

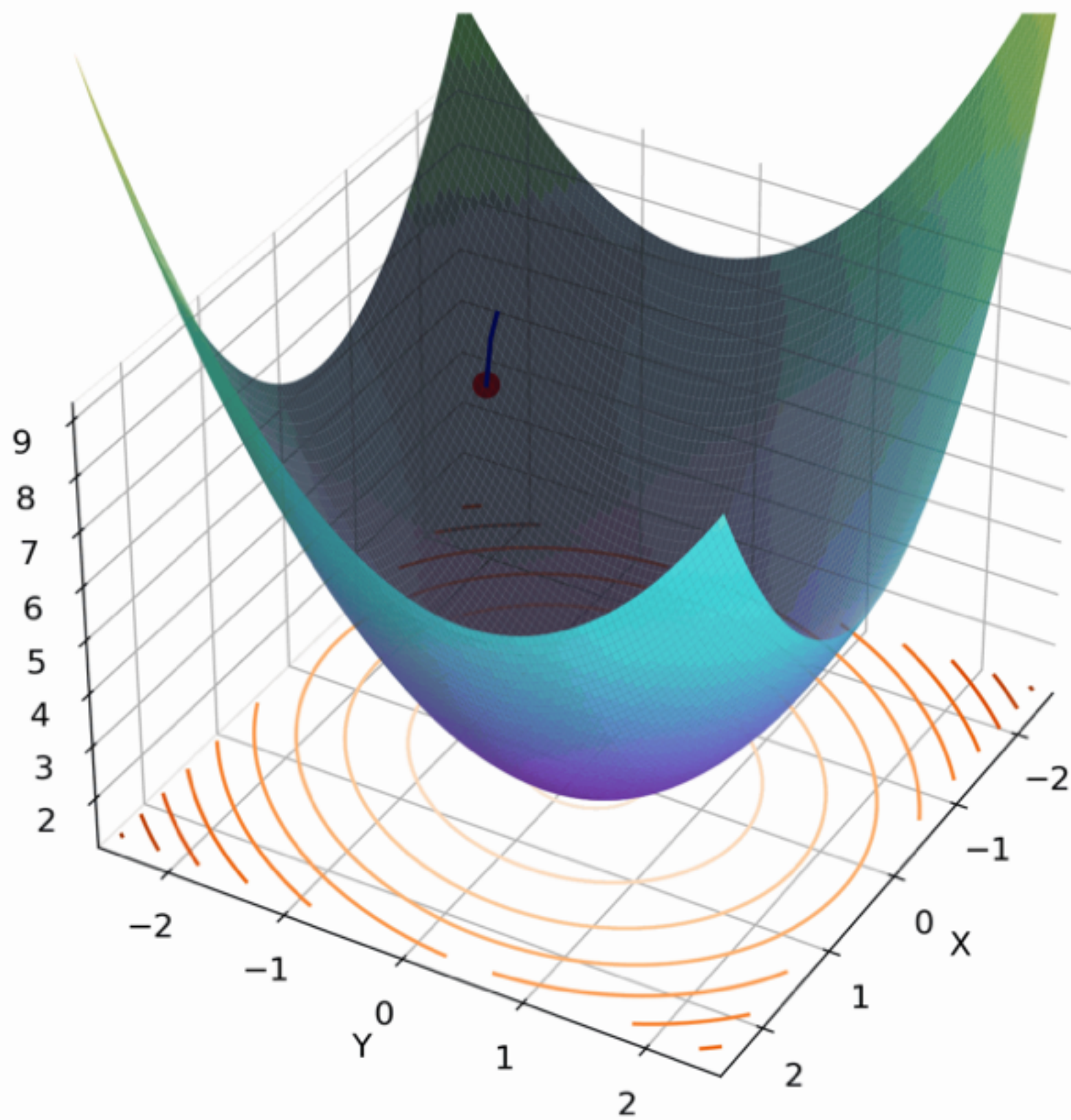




I N S E A







## Hamiltonian Monte-Carlo (*Duane 1987, Neal 1996*)



MC MC: algorithmes avancés



Hamiltonian Dynamics

Engineering, algorithms, Metropolis/MLA, Alex, efficient, space

**I** **d** **é** **e** **:** **a** **u** **l** **i** **e** **u** **d'** **e** **x** **p** **l** **o** **r** **e** **r** **a** **v** **e** **c** **u** **n** **e** **p** **r** **o** **s** **i** **t** **i** **o** **n** **a** **l** **é** **a** **t** **o** **i** **r** **e** **x**, **s** **i** **m** **u** **l** **e** **r** **u** **n** **e** **v** **i** **t** **e** **s** **e** **i** **n** **i** **t** **i** **a** **l** **e** **a** **l** **é** **a** **t** **o** **i** **r** **e**.

En physique, on parle du moment:  $\vec{p} = m\vec{v}$

1. Simulacrum vector **moment** **p**' (direction vite)

2. Déterminer la trajectoire

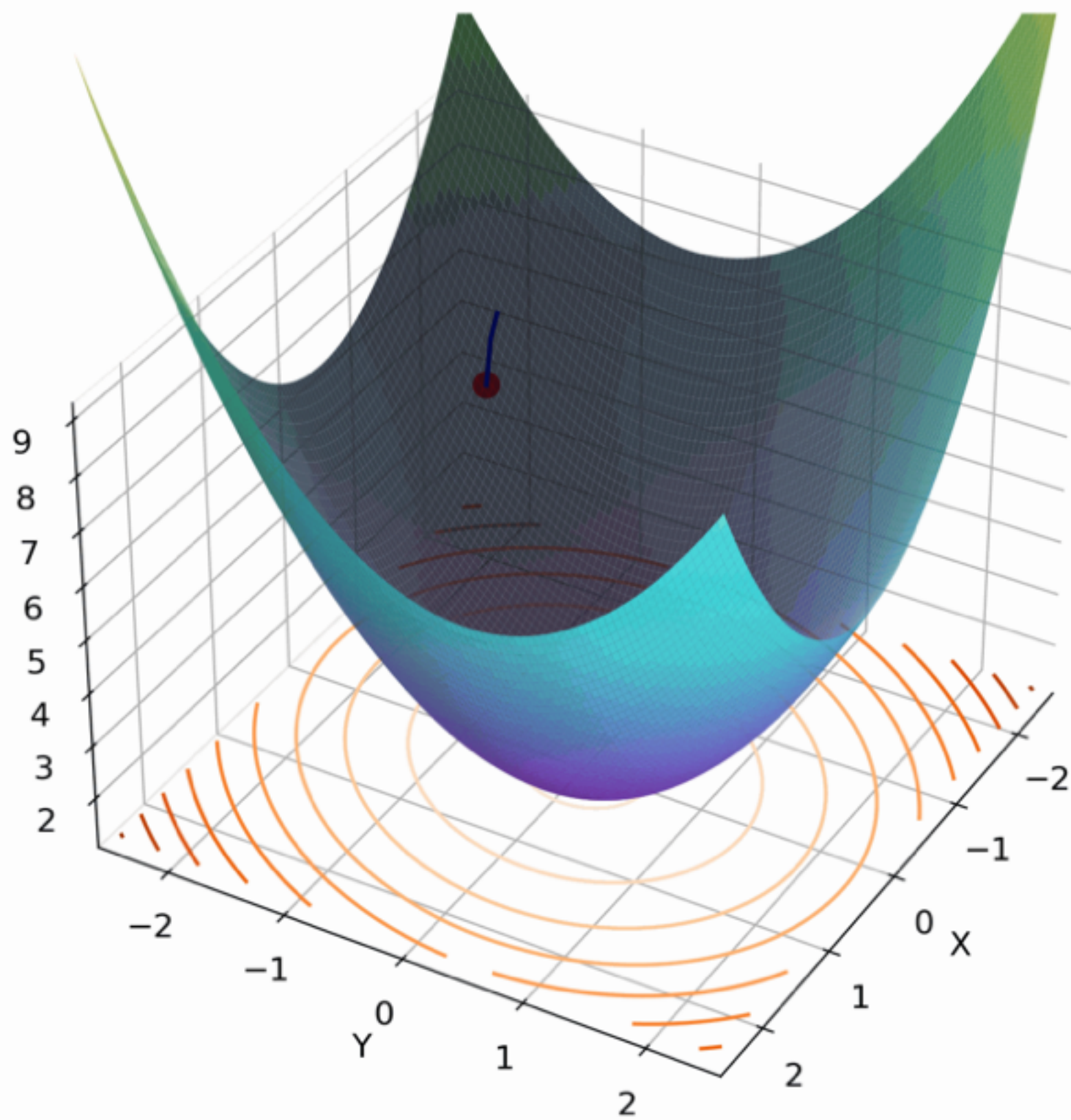
3. Suivre la trajectoire arrêtée (nouveau)

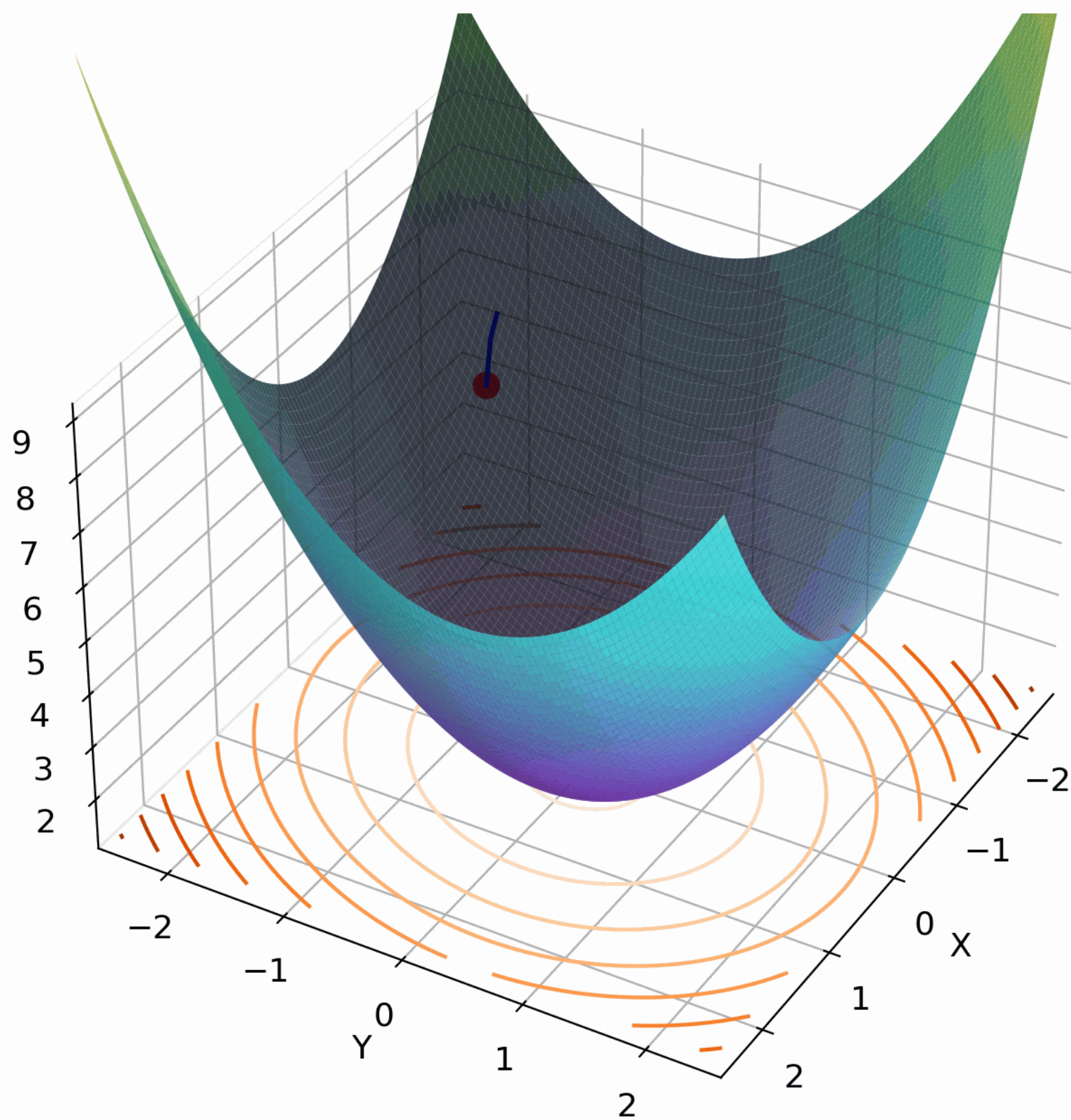
4. **Accepter ou rejeter** ( $x'$ ,  $p'$ )



5. Répéter

*(anninmate)*



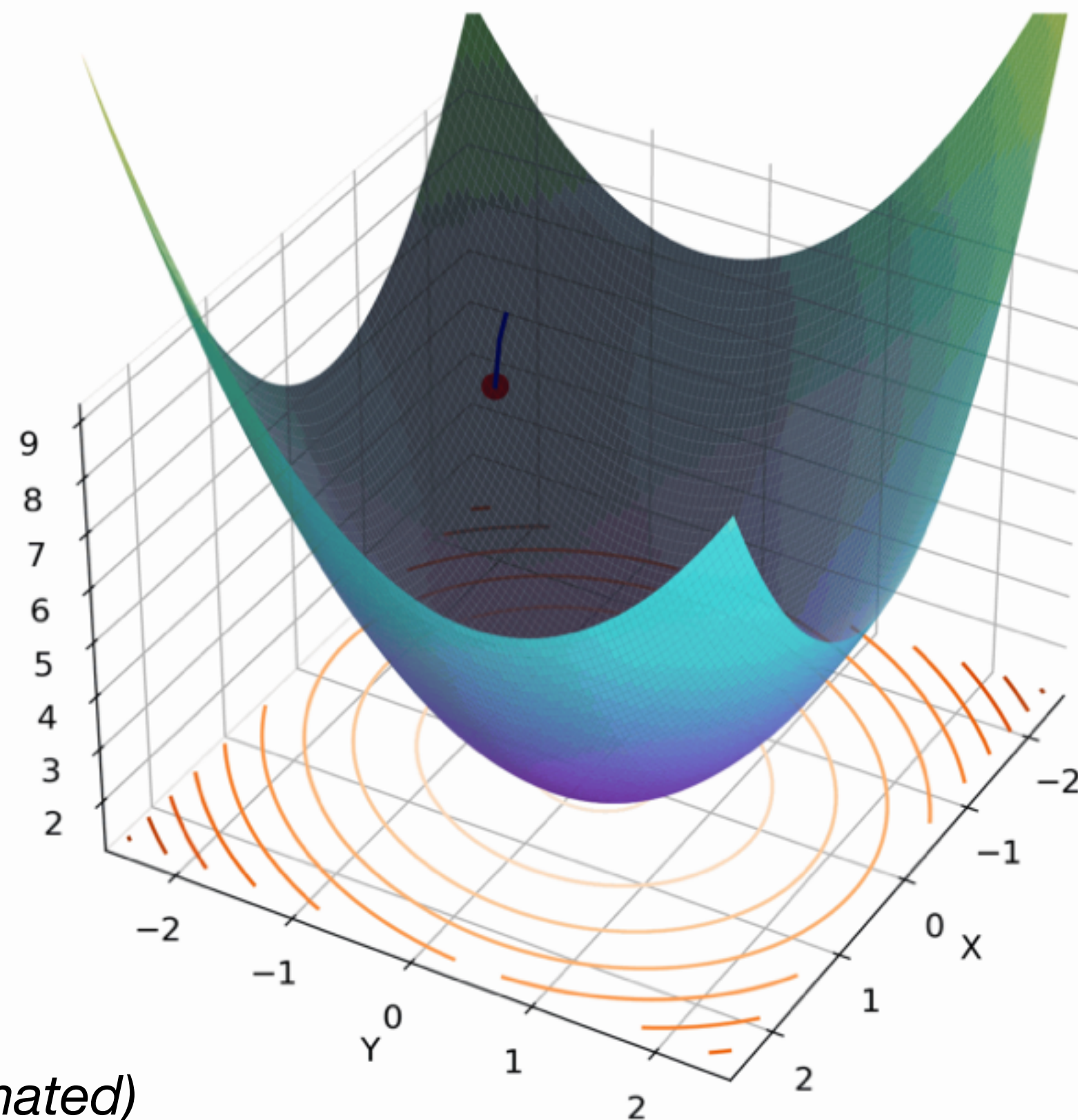




En grande dimension, les algorithmes Metropolis / MALA explorent inefficacement l'espace.

**Idée:** au lieu d'explorer avec une position aléatoire  $x$ , simuler une **vitesse initiale** aléatoire.

En physique, on parle du moment:  $\vec{p} = m\vec{v}$



(animated)

## Hamiltonian Monte-Carlo (Duane 1987, Neal 1996)

1. Simuler un vecteur de **moment  $p'$**  (direction et vitesse)
2. Déterminer la trajectoire
3. Suivre la trajectoire et s'arrêter (**nouveau  $x'$** )
4. Accepter ou rejeter ( **$x'$** ,  **$p'$** )
5. Répéter



1. Pourquoi Monte-Carlo ? (Exemple de modèle hiérarchique)
2. Introduction à la méthode Monte-Carlo (historique, PRNG)
3. Algorithmes de simulation i.i.d (PRNG, transformation, rejet)
4. Méthodes MCMC (Gibbs, Metropolis)
5. Diagnostics de convergence MCMC
6. Méthodes MCMC avancées (Langevin, HMC, NUTS)



## Hamiltonian Monte-Carlo (*Duane 1987, Neal 1996*)

1. Simuler un vecteur de **moment  $p'$**  (direction et vitesse)
2. Déterminer la trajectoire
3. Suivre la trajectoire et s'arrêter (**nouveau  $x'$** )
4. Accepter ou rejeter ( **$x'$** ,  **$p'$** )
5. Répéter

