OAK RIDGE National Laboratory

# A machine learning normalizing flow surrogate model for runaway electron kinetic simulations

D. del-Castillo-Negrete
Fusion Energy Division
ORNL

Work done in collaboration with:
M.Yang, G. Zhang (ORNL)
P. Wang and Y. Cao (Auburn University)

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

OAK RIDGE National Laboratory

# Transport problem of interest

- We are interested in the solution of Fokker-Planck equations of the form

$$\frac{\partial f}{\partial t} + \sum_{i=1}^{d} \frac{\partial}{\partial x^i}[b_i\, f] = \sum_{i,j=1}^{d} \frac{\partial^2}{\partial x^i x^j}\,[K_{ij}\, f]\,, \qquad f(x_i, t=0) = f_0(x_i)$$

  where $b_i = b_i(x_1, \ldots x_d, t)$ is the drift, $K_{ij} = K_{ij}(x_1, \ldots x_d, t)$ the diffusion tensor, $f = f(x_1, \ldots x_d)$ the distribution function in $\mathbb{R}^d$ with coordinates $(x_1, \ldots x_d)$, and $f_0(x_i)$ the initial condition.
- This is a problem of significant interest in many areas, including plasma physics and in particular runaway electrons.
- In general, this is a computational challenging problem because of spatiotemporal multi-scale dynamics in $b_i$ and $K_{ij}$, complex domains, and high-dimesionality.
- Our goal is to present a machine learning method to overcome some of these difficulties.

# Particle-based approach

- The Fokker-Planck partial differential equation of interest is mathematically equivalent to a system of stochastic differential equations of the form

$$d\mathbf{x}_t = \boldsymbol{b}(\mathbf{x}, t)dt + \boldsymbol{\sigma}(\mathbf{x}, t)d\mathbf{W}_t\,, \qquad \mathbf{x}_{t=0} = \mathbf{x}_0$$

  where $d\mathbf{W}_t$ is a Wiener stochastic process (Brownian motion)
- This equivalence is at the heart of the extensively used Monte-Carlo method for the solution of transport problems in plasma physics.
- For example, in the study of runaway electrons the drift term, $\boldsymbol{b}(\mathbf{x}, t)dt$, includes electric field acceleration, Coulomb drag, and velocity drifts, and the stochastic term $\boldsymbol{\sigma}(\mathbf{x}, t)d\mathbf{W}$ models collisional processes, e.g., pitch angle scattering.

$$\begin{aligned} dp &= \left[E\xi - \frac{\gamma p}{\tau}\left(1-\xi^2\right) - C_F + \frac{1}{p^2}\frac{\partial}{\partial p}\left(p^2 C_A\right)\right]dt + \sqrt{2C_A}\,dW_p \\ d\xi &= \left[\frac{E\left(1-\xi^2\right)}{p} + \frac{\xi\left(1-\xi^2\right)}{\tau\gamma} - 2\xi\frac{C_B}{p^2}\right]dt + \frac{\sqrt{2C_B}}{p}\sqrt{1-\xi^2}\,dW_\xi \end{aligned}$$

# Green's function formulation

- Machine learning (ML) methods for this problem are typically based on using known or precomputed solutions to train an algorithm (e.g., a neural network) that could then be used as a black box to obtain new solutions.
- Two potential limitations of this approach are the cost of the training and the ability of the algorithm to go beyond the domain of the training.
- A particular challenge of the initial value problem of the Fokker-Planck equation is that the solution $f(x_i, t)$ depends on $f_0(x)$ and every time the initial condition changes the algorithm must be retrained.
- To circumvent this limitation we will consider the solution in terms of the Green's function $G(x_i, x'_j, t)$

$$f(x_i, t) = \int dx'_1 \ldots \int dx'_d \, G(x_i, x'_j, t) f_0(x'_j)$$

and develop a machine learning surrogate model of the Green's function.

# Machine learning based surrogate model of transition probability

- In the particle based formulation the Green's function $G(x_i, x_j', t)$ corresponds to the conditional (transition) probability $p(\mathbf{x}_t|\mathbf{x}_0)$ for a particle to be at $\mathbf{x}_t$ at time $t$ if it was $\mathbf{x}_0$ at time $t = 0$.
- Given an ensemble of particles with initial conditions $\{\mathbf{x}_0\}$ with probability density $p(\mathbf{x}_0)$, the probability density, $p(\mathbf{x}_t)$, of the final positions of the particles $\{\mathbf{x}_t\}$ evolving under the stochastic differential equation

$$d\mathbf{x}_t = \boldsymbol{b}(\mathbf{x}, t)dt + \boldsymbol{\sigma}(\mathbf{x}, t)d\mathbf{W}_t\,, \qquad \mathbf{x}_{t=0} = \mathbf{x}_0$$

  is given by

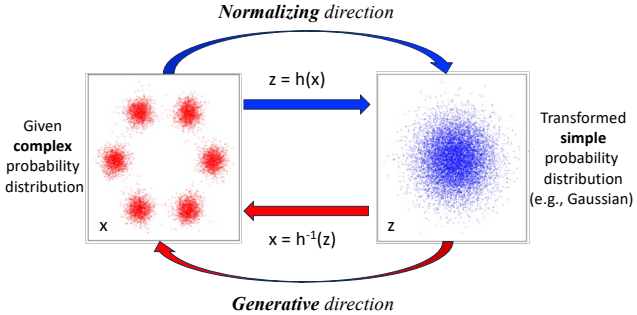
$$p(\mathbf{x}_t) = \int d\mathbf{x}_0\, p(\mathbf{x}_t|\mathbf{x}_0)p(\mathbf{x}_0)$$

- Our goal is to develop a surrogate model for $p(\mathbf{x}_t|\mathbf{x}_0)$ based on normalizing flows, a powerful machine learning method to estimate and sample arbitrary probability distribution functions.

# Normalizing flow

A normalizing flow is an invertible transformation that maps a complex probability distribution into a simple distribution

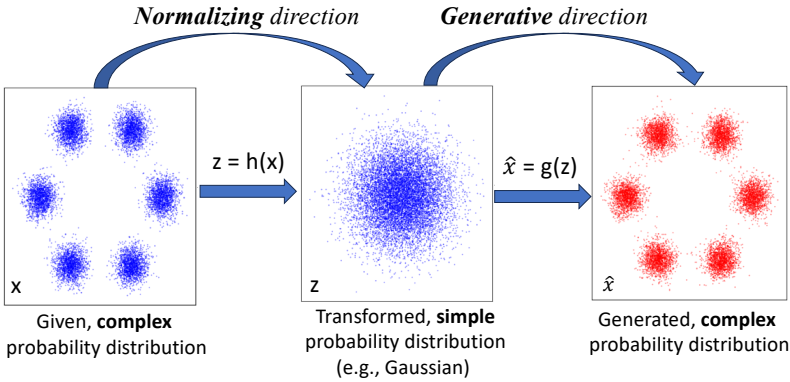
$$p_X(\boldsymbol{x}) = p_Z(\boldsymbol{z}) \left| \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} \right| = p_Z(\boldsymbol{h}(\boldsymbol{x})) |\det \mathbf{J}_h(\boldsymbol{x})|$$




With a normalizing flow we can generate samples $\{\boldsymbol{x}_m\}_{m=1}^M$ of $p_X(\boldsymbol{x})$ by mapping samples $\{\boldsymbol{z}_m\}_{m=1}^M$ of a simple (e.g., Gaussian) distribution using $\boldsymbol{x} = \boldsymbol{h}^{-1}(\boldsymbol{z})$

# Pseudo-reversible normalizing flow

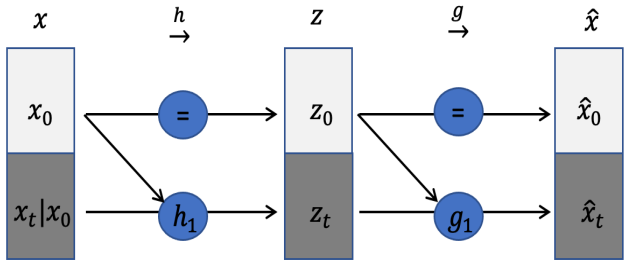
In general the inverse transformation is not known or it is computationally expensive



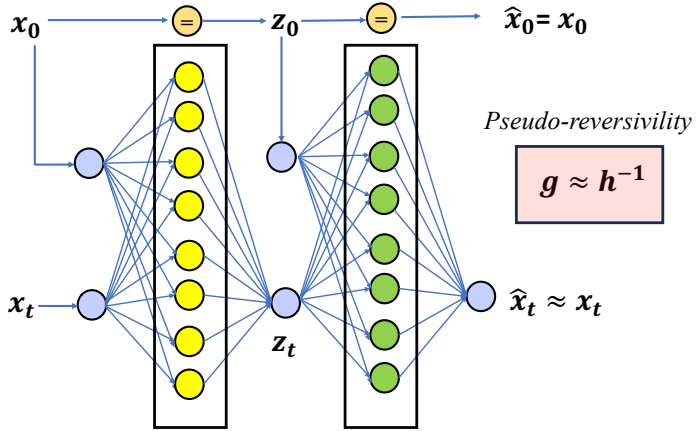

Pseudo-reversible normalizing flows relax the exact invertibility by imposing $\mathbf{g} \approx \mathbf{h}^{-1}$

# Architecture of pseudo-reversible flow surrogate model for $p(\mathbf{x}_t|\mathbf{x}_0)$

- We start with a uniform distribution of initial conditions $\{\mathbf{x}_0^{(i)}\}$ for $i = 1, \ldots N$ and solve the SDE to get $\{\mathbf{x}_t^{(i)} \mid \mathbf{x}_0^{(i)}\}$ which by definition are distributed according to the (unknown) transition probability $p(\mathbf{x}_t|\mathbf{x}_0)$
- The normalizing flow function $\mathbf{h}_1$ transforms $\{\mathbf{x}_t^{(i)} \mid \mathbf{x}_0^{(i)}\}$ into the Gaussian distributed random variable $\{\mathbf{z}_t^{(i)}\}$.
- The function $\mathbf{g}_1 \approx \mathbf{h}_1^{-1}$ transform the Gaussian variable, $\mathbf{z}_t$, and the initial condition, $\mathbf{z}_0 = \mathbf{x}_0$, into $\{\hat{\mathbf{x}}_t^{(i)}\} \approx \{\mathbf{x}_t^{(i)} \mid \mathbf{x}_0^{(i)}\}$ keeping $\hat{\mathbf{x}}_0 = \mathbf{z}_0 = \mathbf{x}_0$
- Once trained, the function $\mathbf{g}_1$ is the surrogate model of $p(\mathbf{x}_t|\mathbf{x}_0)$.



# Neural Network computation of pseudo-reversible flow functions




The parameters $\{\theta_{\mathbf{h}}, \theta_{\mathbf{g}}\}$ are obtained by minimizing the NN loss function

# Neural Network loss function

- Given a training data set $\mathcal{V}_{\text{train}} = \left\{ \boldsymbol{x}_0^{(n)}, \boldsymbol{x}_t^{(n)} \right\}_{n=1}^{N}$, the NN parameters, $\{\theta_{\mathbf{h}}, \theta_{\mathbf{g}}\}$, are obtained by minimizing the loss function

$$\mathcal{L} = \mathcal{L}_1(\theta_{\mathbf{h}}) + \lambda \mathcal{L}_2(\theta_{\mathbf{h}}, \theta_{\mathbf{g}})$$

- $\mathcal{L}_1(\theta_{\mathbf{h}}) = -\frac{1}{N} \log L(\theta_{\mathbf{h}})$ where $L$ is the likelihood

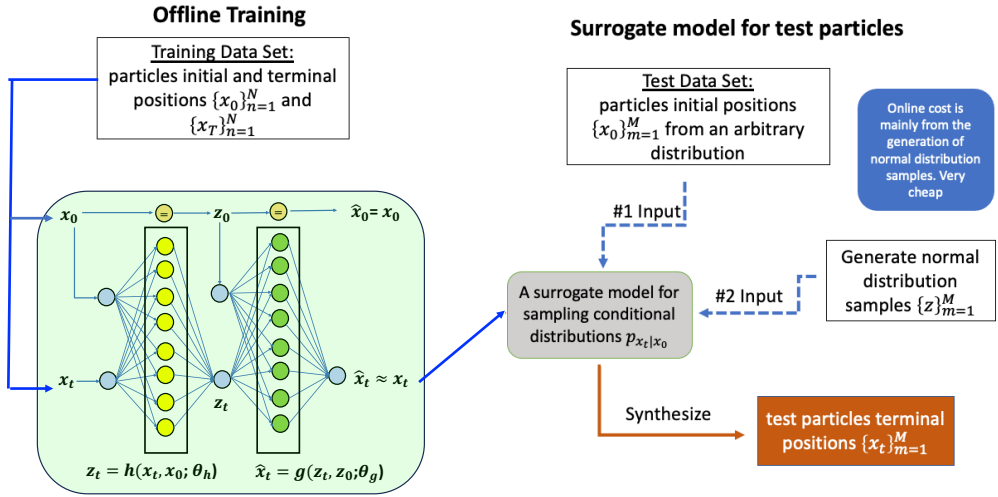
$$L(\theta_{\mathbf{h}}) = \prod_{n=1}^{N} p_X(\boldsymbol{x}_t^{(n)}) = \prod_{n=1}^{N} p_Z(\boldsymbol{h}_1(\boldsymbol{x}_t^{(n)}; \theta_h)) \left| \det \mathbf{J}_{\boldsymbol{h}}(\boldsymbol{x}_t^{(n)}; \theta_h) \right|$$

- $\mathcal{L}_2(\theta_{\mathbf{h}}, \theta_{\mathbf{g}})$ quantifies the reversibility of the NN

$$\mathcal{L}_2 = \frac{1}{N} \sum_{n=1}^{N} \left( \left\| \boldsymbol{x}^{(n)} - \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}^{(n)}; \theta_h); \theta_g) \right\| + \left| \det \mathbf{J}_{\boldsymbol{g}}(\boldsymbol{h}(\boldsymbol{x}^{(n)})) \det \mathbf{J}_{\boldsymbol{h}}(\boldsymbol{x}^{(n)}) - 1 \right| \right),$$

- Minimizing $\mathcal{L}$ maximizes the likelihood and minimizes error in $\mathbf{g} \approx \mathbf{h}^{-1}$.
- $\lambda$ controls the relative importance of likelihood and invertibility, and it is selected by minimizing the cross entropy.

# Normalizing flow surrogate model workflow



# Benchmark, convergence study

We consider the one-dimensional Fokker-Planck equation for $f = f(x,t)$ with $x \in (0, L)$

$$\frac{\partial f}{\partial t} = -\frac{\partial}{\partial x}[b\, f] + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left[\sigma^2 f\right] ,$$

with corresponding stochastic equation

$$dx = b(x)dt + \sigma(x)dW\,.$$

For $b(x) = 2\sqrt{x} + 1$ and $\sigma(x) = 2\sqrt{x}$ this problem has the exact solution

$$x_t = (\sqrt{x_0} + t + W_t)^2,$$

with transition probability

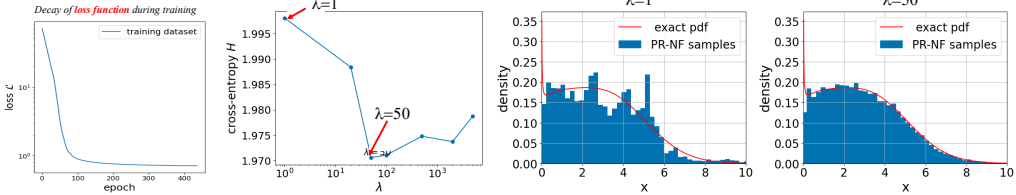
$$p(x,t|x_0,t_0) = \frac{1}{2\sqrt{2\pi t x}}\left[\exp\left(\frac{-(\sqrt{x}-\sqrt{x_0}-t)^2}{2t}\right) + \exp\left(\frac{-(\sqrt{x}+\sqrt{x_0}+t)^2}{2t}\right)\right]$$

and thus, the solution of the initial value problem $f(x, t=0) = f_0(x)$ is

$$f(x,t) = \int_0^L p(x,t|x_0,t_0) f_0(x)\, dx$$

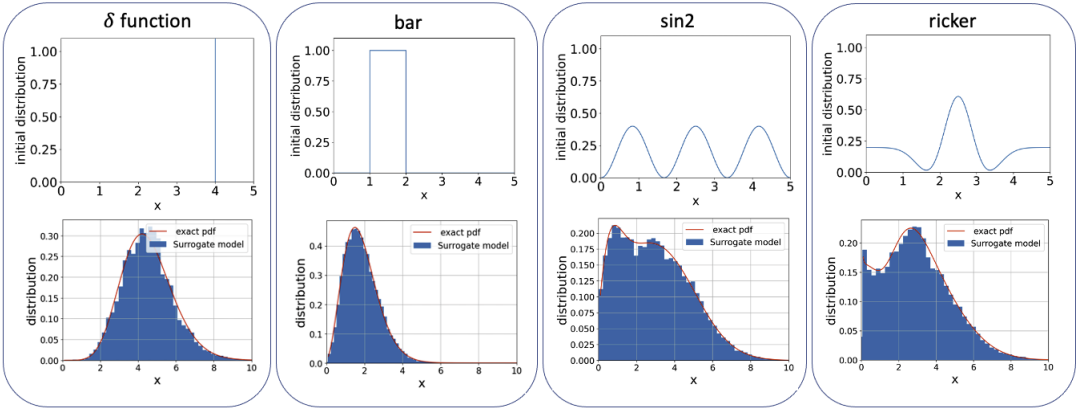
# Decay of loss function and optimal lambda

- Training: $N_{\text{train}} = 20000$ initial positions $\{X_0^{(n)}\}_{n=1}^{N_{\text{train}}}$ sampled from uniform distribution over $\mathcal{D} = [0, 5]$, and $\{X_t^{(n)}\}_{n=1}^{N_{\text{train}}}$ computed by integrating the SDE.
- Neural network has $N_{\text{layer}} = 1$ hidden layer with $N_{\text{neuron}} = 256$ neurons.
- $\lambda$ in $\mathcal{L} = \mathcal{L}_1(\theta_{\mathbf{h}}) + \lambda \mathcal{L}_2(\theta_{\mathbf{h}}, \theta_{\mathbf{g}})$ selected by minimizing the cross-entropy $H_c(p_{exact}, p_{approx}) = \int p_{exact}(x) \log p_{approx}(x)\, dx$



# Solution for different initial conditions

Once trained, the surrogate model can be use to solve the problem for different initial conditions by transforming samples from a Gaussian distribution



# Convergence of method

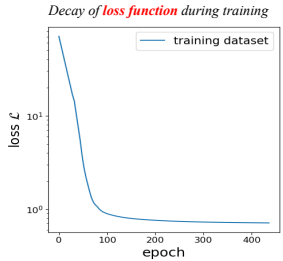
Introducing the entropy, $H(p_X)$ and cross entropy $H_c(p_X, \widetilde{p_X})$, of the exact $p_X$ and the surrogate model $\widetilde{p_X}$ distributions,

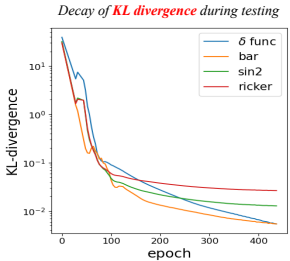
$$H(p_X) = \int_{\mathbb{R}^d} p_X(\boldsymbol{x}) \log p_X(\boldsymbol{x}) d\boldsymbol{x}\,, \qquad H_c(p_X, \widetilde{p_X}) = \int_{\mathbb{R}^d} \widetilde{p_X} \log p_X(\boldsymbol{x}) d\boldsymbol{x}$$

the KL-divergence,

$$D_{\mathrm{KL}}(p_X \,\|\, \widetilde{p_X}) = H(p_X) - H_c(p_X, \widetilde{p_X})$$

is a measure of the difference between the two PDFs



# Hot tail generation of runaway electrons during thermal quench

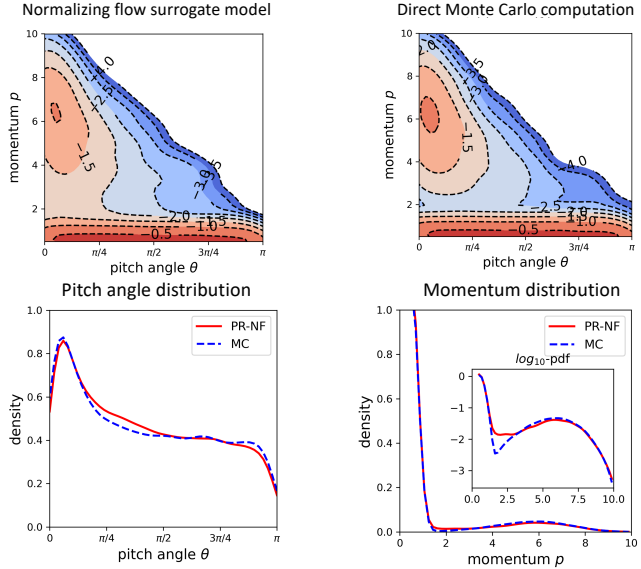
- 2D Fokker-Planck for $f$ as function of momentum, $p$, and pitch angle $\xi$

$$\frac{\partial f}{\partial t} = -E\left[\xi\frac{\partial f}{\partial p} + \frac{(1-\xi^2)}{p}\frac{\partial f}{\partial \xi}\right] + \frac{1}{p^2}\frac{\partial}{\partial p}\left\{p^2\left[C_A\frac{\partial f}{\partial p} + C_F f\right] + \frac{C_B}{p^2}\frac{\partial}{\partial \xi}\left[(1-\xi^2)\frac{\partial f}{\partial \xi}\right]\right\}$$

- Assume fast thermal quench with plasma parameters depending on final plasma temperature $T_f$.
- Electric field from Ohm's law with temperature dependent Spitzer resistivity
- Hot tail generation of runaway electrons studied by computing the evolution for an initial Maxwellian at temperature $T_0 > T_f$
- Training performed by solving stochastic equations

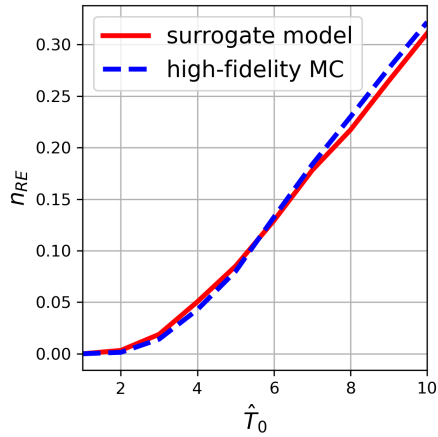
$$\begin{aligned} dp &= \left[E\xi - \frac{\gamma p}{\tau}\left(1-\xi^2\right) - C_F + \frac{1}{p^2}\frac{\partial}{\partial p}\left(p^2 C_A\right)\right]dt + \sqrt{2C_A}\,dW_p \\ d\xi &= \left[\frac{E\left(1-\xi^2\right)}{p} + \frac{\xi\left(1-\xi^2\right)}{\tau\gamma} - 2\xi\frac{C_B}{p^2}\right]dt + \frac{\sqrt{2C_B}}{p}\sqrt{1-\xi^2}\,dW_\xi \end{aligned}$$

# Surrogate model reproduces high-fidelity direct simulation



# Surrogate model also reproduces expensive high-fidelity simulations of hot-tail RE production rate as function of temperature

- Exploration of the hot-tail generation of REs requires the solution of this problem for different initial conditions.
- **Production Rate:** $n_{\mathrm{RE}} = \int_{-1}^{1} \int_{p^*}^{p_{\max}} f_{t_{\max}}(p, \xi) dp d\xi$, for different values of $T_0$, where $p^* = 1.75$.
- The total running time of the surrogate model consists of two parts, the offline cost is around $C_{\mathrm{offline}} = 1354$ sec and the online cost is around $C_{\mathrm{online}} = 12$ sec.
- The total running time of the MC method is around $C_{\mathrm{MC}} = 4000$ sec. MC method depends on $t_{\max}$, temporal step size, ...



# Transport in the presence of a 3D chaotic flow

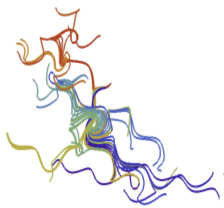
- To test the surrogate model in the case of complex 3D transport we consider

$$\frac{\partial f(t,\boldsymbol{x})}{\partial t} + \boldsymbol{v}\cdot\nabla f(t,\mathbf{x}) = D\nabla^2 f(t,\boldsymbol{x}),$$

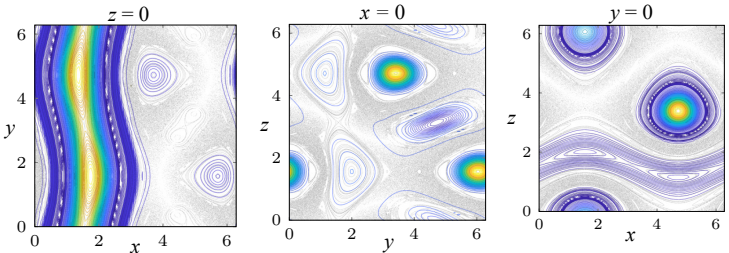
$$v_x = [A\sin z + C\cos y],\, v_y = [B\sin x + A\cos z],\, v_z = [C\sin y + B\cos x]\,.$$

- This ABC velocity field is known to exhibit very complex 3D chaotic trajectories
- Problem of interest to fluid models of RE

*Mixing in the ABC flow*



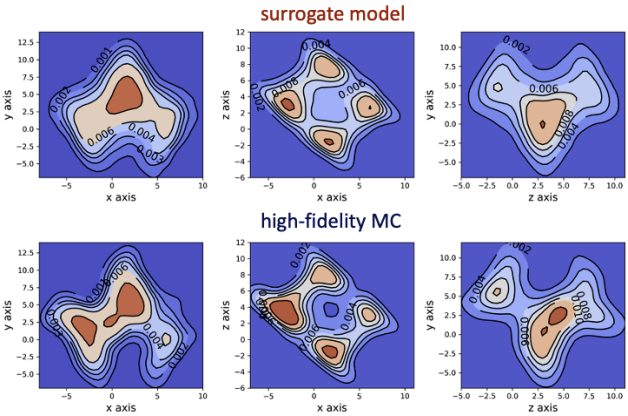
*Poincare plots of chaotic ABC flow*

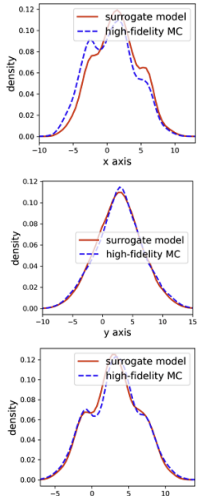



[*Adapted from* Katsanoulis et al., *J. Fluid Mech.* (2023), *vol.* 954, A28]

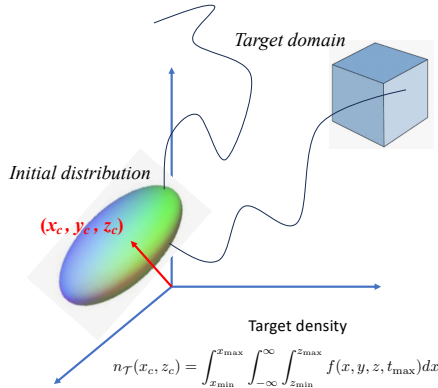
# Surrogate model reproduces high-fidelity direct simulations

$$f(x,y,z,t_0) = H(x,y,z)\exp\left[-\left(\frac{x-x_c}{\sigma_x}\right)^2 - \left(\frac{y-y_c}{\sigma_y}\right)^2 - \left(\frac{z-z_c}{\sigma_z}\right)^2\right]$$



# Surrogate model reproduces high-fidelity direct simulations

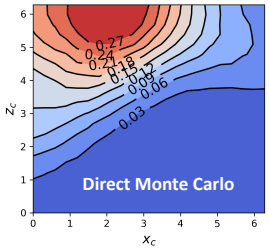


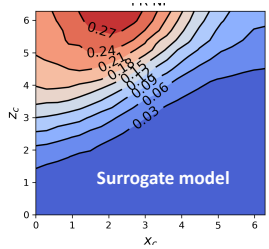

Target density

$$n_{\mathcal{T}}(x_c, z_c) = \int_{x_{\min}}^{x_{\max}} \int_{-\infty}^{\infty} \int_{z_{\min}}^{z_{\max}} f(x, y, z, t_{\max}) dx dy dz$$

Initial distribution

$$f(x, y, z, t_0) = H(x, y, z) \exp\left[ -\left(\frac{x - x_c}{\sigma_x}\right)^2 - \left(\frac{y - y_c}{\sigma_y}\right)^2 - \left(\frac{z - z_c}{\sigma_z}\right)^2 \right]$$




Running time: $C_{\mathrm{MC}} = 5320$ sec. Surrogate model $C_{\mathrm{offline}} = 2200$ sec, and $C_{\mathrm{online}} = 50$ sec.

# Summary

- We proposed an accurate and efficient surrogate method for the solution of initial value problems of the Fokker-Planck equation with arbitrary initial conditions.
- The method is based on normalizing flows, a powerful machine learning generative model.
- We presented a 1D benchmark/convergence example, and applications to hot-tail generation of runaway electrons in 2D, and transports in 3D chaotic flows.
- The surrogate model reproduces the time-consuming high-fidelity Monte-Carlo simulations.
- Further details can be found in

  M. Yang, P. Wang, D. del-Castillo-Negrete, Y. Cao and G. Zhang, "*A pseudo-reversible normalizing flow for stochastic dynamical systems with various initial distributions.*" Submitted to SIAM Journal of Scientific Computing (2023). **https://arxiv.org/pdf/2306.05580.pdf**