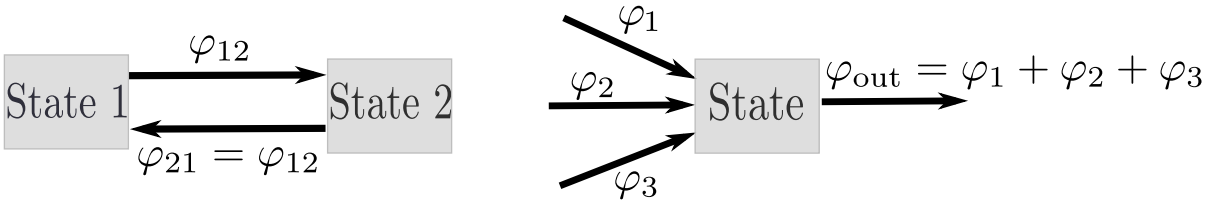


Figure 1.2: On the left, an illustration of the *detailed-balance* condition. On the right, an illustration of the *global-balance* condition. The quantity φ_{ab}, in a discrete case, represents the probability $\pi_a P_{ab}$ that flows from state a to state b in one step, π_a being the probability to be in state a and P_{ab} being the probability to go from state a to state b. One can easily check that the first condition implies the second one.

PDMP-based Monte Carlo

ECMC and more recent algorithms such as the Bouncy Particle Sampler (Alexandre Bouchard-Côté and Doucet, 2018) or the Zig-Zag Sampler (Bierkens et al., 2016) can be characterized as a certain class of class of stochastic processes, namely *Piecewise Deterministic Markov Processes* (PDMP) Davis (1984). PDMP are continuous-time stochastic processes $(\mathbf{Z}_t)_{t\geq 0}$, with values denoted $\mathbf{z}_t$, which evolves deterministically between random events that are drawn according to an inhomogeneous Poisson process. PDMPs therefore consist of four quantities:

- an initial condition $\mathbf{z}_0$;
- a deterministic dynamics governed by an Ordinary Differential Equation between the random events
$$\frac{d\mathbf{z}_t}{dt} = \phi(\mathbf{z}_t),$$
which induces a differential flow Φ ;

Algorithm 3 Building a list of samples with a lifted Metropolis-Hastings algorithm and random starting point.

Require: $I,\ \tilde{\pi},\ \Phi$

Draw $\mathbf{x}^0 \sim \mathcal{N}(0, I_d)$
Draw $\mathbf{v} \sim \nu$
for $i \in \{1, ..., I\}$ **do**
    Draw $\mathbf{x}' \sim \Phi(\mathbf{x}^{i-1}, \mathbf{v})$ ▷ Propose state using current position and lifting variable
    $p_A \leftarrow \min\left(1, \frac{\pi(\mathbf{x}')}{\pi(\mathbf{x}^{i-1})}\right)$ ▷ Accept-reject step
    Draw $u \sim \mathcal{U}[0, 1]$
    if $u \leq p_A$ **then**
        $\mathbf{x}^i \leftarrow \mathbf{x}'$ ▷ Add new state to the chain
    else
        $\mathbf{v} \leftarrow -\mathbf{v}$ ▷ Flip direction of proposal
        $\mathbf{x}^i \leftarrow \mathbf{x}^{i-1}$ ▷ Replicate previous sample in the chain
    end if
end for
Return $\mathbf{x}_0, ..., \mathbf{x}_I$

Algorithm 4 Building a list of samples with a straight Event-Chain algorithm and random starting point.

Require: $I,\ \delta t_{\text{ref}}$

Draw $\mathbf{x}^0 \sim \mathcal{N}(0, I_d)$
$\mathbf{x}' \leftarrow \mathbf{x}^0$
for $i \in \{1, ..., I\}$ **do**
    $t_{\text{toRef}} \leftarrow \delta t_{\text{ref}}$
    Draw $\mathbf{e} \sim \mathcal{U}(\mathcal{S}^{d-1})$ ▷ Draw random direction
    bool $\leftarrow$ True
    while bool **do**
        $\mathbf{x} \leftarrow \mathbf{x}'$
        Draw $u \sim \mathcal{U}[0, 1]$
        $\Delta E \leftarrow -\ln u$
        $\Delta s \leftarrow \min_T \int_0^T \max(0, \langle \nabla E(\mathbf{x} + s\mathbf{e}), \mathbf{e} \rangle) ds$
        if $t_{\text{toRef}} < \Delta s$ **then** ▷ A sample needs to be saved
            $\mathbf{x}' \leftarrow \mathbf{x} + \mathbf{t}_{\text{toRef}}\mathbf{e}$
            $\mathbf{x}^k \leftarrow \mathbf{x}'$
            bool $\leftarrow$ False
        else ▷ An event needs to be solved
            $\mathbf{x}' \leftarrow \mathbf{x} + \Delta s\mathbf{e}$
            $t_{\text{toRef}} \leftarrow t_{\text{toRef}} - \Delta s$
            $\mathbf{e} \leftarrow -\mathbf{e}$ ▷ Update velocity by reversing its sign
        end if
    end while
end for
Return $\mathbf{x}_0, ..., \mathbf{x}_I$

- event times decided by an inhomogeneous Poisson process of rate $\lambda(\mathbf{z}_t)$;
- a transition kernel $Q(\cdot|\mathbf{z}_t)$ to decide the modification of $\mathbf{z}_t$ at events.

Several non-reversible Monte Carlo algorithms based on PDMPs have been developed. They are called *Piecewise Deterministic Monte Carlo methods* (PDMC). These algorithms involve exploring the target distribution by following ballistic movements between each event, where the direction is redrawn. In all these algorithms, one needs to extend the space of parameters of interest, with positions denoted $\mathbf{x}$, with additional velocities or directions, i.e. normalized velocities, denoted $\mathbf{v}$. This is very close in approach to the lifting framework discussed above, except this time the underlying mathematical process is continuous. The event times are decided by simulating an inhomogeneous Poisson process. A PDMC algorithm therefore consists of three ingredients:

1. an initial state $\mathbf{z}_0 = (\mathbf{x}_0, \mathbf{v}_0)$;
2. a deterministic dynamics governed by an Ordinary Differential Equation between the random events
$$\frac{d\mathbf{z}_t}{dt} = \phi(\mathbf{z}_t),$$
which induces a differential flow Φ. In popular PDMC techniques, these are often translations (Bierkens et al., 2016; Alexandre Bouchard-Côté and Doucet, 2018; Michel et al., 2020);
3. an inhomogeneous Poisson process with rate $\lambda(t)$;
4. a kernel $Q(\mathbf{x})$ to decide on changes in the direction of events.

Sampling then takes place uniformly along the continuous trajectory drawn by the process $(\mathbf{x}_t)_{t\geq 0}$, as illustrated in Figure 1.3. Interestingly, PDMC algorithms do not need a Metropolis accept-reject step, as ergodicity comes from the right sampling of the event times and following the deterministic process defined by the underlying PDMP. They are thus referred to as *rejection-free* methods.

Formally, PDMPs can be described by an infinitesimal generator, describing its evolution in the infinitesimal time limit. For all smooth and bounded test function g, denoting $\mathbf{z} := (\mathbf{x}, \mathbf{v})$, it takes the following form (Davis, 1984):

$$\begin{aligned}\mathcal{U}g(\mathbf{z}) &= \lim_{t\to 0^+} \frac{\mathbb{E}[g(\mathbf{z}(t))] - g(\mathbf{z}(t))}{t} \\ &= \langle \phi(\mathbf{z}), \nabla g(\mathbf{z})\rangle + \lambda(\mathbf{z}) \int [g(\mathbf{z}') - g(\mathbf{z})] Q(\mathbf{z}', \mathbf{z}) d\mathbf{z}'.\end{aligned}$$

The first term describes the deterministic evolution induced by the differential flow Φ. The second term describes the velocity changes at events, which occur at rate λ according to the Q kernel. It can be proven (Davis, 1984) that the process admits ρ as an invariant distribution if and only if, for all test function g,
$$\int \mathcal{U}g(\mathbf{z})\rho(\mathbf{z})d\mathbf{z} = 0.$$

Some classical PDMC algorithms

Out of the PDMC litterature, two main samplers have emerged: the Bouncy Particle Sampler (Peters and de With, 2012; Alexandre Bouchard-Côté and Doucet, 2018) and the Zig-Zag sampler (Bierkens et al., 2016). They are built around two different PDMPs and differ in the way new directions are drawn.

Zig-Zag Sampler. The process starts at a position $\mathbf{x}_0$ with initial velocity $\mathbf{v}_0 \in \{-1, 1\}^d$. The idea is to simulate d independent Poisson clocks distributed, for all $i \in \{1, ..., d\}$, according to the cumulative

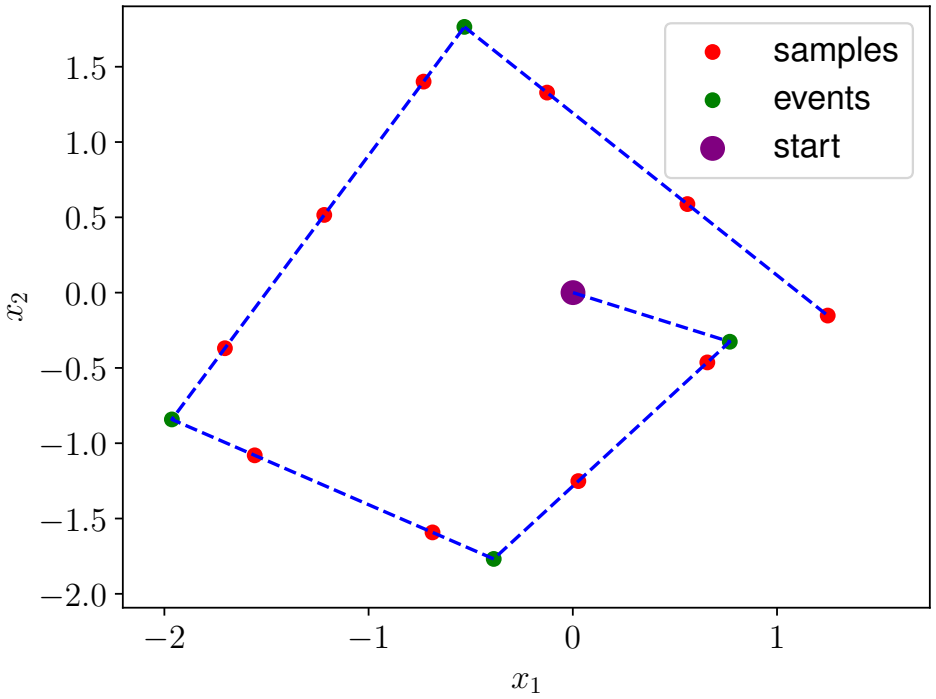


Figure 1.3: Trajectory drawn by a PDMC sampler with ballistic moves in between random events and periodic sampling along the trajectory. The target is a standard Normal distribution $\mathcal{N}(0, I_2)$.

distribution function $\exp\left(-\int_0^t \lambda_i dt\right)$. The first time one clock with index i rings corresponds to one event, at which the direction sign along the i-th component of the velocity is flipped. The process thus describes some sort of *zig-zags* in parameters space.

Bouncy Particle Sampler. Starting at position $\mathbf{x}_0 \in \mathbb{R}^d$ with initial velocity $\mathbf{v}_0 \in \mathbb{R}^d$, the BPS procedure consists in following ballistic moves in between events. At an event, occuring at position $\mathbf{x}^*$ and velocity $\mathbf{v}^*$, a new velocity $\mathbf{v}'$ is drawn by performing a reflection (a so-called *bounce*) against the orthogonal plane to the gradient, as:

$$\mathbf{v}' = \mathbf{v}^* - 2\frac{\langle \mathbf{v}^*, \nabla E(\mathbf{x}^*)\rangle}{||\nabla E(\mathbf{x}^*)||^2}\nabla E(\mathbf{x}^*).$$

For ergodicity purpose, an additional refreshment step is included: at fixed interval δt_{ref}, a new random velocity is drawn. This is necessary for example in the case where the target distribution is symmetrical and the process can get stuck in a single energy level. It is worth mentioning that one can make connections between the BPS process and Hamiltonian dynamics (Deligiannidis et al., 2020): Randomized Hamiltonian Monte Carlo can be derived as the scaling limit of the BPS process, which could give some guidance about theoretical motivations so as to why following Hamiltonian dynamics is a relevant choice when targeting complex distributions.

Forward-Event Chain Sampler. This is a class of sampling algorithms which generalizes Piecewise Deterministic Monte Carlo samplers and whose strength lies in the introduction of stochasticity at events (Michel et al., 2020). Indeed, at each event, the direction is re-drawn by randomly decomposing the velocity vector into a component orthogonal to the gradient and a component parallel to it, as illustrated in Figure 1.4. More precisely, noting $\mathbf{e}_0$ the incident direction, the outgoing direction is written as:

$$\mathbf{e}_1 = a\mathbf{n}_{\text{perp}} - b\mathbf{n}_{\text{par}}$$

with :

- $a = \nu^{1/(d-1)}$, $b = \sqrt{1-a^2}$ and ν a number drawn uniformly between 0 and 1 ;
- $\mathbf{n}_{\text{par}} = \frac{\nabla E(\mathbf{x})}{||\nabla E(\mathbf{x})||}$ the normalized gradient vector;
- $\mathbf{n}_{\text{perp}} = \frac{\mathbf{e}_0 - \langle \mathbf{n}_{\text{par}}, \mathbf{e}_0 \rangle \mathbf{n}_{\text{par}}}{||\mathbf{e}_0 - \langle \mathbf{n}_{\text{par}}, \mathbf{e}_0 \rangle \mathbf{n}_{\text{par}}||}$ the normalized projection of the incident direction onto the plane orthogonal to the gradient.

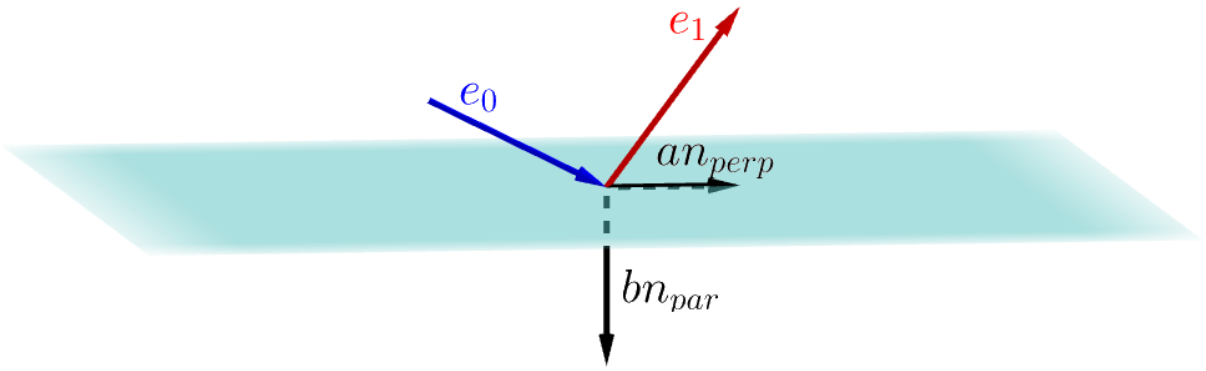


Figure 1.4: Direction change at an event with a Forward Event Chain Monte Carlo algorithm. In sky blue, the orthogonal plane to the gradient vector, which is parallel and pointing in the same direction as $\mathbf{n}_{\text{par}}$. Numbers a and b are drawn randomly at each event.

In particular, these algorithms generalize the Bouncy Particle Sampler (Alexandre Bouchard-Côté and Doucet, 2018), which can be seen as an instance of this class in which the perpendicular component of the outgoing vector is the same as that of the incident vector and the parallel component of the outgoing vector is the opposite of that of the incident vector.

Event times computation in PDMC

The most difficult step concerns the simulation of the inhomogeneous Poisson process, to draw the event times. The rate of the inhomogeneous Poisson process is given by $\lambda(t) = \max(0, \langle \nabla E(\mathbf{x} + t\mathbf{e}), \mathbf{e} \rangle)$. Here, in the second term of the maximum function, we recognise the directional derivative of E along direction $\mathbf{e}$ starting at position $\mathbf{x}$. So, it is about finding t such that $\int_0^t \lambda(s)ds = -\ln u$, for $u \sim \mathcal{U}]0,1[$. In very specific cases, for instance when the energy function E is analytically known, it is possible to solve the equation exactly. In more general cases, it is a hard problem.

Another way to do this is to run a thinning algorithm. Suppose one has access to a global bound $\overline{\lambda} = \sup_t \lambda(t)$ on the event rate. Then, one can simulate an IPP with rate $\lambda(t)$ by first simulating a Poisson process with fixed rate $\overline{\lambda}$ and then accepting each point with probability $\frac{\lambda(t)}{\overline{\lambda}}$ (Lewis and Shedler, 1979). The thinning procedure is detailed in Algorithm 5. The main difficulty here is to find such global bound: in most cases, this quantity is intractable so one cannot derive it analytically. Also, the bound must be sharp for the algorithm to be numerically efficient. To circumvent this issue, one idea consists in turning this task into a local problem (Corbella et al., 2022).

Let us fix a timestep $t_{\max} > 0$. If one is able to derive a local bound in each interval of the form $[t, t + t_{\max}]$, then the idea is to use such a bound to reduce ourselves to the simulation of a homogeneous Poisson process, which we can then *thin* to get the required inhomogeneous Poisson process (IPP) with rate $\lambda(t)$.

Algorithm 5 Simulating a N-point inhomogeneous Poisson process with rate $\lambda(t)$ using a thinning procedure and knowing a global bound $\overline{\lambda}$ on the rate.

Require: $\overline{\lambda}$, n.
$t \leftarrow 0$
$n \leftarrow 0$
while $n < N$ **do**
  Draw $u \sim \mathcal{U}]0,1[$
  $t \leftarrow t - \frac{\ln u}{\overline{\lambda}}$
  Draw $v \sim \mathcal{U}]0,1[$
  if $v \leq \frac{\lambda(t)}{\overline{\lambda}}$ **then**
    $n \leftarrow n + 1$
    $t_n \leftarrow t$
  end if
end while
Return $t_1, ..., t_N$

Local thinning strategy thus consists in simulating an inhomogeneous Poisson process using a local bound in order to derive the next event time:

- If no event time is sampled in the interval, then the process goes directly to $\mathbf{x}_{t=t_0+t_{\max}}$ following its ballistic trajectory with no direction update: this will be referred to as a *fake event*.
- if one event t_e is sampled in the interval, then the process goes directly to $\mathbf{x}_{t=t_0+t_e}$ and a new direction is picked: this will be referred to as a *true event*.

And sampling of event times continues with computation of a new bound relative to the new position and direction.

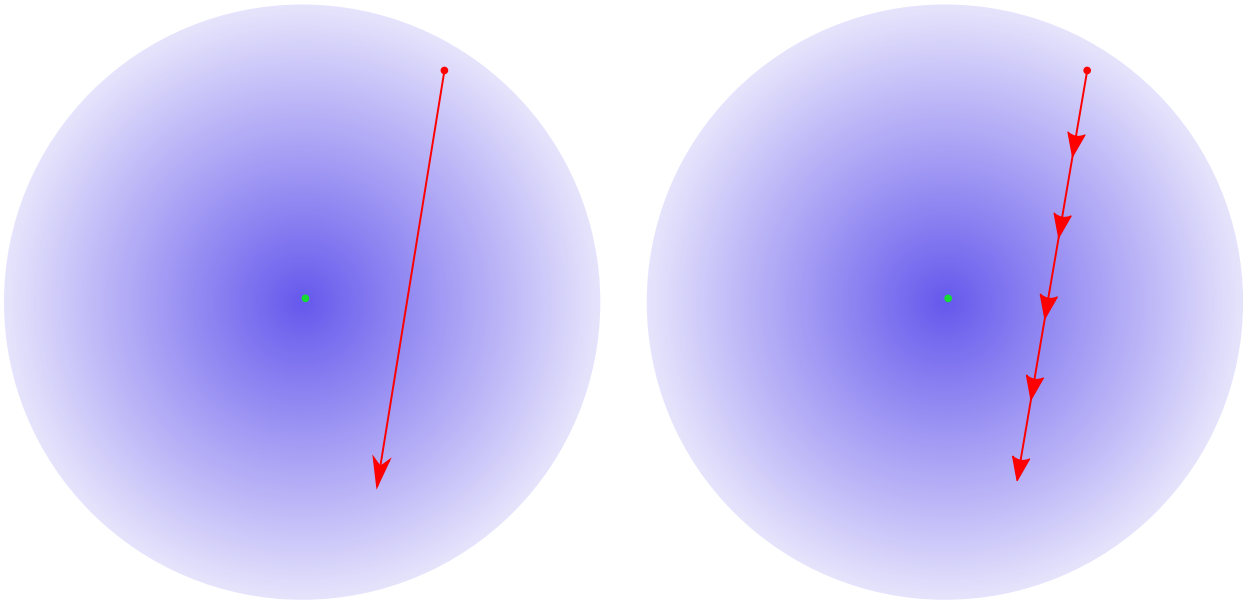
The introduction of a thinning procedure will inevitably provoke numerical difficulties. If one could derive the event times analytically, then PDMC would be efficient at finding the typical set by successive reflections by following the gradient of the target with ballistic moves, which is an improvement over a classical Hamiltonian dynamics. However, our version is limited by the fact that one has to set a maximum timestep $t_{\max}$ requiring at least one gradient evaluation. At each step, the proces cannot run through more than $t_{\max}$. If the latter is too small, then it will struggle moving forward, which is particularly inefficient when descending to the mode, i.e. when the direction is opposed to the gradient. On the other hand, when it has reached the target region, then setting $t_{\max}$ too big will result in many unnecessary ratio computations within the thinning procedure. An illustration is proposed in Figure 1.5. A sweet spot has to be found, which depends on the phase of the trajectory. Our algorithm is specifically tuned to be efficient once the process lives on the typical set. However, such choice results in very slow exploration outside from this region. Hence the necessity to adapt the thinning strategy to the current location of the process.

In the absence of analytical expression for the bound or theoretical properties about the energy landscape associated to the problem, one possibility is to rely on an automatic procedure for efficient thinning (Corbella et al., 2022; Pagani et al., 2024). The purpose of these algorithms is to be applied on problems for which only few knowledge about the target distribution is available. They rely on an optimization algorithm for finding a local maximum and then perform local thinning. The clear advantage is the algorithm capacity to target any smooth distribution. The main drawbacks are listed below:

- Numerical cost: running the optimization procedure induces a cost. At each iteration, one needs multiple evaluations of the gradient associated to the distribution. This is the most obvious

limitation of the procedure which in practice makes the use of an automatic PDMC algorithm prohibitive beyond a certain dimensionality or model complexity.

- Convergence of the optimization procedure: it may suffer from dramatic issues if the energetic landscape is too complex. In the case of highly-multimodal distributions, the procedure can get stuck in a local extremum thus giving a wrong local bound. However, choosing the $t_{\max}$ parameter sufficiently small, one may hope that the energy has some nice properties in intervals of the form $[t, t + t_{\max}]$ - increasing, decreasing, convex, concave, etc.
- Additional hyperparameters: one needs to properly tune the new parameters corresponding to the optimization process, in particular the choice of the tolerance error for convergence. There must be a trade-off between the computational cost and the quality of the bound.
- Overall dynamics: the use of a numerical optimizer does not mitigate the problem of exploring efficiently the parameters space. In particular, the main issue is the choice of the initial direction. If it is not properly aligned with the target mode region, the algorithm may escape from the target region and end up being lost in the tail of the distribution. One possible option is to combine an automatic PDMC with a mode-searcher to get to the right region at the beginning of the run.



(a) Without thinning (b) With thinning

Figure 1.5: If the event times computation is analytic (Left), then the trajectory is able to follow long ballistic moves during mixing. On the other hand, the thinning procedure (Right), which introduces an artificial $t_{\max}$ parameter, limits the exploratory capacity of the algorithm, at fixed computational budget.

Refreshment step

To make sure that the process is ergodic, some PDMC algorithms such as the Bouncy Particle Sampler of the Forward Event Chain need to introduce a refreshment of direction. This is to avoid being trapped in a constant energy level, as it may happen with symmetrical distributions. Any refreshment strategy is valid as long as the target distribution is left invariant. One popular option is to refresh the direction at fixed time interval δt_{ref} or, similarly, at the arrival of a homogeneous Poisson process with rate δt_{ref} (Michel et al., 2014; Alexandre Bouchard-Côté and Doucet, 2018; Michel et al., 2020). In Michel et al.

(2020), the authors explain how refreshment can be performed at events instead. This can be achieved by fixing a refreshment probability p_{ref} and refresh with such probability when an event occurs. Another option is to set up a clock initialized δt_{ref} and refresh at the next event after the clock has reached 0.

Refreshment of direction can be fully random, in the sense that it does not take into account any information about the process location or the initial direction. However, it can be advantageous to slightly alter a good initial direction naturally picked by the underlying PDMP process.

1.3 Generative models

Generative models have become a ubiquitous tool for sampling unknown probability distributions. Their success lies in their formidable ability to learn from massive data, in particular by exploiting their intrinsic symmetries (Bronstein et al., 2021). More precisely, in the classical case of interest here, it is assumed that the data consist of independent realizations of a target probability distribution π, which may be completely unknown. A generative model is an architecture based on artificial neural networks which, starting from inputs, produces outputs that are distributed according to the target distribution of interest (Goodfellow et al., 2016). The deviation of the model distribution from the target is measured by a certain distance or rather a divergence since it is not symmetrical. The training phase consists of adjusting the model's parameters, i.e. the weights and biases of its neurons, to minimize this divergence, turning this task into an optimization problem. Once properly trained, the model is then able to generate original samples that resemble those from the training set.

1.3.1 Artificial Neural Networks

Perceptrons

Learning from data requires building models that are able to learn complex relations from data and can approximate a large number of functions classes. Inspired by biological brains, artificial neural networks consist in interconnected individual units called *neurons*, often aggregated into *layers*. Each neuron receives signals from its neighbors and outputs another signal. The first model of neuron called *perceptron* dates back to the middle of the 20th century (Rosenblatt, 1958). A single neuron is modeled as in Figure1.6. Given inputs under the form of real numbers $x_1, ..., x_n \in \mathbb{R}$, the neuron outputs another real number computed as $o(x_1, ..., x_n) = s\left(b + \sum_{k=1}^{n} a_k x_k\right)$. Parameters $a_1, ..., a_k \in \mathbb{R}$ are called the *weights* as they give more or less importance to each unit of input information. Parameter b is called the bias and it is here to add more flexibility to the model. Finally, a non-linear function s is applied to the sum so that the model can learn non-linear transformations. This function is often called the *activation* by analogy with biological brain where a neuron is only activated beyond a certain limit.

Multilayer perceptrons

In practice, though, models often consist of multiple neurons aggregated in complex architectures. The most basic example is that of a *MultiLayer Perceptron*. A MLP consists in multiple layers made of neurons: an input layer, one ore more hidden layers and finally an output layer. The number of neurons in a layer is called the width and the total number of layers is called the depth. An illustration can be found in Figure 1.7. In the deep learning paradigm, As such, a MLP is often described by its dimension. For instance, a MLP with size $(1, 100, 200, 2)$ is made of:

- an input layer with 1 neuron;
- two hidden layers, the first one with 100 neurons and the second one with 200 neurons;
- an output layer made of 2 neurons.

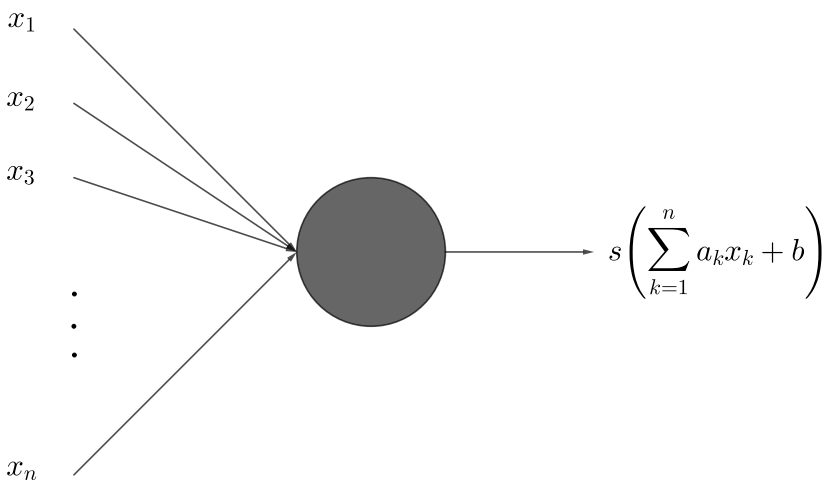


Figure 1.6: Schematic model of a single neuron.

Under mild conditions, MultiLayer Perceptrons equipped with non-linear activation functions have the capacity to learn any continuous function (Cybenko, 1989). Neural networks with this property are called *universal approximators*. More complex types of architectures exist, each one suited to solving different kind of problems depending on the invariance or equivariance properties that need to be preserved (Bronstein et al., 2021). For instance, Convolutional Neural Networks (Fukushima, 1980) are known to be translation-invariant, hence their ubiquitous use in image classification problems (Ciregan et al., 2012).

Training with backpropagation

Given an objective L, often called the loss function, the training step consists in optimizing the model parameters, i.e. its weights and biases, in order to minimize the loss. This is made through an algorithm called *backpropagation* (Werbos, 1994; Rumelhart et al., 1986; Lecun, 1987). This learning procedure is at the heart of Machine Learning. Intuitively, it consists in performing a gradient descent on the model parameters in order to find the minimum of the objective function. It turns the learning task into an optimization problem, in dimension d, where d is the number of trainable parameters. Now, one can immediately see that having a complex model will make it more flexible but at the cost of training. We are once again back to the same issues as before: the dimensionality of the optimization procedure will directly impact the difficulty of the learning step.

In practice, backpropagation works as follows:

- **Step 1: forward pass.** Let x be an input. For a generative model, this would correspond to a sample from the training dataset, i.e. from the target distribution. This input is fed into the neural network and transformed into $y = NN(x)$, the corresponding output.
- **Step 2: computing gradient.** For the sake of simplicity, let us assume that the model has no biases and only weights. Now, one needs to evaluate the partial derivative of the objective function L with respect to each weight w_{ij}^k. Here, w_{ij}^k is the weight for the connection between the i-th neuron from layer $k-1$ and neuron j from layer k. Using the chain rule, one has:

$$\frac{\partial L}{\partial w_{i,j}^k} = \frac{\partial L}{\partial z_j} \frac{\partial z_j^k}{\partial w_{ij}^k} = \frac{\partial L}{\partial z_j^k} \frac{\partial z_j^k}{\partial n_j^k} \frac{\partial n_j^k}{\partial w_{ij}^k}.$$

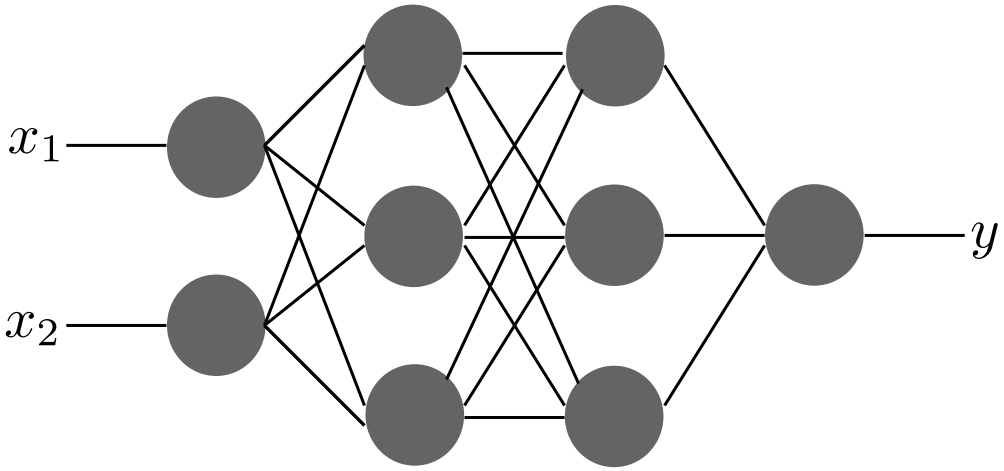


Figure 1.7: Example of MLP which transforms a 2-dimensional input (x_1, x_2) into a 1-dimensional output y. The network is made of an input layer with 2 neurons, two hidden layers with 3 neurons each and an output layer with 1 neuron. The architecture could for instance be used to map a 2-dimensional real vector onto a scalar number representing its energy.

Here, we denoted $z_j^k = s\left(\sum_m w_{mj}^k x_m\right)$ and $n_j^k = \sum_m w_{mj}^k x_m$.

- **Step 3: optimization step.** Use a gradient descent step to update the model parameters. Multiple schemes can be implemented, among which the most popular is the so-called Adam algorithm (Kingma and Ba, 2015).

Various strategies regarding the training procedure can be done. They differ in the number of updates over one epoch.

- Online learning: in this case, training samples are taken one by one and the parameters are updated after each example pass.
- Full batch learning: we compute the gradients for each training example, and then sum them together. The parameters are computed using this aggregated gradient.
- Minibatch learning: this is an intermediate strategy in which the gradients are updated multiple times during one epoch. One needs to fix a minibatch size corresponding to the number of training examples used to perform one gradient update. The resulting gradient is thus an approximation of the true gradient.

Full learning certainly has the most desired theoretical properties in the sense that it computes the true gradient and it allows for smooth updates following the steepest slope direction. However, this smoothness may be an obvious drawback in the case of complex optimization landscapes with multiple local minima. Also, it requires being able to retain in memory a lot of information and it may not be suited to large datasets and/or high-dimensional problems. On the opposite, online learning may be too noisy and is highly dependent on the training example. Hence a certain lack of reproducibility. Minibatch training is thus a common strategy even if it is not suited to particular data, for instance connected datasets. One drawback is that it requires an additional hyperparameter, i.e. the minibatch size, to perform efficient learning.

1.3.2 Popular models

We review popular and interesting architectures for generative modeling in regards of this manuscript. We specifically highlight a few applications in astrophysics.

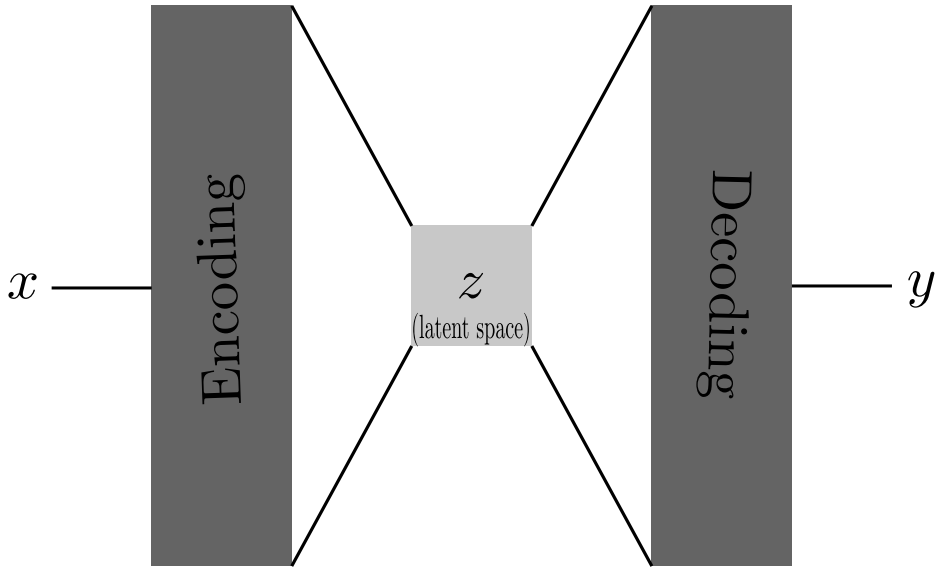


Figure 1.8: Illustration of the Autoencoder architecture.

Autoencoders

They consist of two parts: an Encoder function E, which takes a training sample and sends it to a lower-dimensional space, the latent space; and a Decoder function D, which re-maps this representation to a space of same dimension as the original (Rumelhart and McClelland, 1987; Kramer, 1991). In practice, E and D are parameterized by neural networks. An illustration can be found in Figure 1.8. Training is made by minimizing the reconstruction loss, which is written as follows:

$$R = \mathbb{E}_{\mathbf{x}\sim\text{data}}\left[\ell(\mathbf{x}, D \circ E(\mathbf{x}))\right].$$

Popular choices for the distance ℓ include the squared norm. To generate new samples, one can use the latent code to generate encoded representations of the target distribution, which may for instance refer to the mean and standard deviation of a Gaussian distribution, and then pass it through the trained Decoder. This class of models is also interesting for its Encoder part, which will be found in the architecture at the heart of this thesis, where it is used to perform information compression from a high-dimensional space. In the context of astrophysics, generative Autoencoders can be used for instance to model galaxy images (Ravanbakhsh et al., 2016).

Generative Adversarial Networks (GANs)

GANs (Goodfellow et al., 2014) are composed of two antagonistic neural networks:

- the Generator aims at producing data as close as possible to the target distribution;
- the Discriminator aims at making a distinction between training data and data artificially produced by the Generator.

An illustration of the architecture is presented in Figure 1.9. The Generator's aim is therefore to deceive the Discriminator, while the Discriminator's aim is to confuse the Generator. Training the model is equivalent to solving a two-players minimax game described as:

$$\min_G \max_D V(G, D) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log G(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\text{prior}}}[\log(1 - D(G(\mathbf{z})))]$$

In practice, the Discriminator G and the Generator G are trained simultaneously according to the procedure described in Algorithm 6.

Algorithm 6 Training phase for a GAN

for $i \in \{1, ..., N_{\text{epochs}}\}$ **do**
  for $j \in \{1, ..., \lceil \frac{\text{size training dataset}}{\text{minibatch size}} \rceil \}$ **do**
    Sample $\mathbf{z} \sim p_{\text{prior}}$ with labels $\mathbf{0}$
    Take minibatch $\mathbf{x}$ from training dataset with labels $\mathbf{1}$
    Concatenate $\mathbf{x}$ and $\mathbf{z}$ into $\mathbf{y}$
    Concatenate labels of $\mathbf{x}$ and $\mathbf{z}$ into $\mathbf{y}_{\text{labels}}$
    # Training Discriminator
    $\mathbf{d} = D(\mathbf{y})$
    Compute BCE loss between $\mathbf{d}$ and $\mathbf{y}_{\text{labels}}$
    Optimize the weights of the Discriminator accordingly ▷ Adam optimizer is often chosen
    # Training Generator
    Sample $\mathbf{z} \sim p_{\text{prior}}$
    $\mathbf{g} = G(\mathbf{z})$
    $\mathbf{d} = D(\mathbf{g})$
    Compute BCE loss between $\mathbf{d}$ and $\mathbf{1}$
    Optimize the weights of the Generator accordingly ▷ Adam optimizer is often chosen
  end for
end for

In the above algorithm, the Binary-Cross Entropy (BCE) loss between $\mathbf{x} = (x_1, ..., x_n)$ and $\mathbf{y} = (y_1, ..., y_n)$ is defined as:

$$\text{BCE}(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{k=1}^{n} \left[y_k \log(x_k) + (1 - y_k) \log(1 - x_k) \right].$$

In cosmology, they have been used to model the complex large-scale structure (Rodriguez et al., 2018; Feder et al., 2020). Although powerful, GANs are prone to mode-collapse phenomena, even in the case of very simple distributions (Eghbal-zadeh et al., 2019). They are also difficult to train. One possible improvement is to consider more stable types of objective functions, for instance based on Wasserstein distances (Arjovsky et al., 2017). Overall, these various drawbacks led us to investigate other types of architectures.

Diffusion models

Diffusion models (Sohl-Dickstein et al., 2015) are state-of-the-art in terms of image generation Ho et al. (2020). They involve starting with a training sample image and gradually destroying it by adding noise. Conversely, once trained, the model starts with any white Gaussian noise and maps it to an image. These techniques are based on fundamental principles from thermodynamics and more particularly on two key observations about diffusion processes (Langevin, 1908):

1. they destroy information;
2. they are reversible at the microscopic scale.

These models are technically based on the use of Markov chains, which must be able to be inverted, to add noise to a signal. More specifically, diffusion models rely on a forward trajectory, which adds noise to the data, and a reverse trajectory, which is trained to reconstruct data from noise. The architecture is summarized in Figure 1.10 Let us call $\pi(\mathbf{x}_T)$ the target data distribution and π_0 the noise distribution. The forward process consists in applying repeatedly a Markov diffusion kernel. The probability distribution of the forward process reads:

$$\pi_f(\mathbf{x}_{(T...0)}) = \pi(\mathbf{x}_T) \prod_{t=1}^{T} g_t(\mathbf{x}_{t-1} | \mathbf{x}_t),$$

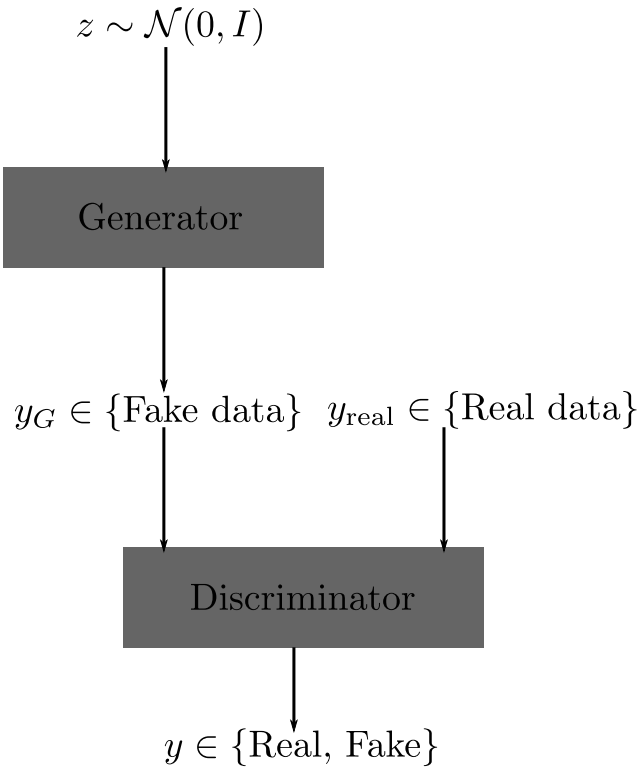


Figure 1.9: Generative Adversarial Networks architecture.

with g_t the density of a Gaussian. The backward process is trained to match the reversed-forward dynamics, i.e. it is required to respect the two following conditions:

$$\pi_b(\mathbf{x}_0) = \pi_0(\mathbf{x}_0)$$
$$\pi_b(\mathbf{x}_{0\ldots T}) = \pi_b(\mathbf{x}_0) \prod_{t=1}^{T} h_t(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

A classical result (Feller, 1949) ensures that the h_t are also Gaussian distributions, for which only mean and variance need be estimated. The training phase simply amounts to maximizing the model log-likelihood (Sohl-Dickstein et al., 2015). As a result, the generative process itself is costly, since this Markov chain must be inverted to produce a sample. It is this non-negligible numerical cost which, even if it can be partially amortized (Song et al., 2022), may lead us to consider other types of models. Another interesting aspects of diffusion models is that they rely on physical diffusion processes, underlying their interpretability properties. In astrophysics, they have been tested in Mudur and Finkbeiner (2022) for the modeling of the dark matter density field or in Cuesta-Lazaro and Mishra-Sharma (2024) for the modeling of galaxy distribution.

1.3.3 Normalizing Flows

In this thesis, we have been mostly interested in Normalizing Flows, a type of generative models that allow for robust sampling and inference by transporting a reference base distribution onto the target. As examples of applications in cosmology, NF models have been used for the modeling of the field-level likelihood of weak lensing (Dai and Seljak, 2022) or the inference of cosmological parameters from gravitational wave events (Stachurski et al., 2024). They are a robust alternative to Generative Adversarial Networks which, despite showing impressive results in image generation, are notoriously hard to train and may suffer from mode-collapse (Lin et al., 2018; Arjovsky and Bottou, 2017; Berard et al., 2020). Also, the numerical cost of diffusion models may be prohibitive in some cases.

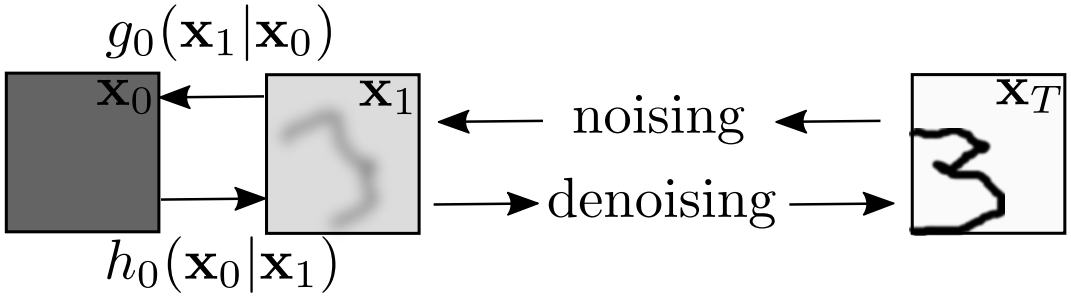


Figure 1.10: Diffusion model architecture.

Background

Normalizing flows (Tabak and Vanden-Eijnden, 2010; Dinh et al., 2014; Rezende and Mohamed, 2015) are generative models that map a complex target distribution π onto a known prior (also called *base*) distribution π_0 which is easy to sample (Papamakarios et al., 2022). In general, π_0 is a unimodal Normal distribution but expressivity requirements or a more precise knowledge of the target distribution may lead to other choices. The mapping is a series of smooth invertible transformations $\mathcal{T}_1, ..., \mathcal{T}_L$. Once the model is trained, one can reverse the learned dynamics to generate samples from the target distribution starting from the prior. If $X = \mathcal{T}_L \circ ... \circ \mathcal{T}_1(Z)$, where $Z \sim \pi_0$, then according to the change of variable formula the density followed by X, i.e. the model density, reads

$$m(x) = \pi_0 \left(\mathcal{T}_1^{-1} \circ ... \circ \mathcal{T}_L^{-1}(x)\right) \times \prod_{k=1}^{L} \left|\det J_{\mathcal{T}_k^{-1}}(x)\right|.$$

The model parameters to optimize are denoted Θ. In most unsupervised settings (Papamakarios et al., 2022), the goal is to minimize the Kullback-Leibler divergence between the target distribution π and the model distribution m with respect to Θ, i.e. minimizing:

$$\begin{aligned}\mathcal{L}(\Theta) &= \mathbf{E}_\pi \left[\log \pi(X) - \log m(X;\Theta)\right] \\ &= -\mathbf{E}_\pi \left[\log \pi_0 \left(\mathcal{T}_1^{-1} \circ ... \circ \mathcal{T}_T^{-1}(X;\Theta)\right) + \sum_{k=1}^{T} \left|\det J_{\mathcal{T}_k^{-1}}(X)\right|\right] + C.\end{aligned}$$

One can use samples from the target distribution in order to get a Monte Carlo estimation of the above loss under the form

$$\mathcal{L}(\Theta) \approx \frac{-1}{N} \sum_{i=1}^{N} \left[\log \pi_0 \left(\mathcal{T}_1^{-1} \circ ... \circ \mathcal{T}_L^{-1}(x_i;\Theta)\right) + \sum_{k=1}^{L} \left|\det J_{\mathcal{T}_k^{-1}}(x_i;\Theta)\right|\right]$$

and minimize it with a gradient descent algorithm (Kingma and Ba, 2015) through backpropagation of error (Rumelhart et al., 1986).

At this point, transformations are to some extent arbitrary. The choice is guided by multiple considerations:

1. The transformations must be *invertible* and *smooth* enough. This is because one requires performing a change of variable, which is in general valid for invertible functions with the additional property of being at least $\mathcal{C}^1$-diffeomorphisms (Rudin, 1987).

2. They also need to be *computationnally not costly*, the most demanding part corresponding to the computation of the Jacobian determinant of the mapping. In practice, one needs to choose transformations whose Jacobian matrix has good properties - diagonal, triangular -, as it is the case with models such as Real NVPs (Dinh et al., 2017).

3. The chain of transformations needs be *expressive* in order to map the (Gaussian) prior onto any target distribution. Ideally, one would want to have theoretical results of the form *in the limit of an infinite number of model parameters and training capacity, the KL-divergence of the model tends to 0.*

4. Finally, one would ideally want the learnt transformation to be *interpretable.* This property, not much studied in the Normalizing Flows literature, is a crucial aspect of Artificial Intelligence, as understanding the decision made by models is important for many fields from physics to medical science.

Some classical flow-based models

Let us present two popular types of Normalizing Flows, namely Non-linear Component Estimation (NICE) (Dinh et al., 2014) and Real-valued non-volume preserving transformations (Real NVP) Dinh et al. (2017). Each of them is based on the use of coupling layers for designing an expressive yet easy to invert model. The purpose of general coupling layers is to design bijective transformations with triangular Jacobian matrix whose determinant is easy to compute (Dinh et al., 2014) since it is the product of diagonal terms. Let us call $x \in \mathbb{R}^D$ an input vector. Let $I_{I_1} = \{1, ..., d\}$ and $I_2 = \{d+1, ..., D\}$ be a partition of $\{1, ..., D\}$. Let $f : \mathbb{R}^d \to \mathcal{Y}$ any map and $g : \mathbb{R}^{D-d} \times f(\mathbb{R}^d) \to \mathbb{R}^{D-d}$ an invertible function with respect to its first component, for all fixed second component. Now one can define the output vector $y = (y_{I_1}, y_{I_2})$ by fixing its first d terms to the same value as the first d terms of the input vector, and the last $D - d$ terms by a transformation depending on the whole input vector as:

$$\begin{cases} y_{I_1} &= x_{I_1} \\ y_{I_2} &= g(x_{I_2}, f(x_{I_1})) \end{cases}$$

Also, inverting the mapping is straightforward since:

$$\begin{cases} x_{I_1} &= y_{I_1} \\ x_{I_2} &= g^{-1}(y_{I_2}, f(y_{I_1})) \end{cases}$$

One can check that the Jacobian matrix associated to this transformation is indeed triangular which allows to compute its determinant easily:

$$J = \begin{pmatrix} I_d & 0 \\ \frac{\partial y_{I_2}}{\partial x_{I_1}} & \frac{\partial y_{I_1}}{\partial x_{I_2}} \end{pmatrix}$$

In practice, multiple coupling layers are chained up for more expressivity. Now, let us investigate the choice of coupling layers for the two models cited above:

- The NICE model is based on the use of *affine* coupling layers. The transformation is described as follows, where f is a deep neural network with ReLU activation:

$$\begin{cases} y_{I_1} &= x_{I_1} \\ y_{I_2} &= x_{I_2} + f(x_{I_1}) \end{cases}$$

  Its inverse is given by:

$$\begin{cases} x_{I_1} &= y_{I_1} \\ x_{I_2} &= y_{I_2} - f(x_{I_1}) \end{cases}$$

  As a consequence, the model has a unit Jacobian determinant, i.e. it is volume-preserving.

- As for the Real NVP model, its coupling transformation writes:

$$\begin{cases} y_{I_1} &= x_{I_1} \\ y_{I_2} &= x_{I_2} \cdot \exp(s(x_{I_1})) + t(x_{I_1}) \end{cases}$$

  where $s, t : \mathbb{R}^d \to \mathbb{R}^{D-d}$ are two functions referred to as the *scale* and *translation*, respectively. The inverse is given by:

$$\begin{cases} x_{I_1} &= y_{I_1} \\ x_{I_2} &= (y_{I_2} - t(y_{I_1})) \cdot \exp(-s(y_{I_1})) \end{cases}$$

  This time, the Jacobian of the transformation is slightly more complex:

$$J = \begin{pmatrix} I_d & 0 \\ \frac{\partial y_{I_2}}{\partial x_{I_1}} & \operatorname{diag}(s(x_{I_1})) \end{pmatrix}$$

  leading to $\det(J) = \prod_{k=1}^{D-d} \operatorname{diag}(s(x_{I_1}))_k$, which is not constant in general.

Building a flow-based model with the four properties mentioned below is hard in practice and the choice should be guided by the user's needs. For instance, none of the above architecture is highly-interpretable, in the sense that the learned mappings are a priori uninformative about the target distribution that the model is trying to fit. Here, the difference in the coupling transformations has a more fundamental impact on the overall expressivity of the model. Experiments suggest that the Real NVP model is by far more expressive than the NICE version. This is because the latter is volume-preserving, thus limiting its expressivity power. Indeed, it has been show that volume-preserving flows are prone to highly-limiting issues such as the incapacity to model multimodal distribution when starting from a unimodal base distribution. A model relying on volume-preserving mapping is thus subject to an incompressible bias regarding the sampling of complex multimodal distributions (Draxler et al., 2024). One way to mitigate this problem is to consider multimodal base distribution, which is only helpful when one has access to the number of modes of the target.

Conclusion

In this first chapter, we introduced the main mathematical tools as well as the sampling algorithms that are used in our contributions. We focused on two kinds of sampling techniques. First, MCMC methods, with emphasis on non-reversible methods as a way to boost the exploration of the parameters space for complex high-dimensional distributions. And finally, generative machine learning models, which have shown great promises for the modeling of complex probability distributions, provided that large amount of training data are available. We specifically focused on flow-based approaches as a robust and numerically efficient alternative to traditional models such as GANs and diffusion models.

Chapter 2

The cosmological framework and BORG algorithm

> *The library will endure; it is the universe. As for us, everything has not been written; we are not turning into phantoms. We walk the corridors, searching the shelves and rearranging them, looking for lines of meaning amid leagues of cacophony and incoherence, reading the history of the past and our future, collecting our thoughts and collecting the thoughts of others, and every so often glimpsing mirrors, in which we may recognize creatures of the information.*
>
> Jorge Luis Borges, *The Library of Babel*

Cosmology offers a great playground for developing sampling algorithms as it deals with complex physical models and high-dimensional spaces. The probabilistic characterization of the highly non-trivial large-scale structure of the universe is thus a challenging problem and a nice example to compare the performance of different kinds of samplers. This chapter deals with an introduction to the Big Bang theory and then details a cosmological forward-model of gravitation as a way to map the primordial fluctuations to the present universe. It ends with an introduction to the BORG machinery which is based on reversible Monte Carlo samplers to infer the primordial fluctuations from astronomical data.

2.1 The Big Bang Theory

This section deals with an introduction to the standard model of cosmology. We also describe the large-scale structure along with some statistical techniques to model it.

2.1.1 The Cosmic Microwave Background and early universe

In 1929, Edwin Hubble concluded after a series of observations that there was a relation between the distance and the radial velocity of out-of-the-galaxy nebulæ(Hubble, 1929). The so-called *Hubble law* is a solid confirmation of earlier theoretical work by Friedmann and Lemaître who solved the equations of General relativity in the case of an expanding isotropic and homogeneous universe, thus explaining this relation (Friedmann, 1922; Lemaître, 1927).

Going backward in time, the expansion of the universe means that at some point, the universe as we know it was much denser and hotter, eventually leading to a singularity. This event, the beginning of ages, is called the Big Bang. In such cosmogony, the early universe was a hot dense soup of matter in which photons were constantly emitted and absorbed. The universe remained opaque until it

became less dense and less hot, making it possible for stable atomic structures to form and photons to circulate, an era named recombination. The fossil trace of this early signal is still visible, but within a low-frequency part of the spectrum because of the spatial expansion causing a redshift distortion.

Historically, the so-called Cosmic Microwave Background (CMB) was predicted in 1948 (Alpher and Herman, 1948) and accidentally observed for the first time in 1964 (Penzias and Wilson, 1965). The CMB corresponds to a black-body system with a temperature of about 2.7 K (Fixsen, 2009; Planck Collaboration et al., 2016) and it played a great role in the comprehension of the cosmological history, as a striking evidence in favor of the Big Bang theory. It also accounts for a precise description of the early universe, whose characteristics are imprinted withing the last-scattering surface and manifest itself in the present universe as an homogeneous isotropic signal in the microwave part of the electromagnetic spectrum, see Figure 2.1. It is widely admitted that the density fluctuations that we observe come from the post-inflation era, whose field can be described by a Gaussian distribution with great accuracy, as the result of a combination of many independent quantum fields.

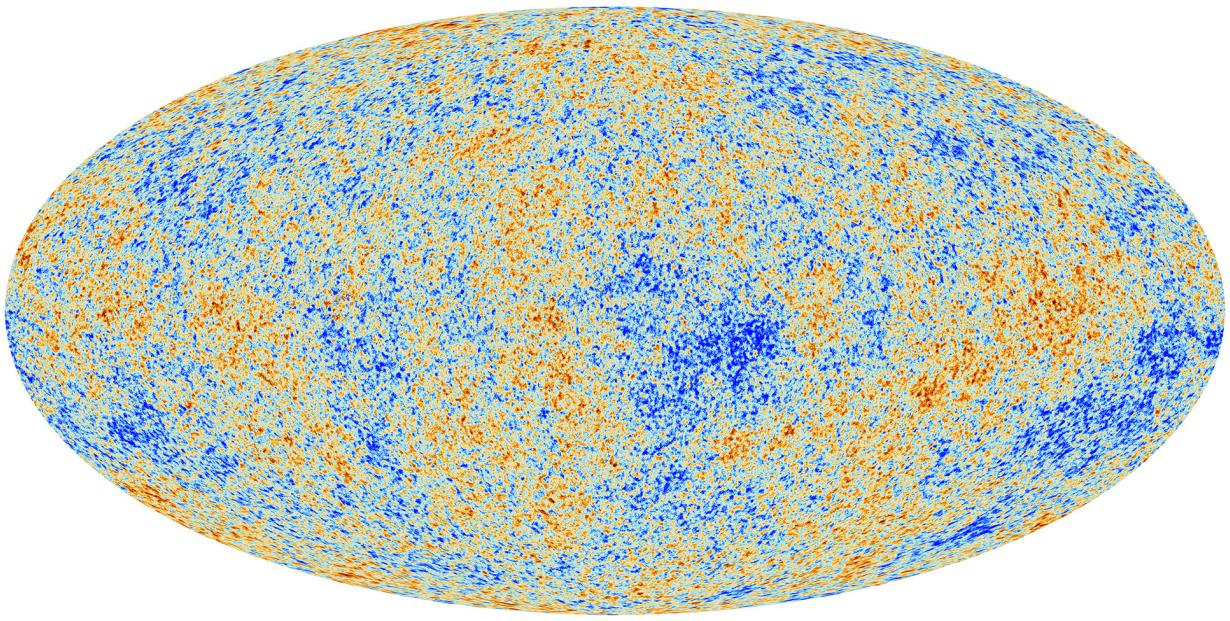


Figure 2.1: The Cosmic Microwave Background reconstructed from observations made with European Space Agency telescope Planck. The difference in colors correspond to very small inhomogeneities in the signal of order 10^{-5} K.

An accurate knowledge of the early universe, whose characteristics correspond to the CMB, allows us to design a well-supported prior knowledge for the Bayesian problem that we are trying to solve, i.e. the probabilistic characterization of the initial conditions given astronomical data. Then, thanks to a forward-model of gravitation, it will be possible to get the characterization of today's universe.

2.1.2 The Λ-CDM model

In a very general setting, General Relativity claims that the dynamics of the Universe is governed by Einstein's equation (Einstein, 1916) which gives a way to determine the metrics of space-time from its energetic content. It can be written as:

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}, \tag{2.1}$$

In the above,

- $R_{\mu\nu}$ is the Ricci's tensor;

- R is the scalar curvature of space-time;
- $g_{\mu\nu}$ is the metric tensor;
- Λ is the cosmological constant;
- G is the Newtonian constant of gravitation;
- c is the speed of light;
- $T_{\mu\nu}$ is the stress-energy tensor.

Let us start with the description of the movement of a very large number of particles with mass m in gravitational interaction. The movement of one particle with position $\mathbf{r}$ and speed $\mathbf{v}$ is given by:

$$\frac{d\mathbf{v}}{dt} = -\frac{\partial \phi}{\partial \mathbf{r}}, \tag{2.2}$$

with ϕ the gravitational potential induced by the local density $\rho(\mathbf{r})$.

In the context of an expanding universe, we have to consider the comoving coordinates $\mathbf{x}$ which are related to the physical coordinates by $\mathbf{r} = a(\tau)\mathbf{x}$, with a the cosmological scale factor and τ the conformal time given by $d\tau = dt/a(\tau)$. Let us define the conformal expansion rate as $\mathcal{H} = a(\tau)H$, H being the Hubble constant. The Λ-CDM model, also called the standard model of cosmology because it supports observations of the sky with great accuracy, is based on a Friedmann–Lemaître–Robertson–Walker metric which describes a homogeneous isotropic and expanding universe (Friedmann, 1922, 1924; Lemaître, 1927; Robertson, 1935; Walker, 1937). These hypotheses lead to the following equations of motion, also known as Friedman-Lemaître's equations (Friedmann, 1922; Lemaître, 1927), that are a re-expression of Einstein's equation in a homogeneous and isotropic Universe solely made of matter and with a cosmological constant:

$$\frac{\partial \mathcal{H}(\tau)}{\partial \tau} = \left(\Omega_\Lambda(\tau) - \frac{\Omega_m(\tau)}{2}\right)\mathcal{H}(\tau)^2, \tag{2.3}$$

$$\left(\Omega_{tot}(\tau) - 1\right)\mathcal{H}(\tau)^2 = k. \tag{2.4}$$

In the above, k is the curvature of the universe. This set of equations can be used to determine the content of the universe in a Λ-CDM model. Indeed, from the above equations, it is possible to derive the present-day critical density ρ_c which gives a null curvature assuming $\Lambda = 0$. Denoting with a subscript x a component of the universe (matter, baryons, radiation, dark energy), one defines a density parameter as (Frieman et al., 2008):

$$\Omega_x := \frac{\rho_x(\text{present time})}{\rho_c}.$$

For the usual Λ-CDM model, Friedmann's equations yield:

$$\frac{\dot{a}}{a} = H\sqrt{\Omega_m a^{-3} + \Omega_{\text{rad}} a^{-4} + \Omega_\Lambda}.$$

In conclusion, the Λ-CDM model is based on the following key ingredients:

- the existence of a constant Λ which accounts for the acceleration of the universe expansion;
- the presence of cold dark matter, which amounts to 27% of the universe content, and ordinary matter, which accounts for 5% of its content. The remaining is made of 68% of dark energy that drives the universe expansion.

The Λ-CDM model is built upon the hypothesis that the theory of General relativity is the correct description of the gravitational force. It is possible to characterize it by means of six independent cosmological parameters, governing the evolution of the cosmic history. They are detailed in the table below and their value was determined by the Planck collaboration (Planck Collaboration et al., 2020). Note that in the latter are reported the values of $\omega_x := \Omega_x h^2$, where h is the normalized Hubble's constant.

Symbol	Description	Value
ω_b	Baryon density	0.02242 ± 0.00014
ω_c	Cold dark matter density	0.11933 ± 0.00091
t_0	Age of the universe	$(13.787 \pm 0.020) \times 10^9$ years
τ	Reionization optical depth	0.0561 ± 0.0071
$\ln(10^{10} A_s)$	Initial super-horizon amplitude of curvature perturbations	3.047 ± 0.014
n_s	Primordial spectral index	0.9665 ± 0.0038

The six parameters can be used to determine the value of other parameters of interest. The determination of cosmological parameters from astronomical data is another important aspect of cosmological inference as the knowledge of the parameters controlling the universe dynamics is another way of making theoretical predictions about its structure and evolution. Such task can be carried out using probabilistic sampling algorithms. A classical example is the inference of Ω_m (matter density) and h (expansion velocity) from type Ia supernovae observations using MCMC techniques (Lewis and Bridle, 2002).

2.1.3 The present large-scale structure

At very large scale, of order greater than 100 Mpc, we often make the two following assumptions about the universe, also referred to as cosmological principles:

- it is **isotropic**, meaning that there is no preferred direction;
- it is **homogeneous**, meaning that there is no preferred position.

More precisely, this is an asymptotic condition since there is no strict homogeneity, even at very large-scale. Indeed, despite these two simplifying assumptions, the density field of today's universe cannot be described by simple statistics. Its complexity results from the gravitational amplification of the primordial Gaussian anisotropies imprinted within the Cosmic Microwave Background. The resulting large-scale structure consists of long filaments of matter and dark matter that range across hundreds of millions light-years, a structure called the cosmic web (Bond et al., 1996). At the intersection of these filaments, so at the highest density regions, lie galaxy clusters. Figure 2.2 illustrates the distribution of galaxies from real astronomical data by the Sloan Digital Sky Survey[1] (York et al., 2000). Another illustration made by a computer simulation is presented in Figure 2.3.

The distribution of galaxies actually fits the underlying density field with great accuracy. We refer to them as *matter tracers*. Characterizing the complex large-scale structure from galaxy observations is a challenging Bayesian problem. One possible approach is to make hypotheses regarding the nature of the posterior probability distribution that best describes the highly non-linear density field. First attempts with large datasets employed Wiener filters that showed promising results for reconstructing the full three-dimensional evolved matter field (Lahav et al., 1994; Zaroubi et al., 1995; van de Weygaert and Bertschinger, 1996; Kitaura and Enßlin, 2008; Kitaura et al., 2009). However, this technique relies on the strong assumption that the posterior is a Gaussian, which is valid at large scale but lacks precision when considering non-linear structures at smaller scales. An improvement consists in

[1] https://www.sdss.org/

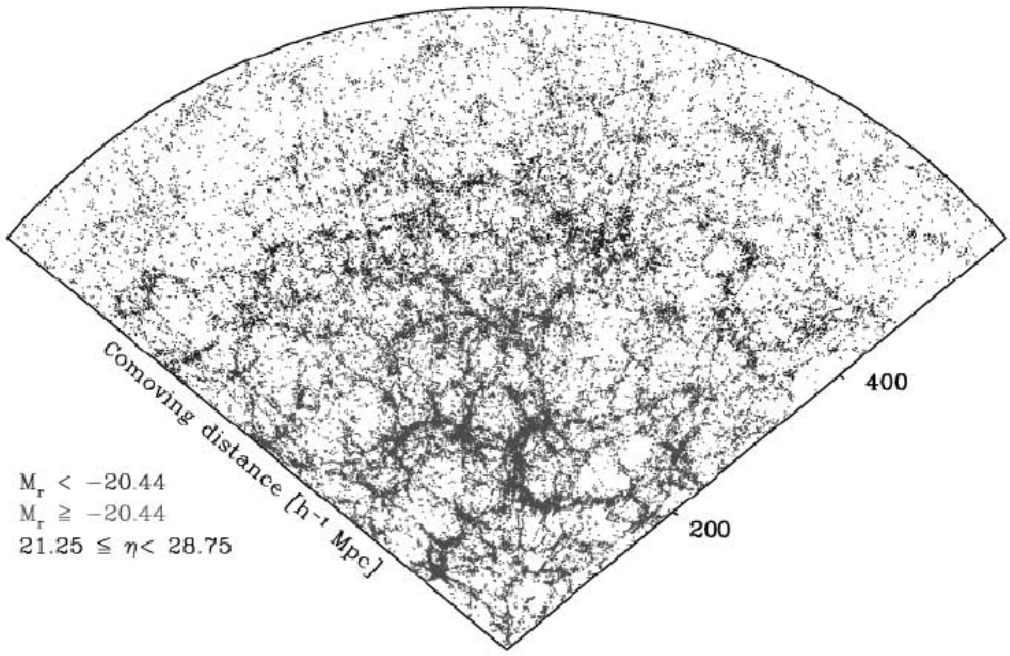


Figure 2.2: Distribution of galaxies within a slice from the Sloan Digital Sky Survey data, image from Park et al. (2005). The radial coordinate is the comoving distance, the angular variable is the instrument longitude, the bottom corner corresponds to our galaxy.

using the log-normal family which has been found to be quite in agreement with galaxy distributions (Hubble, 1934; Peebles, 1980; Kayo et al., 2001). When combining a log-normal prior with a Poissonian likelihood, one gets a log-normal posterior that allows to describes the non-linear density field with great accuracy (Jasche and Kitaura, 2010).

However, these methodologies are based on a prior knowledge that is not straightforward to justify. On the other hand, one can safely rely on a strong prior knowledge regarding the early universe. Recently, new lines of research have been explored (Jasche and Wandelt, 2013; Jasche and Lavaux, 2019). They re-cast the problem of inferring the present density field into an initial conditions Bayesian problem. This approach will be mathematically derived in Section 2.3.2. It requires a forward model of gravitation linking the early and the present universe.

2.2 Cosmological forward-models

The initial conditions of the universe are linked to that of the present universe through a deterministic map corresponding to a forward model of gravitation. The purpose of this section is to present such models.

2.2.1 Basics of cosmology and Vlasov equation

In the present section, we are interested in a model for the dynamics of perturbations from a Friedmann-Lemaître-Robertson-Walker universe. Let us define the following quantities:

- *The density contrast* $\delta(\mathbf{x})$ by ($\overline{\rho}$:= mean density)

$$\rho(\mathbf{x}, \tau) = \overline{\rho}(\tau)(1 + \delta(\mathbf{x})); \tag{2.5}$$

- *The peculiar velocity* $\mathbf{u}(\mathbf{x}, \tau)$ by:

$$\mathbf{v}(\mathbf{x}, \tau) = \mathcal{H}(\tau)\mathbf{x} + \mathbf{u}(\mathbf{x}, \tau); \tag{2.6}$$

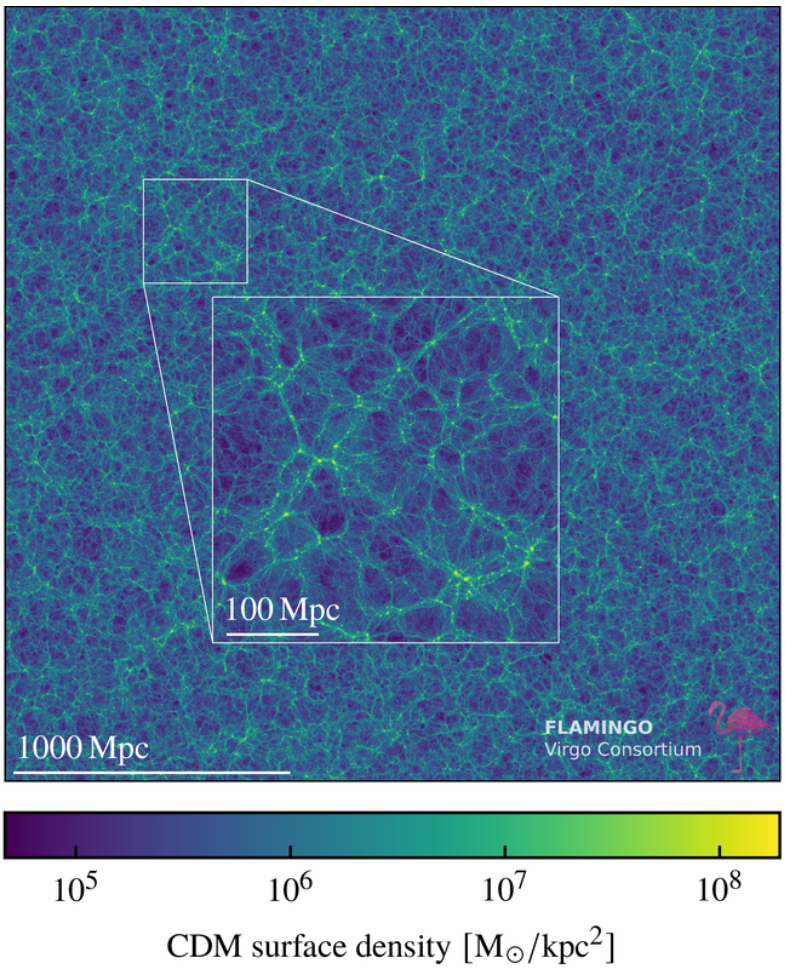


Figure 2.3: The CDM surface density map at redshift $z = 0$ of a 20 Mpc-wide slice with side 2.8 Gpc. The image is the result of a hydrodynamical simulation for the large-scale structure as part of the FLAMINGO project (Schaye et al., 2023).

- *The cosmological gravitational potential* $\Phi(\mathbf{x}, \tau)$ by:

$$\phi(\mathbf{x}, \tau) = -\frac{1}{2}\frac{\partial \mathcal{H}(\tau)}{\partial \tau}\mathbf{x}^2 + \Phi(\mathbf{x}, \tau). \tag{2.7}$$

The consequence of injecting these coordinate changes in the perturbative form inside Einstein's equation yields Poisson's equation. The latter implies that the cosmological potential is only sourced by density fluctuations as:

$$\nabla^2\Phi(\mathbf{x}, \tau) = \frac{3}{2}\Omega_m(\tau)\mathcal{H}(\tau)^2\delta(\mathbf{x}, \tau). \tag{2.8}$$

By introducing the momentum $\mathbf{p} = am\mathbf{u}$, one gets a new equation of motion:

$$\frac{d\mathbf{p}}{d\tau} = -am\nabla\Phi. \tag{2.9}$$

Now, introducing the particle number density in phase-space function $b = b(\mathbf{x}, \mathbf{p}, \tau)$, and invoking the conservation of this quantity through time, we obtain the Vlasov equation:

$$\frac{d\mathbf{p}}{d\tau} = \frac{\partial b}{\partial \tau} = \frac{\mathbf{p}}{am}.\nabla b - am\nabla\Phi.\frac{\partial b}{\partial \mathbf{p}} = 0. \tag{2.10}$$

This partial differential equation is hard to solve as it is highly non-linear (because of Φ) and b depends on seven variables (3 for $\mathbf{x}$, 3 for $\mathbf{p}$, 1 for τ).

2.2.2 Eulerian Perturbation Theory

We would like to express the evolution of the density field from Vlasov equation. This can be done in a Eulerian framework. In general, we prefer solving the evolution of the spatial distribution rather than finding solutions in the full phase-space (Bernardeau et al., 2002). This can be done easily by taking the first momentum moments of the distribution function f.

- 0^{th} order: $\int b(\mathbf{x}, \mathbf{p}, \tau)d^3\mathbf{p} = \rho(\mathbf{x}, \tau)$;
- 1^{st} order: $\int \frac{p}{am} b(\mathbf{x}, \mathbf{p}, \tau)d^3\mathbf{p} = \rho(\mathbf{x}, \tau)\mathbf{u}(\mathbf{x}, \tau)$;
- 2^{nd} order: $\int \frac{p_i p_j}{a^2 m^2} b(\mathbf{x}, \mathbf{p}, \tau)d^3\mathbf{p} = \rho(\mathbf{x}, \tau)u_i(\mathbf{x}, \tau)u_j(\mathbf{x}, \tau) + \sigma_{ij}(\mathbf{x}, \tau)$.

The equations for $\mathbf{u}$ and (σ_{ij}) (the stress tensor) are given via taking the moments of the Vlasov equation (2.10). The 0^{th} moment describes the conservation of mass:

$$\frac{\partial \delta(\mathbf{x}, \tau)}{\partial \tau} + \nabla.\left((1 + \delta(\mathbf{x}, \tau))\mathbf{u}(\mathbf{x}, \tau)\right) = 0. \tag{2.11}$$

The first moment describes the conservation of momentum:

$$\frac{\partial \mathbf{u}(\mathbf{x}, \tau)}{\partial \tau} + \mathcal{H}(\tau)\mathbf{u}(x, \tau) + \mathbf{u}(x, \tau).\nabla\mathbf{u}(x, \tau) = -\nabla\Phi(\mathbf{x}, \tau) - \frac{1}{\rho}\nabla_j(\rho\sigma_{ij}). \tag{2.12}$$

In linear regime, these equations become:

$$\frac{\partial \delta(\mathbf{x}, \tau)}{\partial \tau} + \theta(\mathbf{x}, \tau) = 0, \tag{2.13}$$

where $\theta := \nabla.\mathbf{u}(\mathbf{x}, \tau)$, and

$$\frac{\partial \mathbf{u}(\mathbf{x}, \tau)}{\partial \tau} + \mathcal{H}(\tau)\mathbf{u}(x, \tau) = -\nabla\Phi(\mathbf{x}, \tau). \tag{2.14}$$

This latter equation will be used in the following. The velocity field can be fully described by its divergence θ and its vorticity $\mathbf{w} := \nabla \times \mathbf{u}$. Taking the divergence of Equation (2.14) and using the Poisson equation (2.8) leads to:

$$\frac{\partial \theta(\mathbf{x},\tau)}{\partial \tau} + \mathcal{H}(\tau)\theta(\mathbf{x},\tau) + \frac{3}{2}\Omega_m(\tau)\mathcal{H}(\tau)^2\delta(\mathbf{x},\tau) = 0, \tag{2.15}$$

and now taking the curl of Equation (2.14) leads to:

$$\frac{\partial \mathbf{w}(\mathbf{x},\tau)}{\partial \tau} + \mathcal{H}(\tau)\mathbf{w}(\mathbf{x},\tau) = 0. \tag{2.16}$$

Equation (2.16) describes the evolution of the vorticity field through time. To get the evolution equation for the density field, one can use the derivative of Equation (2.15) and replace it in Equation (2.13).

2.2.3 Lagrangian Perturbation Theory

One can re-write the equations obtained previously using an other framework: the Lagrangian point of view (Moutarde et al., 1991; Bouchet et al., 1995; Bernardeau et al., 2002). It consists in following the trajectory of particles instead of studying the dynamics of the density and the velocity fields. This is the technique used in (Jasche and Wandelt, 2013) to explicit the final density contrasts G_k. Particles are placed on the grid and their position is updated according to the Perturbation theory model. We need to place ourselves in the Lagrangian framework so that each particle can be followed individually.

In this new framework, the main object of interest is the mapping $\mathbf{\Psi} = (\Psi_1, \Psi_2, \Psi_3)$ which maps the initial particle Lagrangian position $\mathbf{q}$ into its final Eulerian position $\mathbf{x}$. It is defined by:

$$\mathbf{\Psi}(\mathbf{q},\tau) = \mathbf{x}(\tau) - \mathbf{q}. \tag{2.17}$$

Following Equation (2.14), we can write that:

$$\frac{d^2\mathbf{x}(\tau)}{d\tau^2} + \mathcal{H}(\tau)\frac{d\mathbf{x}}{d\tau} = -\nabla\Phi(\mathbf{x},\tau). \tag{2.18}$$

If we assume that the Lagrangian mass element is conserved, then $\overline{\rho}(1+\delta(\mathbf{x}))d^3\mathbf{x} = \overline{\rho}d^3\mathbf{q}$. Hence:

$$1 + \delta(\mathbf{x}) = \frac{1}{J(\mathbf{q},\tau)}, \tag{2.19}$$

where J is the Jacobian of the transformation between Eulerian and Lagrangian space. It immediately follows that $\delta(\mathbf{x}) = (1 - J(\mathbf{q},\tau))/J(\mathbf{q},\tau)$.
Now, let us take the divergence of Equation (2.18). Using the Poisson equation (2.8), one gets:

$$J(\mathbf{q},\tau)\nabla_{\mathbf{x}}.\left(\frac{\partial^2\mathbf{\Psi}(\mathbf{q},\tau)}{\partial\tau^2} + \mathcal{H}(\tau)\frac{\partial\mathbf{\Psi}(\mathbf{q},\tau)}{\partial\tau}\right) = \frac{3}{2}\Omega_m(\tau)\mathcal{H}(\tau)^2(1 - J(\mathbf{q},\tau)). \tag{2.20}$$

This equation is difficult to solve because the gradient is taken with respect to the Eulerian variable $\mathbf{x}$ which depends on $\mathbf{q}$ (see (Leclercq, 2015)). Let us now place ourselves in the famous Zel'dovich approximation framework (Zel'dovich, 1970). The key idea is to consider the linear solution of Equation (2.20) as the displacement field. At linear order, one can write $J(\mathbf{q},\tau)\nabla_{\mathbf{x}} = \nabla_{\mathbf{q}}$ and $J(\mathbf{q},\tau) = 1 + \nabla_{\mathbf{q}}.\mathbf{\Psi}(\mathbf{q},\tau)$.

This yields another equation:

$$\nabla_{\mathbf{q}}.\left(\frac{\partial^2\mathbf{\Psi}(\mathbf{q},\tau)}{\partial\tau^2} + \mathcal{H}(\tau)\frac{\partial\mathbf{\Psi}(\mathbf{q},\tau)}{\partial\tau}\right) = \frac{3}{2}\Omega_m(\tau)\mathcal{H}(\tau)^2\nabla_{\mathbf{q}}.\mathbf{\Psi}(\mathbf{q},\tau). \tag{2.21}$$

Now, let us introduce the quantity $\psi = \nabla_{\mathbf{q}}.\mathbf{\Psi}$. Equation (2.21) becomes:

$$\psi'' + \mathcal{H}(\tau)\psi' - \frac{3}{2}\Omega_m(\tau)\mathcal{H}(\tau)^2\psi = 0, \tag{2.22}$$

which is a second-order differential equation. Its solution ψ^1 can be expressed as:

$$\psi^1(\mathbf{x},\tau) = \nabla_{\mathbf{q}}.\mathbf{\Psi}^1(\mathbf{x},\tau) = -D_1(\tau)\delta(\mathbf{q}), \tag{2.23}$$

where $\delta(\mathbf{q})$ is the density field imposed by the initial conditions and $\mathbf{\Psi}^1$ is the linear solution of Equation (2.20) which is identified with the displacement field.

It is convenient to define the Lagrangian potential ϕ^1 from which $\mathbf{\Psi}^1$ derives as:

$$\mathbf{\Psi}^1(\mathbf{q},\tau) = -D_1(\tau)\nabla_{\mathbf{q}}\phi^1(\mathbf{q}), \tag{2.24}$$

which satisfies the Poisson equation:

$$\nabla^2_{\mathbf{q}}\phi^1(\mathbf{q}) = \delta(\mathbf{q}). \tag{2.25}$$

Hence the final important relation between Eulerian and Lagrangian coordinates:

$$\mathbf{x}(\tau) = \mathbf{q} - D_1(\tau)\nabla_q\phi^1(\mathbf{q}). \tag{2.26}$$

This theory gives a way to find the final distribution of particles evolving through a gravitational potential. The final distribution of the particles on the grid will be used to determine the final density contrasts in each cell. Of course, we need meaningful mathematical tools to go back and forth between the continuous density contrast and the discrete contrasts on the grid.

2.3 The BORG algorithm

2.3.1 Quantities defined on a grid

As it was said before, the purpose of introducing Lagrangian Perturbation Theory is to compute the values of the final density contrasts on the grid given an initial realization of the density contrasts. One main difficulty is to go back and forth between continuous quantities and quantities defined on a grid. Given an initial realization of the density contrasts $\{\delta^i_j\}$ on the grid, the idea is to place artificial cold-dark matter particles on the grid and let them evolve according to LPT.

Let's consider a cubic box with volume $V = L^3$. The number of cells on the grid equals N and Δx stands for the side length of a cubic cell. We place N_p particles in the box. The goal is to determine the continuous density field from the discrete distribution of such particles.

The process relies on 2 steps: A **mesh assignment** scheme which is a way to assign to the grid a quantity carried by the particles; and an **interpolation scheme** which seeks to distribute to the particles a quantity defined on the grid (Hockney and Eastwood, 1988).

- **Mesh-assignment:** Each particle is assumed to have a "shape" represented by a function S. In the 1D case, the fraction of the particle at position x_p assigned to the cell with center x_c is the average over this cell of the shape function:

$$W(x_p - x_c) = \int_{x_c-\Delta x/2}^{x_c+\Delta x/2} S(x' - x_p)dx'. \tag{2.27}$$

  In the 3D case, the expression is rewritten as follows (we keep the same notation for W):

$$W_3(\mathbf{x}_p - \mathbf{x}_c) = W(x_p - x_c) \times W(y_p - y_c) \times W(z_p - z_c). \tag{2.28}$$

For any quantity Q, if Q_p $(1 \leq p \leq N_p)$ is the quantity carried by the particle at position x_p, then the following quantity is assigned to each cell:

$$Q_c = Q(\mathbf{x}_c) = \sum_{p=1}^{N_p} Q_p W_3(\mathbf{x}_p - \mathbf{x}_c), \qquad \forall\ 1 \leq c \leq N. \tag{2.29}$$

In the present case, each particle carries the same mass m. The density in each cell can be computed using:

$$\rho(\mathbf{x}_c) = \frac{m}{(\Delta x)^3} \sum_{p=1}^{N_p} W_3(\mathbf{x}_p - \mathbf{x}_c). \tag{2.30}$$

From the definition $\delta := \frac{\rho}{\bar{\rho}} - 1$, we reconstruct the density contrast in each cell from the density:

$$\delta(\mathbf{x}_c) = \left(\frac{N}{N_p} \sum_{p=1}^{N_p} W_3(\mathbf{x}_p - \mathbf{x}_c) \right) - 1. \tag{2.31}$$

- **Interpolation:** The *"conjugate problem"* called interpolation relies on a similar process except this time the summation is taken over the cells. For any quantity Q, if Q_c $(1 \leq c \leq N_g)$ is the quantity in the cell with centre $\mathbf{x}_c$, then quantity assigned to the particle with position $\mathbf{x}_p$ is:

$$Q_p = Q(\mathbf{x}_p) = \sum_{c=1}^{N} Q_c W_3(\mathbf{x}_p - \mathbf{x}_c), \qquad \forall\ 1 \leq c \leq N_p. \tag{2.32}$$

Now, one must decide which kernel W to choose for the procedure (Hockney and Eastwood, 1988; Leclercq, 2015). One popular option (Hockney and Eastwood, 1988; Jasche and Wandelt, 2013; Jasche and Lavaux, 2019) is the Cloud-In-Cell (CIC) kernel whose expression is:

$$W(x) = \begin{cases} 0 & \text{if } |x| > \Delta x \\ 1 - |x| & \text{else} \end{cases} \tag{2.33}$$

In 1D, the CIC kernel (Birdsall and Fuss, 1969) presents some nice properties that are conformal with the definition of a *good* kernel:

- it is easy to compute numerically;
- it preserves the mass on the grid;
- it is piecewise affine;
- it limits the error for long-range interactions (Hockney and Eastwood, 1988).

For computations, we consider that $W'(x) = \begin{cases} 0 & \text{if } |x| > \Delta x \text{ or } x = 0 \\ 1 & \text{if } -1 \leq x < 0 \\ -1 & \text{if } 0 < x \leq 1 \end{cases}$ and that $W'' = 0$.
These properties (except for the third and the derivatives computations) are conserved when considering the 3D extension of this kernel:

$$W_3(x, y) = \begin{cases} 0 & \text{if } |x| > c \text{ or } |y| > c \text{ or } |z| > c \\ (1 - |x|) \times (1 - |y|) \times (1 - |z|) & \text{else} \end{cases} \tag{2.34}$$

2.3.2 The cosmological Bayesian problem

Let us consider a cubic box of length L (a few hundreds Mpc/h in practice) divided into d regular cubic cells. In each cell k, we denote N_k^g the number of galaxies, or other matter tracer, that is observed. Ideally, one would like to sample from the posterior probability of the present density fluctuations in each cell, δ_k, being correct given astromomical observations $\{N_k^g\}_{k=1}^d$, that is:

$$\pi\left(\{\delta_k\}_{k=1}^d|\{N_k^g\}_{k=1}^d\right) \underset{\text{Bayes}}{=} \frac{\pi\left(\{\delta_k\}_{k=1}^d\right) \times \pi\left(\{N_k^g\}_{k=1}^d|\{\delta_k\}_{k=1}^d\right)}{\pi\left(\{N_k^g\}_{k=1}^d\right)} \tag{2.35}$$

It is difficult however to express our a priori on the prior term for the present density fluctuations. But gravitational theory teaches us that the filamentary cosmic structure observed in the present LSS of the universe arises from an amplification of the initial density fluctuations in the primordial universe. Also, present and initial conditions are linked via a purely deterministic process, that is a cosmological model G. Finally, primordial fluctuations exhibit very simple statistical properties, as confirmed experimentally by Planck's observations of the cosmological microwave background (Planck Collaboration et al., 2016). Therefore the idea (Jasche and Wandelt, 2013; Jasche and Lavaux, 2019) to recast the problem (2.35) into an initial conditions inference problem. Let us consider the joint posterior of both initial and present density fluctuations, knowing data:

$$\pi\left(\{\delta_k^0\}_{k=1}^d, \{\delta_k\}_{k=1}^d|\{N_k^g\}_{k=1}^d\right) = \frac{\pi\left(\{\delta_k^0\}_{k=1}^d, \{\delta_k\}_{k=1}^d\right) \times \pi\left(\{N_k^g\}_{k=1}^d|\{\delta_k^0\}_{k=1}^d, \{\delta_k\}_{k=1}^d\right)}{\pi\left(\{N_k^g\}_{k=1}^d\right)} \tag{2.36}$$

Now, let us discuss each term on the right side of Equation (2.36):

- the prior term can be rewritten as $\pi\left(\{\delta_k^0\}\right) \times \pi\left(\{\delta_k\}_{k=1}^d|\{\delta_k^0\}_{k=1}^d\right)$ and as the process is deterministic, i.e. $\delta_j = G_j\left(\{\delta_k^0\}_{k=1}^d\right)$, this leads to:

$$\pi\left(\{\delta_k^0\}_{k=1}^d, \{\delta_k\}_{k=1}^d\right) = \pi\left(\{\delta_k^0\}\right) \times \prod_{j=1}^d \delta^D\left(\delta_j - G_j\left(\{\delta_k^0\}_{k=1}^d\right)\right)$$

- We also make the hypothesis that the likelihood term solely depends on the final conditions, i.e. $\pi\left(\{N_k^g\}_{k=1}^d|\{\delta_k^0\}_{k=1}^d, \{\delta_k\}_{k=1}^d\right) = \pi\left(\{N_k^g\}_{k=1}^d|\{\delta_k\}_{k=1}^d\right) = \pi\left(\{N_k^g\}_{k=1}^d|\{G_k\left(\{\delta_l^0\}_{l=1}^d\right)\}_{k=1}^d\right)$

Putting it all together, one gets:

$$\pi\left(\{\delta_k^0\}_{k=1}^d, \{\delta_k\}_{k=1}^d|\{N_k^g\}_{k=1}^d\right) = \frac{\pi\left(\{\delta_k^0\}\right) \times \prod_{j=1}^d \delta^D\left(\delta_j - G_j\left(\{\delta_k^0\}_{k=1}^d\right)\right) \times \pi\left(\{N_k^g\}_{k=1}^d|\{G_k\left(\{\delta_l^0\}_{l=1}^d\right)\}_{k=1}^d\right)}{\pi\left(\{N_k^g\}_{k=1}^d\right)} \tag{2.37}$$

Marginalizing over the final density fluctuations, one gets the desired posterior for LSS:

$$\pi\left(\{\delta_k^0\}_{k=1}^d|\{N_k^g\}_{k=1}^d\right) = \frac{\pi\left(\{\delta_k^0\}\right) \times \pi\left(\{N_k^g\}_{k=1}^d|\{G_k\left(\{\delta_l^0\}_{l=1}^d\right)\}_{k=1}^d\right)}{\pi\left(\{N_k^g\}_{k=1}^d\right)} \tag{2.38}$$

Multiple choices can be made for the likelihood. Gaussian or Poissonian distributions are typically chosen (Jasche and Wandelt, 2013). As for the prior term, it should be noted that the initial density fluctuations live in a latent space which corresponds to a post-inflation era. The primordial field obeys standard Gaussian statistics with a high level of precision, meaning for us that:

$$\pi\left(\{\delta_k^0\}\right) = \frac{\exp\left(-\frac{1}{2}||\{\delta_k^0\}||^2\right)}{2\pi}.$$

The forward model G which maps the initial inferred conditions onto the final fluctuations is thus made of multiple part :

- a *Primordial* function that gives the fluctuations of the gravity field created by the inflation era;
- a *Eisenstein-Hu* transformation that give the convolution kernel for linearly mapping the latent space fluctuations and bring them to a statistically correct regime for the beginning of the matter-dominated era just after the CMB. This is an approximation of the solutions from a set of Ordinary Differential Equations. The solver is named Einstein-Boltzmann (Lesgourgues, 2011);
- a gravitational forward-model, here Lagrangian Perturbation Theory (LPT), for mapping the post-CMB fluctuations onto the present day.

As a result, one may write:

$$G = \text{LPT} \circ \text{Eisenstein-Hu} \circ \text{Primordial}.$$

In the end, it comes to generating samples from a probability distribution Π on $\mathbb{R}^d$ whose probability density function is known up to a normalisation constant Z and can be written under the generic form:

$$\pi(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{Z}$$

This is a classical inference problem that can be solved numerically by running a MCMC algorithm.

2.3.3 Hamiltonian Monte Carlo for BORG

The BORG machinery relies on a HMC algorithm for sampling plausible realizations of the primordial universe from astronomical observations (Jasche and Wandelt, 2013; Jasche and Lavaux, 2019). In this work, we tested a special version of the sampler implemented withing the BORG framework called `HMCRealDensitySampler`. We describe here its main features.

Gradient evaluation

This version of HMC deals with samples in real space. However, for making computations easier, most of the numerical effort is made in Fourier space before transforming back the samples into real space through the use of Fourier transform. HMC requires evaluating the gradient of the negative logarithm of the posterior distribution. The latter is made of the contribution of the prior term and the likelihood term. Indeed, from Equation (2.38) one has

$$\nabla E(\mathbf{x}) = \nabla \left(-\ln \pi_0 \left(\mathbf{x}\right)\right) + \nabla \left(-\ln \pi \left(\text{data}|\mathbf{x}\right)\right).$$

Assuming a standard Gaussian prior distribution, the first term is the gradient of a quadratic form whose explicit expression is straightforward:

$$-\ln \pi_0 \left(\mathbf{x}\right) = \frac{1}{2}||\mathbf{x}||^2 + \text{cst}$$

and thus:

$$\nabla \left(-\ln \pi_0 \left(\mathbf{x}\right)\right) = \mathbf{x}.$$

As for the second term, which is nothing but the gradient of the negative log-likelihood, assuming a Gaussian likelihood will also yield a straightforward expression:

$$-\ln \pi \left(\text{data}|\mathbf{x}\right) = \sum_{i=1}^{d} \frac{(\rho(\mathbf{x})_i - \rho(\text{data})_i)^2}{2\sigma^2},$$

where $\rho(\mathbf{x})_i$ is the final density fluctuation in cell number i, from initial seed $\mathbf{x}$, and $\rho(\text{data})_i$ is the final density fluctuation in cell number i, from true mock seed data. The evaluation of these quantities requires applying the forward gravitational model. As for evaluating the gradient of this function with respect to the $\mathbf{x}$ variable, this is done with an adjoint method. This is crucially the most expensive part of the procedure.

Symplectic integration

As seen from previous sections, HMC also relies on three hyperparameters used in the numerical integration of Hamilton's equations with a Leapfrog integrator: the number of steps per iteration L, the integration timestep δt and the mass matrix used to draw artificial momenta $\mathcal{M}$. The number of steps per iteration directly controls the numerical cost of a whole run. Indeed, as the Leapfrog steps are chained up, and one can re-use the gradient from the last step to inject it into the new one, the number of steps per sample is equal to $L + 1$. The tuning of the three parameters dramatically affects the overall performance of the algorithm. Choosing the mass matrix is simple. With the BORG framework, it is set to I_d. As for L and δt, their value is chosen to minimize the integrated autocorrelation time at fixed computational budget. Formally, the user does not provide a fixed value for these two parameters. Rather, the input values of L and δt is the maximum value of the parameters. At each iteration, a random number is drawn uniformly between 1 and L (for the number of steps) and 0 and δt (for the integration timestep). This is to avoid resonant trajectories to occur.

Optimal tuning is different whether one is in the thermalized regime or not. This is why the ideal set of parameters used to explore the typical set may be highly inefficient when starting far from this region, thus leading to a high rejection rate. In a recent update, an automatic adaptive scheme has been implemented to adjust parameter δt, allowing the process not to spend too much time in the mixing phase. Another option is to divide the HMC run in two distinct step: the mode-searching phase, with small δt for reducing the rejection rate, followed by a typical set exploration once the correct region has been found, with usual tuning of the parameters. The second strategy is of course more costly since it requires tuning for both warm-up and relaxation.

Towards other Hamiltonian-based samplers for cosmology

Recently, new algorithms based on Hamiltonian dynamics have emerged for field-level inference. These involve Microcanonical Langevin Monte Carlo (Robnik and Seljak, 2023; Bayer et al., 2023) or hybrid methods combining HMC with a proposal learned by a neural network (Gabrié et al., 2022; Modi et al., 2023). This is because using Hamiltonian dynamics for exploring complex parameters space have proven efficient in so many cases (Betancourt, 2018). Also, Hamiltonian dynamics is based on solid and well-understood principles, making these algorithms prone to be highly-interpretable.

The first method addresses the fact that ensuring ergodicity by resampling the velocity component according to a Gaussian distribution may result in inefficiencies (Betancourt, 2018). An alternative is given for instance by Microcanonical Hamiltonian Monte Carlo (MCHMC) (Robnik et al., 2024). Unlike HMC, it generates samples from the target distribution by keeping constant energy, which is achieved by integrating the following equations of motion:

$$\begin{cases} \frac{d\mathbf{x}}{dt} &= \mathbf{p} \\ \frac{d\mathbf{p}}{dt} &= \frac{-1}{d-1}(I_d - ||\mathbf{p}||^2)\nabla E(\mathbf{x}) \end{cases}$$

However, ergodicity is not ensured since constant energy could imply being trapped in an energy sublevel, for instance in the case of symmetrical distributions. One solution to this problem is to consider refreshment of momenta, as in the standard HMC procedure. Another approach is to consider physical dynamics that already exhibit some stochastic behavior. This motivated the design of Microcanonical Langevin Monte Carlo sampler (Robnik and Seljak, 2023). It is a modification of MCHMC which relies on Langevin dynamics. The equations of motions are now:

$$\begin{cases} d\mathbf{x} &= \mathbf{p}dt \\ d\mathbf{p} &= (I_d - ||\mathbf{p}||^2)(-\nabla E(\mathbf{x}) + \eta d\mathbf{W}) \end{cases}$$

with $\mathbf{W}$ a vector drawn from a standard Gaussian distribution in dimension d. This Gaussian diffusion term adds stochasticity to the process for accelerating its exploration. The process is ergodic under

mild assumptions regarding the target distribution (Robnik and Seljak, 2023). Such algorithm has been tested in the context of field-level inference (Bayer et al., 2023), both for inferring the modes of the initial density fluctuations field or for the estimations of parameters from the Lambda-CDM model. In the first scenario, for dimensions 32^3 and 64^3, numerical experiments exhibit improved efficiency compared to a baseline HMC by multiple dozen of orders of magnitude. The explanation given by the authors is that the optimal step size within the HMC integrator decreases as the number of dimensions gets bigger. MCLMC does not have such limitation. One can thus spot an increase in the relative performance of the two algorithms with dimension.

As for the second method, it tries to combine the best of both worlds between exactness of MCMC methods and cheaper numerical cost of generative models (Gabrié et al., 2022). It is based on a two-step process:

- **Step 1: Training.** A vanilla HMC is run. Generated samples are used to train the generative model-based proposal through maximization of the samples log-probability under the variational distribution.
- **Step 2: Sampling.** With probability p, the sampling procedure alternates between proposals from the usual HMC and from the model. The variational model is still updated on the fly.

These different alternatives offer multiple perspectives and could, in the middle term, be integrated within the BORG framework as many of the requirements for an efficient and straightforward implementation are met: an object-oriented framework, a HMC sampler and lots of experience regarding the calibration of Hamiltonian-based algorithms.

Conclusion

In this chapter, we introduced the cosmological tools needed for studying the large-scale structure of the universe in a Bayesian perspective using a forward gravitational model. We also presented the BORG machinery along with its HMC sampler for inferring the initial conditions of the universe from data. Generating samples of the primordial density field is of great interest for cosmology. Indeed, such samples can be used to make re-simulations of the cosmic history that may be crossed with additional astronomical data and highlighted a few research lines for the future, based on recent advances in the MCMC and Machine Learning litterature. These correlations can in turn be used to constrain the parameters of physical models (Desmond et al., 2018; Bartlett et al., 2021a,b, 2022). The sampling procedure, however, relies on a reversible MCMC algorithm with a Metropolis accept-reject step that may suffer from diffusive behavior in high-dimensional spaces. In the next chapter, we introduce a non-reversible MCMC samplers as an alternative to the baseline HMC used in BORG.

Chapter 3

Piecewise Deterministic Monte Carlo for inferring the initial conditions of the universe

They act as vacuums in which people disappear. But where do they go? What lies behind a black hole? Along with things, do space and time also vanish there? Or would space and time be tied together and be part of an endless cycle? What if everything that came from the past were influenced by the future?

H.G. Tannhaus, *Dark, Season 1: Double Lives*

This work aims at discussing the implementation of a non-reversible sampler in the context of field-level inference. The goal is to introduce a PDMC algorithm for inferring the primordial fluctuations of the universe. We provide numerical comparisons between this new algorithm and a baseline Hamiltonian Monte Carlo sampler, which is the current state-of-the-art. We also explain how to fine-tune and use this algorithm in more general frameworks.

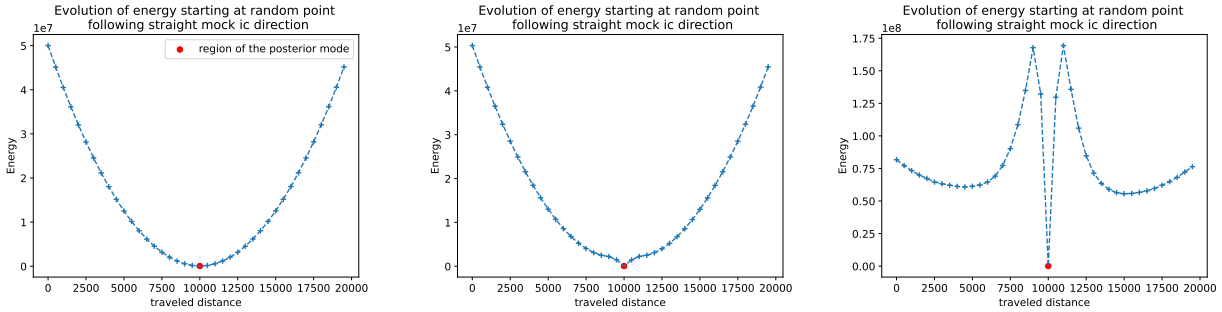
3.1 Description of the target distribution

The goal of this project is to generate samples from the LSS Bayesian posterior derived in Section 2. We are particularly interested in two properties: the convexity of the posterior, as it would allow the design of an efficient thinning strategy; and the anisotropy of the problem, which is an indication of its complexity: the more anisotropic, the more differences should be spotted between PDMC and HMC-BORG.

3.1.1 Convexity analysis

Let us investigate the convexity of the inference problem. Our tests consist in using a PDMC dynamics and check that in between two events, the directional derivative is increasing. Also, one may plot the energy function along a fixed direction to check whether the profile is convex or not. Passing these tests is important since the convexity assumption is at the basis of the thinning scheme that we further propose. However, even if this was not ensured, we could still argue that the algorithm is valid up to some error bar which could be quantified. We can decompose our findings in two conclusions, depending on the location of the process:

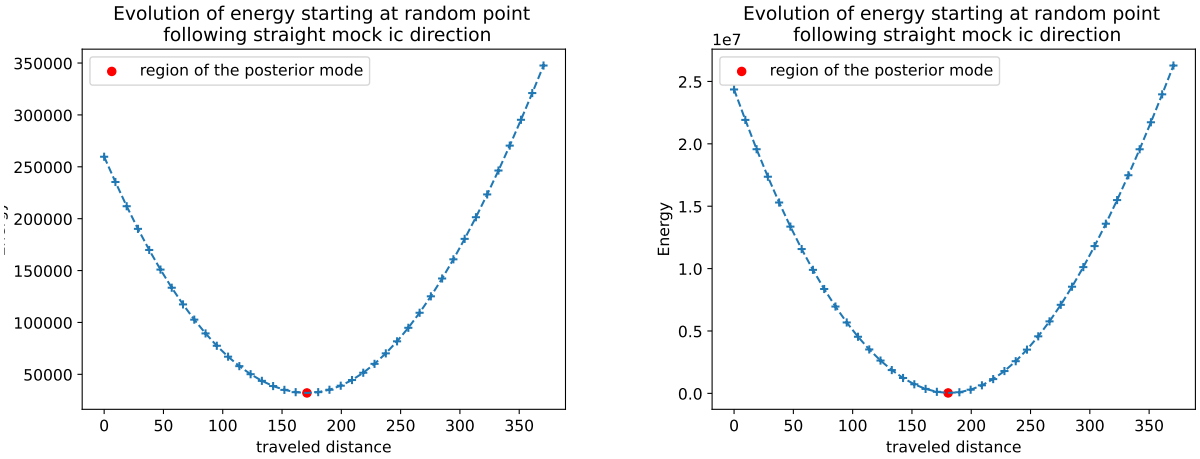
- When starting far from the mode, the process has to explore a complex energy landscape to reach the region of interest. Indeed, the problem is not convex as the noise gets smaller and smaller. See for instance in Figure 3.1 the evolution of energy $E(\mathbf{x}) = 0.5||\mathbf{x}||^2 + \text{NLL}(\mathbf{x})$ starting from a random position $\mathbf{x}$ and following a straight line in the direction of the mode region. Such behavior suggests that the target energetic landscape is very hard to explore. Also, the convexity assumption is not suited to the exploration of out-of-the-mode regions. Efficient mixing will thus demand careful initialization and a smart thinning strategy.



(a) $\sigma = 1.0$. (b) $\sigma = 0.1$. (c) $\sigma = 0.01$.

Figure 3.1: While being locally convex around the mode, this is not the case anymore when getting far from the typical set region in a small noise regime.

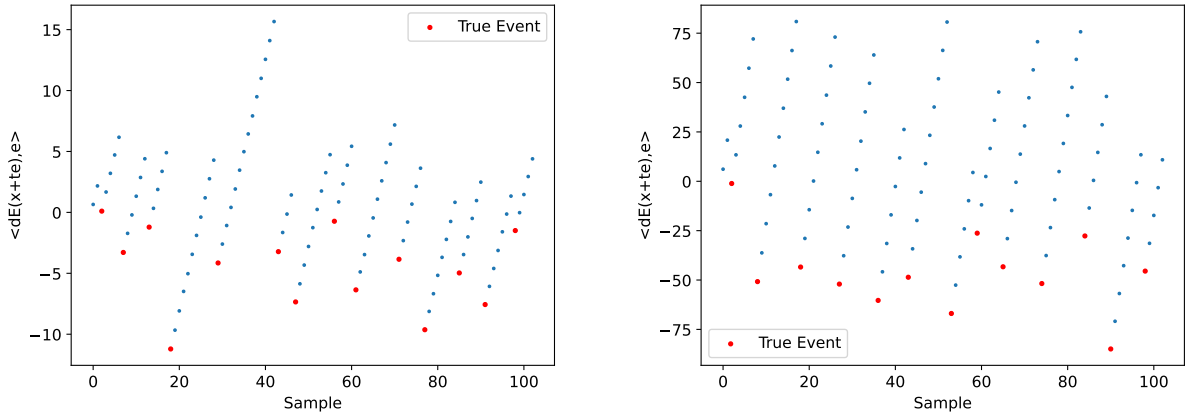
- However, when starting close to the mode, for example at the mock initial conditions, then no pathological behavior was ever spotted. For the two problematic noise levels $\sigma = 0.1$ and $\sigma = 0.01$ mentioned below, a direct observation of the energy landscape seems to indicate that the region is convex, see Figure 3.2. Also, when running a PDMC sampler starting in this region, the directional derivatives are always increasing indicating that the trajectory remains confined to this smooth region, see Figure 3.3.



(a) $\sigma = 0.1$. (b) $\sigma = 0.01$.

Figure 3.2: While being globally non-convex, the typical set around the target mode seems convex even for small noise regimes.

It is interesting to note that decreasing the noise level ends up in drawing a highly challenging energy landscape. The resulting target log-distribution is multimodal with global maximum surrounded by big energy peaks: starting very far from this region will lead to exploring the tails of the distribution, with absolutely no guarantee of targeting the right region of interest with a fixed reasonable amount of



(a) $\sigma = 0.1$. (b) $\sigma = 0.01$.

Figure 3.3: Convexity around the mode for small noise regimes is assessed by tracking the successive directional derivatives that are increasing in between events, i.e. along each direction explored by a PDMC sampler starting from a point located near the mode.

computational resource. In our experiments, we observed that the process could easily escape from the region of interest if the initial direction was not aligned with the mode direction.

The desired behavior is to explore the region around the global minimum as it corresponds to realizations of the density field that have a physical meaning. This is why one does not need to be fully ergodic, in the sense that one does not seek to explore regions that are outside this convex region. Instead, the ultimate goal is to reach the global minimum region rapidly and then explore the region around it in order to generate samples associated with a plausible beginning for cosmic history.

3.1.2 Anisotropy analysis

For the comparison to be relevant, the two algorithms must be tested in a sufficiently *difficult* regime. The difficulty of the problem is linked to the characteristic size of the displacement associated with the gravitational model used, in this case Lagrangian Perturbation Theory (LPT). Morally, any resolution smaller than this characteristic size will be unable to see the non-linear effects caused by this displacement. Also, the noise level combines with the previous effect. Indeed, if the noise is very high, then the problem becomes Gaussian again, and can even compensate for very high resolution. Conversely, if the noise is very low, then the target posterior distribution cannot be obtained simply by inverting the problem's covariance matrix.

In the inference problem that we are trying to solve, it appears that the target distribution is still close to a high-dimensional isotropic Gaussian, even for high resolution and small noise regimes. This is illustrated in Figure 3.4 below, showing a histogram of variances along each dimension of samples produced by a baseline HMC as well as a PDMC in a thermalized regime. This plot is made in a regime where the noise is about 20 smaller than the required noise for the problem to be characterized as difficult. Even for such difficult problem, the distribution of variances remains bounded between extremal values not far from each other, namely between 0.003 and 0.040.

Anisotropy directly governs the performance of the two algorithms that we are comparing. Indeed, the non-reversible ballistic trajectories drawn by a PDMC algorithm outperform the reversible Hamiltonian exploration of HMC, especially on high-dimensional anisotropic distributions (Michel et al., 2020). However, there is a trade-off between the performing dynamics of the PDMC and the numerical cost

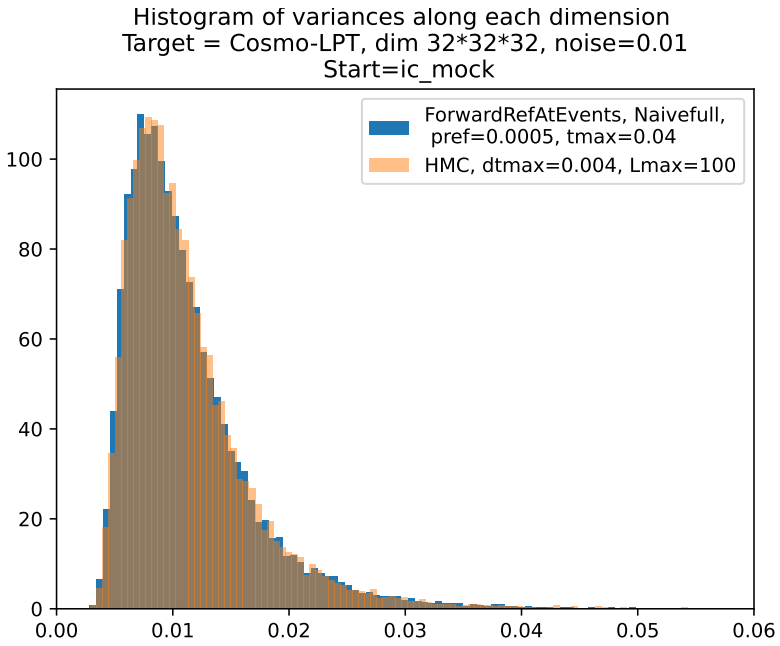


Figure 3.4: Histograms of samples variances along the different dimensions.

needed to implement it, as will be discussed in Sections 3.2 and 3.3.

3.2 Forward-Event Chain sampler for BORG

We describe here the main features of PDMC-BORG, a new non-reversible sampling algorithm for inferring the initial conditions of the universe from data.

3.2.1 General description of the sampler

Type of sampler

PDMC-BORG is a non-reversible Piecewise Deterministic Monte Carlo sampler. The idea is to test how this type of algorithms behave on a complex Bayesian inference problem, and compare their performance to that of a baseline Hamiltonian Monte Carlo. We decided to implement an instance of *Forward Event Chain Monte Carlo* (Michel et al., 2020). This non-reversible and rejection-free class of Event-Chain Monte Carlo allows to reduce extra-randomization of the process necessary for ergodicity purpose to a minimum and exhibited clear acceleration and better scaling with dimensions compared to classical MCMC methods and other PDMC samplers (Michel et al., 2020). By design, this class of samplers could be a challenger to Hamiltonian Monte Carlo in the context of high-dimensional field-level Bayesian inference and we test this hypothesis with numerical experiments.

Since it is implemented within the BORG framework, PDMC-BORG uses some features that have already been introduced when presenting HMC-BORG. It notably relies on the energy function E associated to the target LSS posterior and its gradient ∇E, as explained in Subsection 2.3.3. For testing algorithms, it is common to use mock data, i.e. artificial data generated from a random seed $\mathbf{s}$ corresponding to a sample from the primordial field, that is transformed into the evolved field with a cosmological forward-model. This will be the approach for our numerical experiments.

Computing next-event time

The computation of event times is a challenging part in PDMC algorithms, see Subsection 1.2.3. For the Bayesian problem at stake, exact computation of the event times in not possible since the quantity $\int_0^t \max\left(0, \langle \nabla E(\mathbf{x} + s\mathbf{e}), \mathbf{e}\rangle\right) ds$ is analytically intractable. As explained in Subsection 1.2.3, we instead derive a local thinning strategy, which requires finding a bound on the above integrand. In a first version of PDMC-BORG, this bound is automatically computed with a bounded numerical optimizer. But one can also rely on the fact that the target posterior distribution is convex around the region of interest. Setting $t_{\max} > 0$ the maximum horizon, a local bound on the directional derivative can thus be derived as follows: if the system is at $(\mathbf{x}_t, \mathbf{e}_t)$ at time t, then a local bound is given by

$$\lambda_t^* = \max\left(0, \langle \nabla E(\mathbf{x}_t + t_{\max}\mathbf{e}_t), \mathbf{e}_t\rangle\right).$$

This leads to Algorithm 7, which is a direct adaptation of a local thinning algorithm (Lewis and Shedler, 1979; Corbella et al., 2022) to the case where the target energy is convex.

Algorithm 7 Local thinning algorithm for convex energy function in PDMC.

Require: $\mathbf{x}$, $\mathbf{e}$, $t_{\max}$, ∇E
$t \leftarrow 0$
$\tilde{\chi} \leftarrow \max\left(0, \langle \nabla E(\mathbf{x} + t_{\max}\mathbf{e}), \mathbf{e}\rangle\right)$
if $\tilde{\chi} > 0$ **then**
  while True **do**
    Draw $u \sim \mathcal{U}[0,1]$
    $w \leftarrow \frac{-\log(u)}{\tilde{\chi}}$
    $t \leftarrow t + w$
    if $t > t_{\max}$ **then**
      Return $t_{\max}$
    end if
    Draw $D \sim \mathcal{U}[0,1]$
    $R = \frac{\max(0, \langle \nabla E(\mathbf{x}+t\mathbf{e}), \mathbf{e}\rangle)}{\tilde{\chi}}$
    if $D < R$ **then**
      Return t
    end if
  end while
end if
Return $t_{\max}$

Refreshment strategy

To make sure that PDMC-BORG is ergodic, we introduce a refreshment step. We implemented two different choices.

- The first one consists in a refreshment of the perpendicular component **at each event** with probability p_{ref}. In this case, a valid option (Michel et al., 2020) is to draw $\mathbf{u} \sim \mathcal{N}(0, I_d)$ and take :

$$\mathbf{n}_{\text{perp}} = \frac{\mathbf{u} - \langle \mathbf{u}, \mathbf{n}_{\text{par}}\rangle \mathbf{n}_{\text{par}}}{||\mathbf{u} - \langle \mathbf{u}, \mathbf{n}_{\text{par}}\rangle \mathbf{n}_{\text{par}}||}$$

  The resulting direction change at events function is detailed in Algorithm 8.

- Another possibility is to refresh direction **at every fixed timestep** δt_{ref}. This is a popular option among the PDMC sampling community (Michel et al., 2014, 2020; Monemvassitis et al., 2023). One may choose for instance a global refreshment by picking a random vector uniformly

Algorithm 8 Direction update in PDMC-BORG with refreshments at events

Require: $\mathbf{x}$, $\mathbf{e}_0$, $\mathbf{n}_{\text{par}}$, p_{ref}, ∇E
Draw $\nu \sim \mathcal{U}[0,1]$
$a \leftarrow \nu^{1/(d-1)}$
$b \leftarrow \sqrt{1-a^2}$
$\mathbf{n}_{\text{par}} \leftarrow \frac{\nabla E(\mathbf{x})}{||\nabla E(\mathbf{x})||}$
Draw $u \sim \mathcal{U}[0,1]$
if $u < p_{\text{ref}}$ **then**
    Draw $\mathbf{v} \sim \mathcal{N}(0, I_d)$
    $\mathbf{v} \leftarrow \mathbf{v} - \langle \mathbf{n}_{\text{par}}, \mathbf{v} \rangle \, \mathbf{n}_{\text{par}}$
    $\mathbf{n}_{\text{perp}} \leftarrow \frac{\mathbf{v}}{||\mathbf{v}||}$
    Return $a\mathbf{n}_{\text{perp}} - b\mathbf{n}_{\text{par}}$
end if
$\mathbf{v} \leftarrow \mathbf{e}_0 - \langle \mathbf{n}_{\text{par}}, \mathbf{e}_0 \rangle \, \mathbf{n}_{\text{par}}$
$\mathbf{n}_{\text{perp}} \leftarrow \frac{\mathbf{v}}{||\mathbf{v}||}$
Return $a\mathbf{n}_{\text{perp}} - b\mathbf{n}_{\text{par}}$

on the unit sphere $\mathcal{S}^{d-1}$. Note that such refreshment could also be performed at the arrival of an independent homogeneous Poisson process with rate corresponding to the inverse of the distance δt_{ref}.

3.2.2 An automatic version of PDMC-BORG

The first version we propose is a general version of PDMC-BORG which does not rely on any convexity assumption about the target distribution. We call it *Automatic PDMC-BORG*. This algorithm uses an optimization procedure with bounded constraints to compute the next event time using a local thinning strategy. More precisely, the optimizer solves the following maximization problem within an interval $[0, t_{\max}]$, with $t_{\max} > 0$ a fixed horizon:

$$\max_{t \in [0, t_{\max}]} (0, \langle \nabla E(\mathbf{x}_t + t_{\max}\mathbf{e}_t), \mathbf{e}_t \rangle).$$

Note that we want to find a maximum of a certain function but this problem is equivalent to finding the minimum of its opposite. In practice, our solution relies on an instance of Brent's algorithm (Brent, 1972) which is implemented within the Python SciPy library (Jones et al., 2001) as the `scipy.optimize.minimize_scalar` function with the 'Bound' option. Let us call $f : \mathbb{R} \longrightarrow \mathbb{R}$ the objective function for which a local minimum needs to be found. The procedure is a combination between successive parabolic interpolation and Golden-section search. Successive parabolic interpolation works as follows: given a current position t_i within a given interval, the guess at iteration $i+1$ depends on the guess at iteration i, $i-1$ and $i-2$ as

$$t_{i+1} = t_i + \frac{1}{2}\left[\frac{(t_{i-1}-t_i)^2(f(t_i)-f(t_{i-2})) + (t_{i-2}-t_i)^2(f(t_{i-1})-f(t_i))}{(t_{i-1}-t_i)(f(t_i)-f(t_{i-2})) + (t_{i-2}-t_i)(f(t_{i-1})-f(t_i))}\right]$$

If the next candidate for the minimization is rejected, which may happen because it lies outside from the given interval or because two of the three points are the same, then the algorithm computes the next candidate using a Golden-section search: given an interval of points $[t_a, t_b]$, two points are defined within the interval as:

$$t_c = a + \frac{(b-a)}{r}$$
$$t_d = b - \frac{(b-a)}{r}$$

where $r = \frac{1+\sqrt{5}}{2}$ is the golden ratio. Then the local minimum among these points lies in an interval formed by the two closest neighbors to the smallest evaluated candidate so far. These two neighbors define new bounds a and b and the process can be repeated.

The procedure stops when a stopping criterion is reached, for instance when the difference between two successive candidates is smaller than a certain tolerance threshold.

In the case of BORG, some acceleration strategies could be investigated as the energy landscape seems locally not too complex. For instance, if we assume that for a sufficiently small $t_{\max}$ the function can have at most one mode within the optimization interval, then one may want to check the sign of the directional derivatives at the bounds of the interval. If they are of opposite sign, then one may want to run Brent's algorithm. If they have same sign, then it is an indication that the function is monotonic. In this case, the minimum necessarily lies at one bound of the interval.

While the implementation is available, we chose not to further investigate such strategy as its numerical cost would be prohibitive. Indeed, if the numerical optimizer requires n_s steps on average, i.e. n_s gradient evaluations, then this would make the computation of each event n_s times expensive than a standard thinning procedure, which itself may be costly. For the target at stake, we found that using additional information about the convexity of the target could leap to a less expensive sampler competitive with HMC-BORG. We thus leave for future work precise detailed numerical experiments on Automatic PDMC-BORG.

3.2.3 A two-step version PDMC-BORG

The second version is based on the hypothesis that the energy landscape around the mode is convex. So one possibility is to set up a two-step algorithm with a mode-searching phase followed by an exploration of the typical set. This leads to a sampler with reduced numerical cost that makes it directly comparable with the baseline HMC-BORG in terms of numerical efficiciency. We call this version *two-step PDMC-BORG*.

Mode-searching phase

One needs to be careful about the fact that the convexity hypothesis does not hold far from the target mode. However, starting not too far from the mode means exploring a region that is not overly complex. The main difficulty is to pick a good initial direction in order to target the mode region. Indeed, choosing a random direction will lead to an escaping trajectory that may end up in the tail of the distribution, or event trapped in a local minimum. One possibility is thus to design a straightforward procedure to get closer to the mode at the very beginning of the run. We decided to implementent a *biased* PDMC for the warm-up phase. The idea is to run PDMC-BORG assuming that the region is convex, and perform a refreshment of direction at fixed small timestep δt_{ref} that is biased by the local gradient. We propose to use a noisy gradient descent for refreshing the direction: at every fixed δt_{ref}, one starts by drawing $u \sim \mathcal{U}[0,1]$. Then, one sets $a = u$, $b = \sqrt{1-a^2}$ and draw $\varepsilon \sim \mathcal{N}(0, I_d)$. The new direction is written as:

$$\mathbf{e}_{\text{new}} = -a \cdot \frac{\nabla E(\mathbf{x})}{||\nabla E(\mathbf{x})||} + b \cdot \frac{\varepsilon}{||\varepsilon||}.$$

Note that this scheme is not exact in the sense that refreshment is biased by the gradient. Also, we use a convexity assumption to draw the next event time even though convexity is not ensured when not starting close to the mode. Our choices are motivated by the following facts: first, as the process does not start too far from the mode, if the direction is chosen correctly, then the convexity assumption is not a problem since we remain in a smooth almost-convex region. This claim is particularly true if one follows the gradient direction, as shown in Figure 3.5 which illustrates how the directional derivatives are slightly increasing at the beginning of a run starting close to 0, using this warm-up procedure. This

indicates that the biased PDMC follows an almost convex trajectory towards the mode. And second, as the main objective is to get closer to the mode, we do not put to much attention on exactness of the procedure. Indeed, the very first samples will be progressively discarded in the Monte Carlo estimates as we reach the convex region and start becoming exact.

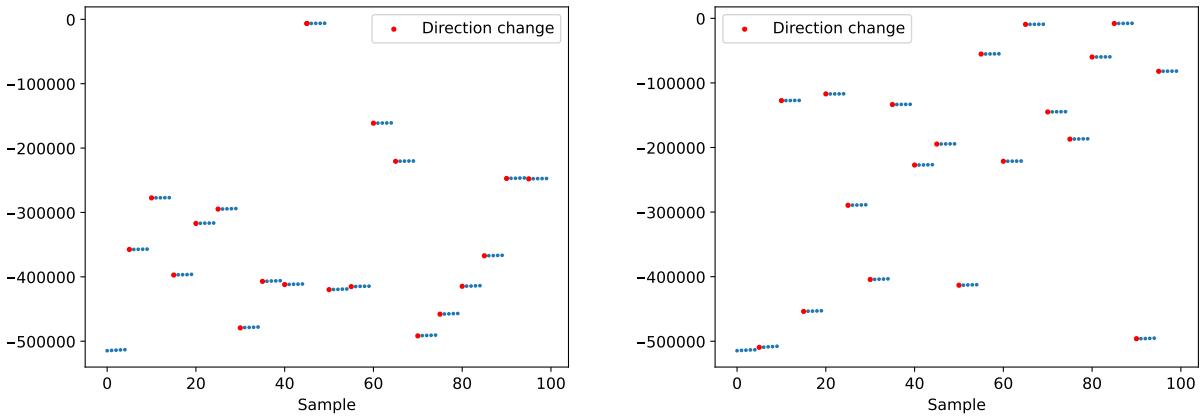


Figure 3.5: Successive directional derivatives at the beginning a warm-up trajectory for two different starting positions drawn from a multivariate Normal with mean 0 and covariance matrix $0.01 I_d$. Starting close to 0 and using the mode-searcher leads to explore a quasi-convex region, as the directional derivatives are slightly increasing in between events. The convexity assumption is thus justified.

We leave for future work the exploration of other kinds of warm-up strategies such as:

- Run a HMC. In this case, an automatic tuning strategy for setting the timestep for the Leapfrog integration would be required.
- A full noisy gradient descent strategy, without use of the thinning procedure and computation of events. This would make the procedure even more biased but could save computational resources as one does not have to compute the event times.

The procedure for relaxation is described by Algorithm 9. It is run until a stopping criterion is reached. The latter is discussed in the next paragraph.

Transition between mode-searching and relaxation phases

The two questions that need to be addressed are the following:

1. at which point do we make the transition between the two phases?
2. what direction should we choose to make the transition?

To address the first point, we propose to use the same criterion as the one used in HMC-BORG: the process is considered to be in its warm-up phase as long as the variance of the samples is less than a threshold value t. This is justified because in the BORG framework, the samplers are initialized close to 0, so the variance of samples is increasing at the beginning of the trajectory. In practice, one chooses $t \approx 0.85$, as the variance of the samples in the thermalized regime is close to 1. During warm-up, the variance will keep increasing until it eventually reaches the desired value. This is an indication that we are getting closer to the global minimum region, and that we are entering a convex enough region for

Algorithm 9 Sampling with PDMC-BORG during warm-up phase using a local thinning procedure.

Require: $t_{\max},\ \delta t_{\text{samples}}, \delta t_{\text{ref}},$ Stopping Criterion

Draw $\mathbf{x} \sim \mathcal{N}(0, 0.01 I_d)$
$\mathbf{e} \leftarrow \frac{\nabla E(\mathbf{x})}{||\nabla E(\mathbf{x})||}$
$t \leftarrow 0$ # process time
$t_{\text{tosample}} \leftarrow \delta t_{\text{samples}}$
$t_{\text{toref}} \leftarrow \delta t_{\text{ref}}$
while not Stopping Criterion **do**
    Draw dt (either next event time or $t_{\max}$) with Algorithm 7
    isEvent $\leftarrow$ False
    $t_{\text{toevent}} \leftarrow dt$
    while not isEvent **do**
        $m \leftarrow \min\{t_{\text{toevent}}, t_{\text{sample}}, t_{\text{ref}}\}$
        if $m = t_{\text{toevent}}$ **then**
            $t \leftarrow t + t_{\text{toevent}}$
            $t_{\text{tosample}} \leftarrow t_{\text{tosample}} - t_{\text{toevent}}$
            $t_{\text{toref}} \leftarrow t_{\text{toref}} - t_{\text{toevent}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{toevent}}\mathbf{e}$
            if $dt < t_{\max}$ **then**
                Update direction $\mathbf{e}$ with Algorithm 8 and $p_{\text{ref}} = 0$
            end if
            isEvent $\leftarrow$ True
        else if $m = t_{\text{tosample}}$ **then**
            $t \leftarrow t + t_{\text{tosample}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{tosample}}\mathbf{e}$
            $\mathbf{s}_{n_{\text{samples}}} \leftarrow \mathbf{x}$
            $n_{\text{samples}} \leftarrow n_{\text{samples}} + 1$
            $t_{\text{toevent}} \leftarrow t_{\text{toevent}} - t_{\text{tosample}}$
            $t_{\text{toref}} \leftarrow t_{\text{toref}} - t_{\text{tosample}}$
            $t_{\text{tosample}} \leftarrow \delta t_{\text{samples}}$
        else if $m = t_{\text{toref}}$ **then**
            $t \leftarrow t + t_{\text{toref}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{toref}}\mathbf{e}$
            Draw $u \sim \mathcal{U}[0,1]$ and $\varepsilon \sim \mathcal{N}(0, I_d)$
            $a \leftarrow u$
            $b \leftarrow \sqrt{1 - a^2}$
            $\mathbf{e} \leftarrow -a \cdot \frac{\nabla E(\mathbf{x})}{||-\nabla E(\mathbf{x})||} + b \cdot \frac{\varepsilon}{||\varepsilon||}$
            $t_{\text{tosample}} \leftarrow t_{\text{tosample}} - t_{\text{toref}}$
            $t_{\text{toref}} \leftarrow \delta t_{\text{ref}}$
            isEvent $\leftarrow$ True ▷ Direction has changed so new event time needs to be computed
        end if
    end while
end while
Return $\mathbf{s}_1, \mathbf{s}_2, ...$

running the usual version of PDMC-BORG.

As for choosing the transition direction, we chose to set it to a random normalized vector. It would be for example inappropriate to align it with the gradient direction since the trajectory would then fall in the global minimum. Ideally, one would like to align with the direction orbiting around the mode, since most of the probability mass will lie on a thin hypersurface around the global maximum. This is possible if one chooses a random direction which will then use the directional derivatives to draw persisting moves in the right direction and orbit around the mode.

Relaxation phase

For the BORG inference-problem, remember from Subsection 3.1.1 that we make the following strong assumption:

The energy landscape to be explored is convex in a neighborhood of the global minimum.

This is essentially the same as assuming that the target posterior distribution is locally log-concave - for example, a Gaussian. This assumption is justified by careful numerical experiments on this target: indeed, it appears that the directional derivative is increasing in each direction explored by the algorithm when testing ballistic trajectories around the mode.

The general procedure for running a PDMC and sampling uniformly along its trajectory is summed up in Algorithm 10 for a procedure with refreshment at events, or in Algorithm 11 for a procedure with refreshment outside events. It differs from the mode-searching phase in the way it deals with refreshments of direction. First, the new scheme is measure-preserving, hence the process targets the correct posterior distribution. And finally refreshment is done at events. This is because of numerical reasons: indeed, as one already computes the gradient at the position where direction will be updated during the thinning procedure, then one can re-use this result to solve the event. If one chooses instead to refresh outside of events, then it is necessary to compute the gradient at the event time twice: during the thinning procedure and once the process reaches the event position.

3.3 HMC vs. PDMC

We now investigate the numerical performance of the two-step PDMC-BORG and compare them to that of a baseline HMC. The following experiments should be seen as a proof-of-concept for PDMC-BORG. The goal is to check whether it is able to solve the Bayesian sampling problem of interest. Also, it may serve as an illustration of the nice features of non-reversible samplers that may incite cosmologist practitioners to use them.

Let us place ourselves in the following regime, considered as difficult because of the box resolution and the noise level:

- Physical cubic box with side $L = 677.7$ Mpc$/h$;
- Dimension $d = 32^3 = 32768$, which corresponds to the number of cubic cells inside the volume;
- Standard deviation of the noise $\sigma = 0.01$. This is about 20 times smaller than the required noise for the problem to be considered as difficult.

The goal is to infer the posterior distribution of the primordial density field given artificial initial mock data generated from a standard Gaussian. We generate $20,000$ samples in each experiment. We first present some metrics used for comparisons and explain how to tune the two algorithms.

Algorithm 10 Sampling N_{samples} samples, separated by fixed time-interval $\delta t_{\text{samples}}$, with PDMC-BORG during relaxation phase using a local thinning procedure and **refreshment at events**.

Require: $t_{\max}$, p_{ref}, $\delta t_{\text{samples}}, N_{\text{samples}}$

Draw $\mathbf{x}$ close to the mode
Draw $\mathbf{u} \sim \mathcal{N}(0, I_d)$
$\mathbf{e} \leftarrow \frac{\mathbf{u}}{||\mathbf{u}||}$
$t \leftarrow 0$ # process time
$n_{\text{samples}} \leftarrow 0$ # number of samples
$t_{\text{tosample}} \leftarrow \delta t_{\text{samples}}$
while $n_{\text{samples}} < N_{\text{samples}}$ **do**
    Draw dt (either next event time or $t_{\max}$) with Algorithm 7
    isEvent $\leftarrow$ False
    $t_{\text{toevent}} \leftarrow dt$
    while not isEvent **do**
        if $t_{\text{toevent}} < t_{\text{tosample}}$ **then**
            $t \leftarrow t + t_{\text{toevent}}$
            $t_{\text{tosample}} \leftarrow t_{\text{tosample}} - t_{\text{toevent}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{toevent}}\mathbf{e}$
            if $dt < t_{\max}$ **then**
                Update direction $\mathbf{e}$ with Algorithm 8
            end if
            isEvent $\leftarrow$ True
        else
            $t \leftarrow t + t_{\text{tosample}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{tosample}}\mathbf{e}$
            $\mathbf{s}_{n_{\text{samples}}} \leftarrow \mathbf{x}$
            $n_{\text{samples}} \leftarrow n_{\text{samples}} + 1$
            $t_{\text{toevent}} \leftarrow t_{\text{toevent}} - t_{\text{tosample}}$
            $t_{\text{tosample}} \leftarrow \delta t_{\text{samples}}$
        end if
    end while
end while
Return $\mathbf{s}_1, ..., \mathbf{s}_{N_{\text{samples}}}$

Algorithm 11 Sampling with PDMC-BORG during warm-up phase using a local thinning procedure and **refreshment outside events**.

Require: $t_{\max}$, $\delta t_{\text{samples}}, \delta t_{\text{ref}}$, Stopping Criterion

Draw $\mathbf{x}$ close to the mode
$\mathbf{e} \leftarrow \frac{\nabla E(\mathbf{x})}{||\nabla E(\mathbf{x})||}$
$t \leftarrow 0$ # process time
$t_{\text{tosample}} \leftarrow \delta t_{\text{samples}}$
$t_{\text{toref}} \leftarrow \delta t_{\text{ref}}$
while not Stopping Criterion **do**
    Draw dt (either next event time or $t_{\max}$) with Algorithm 7
    isEvent $\leftarrow$ False
    $t_{\text{toevent}} \leftarrow dt$
    while not isEvent **do**
        $m \leftarrow \min\{t_{\text{toevent}}, t_{\text{tosample}}, t_{\text{toref}}\}$
        if $m = t_{\text{toevent}}$ **then**
            $t \leftarrow t + t_{\text{toevent}}$
            $t_{\text{tosample}} \leftarrow t_{\text{tosample}} - t_{\text{toevent}}$
            $t_{\text{toref}} \leftarrow t_{\text{toref}} - t_{\text{toevent}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{toevent}}\mathbf{e}$
            if $dt < t_{\max}$ **then**
                Update direction $\mathbf{e}$ with Algorithm 8 and $p_{\text{ref}} = 0$
            end if
            isEvent $\leftarrow$ True
        else if $m = t_{\text{tosample}}$ **then**
            $t \leftarrow t + t_{\text{tosample}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{tosample}}\mathbf{e}$
            $\mathbf{s}_{n_{\text{samples}}} \leftarrow \mathbf{x}$
            $n_{\text{samples}} \leftarrow n_{\text{samples}} + 1$
            $t_{\text{toevent}} \leftarrow t_{\text{toevent}} - t_{\text{tosample}}$
            $t_{\text{toref}} \leftarrow t_{\text{toref}} - t_{\text{tosample}}$
            $t_{\text{tosample}} \leftarrow \delta t_{\text{samples}}$
        else if $m = t_{\text{toref}}$ **then**
            $t \leftarrow t + t_{\text{toref}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{toref}}\mathbf{e}$
            Draw $\varepsilon \sim \mathcal{N}(0, I_d)$
            $\mathbf{e} \leftarrow \frac{\varepsilon}{||\varepsilon||}$
            $t_{\text{tosample}} \leftarrow t_{\text{tosample}} - t_{\text{toref}}$
            $t_{\text{toref}} \leftarrow \delta t_{\text{ref}}$
            isEvent $\leftarrow$ True ▷ Direction has changed so new event time needs to be computed
        end if
    end while
end while
Return $\mathbf{s}_1, \mathbf{s}_2, ...$

3.3.1 Metrics for evaluating performance

In a D-dimensional space, we consider a set of S samples $\{X_s\}_{s=1}^S := \{(X_{s,1}, ..., X_{s,D})\}_{s=1}^S$ generated by a model and S samples $\{X_s^0\}_{s=1}^S := \{(X_{s,1}^0, ..., X_{s,D}^0)\}_{s=1}^S$ drawn from the true (test) dataset.

First-order statistics.

A first test consists in computing the empirical mean and standard deviation of the generated samples, or their cumulative values.

Power spectra.

This is a classical tool in cosmology for checking the convergence. It is a measure of the matter inhomogeneities in the universe as a function of scale. More specifically, an important quantity is the 2-point correlation function

$$\xi(\mathbf{r}) = \langle \delta(\mathbf{x}), \delta(\mathbf{x}+\mathbf{r}) \rangle = \int_V \delta(\mathbf{x})\delta(\mathbf{x}+\mathbf{r}) d^3\mathbf{x}$$

Power spectrum is defined as the Fourier transform of this correlation function, that is:

$$\xi(\mathbf{r}) = \frac{V}{(2\pi)^3} \int P(\mathbf{k}) e^{-i\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{k}$$

with the fundamental relation

$$P(\mathbf{k}) = |\hat{\xi}(\mathbf{k})|^2$$

In practice, this quantity is estimated using a bin method. By plotting the successive power spectra of each sample, one can check the convergence towards a reference power spectrum (the one obtained with mock initial conditions, for instance).

Autocorrelations.

Recall that for a sequence $(X_1, ..., X_S)$ and a function h, the autocorrelation at lag $k \in \{0, ..., S-1\}$ is defined as

$$C_h(k) = \frac{1}{\hat{\sigma}^2(S-k)} \sum_{i=1}^{S-k} (h(X_i) - \hat{\mu})(h(X_{i+k}) - \hat{\mu})$$

Here, $\hat{\mu}$ and $\hat{\sigma}$ are the sample mean and variance of $(h(X_1), ..., h(X_S))$, respectively. We can plot the autocorrelations for different functions of interest. One wants to get the fastest decorrelation possible, as this is a crucial issue when dealing with MCMC methods. The autocorrelation is indeed used to compute the Effective Sample Size (ESS). This is because in MCMC methods, one needs to compute expectations of the form $\mathbf{E}_\pi[h]$ with correlated samples. ESS formalizes the idea that the chain of samples contains redundant information. One could thus drop some samples and end up with the same amount of information. To do that, one needs to compute the integrated autocorrelation $\sum_k C_h(k)$. For a sequence of S samples, the ESS is defined as:

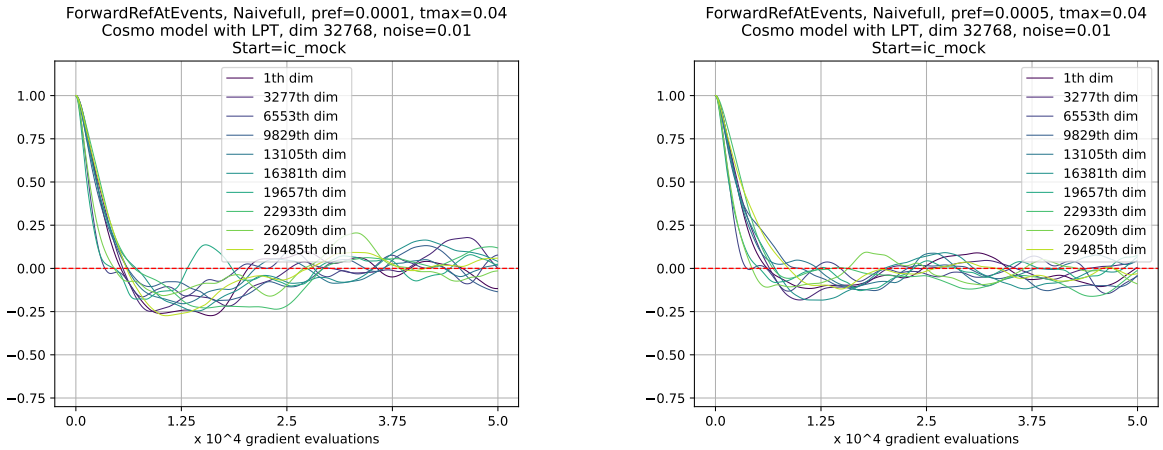
$$\mathrm{ESS}_h = \frac{S}{1 + 2\sum_k C_h(k)}$$

3.3.2 Tuning the algorithms

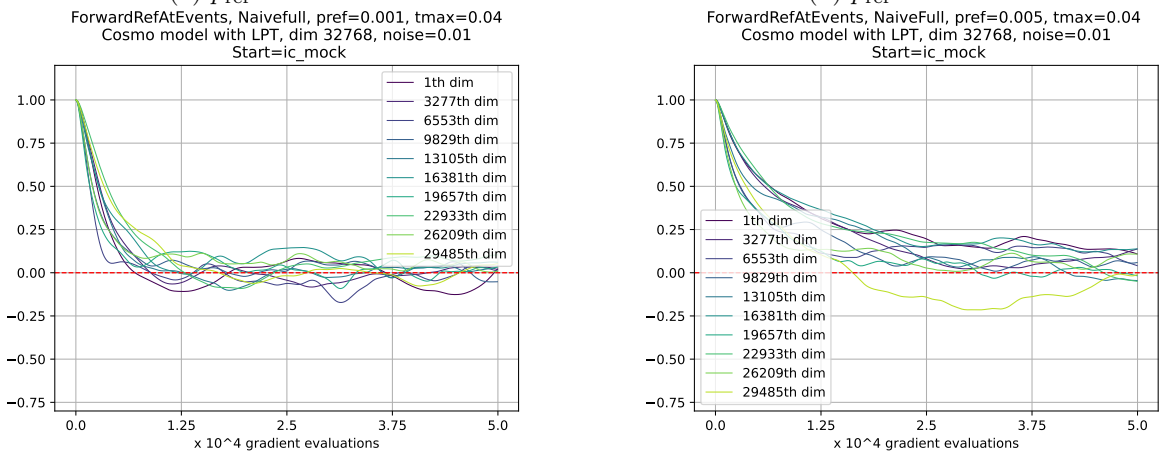
- **Tuning PDMC.** Only two parameters require tuning: the thinning maximal timestep $t_{\max}$ and the refreshment probability p_{ref} or the refreshment distance δt_{ref} if one chooses to refresh outside of events. The former has a direct impact on the number of gradient evaluations per event; the two latter have an influence on the autocorrelation plots. They are chosen so as to minimize the ratio between the number of gradient evaluations and the number of events and to achieve the fastest decorrelation time. The table below guides the choice for the best value for $t_{\max}$. Experiments correspond to short runs starting from the mock initial conditions with a fixed $p_{\text{ref}} = 1 \times 10^{-4}$ as varying the p_{ref} does not significantly affect the values. This leads to choosing $t_{\max} \in [0.03, 0.04]$ as the optimal value.

$t_{\max}$	0.01	0.02	**0.03**	**0.04**	0.05	0.06
#evaluations / #events	7.4	4.7	**4.5**	**4.5**	4.9	5.2

We then plot in Figure 3.6 the autocorrelations along a few dimensions achieved with a PDMC for different choices of p_{ref}. Experiments were conducted at $t_{\max} = 0.04$ and are based on 1M evaluation-runs starting from the mock initial conditions. They lead to choosing $p_{\text{ref}} = 5 \times 10^{-4}$ as a reference.



(a) $p_{\text{ref}} = 1 \times 10^{-4}$ (b) $p_{\text{ref}} = 5 \times 10^{-4}$



(c) $p_{\text{ref}} = 10 \times 10^{-4}$ (d) $p_{\text{ref}} = 50 \times 10^{-4}$

Figure 3.6: Autocorrelations along a few dimensions obtained with PDMC for different choices of p_{ref}.

- **Tuning HMC.** The algorithm has three tunable quantities - which is bigger than the number of tunable parameters for a PDMC: the mass matrix $\mathcal{M}$, the maximum number of Leapfrog steps

per sample L and the maximum timestep per sample δt. The mass matrix is set to $\mathcal{M} = I_d$; as for L and δt, they are chosen in order to achieve an acceptance rate of ~ 0.65 (Beskos et al., 2013). As mentioned before, it also benefits from an automatic warm-up phase allowing it to adjust the step-size in order to reach the mode rapidly. If not, the algorithm keeps rejecting samples as it is unable to find the right direction within a reasonable amount of time, just like the PDMC.

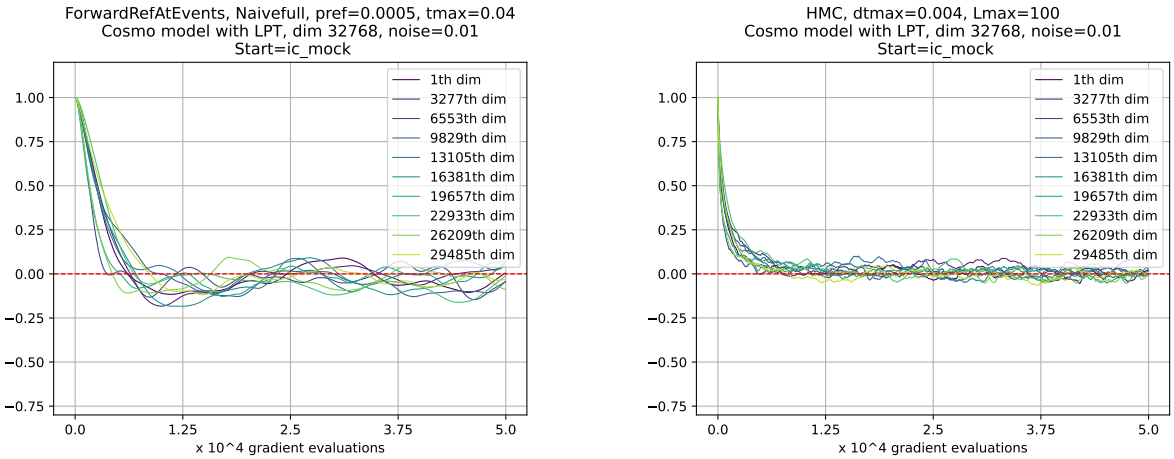
3.3.3 Numerical results

Starting close to the truth

We first present the results of experiments starting close to the value of the mock data. This is to test the relaxation performance of both algorithms and their capacity to efficiently explore the typical set of the distribution once they have found it. Our experiments show that both algorithms are comparable, HMC slightly outperforming PDMC. This is due to a combination of two factors: first, the numerical cost of thinning prevents the PDMC algorithm to outperform the HMC by a relevant factor. Second, by design, the HMC is able to explore the spherical typical set with great efficiency. So the dynamical advantage of PDMC may not visible on such target.

We first present some comparisons at **fixed computational budget**: both HMC and PDMC are run for 1 million gradient evaluations. We then compare the autocorrelations along a few dimensions to assess convergence. As can be seen from Figure 3.7, the HMC converges faster and decorrelates the samples more easily, leading to a higher Effective Sample Size. In this regime though, the limiting factor is clearly the number of gradient evaluations per event which equals ~ 4.5 for PDMC-BORG, preventing the latter to outperform HMC-BORG.

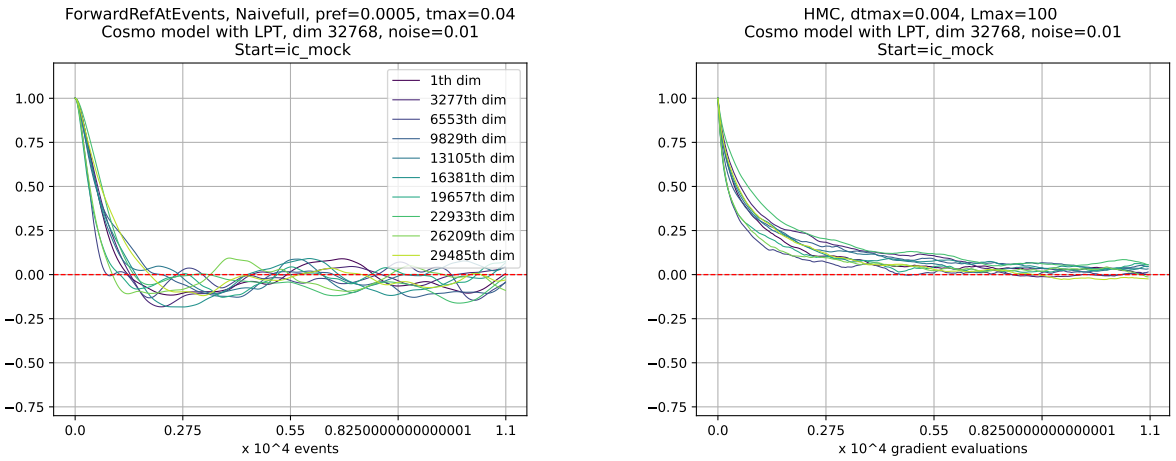
Another kind of comparison would be **in an ideal scenario** for the PDMC sampler where the computation of event times could be done analytically. In this case, the autocorrelation plots would be expressed in terms event number for PDMC-BORG and with the corresponding number of gradient evaluations for HMC-BORG. Such comparison should be used to compare the different dynamics of the algorithms. In this scenario, PDMC-BORG is slightly better at decorrelating the single dimensions, as illustrated in Figure 3.8.



(a) Autocorrelations with PDMC, 1M evaluations.

(b) Autocorrelations with HMC, 1M evaluations.

Figure 3.7: Autocorrelations along a few dimensions obtained with PDMC and a HMC, **at fixed computational budget**. The two plots were made from 20,000 samples, separated on average by ~ 50 gradient evaluations, corresponding to ~ 11 events for PDMC.



(a) Autocorrelations with PDMC, 220k events.

(b) Autocorrelations with HMC, 220k evaluations.

Figure 3.8: Comparisons of PDMC-BORG and HMC-BORG **in an ideal scenario** where the computation of event times could be done analytically for the PDMC sampler. The horizontal axis unit is the number of events for PDMC-BORG (Left) and the corresponding number of gradient evaluations for HMC-BORG (Right).

The **exploratory behavior** of the algorithms are assessed by plotting the first samples produced by the algorithms along the two dimensions with extremal variance. In this setting, dimension 24007 has the smallest variance, which is approximately equal to 3×10^{-3}. On the other hand, dimension 28848 has the biggest variance, which is approximately equal to 8×10^{-2}. The plots are compiled in Figure 3.9. Within 1 million gradient evaluations, it has explored the same region as the HMC with the same number of evaluations. It achieves this results with about 4.5 times less events than the HMC. The different dynamics are clearly visible on the plot.

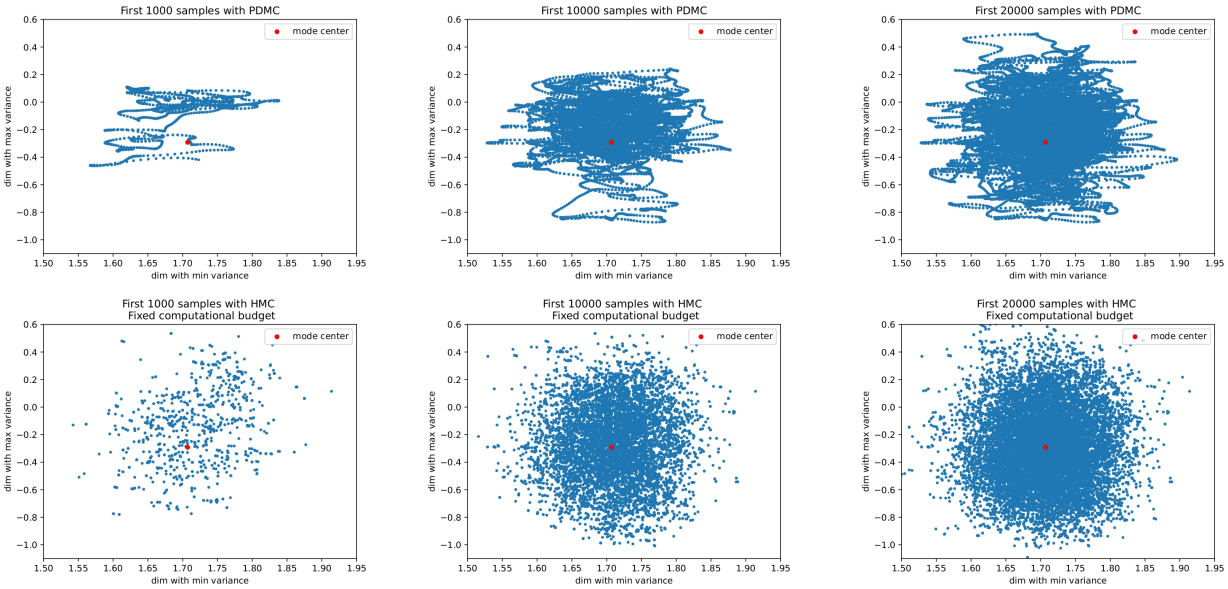
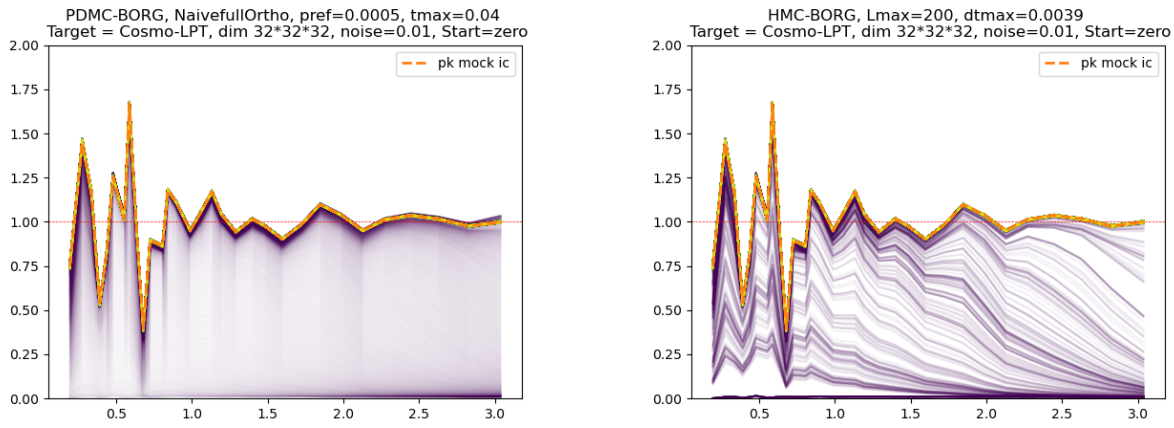


Figure 3.9: First 1,000 (left), 10,000 (middle) and 20,000 (right) samples produced by PDMC (top) and HMC (bottom), projected along the two dimensions with minimal and maximal variance. The baseline PDMC was run for 1 million gradient evaluations, which corresponds to 220,000 events. The HMC was run for 1 million gradient evaluations, which correspond to the same number of events. One can easily visualize the ballistic trajectory of the PDMC and the diffusive behavior of HMC. In the end, the two algorithms have explored the same region.



(a) Successive power spectra with PDMC-BORG.

(b) Successive power spectra with HMC-BORG.

Figure 3.10: Successive power spectra obtained with samples of PDMC-BORG and HMC-BORG starting close to 0. In the two plots, one can observe convergence around the region drawn by the power spectrum of the initial mock conditions.

Starting far from the truth

The last tests consist in comparing PDMC-BORG and HMC-BORG when starting outside the region of interest. By default, the BORG framework defines the starting position as the realization of a Normal distribution with small standard deviation 0.1. The resulting point in a d-dimensional parameters space is close to 0 which does not belong to the typical set. For large values of d and small noise regimes, one thus needs to use the warm-up strategy that has been discussed previously.

In the following experimental results, we use the same setting as before. A HMC and a PDMC are both run for 2 million gradient evaluations, starting from the same initial point in parameters space. Such position is drawn from a Normal distribution with mean 0 and covariance matrix $0.01 I_{32^3}$. In total, $20,000$ samples are generated with each algorithm. The hyperparameters for the warm-up phase lead to approximately 400 samples and $40,000$ gradient evaluations for both algorithms, making them directly comparable. No burn-in is considered, i.e. all the samples are taken into account for observables computations.

Figure 3.10 shows the successive power spectra, from purple to yellow. As expected, these spectra converge around the true power spectrum corresponding to the mock initial conditions. To assess convergence to the same observables, we also represent the cumulative histograms along a few dimensions in Figure 3.11. These plots look the same for PDMC and HMC, which is the desired behavior. Finally, we plot in Figure 3.12 the zoomed autocorrelations along a few dimensions obtained with the two algorithms. At fixed computational budget, they illustrate the similar performance between HMC-BORG and PDMC-BORG on the problem. Interestingly, when representing the autocorrelations in the ideal scenario described above, PDMC-BORG seems to outperform its HMC counterpart. This illustrates that the ballistic exploration of the high-dimensional parameters space is more efficient than a Hamiltonian exploration. In this kind of challenging high-dimensional Bayesian problem, non-reversible PDMC samplers may be a robust alternative to the usual reversible samplers that are popular among the cosmologists community.

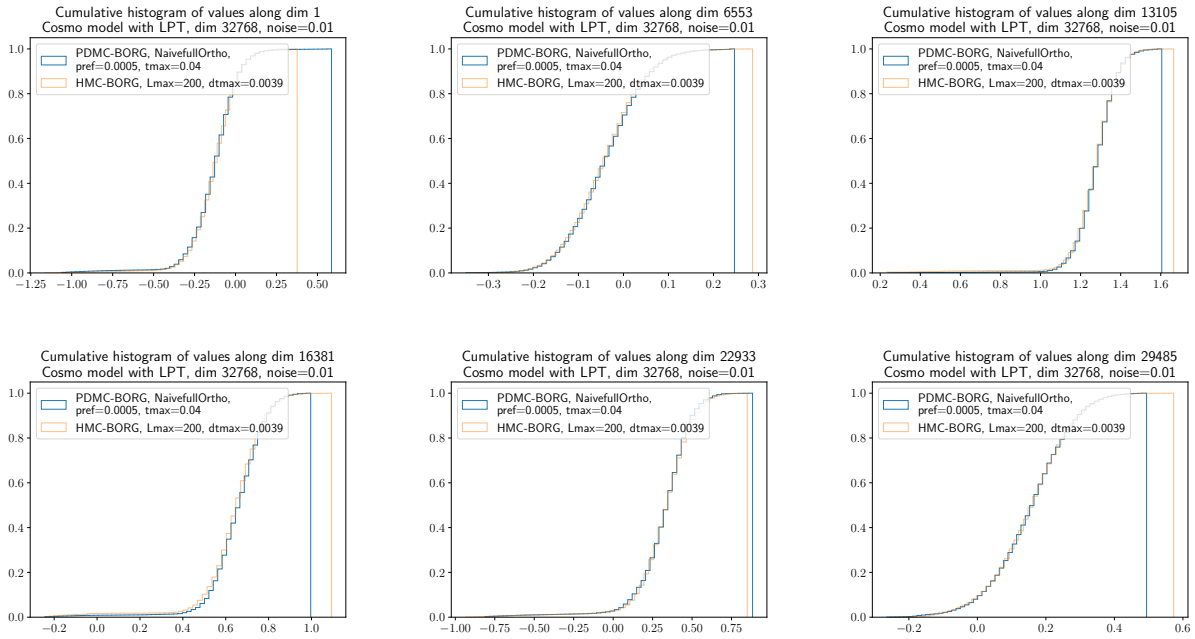


Figure 3.11: Cumulative histograms along a few dimensions obtained with samples generated by the two different algorithms starting close to 0.

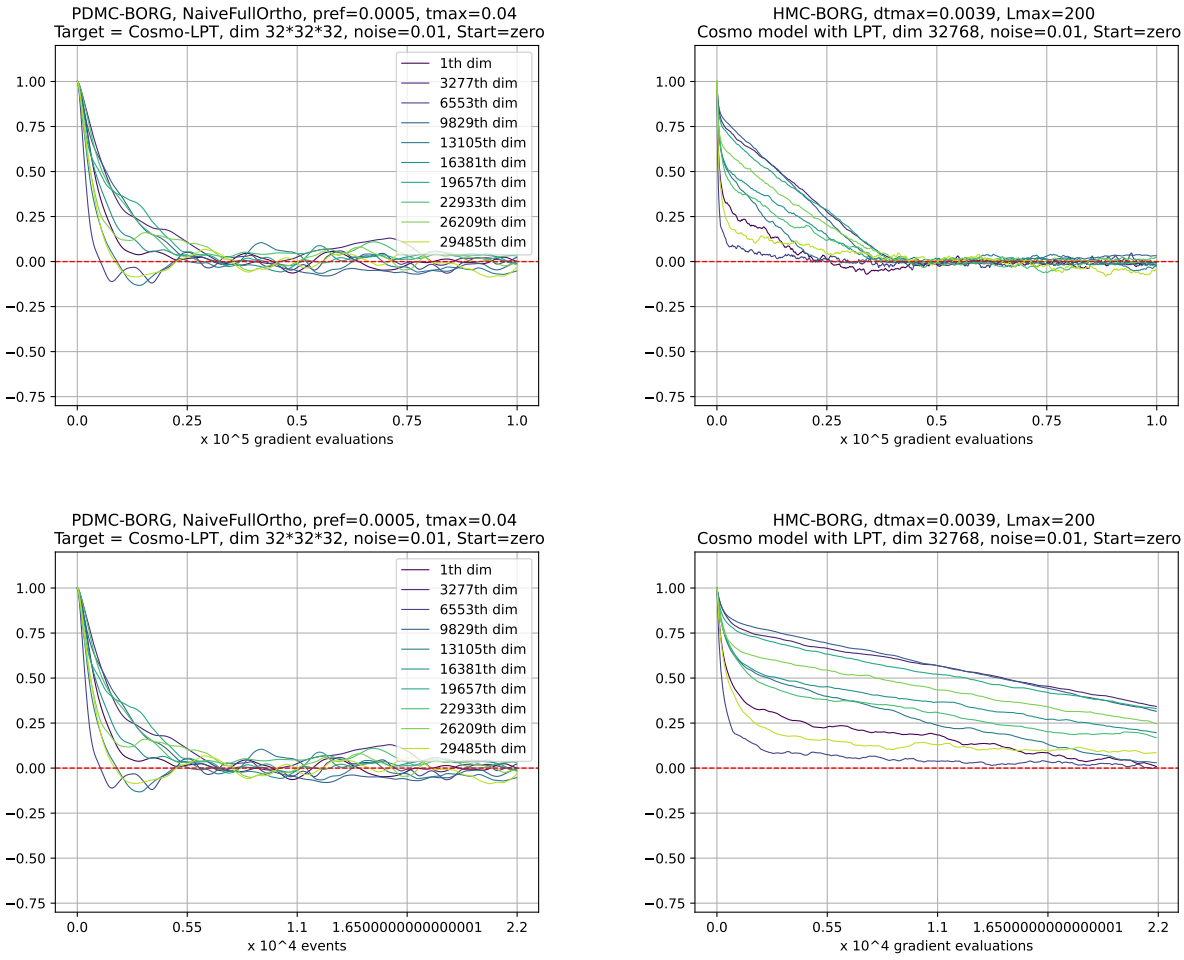


Figure 3.12: Autocorrelations along a few dimensions for PDMC-BORG (left) and HMC-BORG (right) at fixed computational budget (top) and in the ideal scenario (bottom).

Conclusion

We introduced PDMC-BORG, a non-reversible Monte Carlo sampler for large-scale structure inference. Incorporated within the BORG framework, it is straightforward to compare with a baseline HMC-BORG sampler. The results obtained in dimension 32^3 with a small noise level are promising: they illustrate that both PDMC and HMC are close in terms of numerical efficiency. Our comparisons in the ideal case where the event times could be computed analytically show that PDMC-BORG is able to outperform HMC-BORG as confirmed by autocorrelation plots along single dimensions. This is a promising result that could become more and more visible as the dimension increases and the noise level becomes smaller, leading to an anisotropic energy landscape where the ballistic trajectories of PDMC-BORG could turn into a decisive advantage in terms of exploration capacity.

Chapter 4

Fixed-kinetic Neural Hamiltonian Flows for enhanced interpretability and reduced complexity

> *For instance, on the planet Earth, man had always assumed that he was more intelligent than dolphins because he had achieved so much - the wheel, New York, wars and so on - whilst all the dolphins had ever done was muck about in the water having a good time. But conversely, the dolphins had always believed that they were far more intelligent than man - for precisely the same reasons.*
>
> Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

In this work, we investigate the use of Hamiltonian dynamics for building robust and interpretable generative models. We propose a fixed-kinetic version of Neural Hamiltonian Flows for improved robustness and reduced complexity of the model. We evaluate the sampling performance and robustness of such models in high-dimensional image generation problems. Finally, we test a framework for Bayesian inference using Neural Hamiltonian Flows and present numerical results from a cosmological inference problem. This project led to a publication and to a poster presentation at the AISTATS 2024 conference (Souveton et al., 2024). This chapter corresponds to the article enriched with additional comments.

4.1 Normalizing Flows with Hamiltonian transformations

4.1.1 Architecture and training of Neural Hamiltonian Flows

Theoretical motivations

Normalizing Flows consist in training a neural network to map a simple prior distribution onto the desired target through a chain of invertible transformations. They come with interesting characteristics, such as stability and correctness, see for example (Papamakarios et al., 2022). The main limitation comes from the design of an invertible function for the mapping. In particular, computing the Jacobian determinant in the change of variable formula may be costly. Also explainability is now under a growing concern within the community (Gilpin et al., 2018), in particular regarding applications in natural science, as the transformation learned by NF models is commonly hard to interpret.

First motivated by the mitigation of the Jacobian computation limitation, Neural Hamiltonian Flows (NHF) (Toth et al., 2020) are NF models that use Hamiltonian transformations. Indeed, in classical

Newtonian mechanics, the Hamiltonian of a system, composed of a kinetic and a potential energy terms, sets its dynamical evolution, which is reversible and has a Jacobian determinant equal to one. They have exhibited performance similar to RealNVPs in sampling some 2D distributions (Toth et al., 2020). Furthermore, being Physics-driven models, they are expected to enhance interpretability and it is furthermore straightforward to exploit the Hamiltonian properties to include some invariance under symmetrical transformations (Jimenez Rezende et al., 2019). This last assumption comes from a theoretical lemma stated in Jimenez Rezende et al. (2019) based on Noether's theorem Noether (1971): given a Hamiltonian H and symmetry generators g_k, the corresponding Hamiltonian flow $\mathcal{T}_H^{dt}$ pushing the base distribution π_0 onto π is invariant with respect to the generators if $\{g_k, H\} = \{g_k, \pi_0\} = 0$. In the latter expression, the brackets stand for the Poisson bracket. This sufficient condition can be imposed during training by adding a constrained optimization term in the loss function. Jimenez Rezende et al. (2019) exhibit an acceleration in training while using this method on target distributions having for example rotational invariance properties.

The building blocks of NHF come from the idea that learning Hamiltonians, i.e. physical conserved quantities, is a first step towards a better understanding of the physical processes that governs the data generation. Multiple architectures have been proposed in this direction, such as Hamiltonian Neural Networks (Greydanus et al., 2019) or Hamiltonian Generative Networks (Toth et al., 2020). These methods parameterize the Hamiltonian with neural networks and come with useful properties such as exact reversibility and smoothness. They have inspired applications from domain translation (Menier et al., 2022) to fault-detection in industry (Shen et al., 2023). Notably, they can be combined with Markov-Chain Monte Carlo (MCMC) methods, for instance as proposals in the Hamiltonian Monte Carlo (HMC) algorithm (Duane et al., 1987; Dhulipala et al., 2022). We learn here artificial Hamiltonians for sampling and our goal is to extract the negative logarithm of the target distribution into the potential.

In summary, Neural Hamiltonian Flows (NHF, Toth et al., 2020) are a special instance of generative models that use a series of Hamiltonian transformations as normalizing flows. Using Hamiltonian mappings, they come with desirable properties:

- they are invertible by construction and inversion is easy by using a classical numerical integrator, i.e. just reversing the timestep sign;
- their Jacobian determinant is equal to 1, removing the necessity to compute such determinant for each transformation;
- as Physics-inspired models, they have a potential of being highly-interpretable. Also, they may enable to exploit symmetries inside data for better training and improved overall performance (Jimenez Rezende et al., 2019).

Neural Hamiltonian Flows in practice

In practice, NHF is trained on a dataset consisting in realizations from the target distribution. To simulate a Hamiltonian dynamics, one must extend the position space in which live the samples into the phase-space, by adding artificial momenta: this is the role of the Encoder. The dynamics is integrated in phase-space with the Leapfrog integrator. More precisely, during training, NHF takes batches of $\mathbf{q}_T$ from the training dataset as inputs. For each $\mathbf{q}_T$, one $\mathbf{p}_T$ is drawn from a Gaussian distribution whose mean $\mu(\mathbf{q}_T)$ and deviation $\sigma(\mathbf{q}_T)$ depend on the $\mathbf{q}_T$. The resulting point in phase-space then evolves through a series of L Leapfrog steps with integration timestep $-\delta t$. The outputs consist in the final position $\mathbf{q}_0$ and momenta $\mathbf{p}_0$, as well as the initial mean $\mu(\mathbf{q}_T)$, deviation $\sigma(\mathbf{q}_T)$ and $\mathbf{p}_T$ used in the loss computation. Once trained, one can easily define a sampling function that transforms $\mathbf{q}_0$, $\mathbf{p}_0$ into $\mathbf{q}_T$, by changing the sign of integration timestep and moving

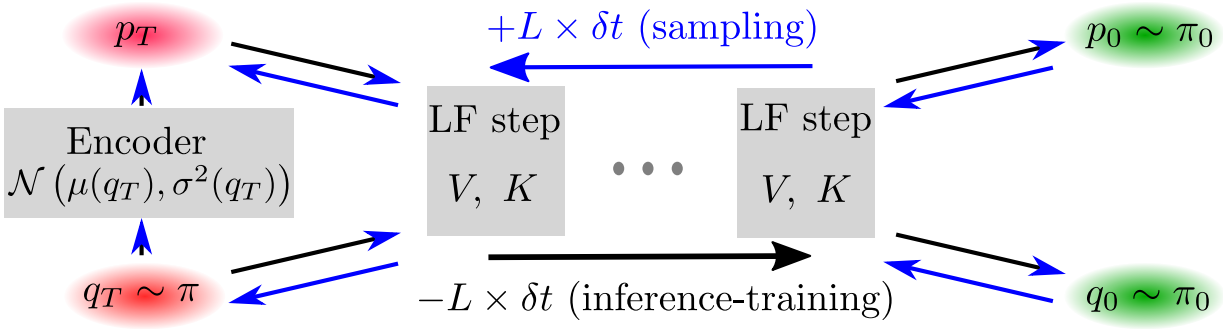


Figure 4.1: Schematic representation of the NHF architecture. In inference-training mode, it consists in starting from training dataset samples, identified as generalized positions, and using the Encoder part to generate artificial generalized momenta. Then, system evolves in phase-space following a Hamiltonian flow that is approximated with a Leapfrog scheme. Once trained, the resulting output in phase-space should follow the prior distribution. It is then possible to sample from the target starting with random samples drawn from the prior distribution and inverting the learned dynamics.

the system through the learned dynamics. An illustration of the architecture can be found in Figure 4.1.

Now regarding the training, following the previous notations, let $f(.|\mathbf{q}_T)$ be the density of a normal distribution $\mathcal{N}(\mu(\mathbf{q}_T), \sigma(\mathbf{q}_T)^2)$, and $\mathcal{T}^{-1}$ the backward transformation of phase-space performed by NHF i.e. $\mathcal{T}^{-1}(\mathbf{q}_T, \mathbf{p}_T) = (\mathbf{q}_0, \mathbf{p}_0)$. Denote Π_0 the joint distribution of $\mathbf{q}_0$, $\mathbf{p}_0$. By adding artificial momenta $\mathbf{p}_T$ (Toth et al., 2020), the distribution modeled by the NHF is $m(\mathbf{q}_T) = \int M(\mathbf{q}_T, \mathbf{p}_T) d\mathbf{p}_T = \int \Pi_0(\mathcal{T}^{-1}(\mathbf{q}_T, \mathbf{p}_T)) d\mathbf{p}_T$. Marginalizing over the momenta is crucial for this volume-preserving flow as it prevents highly non-desirable effects of models equipped with this property, such as the incapacity to create multimodality when using a unimodal base distribution (Draxler et al., 2024). This integral being intractable, one instead maximizes the following ELBO, derived using Jensen's inequality:

$$\begin{aligned}
\mathcal{L}(\mathbf{q}_T) &= \mathbb{E}_f \left[\log \Pi_0(\mathcal{T}^{-1}(\mathbf{q}_T, \mathbf{p}_T)) - \log f(\mathbf{p}_T|\mathbf{q}_T)\right] \\
&\leq \log \int \frac{M(\mathbf{q}_T, \mathbf{p}_T)}{f(\mathbf{p}_T|\mathbf{q}_T)} f(\mathbf{p}_T|\mathbf{q}_T) d\mathbf{p}_T \\
&= m(\mathbf{q}_T).
\end{aligned}$$

This quantity is approximated via Monte Carlo integration. Having learned the transformation, one can reverse the sign of timesteps and use the same potentials to transform the prior distribution into the target distribution. Further improvements can be made regarding the target distribution. In particular, if one seeks to learn symmetries regarding a certain group with generators $g_1, ..., g_K$, then the new loss function is derived as (Jimenez Rezende et al., 2019):

$$\mathcal{L}_{\text{tot}} = \sum_{\mathbf{q}_T \in \text{data}} \mathcal{L}(\mathbf{q}_T) + \sum_{k=1}^{K} \lambda_k \mathbb{E}_{\Pi_0} \left[\{g_k, H\}^2 - \kappa\right].$$

In the above expression, λ_k is the Lagrange multiplier associated to the k-th generator and $\kappa \geq 0$ is the symmetry constraint precision. One may see that if $\kappa = 0$ and the learned mapping is invariant with respect to the desired symmetry group, then the second sum in the right-handed side of the expression will be equal to 0.

To summarize, the first part of the architecture consists in adding artificial momenta, as done by the Encoder, to simulate a Hamiltonian dynamics. Here, μ and σ are approximated by two neural networks. As for the Hamiltonian transformation, it is simulated with a chain of symplectic Leapfrog steps. To do so, one must design the potential energy V and the kinetic energy K of the system.

In Toth et al. (2020), each energy term is parameterized by a neural network. We will discuss this choice in the following section. For now, let us highlight that integrating Hamilton's equations with a symplectic numerical scheme provides flexibility. Most NF architectures rely on a careful architecture design rendering the computation of the Jacobian determinant easy (Dinh et al., 2017). This is not the case with NHF since invertibility and volume-conservation are ensured by the use of a Leapfrog integrator and do not depend on the neural networks that are used to parameterize μ, σ, V and K. The latter aspect also means that the choice of neural networks within the NHF architecture (MLPs, CNNs, etc.) are not guided by usual considerations such as the groups of symmetries one tries to preserve (Bronstein et al., 2021). In this framework, learning equivariant mappings with respect to some symmetry group arises from the use of Hamiltonian transformations (Jimenez Rezende et al., 2019) and can be further enforced by a straightforward modification of the loss function.

Link with Neural ODEs

Another way to look at Neural Hamiltonian Flows is from the perspective of Neural Ordinary Differential Equations (Chen et al., 2018). Compared to the transformation in Neural ODE flow, the Hamiltonian ODE in NHF are volume-preserving, making for a cheaper log-likelihood computation, and can be integrated via symplectic integrators. Let us call $\mathbf{z} = (\mathbf{q}, \mathbf{p})$ the points in phase-space. Integrating Hamiltonian's dynamics boils down to solving the following initial condition ODE:

$$\begin{cases} \frac{d\mathbf{z}}{dt} = J\nabla H(\mathbf{z}) \\ \mathbf{z}(0) = \mathbf{z}_0 \in \mathbb{R}^{2d} \end{cases}$$

The idea of NHF is to parameterize $f(\mathbf{z}) \overset{\text{def}}{=} J\nabla H(\mathbf{z})$ by neural networks, respectively the two neural networks that represent the kinetic and potential energies. In this case, NHF becomes a solver of the above system for the Hamiltonian defined by the energetic landscape of the target distribution. Indeed,

$$\mathbf{z}(T) = \mathbf{z}_0 + \int_0^T J\nabla H(\mathbf{z}(t))dt = \text{NHF}(\mathbf{z}_0).$$

Note that NHF is trained at fixed integration time $T = L \times \delta t$. However, after sufficient training, the model has learned the energetic landscape associated to the target distribution. In theory, it is thus able to simulate Hamiltonian trajectories through this landscape and integration time can be made as big (or small) as one wants. This is another way of illustrating the model robustness since its efficiency will no longer depend on the model hyperparameters. In practice though, as there might be biased due to insufficient training, this aspect needs to be tempered.

4.1.2 Designing new versions of NHF

The NHF architecture is made of four neural networks black-boxes that may render difficult the interpretation of the learned dynamics and energies. In particular, the learnt potential energy is not guaranteed to correspond to the corresponding physical potential energy of the data, when writing their probability distribution as a Boltzmann one. Even if it was numerically shown to transfer multimodality from the target distribution to the potential energy in some 2D cases (Toth et al., 2020), this property is not ensured. We focus here precisely on the transfer of the negative logarithm of the target distribution into the learned potential, which should be the case in any physical systems. This kind of motivations comes from the field of Explainable Artificial Intelligence (XAI) which deals with the problem of understanding the decisions made by an Artificial Intelligence (Samek and Müller, 2019). Indeed, complex architectures made of multiple (deep) neural network are often easier to train than to understand. Some solutions involve surrogate techniques (Ribeiro et al., 2016), local perturbations (Ancona et al., 2022) or meta-explanations (Lapuschkin et al., 2019). Including physical prior knowledge into neural networks may be another solution to understand the model (Raissi et al., 2019; Toth

et al., 2020). In this work, we build on that idea and try to make the model as explainable as possible by fixing its kinetic energy and thus enforcing a classical Mechanics knowledge into the architecture.

MLPK-NHF. If the kinetic energy is chosen to be a MLP (Toth et al., 2020), then the model contains two black-boxes that are not easy to interpret *a priori*, namely kinetic and potential energies K and V. In particular, when sampling a multimodal distribution from a unimodal prior, the learnt potential V may not reflect the multimodal distribution.

FK-NHF. By fixing the kinetic energy inside NHF, we gain interpretability on the learned flow by forcing the latter to obey some Physics principles. In this model, K is no longer a MLP but a quadratic function $K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathcal{M}^{-1}\mathbf{p}$, with $\mathcal{M}$ a symmetric positive matrix. Starting from $(\mathbf{q}_0, \mathbf{p}_0)$ drawn from a unimodal prior distribution and imposing a quadratic kinetic energy significantly reduces the possibilities for the potential energy to recover a multimodal $\mathbf{q}_T$. We indeed aim at enforcing these energies to be classical from a Physics perspective, i.e. making the learned kinetic energy to be of a quadratic form and the learned potential to be the negative logarithm of the target distribution $-\log\pi$ (or an approximation), as it is the case for diffusion models. It is noteworthy that by Liouville theorem and the ergodic hypothesis, if the learnt potential is $-\log\pi$ and K of a classical form, then any initial distribution of $\mathbf{q}_0$ can be mapped to the distribution π of the $\mathbf{q}_T$ given the distribution of $\mathbf{p}_0$ is rich enough (e.g. a Normal law). This is also at the basis of the HMC method (Duane et al., 1987; Neal, 2012): in this setting, Hamiltonian dynamics with momenta refreshment yields an ergodic exploration of phase-space that leaves the canonical distribution invariant. Canonical distribution invariance uses volume-preservation in phase-space, i.e. Liouville's theorem, and ergodicity comes from momenta refreshment. Therefore the FK variant can learn any distribution of $\mathbf{q}_T$.

Encoder-free NHF. In addition to the previous choices for the kinetic energy, one has flexibility regarding the way auxiliary momenta are drawn. In Toth et al. (2020), they are generated as the realization of a Normal distribution $\mathcal{N}(\mu(\mathbf{q}_T), \sigma^2(\mathbf{q}_T))$. Reducing the numerical cost of the model can be achieved by removing the neural networks that parameterize μ and σ. Such choice leads to the design of *Encoder-free* models.

Keeping track of the transformation dynamics, and the analogy with the familiar classical mechanics framework, makes possible the interpretability of the learned potential. Also, learning the energy landscape associated to the target distributions offers guarantees in terms of control of the discretization scheme, avoiding chaotic behaviour and making the model less sensitive to the choice of Leapfrog hyperparameters as we show in Section 4.2. By doing so, both interpretability, sparsity and robustness are gained through FK-NHF. It is possible to create different versions of NHF by fixing its kinetic energy. One could, for instance, use a relativistic kinetic energy instead of a classical one. We now numerically show how the classical choice for kinetic energy yields an interpretable potential V and such in a robust manner.

4.1.3 Metrics for evaluating sampling quality

Divergence between two probability measures

A natural (pseudo-)distance is the KL-divergence $D_{KL}(\pi||m)$ between the true target distribution with density π and the model distribution with density m. This pseudo-distance quantifies the loss of information when using the model distribution instead of the true target for describing the data - so the lower the better. The KL-divergence is defined as

$$D_{KL}(\pi||m) = \int \pi(x) \ln \frac{\pi(x)}{m(x)} dx \geq 0.$$

When one of these probability distributions is intractable, as it is the case in our setting, we estimate D_{KL} by comparing samples from the true target distribution with samples from the model distribution.

The procedure described in (Perez-Cruz, 2008) proceeds according to a k-neighborhood density estimate of the two distributions:

$$D_{KL}(\pi||m) \approx -\frac{D}{S}\sum_{s=1}^{S}\ln\frac{r_k(X_s)}{s_k(X_s)} + \ln\frac{S}{S-1}$$

where $r_k(X_s)$ is the k-th closest neighbor of X_s in $\{X_s\}_{s=1}^{S}\backslash\{X_s\}$ and $s_k(X_s)$ is the k-th closest neighbor of X_s in $\{X_s^0\}_{s=1}^{S}$. Parameters chosen in our estimation are $S = 1024$ and $k = 1$.

Evaluating images quality

For evaluating the quality of images generated by our models, we compute the number of bits per pixel, decaying as the quality of image increases. We start by a pre-processing step. First, pixels of training images are dequantized by adding uniform noise $\varepsilon \sim \mathcal{U}]0,1[$ and ranging them back to interval $[0,1]$ as $x \leftarrow (255x + \varepsilon)/256$. Then, our models are trained on the target distribution in logit space by transforming the resulting noisy pixels as $x \leftarrow \text{logit}((1-2\lambda)x + \lambda)$ with $\lambda = 10^{-6}$.
Once trained, we evaluate the number of bits per pixel of a pre-processed image $\tilde{X}_s^0 := (\tilde{X}_{s,1}^0, ..., \tilde{X}_{s,D}^0)$ in logit space from the test dataset following (Papamakarios et al., 2017):

$$b(\tilde{X}_s^0) = -\frac{\ln m(\tilde{X}_s^0)}{D\ln 2} - \log_2(1-2\lambda) + \frac{1}{D}\sum_{d=1}^{D}\left[\log_2(\text{logit}(\tilde{X}_{s,d}^0)) + \log_2(1-\text{logit}(\tilde{X}_{s,d}^0))\right].$$

In the above equation, we evaluate the probability distribution in logit space of the model, namely $m(\tilde{X}_s^0)$. For a classical flow-based model, it is straightforward since $m(\tilde{X}_s^0) = \pi_0(T^{-1}(\tilde{X}_s^0)) \times \left|\text{Jac}_{T^{-1}}(\tilde{X}_s^0)\right|$ where T^{-1} is the transformation from logit space to latent space learned by the model.
However, for NHF, the change of variable formula is only valid in phase-space for the model joint distribution of the positions and momenta: $M(\tilde{X}_s^0, V_s) = \Pi_0(T^{-1}(\tilde{X}_s^0, V_s)) \times 1$. We use a Monte Carlo approximation of $m(\tilde{X}_s^0)$ by drawing N momenta $V_{s,1}, ..., V_{s,N}$ from the Gaussian distribution $f(.|\tilde{X}_s^0)$ parameterized by the Encoder as:

$$m(\tilde{X}_s^0) = \int M(\tilde{X}_s^0, V_s)dV_s \approx \frac{1}{N}\sum_{i=1}^{N}\frac{M(\tilde{X}_s^0, V_{s,i})}{f(V_{s,i}|\tilde{X}_s^0)}.$$

For evaluating the number of bits per pixel of a model on the two MNIST datasets, we average the bits per pixel values obtained with 1024 images from the test dataset, using $N = 10$ for NHF.

4.2 Testing interpretability and robustness

We present the results of numerical experiments for sampling a 2D Gaussian mixture with 9 modes (see Figure 4.2). Such example, similarly studied in (Toth et al., 2020), enables to understand important aspects of NHF, like sensitivity to the choice of hyperparameters and, more importantly, interpretability. Also, traditional generative models like GANs may suffer from mode-collapse problems even in simple multimodal 2D settings (Eghbal-zadeh et al., 2019), mode-collapses which were never observed with NHF experiments. We tested four different NHF models:

- MLP-kinetic NHF with Encoder: μ and σ are MLPs with size $(2, N, N, 2)$, V and K are MLPs with size $(2, N, N, 1)$. According to the experiments, we used $N = 8,\ 32,\ 128$.
- Fixed-kinetic NHF with Encoder: μ and σ are MLPs with size $(2, N, N, 2)$, V is a MLP with size $(2, N, N, 1)$. Kinetic energy K is a positive quadratic form whose mass matrix is learned during training. According to the experiments, we used $N = 8,\ 32,\ 128$.

- MLP-kinetic NHF without Encoder: we use this model for experiments in Section 4.2.2. Artificial momenta $\mathbf{p}_T$ are drawn from a $\mathcal{N}(0, C)$ where $C = \mathrm{Diag}(s_1^2, s_2^2)$, s_1, s_2 being learned during training. V and K are MLPs with size $(2, 156, 156, 1)$. The number of neurons per hidden layer was chosen so that the resulting model has about the same number of parameters as an Encoder-based Fixed-kinetic NHF with $N = 128$.

- Fixed-kinetic NHF without Encoder: this model is considered in Section 4.2.2. Artificial momenta $\mathbf{p}_T$ are drawn from a $\mathcal{N}(0, C)$ where $C = \mathrm{Diag}(s_1^2, s_2^2)$, s_1, s_2 being learned during training. V is a MLP with size $(2, 220, 220, 1)$. Kinetic energy K is a positive quadratic form whose mass matrix is learned during training. The number of neurons per hidden layer was chosen so that the resulting model has about the same number of parameters as an Encoder-based Fixed-kinetic NHF with $N = 128$.

We used Softplus activation functions in between hidden layers. All models were trained on a 5,000 points dataset with minibatches of size 512. Weights and biases were optimized with the Adam algorithm (Kingma and Ba, 2015), setting the learning rate to 5×10^{-4}. Our experiments were run on a HPC cluster, each of them using one GPU.

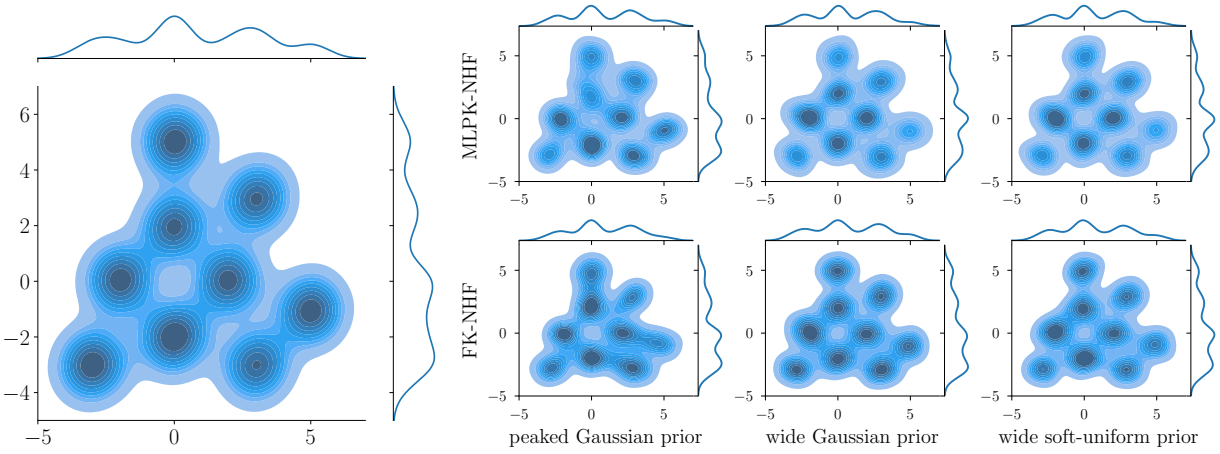


Figure 4.2: Density estimation with its marginals of (Left) the target 2D multimodal distribution (9 equally-weighted Gaussians with same covariance matrix $0.5^2 I_2$) and (Right) of the samples produced by two different NHF models with various choices of prior.

4.2.1 Impact of Leapfrog-hyperparameters and model complexity

Let us discuss the effect of Leapfrog-hyperparameters L (number of Leapfrog steps) and $T = L \times \delta t$ (integration time) on the optimization, but also the impact of the model complexity. The latter is governed by the total number of neurons in the model, this number being an increasing function of N, the number of neurons per hidden layer in each MLP of the model. If the model is complex enough, we expect to learn how to adjust to the number of Leapfrog steps and choice of integration time. If not, the model may have better performance by increasing the number of steps, i.e. increasing L.

We tested both FK-NHF and MLPK-NHF with various choices of L, T and N and a soft-uniform prior $\propto\ s(x+3)s(-x+3)$, where s is the sigmoid function. The corresponding loss decays are illustrated in Figure 4.3.
First, FK-NHF is more robust than the MLPK one to the choices of L and T, at fixed N, as discrepancy in the loss decay more clearly appears especially with $N = 8$. Then, regarding the tuning of the Leapfrog scheme, at fixed-integration time, models with $L = 1$ always reach higher final value of the loss, this effect being less visible with FK-NHF. Increasing the number of leapfrog steps leads to better final performance even if the effect disappears once the number of Leapfrog steps gets sufficient and no

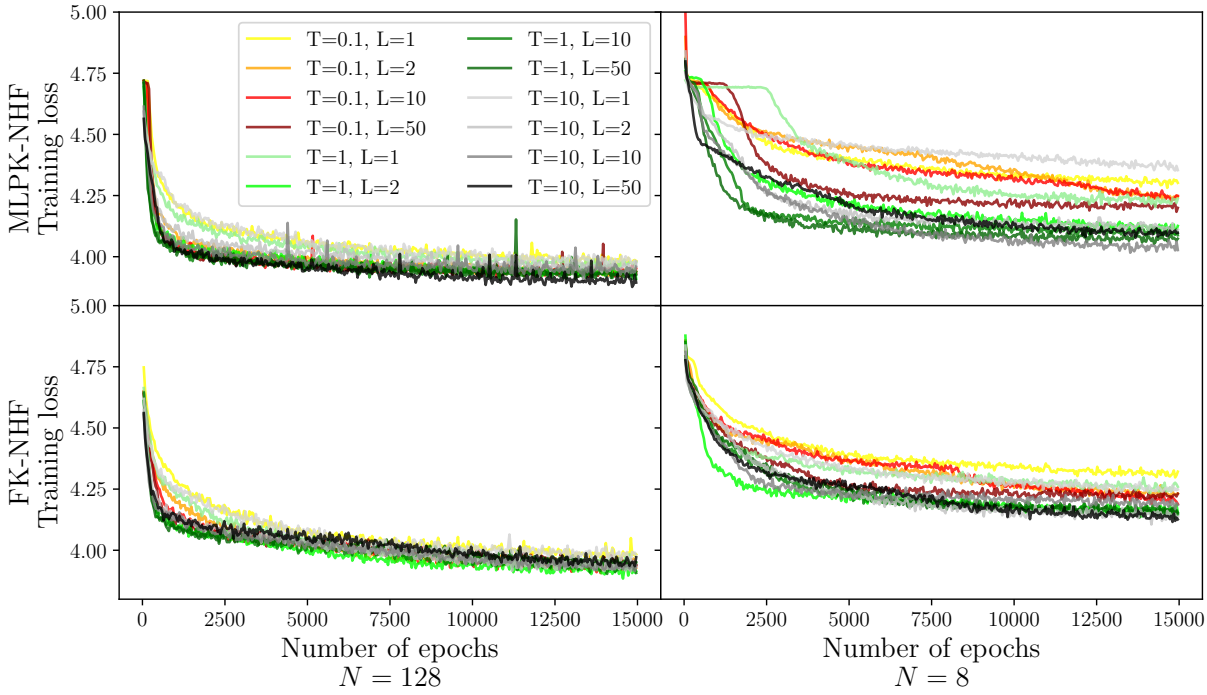


Figure 4.3: Training loss as a function of epochs for models with different N (number of neurons per hidden layer in each neural network of the model), L (number of Leapfrog steps) and T (integration time).

further expressivity can be achieved. Finally, as for the effect of integration time T, it barely appears for FK-NHF, showing that the latter efficiently adjusts to this parameter. As for the MLPK-NHF, the effect of the integration time is clearer, but mostly at $N = 8$, where performance improves for $T = 1, 10$ compared to $T = 0.1$. Overall, as the number of parameters in the model is increasing, the impact of the integration time becomes limited.

Thus, there are four hyperparameters that require tuning: three are usual in learning (minibatch size, learning rate and number of neurons per hidden layer, i.e. number of learning parameters of the model) and only one is specific to NHF: the number of Leapfrog steps, whose tuning is less sensitive when using FK-NHF. Furthermore, compared to diffusion models (Sohl-Dickstein et al., 2015), the required amount of steps is quite low.

4.2.2 Impact of the prior distribution on the learned dynamics

We illustrate the impact of the prior choice on the transfer of characteristics of the target distribution on the potential V, especially regarding the multimodality nature. All models were trained for 15,000 epochs using $N = 128$, $T = 1$ and $L = 10$, with a 5,000 points training dataset and with the soft-uniform prior, a peaker Gaussian prior $\mathcal{N}(0, I_2)$ and a wide Gaussian prior $\mathcal{N}(0, 2.5^2 I_2)$.

All considered schemes recover the nine correct modes from the target distribution, as illustrated in Figure 4.2. We now consider the learned potential V. As the Hamiltonian evolution only involves its derivative, we represented a shifted version in Figure 4.4. Choosing a relatively flat soft-uniform prior distribution that covers the target region, multimodality transfers to the potential energy for both FK-NHF and MLPK-NHF. The potential exhibits local extrema centered at the modes of the target, which can either be minima or maxima for the MLPK-NHF but are minima for the FK one. Indeed, with a MLPK model, the orientation of the learned energies may change from one numerical experiment to another, as we do not enforce the positiveness of the output of V and K. Similar results

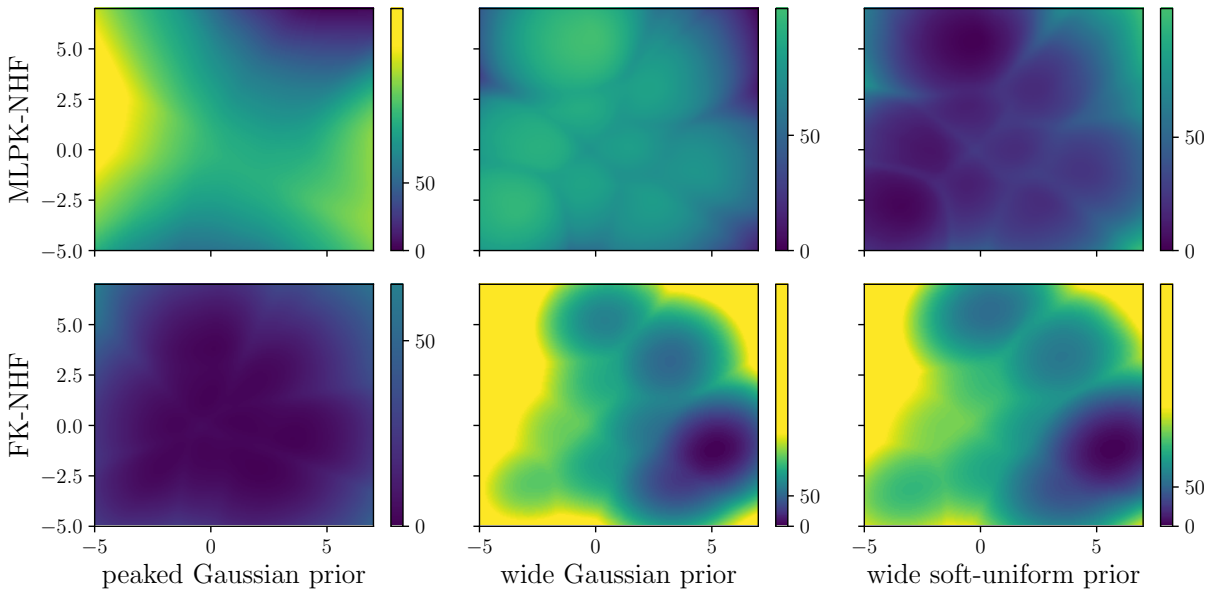


Figure 4.4: Shifted potential energies learned by the six models previously defined.

were obtained using the wide Gaussian prior with variance large enough to cover the support of the target distribution, which stresses the impact of the spatial expansion rather than the nature of the prior distribution.

On the other hand, with a "peaked" prior distribution $\mathcal{N}(0, I_2)$ for the MLPK-NHF, the momenta $\mathbf{p}_T$ generated by the Encoder inherit from the multimodality of the target distribution, with the same number of modes (see Figure 4.5). The learned energies are then different from the classical Physical ones and differs from one model to another. In the case of FK-NHF, multimodality is transferred to the potential energy, showing the robustness of the model to the choice of prior distribution.

Thus, using FK-NHF allows to more robustly transfer important properties of the target distribution into the learned potential. More specifically, the model is able to learn an interpretable potential with extrema centered at the modes of the data. When the learned potential is not multimodal, it is an indication that multimodality has been transferred instead to the artificial momenta $\mathbf{p}_T$ generated by the Encoder.

Finally, learning a potential approximating $-\log \pi$ is interesting in terms of interpretability but also renders the model more robust to hyperparameters. Figures 4.6 and 4.7 illustrate how learning an interpretable multimodal potential makes the model less sensitive to the choice of Leapfrog steps and robust to some dynamics extrapolation.

As shown in Figure 4.6, we also investigated the possibility of enforcing the transfer of multimodality to V by removing the Encoder and having $\mathbf{p}_T$ drawn from a $\mathcal{N}(0, s^2 I)$, s being learned during training. Such models are efficient at recovering a multimodal potential as illustrated in Figure 4.8.
While it improves MLPK-NHF for a peaked prior, we find it is less efficient and directly interpretable than fixing K. When fixing K, an Encoder-free model comes with a reduced complexity but we find that leaving as much flexibility as possible in the generation of momenta is the relevant option, especially for challenging problems and a small number of leapfrog steps.

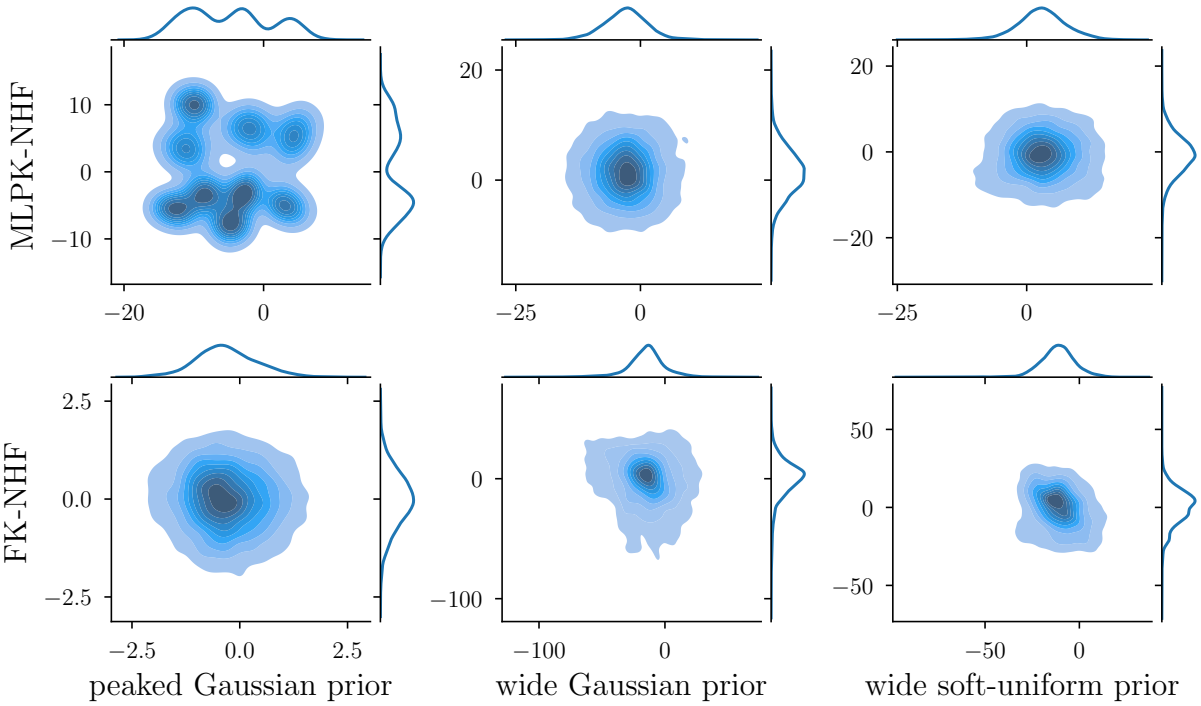


Figure 4.5: Density estimation of the artificial momenta from Encoder for the six models previously defined.

4.3 NHF for image generation

4.3.1 Numerical results and comparisons with a baseline Real NVP

To the best of our knowledge, NHF models have not been tested on high-dimensional image generation problems. We run experiments for sampling the MNIST handwritten digits (Deng, 2012) dataset, as well as additional tests on the Fashion MNIST dataset (Xiao et al., 2017). FK-NHF and MLPK-NHF are compared to another flow-based architecure, namely RealNVP (Dinh et al., 2017) with a similar number of learnable parameters. Our models were tested on the MNIST handwritten digits and Fashion MNIST datasets, which contain 60,000 images of size 28×28. The energy functions within the MLP-kinetic and Fixed-kinetic NHF are parameterized by 3-hidden layer MLPs with size $(784, 512, 256, 128, 1)$. As for μ and σ, they are 3-hidden layer MLPs with size $(784, 256, 256, 256, 784)$. We use LeakyReLU activation functions in between hidden layers with slope 0.1 for μ and σ and Softplus activation functions in between hidden layers for the energies. For the Fixed-Kinetic model, the mass matrix is optimized on the fly during training by learning its Cholesky decomposition, which has $D(D+1)/2$ parameters, D being the dimension of data, as was done in (Celledoni et al., 2023). This represents a grand total of 1.94 million learnable parameters for Fixed-kinetic NHF and 2.20 million for MLP-Kinetic NHF, that both use 10 Leapfrog steps with integration timestep $dt = 0.1$. As for the Real NVP to which they are compared, it uses 6 coupling layers and 32 planes for a total of 2.27 million learnable parameters. We adapted an architecture from a public open source GitHub project available at `https://github.com/bjlkeng/sandbox/tree/master/realnvp` under MIT Licence. All models were trained using a $\mathcal{N}(0, I_{784})$ base distribution, except for the NHF models on the MNIST handwritten digits which use a $\mathcal{N}(0, 2^2 I_{784})$. All models were trained for 50 epochs on minibatches with size 32 and optimization was performed using the Adam algorithm (Kingma and Ba, 2015). Our experiments were run on a HPC cluster, each of them using one GPU. We use a pre-processing step prescribed in (Dinh et al., 2017) consisting of learning the dequantized target distribution in logit space. The choice of mass matrix for FK-NHF is important for expressivity. This behaviour should be familiar to the HMC community (Duane et al., 1987) for which an optimal mass matrix is important

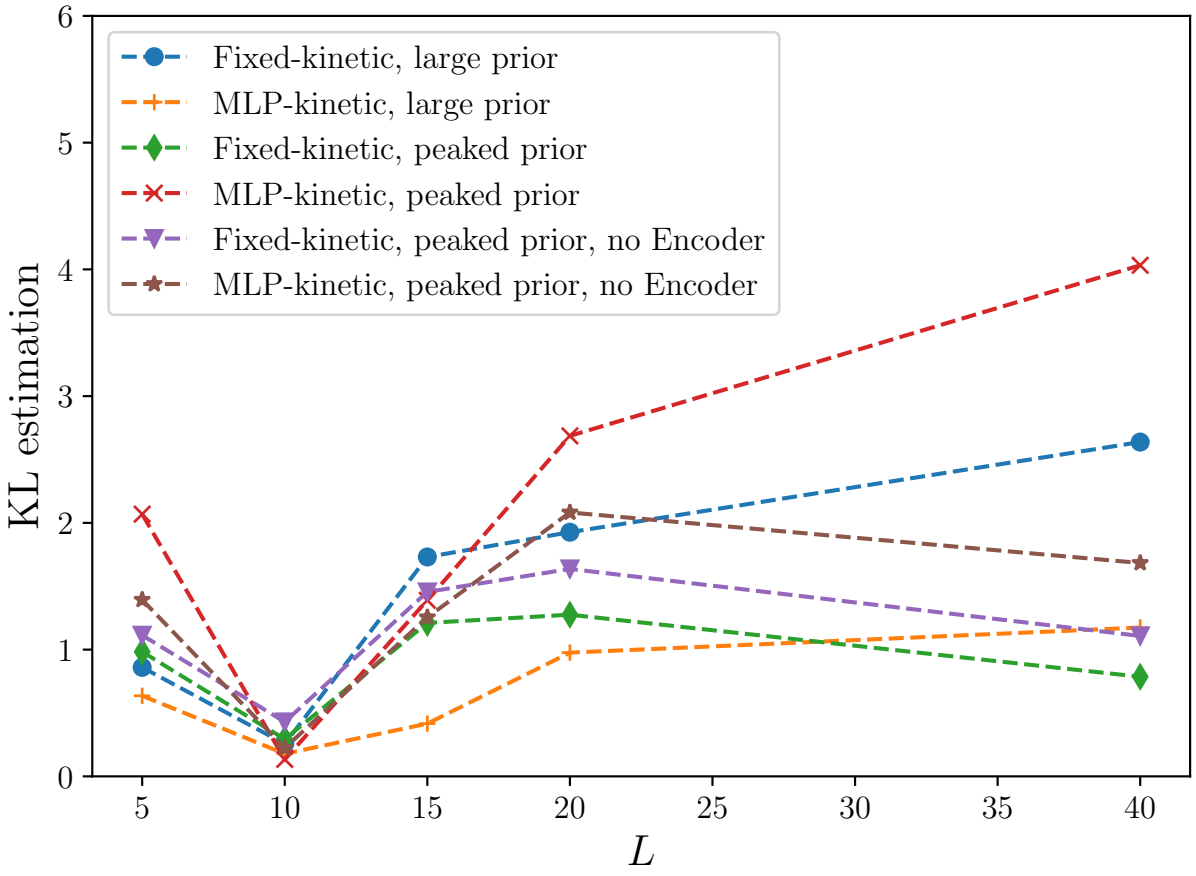


Figure 4.6: Evolution of KL estimation between the true and the model distribution as L increases, for models trained with $L = 10$. MLPK-NHF models with Encoder have $\sim 70,000$ learnable parameters while all the others have $\sim 50,000$ learnable parameters.

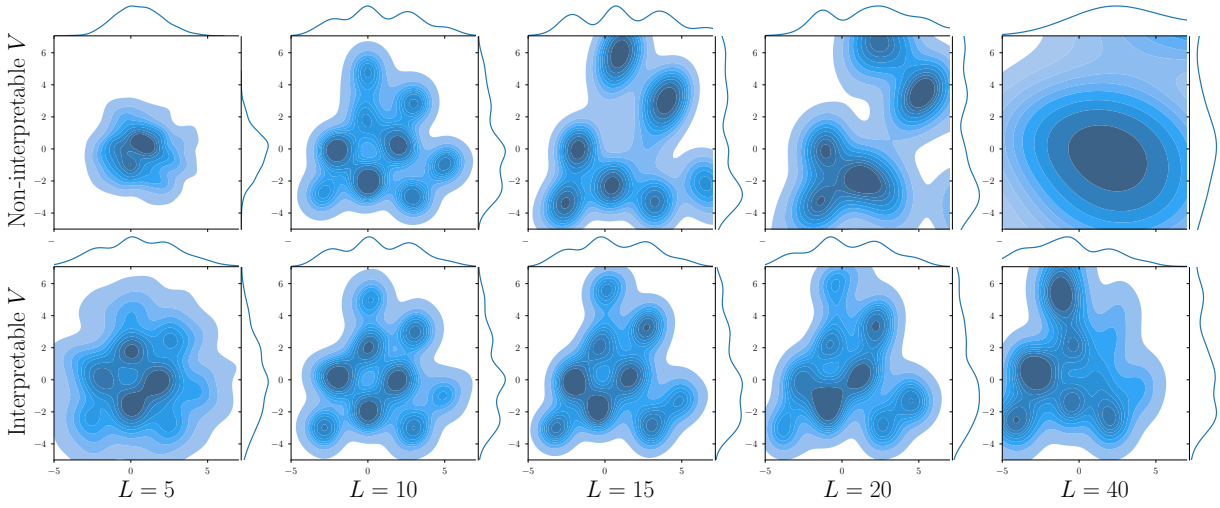


Figure 4.7: Density estimations of samples generated by MLPK-NHF, one that has learned a non-interpretable potential, and one that has learned an interpretable potential, trained with $L = 10$, as L increases.

for efficient exploration (Neal, 2012).

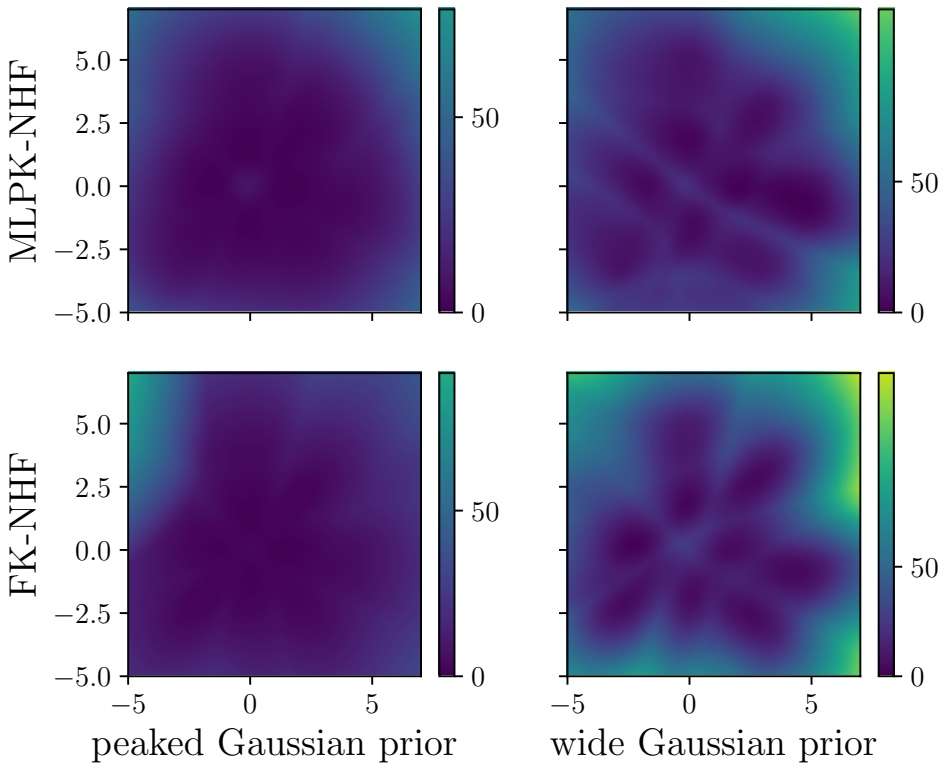


Figure 4.8: Shifted potential energy learned by four different Encoder-free models. All of them have recovered the 9 correct modes of the data which correspond to local minima of the potential.

Quality of sampling is quantitatively assessed by the KL divergence, estimated following (Perez-Cruz, 2008), and the number of bits per pixel (Papamakarios et al., 2017). Our experiments (Figure 4.9 and Appendices) show that RealNVP and NHF slightly outperform each other depending on the metric. The performance of FK-NHF is achieved with a simpler architecture though.

4.3.2 High-dimensional interpretability

Furthermore, the learned potentials have extrema located at the modes of the target distribution, see Figure 4.10, underlying interpretability and robustness to extrapolation of the number of Leapfrog steps. As can be seen, the fixed-kinetic model has learned local minima at the modes of data while the usual MLP-kinetic model has learned local maxima. Imposing a positive quadratic term for the kinetic energy ensures that it will always be minima.
Again, by learning an interpretable potential we observe that the model is less sensitive to the number of Leapfrog steps. This is particularly visible for MLP-kinetic NHF on both MNIST handwritten digits and Fashion MNIST datasets, as well as for FK-NHF especially on Fashion MNIST, see Figure 4.11.

4.3.3 Comparisons with diffusion models

Finally, we thought that it would be relevant to compare an ODE-driven flow-based models, such as NHF, with SDE-driven diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020), which are state of the art in image generation problems. The FK-NHF we present can however be understood as some ODE counterpart of diffusion models where the transformation is governed by a Langevin SDE. Both transformations agree for one discrete time step and this situation is reminiscent from the one in sampling with HMC (Neal, 2012) and Metropolis-adjusted Langevin algotihm (Rossky et al., 1978). We primarily focused on the numerical complexity needed for producing good samples. Our experiments, were run with an architecture which is an implementation of the latter paper based on an open-source

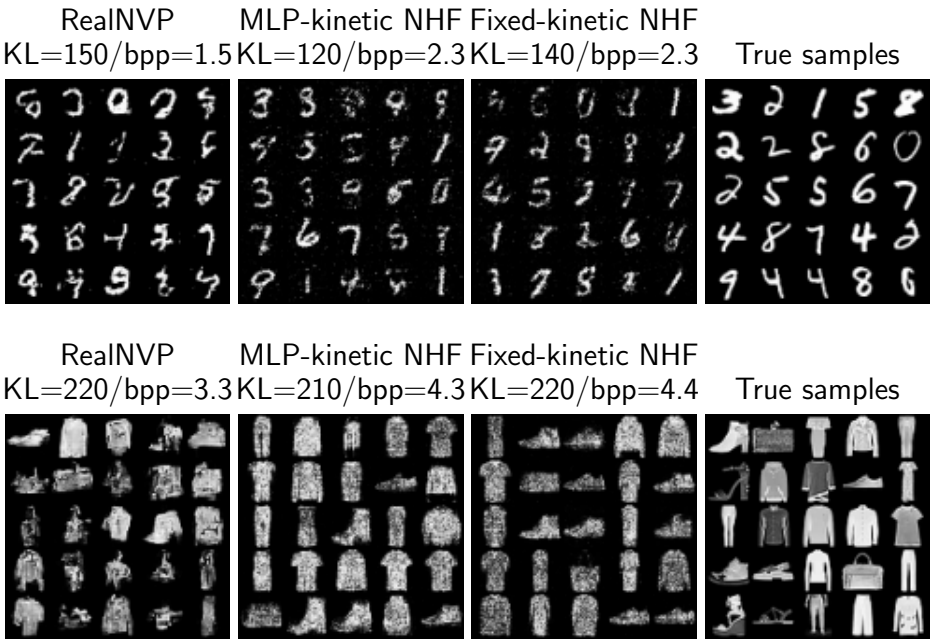


Figure 4.9: Samples produced after training on the MNIST and Fashion-MNIST datasets along with an estimation of the KL divergence between the true and model distributions and the number of bits per pixel.

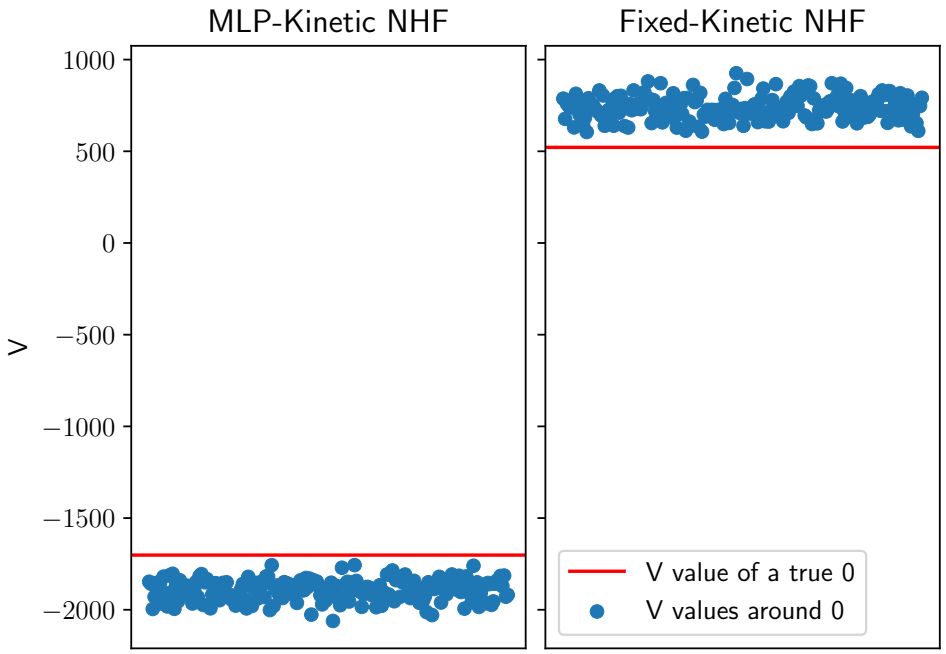


Figure 4.10: For two NHF models trained on the MNIST handwritten digits, we plot the value of the potential for a true '0' corresponding to a point x_0 in a 784-dimensional logit space, as well as the values of the potential for 200 perturbations of the form $x = x_0 + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, 2^2 I_{784})$.

GitHub repository https://github.com/lucidrains/denoising-diffusion-pytorch/tree/main. We adapted it so that it has 2.7 million parameters (same order as the other algorithms) and uses our usual pre-processing step. Our experiments ran on a HPC cluster with the same number of epochs and batch size than for our previous experiments with normalizing flows. We tested the model with $L = 10$ and $L = 100$ denoising steps. The results compiled in Figure 4.12 indicate that multiple dozen

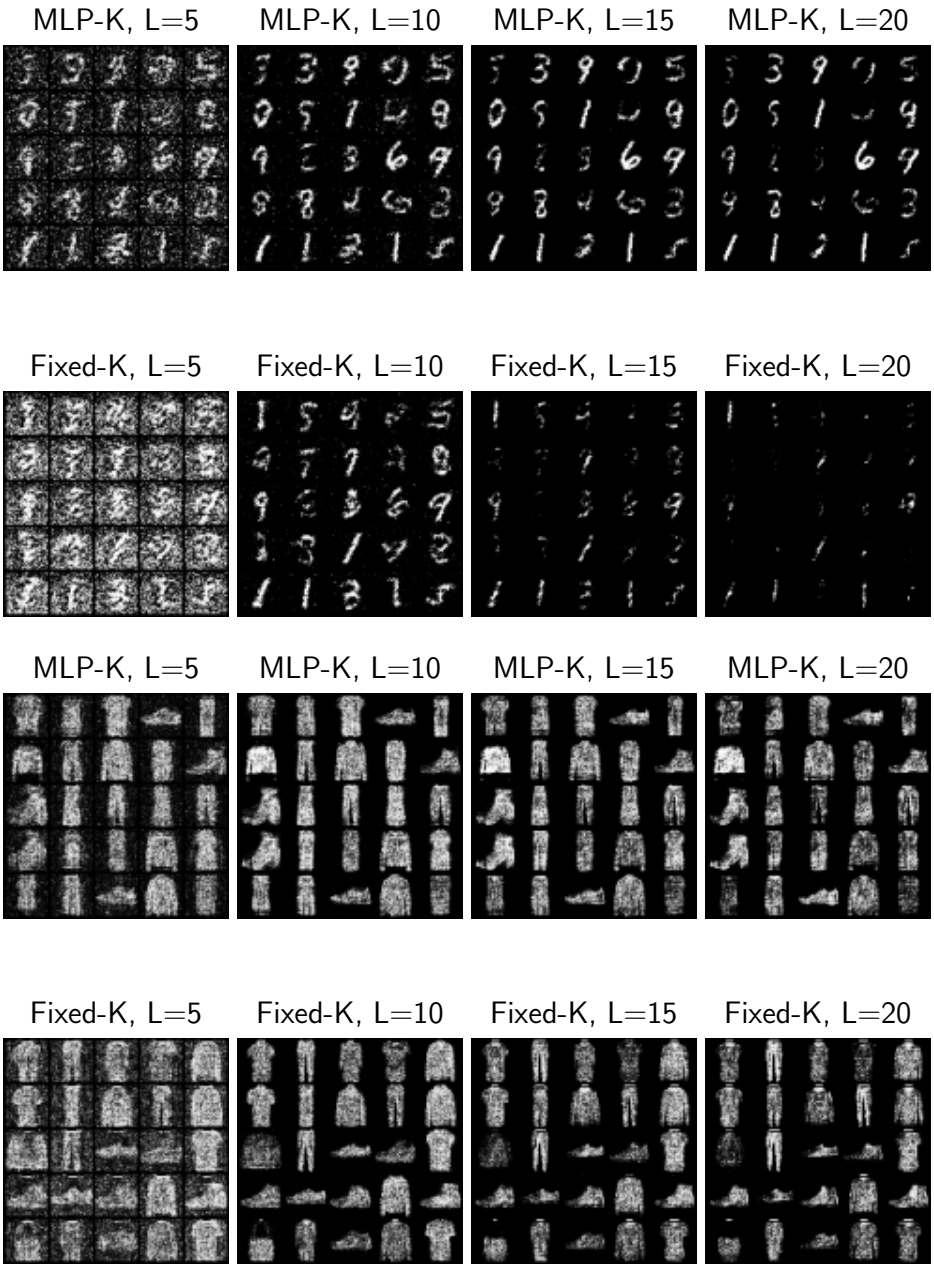


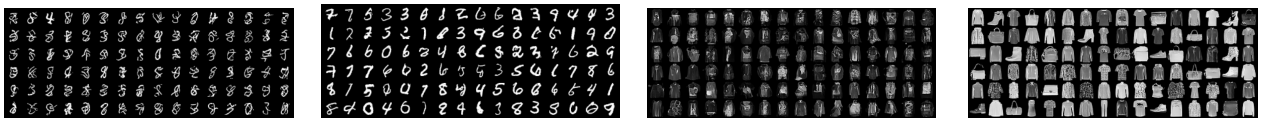
Figure 4.11: Stability of both NHF models trained with $L = 10$ to the number of Leapfrog steps, for the MNIST handwritten digits dataset (first and second rows) and the Fashion MNIST dataset (third and fourth rows).

of denoising steps are required for a diffusion model. This should be directly compared with the 10 Leapfrog steps used within our NHF models. Another major difference is that sampling with a trained diffusion model is not free since it requires approximating the reverse process from noise to images with a Markov chain.

4.4 Adapting NHF for Bayesian inference

4.4.1 Bayesian inference with generative models

Traditional MCMC methods (Robert and Casella, 2004) are very popular because they come with guarantees in terms of convergence and many progress have been made regarding their tuning (Homan



(a) $L = 10$, KL $= 250$ (b) $L = 100$, KL $= 70$ (c) $L = 10$, KL $= 340$ (d) $L = 100$, KL $= 70$

Figure 4.12: Samples generated by diffusion models. Multiple dozen of denoising steps are necessary to get high quality results.

and Gelman, 2014; Carpenter et al., 2017). NF architectures have also been proposed in this framework (Rezende and Mohamed, 2015; Winkler et al., 2019). Here, we adapt NHF to sampling Bayesian posterior distributions by transforming the prior distribution into the posterior with no access to samples from the target. In this framework, the parameter of interest is modeled by a random vector $\mathbf{Q}$. In the presence of data $\boldsymbol{d}$, Bayesian inference consists in inferring the posterior distribution $\pi(\mathbf{q}|\boldsymbol{d})$ which describes the probability that the parameter of interest takes certain values, knowing data. Such distribution can be re-written using Bayes' theorem as:

$$\pi(\mathbf{q}|\boldsymbol{d}) = \frac{\pi_0(\mathbf{q}) \times \ell(\boldsymbol{d}|\mathbf{q})}{Z} \tag{4.1}$$

This quantity is the product of a prior distribution $\pi_0(\mathbf{q})$, describing our knowledge on the parameter of interest in the absence of data, and a likelihood term $\ell(\boldsymbol{d}|\mathbf{q})$, which quantifies how probable the data are if we assume some fixed value for the parameter of interest. The overall quantity is divided by a constant Z which is often intractable.

There is a rich literature dealing with stochastic procedures for solving this task. Markov Chain Monte Carlo algorithms (Metropolis et al., 1953; Duane et al., 1987; Robert and Casella, 2004) consist in building a Markov chain with invariant distribution the target. Such method, while being exact, require a lot of computing effort to produce samples one by one. More recently, Machine Learning methods (Noé et al., 2019) have been used to tackle these problems.

4.4.2 Methodology, derivation of the new loss function

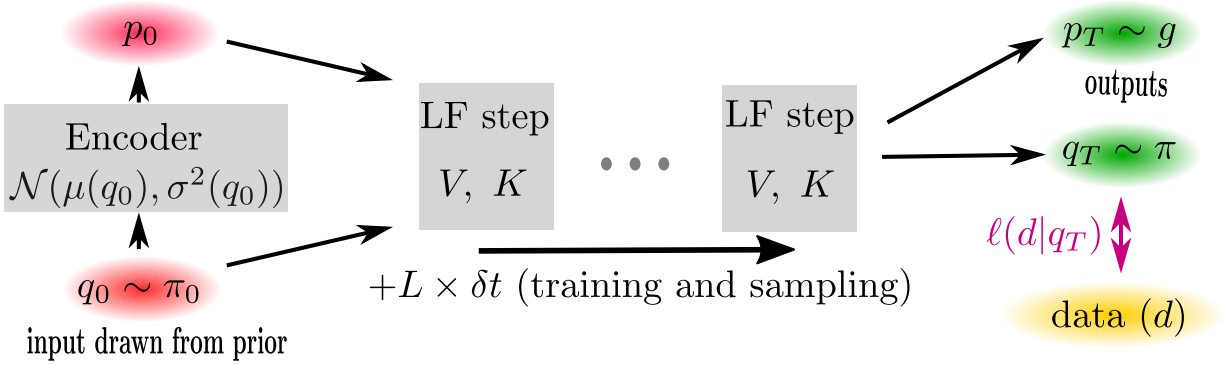


Figure 4.13: Schematic representation of NHF for Bayesian inference.

NHF can be used to perform Bayesian inference, by using Hamiltonian flows to transform the prior distribution, in the sense of Bayes' theorem, π_0 of some vector of parameters $\mathbf{q}$ into the target posterior distribution $\pi(\mathbf{q}|\boldsymbol{d})$ of these parameters, knowing some data $\boldsymbol{d}$ and likelihood distribution ℓ (see Figure 4.13). The main difference with the above-described NHF lies in the loss inspired by the KL phase in Boltzmann Generators (Noé et al., 2019), as well as in the learning procedure. The NHF becomes a generator of a family of functions for variational inference. During training, this NHF takes batches of $\mathbf{q}_0$ from the prior distribution as inputs. For each $\mathbf{q}_0$, one $\mathbf{p}_0$ is drawn from a Gaussian

distribution whose mean and deviation depend on the $\mathbf{q}_0$. The resulting point in phase-space evolves through L Leapfrog steps with integration time δt. The outputs consist in the final positions $\mathbf{q}_T$ and momenta $\mathbf{p}_T$, as well as the initial mean $\mu(\mathbf{q}_0)$, deviation $\sigma(\mathbf{q}_0)$ and $\mathbf{p}_0$. All these outputs, as well as the data $\boldsymbol{d}$, are used in the loss computation. Once trained, it can transform the prior into the desired posterior distribution of the parameters. Thus, both training and sampling are now made following the forward-direction flow from the prior to the posterior.

Computing the loss requires access to the likelihood distribution ℓ of the model, which encapsulates the covariance matrix of the data as well as the underlying physical mapping between vectors of parameters and the corresponding data. In the framework of Hamiltonian dynamics, the full system is made of both positions (the parameters of interest) and artificial momenta. We call $\mathbf{q}_0, \mathbf{p}_0$ the initial position and momentum, respectively, and $\mathbf{q}_T, \mathbf{p}_T$ the corresponding final position and momentum, respectively, obtained after L Leapfrog transformations $\mathcal{T}_1^{\delta t}, ..., \mathcal{T}_L^{\delta t}$ with timestep δt, i.e: $(\mathbf{q}_T, \mathbf{p}_T) = \mathcal{T}_L^{\delta t} \circ ... \circ \mathcal{T}_1^{\delta t}(\mathbf{q}_0, \mathbf{p}_0) := \mathcal{T}(\mathbf{q}_0, \mathbf{p}_0)$. Also, we introduce the notations for the projections along the final positions and momenta, i.e. $\mathbf{q}_T := \mathcal{T}_q(\mathbf{q}_0, \mathbf{p}_0)$ and $\mathbf{p}_T := \mathcal{T}_p(\mathbf{q}_0, \mathbf{p}_0)$. By changing the variables, the model joint distribution M may be written as:

$$\begin{aligned} M(\mathbf{q}_T, \mathbf{p}_T) &= 1 \times \Pi_0(\mathcal{T}_1^{-\delta t} \circ ... \circ \mathcal{T}_L^{-\delta t}(\mathbf{q}_T, \mathbf{p}_T)) \\ &= \Pi_0(\mathbf{q}_0, \mathbf{p}_0) = \pi_0(\mathbf{q}_0) \times f(\mathbf{p}_0|\mathbf{q}_0), \end{aligned}$$

where Π_0 is the joint prior distribution, π_0 the prior distribution of the parameters of interest and f the Gaussian distribution of the Encoder. We fix the target density of the final momenta $g(\mathbf{p})$ (e.g. Gaussian). We then minimize the KL-divergence between the model joint distribution and the desired target joint distribution conditioned on data $\Pi(\mathbf{q}, \mathbf{p}|\boldsymbol{d}) = \pi(\mathbf{q}|\boldsymbol{d})g(\mathbf{p})$. We write the latter as the product of a density depending on $\mathbf{q}$ and one depending on $\mathbf{p}$. Using Bayes' theorem, this yields:

$$\begin{aligned} &D_{KL}(M(\mathbf{q}_T, \mathbf{p}_T) \;||\; \pi(\mathbf{q}_T|\boldsymbol{d})g(\mathbf{p}_T)) \\ &= \int M(\mathbf{q}_T, \mathbf{p}_T) \log M(\mathbf{q}_T, \mathbf{p}_T) d\mathbf{q}_T d\mathbf{p}_T \\ &- \int M(\mathbf{q}_T, \mathbf{p}_T) \left[\log \pi(\mathbf{q}_T|\boldsymbol{d}) + \log g(\mathbf{p}_T)\right] d\mathbf{q}_T d\mathbf{p}_T \\ &= \int \Pi_0(\mathcal{T}^{-1}(\mathbf{q}_T, \mathbf{p}_T)) \log \Pi_0(\mathcal{T}^{-1}(\mathbf{q}_T, \mathbf{p}_T)) d\mathbf{q}_T d\mathbf{p}_T \\ &- \int M(\mathbf{q}_T, \mathbf{p}_T) \left[\log \pi_0(\mathbf{q}_T) + \log \ell(\boldsymbol{d}|\mathbf{q}_T) - \log p(\boldsymbol{d}) + \log g(\mathbf{p}_T)\right] d\mathbf{q}_T d\mathbf{p}_T \qquad (4.2) \\ &= \int \Pi_0(\mathbf{q}_0, \mathbf{p}_0) \left[\log \pi_0(\mathbf{q}_0) + \log f(\mathbf{p}_0|\mathbf{q}_0)\right] d\mathbf{q}_0 d\mathbf{p}_0 \\ &- \int M(\mathbf{q}_T, \mathbf{p}_T) \left[\log \pi_0(\mathbf{q}_T) + \log \ell(\boldsymbol{d}|\mathbf{q}_T) + \log g(\mathbf{p})\right] d\mathbf{q}_T d\mathbf{p}_T + \text{cst} \\ &= \int \Pi_0(\mathbf{q}_0, \mathbf{p}_0) [\log \pi_0(\mathbf{q}_0) + \log f(\mathbf{p}_0|\mathbf{q}_0) - \log \pi_0(\mathcal{T}_q(\mathbf{q}_0, \mathbf{p}_0)) - \log \ell(\boldsymbol{d}|\mathcal{T}_q(\mathbf{q}_0, \mathbf{p}_0)) \\ &- \log g(\mathcal{T}_p(\mathbf{q}_0, \mathbf{p}_0)] d\mathbf{q}_0 d\mathbf{p}_0 + \text{cst}. \end{aligned}$$

We can also adapt the ELBO to our inference framework, leading to a different loss function:

$$\begin{aligned} \ln \pi_0(\mathbf{q}_0) &= \ln \int \Pi_0(\mathbf{q}_0, \mathbf{p}_0) d\mathbf{p}_0 \\ &= \ln \int \frac{\Pi_0(\mathbf{q}_0, \mathbf{p}_0)}{f(\mathbf{p}_0|\mathbf{q}_0)} f(\mathbf{p}_0|\mathbf{q}_0) d\mathbf{p}_0 \\ &= \ln \mathbb{E}_f \left[\frac{\Pi_0(\mathbf{q}_0, \mathbf{p}_0)}{f(\mathbf{p}_0|\mathbf{q}_0)}\right] \\ &\geq \mathbb{E}_f \left[\ln \Pi_0(\mathbf{q}_0, \mathbf{p}_0) - \ln f(\mathbf{p}_0|\mathbf{q}_0)\right] \\ &= \mathbb{E}_f \left[\ln M(\mathcal{T}(\mathbf{q}_0, \mathbf{p}_0)) - \ln f(\mathbf{p}_0|\mathbf{q}_0)\right]. \end{aligned}$$

Then, expliciting $M(q,p) = \pi_0(q)\ell(d|q)g(p)$:

$$\mathrm{ELBO}(\mathbf{q}_0) = \mathbb{E}_f\left[\ln\left[\pi_0(T_q(\mathcal{T}(\mathbf{q}_0,\mathbf{p}_0)))\ell(d|T_q(\mathcal{T}(\mathbf{q}_0,\mathbf{p}_0)))g(T_p(\mathcal{T}(\mathbf{q}_0,\mathbf{p}_0)))\right] - \ln f(\mathbf{p}_0|\mathbf{q}_0)\right]. \tag{4.3}$$

4.4.3 Application to cosmology

We apply the above architecture to cosmological analysis: the determination of the cosmic expansion, and more generally of the cosmological parameters, from the observation of brightness and recession velocity of Type Ia supernovæ (e.g. Riess et al., 1998; Betoule et al., 2014). While this model used so far has been simple, it may be expanded in very complicated directions for which sampling from the probability distribution becomes complex. New observatories are presently being built and expected to deliver tens of thousands of new supernovæ Ia over the next decade (Abell et al., 2009).

According to the Λ-CDM model, the relation between the distance and the brightness of Type Ia supernovae is of great interest because it depends on two cosmological parameters: the matter density parameter Ω_m and the adimensional Hubble parameter h. To be more specific, a database of type Ia supernovae reports the distance modulus μ. This quantity is defined as the difference between the apparent and the absolute magnitude of an astronomical object and is directly related to luminosity distance (Weinberg, 1972) and thus a function of the redshift z, Ω_m and h:

$$\mu(z,\Omega_m,h) = 5\log_{10}\left(\frac{D_L^*(z,\Omega_m)}{h10\mathrm{pc}}\right)$$

where

$$D_L^*(z,\Omega_m) = \frac{c(1+z)}{H_0}\int_0^z \frac{ds}{\sqrt{1-\Omega_m+\Omega_m(1+s)^3}},$$

and $H_0 = 100$ km s^{-1} Mpc^{-1}, c being the speed of light.
In practice, we avoid computing the integral in D_L^* by using an approximation from (Pen, 1999) which is only valid for a flat Universe:

$$D_L^*(z,\Omega_m) = \frac{c(1+z)}{H_0}\left[\eta(1,\Omega_m) - \eta\left(\frac{1}{1+z},\Omega_m\right)\right],$$

with

$$\eta(a,\Omega_m) = 2\sqrt{1+s^3}\left(\frac{1}{a} - 0.1540\frac{s}{a^3} + 0.4304\frac{s^2}{a^2} + 0.19097\frac{s^3}{a} + 0.066941s^4\right).$$

Note that the formal definition of these quantities imposes constraints on the possible values of the parameters, that can only be comprised between zero and one. We avoid the problem by outputting a sigmoid of these parameters.

We aim to sample from the posterior distribution $\pi(\Omega_m, h|\mathrm{data})$ quantifying the probability that we are living in a universe whose mean density and expansion is equal to Ω_m and h given D observations $\mathrm{data} = \{z_i,\mu_i\}_{1\leq i\leq D}$ of type Ia supernovae, and the covariance matrix C of the observed distance moduli. We assume for simplicity that the likelihood of the problem is Gaussian, i.e. the observed data and the simulated output from parameters differ up to Gaussian noise. We note that the exact simulation for cosmological analysis of the supernova brightness is a complicated and expensive procedure, involving many nuisance parameters which participate in the final noise. It is also a toy model for more complex inference procedures, such as one relying on galaxy clustering, or weak lensing. It is thus crucial to use the least amount of parameters, and the least simulations possible to run the inference, which is the aim of this section. The final momenta distribution g is set to a Normal distribution. The trace plots in figure 4.14 compare the performance of an FK and MLPK-NHF with an HMC. They represent the cumulative means and standard deviations of the set of samples, which gets bigger as more and more are generated by the models. Such plots are biased with NHF models since they

minimize the KL-divergence between the model distribution and the target, but less than with an ELBO.

We leave for future work a possible correction using importance sampling methods at the end of training. However, there will always be a tradeoff between the quality of sampling and the computational cost. In exact MCMC methods, a constant cost needs to be paid every time a sample is generated. With generative models like ours, this cost comes from training because once trained, sampling is almost free. NHF for Bayesian inference should be understood as follows: they are compact models that encapsulate an approximate knowledge - yet quite precise as the number of data increases along with sufficient training - of the parameters of interest, according to Bayes' formula that updates our beliefs from data in a logical and consistent manner.

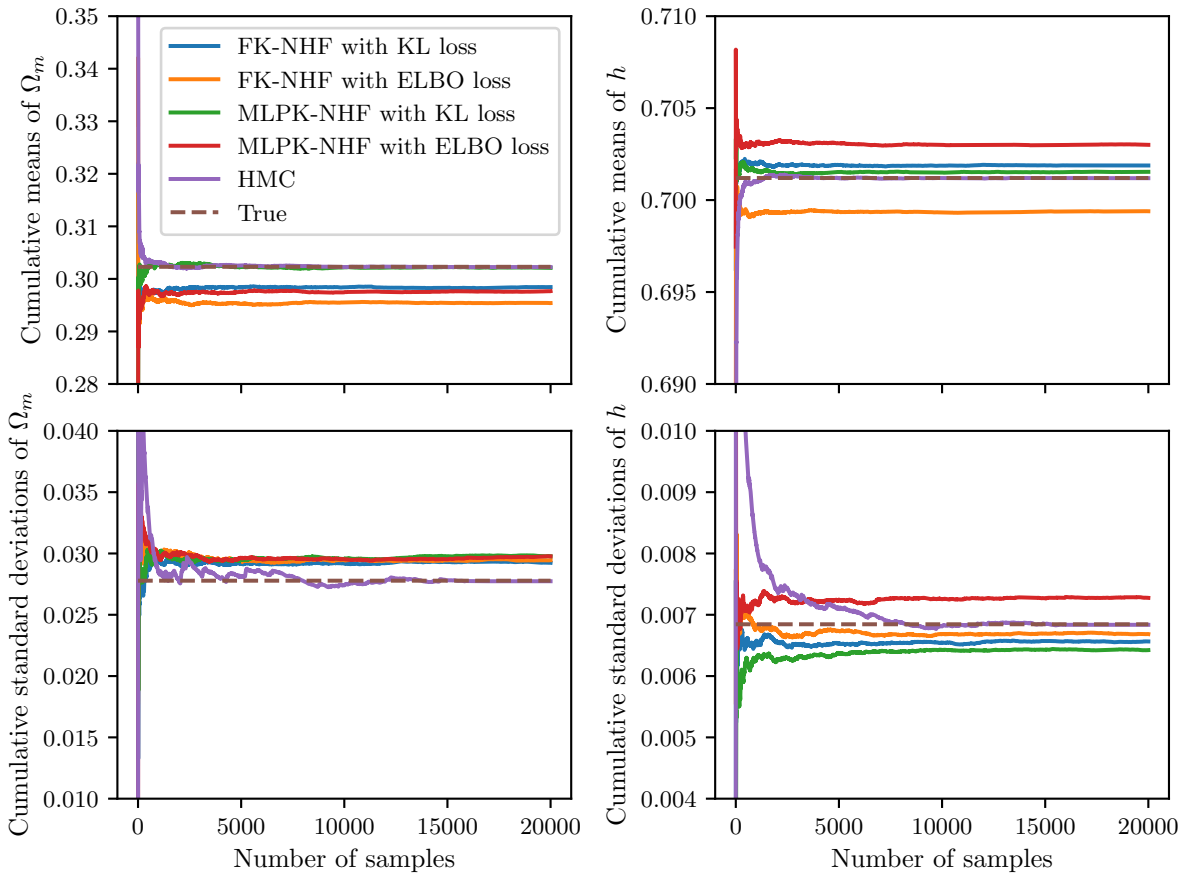


Figure 4.14: Trace plots of means and standard deviations of Ω_m and h produced by trained NHF models and for an HMC on a 20,000-sample dataset, compared to the ground truth. Soft-uniform prior, 30,000 training epochs, $g \sim \mathcal{N}(0, I_2)$.

Conclusion

In this work, we analyzed and improved Normalizing Hamiltonian Flows algorithms for Generative modeling. The main advantage of these methods is twofold. First, the volume-preservation in phase-space avoids the costly computation of Jacobian determinants. Then, as reversibility is ensured by the symplectic integrator, they allow for flexibility in the neural network architecture. This flexibility allowed us to propose a NHF variant based on classical kinetic energy. By exploring a 2D mixture problem, we illustrated how the explicit classical design of the kinetic energy is a way to increase robustness and facilitate interpretability while reducing the computational cost. While testing NHF

models for image generation, both show similar generative performance and are able to preserve their interpretability properties. It is noteworthy that, compared to diffusion models, they only require a short dynamics integration. Finally, we explained how to adapt NHF to the context of Bayesian inference to obtain a sampler of the posterior distribution. Further work will address methodological issues as to how the bias generated by a trained model could be corrected by importance sampling techniques, typically on high dimensional cosmological models but also more fundamental questions regarding a more precise comparison of NHF with diffusion models.

General conclusion

I knew exactly what to do. But in a much more real sense, I had no idea what to do.

Michael Scott, *The Office, Season 5: Stress Relief*

The topic of this thesis is the development and implementation of two samplers within the context of large-scale structure inference. The first algorithm, called PDMC-BORG, is a non-reversible Monte Carlo algorithm. It is used to perform exact sampling of the primordial flucuations field. We compared its performance to that of a Hamiltonian Monte Carlo sampler which is the current state of the art. The second one, called Fixed-kinetic NHF, is an interpretable generative model based on symplectic Hamiltonian flows. We tested it in the context of image generation but also showed how it could be adapted to Bayesian statistical inference through the determination of cosmological parameters from the standard model. As already highlighted, they are both different by nature and applications. As a scalable Monte Carlo method, PDMC-BORG is suited to statistical inference in very high-dimensional spaces - with more than 10^4 free parameters in our examples. This is not the case of Fixed-kinetic NHF. Generative models in general suffer from high-dimensionality and sometimes they are simply restricted by architecture limitations rather than by the intrinsic particular geometry of such spaces. PDMC-BORG inherits from the exactness properties that come with Markov Chain Monte Carlo methods. However, generating one sample requires paying a fixed computational budget. On the other hand, Fixed-kinetic NHF does not come with exact convergence guarantees. But once the cost of training is, the model is able to generate as many samples as one wants for almost free. They have in common that they are both interpretable and suited to cosmological Bayesian inference, i.e. building knowledge about the universe from astronomical data using the aggregation rules given by Bayes' formula. They can be used for testing different cosmological models, a task that would be replaced by a Bayesian model selection operation. The very first conclusion of this thesis is thus very general: it illustrates the difference between classical MCMC methods and more recent learning techniques.

In the field-level inference approach, PDMC-BORG showed promising performance. Its ballistic dynamics is particularly well suited to the exploration of high-dimensional spaces and it has less parameters to tune than a vanilla Hamiltonian Monte Carlo. Getting rid of the detailed-balance position, one could expect enormous gains. However, the algorithm is limited for both theoretical and practical reasons. First, the posterior region that is explored seems convex and isotropic. In this context, HMC is able to perform just fine as it is particularly suited to the spherical shape of the typical set. Second, computing the events time at which direction changes occur is analytically intractable and thus requires the use of a costly thinning procedure which does not exist for HMC-BORG. This extra cost makes PDMC-BORG as performing as the HMC in the end. Computing the directional derivative bounds when not much information is available can be a real burden impeding these methods to fully exploit their potential and outperform classical MCMC methods by a significative factor. However, as the dimensionality of the problem increases in future experiments, we expect to spot relevant differences between the two algorithms.

Regarding the use of generative models, we developed a better understanding of Hamiltonian flow-based models. As they rely on symplectic transformations, Neural Hamiltonian Flows are a very interesting type of flexible and numerically-efficient samplers. This led us to design a variety of alternatives to the usual NHF architecture. While focusing on robustness, we also demonstrated their relevance in the context of image generations as well as Bayesian inference. Being non-exact, they still have a reasonable expressive power. They should be seen as compressed approximators to the true target distribution they are trying to model. All of this comes with appealing interpretability properties. NHF, and in particular its fixed-kinetic version, form a sub-class of more general Physics-informed models. The Hamiltonian formalism in this case makes it straightforward to incorporate symmetry invariance prior knowledge inside the model and improve training significantly.

The different projects investigated during this PhD thesis naturally open new research paths. Some of them are straightforward and could be conducted in the short term: they involve further tests regarding the performance of the algorithms. Some are more prospective: they deal with theoretical analysis of the mathematical objects that have been introduced and their inherent properties. We thus end the manuscript by specifying a few possible future directions regarding this work.

- **Automatic tuning of PDMC-BORG.** The $t_{\max}$ and p_{ref} or δt_{ref} parameters could be automatically tuned on the fly during the run according to certain criteria. For $t_{\max}$, that would be minimizing the number of evaluations per event. As for the others, it should make the integrated autocorrelation time as small as possible.

- **Testing PDMC-BORG in higher dimensions.** Our tests were run in dimension $32^3 = 32768$. However, state of the art samplers are able to perform inference in spaces of order 10^6 and more free parameters. That would require adapting the `Python` code for acceleration, probably using `C++` functions.

- **Testing PDMC-BORG on real data.** As for now, our performance tests were conducted using artificial mock data corresponding to the primordial field. It would be interesting to test the algorithm on real astronomical data, for example galaxy counts from LSST surveys (Abell et al., 2009).

- **Learning the energy landscape for PDMC-BORG.** Since finding an appropriate bound is a complicated task, one option could consist in learning the energy of the system with a Hamiltonian neural network (Greydanus et al., 2019) in order to propose adequate bounds. Such procedure could also be applied to HMC-BORG for proposing moves without having to compute the whole gradient.

- **Bias analysis for NHF models.** It has been demonstrated that volume-preserving Normalizing flows have an incompressible bias that fundamentally limits their expressive power (Draxler et al., 2024). It is unclear, though, if this results directly applies to our framework since volume-preservation occurs in the whole phase-space and the momenta are marginalized out in the final expression of the parameters density.

- **Interpretability analysis of diffusion models.** As they rely on physical diffusion processes, probabilistic diffusion models may be highly interpretable, in the sense that the learned mapping could give us relevant information about the target distribution. A similar analysis to the one done with NHF could be performed on diffusion models.

Résumé détaillé de la thèse (Français)

À l'origine fut la vitesse, le pur mouvement furtif, le "vent-foudre". Puis le cosmos décéléra, prit consistance et forme, jusqu'aux lenteurs habitables, jusqu'au vivant, jusqu'à vous. Bienvenue à toi, lent homme lié, poussif tresseur des vitesses.

Alain Damasio, *La Horde du Contrevent*

Titre : *Algorithmes d'échantillonnage non-réversibles et génératifs. Application à l'inférence de paramètres cosmologiques.*

Abstract : Cette thèse de doctorat est consacrée au développement et à l'analyse d'algorithmes d'échantillonnage appliqués à l'inférence de paramètres cosmologiques. Nous passons d'abord en revue les outils mathématiques ainsi que le cadre cosmologique. Ensuite, nous introduisons deux algorithmes. Le premier est appelé PDMC-BORG. Il s'agit d'un échantillonneur de Monte Carlo par chaîne de Markov non-réversible utilisé pour l'inférence de structures à grande échelle. Il s'appuie sur la machinerie BORG développée par le consortium Aquila pour inférer le champ de densité primordial à partir de données astronomiques. Nous détaillons les principales caractéristiques de l'algorithme, expliquons comment le régler et montrons que ses performances sont similaires à celles d'un échantillonneur classique de type Monte Carlo hamiltonien. Nous présentons ensuite une variante à énergie cinétique fixée de Neural Hamiltonian Flows, un type de modèle génératif qui utilise des transformations hamiltoniennes symplectiques pour transformer une distribution de base en n'importe quelle cible. Notre modification permet d'améliorer l'interprétabilité du modèle tout en réduisant sa complexité numérique. Nous testons ses performances dans la génération d'images et expliquons comment utiliser le Neural Hamiltonian Flows et ses variantes dans le contexte de l'inférence bayésienne, en illustrant la méthode sur l'inférence de deux paramètres cosmologiques à partir d'observations de supernovae.

Contexte général : Cette thèse, débutée en octobre 2021, a pour objectif l'étude, l'amélioration et l'implémentation d'algorithmes d'échantillonnage pour la cosmologie. Elle a été conduite au sein du Laboratoire de Mathématiques Blaise Pascal[1], sous la direction d'Arnaud Guillin et de Manon Michel ainsi que d'un encadrant de l'Institut d'Astrophysique de Paris[2], Guilhem Lavaux. Ce travail a été réalisé en étroite collaboration avec le consortium Aquila[3], qui regroupe actuellement une trentaine de cosmologistes et statisticiens, avec l'objectif de développer des méthodes robustes d'analyse de données pour l'étude de la structure à grande échelle de l'Univers. Le but de ce résumé détaillé est de fournir au lecteur francophone une description générale mais néanmoins précise de la thèse. Il situe le travail de recherche par rapport à d'autres travaux existants et précise les contributions scientifiques apportées. Cela peut donc constituer une bonne base pour appréhender les idées générales qui ont

[1] https://lmbp.uca.fr/
[2] https://www.iap.fr/
[3] https://www.aquila-consortium.org/

présidé à la réalisation de ces travaux. Pour plus de détails techniques sur les outils mathématiques et cosmologiques utilisés, les commentaires théoriques et les expériences numériques menées, le lecteur pourra se référer à la partie précédente du manuscrit rédigée en anglais, qui est indépendante et bien plus exhaustive.

Inférence statistique pour la cosmologie

L'utilisation d'algorithmes d'échantillonnage probabilistes pour la résolution de problèmes issus de la cosmologie n'est guère surprenante. Depuis l'émergence des algorithmes de Monte Carlo (Metropolis et al., 1953), l'utilisation du hasard pour modéliser des phénomènes complexes issus de la Physique n'a cessé de se développer, jusqu'à l'avénement récent des modèles génératifs (Goodfellow et al., 2016). Les prochains défis en cosmologie vont demander de manipuler de grands volumes de données et de caractériser des espaces de dimension gigantesque - jusqu'à plusieurs milliards de paramètres. Cela ne sera vraisemblablement possible qu'à travers l'utilisation d'outils numériques robustes, exploitant habilement l'exploration stochastique et s'appuyant sur de puissantes infrastructures de calcul. Et réciproquement, ces nouveaux défis posés par la cosmologie offrent un terrain de jeu privilégié pour tester les limites des algorithmes sur des problèmes difficiles. L'objectif de cette thèse est d'étudier l'application de nouvelles classes d'algorithmes à des problèmes d'inférence statistique en cosmologie. Commençons donc par situer le problème qui nous intéresse et les différentes méthodes existantes pour le résoudre, avec leurs limitations.

Structure à grande échelle de l'univers

D'après le modèle standard de la cosmologie (Einstein, 1916; Friedmann, 1922, 1924; Lemaître, 1927; Robertson, 1935; Walker, 1937), l'univers est né d'un Big Bang il y a environ 13,8 milliards d'années, selon les meilleures estimations (Spergel et al., 2003; Planck Collaboration et al., 2020). Après une courte phase d'inflation puis de nucléosynthèse primordiale, l'univers âgé de 380 000 ans est devenu suffisamment froid et peu dense pour que des atomes stables puissent se former et que le rayonnement électromagnétique puisse circuler librement. Cette lueur originelle baigne aujourd'hui tout le cosmos sous le nom de Fond Diffus Cosmologique - en anglais, Cosmic Microwave Background (CMB) (Penzias and Wilson, 1965). Si le ciel nocturne nous apparaît sombre, ce n'est donc que parce que l'œil humain n'est pas sensible au rayonnement micro-onde qui caractérise cet objet, assimilable à un corps noir rayonnant à une température d'environ $2,7$ degrés Kelvin (Penzias and Wilson, 1965; Peebles, 1980; Fixsen, 2009). Très isotrope et homogène, avec des fluctuations de température de l'ordre de quelques cent millièmes de degrés, ce signal présente cependant de très légères anisotropies de l'ordre de 10^{-5}, caractérisant de très faibles variations locales du potentiel gravitationnel, sous l'effet de Sachs-Wolfe (Sachs and Wolfe, 1967). Cela se traduit par d'infimes différences de température dans le signal du CMB, comme montré sur la Figure 4.15.

L'univers présent, quant à lui, est bien différent. À très grande échelle, de l'ordre de plusieurs centaines de mégarparsecs, on observe de grandes variations de densité. La structure observée est en réalité filamentaire : la matière s'organise autour d'un réseau complexe et difficile à décrire statistiquement. À l'intersection de ces longs filaments, là où la densité est la plus forte, on trouve des amas de galaxies. On parle de la *toile cosmique* (Bond et al., 1996). Obtenir une caractérisation de cette structure filamentaire, représentée sur la Figure 4.16, revêt une grande importance en cosmologie. Une première approche pour une meilleure compréhension de cette structure à grande échelle est celle de l'inférence des paramètres caractérisant le modèle standard de la cosmologie (Einstein, 1916; Friedmann, 1922, 1924; Lemaître, 1927; Robertson, 1935; Walker, 1937). Une autre, plus ambitieuse, concerne la description du champ des fluctuations de densité de l'univers présent.

Mais ce dernier problème est compliqué pour de multiples raisons. Premièrement, un observateur du

cosmos est nécessairement interne au système qu'il étudie ; de plus, il n'a accès qu'à une partie de son objet d'étude, à cause de l'expansion de ce dernier et de la vitesse finie à laquelle se propagent les signaux électromagnétiques. Deuxièmement, les biais observationnels et de sélection rendent difficiles la cartographie précise de l'univers à grande échelle (York et al., 2000). Enfin, puisqu'il est fait pour la majeure partie de composants qui échappent à l'observation directe et aux instruments de mesure - matière noire et énergie sombre -, (Bertone et al., 2005) il est vain de vouloir l'observer directement.

Or, il se trouve que cette structure riche provient en réalité de l'amplification des micro-fluctuations de densité de notre jeune univers, sous l'effet de mécanismes d'instabilité gravitationnelle. Et si la structure présente semble hors de portée, ce n'est pas le cas de l'univers jeune. La théorie (Linde, 2008), confirmée par de solides expériences (Komatsu et al., 2011; Planck Collaboration et al., 2020), suggère que l'univers âgé de 380,000 ans peut être décrit par des statistiques très simples. Ainsi, un relevé à grande échelle de galaxies devraient être en accord avec la théorie des fluctuations initiales et du modèle de gravitation qui lie les conditions initiales avec celles de l'univers présent. En effet, la répartition des galaxies dans l'univers suit la répartition de ces filaments, à tel point que les galaxies constituent un excellent traceur de matière. Jusqu'à peu, l'approche standard consistait à comparer les observations et la théorie au moyen d'outils statistiques comme les spectres de puissance (Peebles, 1980; Tegmark et al., 2004). Cela requiert notamment de savoir comment construire ce spectre de puissance à partir des données, sachant des contraintes supplémentaires (masques, par exemple). Aussi, il s'agit de connaître la forme de ce spectre, ce qui repose sur une connaissance théorique de la distribution -biaisée - des galaxies dans l'univers présent. Les approches traditionnelles font des hypothèses sur la distribution de ce spectre.

Puisque ces approches sont insuffisantes pour modéliser des statistiques d'ordre élevé, il a été proposé (Jasche and Wandelt, 2013) de suivre le principe suivant : partir des données astronomiques pour contraindre une histoire possible de notre univers. Comme nous allons le voir, cette approche prend tout son sens dans le cadre de l'inférence Bayésienne. Elle fait suite à de nombreux travaux par exemple sur la caractérisation des données du télescope spatial Wilkinson Microscopic Anisotropy Probe (WMAP) (Komatsu et al., 2011) avec l'outil COMMANDER (O'Dwyer et al., 2004), ou encore sur le rayonnement radio avec NIFTY (Selig et al., 2013).

Cette thèse s'inscrit donc dans un projet global de construction de *machines algorithmiques* destinées à mieux comprendre la structure à grande échelle de l'univers, en caractérisant la *distribution de probabilité* qui décrit les fluctuations de densité de l'univers, sachant des données astronomiques. Cela demande plusieurs ingrédients : **1)** un modèle d'évolution gravitationnel précis et différentiable, **2)** des algorithmes d'échantillonnage performants pour produire des échantillons de cette histoire cosmique et enfin **3)** des outils robustes pour analyser la qualité des échantillons produits. Dans ce manuscrit, c'est principalement sur le deuxième que nos efforts se sont portés. L'idée est de fournir à la communauté des cosmologistes des algorithmes d'échantillonnage puissants ainsi que des méthodes robustes pour analyser les échantillons produits. Avant de présenter les méthodes retenues, commençons par mieux expliciter le problème Bayésien sous-jacent ainsi que les méthodes traditionnellement utilisées pour le résoudre. Ce problème, par sa complexité, offre une formidable occasion de tester les limites et de mieux appréhender le comportement de différentes méthodes d'échantillonnage.

Traduction en un problème Bayésien aux conditions initiales

L'idée fondamentale est donc de trouver des débuts d'univers possibles d'après les contraintes posées par les observations astronomiques de l'univers présent. Du point de vue Bayésien, cela constitue un problème bien posé (Robert, 2001). Cela revient à construire des débuts d'univers *probables* sachant les observations astronomiques présentes, puis de les faire évoluer selon un modèle gravitationnel jusqu'au temps présent. Cette approche permet aussi de résoudre un problème fondamentale de la Cosmologie : l'histoire cosmique n'a eu lieu qu'une seule fois, donnant lieu à un seul point de données - l'univers tel que nous le connaissons. Le Bayésien est capable de donner une estimation cohérente de ses croyances

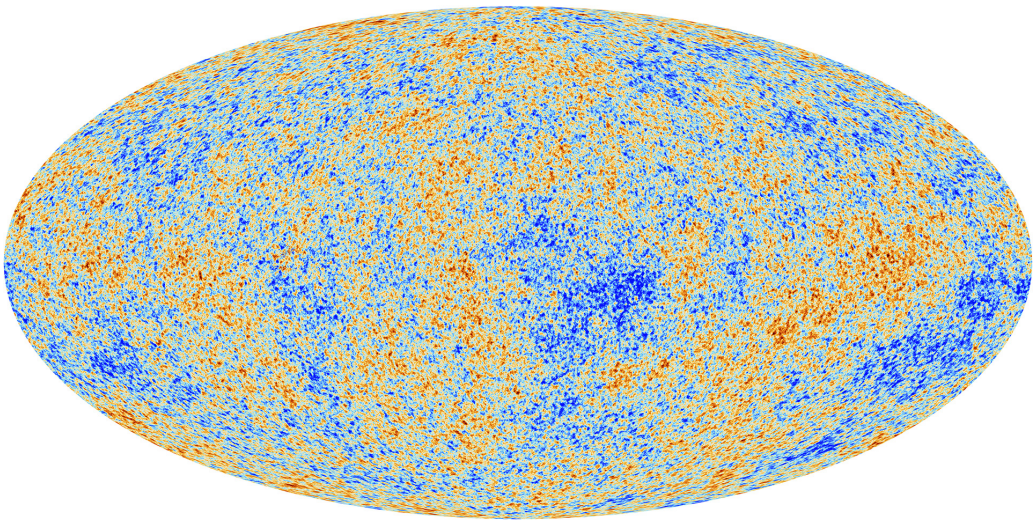


Figure 4.15: Le Fond Diffus Cosmologique reconstitué à partir d'observations du satellite Planck, de l'European Space Agency.

et de leur incertitude associée, grâce à la fameuse formule de Bayes qui permet de mettre à jour son savoir de manière logique à mesure que de nouvelles données lui parviennent :

$$\mathbb{P}(\text{Théorie}|\text{Données}) = \frac{\mathbb{P}(\text{Théorie}) \times \mathbb{P}(\text{Données}|\text{Théorie})}{\mathcal{Z}}$$

La quantité d'intérêt ici est la *distribution postérieure* $\mathbb{P}(\text{Théorie}|\text{Données})$ qui décrit la totalité des croyances en la validité d'une certaine Théorie, sachant les Données observationnelles à notre disposition. Dans le cadre de cette thèse, la Théorie va concerner soit la valeur de certains des six paramètres du modèle standard de la cosmologie, soit la répartition des fluctuations de densité de l'univers, et les Données correspondront à des observations astronomiques. Cette quantité peut s'exprimer comme le produit et quotient de plusieurs termes.

- D'abord, l'*a priori* $\mathbb{P}(\text{Théorie})$ qui décrit la probabilité assignée à la Théorie avant que les Données soient observées. Le choix de l'a priori occupe un pan entier de la littérature sur les méthodes Bayésiennes. Parmi les méthodes proposées, on peut citer les a priori conjugués, qui permettent d'obtenir une expression simple de la distribution postérieure si l'a priori et la vraisemblance appartiennent à certaines familles de distributions (Minton et al., 1961). Ou encore les a priori de Jeffreys, qui sont utilisés dans le cas où l'on dispose de très peu d'informations sur les paramètres à modéliser (Jeffreys, 1946). Pour l'inférence du champ primordial, les a priori utilisés résultent de notre bonne connaissance des propriétés de l'univers primordial et en particulier du Fond Diffus Cosmologique.

- Ensuite, la *vraisemblance* $\mathbb{P}(\text{Données}|\text{Théorie})$ qui donne la probabilité que les Données soient observées partant de l'hypothèse que notre Théorie est correcte. La construction de vraisemblances pertinentes pour le problème qui nous intéresse a été largement étudiée, notamment dans Jasche and Kitaura (2010) et Jasche and Wandelt (2013).

- Enfin, cette quantité est divisée par une constante de normalisation $\mathcal{Z}$, souvent difficile à calculer analytiquement et inaccessible numériquement, et qui justifie l'utilisation de méthodes algorithmiques robustes pour résoudre des problèmes d'inférence dans ce cadre.

Dans le cas de l'inférence du champ des fluctuations de densité de l'univers présent sachant des observations astronomiques, une double question se pose immédiatement :

1. *Comment discrétiser un champ continu, pour rendre le problème résoluble par un ordinateur ?*

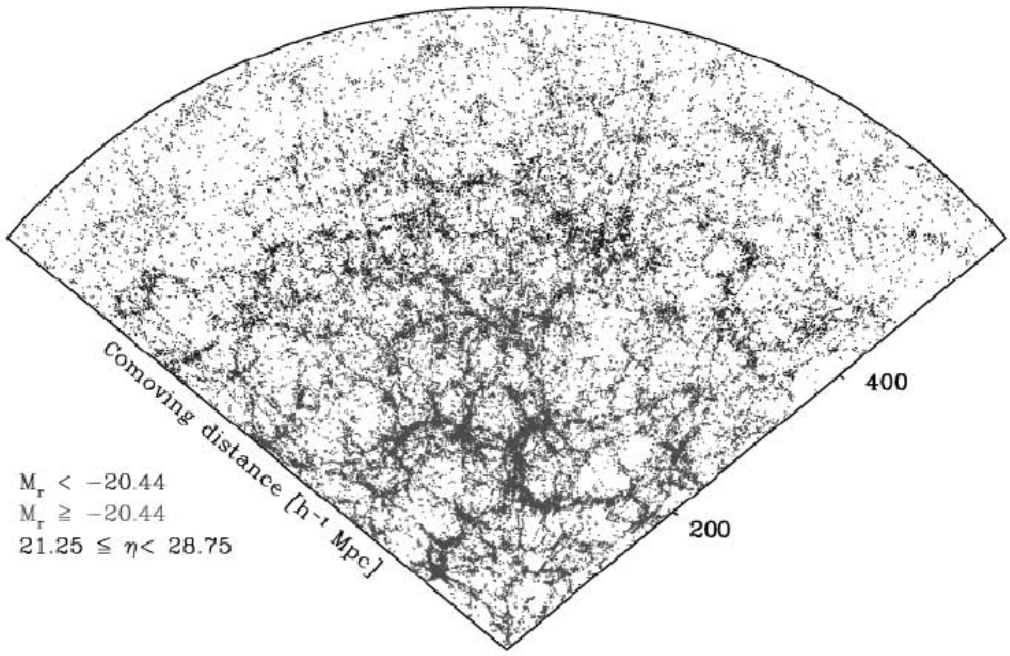


Figure 4.16: Distribution de galaxies dans une tranche réalisée à partir de données du Sloan Digital Sky Survey. L'image provient de Park et al. (2005). La coordonnée radiale correspond à la distance comobile, la coordonnée angulaire à la longitude de l'instrument. Le coin en bas correspond au lieu d'observation, c'est-à-dire à un point de notre galaxie. On observe clairement la structure filamentaire caractéristique de l'univers présent vu à grande échelle.

2. *Quels objets astronomiques sont pertinents pour tracer ces fluctuations de densité ?*

La réponse au premier problème est apportée par une méthode de dicrétisation : plutôt que d'étudier le champ continu des fluctuations de densité, il convient de travailler dans une boîte d'univers cubique plus ou moins grande selon le problème, et de la subdiviser régulièrement en plusieurs petites cellules cubiques elles aussi. La taille de ces dernières dépendra du niveau de résolution souhaité. L'idée consiste alors à inférer dans chacune de ces cellules, indexées de 1 jusqu'à d, un nombre scalaire δ_k correspondant à la valeur du contraste de densité de l'univers présent dans la cellule k. C'est aussi ce choix de résolution qui déterminera en partie le niveau de difficulté du problème : en effet, plus le nombre de cellules est élevé, plus la dimension de la distribution cible est grande : de plus, aux toutes petites échelles, il peut être difficile de caractériser les distortions du champ de fluctuation causées par le modèle de gravitation. Nous y reviendrons.

Quant au second problème de savoir quels traceurs sont les plus pertinents, il convient de remarquer que les fluctuations de densité suivent globalement la distribution des galaxies. Aussi, un traceur approprié est le nombre de galaxies N_k^g observées dans chacune des cellules k de notre boîte d'univers (Jasche and Wandelt, 2013).

Finalement, étant donnée une boîte d'univers et sa subdivision en cellules régulières cubiques indexées de 1 jusqu'à d, le problème revient à construire des échantillons qui suivent la distribution postérieure $\mathbb{P}\left(\{\delta_k\}_{k=1}^d | \{N_k^g\}_{k=1}^d\right)$. D'après la formule de Bayes, cette quantité peut se réécrire sous la forme :

$$\mathbb{P}\left(\{\delta_k\}_{k=1}^d | \{N_k^g\}_{k=1}^d\right) \underset{\text{Bayes}}{=} \frac{\mathbb{P}\left(\{\delta_k\}_{k=1}^d\right) \times \mathbb{P}\left(\{N_k^g\}_{k=1}^d | \{\delta_k\}_{k=1}^d\right)}{\mathbb{P}\left(\{N_k^g\}_{k=1}^d\right)}$$

La véritable limitation de cette formulation vient du fait qu'il est extrêmement difficile d'obtenir une bonne estimation de l'a priori concernant les contrastes de densité de l'univers présent. A très grande

échelle, l'emploi de lois normales s'est révélé pertinent alors que dans le régime non-linéaire, l'utilisation de lois log-normales a été explorée. En revanche, comme évoqué précédemment, les contrastes de densité finaux $\{\delta_k\}_{k=1}^d$ sont directement liées aux contrastes de densité initiaux $\{\delta_k^0\}_{k=1}^d$ par un processus déterministe, à savoir un modèle d'évolution gravitationnel que l'on note $\delta_j = G_j\left(\{\delta_k^0\}_{k=1}^d\right)$, pour tout $1 \leq j \leq d$. Plus précisément, il est possible d'appliquer une suite de transformations pour amener les conditions initiales post-inflation sur celles de l'Univers présent : un modèle donnant les fluctuations du champ de gravité créé par la période dominée par l'inflation, suivi d'une transformation amenant ces conditions dans une période dominée par la matière et enfin, au cœur du dispositif, un modèle d'évolution gravitationnel pour amener les fluctuations résultantes d'un redshift $z = 1000$ (juste après l'époque du Fond Diffus Cosmologique) jusqu'au temps présent, c'est-à-dire à $z = 0$ (Zel'dovich, 1970). Or il est beaucoup plus aisé de construire un a priori sur ces conditions initiales puisque l'univers jeune obéissait à des statistiques beaucoup plus simples. Ainsi, il est possible de traduire le problème en un problème d'inférence Bayésienne aux conditions initiales (Jasche and Wandelt, 2013; Jasche and Lavaux, 2019) :

$$\mathbb{P}\left(\{\delta_k^0\}_{k=1}^d | \{N_k^g\}_{k=1}^d\right) = \frac{\mathbb{P}\left(\{\delta_k^0\}\right) \times \mathbb{P}\left(\{N_k^g\}_{k=1}^d | \{G_k\left(\{\delta_l^0\}_{l=1}^d\right)\}_{k=1}^d\right)}{\mathbb{P}\left(\{N_k^g\}_{k=1}^d\right)}$$

Cette distribution postérieure constitue un objet central de notre étude. Il s'agit de générer des échantillons (indépendants) qui suivent cette loi, vivant dans un espace de très grande dimension. Par la suite, dans un souci de simplification, on identifiera souvent la loi de probabilité avec sa densité.

Méthodes numériques d'échantillonnage

Étant donnée une certaine loi de probabilité de densité $\pi : \mathbb{R}^d \to [0, 1]$, comment produire des échantillons distribués selon une loi de densité π ? En particulier dans les cas où :

- *la dimension d est (très) grande ;*
- *la densité π n'admet pas d'expression analytique facilement calculable.*

Les deux points ci-dessus correspondent précisément au cadre du problème d'inférence du champ des fluctuations primordiales détaillé dans la partie précédente. En effet, puisqu'il s'agit d'approximer un champ continu, il convient d'avoir un nombre très élevé de cellules dans chaque boîte d'univers, de l'ordre de 10^4 et jusqu'à 10^9 pour les simulations les plus récentes. Aussi, la distribution cible peut s'écrire sous la forme $\pi(x) = \frac{\exp(-E(x))}{\mathcal{Z}}$, avec $\mathcal{Z}$ une constante de normalisation dont il est difficile d'obtenir une expression analytique et qui est inaccessible par des méthodes d'approximation numérique.

Méthodes de Monte Carlo par Chaînes de Markov

Généralités. L'idée des méthodes Markov Chain Monte Carlo (MCMC) est de construire un objet mathématique probabiliste, appelé une *chaîne de Markov*, qui converge en loi vers la distribution postérieure d'intérêt (Robert and Casella, 2004). Supposons, pour simplifier, que l'on travaille dans un espace d'états continu à temps discret. Mathématiquement parlant, une chaîne de Markov est une suite $(X_n)_{n\geq 0}$ de variables aléatoires à valeurs dans un espace d'états, caractérisée par un état initial, qui est une réalisation aléatoire de la loi de X_0, et une probabilité de transition qui à tout instant ne dépend que de l'état courant :

$$\mathbb{P}(X_{n+1}|X_0, ..., X_n) = \mathbb{P}(X_{n+1}|X_n).$$

Si de plus cette dernière quantité ne dépend pas de n, alors on parle de chaîne de Markov *homogène*. De tels objets sont entièrement déterminés par la loi de X_0 et celle de la probabilité de transition.

Le but d'un algorithme MCMC est donc de parcourir l'espace d'états, ici l'espace des paramètres d'intérêt, et de générer une *chaîne* d'échantillons $\{x_0, x_1 ..., x_n\}$ dont la densité va être proportionnelle à celle de la distribution cible π. Pour construire une telle chaîne, il faut alors respecter trois principes :

1. **Invariance de la cible :** la distribution cible doit être un point fixe de l'opérateur de transition d'un état x' vers un état x, noté $\mathcal{T}(x', x)$, soit, si toutes les quantités admettent une densité :
$$\pi(x) = \int \mathcal{T}(x', x)\pi(x')dx'$$
2. **Irréductibilité :** pour toute paire d'états, la probabilité d'aller de l'un vers l'autre, éventuellement en plusieurs étapes, est non nulle.
3. **Apériodicité :** pour toute paire d'états $x, y \in E$, il existe $n_{x,y} \in \mathbf{N}$ tel que la probabilité d'atteindre y depuis x en n étapes est strictement positive dès que $n \geq n_{x,y}$.

Ces deux propriétés assurent l'existence et l'unicité d'une distribution invariante, celle de densité π. Ainsi, partant d'un point aléatoire dans l'espace des paramètres, la chaîne de Markov va progressivement se rapprocher des zones de forte densité de la distribution cible. Elle va ensuite générer des états qui seront distribués selon la distribution visée. La construction d'un algorithme MCMC efficace repose donc en grande partie sur un bon choix de $\mathcal{T}$. Deux tels algorithmes sont présentés dans la suite.

Metropolis-Hastings. L'idée de cet algorithme est la suivante : on se donne un point de départ x_0 dans l'espace des paramètres d'intérêt et une distribution $Q(x'|x)$ pour proposer un point candidat x' à partir du point courant x. A chaque étape, on tire un candidat x' selon l'état x. Ce nouveau candidat est alors accepté avec probabilité
$$p_{\text{acc}}(x, x') = \min\left(1, \frac{\pi(x')}{\pi(x)} \times \frac{Q(x|x')}{Q(x'|x)}\right)$$
On vérifie alors aisément que cela implique la condition d'invariance notée plus haut, sous une forme plus restrictive appelée la condition d'*équilibre détaillé*, sur laquelle nous ferons de nombreux commentaires dans la partie suivante.

Notons que cet algorithme se prête particulièrement bien au cadre Bayésien, où la distribution d'intérêt n'est pas analytiquement connue mais peut s'écrire sous la forme $\tilde{\pi} \times \mathcal{Z}^{-1}$, la quantité $\tilde{\pi}$ désignant le produit de l'a priori par la vraisemblance. Le rapport $\frac{\pi(x')}{\pi(x)}$ est alors égal à $\frac{\tilde{\pi}(x')}{\tilde{\pi}(x)}$. Enfin, dans le cas où la proposition Q est symétrique, c'est-à-dire où $Q(x|x') = Q(x'|x)$ pour tous x, x', alors on retrouve l'algorithme de Metropolis tel que formulé pour la première fois en 1953 par Metropolis et ses collaborateurs (Metropolis et al., 1953). La généralisation au cas non-symétrique a été faite par Hastings en 1970 (Hastings, 1970), ajoutant son nom à l'algorithme présenté ci-dessous sous forme de pseudo code 12. La principale limitation est que si le choix de Q n'est pas bien adapté à la distribution cible, alors l'exploration engendrée par l'algorithme sera peu efficace. En particulier, en grande dimension, cet algorithme souffre d'un comportement diffusif provoquant une très lente exploration des zones d'intérêt dans l'espace des paramètres.

Hamiltonian Monte Carlo. Le principe du HMC (Duane et al., 1987) est de s'appuyer sur la mécanique classique afin d'explorer efficacement l'espace des paramètres. Commençons par rappeler certains principes de la mécanique Hamiltonienne. Il s'agit d'augmenter l'espace des paramètres, assimilés à des positions $\mathbf{q}$, avec des momenta $\mathbf{p}$ (masse $\times$ vitesse) pour obtenir un espace de phases. Le système est alors complètement décrit par la liste de ses positions et momenta à laquelle on associe une grandeur scalaire appelée *Hamiltonien* qui exprime son énergie totale. On supposera ici que cette quantité H peut s'écrire comme la somme d'une énergie potentielle V, ne dépendant que des positions, et d'une énergie cinétique K, ne dépendant que des momenta. Autrement dit,
$$H(\mathbf{q}, \mathbf{p}) = V(\mathbf{q}) + K(\mathbf{p})$$

Algorithm 12 Construction d'une liste d'échantillons avec un Metropolis-Hastings avec position initiale aléatoire.

Require: $I,\ \tilde{\pi},\ Q$
Tirer $\mathbf{x}^1 \sim \mathcal{N}(0, I_d)$
$X \leftarrow [\mathbf{x}^1]$
for $i \in \{1, ..., I\}$ **do**
    Tirer $x' \sim Q(x'|x^i)$ ▷ Proposer candidat
    $p_A \leftarrow \min\left(1, \frac{\pi(x')}{\pi(x)} \times \frac{Q(x|x')}{Q(x'|x)}\right)$ ▷ Étape d'acceptation-rejet
    Tirer $u \sim \mathcal{U}[0, 1]$
    if $u \leq p_A$ **then**
        $\mathbf{x}^{i+1} \leftarrow \mathbf{x}'$ ▷ Si accepté, ajouter le nouveau candidat à la liste d'échantillons
    else
        $\mathbf{x}^{i+1} \leftarrow \mathbf{x}^i$ ▷ Sinon, répéter l'échantillon dans la liste
    end if
    Ajouter $\mathbf{x}^{i+1}$ à la liste X
end for
Renvoyer X

Le système évolue alors dans l'espace des phases suivant les équations d'Hamilton, qui lient les dérivées premières du Hamiltonien aux dérivées temporelles des positions et momenta :

$$\frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}},\ \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}}$$

En notant $\mathbf{z} = (\mathbf{q}, \mathbf{p})$ les points dans l'espace des phases, on peut ré-écrire le système comme :

$$\frac{d\mathbf{z}}{dt} = J\nabla H(\mathbf{z})$$

avec $J := \begin{pmatrix} 0 & I_N \\ -I_N & 0 \end{pmatrix}$.

En commençant à $(\mathbf{q}_0, \mathbf{p}_0)$ dans l'espace des phases, ce système définit un unique flot Hamiltonien $\mathcal{T}_H^t(\mathbf{q}_0, \mathbf{p}_0)$ qui lie les positions initiales à celles de nimporte quel temps $t > 0$,pourvu que H soit assez régulier ($\mathcal{C}^1$, par exemple) :

$$(\mathbf{q}(t), \mathbf{p}(t)) = \mathcal{T}_H^t(\mathbf{q}_0, \mathbf{p}_0)$$

notons que la transformation $\mathcal{T}_H^t$ est inversible en changeant simplement le signe du temps dans les équations d'Hamilton.

Une autre propriété importante de ces transformations est qu'elles rendent constante la probabilité de se trouver dans un volume élémentaire $d\mathbf{q}\ d\mathbf{p}$. Autrement dit, elles préservent le volume dans l'espace des phases, un résultat aussi connu sous le nom de *théorème de Liouville*. En fait, cela vient d'une propriété encore plus générale qui affirme que ces transformations sont *symplectiques*. En appelant B_t la matrice Jacobienne de la transformation, on a :

$$B_t^T J B_t = J$$

Cela implique la préservation du volume car $\det\left(B_t^T J B_t\right) = \det J \Longrightarrow |\det B_t| = 1$.

De plus, ces propriétés sont conservées quand on approxime les équations d'Hamilton avec une schéma numérique symplectique comme le Leapfrog :

$$\begin{cases} \mathbf{p}_{n+\frac{1}{2}} &= \mathbf{p}_n - \nabla V(\mathbf{q}_n) \times \frac{\delta t}{2}, \\ \mathbf{q}_{n+1} &= \mathbf{q}_n + \nabla K(\mathbf{p}_{n+\frac{1}{2}}) \times \delta t, \\ \mathbf{p}_{n+1} &= \mathbf{p}_{n+\frac{1}{2}} - \nabla V(\mathbf{q}_{n+1}) \times \frac{\delta t}{2}. \end{cases}$$

L'algorithme HMC assimile le logarithme négatif de la distribution cible à un potentiel V le long duquel on peut se déplacer en intégrant les équations d'Hamilton. On commence par ajouter des variables auxiliaires, des momenta, et on décrit l'énergie du système sous forme du Hamiltonien suivant, où l'énergie cinétique correspond cette fois à une forme quadratique avec matrice de masse $\mathcal{M}$:

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2}\mathbf{p}^T \mathcal{M}^{-1} \mathbf{p} + V(\mathbf{q})$$

Le système part d'une position $\mathbf{q}_0$ choisie aléatoirement ou non. A chaque itération, on tire un momentum $\mathbf{p}$ selon la loi de probabilité associée au terme cinétique, qui n'est autre qu'une loi normale multidimensionnelle de moyenne nulle et de matrice de covariance $\mathcal{M}$. Ensuite, le système évolue en suivant les équations d'Hamilton, intégrées numériquement avec un schéma symplectique pendant un pseudo-temps d'intégration $L \times \delta t$, où L désigne le nombre d'étapes de Leapfrog et δt le pas de temps de ce schéma. Le nouveau point $(\mathbf{q}', \mathbf{p}')$ dans l'espace des phases est alors accepté avec probabilité :

$$p_{\text{acc}} = \min\left(1, \exp(-H(\mathbf{q}', \mathbf{p}') + H(\mathbf{q}, \mathbf{p}))\right)$$

Cette dernière étape sert à corriger les erreurs commises par l'utilisation d'un schéma numérique qui ne préserve pas l'énergie totale du système. Puisque le HMC utilise l'information du gradient de la distribution cible, il est beaucoup plus efficace qu'un algorithme de Metropolis-Hastings. En revanche, il est sensible au choix de la matrice de masse $\mathcal{M}$, et des paramètres du schéma Leapfrog L et δt. Certaines versions proposent de simplifier cette phase de réglage en s'appuyant par exemple sur le fait que si la trajectoire commence à revenir sur ses pas, alors il faut arrêter d'intégrer : c'est le principe du No-U-Turn Sampler (Homan and Gelman, 2014). L'algorithme dans sa version de base est résumé dans le pseudo-code 13.

On notera enfin que le choix de viser la distribution canonique dans l'espace des phases tout entier peut être assoupli, étant donné que seul l'espace des positions nous intéresse. Il existe des algorithmes (Robnik and Seljak, 2023; Robnik et al., 2024) bâtis sur ce constat, dont certains ont montré de meilleures performances que le NUTS sur différentes cibles, et utilisés pour résoudre des problèmes d'inférence en cosmologie (Bayer et al., 2023).

Modèles génératifs

La deuxième grande famille d'algorithmes d'échantillonnage étudiée dans cette thèse est celle des modèles génératifs. Ces derniers sont devenus un outil classique pour l'échantillonnage de distributions de probabilités inconnues. Leur succès tient en leur formidable capacité à apprendre de données massives, notamment en exploitant leurs symétries intrinsèques (Bronstein et al., 2021). Plus précisément, dans le cas classique qui nous intéresse ici, on fait l'hypothèse que les données consistent en des réalisations indépendantes d'une distribution de probabilité cible π, qui peut être totalement inconnue. Un modèle génératif est une architecture reposant sur des réseaux de neurones (cf. illustration en Figure 4.17) qui consiste produire des échantillons dont la distribution est proche de la distribution cible d'intérêt (Goodfellow et al., 2016). L'écart entre la distribution du modèle et la cible est mesurée par une certaine distance, ou plutôt une divergence - qui n'est pas symétrique. La phase d'entraînement consiste à ajuster les paramètres du modèle, c'est-à-dire les poids et les biais des neurones qui le composent, afin de minimiser cette divergence. Une fois bien entraîné, le modèle est ainsi capable de générer des échantillons originaux qui ressemblent à s'y méprendre à ceux issus du jeu d'entraînement.

La littérature foisonne de modèles différents, chacun avec ses avantages et inconvénients. Donnons quelques exemples utiles à la suite de ce manuscrit.

- Les *Autoencodeurs* (Lecun, 1987; Kramer, 1991) sont composés de deux parties : un Encodeur, qui, partant d'un échantillon d'entraînement, l'envoie sur un espace de plus petite dimension, l'espace latent ; puis un Décodeur, qui se charge de renvoyer cette représentation vers un espace

Algorithm 13 Construction d'une liste d'échantillons avec un HMC avec position initiale aléatoire.

Require: $I,\ \varepsilon,\ L,\ H,\ \nabla V, \mathcal{M}$

Tirer $\mathbf{x}^1 \sim \mathcal{N}(0, I_d)$
Tirer $\mathbf{p}^1 \sim \mathcal{N}(0, \mathcal{M})$
$X \leftarrow [\mathbf{x}^1]$
for $i \in \{1, ..., I\}$ **do**
    for $j \in \{1, ..., L\}$ **do** ▷ Intégration Leapfrog pour proposer le prochain candidat
        $\mathbf{p}^{j+1} \leftarrow \mathbf{p}^j - \frac{\varepsilon}{2}\nabla V(\mathbf{x}^j)$
        $\mathbf{x}^{j+1} \leftarrow \mathbf{x}^j + 2\varepsilon\mathbf{p}^{j+1}$
        $\mathbf{p}^{j+1} \leftarrow \mathbf{p}^{j+1} - \frac{\varepsilon}{2}\nabla V(\mathbf{x}^{j+1})$
    end for
    $p_A \leftarrow \min\{1, \exp(-(H(\mathbf{x}^{L+1}, \mathbf{p}^{L+1}) - H(\mathbf{x}^1, \mathbf{p}^1)))\}$ ▷ Étape d'acceptation-rejet
    Tirer $u \sim \mathcal{U}[0, 1]$
    if $u \leq p_A$ **then**
        Ajouter $\mathbf{x}^{L+1}$ à la liste X ▷ Si accepté, ajouter le nouveau candidat à la liste d'échantillons
        $\mathbf{x}^1 \leftarrow \mathbf{x}^{L+1}$
    else
        Ajouter $\mathbf{x}^1$ à la liste X ▷ Sinon, répéter l'échantillon précédent dans la liste
    end if
    Tirer $\mathbf{p}^1 \sim \mathcal{N}(0, \mathcal{M})$ ▷ Tirer un nouveau momentum aléatoire
end for
Renvoyer X

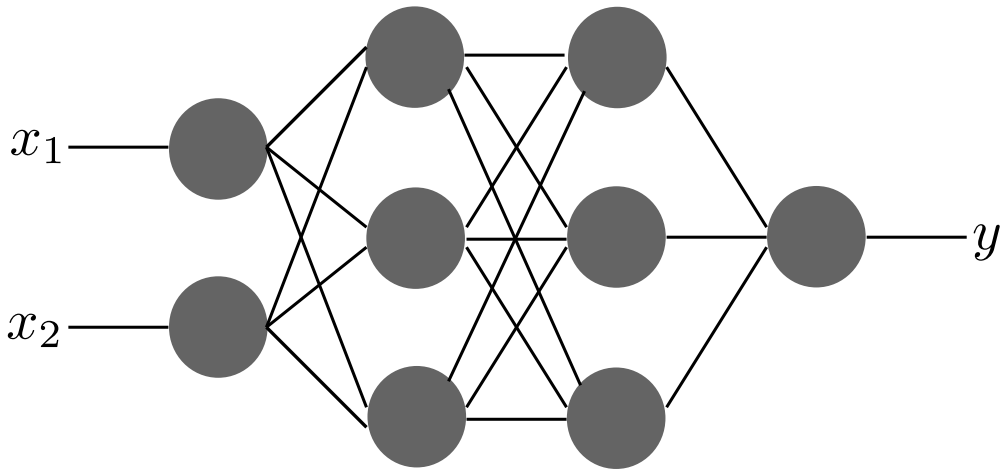


Figure 4.17: Exemple de réseau de neurones entièrement connecté qui transforme une donnée d'entrée (x_1, x_2) comportant deux dimension en une donnée de sortie y qui n'en compte qu'une seule. Le réseau est constitué d'une couche d'entrée avec 2 neurones, de deux couches cachées avec 3 neurones et d'une couche de sortie avec 1 neurone. Sachant des entrées $e_1, ..., e_n$, un neurone renvoie une sortie de la forme $o(e_1, ..., e_n) = s\left(b + \sum_{k=1}^{n} a_k e_k\right)$. La fonction s est une fonction dite *d'activation* servant à introduire de la non-linéarité. Les paramètres $a_1, ..., a_k$ sont appelés les *poids* et le paramètre b est appelé le *biais*. Une telle architecture pourrait par exemple être utilisée pour associer à chaque vecteur de $\mathbb{R}^2$ un nombre représentant une énergie.

de même dimension que celui d'origine. Pour échantillonner, il suffit de partir d'un échantillon de l'espace latent et de le faire passer à travers le Décodeur. Cette classe de modèles est intéressante notamment pour sa partie Encodeur, qui va nous servir dans l'architecture au cœur de cette thèse.

- Une des premières classes de modèles à avoir attiré l'attention de la communauté furent les *Generative Adversarial Networks* (GANs) (Goodfellow et al., 2014). Leur performance en génération d'images a constitué une véritable révolution dans le domaine de l'Intelligence Artificielle. Les GANs sont composés de deux réseaux de neurones antagonistes : le Générateur produit les échantillons et le Discriminateur tente de déterminer, étant donné un échantillon d'entrée, si ce dernier provient du vrai jeu de données ou s'il a été produit par le Générateur. Le but du Générateur est donc de tromper le Discriminateur, alors que le Discriminateur a pour objectif de confondre le Générateur. Bien que très puissants, les GANs sont prompts à des phénomènes de mode-collapse, même dans le cas de distributions très simples (Eghbal-zadeh et al., 2019). De plus, ils sont délicats à entraîner. Ces différents inconvénients nous ont conduit à étudier d'autres types d'architectures.

- Plus récemment, l'état de l'art en terme de génération d'images a été atteint grâce aux *modèles de diffusion* (Sohl-Dickstein et al., 2015). Ils consistent à partir d'une image de l'échantillon d'entraînement et de la perturber progressivement en y ajoutant du bruit. Inversement, une fois entraîné, le modèle part d'un bruit blanc Gaussien quelconque et le débruite jusqu'à obtenir une nouvelle image. Ces modèles reposent techniquement sur l'utilisation de chaînes de Markov, qu'il faut être capable d'inverser, pour bruiter un signal. Ainsi, le processus génératif en lui-même est coûteux, puisqu'il faut inverser l'évolution temporelle de cette chaîne de Markov pour produire un échantillon. C'est ce coût numérique non négligeable qui, même s'il peut être en partie amorti (Song et al., 2022), peut nous pousser à considérer d'autres types de modèles.

- Enfin, les *modèles de flots* (Tabak and Vanden-Eijnden, 2010; Dinh et al., 2014; Papamakarios et al., 2022) au cœur de cette thèse consiste à transformer une distribution de base, souvent une Gaussienne, en la distribution cible au moyen d'une bijection bien choisie. Nous verrons dans la dernière partie comment il est possible de les adapter au cadre bayésien.

Un algorithme de Monte Carlo non réversible pour l'inférence des conditions initiales de l'Univers

Nous introduisons PDMC-BORG, un échantillonneur non-réversible basé sur des processus de Markov déterministes par morceaux, pour l'inférence Bayésienne du champ de fluctuations de densité primordial. L'objectif est de comparer cet algorithme à un HMC de référence.

Vers la non-réversibilité

Pour que la chaîne de Markov converge vers la distribution cible, il est nécessaire que le flot de probabilité entrant sur chaque état soit égal au flot de probabilité sortant de ce même état, une condition appelée *équilibre global*. Notons K le noyau de Markov décrivant la transition d'un état vers un autre. Mathématiquement, elle s'écrit de la façon suivante :

$$\pi = \pi K \tag{4.4}$$

Une condition suffisante pour que l'équation 4.4 soit vérifiée est que la chaîne de Markov satisfasse une condition encore plus forte, imposant que le flot entre chaque paire d'états (x_1, x_2) se compense. Cette condition s'appelle l'*équilibre détaillé* et elle conduit à ce que la chaîne de Markov soit réversible, en ajoutant une contrainte de symétrie locale tout à fait artificielle. Mathématiquement, elle s'écrit :

$$\pi(dx_1)K(x_1, dx_2) = \pi(dx_2)K(x_2, dx_1) \tag{4.5}$$

Un tel comportement présente une interprétation physique : à l'échelle microscopique, on fait classiquement l'hypothèse que les processus sont en effet réversibles, bien que le phénomène macroscopique émergent correspondant ne le soit pas nécessairement. D'un point de vue échantillonnage, la condition d'*équilibre détaillé*, sur laquelle repose beaucoup de méthodes MCMC - algorithmes de Metropolis, Metropolis-Hastings et Hamiltonian Monte Carlo, notamment - peut constituer une limitation. En imposant des contraintes locales et artificielles de symétrie, elle peut être à l'origine d'une exploration lente de l'espace des paramètres, selon un processus diffusif. Pour des distributions en grande dimension et/ou présentant de nombreux modes, ce défaut peut devenir une véritable malédiction, empêchant une exploration suffisante du paysage énergétique associé à la cible en des temps raisonnables. Cet effet est particulièrement visible sur des distributions anisotropiques dans des espaces de grande dimension, voir par exemple (Michel et al., 2020).

Une première étape importante a été franchie avec l'introduction du concept de *lifting* (Chen et al., 1999; Diaconis et al., 2000). Ces méthodes reposent sur le constat que le rejet d'un échantillon avec une étape de Metropolis-Hastings survient quand la zone explorée a une faible masse de probabilité, voire qu'elle correspond à une configuration impossible. D'où l'idée d'introduire une variable auxiliaire, dite de lifting, comme c'est le cas pour le HMC, afin d'amener le processus dans des zones où la probabilité d'acceptation sera plus élevée. Cela a par exemple abouti à une version liftée de l'algorithme de Metropolis-Hastings (Gustafson, 1998; Turitsyn et al., 2011). L'objectif est de remplacer les rejets dus à l'étape de Metropolis par un ré-échantillonnage de la variable auxiliaire.

Ces algorithmes, brisent effectivement la condition d'équilibre détaillée mais reposent toujours sur des symétries locales artificielles, bien que sous une forme moins contraignante. L'introduction d'algorithmes dits *Event Chain Monte Carlo* dans le contexte de la Physique statistique a permis une avancée décisive. Dans ce cas, il n'y a plus de rejet mais une succession (une *chaîne*) d'événements qui correspondent à des changements de direction du processus, lequel suit une trajectoire balistique entre ces événements (Bernard et al., 2009; Michel et al., 2014). Puisqu'ils renoncent à introduire des symétries artificielles, ces algorithmes perdent leur propriété de réversibilité et sont bien plus performants dans leur exploration.

Formellement, ces algorithmes reposent sur des processus stochastiques et leur utilisation dans un cadre d'échantillonage statistique a depuis été caractérisée par la communauté mathématique (Bierkens et al., 2016; Alexandre Bouchard-Côté and Doucet, 2018; Michel et al., 2020) comme une certaine instance de processus stochastiques Markoviens.

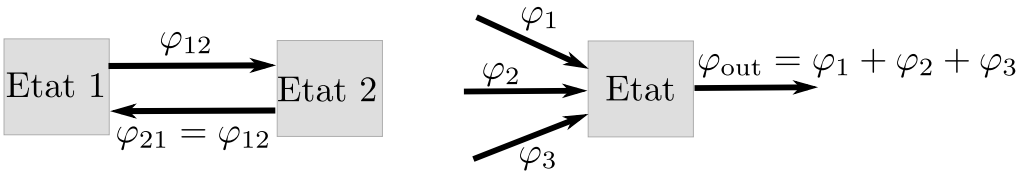


Figure 4.18: A gauche, une illustration de la condition d'*équilibre détaillé*. A droite, une illustration de la condition d'*équilibre global*. Les φ représentent des flots de probabilité. On visualise clairement que la première condition implique la seconde.

Implémentation d'un algorithme Piecewise Deterministic Markov Chain pour la cosmologie

Une des façons de construire des échantillonneurs robustes et s'adaptant bien à la grande dimension est l'utilisation de processus stochastiques de type *Piecewise Deterministic Markov Processes* (PDMP). Un PDMP (Davis, 1984) est un processus stochastique à temps continu $(Z_t)_{t\geq 0}$, à valeurs notées $\mathbf{z}_t$, qui évolue de manière déterministe entre des événements aléatoires qui sont tirés selon un processus de Poisson inhomogène. Les PDMP consistent donc en la donnée de quatre quantités :

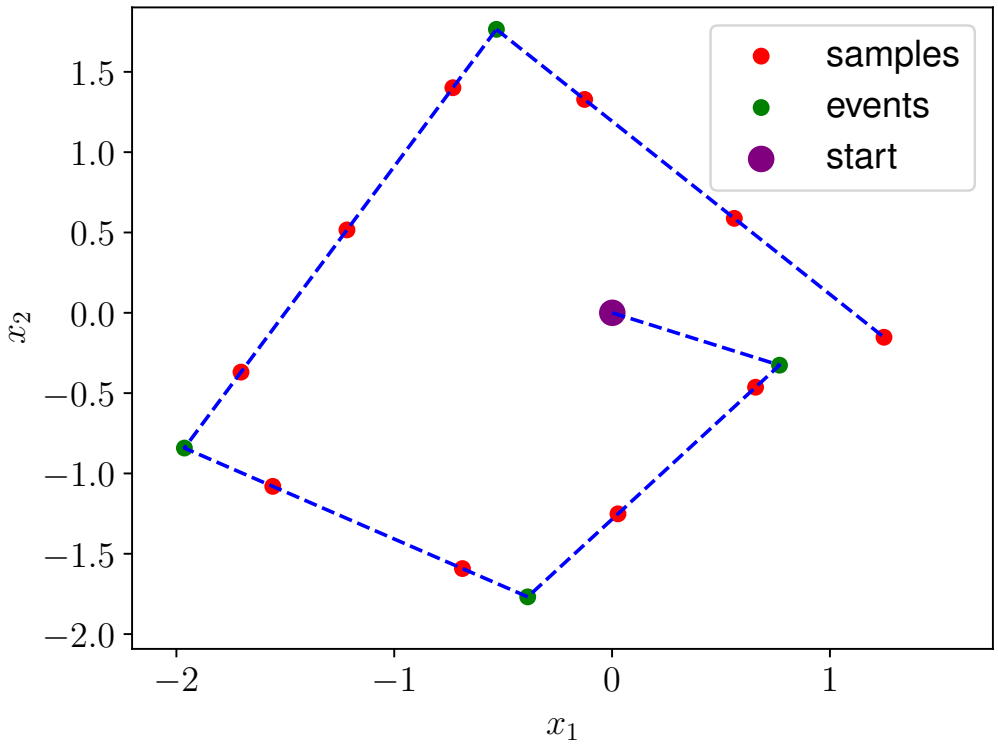


Figure 4.19: Trajectoire dessinée par un algorithme PDMC avec mouvement ballistique entre événements aléatoires et échantillonnage régulier le long de la trajectoire. Ici, la cible est une loi normale standard $\mathcal{N}(0, I_2)$.

- une condition initiale $\mathbf{z}_0$;
- une dynamique déterministe gouvernée par une Équation Différentielle Ordinaire entre les événements aléatoires
$$\frac{d\mathbf{z}_t}{dt} = \phi(\mathbf{z}_t),$$
qui induit un flot différentiel Φ ;
- des temps d'événements décidés par un processus de Poisson inhomogène de taux $\lambda(\mathbf{z}_t)$;
- un noyau de transition $Q(\cdot|\mathbf{z}_t)$ pour décider de la modification des $\mathbf{z}_t$ aux événements.

Plusieurs algorithmes de Monte Carlo non réversibles basées sur des PDMP ont été développés. Ces algorithmes consistent à explorer la distribution cible en suivant des mouvements balistiques entre chaque événement, où la direction est re-tirée. Parmi les plus populaires, on peut mentionner le Bouncy Particle Sampler (Alexandre Bouchard-Côté and Doucet, 2018) dont le changement de direction consiste en une réflexion le long du plan orthogonal dessiné localement par le vecteur gradient. Quant à l'Automatic Zig-Zag sampler (Bierkens et al., 2016), il s'agit d'inverser aux événements le signe d'une composante du vecteur direction, la trajectoire dessinant alors des *zig-zags* dans l'espace des paramètres. Il faut donc, dans tous ces algorithmes, étendre l'espace des paramètres d'intérêt, notés $\mathbf{x}$, avec des variables additionnelles, pouvant par exemple représenter la direction de la trajectoire, notée $\mathbf{e}$. Les temps d'événements sont décidés à partir de la simulation d'un processus de Poisson inhomogène dont le taux dépend de la dérivée directionnelle du processus. Un algorithme PDMC consiste donc en la donnée de trois ingrédients :

1. un état initial $(\mathbf{x}_0, \mathbf{e}_0)$;

2. une dynamique déterministe gouvernée par une Équation Différentielle Ordinaire entre les événements aléatoires
$$\frac{d\mathbf{z}_t}{dt} = \phi(\mathbf{z}_t),$$
qui induit un flot différentiel Φ. Dans les algorithmes PDMC classiques, ce sont souvent des translations (Bierkens et al., 2016; Michel et al., 2020; Alexandre Bouchard-Côté and Doucet, 2018) ;

3. un processus de Poisson inhomogène avec taux $\lambda(t)$;

4. un noyau $Q(\mathbf{x})$ pour décider des changements de direction aux événements.

L'échantillonnage se fait ensuite uniformément le long de la trajectoire continue dessinée par le processus $(\mathbf{x}_t)_{t\geq 0}$, comme illustré sur la Figure 4.19.

Parmi les différentes méthodes PDMC, notre choix s'est porté sur la classe d'algorithmes *Forward Event Chain Monte Carlo* (Michel et al., 2020), qui a montré des performances supérieures au Bouncy Particle Sampler et à l'Automatic Zig-Zag sampler, notamment concernant l'échantillonnage de distributions fortement anisotropiques en grande dimension (Michel et al., 2020). Les algorithmes de Forward Event Chain généralisent les méthodes ECMC. Leur force repose sur l'introduction de stochasticité aux événements. En effet, à chaque événement, la direction est re-tirée en décomposant aléatoirement le vecteur vélocité selon une composante orthogonale au gradient et une composante parallèle, comme illustré sur la Figure 4.20. Plus précisément, en notant $\mathbf{e}_0$ la direction incidente, la direction sortante s'écrit :
$$\mathbf{e}_1 = a\mathbf{n}_{\text{perp}} - b\mathbf{n}_{\text{par}}$$
avec :

- $a = \nu^{1/(d-1)}$, $b = \sqrt{1-a^2}$ et ν un nombre tiré uniformément entre 0 et 1 ;
- $\mathbf{n}_{\text{par}} = \frac{\nabla E(\mathbf{x})}{||\nabla E(\mathbf{x})||}$ le vecteur gradient normalisé ;
- $\mathbf{n}_{\text{perp}} = \frac{\mathbf{e}_0 - \langle \mathbf{n}_{\text{par}}, \mathbf{e}_0 \rangle \mathbf{n}_{\text{par}}}{||\mathbf{e}_0 - \langle \mathbf{n}_{\text{par}}, \mathbf{e}_0 \rangle \mathbf{n}_{\text{par}}||}$ la projection normalisée de la direction incidente sur le plan orthogonal au gradient.

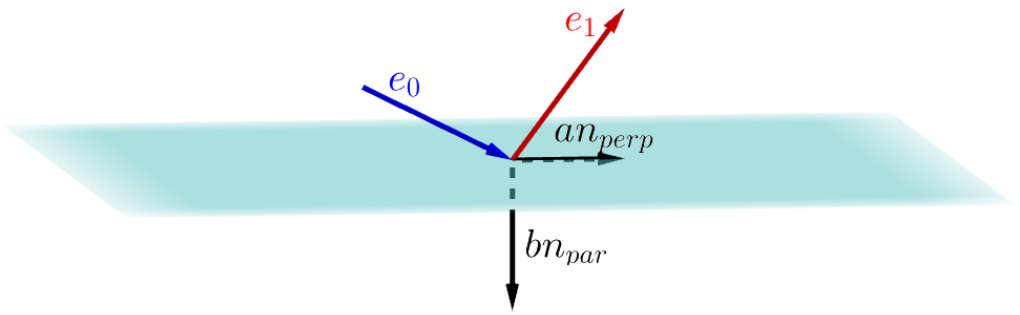


Figure 4.20: Changement de direction à un événement avec un algorithme Forward Event Chain Monte Carlo. En bleu ciel, le plan orthogonal au vecteur gradient, qui est de même sens et même direction que $\mathbf{n}_{\text{par}}$. Les nombres a et b sont aléatoires et diffèrent a priori à chaque événement.

Afin d'assurer l'ergodicité de l'exploration, une étape de rafraîchissement est introduite. Plusieurs stratégies peuvent être considérées. On peut par exemple retirer aux événements la composante perpendiculaire au gradient, avec une certaine probabilité p_{ref} qui devient un hyperparamètre de

l'algorithme. Dans ce cas, une option correcte (Michel et al., 2020) est de tirer $\mathbf{u} \sim \mathcal{N}(0, I_d)$ et de prendre :

$$\mathbf{n}_{\text{perp}} = \frac{\mathbf{u} - \langle \mathbf{u}, \mathbf{n}_{\text{par}} \rangle \mathbf{n}_{\text{par}}}{||\mathbf{u} - \langle \mathbf{u}, \mathbf{n}_{\text{par}} \rangle \mathbf{n}_{\text{par}}||}.$$

Une autre possibilité, populaire dans la communauté PDMC (Michel et al., 2014, 2020; Monemvassitis et al., 2023) consiste à rafraîchir à intervalle de temps fixé δt_{ref} la direction selon la loi uniforme sur la sphère unité. Dans ce cas, δt_{ref} devient à son tour un hyperparamètre. En pratique, cela revient à tirer $\mathbf{u} \sim \mathcal{N}(0, I_d)$ et à prendre :

$$\mathbf{e} = \frac{\mathbf{u}}{||\mathbf{u}||}.$$

L'étape la plus difficile concerne la simulation du processus de Poisson inhomogène, pour tirer les temps des événements. Pour cela, nous avons besoin de trouver une borne sur le taux $\lambda(t) = \max(0, \langle \nabla E(\mathbf{x} + t\mathbf{e}), \mathbf{e} \rangle)$. Une stratégie possible (Corbella et al., 2022) est de trouver une borne locale. On fait alors l'hypothèse forte suivante :

Le paysage énergétique à explorer est convexe autour du mode cible.

Cela revient essentiellement à supposer que la distribution postérieure cible est log-concave - par exemple, une Gaussienne. Cette hypothèse est justifiée par des expériences numérique minutieuses sur cette cible : il apparaît en effet qu'au voisinage du mode cible, la dérivée directionnelle est croissante selon chaque direction explorée par l'algorithme. Ce constat permet alors de trouver une borne locale de λ dans un intervalle de temps de taille $t_{\max}$. Ainsi, si le système est à $(\mathbf{x}_t, \mathbf{e}_t)$ à l'instant t, alors une telle borne est donnée par $\lambda_t^* = \max(0, \langle \nabla E(\mathbf{x}_t + t_{\max}\mathbf{e}_t), \mathbf{e}_t \rangle)$. L'idée consiste alors à utiliser une telle borne pour se ramener à la simulation d'un processus de Poisson homogène, que l'on peut ensuite *thinner* pour obtenir la bonne loi (Lewis and Shedler, 1979). Partant d'un point $(\mathbf{x}, \mathbf{e})$, on génère ainsi le prochain temps d'événement $t_e > 0$:

- si $t_e < t_{\max}$, alors on parle de *vrai événement* et le processus est amené à la position $\mathbf{x} + t_e\mathbf{e}$ **avec** changement de direction ;
- sinon, on parle de *faux événement* et le processus est amené à la position $\mathbf{x} + t_{\max}\mathbf{e}$ **sans** changement de direction.

Pour être tout à fait précis, l'algorithme PDMC-BORG que nous proposons est composé de deux phases : une phase de recherche du mode, qui ne préserve pas la mesure cible mais permet de converger rapidement vers la zone d'intérêt, et une phase de relaxation dont une version avec rafraîchissement aux événements est décrite par l'Algorithme 30.

Résultats et comparaisons avec un HMC

Nous présentons ici des résultats d'expériences numériques consistant en la comparaison des performances du HMC traditionnellement utilisé dans BORG avec notre algorithme PDMC. Pour que la comparaison soit pertinente, il faut que les algorithmes soient testés dans un régime suffisamment *difficile*. La difficulté du problème est lié à la taille caractéristique du déplacement lié au modèle gravitationnel utilisé, ici la Lagrangian Perturbation Theory (LPT). Moralement, toute résolution inférieure à cette taille caractéristique sera dans l'incapacité de voir les effets non-linéaires causés ce déplacement. De plus, le niveau de bruit se combine à l'effet précédent. En effet, dans le cas où le bruit est très fort, alors le problème redevient Gaussien, et peut même compenser une très forte résolution. A l'inverse, si le bruit est très faible, alors l'obtention de la distribution postérieure cible ne peut pas se faire en inversant simplement la matrice de covariance du problème.

Pour travailler dans un cas non trivial, on se place alors dans une boîte cubique d'univers de côté $L = 677.7$ Mpc$/h$ et subdivisée en $d = 32^3 = 32768$ cellules, ce qui constitue la dimension du problème.

Algorithm 14 Échantillonage de N_{samples} échantillons, séparés par un intervalle temporel fixé $\delta t_{\text{samples}}$, avec un algorithme PDMC et une stratégie de thinning local et un rafraîchissement de la direction aux événements.

Require: $t_{\max},\ \delta t_{\text{samples}}, N_{\text{samples}}$

Tirer $\mathbf{x} \sim \mathcal{N}(0, I_d)$
Tirer $\mathbf{u} \sim \mathcal{N}(0, I_d)$
$\mathbf{e} \leftarrow \frac{\mathbf{u}}{||\mathbf{u}||}$
$t \leftarrow 0$ ▷ temps du processus
$n_{\text{samples}} \leftarrow 0$ ▷ nombre d'échantillons
$t_{\text{tosample}} \leftarrow \delta t_{\text{samples}}$ ▷ temps avant prochain échantillon
while $n_{\text{samples}} < N_{\text{samples}}$ **do**
    Tirer temps du prochain événement dt ▷ avec thinning local
    isEvent ← False
    $t_{\text{toevent}} \leftarrow dt$
    while not isEvent **do**
        if $t_{\text{toevent}} < t_{\text{tosample}}$ **then**
            $t \leftarrow t + t_{\text{toevent}}$
            $t_{\text{tosample}} \leftarrow t_{\text{tosample}} - t_{\text{toevent}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{toevent}}\mathbf{e}$
            if $dt < t_{\max}$ (C'est un vrai événement) **then**
                Changer la direction $\mathbf{e}$
            end if
            isEvent ← True
        else
            $t \leftarrow t + t_{\text{tosample}}$
            $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{tosample}}\mathbf{e}$
            $\mathbf{s}_{n_{\text{samples}}} \leftarrow \mathbf{x}$
            $n_{\text{samples}} \leftarrow n_{\text{samples}} + 1$
            $t_{\text{toevent}} \leftarrow t_{\text{toevent}} - t_{\text{tosample}}$
            $t_{\text{tosample}} \leftarrow \delta t_{\text{samples}}$
        end if
    end while
end while
Renvoyer $\mathbf{s}_1, ..., \mathbf{s}_{N_{\text{samples}}}$

Les données astronomiques sont générées artificiellement et on utilise une vraisemblance Gaussienne avec un bruit d'écart-type 0.01, ainsi qu'un a priori Gaussien standard. L'objectif est de comparer les performances du HMC avec celles du PDMC à :

- à nombre d'évaluations de gradients fixé, cette opération étant de loin la plus coûteuse puisqu'elle fait intervenir le modèle d'évolution cosmologique. Ces tests ont pour but de comparer leurs performances *numériques* ;
- à nombre d'événements fixé, pour comparer leurs performance en terme de *dynamiques*, c'est-à-dire leur capacité à explorer efficacement l'espace des paramètres. Cela sert à comparer les algorithmes dans un cas idéal pour le PDMC où le calcul du temps d'événement est analytique et correspond à exactement une évaluation de gradient.

Métriques. Afin d'évaluer la qualité des échantillons produits et la vitesse de convergence vers la cible, plusieurs outils sont utilisés :

1. Comparaisons visuelles. La machinerie BORG offre la possibilité de visualiser le champ final correspondant aux conditions initiales échantillonnées. Si le bruit est suffisamment faible, alors l'observation de ce champ devrait être comparable entre le PDMC et le HMC : leur structure filamentaire devrait suivre les mêmes tendances.

2. Moyennes et histogrammes cumulées des échantillons le long de chacune des dimensions. Étant donnée une suite de points $(x_1, ..., x_n) \in \mathbb{R}^n$, il peut être utile d'étudier les statistiques d'ordre 1 de cette série. En particulier, la moyenne empirique est définie comme :

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

et la variance empirique comme :

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{\mu})^2.$$

Ici, les x_i représentent une quantité scalaire associée à un échantillon, qui est une grandeur vectorielle, par exemple la valeur d'une de ses dimensions. Un test simple peut être de tracer l'évolution de cette quantité à mesure que le nombre n d'échantillons générés par l'algorithme augmente.

3. Comparaison des spectres de puissance. Il s'agit d'un outil classique en cosmologie qui mesure les inhomogénéités de matière dans l'univers en fonction de l'échelle d'observation. En particulier, on définit la fonction de corrélation à 2 points de la manière suivante:

$$\xi(\mathbf{r}) = \langle \delta(\mathbf{x}), \delta(\mathbf{x} + \mathbf{r}) \rangle = \int_V \delta(\mathbf{x}) \delta(\mathbf{x} + \mathbf{r}) d^3\mathbf{x}$$

Le spectre de puissance correspond alors à la transformée de Fourier de cette fonction de corrélation, c'est-à-dire :

$$\xi(\mathbf{r}) = \frac{V}{(2\pi)^3} \int P(\mathbf{k}) e^{-i\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{k}$$

avec la relation fondamentale

$$P(\mathbf{k}) = |\hat{\xi}(\mathbf{k})|^2$$

En pratique, cette quantité est estimé en utilisant une technique de boîtes. En représentant les spectres de puissance successifs associés à chaque échantillon, on peut avoir une idée de la convergence de l'algorithme, ce qui se traduit par une suite de spectres de puissance oscillant autour d'un spectre de référence, celui obtenu avec les conditions initiales artificielles générées pour le problème, par exemple.

4. Autocorrélation le long de chacune des dimensions. On se rappelle qu'étant donnée une suite de points $(x_1, ..., x_n)$ et une fonction h, l'autocorrélation au lag $k \in \{0, ..., n-1\}$ est définie comme

$$C_h(k) = \frac{1}{\hat{\sigma}^2(n-k)} \sum_{i=1}^{n-k} (h(x_i) - \hat{\mu})(h(x_{i+k}) - \hat{\mu})$$

Ici, $\hat{\mu}$ and $\hat{\sigma}^2$ désignent respectivement la moyenne et la variance empirique de $(h(x_1), ..., h(x_n))$. L'étude des autocorrélations relève de la plus grande importance. La capacité d'un algorithme d'échantillonnage de type Monte Carlo à décorréler rapidement les échantillons produits est cruciale. En effet, il s'agit idéalement de produire des échantillons indépendants et identiquement distribués selon la postérieure cible. Un tel comportement est impossible à atteindre en pratique puisque l'échantillon suivant est à chaque étape tiré en fonction de l'état courant.

Calibrage des algorithmes. La partie délicate dans le HMC consiste en le réglage des différents paramètres, à savoir la matrice de masse $\mathcal{M}$, le nombre d'étapes du Leapfrog L ainsi que le pas de temps de ce schéma δt. L'objectif est d'obtenir une décorrélation la plus rapide possible des observables d'intérêt. On fixe ici la matrice de masse à la matrice identité. Quant à L et δt, une bonne heuristique est de les ajuster de sorte que le taux d'acceptation soit environ 0.65 (Beskos et al., 2013).

Quant au PDMC, il convient de régler seulement deux paramètres, ce qui constitue déjà un avantage : la probabilité p_{ref} avec laquelle un rafraîchissement a lieu lors d'un événement ainsi que le paramètre temporel $t_{\max}$ qui gouverne le thinning du processus de Poisson inhomogène. On rappelle en effet que l'on suppose que le problème d'inférence à résoudre est convexe. L'objectif est de choisir ces paramètres de sorte à minimiser le nombre d'évaluations de gradient par vrai événement. Pour régler ces deux paramètres, on peut effectuer une simulation test en partant du régime thermalisé. Les valeurs optimales pour ce régime amènent à environ 4.5 évaluations de gradient par événement.

Résultats des expériences. S'il est difficile d'observer des différences majeures de comportement entre les algorithmes à budget computationnel fixé, c'est parce que le problème d'inférence statistique sous-jacent présente de bonnes propriétés dans la région d'intérêt : il semble bien convexe, d'une part, et d'autre part il est plutôt isotropique. Nous présentons ici les résultats d"expériences numériques afin de comparer PDMC-BORG avec l'algorithme de référence HMC-BORG. Les trajectoires partent du même point, proche de 0. La phase de mixage représente environ 400 échantillons pour les deux algorithmes. En tout, 20000 échantillons sont générés par chacun des algorithmes. Pour s'assurer de la convergence des deux algorithmes vers les mêmes observables, on représente dans la Figure 4.21 les histogrammes cumulés le long de quelques dimensions. Ces graphiques semblent identiques pour PDMC et HMC-BORG, ce qui est le comportement attendu. Enfin, on représente dans la Figure 4.22 un zoom sur les autocorrélations obtenues le long de quelques dimensions. A **budget computationnel fixé**, cela illustre le comportement similaire entre PDMC et HMC-BORG sur ce problème. De façon intéressante, on peut aussi représenter les autocorrélations **dans un scénario idéal** pour PDMC-BORG, c'est-à-dire dans une situation où le calcul d'un événement correspond à une seule évaluation de gradient. Cela revient à zoomer sur les schémas d'autocorrélation du HMC d'un facteur 4.5 qui correspond au nombre moyen d'évaluations de gradient par calcul d'événement pour PDMC-BORG. Dans ce cas, PDMC-BORG semble présenter de meilleurs résultats que HMC-BORG. Cela met en lumière l'efficacité d'une exploration balistique de l'espace des paramètres. Pour ce type de problème Bayésien en grande dimension, les algorithmes de Monte-Carlo non-réversibles constituent donc une alternative intéressante aux méthodes MCMC réversibles classiquement utilisées par la communauté des cosmologistes.

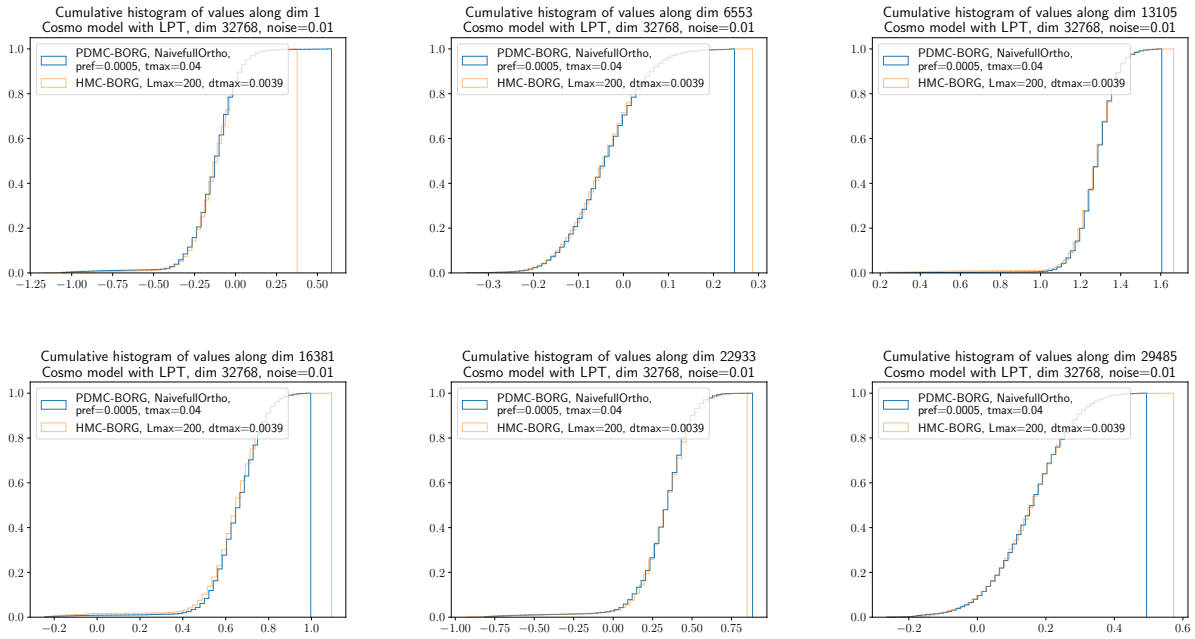


Figure 4.21: Histogrammes cumulés le long de quelques dimensions obtenus avec des échantillons produits par les deux algorithmes.

Résumé des principales contributions

- Implémentation d'un algorithme de Monte Carlo non-réversible pour l'inférence Bayésienne des conditions initiales de l'univers, autorisant à terme l'intégration définitive de ce code dans la machinerie BORG.
- Arguments numériques et heuristiques concernant l'efficacité d'un tel algorithme dans un contexte de grande dimension avec une distribution cible compliquée.
- Comparaisons numériques entre ce nouvel algorithme et le traditionnel Hamiltonian Monte Carlo utilisé par les cosmologistes du consortium Aquila.

Modèles génératifs interprétables pour l'inférence de paramètres cosmologiques

Dans ce travail, nous étudions l'utilisation de la dynamique Hamiltonienne pour construire des modèles génératifs robustes et interprétables. Nous proposons une version à énergie cinétique fixée des Neural Hamiltonian Flows et évaluons la performance de cette famille de modèles sur des problèmes de génération d'images. Enfin, nous testons leur utilisation pour l'inférence Bayésienne en cosmologie. Ce travail a donné lieu à la rédaction d'un article publié à la conférence AISTATS 2024 (Souveton et al., 2024). Aussi, une version publique du code est disponible[4].

Modèles de flots et interprétabilité

Dans cette thèse, nous nous sommes tout particulièrement intéressés à des modèles de flots dits *Normalizing Flows* (Tabak and Vanden-Eijnden, 2010; Dinh et al., 2014; Papamakarios et al., 2022). Ces derniers ont remporté un grand succès dans la communauté du fait de leur robustesse. Ces méthodes

[4] https://plmlab.math.cnrs.fr/stoch-algo-phys/generative-models/fixed-kinetic-NHF/-/tree/main?ref_type=heads

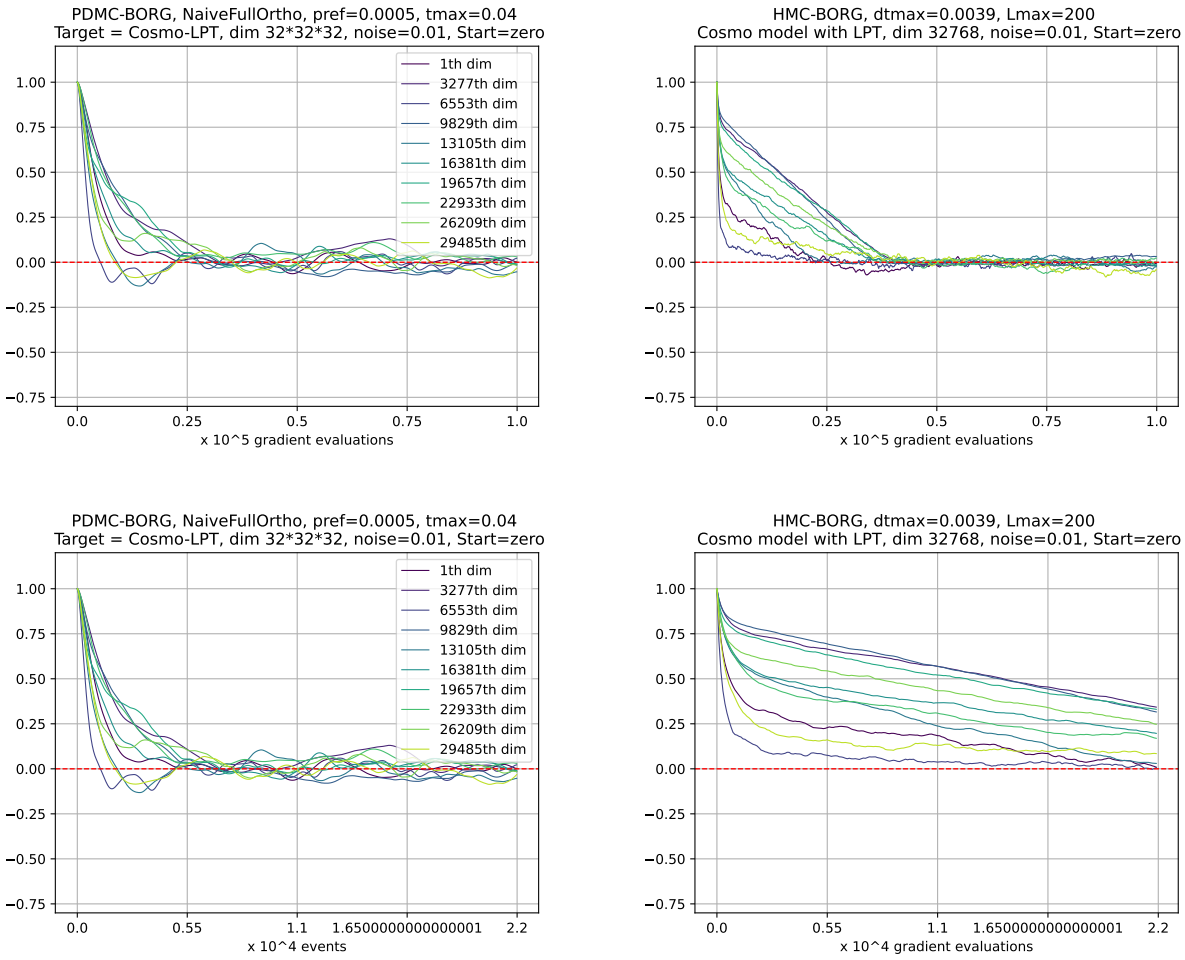


Figure 4.22: Autocorrélations le long de quelques dimensions à budget computationnel fixé (en-haut) et dans un scénario idéal (en-bas) pour PDMC-BORG (à gauche) et HMC-BORG (à droite).

consistent à transformer une distribution simple de densité π_0, typiquement une Gaussienne, en une distribution cible plus compliquée au moyen d'une suite finie de transformations inversibles $\mathcal{T}_1, ..., \mathcal{T}_L$. Une fois le modèle entraîné, il est possible d'inverser la dynamique apprise pour générer des échantillons de la cible partant de la distribution Gaussienne de base, aussi appelée *a priori*. Si $X = \mathcal{T}_L \circ ... \circ \mathcal{T}_1(Z)$, où $Z \sim \pi_0$, alors la densité de X s'écrit $m(x) = \pi_0\left(\mathcal{T}_1^{-1} \circ ... \circ \mathcal{T}_L^{-1}(x)\right) \times \prod_{k=1}^{L} \left|\det J_{\mathcal{T}_k^{-1}}(x)\right|$. Appelons Θ les paramètres du modèle. Le but est de minimiser la divergence de Kullback-Leibler entre la distribution cible π et celle du modèle m par rapport à Θ, c'est à dire minimiser la quantité suivante :

$$\begin{aligned}\mathcal{L}(\Theta) &= \mathbf{E}_\pi\left[\log \pi(X) - \log m(X;\Theta)\right] \\ &= -\mathbf{E}_\pi\left[\log \pi_0\left(\mathcal{T}_1^{-1} \circ ... \circ \mathcal{T}_L^{-1}(X;\Theta)\right) + \sum_{k=1}^{L} \left|\det J_{\mathcal{T}_k^{-1}}(X;\Theta)\right|\right] + C.\end{aligned}$$

On utilise alors un estimateur Monte Carlo pour estimer cette quantité à partir d'échantillons $x_1, ..., x_N$ issus du jeu d'entraînement :

$$\mathcal{L}(\Theta) \approx \frac{-1}{N} \sum_{i=1}^{N} \left[\log \pi_0\left(\mathcal{T}_1^{-1} \circ ... \circ \mathcal{T}_L^{-1}(x_i;\Theta)\right) + \sum_{k=1}^{L} \left|\det J_{\mathcal{T}_k^{-1}}(x_i;\Theta)\right|\right]$$

et la minimiser à l'aide d'un algorithme classique de descente de gradient, comme Adam (Kingma and

Ba, 2015).

Dans ces architectures, le choix de la chaîne de transformations inversibles est crucial. Il est guidé par de multiples considérations :

1. les transformations doivent être *inversibles* et suffisamment *régulières* ;
2. elles doivent aussi être *peu coûteuse* en calcul que possible. La partie exigeante concerne le calcul du déterminant Jacobien de la transformation. En pratique, on choisit des transformations dont la matrice Jacobienne possède de bonnes propriétés - diagonales, triangulaires -, comme c'est le cas avec les modèles Real NVP (Dinh et al., 2017) ;
3. la chaîne de transformations doit être suffisamment *expressive* afin d'être capable d'envoyer la Gaussienne sur n'importe quelle type de distribution cible ;
4. enfin, on voudrait que la transformation apprise soit *interprétable*. Cette propriété, peu étudiée dans la littérature sur les Normalizing Flows, constitue pourtant un enjeu essentiel de l'Intelligence Artificielle, où la compréhension des modèles entraînés par des humains est importante dans des domaines allant des sciences physiques jusqu'à la médecine.

Le dernier point a largement guidé nos approches. L'interprétabilité est liée à la capacité d'un être humain de comprendre les différentes étapes qui conduisent la machine à produire une certaine sortie, étant donnée une entrée connue. Plus précisément, dans le cas de réseaux de neurones profonds, il est souvent plus facile d'entraîner le modèle que de comprendre son fonctionnement interne. Plusieurs approches existent, des substituts (Ribeiro et al., 2016) aux perturbations locales (Ancona et al., 2022) en passant par les méta-explications (Lapuschkin et al., 2019). Ces différentes techniques forment d'ailleurs un champ disciplinaire appelé *Explainable Artificial Intelligence (XAI)*.

Fixed-kinetic Neural Hamiltonian Flows pour l'apprentissage non supervisé

Architecture

L'idée des Neural Hamiltonian Flows (Toth et al., 2020) est de considérer des transformations Hamiltoniennes. Il s'agit donc d'augmenter l'espace des paramètres, assimilés à des positions $\mathbf{q}$, en un espace de phases qui combinent des positions et des momenta (masse $\times$ vitesse) $\mathbf{p}$. Comme vu précédemment, les transformations Hamiltoniennes possèdent des propriétés en faisant des candidats intéressants pour des Normalizing flows :

1. ce sont des transformations inversibles et suffisamment régulières, pour peu que l'on suppose des hypothèses assez peu contraignantes sur l'Hamiltonien du système ;
2. elles sont également symplectiques, ce qui implique en particulier la préservation du volume dans l'espace des phases. Autrement dit, le déterminant Jacobien de la transformation est égal à 1, ce qui permet d'annuler le coût computationnel lié au calcul de cette quantité qui intervient dans la formule de changement de variable. Notons qu'une telle propriété demande de la prudence car elle peut aboutir à l'introduction de biais dans le modèle (Draxler et al., 2024) ;
3. l'utilisation de transformations Hamiltoniennes pour transporter une distribution sur une autre est à la base de certaines méthodes MCMC robustes (Duane et al., 1987) ;
4. enfin, parce qu'elles sont à la base de la Mécanique classique, on s'attend à ce qu'elles soient interprétables, dans le sens où les énergies apprises par le modèle peuvent être assimilées à des quantité bien connues des physiciens.

Il a donc été proposé (Toth et al., 2020) d'utiliser une architecture de type Normalizing Flows reposant sur ce principe. En mode entraînement, on part d'un point de données $\mathbf{q}_T$ issu de la distribution cible. On utilise un Encodeur, paramétré par deux réseaux de neurones μ et σ, pour générer un momentum $\mathbf{p}_T$ selon une loi normale $\mathcal{N}(\mu(\mathbf{q}_T), \sigma(\mathbf{q}_T)^2 I)$. Ensuite, on fait évoluer le point dans l'espace des phases en intégrant les équations d'Hamilton grâce à un schéma symplectique. Les énergies potentielle V et cinétique K du système sont elles aussi paramétrées chacune par un réseau de neurones. Le point résultant $(\mathbf{q}_0, \mathbf{p}_0)$ dans l'espace des phases doit, une fois le modèle entraîné, suivre une approximation de la distribution a priori jointe - une Gaussienne la plupart du temps. Le NHF est entraîné par backpropagation avec pour objectif de minimiser la fonction de perte de type Evidence Lower BOund (ELBO) suivante :

$$\mathcal{L}(\mathbf{q}_T) = \mathbf{E}_f \left[\log \Pi_0(\mathcal{T}^{-1}(\mathbf{q}_T, \mathbf{p}_T)) - \log f(\mathbf{p}_T | \mathbf{q}_T) \right].$$

L'architecture est illustrée dans la Figure 4.23.

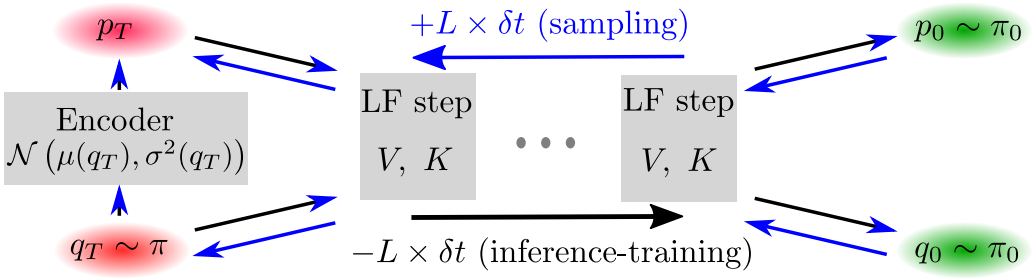


Figure 4.23: Représentation schématique de l'architecture du NHF.

La force de ce modèle est que ses bonnes propriétés sont assurées par l'emploi d'un schéma symplectique pour l'intégration numérique des équations d'Hamilton. Plus précisément, les propriétés 1 à 3 énoncées plus haut ne dépendent absolument pas du choix de l'architecture retenue pour les réseaux de neurones qui composent le NHF. En effet l'utilisation d'architectures pertinentes relativement au problème étudié est à la base de nombreuses méthodes d'apprentissage en grande dimension. Par exemple, cette observation constitue le fondement du Geometric Deep Learning (Bronstein et al., 2021), qui préconise l'emploi d'*a priori* géométriques pour la construction de réseaux de neurones afin de résoudre un problème donné. Ici, cependant, les bonnes propriétés nécessaires à l'apprentissage efficace en grande dimension au moyen d'un modèle de flot sont assurées par l'emploi du schéma Leapfrog. Ce dernier permet l'inversibilité et la régularité de la transformation Hamiltonienne apprise ; il assure aussi la préservation du volume dans l'espace des phases, et donc l'efficacité numérique du processus ; enfin, il permet une exploration efficace du paysage énergétique associé à la distribution cible en exploitant le gradient de cette dernière.

On notera aussi que l'emploi de principes issus de la mécanique classique semble être une bonne idée pour améliorer l'interprétabilité. En effet, l'architecture originelle du NHF repose sur 4 réseaux de neurones, *a priori* des boîtes noires qu'il est difficile d'interpréter. Cependant, sur des distributions bidimensionnelles multimodales, on remarque parfois que l'énergie potentielle apprise présente dans certains cas des extrema locaux correspondant aux différents modes des données (Toth et al., 2020). Ce comportement souhaitable n'est en revanche pas assuré en permanence, comme le montrent certaines de nos expériences menées sur des mélanges Gaussiens en deux dimensions.

C'est pourquoi nous proposons d'introduire une version du NHF dans laquelle l'énergie cinétique n'est plus une boîte noire mais une forme quadratique définie positive, comme c'est le cas en mécanique classique. Ce faisant, on espère que le modèle sera plus à même d'apprendre une dynamique classique dans laquelle l'énergie potentielle deviendra à son tour interprétable : en effet, forcer l'énergie cinétique à être une forme quadratique $K(\mathbf{p}) = \frac{1}{2}\mathbf{p}\mathcal{M}^{-1}\mathbf{p}$ doit permettre à l'énergie potentielle de devenir une approximation du logarithme négatif de la cible, $V(\mathbf{q}) = -\log \pi(\mathbf{q})$. Nous appelons le modèle résultant

Fixed-kinetic Neural Hamiltonian Flows (FK-NHF).

Tests de robustesse sur un mélange Gaussien bidimensionnel

Nous proposons donc de tester la robustesse de ces modèles sur un exemple similaire à celui de l'article introductif des NHF (Toth et al., 2020) : un mélange Gaussien bidimensionnel avec neuf modes de même variance. Pour cela, nous comparons les performances du modèle NHF tel qu'introduit pour la première fois, que l'on appelle ici *MLP-kinetic Neural Hamiltonian Flows* (MLPK-NHF) (Toth et al., 2020), avec le Fixed-kinetic NHF. Plus précisément :

- MLP-kinetic NHF où μ et σ sont des MLPs de taille $(2, N, N, 2)$, et où V et K sont des MLPs de taille $(2, N, N, 1)$. On prend $N = 8,\ 32,\ 128$, selon les expériences..
- Fixed-kinetic NHF où μ et σ sont des MLPs de taille $(2, N, N, 2)$, V un MLP de taille $(2, N, N, 1)$. L'énergie cinétique K est fixée à une forme quadratique positive dont la matrice de masse est apprise durant l'entraînement. On prend $N = 8,\ 32,\ 128$, selon les expériences.

Nos expériences illustrent clairement la capacité de ces deux modèles à retrouver les 9 modes de la distribution cible après une phase d'entraînement, comme montré sur la Figure 4.24.

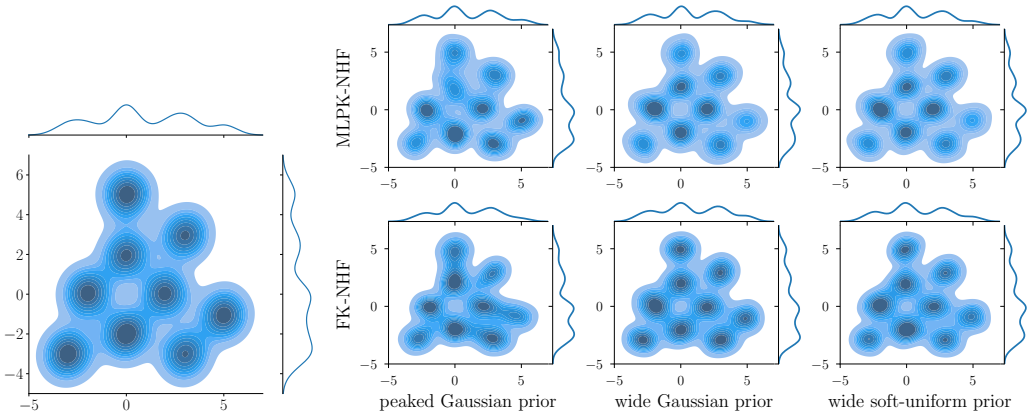


Figure 4.24: Estimation de la densité avec ses marginales de (Gauche) la cible multimodale bidimensionnelle (9 modes avec même poids et même matrice de covariance $0.5^2 I_2$) et (Droite) des échantillons (samples) produits par différents modèles de NHF entraînés avec un choix d'a priori différent.

L'intérêt du Fixed-kinetic NHF réside dans sa grande robustesse au choix des hyperparamètres L (nombre de Leapfrogs) et $T = L \times \delta t$ (temps d'intégration) du schéma Leapfrog. Comme illustré dans la Figure 4.25, on constate que la fonction de perte du modèle à énergie cinétique fixée suit à peu près la même tendance quel que soit le choix des hyperparamètres, à nombre de neurones par couche cachée N fixé. Cela est notamment visible à mesure que le modèle est simple, ce qui est équivalent à un choix de N petit.
Mais le plus grand avantage du Fixed-kinetic NHF réside dans son interprétabilité. Dans la Figure 4.26 sont présentées les estimations de l'énergie potentielle apprise par différents modèles de NHF, à savoir MLP-kinetic et Fixed-kinetic, et des choix d'a priori π_0 différents : une Gaussienne avec faible variance, une Gaussienne avec une variance permettant de recouvrir la distribution cible et une approximation continue d'une loi uniforme recouvrant également le support de la cible. Dans les deux derniers cas, c'est à dire quand la distribution de base est de variance suffisamment grande pour recouvrir de manière satisfaisante le support de la distribution cible, alors les potentiels appris présentent des extrema aux modes de la cible. Pour le MLP-kinetic NHF, ces extrema peuvent être des minima ou des maxima, sans que l'explication de ce phénomène soit clairement identifiée. Pour le Fixed-kinetic, comme attendu, ce sont toujours des minima. En revanche, quand l'a priori est de petite variance, alors l'énergie potentielle apprise par le MLP-kinetic NHF perd toute son interprétabilité, ce qui n'est pas le cas du

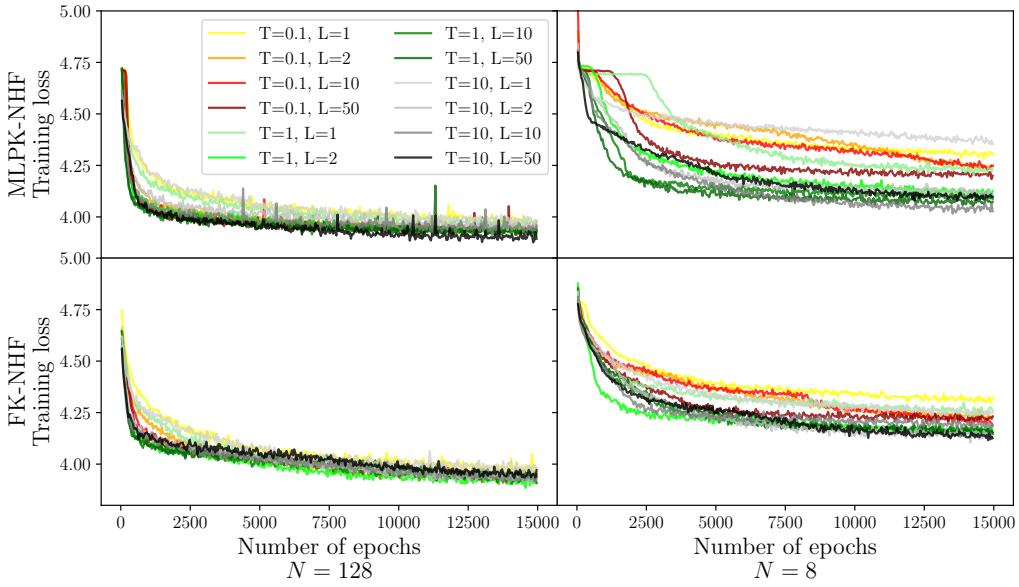


Figure 4.25: Évolution de la fonction de perte en fonction du nombre d'époques pour des modèles de NHF avec différents choix de N (nombre de neurones par couche cachée dans chacun des réseaux constituant le modèle), L (nombre de Leapfrogs) et T (temps d'intégration).

Fixed-kinetic NHF, montrant la robustesse de ce dernier ainsi que son interprétabilité pour une large gamme de régimes.

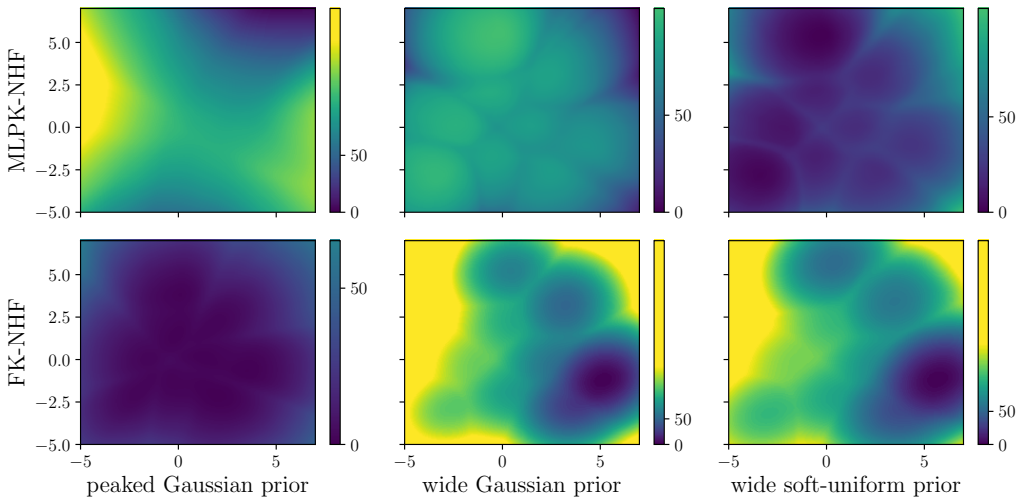


Figure 4.26: Energies potentielles apprises par les deux types de NHF, avec un choix d'a priori différent à chaque fois.

Dans ce genre de situation problématique pour le MLP-kinetic NHF, nous avons remarqué que la multimodalité était transférée sur la distribution des momenta $\mathbf{p}_T$ générés par l'Encodeur. Pour être parfaitement exhaustif, nous avons donc également mené des expériences dans lesquelles nous avons supprimé les réseaux de neurones μ et σ et forcé la loi Normale de l'Encodeur à être une Gaussienne unimodale. Dans ces modèles dits *sans Encodeur*, la multimodalité était toujours transférée dans le potentiel appris, fournissant ainsi une nouvelle manière de simplifier le modèle tout en améliorant son interprétabilité. Néanmoins, pour des problèmes compliqués en grande dimension, nous avons préféré nous focaliser sur des modèles usuels *avec Encodeur* afin de laisser aux NHF suffisamment de flexibilité dans leur phase d'apprentissage.

Tests de performance en génération d'images

A notre connaissance, les modèles de NHF n'ont pas été testés sur des distributions en dimension supérieure à 2. Nous proposons ici de comparer les performances des NHF sur de la génération d'images avec celles d'un modèle de flot classique, un Real NVP (Dinh et al., 2017). L'objectif est d'effectuer les comparaisons à nombre de paramètres fixé. Nous considérons donc un modèle de Real NVP avec 2.27 millions de paramètres, un MLP-kinetic NHF avec 2.20 millions de paramètres et un Fixed-kinetic NHF plus compact, avec 1.94 millions de paramètres. Nous souhaitons en effet que ce dernier soit plus compact que les deux premiers, l'intérêt du Fixed-kinetic NHF étant aussi d'opérer une réduction du nombre de paramètres à optimiser. Les modèles sont entraînés sur 50 époques avec des mini-batchs de taille 32, après une étape de pré-traitement des images (Dinh et al., 2017). Ces images correspondent aux jeux de données MNIST (Deng, 2012), consistant en des chiffres de 0 à 9 écrits à la main, et Fashion-MNIST (Xiao et al., 2017), consistant en des vêtements et accessoires de mode. Une fois entraîné, on s'attend à ce que les trois modèles soient capables de produire des images ressemblant à celles auxquels ils ont été exposés. La qualité de ces échantillons est mesurée grâce à des estimations de KL-divergence (Perez-Cruz, 2008) et de bits/pixel (Papamakarios et al., 2017) (plus ces scores sont faibles, plus la qualité est grande). Les échantillons sont représentés dans la Figure 4.27 et montrent que les performances sont difficilement discernables, bien qu'elles soient atteintes avec une architecture plus simple concernant le Fixed-kinetic NHF.

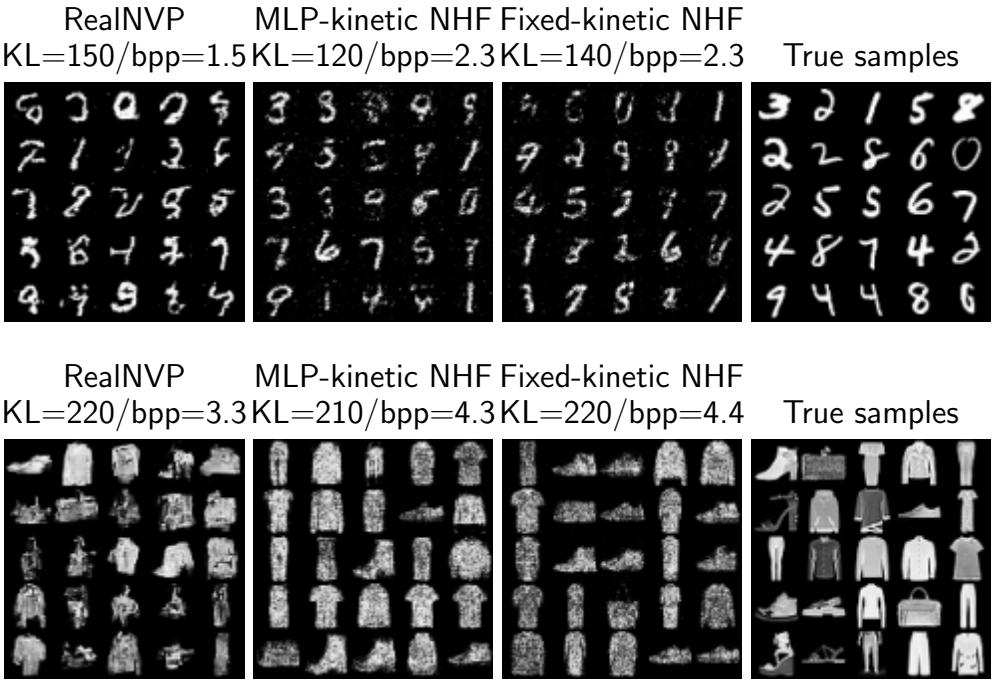


Figure 4.27: Échantillons produits après entraînement sur les jeux de donnée MNIST et Fashion-MNIST avec estimation de la KL-divergence ainsi que le nombre de bits/pixel.

Une propriété intéressante des NHF est qu'il préserve leur interprétabilité en grande dimension. En effet, on remarque que les énergies potentielles apprises présentent encore des extrema aux modes des données. Cette capacité les rend d'autant plus robuste au choix des hyperparamètres du Leapfrog dans le sens où apprendre un potentiel proche du logarithme négatif de la cible leur permet de rester performant dans des régimes qui s'éloignent des hyperparamètres utilisés pour l'entraînement. Ce phénomène est illustré dans la Figure 4.28 qui montre qu'une fois entraîné, les modèles sont capables de produire des échantillons de bonne qualité même dans des situations où le nombre de Leapfrogs utilisé n'est pas le même que celui utilisé pour l'entraînement.
Enfin, puisque l'état de l'art en génération d'images a récemment été atteint grâce à l'utilisation de

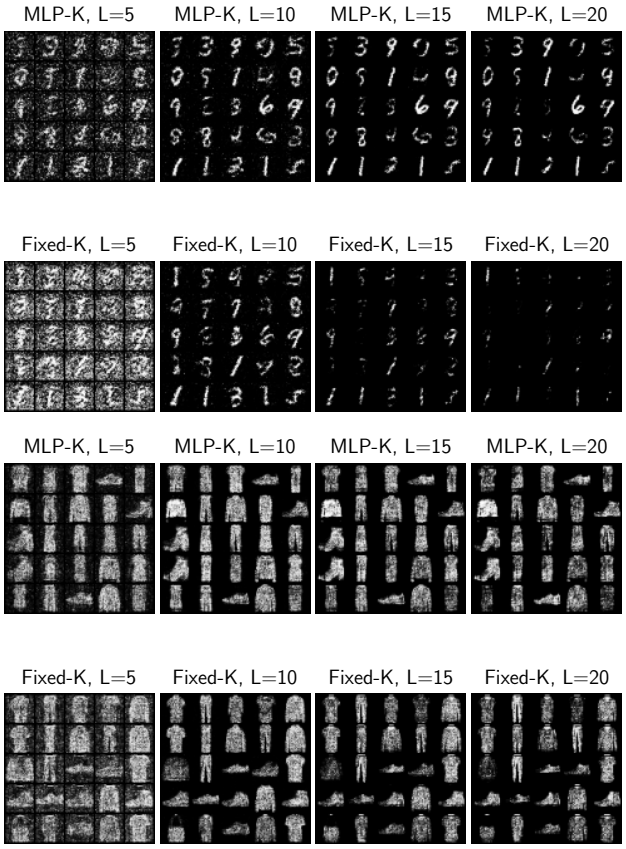
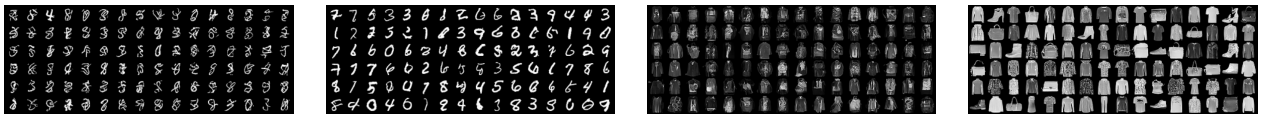


Figure 4.28: Stabilité des modèles de NHF entraînés avec $L = 10$ Leapfrog steps, pour le MNIST (première et seconde lignes) et le Fashion-MNIST (troisième et quatrième lignes).

modèle probabilistes de diffusion (Sohl-Dickstein et al., 2015), nous avons comparé les performances de ces derniers avec les NHF. Une différence fondamentale est que les modèles de diffusion doivent, pour échantillonner, inverser la chaîne de Markov apprise pour bruiter le signal, ce qui est numériquement coûteux (Song et al., 2022). Comme plusieurs dizaines d'étapes sont nécessaires pour obtenir de bons échantillons, cf. Figure 4.29, ce qui doit être directement comparé aux $L = 10$ étapes de Leapfrog utilisés dans notre modèle, le modèle résultant, bien que performant, ne permet pas d'obtenir des échantillons à faible coût une fois entraîné.



(a) $L = 10$, KL $= 250$ (b) $L = 100$, KL $= 70$ (c) $L = 10$, KL $= 340$ (d) $L = 100$, KL $= 70$

Figure 4.29: Échantillons produits par des modèles de diffusion. Plusieurs dizaines d'étapes de débruitage sont nécessaires pour obtenir une bonne qualité.

Application à l'inférence Bayésienne et utilisation en Cosmologie

Comme largement discuté précédemment, un paradigme important en science est celui de l'inférence Bayésienne. En cosmologie notamment, le traitement de données de plus en plus massives (Abell et al., 2009) nécessite des algorithmes performants et des calculateurs puissants. Si nous avons proposé l'emploi d'algorithmes de Monte Carlo non réversibles pour une inférence robuste et asymptotiquement exacte dans des espaces à plusieurs milliers, voire millions de paramètres, il existe une autre voie, basée

sur des modèles génératifs.

L'idée centrale ici consiste à transformer l'a priori, dans le sens Bayésien du terme, en la distribution cible d'intérêt, sachant des données, en utilisant un modèle de flots. Une méthodologie a été proposée dans le cadre des *Boltzmann generators* (Noé et al., 2019). Nous proposons de l'adapter directement aux NHF, la principale différence étant que la transformation a lieu dans l'espace des phases et non pas dans l'espace des positions comme c'est habituellement le cas. Il s'agit ici de construire un modèle génératif capable de produire des échantillons qui suivent une distribution *proche* de la posterieure cible, dans un sens à définir. Le NHF ainsi construit n'a pas vocation à être un générateur asymptotiquement exact de la distribution cible, comme c'est le cas avec un algorithme de Monte Carlo. L'objectif est d'entraîner un modèle grâce à des données - qui ne suivent d'ailleurs pas la distribution cible mais à partir desquelles on peut évaluer la vraisemblance - pour produire des échantillons *ressemblant* à la distribution cible. Une fois l'entraînement effectué, la génération de tels échantillons devient numériquement quasi-gratuite, contrairement aux méthodes Monte Carlo qui nécessitent de payer un coût constant pour produire un échantillon. L'emploi de ces modèles répond donc à des objectifs différents :

- le modèle ainsi obtenu est un générateur permettant l'estimation rapide mais inexacte d'observables liées à la distribution cible ;
- ce modèle peut à son tour servir comme moyen efficace de proposer de nouveaux états dans un algorithme de Monte Carlo classique (Gabrié et al., 2022).

Partant de réalisations de l'a priori, on utilise donc une successions de transformations Hamiltoniennes pour les transformer en la distribution cible. Durant la phase d'entraînement, on cherche à minimiser la KL-divergence entre la distribution jointe du modèle $M(\mathbf{q}_T, \mathbf{p}_T)$ et la distribution cible, que l'on suppose écrite sous la forme $\Pi(\mathbf{q}_T, \mathbf{p}_T|\mathbf{d}) = \pi(\mathbf{q}_T|\mathbf{d})g(\mathbf{p}_T)$ pour avoir l'indépendance entre les positions et les momenta. Nous avons aussi proposé d'adapter l'ELBO du NHF (Toth et al., 2020) dans ce cadre. Le modèle peut être schématiquement représenté comme dans la Figure 4.30.

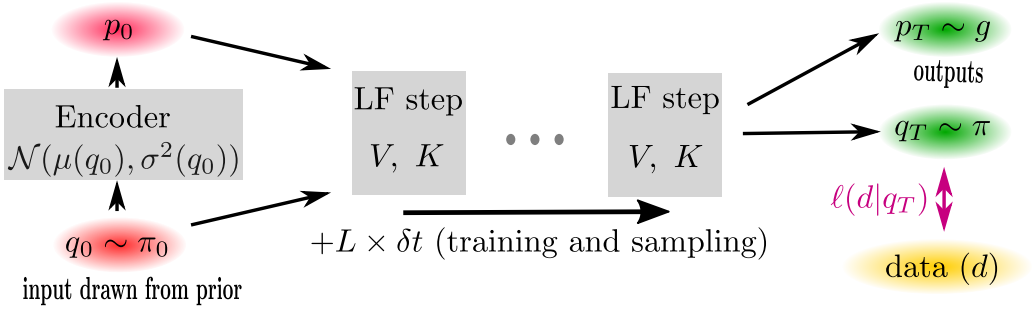


Figure 4.30: Représentation schématique de l'architecture du NHF pour l'inférence Bayésienne.

Il devient alors possible d'utiliser ce cadre pour la cosmologie et en particulier, pour l'inférence de paramètres cosmologiques issus du modèle standard à partir de l'observation de supernovæ (e.g. Riess et al., 1998; Betoule et al., 2014). En effet, selon le modèle Λ-CDM, la relation entre la distance et la luminosité des supernovæ de Type Ia supernovae dépend de deux paramètres cosmologiques: le paramètre gouvernant la densité de matière Ω_m et le paramètre de Hubble h qui contrôle la vitesse d'expansion de l'univers. Pour être plus exhaustif, les bases de données sur des supernovæ de type Ia reportent le module de distance μ, défini comme la différence entre la magnitude apparente et absolue d'un objet astronomique. Cette quantité est directement liée à la distance lumineuse (Weinberg, 1972) et donc une fonction du redshift z, mais aussi de Ω_m et h:

$$\mu(z, \Omega_m, h) = 5 \log_{10} \left(\frac{D_L^*(z, \Omega_m)}{h 10\text{pc}} \right)$$

où

$$D_L^*(z, \Omega_m) = \frac{c(1+z)}{H_0} \int_0^z \frac{ds}{\sqrt{1-\Omega_m + \Omega_m(1+s)^3}},$$

et $H_0 = 100$ km s^{-1} Mpc^{-1}, c étant la vitesse de la lumière dans le vide.
En pratique, on évite de calculer l'intégrale dans D_L^* grâce à une approximation due à (Pen, 1999) valide dans un univers plat :

$$D_L^*(z, \Omega_m) = \frac{c(1+z)}{H_0} \left[\eta(1, \Omega_m) - \eta\left(\frac{1}{1+z}, \Omega_m\right)\right],$$

avec

$$\eta(a, \Omega_m) = 2\sqrt{1+s^3}\left(\frac{1}{a} - 0.1540\frac{s}{a^3} + 0.4304\frac{s^2}{a^2} + 0.19097\frac{s^3}{a} + 0.066941 s^4\right).$$

L'objectif est donc d'échantillonner la distribution postérieure $\pi(\Omega_m, h|\text{data})$ qui quantifie la probabilité que l'on vive dans un univers dont la densité et l'expansion moyennes sont égales à Ω_m and h, sachant D observations data $= \{z_i, \mu_i\}_{1 \leq i \leq D}$ de supernovæ de type Ia, et une matrice de covariance C des modules de distance observés. On fait l'hypothèse que la vraisemblance du problème est Gaussienne, i.e. que les données observées et les valeurs associées des paramètres cosmologiques diffèrent à un bruit Gaussien près. Un tel problème d'inférence est important car la simulation exacte de la luminosité de ces objets astronomiques est compliquée et dépend de beaucoup de paramètres bruitant le signal. Il est donc important d'utiliser le moins de paramètres possible dans notre modèle, ce qui justifie l'emploi du Fixed-kinetic NHF, en tant que modèle compact, robuste et interprétable, comme moyen de résoudre ce problème.

Les résultats sont résumés dans les graphiques de la Figure 4.31 qui illustre les moyennes et écarts-types cumulés de ces paramètres pour différents algorithmes, avec différents choix de fonction de perte. Ils montrent que les modèles du NHF sont capables d'inférer les moyennes et écarts-types corrects de la distribution cible, avec une erreur de l'ordre de moins de 5%. Comme annoncé précédemment, les valeurs retrouvées ne sont pas parfaites puisque, contrairement au HMC qui est un algorithme asymptotiquement exact, les NHF ne font que minimiser une certaine divergence entre la distribution jointe du modèle et celle jointe de la cible.

Résumé des principales contributions

- Introduction d'un modèle alternatif de NHF à énergie cinétique quadratique, le Fixed-Kinetic NHF, pour une meilleure interprétabilité du modèle ainsi qu'un coût computationnel réduit.
- Tests du NHF et du Fixed-kinetic NHF sur des problèmes de génération d'image et comparaison avec des modèles de flots classiques ainsi que des modèles de diffusion.
- Adaptation des NHF au contexte de l'inférence Bayésienne et application à l'inférence de paramètres cosmologiques du modèle standard à partir d'observations astronomiques.

Conclusion générale

L'objet de cette thèse est le développement, l'implémentation et l'étude des performances de deux échantillonneurs, principalement dans le contexte de l'inférence Bayésienne pour la cosmologie. Le premier algorithme, PDMC-BORG, est un algorithme de Monte Carlo non-réversible utilisé pour l'échantillonnage du champ primordial de fluctuations. Le second est un modèle génératif basé

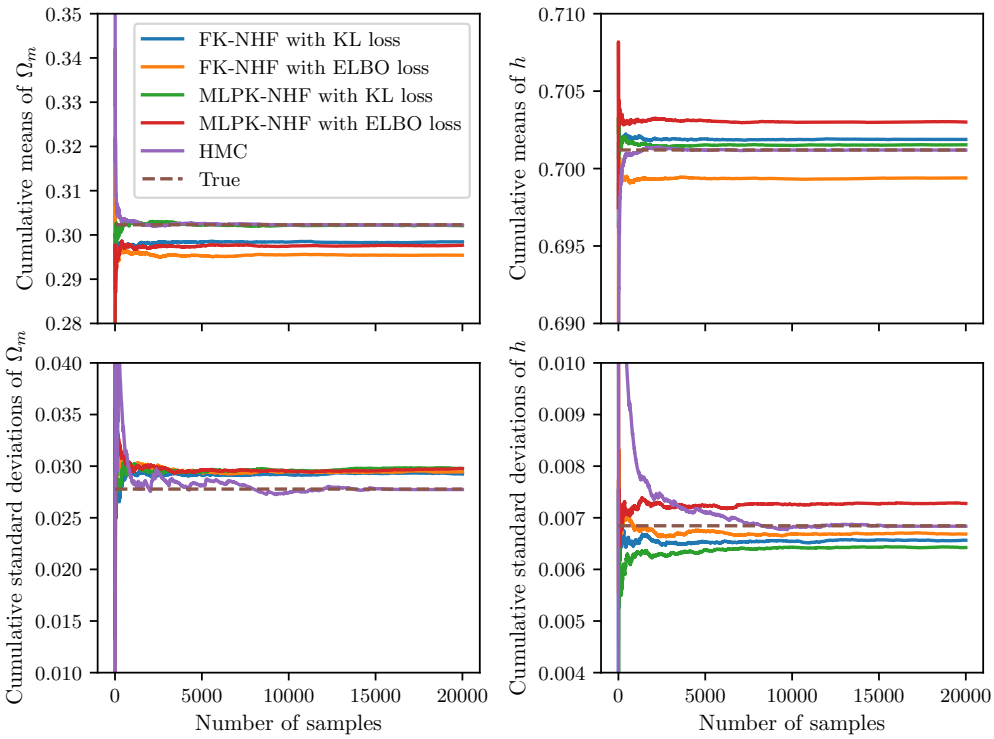


Figure 4.31: Moyennes et écart-types cumulés de Ω_m et h produits par des NHF entraînés et un HMC sur 20,000 échantillons, comparés à la vraie valeur en pointillés. Les modèles sont entraînés sur $30,000$ époques. On utilise un a priori de type soft-uniform et on fixe $g \sim \mathcal{N}(0, I_2)$.

sur des flots Hamiltoniens, que nous avons testé sur des problèmes de génération d'images mais aussi pour l'inférence de paramètres du modèle standard de la cosmologie. Ces deux algorithmes sont différents tant dans leur nature que dans les applications pour lesquelles ils peuvent être utilisés, mais ils ont en commun leur interprétabilité ainsi que leur capacité d'adaptation au cadre d'inférence Bayésienne.

Dans le domaine de l'inférence de la structure à grande échelle de l'univers, PDMC-BORG a démontré des performances prometteuses. Sa dynamique balistique semble particulièrement adaptée à l'exploration d'espaces de paramètre à grande dimension et l'algorithme a montré des performances similaires à celles d'un HMC, qui est l'état de l'art sur ce problème. A mesure que la dimension et la complexité du problème augmenteront dans de futures expériences, on s'attend à ce que les différences de performance deviennent plus marquées entre les deux algorithmes.

Enfin, notre étude de l'architecture NHF nous a permis de développer une meilleure compréhension des modèles de flots Hamiltoniens. Parce qu'ils reposent sur des transformations symplectiques, ces modèles sont particulièrement intéressants en terme de flexibilité et de coût computationnel. Nous nous sommes attachés à démontrer la robustesse de ces modèles et les avons testés sur des problèmes plus complexes de génération d'images. Enfin, nous avons proposé d'adapter le cadre des Boltzmann generators afin d'utiliser les NHF pour l'inférence Bayésienne.

Tous les projets explorés dans cette thèse ouvrent naturellement des prolongements ou des nouvelles pistes de recherche. En voici quelques uns, en guide de conclusion à ce manuscrit :

- le réglage automatique de PDMC-BORG ;
- tester PDMC-BORG sur des problèmes de plus grande dimension ;

- tester PDMC-BORG sur des données astronomiques réelles ;
- apprendre automatiquement le paysage énergétique de la distribution postérieure cible ;
- effectuer une analyse théorique des potentiels biais générés par les NHF ;
- tester l'interprétabilité des modèles de diffusion.

Bibliography

Abell, P. A., Allison, J., Anderson, S. F., Andrew, J. R., Angel, J. R. P., Armus, L., Arnett, D., Asztalos, S. J., et al. (2009). LSST Science Book, Version 2.0. *arXiv e-prints*, page arXiv:0912.0201.

Alexandre Bouchard-Côté, S. J. V. and Doucet, A. (2018). The bouncy particle sampler: A nonreversible rejection-free markov chain monte carlo method. *Journal of the American Statistical Association*, 113(522):855–867.

Alpher, R. A. and Herman, R. K. (1948). Evolution of the universe. *Nature*, 162:774–775.

Amiaux, J., Scaramella, R., Mellier, Y., Altieri, B., Burigana, C., Silva, A. D., Gomez, P., Hoar, J., Laureijs, R., Maiorano, E., Oliveira, D. M., Renk, F., Criado, G. S., Tereno, I., Auguères, J. L., Brinchmann, J., Cropper, M., Duvet, L., Ealet, A., Franzetti, P., Garilli, B., Gondoin, P., Guzzo, L., Hoekstra, H., Holmes, R., Jahnke, K., Kitching, T., Meneghetti, M., Percival, W., and Warren, S. (2012). Euclid Mission: building of a reference survey. In Clampin, M. C., Fazio, G. G., MacEwen, H. A., and Jr., J. M. O., editors, *Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave*, volume 8442, page 84420Z. International Society for Optics and Photonics, SPIE.

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. (2022). *Gradient-Based Attribution Methods*, page 169–191. Springer-Verlag, Berlin, Heidelberg.

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR.

Bartlett, D. J., Bergsdal, D., Desmond, H., Ferreira, P. G., and Jasche, J. (2021a). Constraints on equivalence principle violation from gamma ray bursts. *Phys. Rev. D*, 104:084025.

Bartlett, D. J., Desmond, H., Ferreira, P. G., and Jasche, J. (2021b). Constraints on quantum gravity and the photon mass from gamma ray bursts. *Phys. Rev. D*, 104:103516.

Bartlett, D. J., Kostić, A., Desmond, H., Jasche, J., and Lavaux, G. (2022). Constraints on dark matter annihilation and decay from the large-scale structure of the nearby universe. *Phys. Rev. D*, 106:103526.

Bayer, A. E., Seljak, U., and Modi, C. (2023). Field-Level Inference with Microcanonical Langevin Monte Carlo. In *40th International Conference on Machine Learning*.

Berard, H., Gidel, G., Almahairi, A., Vincent, P., and Lacoste-Julien, S. (2020). A closer look at the optimization landscapes of generative adversarial networks. In *International Conference on Learning Representations*.

Bernard, E. P., Krauth, W., and Wilson, D. B. (2009). Event-chain monte carlo algorithms for hard-sphere systems. *Phys. Rev. E*, 80:056704.

Bernardeau, F., Colombi, S., Gaztañaga, E., and Scoccimarro, R. (2002). Large-scale structure of the universe and cosmological perturbation theory. *Physics Reports*, 367(1):1–248.

Bertone, G., Hooper, D., and Silk, J. (2005). Particle dark matter: evidence, candidates and constraints. *Physics Reports*, 405(5):279–390.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. (2013). Optimal tuning of the hybrid monte carlo algorithm. *Bernoulli*, 19(5A):1501–1534.

Betancourt, M. (2018). A conceptual introduction to hamiltonian monte carlo.

Betoule, M., Kessler, R., Guy, J., Mosher, J., Hardin, D., Biswas, R., Astier, P., El-Hage, P., Konig, M., Kuhlmann, S., Marriner, J., et al. (2014). Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples. *Astronomy & Astrophysics*, 568:A22.

Bierkens, J., Fearnhead, P., and Roberts, G. O. (2016). The zig-zag process and super-efficient sampling for bayesian analysis of big data. *The Annals of Statistics*.

Birdsall, C. K. and Fuss, D. (1969). Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation. *Journal of Computational Physics*, 3(4):494–511.

Bond, J. R., Kofman, L., and Pogosyan, D. (1996). How filaments of galaxies are woven into the cosmic web. *Nature*, 380(6575):603–606.

Bouchet, F. R., Colombi, S., Hivon, E., and Juszkiewicz, R. (1995). Perturbative Lagrangian approach to gravitational instability. *Astronomy & Astrophysics*, 296:575.

Box, G. E. P. and Muller, M. E. (1958). A Note on the Generation of Random Normal Deviates. *The Annals of Mathematical Statistics*, 29(2):610 – 611.

Brent, R. (1972). *Algorithms for Minimization Without Derivatives*. Prentice-Hall series in automatic computation. Prentice-Hall.

Bronstein, M. M., Bruna, J., Cohen, T., and Velickovic, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges.

Candelpergher, B. (2013). *Théorie des probabilités: une introduction élémentaire*. Mathématiques en devenir. Calvage & Mounet.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, 76(1).

Casella, G. and Berger, R. (2001). *Statistical Inference*. Duxbury Resource Center.

Celledoni, E., Leone, A., Murari, D., and Owren, B. (2023). Learning hamiltonians of constrained mechanical systems. *Journal of Computational and Applied Mathematics*, 417:114608.

Chen, F., Lovász, L., and Pak, I. (1999). Lifting markov chains to speed up mixing. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, STOC '99, page 275–281, New York, NY, USA. Association for Computing Machinery.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Ciregan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649.

Corbella, A., Spencer, S. E. F., and Roberts, G. O. (2022). Automatic zig-zag sampling in practice. *Statistics and Computing*, 32(6):107.

Cuesta-Lazaro, C. and Mishra-Sharma, S. (2024). Point cloud approach to generative modeling for galaxy surveys at the field level. *Phys. Rev. D*, 109:123531.

Cybenko, G. V. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314.

Dai, B. and Seljak, U. (2022). Translation and rotation equivariant normalizing flow (TRENF) for optimal cosmological analysis. *Monthly Notices of the Royal Astronomical Society*, 516(2):2363–2373.

Davis, M. H. A. (1984). Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(3):353–388.

Deligiannidis, G., Paulin, D., Bouchard-Côté, A., and Doucet, A. (2020). Randomized hamiltonian monte carlo as scaling limit of the bouncy particle sampler and dimension-free convergence rates. *Annals of Applied Probability*, (Accepted).

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

Desmond, H., Ferreira, P. G., Lavaux, G., and Jasche, J. (2018). Fifth force constraints from the separation of galaxy mass components. *Phys. Rev. D*, 98:064015.

Dhulipala, S. L. N., Che, Y., and Shields, M. D. (2022). Bayesian inference with latent hamiltonian neural networks.

Diaconis, P., Holmes, S., and Neal, R. M. (2000). Analysis of a nonreversible markov chain sampler. *The Annals of Applied Probability*, 10(3):726–752.

Dinh, L., Krueger, D., and Bengio, Y. (2014). Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real NVP. In *International Conference on Learning Representations*.

Doob, J. L. (1942). What is a stochastic process? *The American Mathematical Monthly*, 49(10):648–653.

Draxler, F., Wahl, S., Schnörr, C., and Köthe, U. (2024). On the universality of coupling-based normalizing flows.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Phys. Lett. B*, 195:216–222.

Durrett, R. (1996). *Probability: theory and examples*. Duxbury Press, Belmont, CA, second edition.

Eghbal-zadeh, H., Zellinger, W., and Widmer, G. (2019). Mixture density generative adversarial networks. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Einstein, A. (1916). The foundation of the general theory of relativity. *Annalen Phys.*, 49(7):769–822.

Feder, R. M., Berger, P., and Stein, G. (2020). Nonlinear 3d cosmic web simulation with heavy-tailed generative adversarial networks. *Phys. Rev. D*, 102:103504.

Feller, W. (1949). On the Theory of Stochastic Processes, with Particular Reference to Applications. In *First Berkeley Symposium on Mathematical Statistics and Probability*, pages 403–432.

Fixsen, D. J. (2009). The temperature of the cosmic microwave background. *The Astrophysical Journal*, 707(2):916.

Friedmann, A. (1922). Über die Krümmung des Raumes. *Zeitschrift fur Physik*, 10:377–386.

Friedmann, A. (1924). Über die Möglichkeit einer Welt mit konstanter negativer Krümmung des Raumes. *Zeitschrift fur Physik*, 21(1):326–332.

Frieman, J. A., Turner, M. S., and Huterer, D. (2008). Dark energy and the accelerating universe. *Annual Review of Astronomy and Astrophysics*, 46:385–432.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202.

Gabrié, M., Rotskoff, G. M., and Vanden-Eijnden, E. (2022). Adaptive monte carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10):e2109420119.

Gardner, J. P., Mather, J. C., Abbott, R., Abell, J. S., Abernathy, M., Abney, F. E., Abraham, J. G., Abraham, R., Abul-Huda, Y. M., Acton, S., Adams, C. K., Adams, E., Adler, D. S., Adriaensen, M., Aguilar, J. A., Ahmed, M., Ahmed, N. S., Ahmed, T., Albat, R., Albert, L., Alberts, S., Aldridge, D., Allen, M. M., Allen, S. S., Altenburg, M., Altunc, S., Alvarez, J. L., Álvarez Márquez, J., de Oliveira, C. A., Ambrose, L. L., Anandakrishnan, S. M., Andersen, G. C., Anderson, H. J., Anderson, J., Anderson, K., Anderson, S. M., Aprea, J., Archer, B. J., Arenberg, J. W., Argyriou, I., Arribas, S., Étienne Artigau, Arvai, A. R., Atcheson, P., Atkinson, C. B., Averbukh, J., Aymergen, C., Bacinski, J. J., Baggett, W. E., Bagnasco, G., Baker, L. L., Balzano, V. A., Banks, K. A., Baran, D. A., Barker, E. A., Barrett, L. K., Barringer, B. O., Barto, A., Bast, W., Baudoz, P., Baum, S., Beatty, T. G., Beaulieu, M., Bechtold, K., Beck, T., Beddard, M. M., Beichman, C., Bellagama, L., Bely, P., Berger, T. W., Bergeron, L. E., Bernier, A.-D., Bertch, M. D., Beskow, C., Betz, L. E., Biagetti, C. P., Birkmann, S., Bjorklund, K. F., Blackwood, J. D., Blazek, R. P., Blossfeld, S., Bluth, M., Boccaletti, A., Jr, M. E. B., Bohlin, R. C., Boia, J. J., Böker, T., Bonaventura, N., Bond, N. A., Bosley, K. A., Boucarut, R. A., Bouchet, P., Bouwman, J., Bower, G., Bowers, A. S., Bowers, C. W., Boyce, L. A., Boyer, C. T., Boyer, M. L., Boyer, M., Boyer, R., Bradley, L. D., Brady, G. R., Brandl, B. R., Brannen, J. L., Breda, D., Bremmer, H. G., Brennan, D., Bresnahan, P. A., Bright, S. N., Broiles, B. J., Bromenschenkel, A., Brooks, B. H., Brooks, K. J., Brown, B., Brown, B., Brown, T. M., Bruce, B. W., Bryson, J. G., Bujanda, E. D., Bullock, B. M., Bunker, A. J., Bureo, R., Burt, I. J., Bush, J. A., Bushouse, H. A., Bussman, M. C., Cabaud, O., Cale, S., Calhoon, C. D., Calvani, H., Canipe, A. M., Caputo, F. M., Cara, M., Carey, L., Case, M. E., Cesari, T., Cetorelli, L. D., Chance, D. R., Chandler, L., Chaney, D., Chapman, G. N., Charlot, S., Chayer, P., Cheezum, J. I., Chen, B., Chen, C. H., Cherinka, B., Chichester, S. C., Chilton, Z. S., Chittiraibalan, D., Clampin, M., Clark, C. R., Clark, K. W., Clark, S. M., Claybrooks, E. E., Cleveland, K. A., Cohen, A. L., Cohen, L. M., Colón, K. D., Coleman, B. L., Colina, L., Comber, B. J., Comeau, T. M., Comer, T., Reis, A. C., Connolly, D. C., Conroy, K. E., Contos, A. R., Contreras, J., Cook, N. J., Cooper, J. L., Cooper, R. A., Correia, M. F., Correnti, M., Cossou, C., Costanza, B. F., Coulais, A., Cox, C. R., Coyle, R. T., Cracraft, M. M., Crew, K. A., Curtis, G. J., Cusveller, B., Maciel, C. D. C., Dailey, C. T., Daugeron, F., Davidson, G. S., Davies, J. E., Davis, K. A., Davis, M. S., Day, R., de Chambure, D., de Jong, P., Marchi, G. D., Dean, B. H., Decker, J. E., Delisa, A. S., Dell, L. C., Dellagatta, G., Dembinska, F., Demosthenes, S., Dencheva, N. M., Deneu, P., DePriest, W. W., Deschenes, J., Dethienne, N., Örs Hunor Detre, Diaz, R. I., Dicken, D., DiFelice, A. S., Dillman, M., Disharoon, M. O., Dixon, W. V., Doggett, J. B., Dominguez, K. L., Donaldson, T. S., Doria-Warner, C. M., Santos, T. D., Doty, H., Robert E. Douglas, J., Doyon, R., Dressler, A., Driggers, J., Driggers, P. A., Dunn, J. L., DuPrie, K. C., Dupuis, J., Durning, J., Dutta, S. B., Earl, N. M., Eccleston,

P., Ecobichon, P., Egami, E., Ehrenwinkler, R., Eisenhamer, J. D., Eisenhower, M., Eisenstein, D. J., Hamel, Z. E., Elie, M. L., Elliott, J., Elliott, K. W., Engesser, M., Espinoza, N., Etienne, O., Etxaluze, M., Evans, L., Fabreguettes, L., Falcolini, M., Falini, P. R., Fatig, C., Feeney, M., Feinberg, L. D., Fels, R., Ferdous, N., Ferguson, H. C., Ferrarese, L., Ferreira, M.-H., Ferruit, P., Ferry, M., Filippazzo, J. C., Firre, D., Fix, M., Flagey, N., Flanagan, K. A., Fleming, S. W., Florian, M., Flynn, J. R., Foiadelli, L., Fontaine, M. R., Fontanella, E. M., Forshay, P. R., Fortner, E. A., Fox, O. D., Framarini, A. P., Francisco, J. I., Franck, R., Franx, M., Franz, D. E., Friedman, S. D., Friend, K. E., Frost, J. R., Fu, H., Fullerton, A. W., Gaillard, L., Galkin, S., Gallagher, B., Galyer, A. D., Marín, M. G., Gardner, L. E., Garland, D., Garrett, B. A., Gasman, D., Gáspár, A., Gastaud, R., Gaudreau, D., Gauthier, P. T., Geers, V., Geithner, P. H., Gennaro, M., Gerber, J., Gereau, J. C., Giampaoli, R., Giardino, G., Gibbons, P. C., Gilbert, K., Gilman, L., Girard, J. H., Giuliano, M. E., Gkountis, K., Glasse, A., Glassmire, K. Z., Glauser, A. M., Glazer, S. D., Goldberg, J., Golimowski, D. A., Gonzaga, S. P., Gordon, K. D., Gordon, S. J., Goudfrooij, P., Gough, M. J., Graham, A. J., Grau, C. M., Green, J. D., Greene, G. R., Greene, T. P., Greenfield, P. E., Greenhouse, M. A., Greve, T. R., Greville, E. M., Grimaldi, S., Groe, F. E., Groebner, A., Grumm, D. M., Grundy, T., Güdel, M., Guillard, P., Guldalian, J., Gunn, C. A., Gurule, A., Gutman, I. M., Guy, P. D., Guyot, B., Hack, W. J., Haderlein, P., Hagan, J. B., Hagedorn, A., Hainline, K., Haley, C., Hami, M., Hamilton, F. C., Hammann, J., Hammel, H. B., Hanley, C. J., Hansen, C. A., Hardy, B., Harnisch, B., Harr, M. H., Harris, P., Hart, J. A., Hartig, G. F., Hasan, H., Hashim, K. M., Hashimoto, R., Haskins, S. J., Hawkins, R. E., Hayden, B., Hayden, W. L., Healy, M., Hecht, K., Heeg, V. J., Hejal, R., Helm, K. A., Hengemihle, N. J., Henning, T., Henry, A., Henry, R. L., Henshaw, K., Hernandez, S., Herrington, D. C., Heske, A., Hesman, B. E., Hickey, D. L., Hilbert, B. N., Hines, D. C., Hinz, M. R., Hirsch, M., Hitcho, R. S., Hodapp, K., Hodge, P. E., Hoffman, M., Holfeltz, S. T., Holler, B. J., Hoppa, J. R., Horner, S., Howard, J. M., Howard, R. J., Huber, J. M., Hunkeler, J. S., Hunter, A., Hunter, D. G., Hurd, S. W., Hurst, B. J., Hutchings, J. B., Hylan, J. E., Ignat, L. I., Illingworth, G., Irish, S. M., III, J. C. I., Jr, W. C. J., Jaffe, D. T., Jahic, J., Jahromi, A., Jakobsen, P., James, B., James, J. C., James, L. R., Jamieson, W. B., Jandra, R. D., Jayawardhana, R., Jedrzejewski, R., Jeffers, B. S., Jensen, P., Joanne, E., Johns, A. T., Johnson, C. A., Johnson, E. L., Johnson, P., Johnson, P. S., Johnson, T. K., Johnson, T. W., Johnstone, D., Jollet, D., Jones, D. P., Jones, G. S., Jones, O. C., Jones, R. A., Jones, V., Jordan, I. J., Jordan, M. E., Jue, R., Jurkowski, M. H., Justis, G., Justtanont, K., Kaleida, C. C., Kalirai, J. S., Kalmanson, P. C., Kaltenegger, L., Kammerer, J., Kan, S. K., Kanarek, G. C., Kao, S.-H., Karakla, D. M., Karl, H., Kassin, S. A., Kauffman, D. D., Kavanagh, P., Kelley, L. L., Kelly, D. M., Kendrew, S., Kennedy, H. V., Kenny, D. A., Keski-Kuha, R. A., Keyes, C. D., Khan, A., Kidwell, R. C., Kimble, R. A., King, J. S., King, R. C., Kinzel, W. M., Kirk, J. R., Kirkpatrick, M. E., Klaassen, P., Klingemann, L., Klintworth, P. U., Knapp, B. A., Knight, S., Knollenberg, P. J., Knutsen, D. M., Koehler, R., Koekemoer, A. M., Kofler, E. T., Kontson, V. L., Kovacs, A. R., Kozhurina-Platais, V., Krause, O., Kriss, G. A., Krist, J., Kristoffersen, M. R., Krogel, C., Krueger, A. P., Kulp, B. A., Kumari, N., Kwan, S. W., Kyprianou, M., Labador, A. G., Álvaro Labiano, Lafrenière, D., Lagage, P.-O., Laidler, V. G., Laine, B., Laird, S., Lajoie, C.-P., Lallo, M. D., Lam, M. Y., LaMassa, S. M., Lambros, S. D., Lampenfield, R. J., Lander, M. E., Langston, J. H., Larson, K., Larson, M., LaVerghetta, R. J., Law, D. R., Lawrence, J. F., Lee, D. W., Lee, J., Lee, Y.-N. P., Leisenring, J., Leveille, M. D., Levenson, N. A., Levi, J. S., Levine, M. B., Lewis, D., Lewis, J., Lewis, N., Libralato, M., Lidon, N., Liebrecht, P. L., Lightsey, P., Lilly, S., Lim, F. C., Lim, P. L., Ling, S.-K., Link, L. J., Link, M. N., Lipinski, J. L., Liu, X., Lo, A. S., Lobmeyer, L., Logue, R. M., Long, C. A., Long, D. R., Long, I. D., Long, K. S., López-Caniego, M., Lotz, J. M., Love-Pruitt, J. M., Lubskiy, M., Luers, E. B., Luetgens, R. A., Luevano, A. J., Lui, S. M. G. F., III, J. M. L., Lundquist, R. A., Lunine, J., Lützgendorf, N., Lynch, R. J., MacDonald, A. J., MacDonald, K., Macias, M. J., Macklis, K. I., Maghami, P., Maharaja, R. Y., Maiolino, R., Makrygiannis, K. G., Malla, S. G., Malumuth, E. M., Manjavacas, E., Marini, A., Marrione, A., Marston, A., Martel, A. R., Martin, D., Martin, P. G., Martinez, K. L., Maschmann, M., Masci, G. L., Masetti, M. E., Maszkiewicz, M., Matthews, G., Matuskey, J. E., McBrayer, G. A., McCarthy,

D. W., McCaughrean, M. J., McClare, L. A., McClare, M. D., McCloskey, J. C., McClurg, T. D., McCoy, M., McElwain, M. W., McGregor, R. D., McGuffey, D. B., McKay, A. G., McKenzie, W. K., McLean, B., McMaster, M., McNeil, W., Meester, W. D., Mehalick, K. L., Meixner, M., Meléndez, M., Menzel, M. P., Menzel, M. T., Merz, M., Mesterharm, D. D., Meyer, M. R., Meyett, M. L., Meza, L. E., Midwinter, C., Milam, S. N., Miller, J. T., Miller, W. C., Miskey, C. L., Misselt, K., Mitchell, E. P., Mohan, M., Montoya, E. E., Moran, M. J., Morishita, T., Moro-Martín, A., Morrison, D. L., Morrison, J., Morse, E. C., Moschos, M., Moseley, S. H., Mosier, G. E., Mosner, P., Mountain, M., Muckenthaler, J. S., Mueller, D. G., Mueller, M., Muhiem, D., Mühlmann, P., Mullally, S. E., Mullen, S. M., Munger, A. J., Murphy, J., Murray, K. T., Muzerolle, J. C., Mycroft, M., Myers, A., Myers, C. R., Myers, F. R. R., Myers, R., Myrick, K., Adrian F. Nagle, I., Nayak, O., Naylor, B., Neff, S. G., Nelan, E. P., Nella, J., Nguyen, D. T., Nguyen, M. N., Nickson, B., Nidhiry, J. J., Niedner, M. B., Nieto-Santisteban, M., Nikolov, N. K., Nishisaka, M. A., Noriega-Crespo, A., Nota, A., O'Mara, R. C., Oboryshko, M., O'Brien, M. B., Ochs, W. R., Offenberg, J. D., Ogle, P. M., Ohl, R. G., Olmsted, J. H., Osborne, S. B., O'Shaughnessy, B. P., Östlin, G., O'Sullivan, B., Otor, O. J., Ottens, R., Ouellette, N. N.-Q., Outlaw, D. J., Owens, B. A., Pacifici, C., Page, J. C., Paranilam, J. G., Park, S., Parrish, K. A., Paschal, L., Patapis, P., Patel, J., Patrick, K., Jr, R. A. P., Paul, D. W., Paul, S. J., Pauly, T. A., Pavlovsky, C. M., Peña-Guerrero, M., Pedder, A. H., Peek, M. W., Pelham, P. A., Penanen, K., Perriello, B. A., Perrin, M. D., Perrine, R. F., Perrygo, C., Peslier, M., Petach, M., Peterson, K. A., Pfarr, T., Pierson, J. M., Pietraszkiewicz, M., Pilchen, G., Pipher, J. L., Pirzkal, N., Pitman, J. T., Player, D. M., Plesha, R., Plitzke, A., Pohner, J. A., Poletis, K. K., Pollizzi, J. A., Polster, E., Pontius, J. T., Pontoppidan, K., Porges, S. C., Potter, G. D., Prescott, S., Proffitt, C. R., Pueyo, L., Neira, I. A. Q., Radich, A., Rager, R. T., Rameau, J., Ramey, D. D., Alarcon, R. R., Rampini, R., Rapp, R., Rashford, R. A., Rauscher, B. J., Ravindranath, S., Rawle, T., Rawlings, T. N., Ray, T., Regan, M. W., Rehm, B., Rehm, K. D., Reid, N., Reis, C. A., Renk, F., Reoch, T. B., Ressler, M., Rest, A. W., Reynolds, P. J., Richon, J. G., Richon, K. V., Ridgaway, M., Riedel, A. R., Rieke, G. H., Rieke, M. J., Rifelli, R. E., Rigby, J. R., Riggs, C. S., Ringel, N. J., Ritchie, C. E., Rix, H.-W., Robberto, M., Robinson, G. L., Robinson, M. S., Robinson, O., Rock, F. W., Rodriguez, D. R., del Pino, B. R., Roellig, T., Rohrbach, S. O., Roman, A. J., Romelfanger, F. J., Jr, F. P. R., Rosales, J. J., Rose, P., Roteliuk, A. F., Roth, M. N., Rothwell, B. Q., Rouzaud, S., Rowe, J., Rowlands, N., Roy, A., Royer, P., Rui, C., Rumler, P., Rumpl, W., Russ, M. L., Ryan, M. B., Ryan, R. M., Saad, K., Sabata, M., Sabatino, R., Sabbi, E., Sabelhaus, P. A., Sabia, S., Sahu, K. C., Saif, B. N., Salvignol, J.-C., Samara-Ratna, P., Samuelson, B. S., Sanders, F. A., Sappington, B., Sargent, B. A., Sauer, A., Savadkin, B. J., Sawicki, M., Schappell, T. M., Scheffer, C., Scheithauer, S., Scherer, R., Schiff, C., Schlawin, E., Schmeitzky, O., Schmitz, T. S., Schmude, D. J., Schneider, A., Schreiber, J., Schroeven-Deceuninck, H., Schultz, J. J., Schwab, R., Schwartz, C. H., Scoccimarro, D., Scott, J. F., Scott, M. B., Seaton, B. L., Seely, B. S., Seery, B., Seidleck, M., Sembach, K., Shanahan, C. E., Shaughnessy, B., Shaw, R. A., Shay, C. M., Sheehan, E., Sheth, K., Shih, H.-Y., Shivaei, I., Siegel, N., Sienkiewicz, M. G., Simmons, D. D., Simon, B. P., Sirianni, M., Sivaramakrishnan, A., Slade, J. E., Sloan, G. C., Slocum, C. E., Slowinski, S. E., Smith, C. T., Smith, E. P., Smith, E. C., Smith, K., Smith, R., Smith, S. J., Smolik, J. L., Soderblom, D. R., Sohn, S. T., Sokol, J., Sonneborn, G., Sontag, C. D., Sooy, P. R., Soummer, R., Southwood, D. M., Spain, K., Sparmo, J., Speer, D. T., Spencer, R., Sprofera, J. D., Stallcup, S. S., Stanley, M. K., Stansberry, J. A., Stark, C. C., Starr, C. W., Stassi, D. Y., Steck, J. A., Steeley, C. D., Stephens, M. A., Stephenson, R. J., Stewart, A. C., Stiavelli, M., Jr, H. S., Strada, P., Straughn, A. N., Streetman, S., Strickland, D. K., Strobele, J. F., Stuhlinger, M., Stys, J. E., Such, M., Sukhatme, K., Sullivan, J. F., Sullivan, P. C., Sumner, S. M., Sun, F., Sunnquist, B. D., Swade, D. A., Swam, M. S., Swenton, D. F., Swoish, R. A., Litten, O. I. T., Tamas, L., Tao, A., Taylor, D. K., Taylor, J. M., te Plate, M., Tea, M. V., Teague, K. K., Telfer, R. C., Temim, T., Texter, S. C., Thatte, D. G., Thompson, C. L., Thompson, L. M., Thomson, S. R., Thronson, H., Tierney, C. M., Tikkanen, T., Tinnin, L., Tippet, W. T., Todd, C. W., Tran, H. D., Trauger, J., Trejo, E. G., Truong, J. H. V., Tsukamoto, C. L., Tufail, Y., Tumlinson, J., Tustain, S., Tyra, H., Ubeda, L., Underwood, K., Uzzo,

M. A., Vaclavik, S., Valenduc, F., Valenti, J. A., Campen, J. V., van de Wetering, I., Marel, R. P. V. D., van Haarlem, R., Vandenbussche, B., van Dishoeck, E. F., Vanterpool, D. D., Vernoy, M. R., Costas, M. B. V., Volk, K., Voorzaat, P., Voyton, M. F., Vydra, E., Waddy, D. J., Waelkens, C., Wahlgren, G. M., Jr, F. E. W., Wander, M., Warfield, C. K., Warner, G., Wasiak, F. C., Wasiak, M. F., Wehner, J., Weiler, K. R., Weilert, M., Weiss, S. B., Wells, M., Welty, A. D., Wheate, L., Wheeler, T. P., White, C. L., Whitehouse, P., Whiteleather, J. M., Whitman, W. R., Williams, C. C., Willmer, C. N. A., Willott, C. J., Willoughby, S. P., Wilson, A., Wilson, D., Wilson, D. V., Windhorst, R., Wislowski, E. C., Wolfe, D. J., Wolfe, M. A., Wolff, S., Wondel, A., Woo, C., Woods, R. T., Worden, E., Workman, W., Wright, G. S., Wu, C., Wu, C.-R., Wun, D. D., Wymer, K. B., Yadetie, T., Yan, I. C., Yang, K. C., Yates, K. L., Yeager, C. R., Yerger, E. J., Young, E. T., Young, G., Yu, G., Yu, S., Zak, D. S., Zeidler, P., Zepp, R., Zhou, J., Zincke, C. A., Zonak, S., and Zondag, E. (2023). The james webb space telescope mission. *Publications of the Astronomical Society of the Pacific*, 135(1048):068001.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Geyer, C. (2011). *Introduction to Markov Chain Monte Carlo*, pages 3–48. CRC Press.

Ghosal, S., Ghosh, J. K., and Samanta, T. (1995). On convergence of posterior distributions. *The Annals of Statistics*, 23(6):2145–2152.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. *Advances in neural information processing systems*, 32.

Gustafson, P. (1998). A guided walk metropolis algorithm. *Statistics and computing*, 8:357–364.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

Hockney, R. W. and Eastwood, J. W. (1988). *Computer simulation using particles*. Bristol: Hilger, 1988.

Homan, M. D. and Gelman, A. (2014). The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

Hubble, E. (1929). A relation between distance and radial velocity among extra-galactic nebulae. *Proceedings of the National Academy of Sciences*, 15(3):168–173.

Hubble, E. (1934). The Distribution of Extra-Galactic Nebulae. *Astrophysical Journal*, 79:8.

Jasche, J. and Kitaura, F. S. (2010). Fast Hamiltonian sampling for large-scale structure inference. *Monthly Notices of the Royal Astronomical Society*, 407(1):29–42.

Jasche, J. and Lavaux, G. (2019). Physical bayesian modelling of the non-linear matter distribution: New insights into the nearby universe. *Astronomy and Astrophysics*, 625:A64.

Jasche, J. and Wandelt, B. D. (2013). Bayesian physical reconstruction of initial conditions from large-scale structure surveys. *Monthly Notices of the Royal Astronomical Society*, 432(2):894–913.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 186(1007):453–461.

Jimenez Rezende, D., Racanière, S., Higgins, I., and Toth, P. (2019). Equivariant Hamiltonian Flows. *arXiv e-prints*, page arXiv:1909.13739.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python.

Kayo, I., Taruya, A., and Suto, Y. (2001). Probability distribution function of cosmological density fluctuations from a gaussian initial condition: Comparison of one-point and two-point lognormal model predictions with n-body simulations. *The Astrophysical Journal*, 561.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kitaura, F. S. and Enßlin, T. A. (2008). Bayesian reconstruction of the cosmological large-scale structure: methodology, inverse algorithms and numerical optimization. *Monthly Notices of the Royal Astronomical Society*, 389(2):497–544.

Kitaura, F. S., Jasche, J., Li, C., Enßlin, T. A., Metcalf, R. B., Wandelt, B. D., Lemson, G., and White, S. D. (2009). Cosmic cartography of the large-scale structure with sloan digital sky survey data release 6. *Monthly Notices of the Royal Astronomical Society*, 400(1):183–203.

Komatsu, E., Smith, K. M., Dunkley, J., Bennett, C. L., Gold, B., Hinshaw, G., Jarosik, N., Larson, D., Nolta, M. R., Page, L., Spergel, D. N., Halpern, M., Hill, R. S., Kogut, A., Limon, M., Meyer, S. S., Odegard, N., Tucker, G. S., Weiland, J. L., Wollack, E., and Wright, E. L. (2011). Seven-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation. *The Astrophysical Journal Supplement Series*, 192(2):18.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243.

Lahav, O., Fisher, K. B., Hoffman, Y., Scharf, C. A., and Zaroubi, S. (1994). Wiener Reconstruction of All-Sky Galaxy Surveys in Spherical Harmonics. *Astrophysical Journal Letters*, 423:L93.

Landau, L. and Lifshitz, E. (1982). *Mechanics: Volume 1*. Number vol. 1 in Course of Theoretical Physics. Elsevier Science.

Langevin, P. (1908). Sur la théorie du mouvement brownien. *CR Acad. Sci. Paris*, 146(530-533):530.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096–1096.

Leclercq, F. (2015). *Bayesian large-scale structure inference and cosmic web analysis*. Theses, Université Pierre et Marie Curie - Paris VI.

Lecun, Y. (1987). *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*. Universite P. et M. Curie (Paris 6).

Ledoux, M. (2001). *The concentration of measure phenomenon*. Number 89. American Mathematical Soc.

Lemaître, G. (1927). Un Univers homogène de masse constante et de rayon croissant rendant compte de la vitesse radiale des nébuleuses extra-galactiques. *Annales de la Société Scientifique de Bruxelles*, 47:49–59.

Lesgourgues, J. (2011). The cosmic linear anisotropy solving system (class) i: Overview.

Levin, D. A., Peres, Y., and Wilmer, E. L. (2006). *Markov chains and mixing times*. American Mathematical Society.

Lewis, A. and Bridle, S. (2002). Cosmological parameters from cmb and other data: A monte carlo approach. *Physical Review D*, 66(10):103511.

Lewis, P. A. W. and Shedler, G. S. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413.

Lin, Z., Khetan, A., Fanti, G., and Oh, S. (2018). Pacgan: The power of two samples in generative adversarial networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Linde, A. (2008). Inflationary cosmology. In *Inflationary Cosmology*, pages 1–54. Springer.

Mcculloch, W. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147.

Menier, E., Bucci, M. A., Yagoubi, M., Mathelin, L., and Schoenauer, M. (2022). Continuous Methods : Hamiltonian Domain Translation. *arXiv e-prints*, page arXiv:2207.03843.

Metropolis, N. C., Rosenbluth, A. W., Rosenbluth, M. N., and Teller, A. H. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.

Michel, M., Durmus, A., and Sénécal, S. (2020). Forward event-chain monte carlo: Fast sampling by randomness control in irreversible markov chains. *Journal of Computational and Graphical Statistics*, 29(4):689–702.

Michel, M., Kapfer, S. C., and Krauth, W. (2014). Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps. *The Journal of Chemical Physics*, 140(5):054116.

Minton, P. D., Raiffa, H., and Schlaifer, R. (1961). Applied statistical decision theory. *American Mathematical Monthly*, 69:72.

Modi, C., Li, Y., and Blei, D. (2023). Reconstructing the universe with variational self-boosted sampling. *Journal of Cosmology and Astroparticle Physics*, 2023(03):059.

Monemvassitis, A., Guillin, A., and Michel, M. (2023). Pdmp characterisation of event-chain monte carlo algorithms for particle systems. *Journal of statistical physics.*, 190(3).

Moutarde, F., Alimi, J. M., Bouchet, F. R., Pellat, R., and Ramani, A. (1991). Precollapse Scale Invariance in Gravitational Instability. *The Astrophysical Journal*, 382:377.

Mudur, N. and Finkbeiner, D. P. (2022). Can denoising diffusion probabilistic models generate realistic astrophysical fields?

Muller, J.-M., Brisebarre, N., De Dinechin, F., Jeannerod, C.-P., Lefevre, V., Melquiond, G., Revol, N., Stehlé, D., Torres, S., et al. (2018). *Handbook of floating-point arithmetic*. Springer.

Neal, R. (2012). Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*.

Noether, E. (1971). Invariant variation problems. *Transport Theory and Statistical Physics*, 1(3):186–207.

Norris, J. R. (1998). *Markov chains*. Number 2. Cambridge university press.

Noé, F., Olsson, S., Köhler, J., and Wu, H. (2019). Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Leoni Aleman, F., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Posada Fishman, S., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Kaiser, Ł., Kamali, A., Kanitscheider, I., Shirish Keskar, N., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Hendrik Kirchner, J., Kiros, J., Knight, M., Kokotajlo, D., Kondraciuk, Ł., Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., Ponde de Oliveira Pinto, H., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Petroski Such, F., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Cerón Uribe, J. F., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2023). GPT-4 Technical Report. *arXiv e-prints*, page arXiv:2303.08774.

O'Dwyer, I. J., Eriksen, H., Wandelt, B., Jewell, J., Larson, D., Górski, K., Banday, A., Levin, S., and Lilje, P. (2004). Bayesian power spectrum analysis of the first-year wilkinson microwave anisotropy probe data. *The Astrophysical Journal*, 617(2):L99.

Pagani, F., Chevallier, A., Power, S., House, T., and Cotter, S. (2024). Nuzz: Numerical zig-zag for general models. *Statistics and Computing*, 34.

Pages, G. and Briane, M. (2018). *Analyse - Théorie de l'intégration: Convolution et transformée de Fourier*. LMD MATHS. De Boeck supérieur.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2022). Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(1).

Papamakarios, G., Pavlakou, T., and Murray, I. (2017). Masked autoregressive flow for density estimation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Park, C., Choi, Y.-Y., Vogeley, M. S., III, J. R. G., Kim, J., Hikage, C., Matsubara, T., Park, M.-G., Suto, Y., Weinberg, D. H., and Collaboration), T. S. (2005). Topology analysis of the sloan digital sky survey. i. scale and luminosity dependence. *The Astrophysical Journal*, 633(1):11.

Peebles, P. J. E. (1980). *The large-scale structure of the universe*. Princeton University Press.

Pen, U.-L. (1999). Analytical fit to the luminosity distance for flat cosmologies with a cosmological constant. *The Astrophysical Journal Supplement Series*, 120(1):49–50.

Penzias, A. A. and Wilson, R. W. (1965). A Measurement of Excess Antenna Temperature at 4080 Mc/s. *The Astrophysical Journal*, 142:419–421.

Perez-Cruz, F. (2008). Kullback-leibler divergence estimation of continuous distributions. In *2008 IEEE International Symposium on Information Theory*, pages 1666–1670.

Peters, E. A. J. F. and de With, G. (2012). Rejection-free monte carlo sampling for general potentials. *Phys. Rev. E*, 85:026703.

Planck Collaboration, Ade, P. A. R., Aghanim, N., Arnaud, M., Ashdown, M., Aumont, J., Baccigalupi, C., Banday, A. J., Barreiro, R. B., Bartlett, J. G., Bartolo, N., Battaner, E., Battye, R., Benabed, K., Benoît, A., Benoit-Lévy, A., Bernard, J.-P., Bersanelli, M., Bielewicz, P., Bock, J. J., Bonaldi, A., Bonavera, L., Bond, J. R., Borrill, J., Bouchet, F. R., Boulanger, F., Bucher, M., Burigana, C., Butler, R. C., Calabrese, E., Cardoso, J.-F., Catalano, A., Challinor, A., Chamballu, A., Chary, R.-R., Chiang, H. C., Chluba, J., Christensen, P. R., Church, S., Clements, D. L., Colombi, S., Colombo, L. P. L., Combet, C., Coulais, A., Crill, B. P., Curto, A., Cuttaia, F., Danese, L., Davies, R. D., Davis, R. J., de Bernardis, P., de Rosa, A., de Zotti, G., Delabrouille, J., Désert, F.-X., Di Valentino, E., Dickinson, C., Diego, J. M., Dolag, K., Dole, H., Donzelli, S., Doré, O., Douspis, M., Ducout, A., Dunkley, J., Dupac, X., Efstathiou, G., Elsner, F., Enßlin, T. A., Eriksen, H. K., Farhang, M., Fergusson, J., Finelli, F., Forni, O., Frailis, M., Fraisse, A. A., Franceschi, E., Frejsel, A., Galeotta, S., Galli, S., Ganga, K., Gauthier, C., Gerbino, M., Ghosh, T., Giard, M., Giraud-Héraud, Y., Giusarma, E., Gjerløw, E., González-Nuevo, J., Górski, K. M., Gratton, S., Gregorio, A., Gruppuso, A., Gudmundsson, J. E., Hamann, J., Hansen, F. K., Hanson, D., Harrison, D. L., Helou, G., Henrot-Versillé, S., Hernández-Monteagudo, C., Herranz, D., Hildebrandt, S. R., Hivon, E., Hobson, M., Holmes, W. A., Hornstrup, A., Hovest, W., Huang, Z., Huffenberger, K. M., Hurier, G., Jaffe, A. H., Jaffe, T. R., Jones, W. C., Juvela, M., Keihänen, E., Keskitalo, R., Kisner, T. S., Kneissl, R., Knoche, J., Knox, L., Kunz, M., Kurki-Suonio, H., Lagache, G., Lähteenmäki, A., Lamarre, J.-M., Lasenby, A., Lattanzi, M., Lawrence, C. R., Leahy, J. P., Leonardi, R., Lesgourgues,

J., Levrier, F., Lewis, A., Liguori, M., Lilje, P. B., Linden-Vørnle, M., López-Caniego, M., Lubin, P. M., Macías-Pérez, J. F., Maggio, G., Maino, D., Mandolesi, N., Mangilli, A., Marchini, A., Maris, M., Martin, P. G., Martinelli, M., Martínez-González, E., Masi, S., Matarrese, S., McGehee, P., Meinhold, P. R., Melchiorri, A., Melin, J.-B., Mendes, L., Mennella, A., Migliaccio, M., Millea, M., Mitra, S., Miville-Deschênes, M.-A., Moneti, A., Montier, L., Morgante, G., Mortlock, D., Moss, A., Munshi, D., Murphy, J. A., Naselsky, P., Nati, F., Natoli, P., Netterfield, C. B., Nørgaard-Nielsen, H. U., Noviello, F., Novikov, D., Novikov, I., Oxborrow, C. A., Paci, F., Pagano, L., Pajot, F., Paladini, R., Paoletti, D., Partridge, B., Pasian, F., Patanchon, G., Pearson, T. J., Perdereau, O., Perotto, L., Perrotta, F., Pettorino, V., Piacentini, F., Piat, M., Pierpaoli, E., Pietrobon, D., Plaszczynski, S., Pointecouteau, E., Polenta, G., Popa, L., Pratt, G. W., Prézeau, G., Prunet, S., Puget, J.-L., Rachen, J. P., Reach, W. T., Rebolo, R., Reinecke, M., Remazeilles, M., Renault, C., Renzi, A., Ristorcelli, I., Rocha, G., Rosset, C., Rossetti, M., Roudier, G., Rouillé d´Orfeuil, B., Rowan-Robinson, M., Rubiño-Martín, J. A., Rusholme, B., Said, N., Salvatelli, V., Salvati, L., Sandri, M., Santos, D., Savelainen, M., Savini, G., Scott, D., Seiffert, M. D., Serra, P., Shellard, E. P. S., Spencer, L. D., Spinelli, M., Stolyarov, V., Stompor, R., Sudiwala, R., Sunyaev, R., Sutton, D., Suur-Uski, A.-S., Sygnet, J.-F., Tauber, J. A., Terenzi, L., Toffolatti, L., Tomasi, M., Tristram, M., Trombetti, T., Tucci, M., Tuovinen, J., Türler, M., Umana, G., Valenziano, L., Valiviita, J., Van Tent, F., Vielva, P., Villa, F., Wade, L. A., Wandelt, B. D., Wehus, I. K., White, M., White, S. D. M., Wilkinson, A., Yvon, D., Zacchei, A., and Zonca, A. (2016). Planck 2015 results - xiii. cosmological parameters. *A&A*, 594:A13.

Planck Collaboration, Aghanim, N., Akrami, Y., Ashdown, M., Aumont, J., Baccigalupi, C., Ballardini, M., Banday, A. J., Barreiro, R. B., Bartolo, N., Basak, S., Battye, R., Benabed, K., Bernard, J. P., Bersanelli, M., Bielewicz, P., Bock, J. J., Bond, J. R., Borrill, J., Bouchet, F. R., Boulanger, F., Bucher, M., Burigana, C., Butler, R. C., Calabrese, E., Cardoso, J. F., Carron, J., Challinor, A., Chiang, H. C., Chluba, J., Colombo, L. P. L., Combet, C., Contreras, D., Crill, B. P., Cuttaia, F., de Bernardis, P., de Zotti, G., Delabrouille, J., Delouis, J. M., Di Valentino, E., Diego, J. M., Doré, O., Douspis, M., Ducout, A., Dupac, X., Dusini, S., Efstathiou, G., Elsner, F., Enßlin, T. A., Eriksen, H. K., Fantaye, Y., Farhang, M., Fergusson, J., Fernandez-Cobos, R., Finelli, F., Forastieri, F., Frailis, M., Fraisse, A. A., Franceschi, E., Frolov, A., Galeotta, S., Galli, S., Ganga, K., Génova-Santos, R. T., Gerbino, M., Ghosh, T., González-Nuevo, J., Górski, K. M., Gratton, S., Gruppuso, A., Gudmundsson, J. E., Hamann, J., Handley, W., Hansen, F. K., Herranz, D., Hildebrandt, S. R., Hivon, E., Huang, Z., Jaffe, A. H., Jones, W. C., Karakci, A., Keihänen, E., Keskitalo, R., Kiiveri, K., Kim, J., Kisner, T. S., Knox, L., Krachmalnicoff, N., Kunz, M., Kurki-Suonio, H., Lagache, G., Lamarre, J. M., Lasenby, A., Lattanzi, M., Lawrence, C. R., Le Jeune, M., Lemos, P., Lesgourgues, J., Levrier, F., Lewis, A., Liguori, M., Lilje, P. B., Lilley, M., Lindholm, V., López-Caniego, M., Lubin, P. M., Ma, Y. Z., Macías-Pérez, J. F., Maggio, G., Maino, D., Mandolesi, N., Mangilli, A., Marcos-Caballero, A., Maris, M., Martin, P. G., Martinelli, M., Martínez-González, E., Matarrese, S., Mauri, N., McEwen, J. D., Meinhold, P. R., Melchiorri, A., Mennella, A., Migliaccio, M., Millea, M., Mitra, S., Miville-Deschênes, M. A., Molinari, D., Montier, L., Morgante, G., Moss, A., Natoli, P., Nørgaard-Nielsen, H. U., Pagano, L., Paoletti, D., Partridge, B., Patanchon, G., Peiris, H. V., Perrotta, F., Pettorino, V., Piacentini, F., Polastri, L., Polenta, G., Puget, J. L., Rachen, J. P., Reinecke, M., Remazeilles, M., Renzi, A., Rocha, G., Rosset, C., Roudier, G., Rubiño-Martín, J. A., Ruiz-Granados, B., Salvati, L., Sandri, M., Savelainen, M., Scott, D., Shellard, E. P. S., Sirignano, C., Sirri, G., Spencer, L. D., Sunyaev, R., Suur-Uski, A. S., Tauber, J. A., Tavagnacco, D., Tenti, M., Toffolatti, L., Tomasi, M., Trombetti, T., Valenziano, L., Valiviita, J., Van Tent, B., Vibert, L., Vielva, P., Villa, F., Vittorio, N., Wandelt, B. D., Wehus, I. K., White, M., White, S. D. M., Zacchei, A., and Zonca, A. (2020). Planck 2018 results. VI. Cosmological parameters. *A&A*, 641:A6.

Popper, K. R. (1934). *The Logic of Scientific Discovery*. Hutchinson, London.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep

learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707.

Ravanbakhsh, S., Lanusse, F., Mandelbaum, R., Schneider, J. G., and Póczos, B. (2016). Enabling dark energy science with deep generative models of galaxy images. In *AAAI Conference on Artificial Intelligence*.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Riess, A. G., Filippenko, A. V., Challis, P., Clocchiatti, A., Diercks, A., Garnavich, P. M., Gilliland, R. L., Hogan, C. J., Jha, S., Kirshner, R. P., Leibundgut, B., Phillips, M. M., Reiss, D., Schmidt, B. P., Schommer, R. A., Smith, R. C., Spyromilio, J., Stubbs, C., Suntzeff, N. B., and Tonry, J. (1998). Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant. *The Astronomical Journal*, 116(3):1009–1038.

Robert, C. P. (2001). *The Bayesian Choice : From Decision-Theoretic Foundations to Computational Implementation*. Springer Texts in Statistics. Springer New York., New York, NY, 2nd edition. edition.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer, 2nd edition.

Robertson, H. P. (1935). Kinematics and World-Structure. *Astrophysical Journal*, 82:284.

Robnik, J., De Luca, G. B., Silverstein, E., and Seljak, U. (2024). Microcanonical hamiltonian monte carlo. *Journal of Machine Learning Research*, 24(1).

Robnik, J. and Seljak, U. (2023). Fluctuation without dissipation: Microcanonical langevin monte carlo.

Rodriguez, A. C., Kacprzak, T., Lucchi, A., Amara, A., Sgier, R., Fluri, J., Hofmann, T., and Réfrégier, A. (2018). Fast cosmic web simulations with generative adversarial networks. *Computational Astrophysics and Cosmology*, 5(1):1–11.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Rossky, P. J., Doll, J. D., and Friedman, H. L. (1978). Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633.

Rudin, W. (1987). *Real and complex analysis, 3rd ed.* McGraw-Hill, Inc., USA.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). *Learning internal representations by error propagation*, page 318–362. MIT Press, Cambridge, MA, USA.

Rumelhart, D. E. and McClelland, J. L. (1987). *Learning Internal Representations by Error Propagation*, pages 318–362. MIT Press.

Sachs, R. K. and Wolfe, A. M. (1967). Perturbations of a Cosmological Model and Angular Variations of the Microwave Background. *Astrophysical Journal*, 147:73.

Samek, W. and Müller, K.-R. (2019). Towards explainable artificial intelligence. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22. Springer.

Schaye, J., Kugel, R., Schaller, M., Helly, J. C., Braspenning, J., Elbers, W., McCarthy, I. G., van Daalen, M. P., Vandenbroucke, B., Frenk, C. S., Kwan, J., Salcido, J., Bahé, Y. M., Borrow, J., Chaikin, E., Hahn, O., Husko, F., Jenkins, A., Lacey, C. G., and Nobels, F. S. J. (2023). The FLAMINGO project: cosmological hydrodynamical simulations for large-scale structure and galaxy cluster surveys. *Monthly Notices of the Royal Astronomical Society*, 526(4):4978–5020.

Selig, M., Bell, M. R., Junklewitz, H., Oppermann, N., Reinecke, M., Greiner, M., Pachajoa, C., and Enßlin, T. A. (2013). Nifty–numerical information field theory-a versatile python library for signal inference. *Astronomy & Astrophysics*, 554:A26.

Shen, J., Chowdhury, J., Banerjee, S., and Terejanu, G. (2023). Machine fault classification using hamiltonian neural networks.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France. PMLR.

Song, J., Meng, C., and Ermon, S. (2022). Denoising diffusion implicit models.

Souveton, V., Guillin, A., Jasche, J., Lavaux, G., and Michel, M. (2024). Fixed-kinetic neural Hamiltonian flows for enhanced interpretability and reduced complexity. In Dasgupta, S., Mandt, S., and Li, Y., editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3178–3186. PMLR.

Spergel, D. N., Verde, L., Peiris, H. V., Komatsu, E., Nolta, M. R., Bennett, C. L., Halpern, M., Hinshaw, G., Jarosik, N., Kogut, A., Limon, M., Meyer, S. S., Page, L., Tucker, G. S., Weiland, J. L., Wollack, E., and Wright, E. L. (2003). First-year wilkinson microwave anisotropy probe (wmap)* observations: Determination of cosmological parameters. *The Astrophysical Journal Supplement Series*, 148(1):175.

Stachurski, F., Messenger, C., and Hendry, M. (2024). Cosmological inference using gravitational waves and normalizing flows. *Phys. Rev. D*, 109:123547.

Tabak, E. and Vanden-Eijnden, E. (2010). Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233.

Tegmark, M., Blanton, M. R., Strauss, M. A., Hoyle, F., Schlegel, D., Scoccimarro, R., Vogeley, M. S., Weinberg, D. H., Zehavi, I., Berlind, A., Budavari, T., Connolly, A., Eisenstein, D. J., Finkbeiner, D., Frieman, J. A., Gunn, J. E., Hamilton, A. J. S., Hui, L., Jain, B., Johnston, D., Kent, S., Lin, H., Nakajima, R., Nichol, R. C., Ostriker, J. P., Pope, A., Scranton, R., Seljak, U., Sheth, R. K., Stebbins, A., Szalay, A. S., Szapudi, I., Verde, L., Xu, Y., Annis, J., Bahcall, N. A., Brinkmann, J., Burles, S., Castander, F. J., Csabai, I., Loveday, J., Doi, M., Fukugita, M., III, J. R. G., Hennessy, G., Hogg, D. W., Ivezić, Z., Knapp, G. R., Lamb, D. Q., Lee, B. C., Lupton, R. H., McKay, T. A., Kunszt, P., Munn, J. A., O'Connell, L., Peoples, J., Pier, J. R., Richmond, M., Rockosi, C., Schneider, D. P., Stoughton, C., Tucker, D. L., Berk, D. E. V., Yanny, B., York, D. G., and Collaboration), T. S. (2004). The three-dimensional power spectrum of galaxies from the sloan digital sky survey. *The Astrophysical Journal*, 606(2):702.

Toth, P., Rezende, D. J., Jaegle, A., Racanière, S., Botev, A., and Higgins, I. (2020). Hamiltonian generative networks. In *International Conference on Learning Representations*.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(October):433–60.

Turitsyn, K. S., Chertkov, M., and Vucelja, M. (2011). Irreversible monte carlo algorithms for efficient sampling. *Physica D: Nonlinear Phenomena*, 240(4-5):410–414.

van de Weygaert, R. and Bertschinger, E. (1996). Peak and gravity constraints in Gaussian primordial density fields: An application of the Hoffman-Ribak method. *Monthly Notices of the Royal Astronomical Society*, 281:84.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Walker, A. G. (1937). On Milne's Theory of World-Structure. *Proceedings of the London Mathematical Society*, 42:90–127.

Weinberg, S. (1972). *Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity*. Wiley.

Werbos, P. J. (1994). *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*. Wiley-Interscience, USA.

Winkler, C., Worrall, D., Hoogeboom, E., and Welling, M. (2019). Learning likelihoods with conditional normalizing flows.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv e-prints*, page arXiv:1708.07747.

York, D. G., Adelman, J., Anderson, John E., J., Anderson, S. F., Annis, J., Bahcall, N. A., Bakken, J. A., Barkhouser, R., Bastian, S., Berman, E., Boroski, W. N., Bracker, S., Briegel, C., Briggs, J. W., Brinkmann, J., Brunner, R., Burles, S., Carey, L., Carr, M. A., Castander, F. J., Chen, B., Colestock, P. L., Connolly, A. J., Crocker, J. H., Csabai, I., Czarapata, P. C., Davis, J. E., Doi, M., Dombeck, T., Eisenstein, D., Ellman, N., Elms, B. R., Evans, M. L., Fan, X., Federwitz, G. R., Fiscelli, L., Friedman, S., Frieman, J. A., Fukugita, M., Gillespie, B., Gunn, J. E., Gurbani, V. K., de Haas, E., Haldeman, M., Harris, F. H., Hayes, J., Heckman, T. M., Hennessy, G. S., Hindsley, R. B., Holm, S., Holmgren, D. J., Huang, C.-h., Hull, C., Husby, D., Ichikawa, S.-I., Ichikawa, T., Ivezic, Z., Kent, S., Kim, R. S. J., Kinney, E., Klaene, M., Kleinman, A. N., Kleinman, S., Knapp, G. R., Korienek, J., Kron, R. G., Kunszt, P. Z., Lamb, D. Q., Lee, B., Leger, R. F., Limmongkol, S., Lindenmeyer, C., Long, D. C., Loomis, C., Loveday, J., Lucinio, R., Lupton, R. H., MacKinnon, B., Mannery, E. J., Mantsch, P. M., Margon, B., McGehee, P., McKay, T. A., Meiksin, A., Merelli, A., Monet, D. G., Munn, J. A., Narayanan, V. K., Nash, T., Neilsen, E., Neswold, R., Newberg, H. J., Nichol, R. C., Nicinski, T., Nonino, M., Okada, N., Okamura, S., Ostriker, J. P., Owen, R., Pauls, A. G., Peoples, J., Peterson, R. L., Petravick, D., Pier, J. R., Pope, A., Pordes, R., Prosapio, A., Rechenmacher, R., Quinn, T. R., Richards, G. T., Richmond, M. W., Rivetta, C. H., Rockosi, C. M., Ruthmansdorfer, K., Sandford, D., Schlegel, D. J., Schneider, D. P., Sekiguchi, M., Sergey, G., Shimasaku, K., Siegmund, W. A., Smee, S., Smith, J. A., Snedden, S., Stone, R., Stoughton, C., Strauss, M. A., Stubbs, C., SubbaRao, M., Szalay, A. S., Szapudi, I., Szokoly, G. P., Thakar, A. R., Tremonti, C., Tucker, D. L., Uomoto, A., Vanden Berk, D., Vogeley, M. S., Waddell, P., Wang, S.-i., Watanabe, M., Weinberg, D. H., Yanny, B., Yasuda, N., and SDSS Collaboration (2000). The Sloan Digital Sky Survey: Technical Summary. *The Astronomical Journal*, 120(3):1579–1587.

Zaroubi, S., Hoffman, Y., Fisher, K. B., and Lahav, O. (1995). Wiener Reconstruction of the Large-Scale Structure. *Astrophysical Journal*, 449:446.

Zel'dovich, Y. B. (1970). Gravitational instability: An approximate theory for large density perturbations. *Astronomy & Astrophysics*, 5:84–89.

Zhu, X., Vondrick, C., Fowlkes, C. C., and Ramanan, D. (2016). Do we need more training data? *International Journal of Computer Vision*, 119(1):76–92.