# Soutenance de stage

# Réduction du modèle de Vlasov-Poisson guidée par les données

Guillaume Steimer

***Dirigé par :*** Emmanuel Franck, Vincent Vigon, Laurent Navoret & Nicolas Crouseilles

**M2 CSMI**
**UFR de Mathématiques et d'Informatique**
**Université de Strasbourg**

**26 août 2021**

# Introduction

- modèle de plasma : Vlasov-Poisson $\rightarrow$ EDP,
- discrétisation $\rightarrow$ EDO en grande dimension : solution $u$ paramétrée,
- technique (MOR : Model Order Reduction) :
  - réduction $\rightarrow$ EDO en petite dimension (modèle réduit),
  - solution $\overline{u}$ proche de $u$,
  - pour une plage restreinte de paramètres,
- intérêts :
  - réduire la complexité, le coût calcul numérique,
  - résultats en temps réel ou en grand nombre,
- outils :
  - apprentissage machine profond,
  - mécanique hamiltonienne.

# Introduction

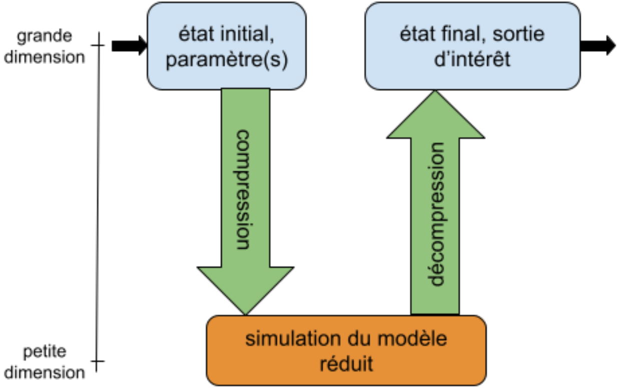



Figure – Schéma du processus

# Introduction

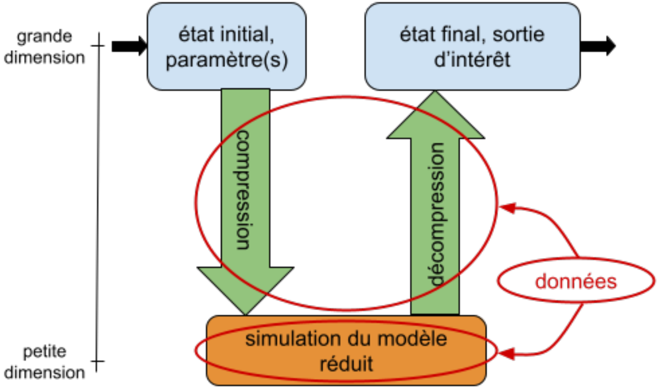



Figure – Schéma du processus, étapes par apprentissage machine en rouge

# Le plasma

- 4ème état de la matière,
- « gaz » chauffé intensément ($10^2 - 10^9 K$),
- $\rightarrow$ « soupe » très active de particules chargées ou non,
- dynamique :
  - dominée par des champs électromagnétiques (auto-induit ou externe),
  - fortement non linéaire,
- 99% de la matière connue (étoiles, nébuleuses...) mais très rare sur Terre.

# Le plasma

- intérêt : plasma dans un réacteur $\rightarrow$ fusion thermonucléaire $\rightarrow$ grande production d'énergie très peu polluante.

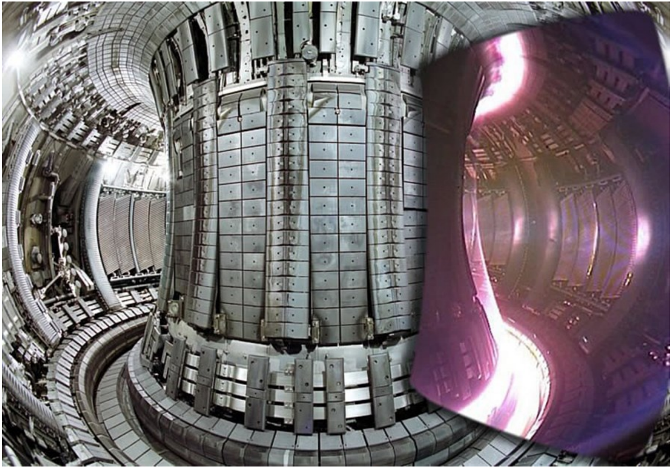


Figure – JET (Joint European Torus) avec et sans plasma

# Modèle de Vlasov-Poisson

- plasma 1D sans collisions, non relativiste, limite électrique $|E + E_{ext}| \gg c|B|$,
- modèle cinétique : distribution statistique $f(x, v, t)$ de particules,
- équation de $\boxed{\text{Vlasov}}$-$\boxed{\text{Poisson}}$ :

$$\boxed{\partial_t f + v\nabla_x f + \frac{q}{m}(E(f) + E_{ext})\nabla_v f = 0},$$

$$E = -\nabla\phi \quad , \boxed{\Delta\phi = -\rho},$$

- $x(t), v(t)$ position et vitesse, $t$ temps, $q, m$ charge et masse,
- $\phi(x(t))$ potentiel électrique, $\rho(x(t)) = q\int f dv$ densité de charge,
- $E, E_{ext}$ champs électriques auto-induit et externe (donné).
- $\rightarrow$ équation de transport (fortement) non linéaire !

# Discrétisation

- distribution discrète $f_N$ de $N$ particules de positions et vitesses $(x_i(t), v_i(t))$ :

$$f_N(x(t), v(t), t) = \sum_{i=1}^{N} w_i \delta(x - x_i(t)) \delta(v - v_i(t))$$

- conditions initiales $(x^0, v^0)$, on aboutit à une EDO en grande dimension $2N$ :

$$\begin{cases} \dot{x}_i = v_i, \\ \dot{v}_i = \frac{q}{m}(E + E_{ext})(x_i), \\ x_i(0) = x_i^0, v_i(0) = v_i^0. \end{cases}$$

- $E$ dépend de tous les $x_i$ !
- point de départ de notre méthode.

- étape suivante : générer des données suivant cette EDO,
- avant : notions de mécanique Hamiltonienne.

# Mécanique Hamiltonienne

- cas d'un système conservatif ou hamiltonien ; on note $x = (x_i), v = (v_i) \in \mathbb{R}^N$,
- $u = \begin{pmatrix} x \\ v \end{pmatrix} \in \mathbb{R}^{2N}$ suit une dynamique hamiltonienne associé à $\mathcal{H} : \mathbb{R}^{2N} \to \mathbb{R}$ si $\dot{u} = \mathbb{J}_{2N} \nabla_u \mathcal{H}(u)$,
- $\mathbb{J}_{2N} := \begin{pmatrix} 0 & \mathbb{I}_N \\ -\mathbb{I}_N & 0 \end{pmatrix}$
- formulation symplectique : $\forall t > 0, \mathcal{H}(u(t)) = \mathcal{H}(u(0))$,
- $\mathcal{H}$ : hamiltonien associé, en physique : énergie du système,
- nécessité de schémas numériques symplectiques $\to$ stabilité numérique.

# Mécanique Hamiltonienne

- ici :

$$\mathcal{H}(u(t)) = \sum_{i=1}^{N} \left[ \frac{1}{2} v_i^2(t) + \frac{q}{m} \phi(x_i(t)) \right]$$

- hamiltonien séparé $\mathcal{H}(u) = \mathcal{H}^1(x) + \mathcal{H}^2(v)$,

# Algorithme PIC

- calcul de $E$ en 3 étapes : calcul de la densité $\rho \to$ du potentiel $\Delta\phi = -\rho \to E = -\nabla\phi$
- difficulté de l'EDO : calcul de $E$ à chaque pas $\to \mathcal{O}(N^2)$,
- idée : passer par un maillage de taille $M \ll N \to \mathcal{O}(MN)$,
- aboutit à un algorithme PIC (Particle In Cell), en 2 étapes répétées :
  - particle mover : résolution de la dynamique sur un pas de temps,
  - field solver : calcul de $E$ en passant par un maillage.

# Particle mover

- résolution de l'EDO sur un pas de temps,
- on note $(x^n, v^n)$ position et vitesse au n-ième pas de temps,
- schéma symplectique : Störmer-Verlet (ordre 2) :

$$\begin{cases} \frac{1}{\Delta t}\left(x^{n+1} - x^n\right) = v^{n+\frac{1}{2}} \\ \frac{1}{\Delta t}\left(v^{n+\frac{1}{2}} - v^{n-\frac{1}{2}}\right) = \frac{q}{m}\left(E_{ext}(x^n) + \boxed{E^n(x^n)}\right) \end{cases}$$

- schéma rapide et précis,
- champ $\boxed{E^n(x^n)}$ auto-consistant connu.

# Field solver

- calcul de $E^n(x^n)$ par un champ moyen $E^n_M$ sur un maillage de $M$ noeuds séparés par $\Delta x$, 4 étapes :
- 1 : calcul de $\rho^n_M = (\rho^n_i)$ à chaque noeud, boucle sur chaque particule :
  - particule de position $\tilde{x}$ : répartir charge $q$ entre les noeuds $i = \lfloor \tilde{x}/\Delta x \rfloor$ et $i+1$ voisins :

$$\rho^n_i \mathrel{+}= \overline{\rho}\frac{\Delta x - h}{\Delta x}, \quad \rho^n_{i+1} \mathrel{+}= \overline{\rho}\frac{h}{\Delta x} \quad \text{où} \quad \overline{\rho} = \frac{q}{\Delta x}.$$

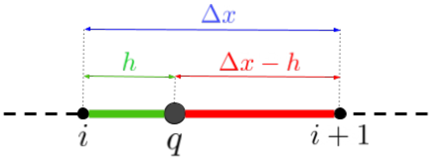



Figure – Illustration de la répartition de la charge

## Field solver

- 2 : calcul de $\Delta\phi_M^n = -\rho_M^n$ par différences finies,
- 3 : calcul de $E_M^n = -\nabla\phi_M^n$ par différences finies centrées,
- 4 : interpolation de $E_M^n$ en $x^n \to E^n(x^n)$.

# Field solver

- illustration du field solver avec conditions aux bords de Neumann, $N = 3, M = 6, q/m = +1$,
- calcul de la densité et du potentiel :

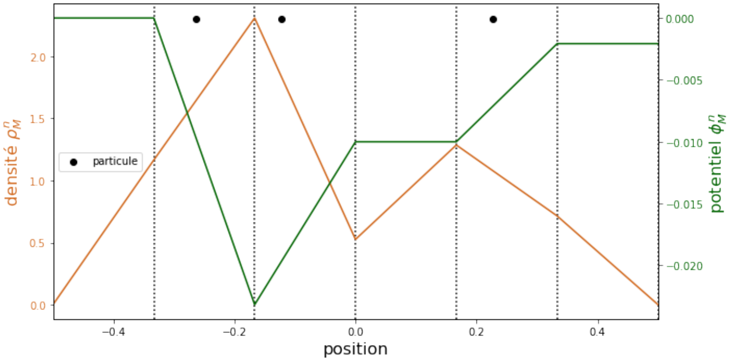



Figure – densité $\rho_M^n$ et potentiel $\phi_M^n$ sur grille

# Field solver

- illustration du field solver avec conditions aux bords de Neumann, $N = 3, M = 6, q/m = +1$,
- calcul du champ électrique sur grille et interpolation aux particules :

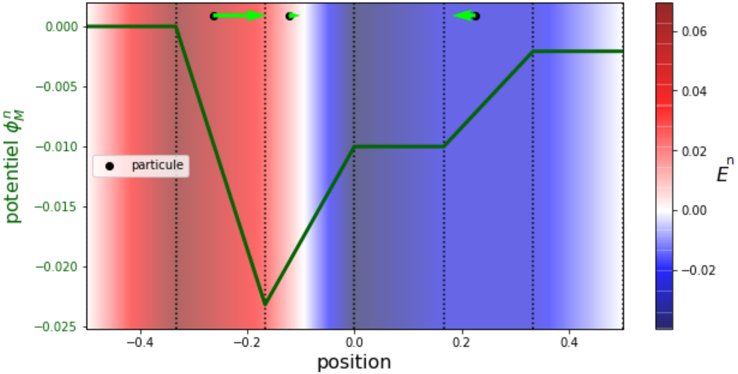



Figure – potentiel $\phi_M^n$ et champ $E_M^n$ sur grille

# Réduction

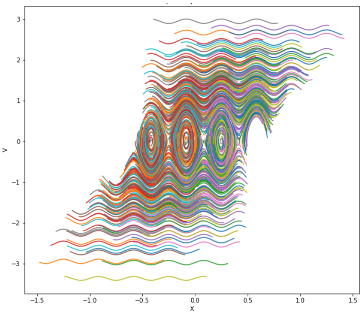
- EDO en dimension $2N$ très coûteuse,
- solution : trouver des vitesses-positions réduites $\overline{u} = \begin{pmatrix} \overline{x} \\ \overline{v} \end{pmatrix} \in \mathbb{R}^{2K}, K \ll N$ qui :
  - suivent une EDO réduite (symplectique) en dimension $2K$,
  - fidèle à la solution $u \in \mathbb{R}^{2N}$ paramétrée (par $t$, $u^0$),
  - MAIS pour un domaine (restreint) de paramètres,
  - compression-décompression $u \leftrightarrow \overline{u}$ rapide,

## Limite des méthodes classiques

- PSD (Proper Symplectic Decomposition) [4] inspirée de la POD (Proper Orthogonal Decomposition),
- hypothèse : linéarité de $u$ par rapport aux paramètres $\rightarrow$ réduction linéaire $u = A\overline{u}$,
- fournit un modèle réduit symplectique,
- hypothèse suffisante avec une EDP linéaire ($E \equiv 0$),
- mais insuffisante dans le cas fortement non linéaire ($E \not\equiv 0$).

# Limite des méthodes classiques

- exemple dans le cas linéaire, $N = 1000$ particules, $K = 10$ et $E_{ext}(x) = 3\cos(6\pi x)$, on fixe $q/m = -1$.



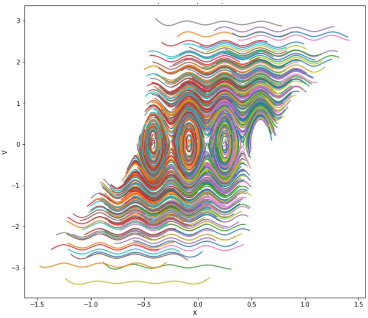




Figure – Trajectoires de référence par PIC (gauche) et obtenues par PSD (droite)

## Limite des méthodes classiques

- exemple : cas non linéaire, $N = 1000$ particules confinées par $E_{ext}$ : zone libre de taille $L_{libre} = 1.2$ et d'intensité $I$,

$$E_{ext}(x) = \begin{cases} I\left(x + \frac{L_{libre}}{2}\right)^3 & \text{si } x < -\frac{L_{libre}}{2}, \\ 0 & \text{si } -\frac{L_{libre}}{2} < x < \frac{L_{libre}}{2}, \\ I\left(x - \frac{L_{libre}}{2}\right)^3 & \text{sinon} \end{cases}$$

- rappel : accélération : $-E_{ext}$

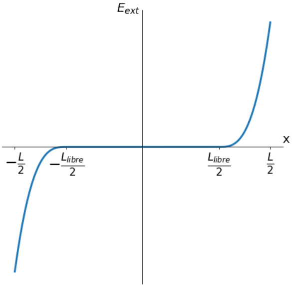



Figure – Champ confinant

- $K = 35$ (grand).

# Limite des méthodes classiques

- trajectoires des particules dans l'espace des phases $(x, v)$ :

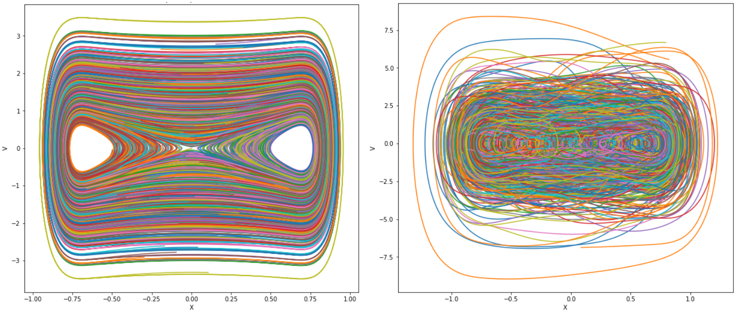



Figure – Trajectoires de référence par PIC (gauche) et obtenues par PSD (droite)

# Réduction par auto-encodeur

- nouvelle technique basée sur des réseaux de neurones = fonction paramétrique :
  - succession de couches, la sortie d'une est l'entrée de la suivante,
  - couche : entrée $z$, sortie $\sigma(Wz + b)$ avec $W, b$ poids et biais, $\sigma$ une non-linéarité,
  - ensemble des $W, b \rightarrow$ paramètres $\theta$ du réseau,
  - $\theta$ ajusté sur des données pour minimiser un coût ou loss $\mathcal{L}$,
- $\rightarrow$ auto-encodeur (AE) !

# Réduction par auto-encodeur

- représentation de l'identité par un code intermédiaire de petite dimension,
- encodeur $\mathcal{F}_\theta(u) = \overline{u}$, décodeur $\mathcal{G}_\theta(\overline{u}) = u$,
- architecture optimisée au problème,

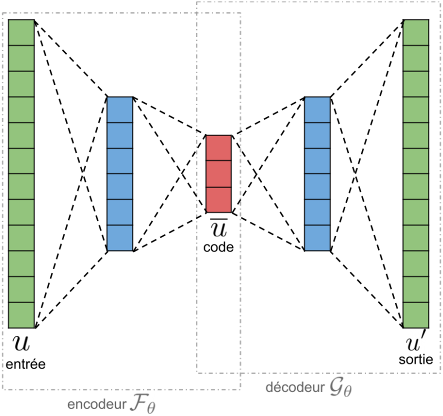



Figure – Auto-encodeur

# Réduction par auto-encodeur

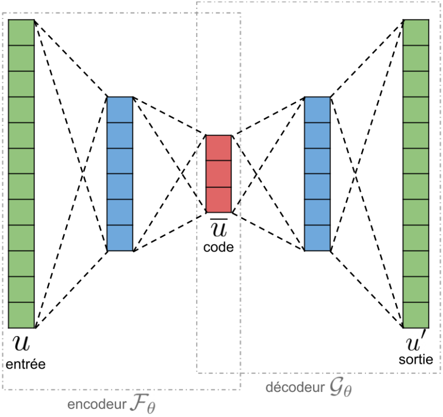



Figure – Auto-encodeur

- entraînement : $\mathcal{L}_{AE} = \|u - (\mathcal{G}_\theta \circ \mathcal{F}_\theta)(u)\|_2$,
- apprentissage non supervisé,
- hypothèse : $\overline{u}$ suit une dynamique symplectique,
- MAIS ne fournit pas le modèle réduit.

# Modélisation de la dynamique réduite

- apprentissage du modèle réduit par réseau de neurones hamiltonien (HNN) [2],
- apprend l'hamiltonien réduit $\mathcal{H}_\theta(\overline{u}) = \mathcal{H}_\theta^1(\overline{x}) + \mathcal{H}_\theta^2(\overline{v})$,
- entrée $\overline{u}$ (encodeur), sortie $\dot{\overline{u}}$ (par différences finies),
- loss :

$$\mathcal{L}_{HNN} = \left\| \nabla_{\overline{v}} \mathcal{H}_\theta^2(\overline{v}(t)) - \frac{\overline{x}(t+\Delta t) - \overline{x}(t-\Delta t)}{2\Delta t} \right\|_2 + \left\| \nabla_{\overline{x}} \mathcal{H}_\theta^1(\overline{x}(t)) + \frac{\overline{v}(t+\Delta t) - \overline{v}(t-\Delta t)}{2\Delta t} \right\|_2$$

- modèle réduit symplectique : stabilité,
- calcul de la dynamique avec formulation symplectique et schéma de Verlet.

# Résumé du processus

- phase hors ligne (coûteuse, lente) :
  - génération d'une base de données par PIC,
  - entraînement AE et HNN,
- phase en ligne (rapide) :
  - compression des conditions initiales par encodeur (AE),
  - simulation du modèle réduit (HNN),
  - décompression des conditions finales ou autre sortie d'intérêt par décodeur (AE).

## Résultats

- $N = 1000, K = 7$ (ratio $\sim$ 140),
- conditions initiales : distribution normal en vitesse et bêta $\mathcal{B}(\alpha, \beta)$ en position, $\beta = 1.5$ fixe,
- paramètre de réduction : $\alpha \in \{2.2, 2.6, 3.0, 3.4, 3.8\}$,
- trajectoire d'une particule :

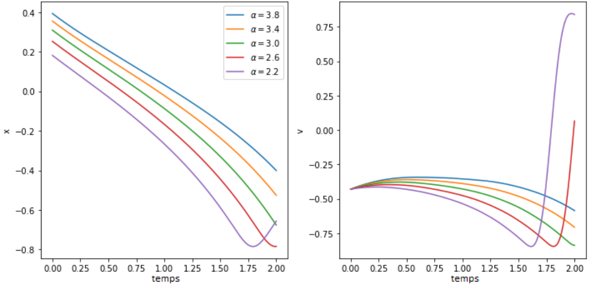
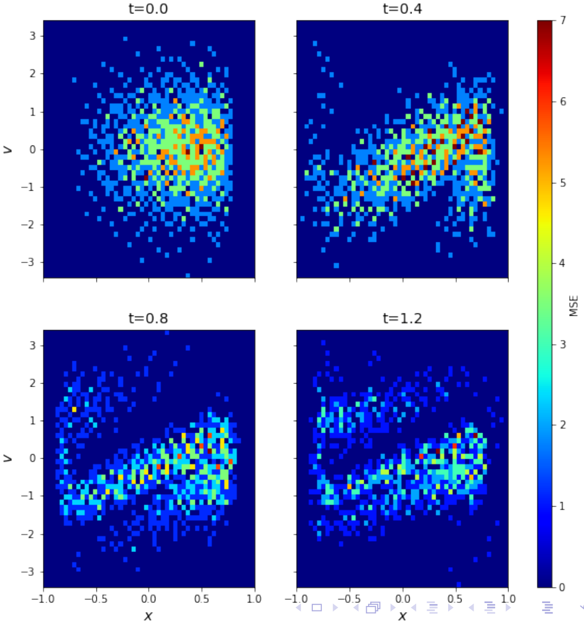



Figure – position (gauche) et vitesse (droite) en fonction du temps

## Résultats

- évolution de $f$ :



# Résultats

- processus 25 fois plus rapide,
- quelques erreurs en fonction du temps :

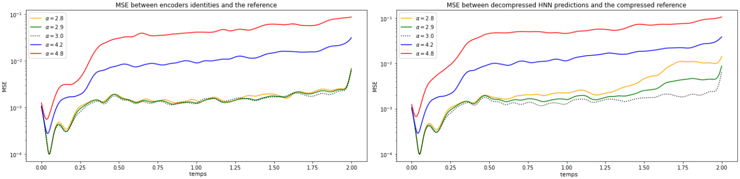



Figure – MSE en fonction du temps pour l'identité de l'AE et la prédiction du HNN

## Résultats

- erreur en fonction de $\alpha$ :

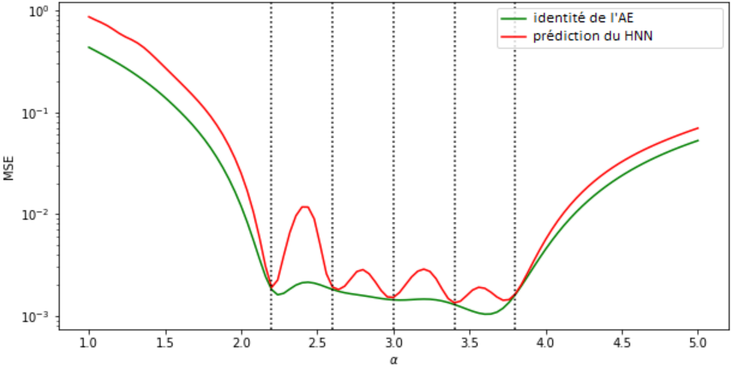



Figure – MSE en fonction du temps pour l'identité de l'AE et la prédiction du HNN

# Conclusion

- atouts : processus efficace, accélération par rapport à PIC, grand ratio de réduction (jusqu'à 700), stable en temps long, scalabilité, paramétrisation,
- faiblesses : construction AE, permutations des particules, domaine périodique,
- ouvertures : passage en 3D avec champ magnétique, dans un tokamak, autre modèle cinétique...

# Références


Régine Barthelmé. « Le problème de conservation de la charge dans le couplage des équations de Vlasov et de Maxwell. ». Thèse. 7, rue René Descartes 67084 Strasbourg Cedex (FRANCE) : Université Louis Pasteur et CNRS (UMR 7501), 2005.

Sam Greydanus, Misko Dzamba et Jason Yosinski. *Hamiltonian Neural Networks*. `https://greydanus.github.io/2019/05/15/hamiltonian-nns/`.

Eric Sonnendrücker. *Numerical Methods for the Vlasov-Maxwell equations*. Springer, 2015.

Tomasz M. Tyranowski et Michael Kraus. « Symplectic model reduction methods for the Vlasov equation ». In : (2019). arXiv : `1910.06026 [physics.comp-ph]`.

# ANNEXE : Algorithme PIC : illustration

- avec un double faisceau perturbé d'électrons, $N = 400\,000$, $M = 1\,000$, $q/m = -1$,
- calcul du potentiel :

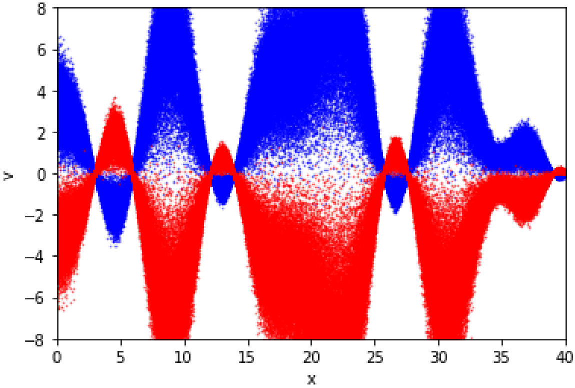



Figure – à $t = 0$

# ANNEXE : Algorithme PIC : illustration

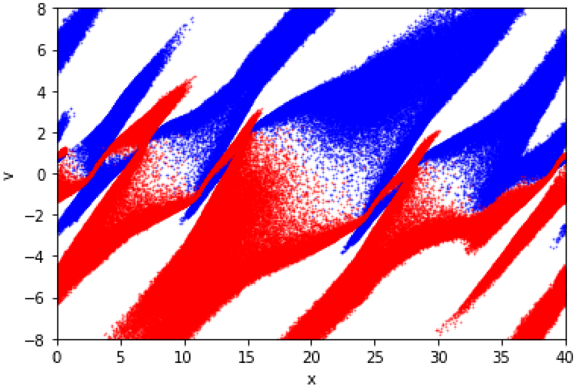



Figure – à $t = 1.5$

# ANNEXE : Algorithme PIC : illustration

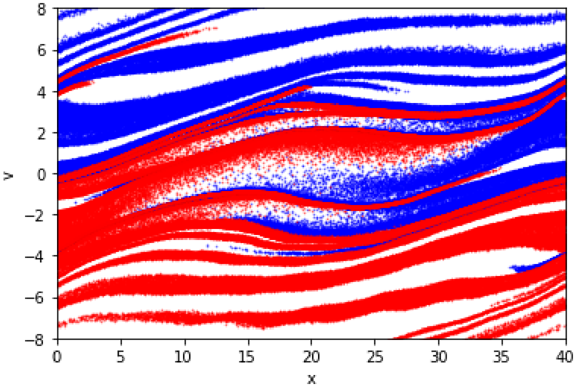



Figure – à $t = 8$

# ANNEXE : Algorithme PIC : illustration

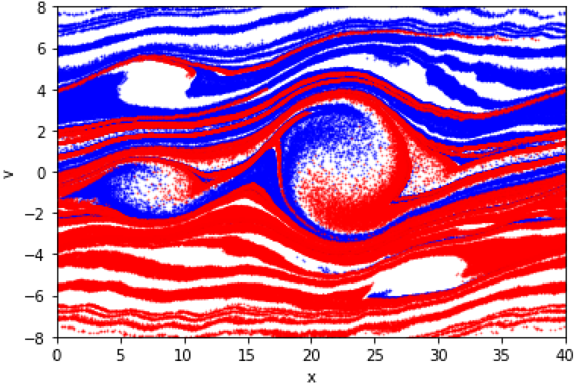



Figure – à $t = 17.5$