Modelling of highly oscillatory phenomenon by neural networks

Maxime BOUCHEREAU - Université de Rennes (IRMAR)

PhD defense - 17^{th} October 2024

Ph. D supervisors:
François CASTELLA - Philippe CHARTIER
Mohammed LEMOU - Florian MEHATS

Motivation

- **Goal:** Approximate solutions of highly oscillatory ODE's, of the form:

$$\begin{cases} \dot{y^\varepsilon}(t) & = & f\left(\frac{t}{\varepsilon}, y^\varepsilon(t)\right) \\ y^\varepsilon(0) & = & y_0 \in \mathbb{R}^d \end{cases} \tag{1}$$

where $\tau \mapsto f(\tau, \cdot)$ is 2π-periodic, ε is a small parameter.

- **Issue with standard methods:** Error bound increasing when $\varepsilon \to 0$, e.g. with Forward Euler method: $y_{n+1} = y_n + hf\left(\frac{nh}{\varepsilon}, y_n\right)$:

$$\underset{0 \leqslant n \leqslant N}{Max} |y_n - y^\varepsilon(nh)| \quad \leqslant \quad \frac{Ch}{\varepsilon} \tag{2}$$

- **Efficient numerical methods exist:** But require complex preliminary computations.

Motivation

- **Goal:** Approximate solutions of highly oscillatory ODE's, of the form:

$$\left\{ \begin{array}{rcl} \dot{y}^\varepsilon(t) & = & f\left(\frac{t}{\varepsilon}, y^\varepsilon(t)\right) \\ y^\varepsilon(0) & = & y_0 \in \mathbb{R}^d \end{array} \right. \tag{1}$$

  where $\tau \mapsto f(\tau, \cdot)$ is 2π-periodic, ε is a small parameter.

- **Issue with standard methods:** Error bound increasing when $\varepsilon \to 0$, e.g. with Forward Euler method: $y_{n+1} = y_n + hf\left(\frac{nh}{\varepsilon}, y_n\right)$:

$$\underset{0 \leqslant n \leqslant N}{Max} |y_n - y^\varepsilon(nh)| \leqslant \frac{Ch}{\varepsilon} \tag{2}$$

- **Efficient numerical methods exist:** But require complex preliminary computations.

Motivation

- **Goal:** Approximate solutions of highly oscillatory ODE's, of the form:

$$\left\{ \begin{array}{rcl} \dot{y^\varepsilon}(t) & = & f\left(\frac{t}{\varepsilon}, y^\varepsilon(t)\right) \\ y^\varepsilon(0) & = & y_0 \in \mathbb{R}^d \end{array} \right. \tag{1}$$

where $\tau \mapsto f(\tau, \cdot)$ is 2π-periodic, ε is a small parameter.

- **Issue with standard methods:** Error bound increasing when $\varepsilon \to 0$, e.g. with Forward Euler method: $y_{n+1} = y_n + hf\left(\frac{nh}{\varepsilon}, y_n\right)$:

$$\underset{0 \leqslant n \leqslant N}{Max} |y_n - y^\varepsilon(nh)| \leqslant \frac{Ch}{\varepsilon} \tag{2}$$

- **Efficient numerical methods exist:** But require complex preliminary computations.

Motivation

- **Goal:** Approximate solutions of highly oscillatory ODE's, of the form:

$$\begin{cases} \dot{y}^\varepsilon(t) &= f\left(\frac{t}{\varepsilon}, y^\varepsilon(t)\right) \\ y^\varepsilon(0) &= y_0 \in \mathbb{R}^d \end{cases} \tag{1}$$

  where $\tau \mapsto f(\tau, \cdot)$ is 2π-periodic, ε is a small parameter.
- **Issue with standard methods:** Error bound increasing when $\varepsilon \to 0$, e.g. with Forward Euler method: $y_{n+1} = y_n + hf\left(\frac{nh}{\varepsilon}, y_n\right)$:

$$\underset{0 \leqslant n \leqslant N}{Max} |y_n - y^\varepsilon(nh)| \leqslant \frac{Ch}{\varepsilon} \tag{2}$$

- **Efficient numerical methods exist:** But require complex preliminary computations.

Motivation: How to avoid these computations ?

Main tools used:

- Function approximations by neural networks
- Backward error analysis for autonomous ODE's
- Averaging theory & Numerical methods for highly oscillatory ODE's

Autonomous case: f is independant from τ: $\dot{y}(t) = f(y(t))$

Definition (Modified field w.r.t. a numerical method)

Let consider a one step numerical method $\Phi_h(\cdot)$. The ***modified vector field w.r.t.*** Φ_h*, denoted $\tilde{f}_h$, is defined by the relation:*

$$\varphi^f_{nh}(y_0) = \left(\Phi_h^{\tilde{f}_h}\right)^n(y_0) \qquad (3)$$

Proposition (Properties of the modified field)

- ***Formal serie w.r.t.*** h*: If Φ is of order p, then* $\tilde{f}_h(y) = f(y) + h^p f_1(y) + h^{p+1} f_2(y) + \cdots$
- ***Truncated modified field:*** *Let consider $\tilde{f}_h^{[q]}$ the truncated modified field $\tilde{f}_h^{[q]}(y) = f(y) + h^p f_1(y) + \cdots + h^{p+q-2} f_{q-1}(y)$. Thus we have*

$$\underset{0 \leqslant n \leqslant N}{Max} \left| \left(\Phi_h^{\tilde{f}_h^{[q]}}\right)^n (y_0) - \varphi^f_{nh}(y_0) \right| \leqslant C h^{p+q-1} \qquad (4)$$

Autonomous case: f is independant from τ: $\dot{y}(t) = f(y(t))$

Definition (Modified field w.r.t. a numerical method)

Let consider a one step numerical method $\Phi_h(\cdot)$. The ***modified vector field w.r.t.*** Φ_h*, denoted $\tilde{f}_h$, is defined by the relation:*

$$\varphi^f_{nh}(y_0) \quad = \quad \left(\Phi_h^{\tilde{f}_h}\right)^n(y_0) \tag{3}$$

Proposition (Properties of the modified field)

- ***Formal serie w.r.t.*** h*: If Φ is of order p, then* $\tilde{f}_h(y) = f(y) + h^p f_1(y) + h^{p+1} f_2(y) + \cdots$
- ***Truncated modified field:*** *Let consider $\tilde{f}_h^{[q]}$ the truncated modified field $\tilde{f}_h^{[q]}(y) = f(y) + h^p f_1(y) + \cdots + h^{p+q-2} f_{q-1}(y)$. Thus we have*

$$\underset{0\leqslant n\leqslant N}{Max}\left|\left(\Phi_h^{\tilde{f}_h^{[q]}}\right)^n(y_0) - \varphi^f_{nh}(y_0)\right| \quad \leqslant \quad Ch^{p+q-1} \tag{4}$$

Autonomous case: f is independant from τ: $\dot{y}(t) = f(y(t))$

Definition (Modified field w.r.t. a numerical method)

*Let consider a one step numerical method $\Phi_h(\cdot)$. The **modified vector field w.r.t.** Φ_h, denoted $\tilde{f}_h$, is defined by the relation:*

$$\varphi^f_{nh}(y_0) \quad = \quad \left(\Phi_h^{\tilde{f}_h}\right)^n (y_0) \tag{3}$$

Proposition (Properties of the modified field)

- ***Formal serie w.r.t.*** *h: If Φ is of order p, then* $\tilde{f}_h(y) = f(y) + h^p f_1(y) + h^{p+1} f_2(y) + \cdots$
- ***Truncated modified field:*** *Let consider $\tilde{f}_h^{[q]}$ the truncated modified field $\tilde{f}_h^{[q]}(y) = f(y) + h^p f_1(y) + \cdots + h^{p+q-2} f_{q-1}(y)$. Thus we have*

$$\underset{0 \leqslant n \leqslant N}{Max} \left| \left(\Phi_h^{\tilde{f}_h^{[q]}}\right)^n (y_0) - \varphi^f_{nh}(y_0) \right| \quad \leqslant \quad C h^{p+q-1} \tag{4}$$

Autonomous case: example of modified field

- **Linear equation:**

$$\dot{y}(t) = ay(t) \tag{5}$$

- **Modified field - Forward Euler:**

$$\tilde{f}_h^{[0]}(y) = ay \tag{6}$$

$$\tilde{f}_h^{[q]}(y) = \left(a + \frac{h}{2}a^2 + \cdots + \frac{h^{q-1}}{q!}a^q\right)y \tag{7}$$

- **Error estimate:**

$$\underset{0\leqslant n\leqslant N}{Max}\left|e^{nha}y(0) - \left(\Phi_h^{\tilde{f}_h^{[q]}}\right)^n(y(0))\right| \leqslant Ch^q \tag{8}$$

Autonomous case: example of modified field

- **Linear equation:**

$$\dot{y}(t) = ay(t) \tag{5}$$

- **Modified field - Forward Euler:**

$$\tilde{f}_h^{[0]}(y) = ay \tag{6}$$

$$\tilde{f}_h^{[q]}(y) = \left(a + \frac{h}{2}a^2 + \cdots + \frac{h^{q-1}}{q!}a^q\right) y \tag{7}$$

- **Error estimate:**

$$\underset{0\leqslant n\leqslant N}{Max} \left| e^{nha}y(0) - \left(\Phi_h^{\tilde{f}_h^{[q]}}\right)^n (y(0)) \right| \leqslant Ch^q \tag{8}$$

Autonomous case: example of modified field

- **Linear equation:**

$$\dot{y}(t) = ay(t) \tag{5}$$

- **Modified field - Forward Euler:**

$$\tilde{f}_h^{[0]}(y) = ay \tag{6}$$

$$\tilde{f}_h^{[q]}(y) = \left(a + \frac{h}{2}a^2 + \cdots + \frac{h^{q-1}}{q!}a^q\right) y \tag{7}$$

- **Error estimate:**

$$\underset{0\leqslant n\leqslant N}{Max} \left| e^{nha} y(0) - \left(\Phi_h^{\tilde{f}_h^{[q]}}\right)^n (y(0)) \right| \leqslant Ch^q \tag{8}$$

Autonomous case: example of modified field

- **Linear equation:**

$$\dot{y}(t) = ay(t) \tag{5}$$

- **Modified field - Forward Euler:**

$$\tilde{f}_h^{[0]}(y) = ay \tag{6}$$

$$\tilde{f}_h^{[q]}(y) = \left(a + \frac{h}{2}a^2 + \cdots + \frac{h^{q-1}}{q!}a^q\right) y \tag{7}$$

- **Error estimate:**

$$\underset{0 \leqslant n \leqslant N}{Max} \left| e^{nha} y(0) - \left(\Phi_h^{\tilde{f}_h^{[q]}}\right)^n (y(0)) \right| \leqslant Ch^q \tag{8}$$

Learning hidden dynamics & structure preservation

- **Learning dynamics:** M. Raissi, P. Perdikaris, G. Em Karniadakis, 2018.
- **Learning dynamics with Backward error analysis:** Y. Zhu, P. Jin, B. Zhu, and Y. Tang, 2020
- **Hamiltonian structure (HNN):** M. David and F. Méhats, 2023.
- **Poisson structure (PNN):** P. Jin, Z. Zhang, I. G Kevrekidis, and G. Em Karniadakis, 2022.
- **Free-divergence structure (VP-Nets):** A. Zhu, B. Zhu, J. Zhang, Y. Tang, and J. Liu, 2022

Learning hidden dynamics & structure preservation

- **Learning dynamics:** M. Raissi, P. Perdikaris, G. Em Karniadakis, 2018.
- **Learning dynamics with Backward error analysis:** Y. Zhu, P. Jin, B. Zhu, and Y. Tang, 2020
- **Hamiltonian structure (HNN):** M. David and F. Méhats, 2023.
- **Poisson structure (PNN):** P. Jin, Z. Zhang, I. G Kevrekidis, and G. Em Karniadakis, 2022.
- **Free-divergence structure (VP-Nets):** A. Zhu, B. Zhu, J. Zhang, Y. Tang, and J. Liu, 2022

Learning hidden dynamics & structure preservation

- **Learning dynamics:** M. Raissi, P. Perdikaris, G. Em Karniadakis, 2018.
- **Learning dynamics with Backward error analysis:** Y. Zhu, P. Jin, B. Zhu, and Y. Tang, 2020
- **Hamiltonian structure (HNN):** M. David and F. Méhats, 2023.
- **Poisson structure (PNN):** P. Jin, Z. Zhang, I. G Kevrekidis, and G. Em Karniadakis, 2022.
- **Free-divergence structure (VP-Nets):** A. Zhu, B. Zhu, J. Zhang, Y. Tang, and J. Liu, 2022

Highly oscillatory ODE's: Structure of solutions of $\dot{y^\varepsilon}(t) = f\left(\frac{t}{\varepsilon}, y^\varepsilon(t)\right)$

Theorem (P. Chartier, A. Murua, and J.M. Sanz-Serna - 2015)

There exists ε_0 s.t. for all $\varepsilon \in (0, \varepsilon_0]$, there exists $n_\varepsilon \in \mathbb{N}$ s.t. For all $t \in [0, T]$:

$$\left| y^\varepsilon(t) - \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t}{\varepsilon}}\left(\varphi_t^{F^{\varepsilon,[n_\varepsilon]}}(y_0)\right)\right| \leqslant M e^{-\frac{\beta\varepsilon_0}{\varepsilon}} \tag{9}$$

where:

- *$F^{\varepsilon,[n_\varepsilon]}$ is the truncation at order $\mathcal{O}(\varepsilon^{n_\varepsilon})$ of **averaged field** F^ε. Structure: $F^\varepsilon(y) = \langle f\rangle(y) + \varepsilon F_1(y) + \varepsilon^2 F_2(y) + \cdots$, where $\langle f\rangle$ is the average field w.r.t. time variable:*

$$\langle f\rangle(y) := \frac{1}{2\pi}\int_0^{2\pi} f(\tau, y)\mathrm{d}\tau \tag{10}$$

- *$\phi_\tau^{\varepsilon,[n_\varepsilon]}$ is the truncation at order $\mathcal{O}(\varepsilon^{n_\varepsilon})$ of **high oscillation generator** $\phi_\tau^{\varepsilon,[n_\varepsilon]}(y) = y + \varepsilon\tilde{\phi}_1(\tau, y) + \varepsilon^2\tilde{\phi}_2(\tau, y) + \cdots$ (Near to identity mapping) and is 2π-periodic w.r.t. τ.*

Highly oscillatory ODE's: Structure of solutions of $\dot{y}^\varepsilon(t) = f\left(\frac{t}{\varepsilon}, y^\varepsilon(t)\right)$

Theorem (P. Chartier, A. Murua, and J.M. Sanz-Serna - 2015)

There exists ε_0 s.t. for all $\varepsilon \in (0, \varepsilon_0]$, there exists $n_\varepsilon \in \mathbb{N}$ s.t. For all $t \in [0, T]$:

$$\left| y^\varepsilon(t) - \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t}{\varepsilon}}\left(\varphi_t^{F^{\varepsilon,[n_\varepsilon]}}(y_0)\right)\right| \leqslant Me^{-\frac{\beta\varepsilon_0}{\varepsilon}} \tag{9}$$

where:

- $F^{\varepsilon,[n_\varepsilon]}$ *is the truncation at order $\mathcal{O}(\varepsilon^{n_\varepsilon})$ of* ***averaged field*** F^ε*. Structure: $F^\varepsilon(y) = \langle f\rangle(y) + \varepsilon F_1(y) + \varepsilon^2 F_2(y) + \cdots$, where $\langle f\rangle$ is the average field w.r.t. time variable:*

$$\langle f\rangle(y) := \frac{1}{2\pi}\int_0^{2\pi} f(\tau, y)\mathrm{d}\tau \tag{10}$$

- $\phi^{\varepsilon,[n_\varepsilon]}_\tau$ *is the truncation at order $\mathcal{O}(\varepsilon^{n_\varepsilon})$ of* ***high oscillation generator*** $\phi^{\varepsilon,[n_\varepsilon]}_\tau(y) = y + \varepsilon\tilde{\phi}_1(\tau, y) + \varepsilon^2\tilde{\phi}_2(\tau, y) + \cdots$ *(Near to identity mapping) and is 2π-periodic w.r.t. τ.*

Highly oscillatory ODE's: UA method

- **Micro-Macro decomposition:**

$$y^\varepsilon(t) = \phi^{[p]}_{\frac{t}{\varepsilon}}(v(t)) + w(t) \tag{11}$$

- **Micro-Macro decomposition system:**

$$\begin{cases} \dot{v}(t) = F^{[p]}(v(t)) \\ \dot{w}(t) = f\left(\frac{t}{\varepsilon}, \phi^{[p]}_{\frac{t}{\varepsilon}}(v(t)) + w(t)\right) \\ \quad - \left(\frac{1}{\varepsilon}\partial_\tau \phi^{[p]}_{\frac{t}{\varepsilon}} + \partial_y \phi^{[p]}_{\frac{t}{\varepsilon}} F^{[p]}\right)(v(t)) \\ (v,w)(0) = (y^\varepsilon(0), 0) \end{cases} \tag{12}$$

- **Numerical integration:** Integrate this system with a standard numerical method of order p giving approximations $(v_n, w_n)_{0\leqslant n\leqslant N}$ of $(v(nh), w(nh))_{0\leqslant n\leqslant N}$. Approximation of $y^\varepsilon(nh)$ given by

$$y^\varepsilon_n := \phi^{[p]}_{\frac{nh}{\varepsilon}}(v_n) + w_n \tag{13}$$

Highly oscillatory ODE's: UA method

- **Micro-Macro decomposition:**

$$y^\varepsilon(t) = \phi^{[p]}_{\frac{t}{\varepsilon}}(v(t)) + w(t) \tag{11}$$

- **Micro-Macro decomposition system:**

$$\begin{cases} \dot{v}(t) = F^{[p]}(v(t)) \\ \dot{w}(t) = f\left(\frac{t}{\varepsilon}, \phi^{[p]}_{\frac{t}{\varepsilon}}(v(t)) + w(t)\right) \\ \qquad - \left(\frac{1}{\varepsilon}\partial_\tau \phi^{[p]}_{\frac{t}{\varepsilon}} + \partial_y \phi^{[p]}_{\frac{t}{\varepsilon}} F^{[p]}\right)(v(t)) \\ (v,w)(0) = (y^\varepsilon(0), 0) \end{cases} \tag{12}$$

- **Numerical integration:** Integrate this system with a standard numerical method of order p giving approximations $(v_n, w_n)_{0\leqslant n\leqslant N}$ of $(v(nh), w(nh))_{0\leqslant n\leqslant N}$. Approximation of $y^\varepsilon(nh)$ given by

$$y_n^\varepsilon := \phi^{[p]}_{\frac{nh}{\varepsilon}}(v_n) + w_n \tag{13}$$

Highly oscillatory ODE's: UA method

- **Micro-Macro decomposition:**

$$y^\varepsilon(t) \quad = \quad \phi^{[p]}_{\frac{t}{\varepsilon}}(v(t)) + w(t) \tag{11}$$

- **Micro-Macro decomposition system:**

$$\left\{\begin{array}{ccc} \dot{v}(t) & = & F^{[p]}(v(t)) \\ \dot{w}(t) & = & f\left(\frac{t}{\varepsilon}, \phi^{[p]}_{\frac{t}{\varepsilon}}(v(t)) + w(t)\right) \\ & - & \left(\frac{1}{\varepsilon}\partial_\tau \phi^{[p]}_{\frac{t}{\varepsilon}} + \partial_y \phi^{[p]}_{\frac{t}{\varepsilon}} F^{[p]}\right)(v(t)) \\ (v,w)(0) & = & (y^\varepsilon(0), 0) \end{array}\right. \tag{12}$$

- **Numerical integration:** Integrate this system with a standard numerical method of order p giving approximations $(v_n, w_n)_{0\leqslant n\leqslant N}$ of $(v(nh), w(nh))_{0\leqslant n\leqslant N}$. Approximation of $y^\varepsilon(nh)$ given by

$$y^\varepsilon_n \quad := \quad \phi^{[p]}_{\frac{nh}{\varepsilon}}(v_n) + w_n \tag{13}$$

Highly oscillatory ODE's: UA method

- **Micro-Macro decomposition:**

$$y^\varepsilon(t) = \phi^{[p]}_{\frac{t}{\varepsilon}}(v(t)) + w(t) \tag{11}$$

- **Micro-Macro decomposition system:**

$$\left\{\begin{array}{rcl} \dot{v}(t) & = & F^{[p]}(v(t)) \\ \dot{w}(t) & = & f\left(\frac{t}{\varepsilon}, \phi^{[p]}_{\frac{t}{\varepsilon}}(v(t)) + w(t)\right) \\ & - & \left(\frac{1}{\varepsilon}\partial_\tau \phi^{[p]}_{\frac{t}{\varepsilon}} + \partial_y \phi^{[p]}_{\frac{t}{\varepsilon}} F^{[p]}\right)(v(t)) \\ (v,w)(0) & = & (y^\varepsilon(0), 0) \end{array}\right. \tag{12}$$

- **Numerical integration:** Integrate this system with a standard numerical method of order p giving approximations $(v_n, w_n)_{0 \leqslant n \leqslant N}$ of $(v(nh), w(nh))_{0 \leqslant n \leqslant N}$. Approximation of $y^\varepsilon(nh)$ given by

$$y^\varepsilon_n := \phi^{[p]}_{\frac{nh}{\varepsilon}}(v_n) + w_n \tag{13}$$

Highly oscillatory ODE's: UA method

Theorem (P. Chartier, M. Lemou, F. Méhats, G. Vilmart - 2020)

$$\underset{0 \leqslant n \leqslant N}{Max} |y_n^\varepsilon - y^\varepsilon(nh)| \quad \leqslant \quad Ch^p \tag{14}$$

where $h = \frac{T}{N}$ *and the constant* C *is independant from* ε.

- **Autonomous ODE:**

$$\begin{cases} \dot{y}(t) & = & f\left(y(t)\right) \\ y(0) & = & y_0 \end{cases} \tag{15}$$

- **Numerical method:** $\Phi_h(\cdot)$, assumed to be of order p.
- **Goal:** Approximate a truncated modified field $\tilde{f}_h$ by a neural network $f_\theta(\cdot, h)$ in order to get an approximated solution $y_{\theta,n} = \left(\Phi_h^{f_\theta(\cdot,h)}\right)^n (y_0)$ very close to the exact solution $y(nh)$.

Neural network structure

- **Structure of f_θ (approximation of $\tilde{f}_h^{[q]}$):**

$$\begin{aligned} f_\theta(y,h) &= f(y) + h^p f_{\theta,1}(y) + h^{p+1} f_{\theta,2}(y) \\ &+ \cdots \\ &+ h^{q+p-2} f_{\theta,q-1}(y) + h^{q+p-1} R_\theta(y,h) \end{aligned} \tag{16}$$

Data creation

- **Repeat for** $0 \leqslant k \leqslant K-1$**.**
- **Select time step** $h^{(k)} \in [h_-, h_+]$ randomly.
- **Select initial condition** $y_0^{(k)} \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi_{h^{(k)}}^f (y_0^{(k)})$ with accurate but expensive integrator.

Data creation

- **Repeat for** $0 \leqslant k \leqslant K-1$**.**
- **Select time step** $h^{(k)} \in [h_-, h_+]$ randomly.
- **Select initial condition** $y_0^{(k)} \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi_{h^{(k)}}^f(y_0^{(k)})$ with accurate but expensive integrator.

Data creation

- **Repeat for** $0 \leqslant k \leqslant K-1$**.**
- **Select time step** $h^{(k)} \in [h_-, h_+]$ randomly.
- **Select initial condition** $y_0^{(k)} \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi_{h^{(k)}}^f(y_0^{(k)})$ with accurate but expensive integrator.

Data creation

- **Repeat for** $0 \leqslant k \leqslant K-1$**.**
- **Select time step** $h^{(k)} \in [h_-, h_+]$ randomly.
- **Select initial condition** $y_0^{(k)} \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi^f_{h^{(k)}}(y_0^{(k)})$ with accurate but expensive integrator.

Data creation

- **Repeat for** $0 \leqslant k \leqslant K-1$**.**
- **Select time step** $h^{(k)} \in [h_-, h_+]$ randomly.
- **Select initial condition** $y_0^{(k)} \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi_{h^{(k)}}^f(y_0^{(k)})$ with accurate but expensive integrator.

Training

- **Training of the neural network:** Optimization of:

$$Loss_{Train} = \frac{1}{K_0} \sum_{k=0}^{K_0-1} \frac{1}{{h^{(k)}}^{2p+2}} \Big| \underbrace{\Phi_{h^{(k)}}^{f_\theta(\cdot,h^{(k)})}\left(y_0^{(k)}\right)}_{=\hat{y_1}^{(k)}} - y_1^{(k)} \Big|^2$$

- **Good training:** $Loss_{Train}$ has the same decay pattern as:

$$Loss_{Test} = \frac{1}{K-K_0} \sum_{k=K_0}^{K-1} \frac{1}{{h^{(k)}}^{2p+2}} \Big| \Phi_{h^{(k)}}^{f_\theta(\cdot,h^{(k)})}\left(y_0^{(k)}\right) - y_1^{(k)} \Big|^2$$

- **At the end of the training:** Good approximation f_θ of $\tilde{f}_h^{[q]}$ (only one training required).

Training

- **Training of the neural network:** Optimization of:

$$Loss_{Train} = \frac{1}{K_0} \sum_{k=0}^{K_0-1} \frac{1}{{h^{(k)}}^{2p+2}} \left| \underbrace{\Phi_{h^{(k)}}^{f_\theta(\cdot, h^{(k)})}\left(y_0^{(k)}\right)}_{=\hat{y_1}^{(k)}} - y_1^{(k)} \right|^2$$

- **Good training:** $Loss_{Train}$ has the same decay pattern as:

$$Loss_{Test} = \frac{1}{K-K_0} \sum_{k=K_0}^{K-1} \frac{1}{{h^{(k)}}^{2p+2}} \left| \Phi_{h^{(k)}}^{f_\theta(\cdot, h^{(k)})}\left(y_0^{(k)}\right) - y_1^{(k)} \right|^2$$

- **At the end of the training:** Good approximation f_θ of $\tilde{f}_h^{[q]}$ (only one training required).

Training

- **Training of the neural network:** Optimization of:

$$Loss_{Train} = \frac{1}{K_0}\sum_{k=0}^{K_0-1}\frac{1}{{h^{(k)}}^{2p+2}}\left|\underbrace{\Phi_{h^{(k)}}^{f_\theta(\cdot,h^{(k)})}\left(y_0^{(k)}\right)}_{=\hat{y_1}^{(k)}}-y_1^{(k)}\right|^2$$

- **Good training:** $Loss_{Train}$ has the same decay pattern as:

$$Loss_{Test} = \frac{1}{K-K_0}\sum_{k=K_0}^{K-1}\frac{1}{{h^{(k)}}^{2p+2}}\left|\Phi_{h^{(k)}}^{f_\theta(\cdot,h^{(k)})}\left(y_0^{(k)}\right)-y_1^{(k)}\right|^2$$

- **At the end of the training:** Good approximation f_θ of $\tilde{f}_h^{[q]}$ (only one training required).

Numerical integration

- **Numerical integration:** We get a small numerical error:

$$e_{\theta,n} = \left(\Phi_h^{f_\theta(\cdot,h)}\right)^n (y_0) - \varphi_{nh}^f(y_0) \tag{17}$$

Denoting the **learning error** by

$$\delta := \underset{(y,h)\in\Omega\times[0,h_+]}{\mathrm{Max}} \frac{\left|\tilde{f}_h^{[q]}(y) - f_\theta(y,h)\right|}{h^p} \tag{18}$$

Theorem (M.B, P.Chartier, M.Lemou, F.Méhats - 2023[1])

Assuming that

- *For any pair smooth vector fields f_1 and f_2, we have*

$$\forall 0 \leq h \leq h_+, \quad \left|\left|\Phi_h^{f_1} - \Phi_h^{f_2}\right|\right|_{L^\infty(\Omega)} \leqslant Ch \left|\left|f_1 - f_2\right|\right|_{L^\infty(\Omega)} \quad (19)$$

  for some positive constant C, independent of f_1 and f_2;

- *For any smooth vector field f, there exists a constant $L > 0$ such that $\forall 0 \leq h \leq h_+, \forall (y_1, y_2) \in \Omega^2$:*

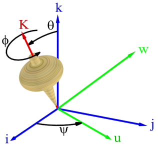
$$\left|\Phi_h^f(y_1) - \Phi_h^f(y_2)\right| \leqslant (1 + Lh)\left|y_1 - y_2\right|. \quad (20)$$

Then there exists positive constant C_θ such that:

$$\underset{0 \leqslant n \leqslant N}{Max} \left|e_{\theta,n}\right| \leqslant C_\theta \delta h^p \quad (21)$$

[1] B, M., Chartier, P., Lemou, M., & Méhats, F. (2023). *Machine Learning Methods for Autonomous Ordinary Differential Equations*

Numerical test: Rigid Body system - Forward Euler



source: Wikipedia.org

$$\begin{cases} \dot{y}_1 &= \left(\frac{1}{I_3} - \frac{1}{I_2}\right) y_2 y_3 \\ \dot{y}_2 &= \left(\frac{1}{I_1} - \frac{1}{I_3}\right) y_1 y_3 \\ \dot{y}_3 &= \left(\frac{1}{I_2} - \frac{1}{I_1}\right) y_1 y_2 \end{cases} \quad (22)$$

with $(I_1, I_2, I_3) = (1, 2, 3)$.

Parameters	
# Math Parameters:	
Interval where time steps are selected:	$[h_-, h_+] = [0.5, 2.5]$
Time for ODE simulation:	$T = 20$
Time step for ODE simulation:	$h = 0.5$
# AI Parameters:	
Domain where data are selected:	$\Omega = \left\{x \in [-2, 2]^2 : 0.98 \leqslant \lvert x \rvert \leqslant 1.02\right\}$
Number of data:	$K = 100\,000\,000$
Proportion of data for training:	80% - $K_0 = 80\,000\,000$
Number of terms in the perturbation (MLP's):	$q = 1$
Hidden layers per MLP:	2
Neurons on each hidden layer:	250
Epochs:	200

Computational time for training: 1 Day 21 h 59 min 51 s

Numerical test: Rigid Body system - Forward Euler

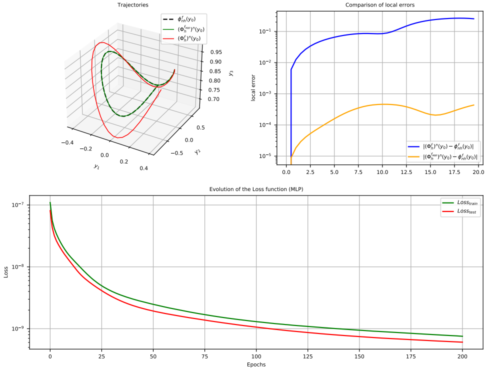


Figure: Comparison between *Loss* decays (green: $Loss_{Train}$, red: $Loss_{Test}$), trajectories (dashed dark: exact flow, red: numerical flow with f, green: numerical flow with $f_\theta(\cdot, h)$) and local error (blue: exact flow and numerical flow with f, yellow: exact and numerical flow with $f_\theta(\cdot, h)$)

Numerical test: Rigid Body system - Forward Euler

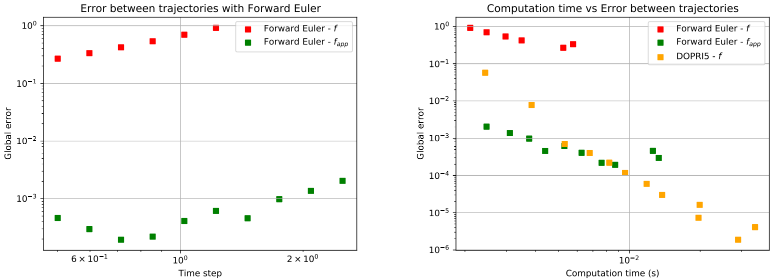
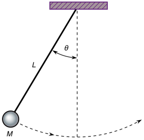


Figure: Left: Integration errors (green: integration with f, red: integration with $f_\theta(\cdot, h)$). Right: Comparison between computational time and integration error (red: numerical method with f, green: integration with $f_\theta(\cdot, h)$, yellow: integration with DOPRI5).

Numerical test: Nonlinear Pendulum - Midpoint



$$\begin{cases} \dot{y_1} = -\sin(y_2) \\ \dot{y_2} = y_1 \end{cases} \tag{23}$$

Hamiltonian function:

$$H : y \mapsto \frac{1}{2}y_1^2 + (1 - \cos(y_2)) \tag{24}$$

Numerical test: Nonlinear Pendulum - Midpoint

Parameters	
# Math Parameters:	
Interval where time steps are selected:	$[h_-, h_+] = [0.05, 0.5]$
Time for ODE simulation:	$T = 20$
Time step for ODE simulation:	$h = 0.25$
# AI Parameters:	
Domain where data are selected:	$\Omega = [-2, 2]^2$
Number of data:	$K = 20\,000\,000$
Proportion of data for training:	80% - $K_0 = 16\,000\,000$
Number of terms in the perturbation (MLP's):	$q = 1$
Hidden layers per MLP:	2
Neurons on each hidden layer:	200
Epochs:	200

Computational time for data creation: 2 Days 21 h 49 min 50 s
Computational time for training: 9 h 47 min 51 s

Numerical test: Nonlinear Pendulum - Midpoint

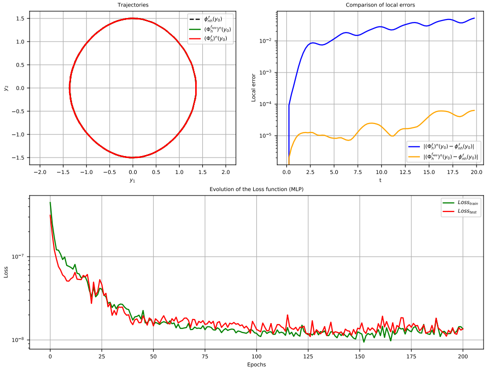


Figure: Comparison between *Loss* decays (green: $Loss_{Train}$, red: $Loss_{Test}$), trajectories (dashed dark: exact flow, red: numerical flow with f, green: numerical flow with $f_\theta(\cdot, h)$) and local error (blue: exact flow and numerical flow with f, yellow: exact and numerical flow with $f_\theta(\cdot, h)$)

Numerical test: Nonlinear Pendulum - Midpoint

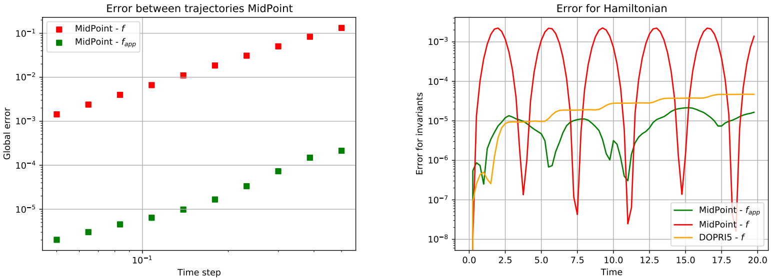


Figure: Left: Integration errors (green: integration with f, red: integration with $f_\theta(\cdot, h)$). Right: Evolution of the error between Hamiltonian $H : y \mapsto (1 - \cos(y_2)) + \frac{1}{2}y_1^2$ over the numerical flow and Hamiltonian at $t = 0$, $H(y_0)$.

- **Goal:** Find an approximation of the solution of

$$\begin{cases} \dot{y}^{\varepsilon}(t) & = & f\left(\frac{t}{\varepsilon}, y^{\varepsilon}(t)\right) \\ y^{\varepsilon}(0) & = & y_0 \end{cases} \tag{25}$$

- **General strategy:** Use the decomposition (9)

$$y^{\varepsilon}(t) \approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t}{\varepsilon}}\left(\varphi_t^{F^{\varepsilon,[n_\varepsilon]}}(y_0)\right) \tag{26}$$

in order to approximate $F^{\varepsilon,[n_\varepsilon]}$ and $\phi_{\cdot}^{\varepsilon,[n_\varepsilon]}(\cdot)$ with neural networks.

- **Goal:** Find an approximation of the solution of

$$\begin{cases} \dot{y}^\varepsilon(t) & = & f\left(\frac{t}{\varepsilon}, y^\varepsilon(t)\right) \\ y^\varepsilon(0) & = & y_0 \end{cases} \tag{25}$$

- **General strategy:** Use the decomposition (9)

$$y^\varepsilon(t) \quad \approx \quad \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t}{\varepsilon}}\left(\varphi_t^{F^{\varepsilon,[n_\varepsilon]}}(y_0)\right) \tag{26}$$

in order to approximate $F^{\varepsilon,[n_\varepsilon]}$ and $\phi^{\varepsilon,[n_\varepsilon]}_{\cdot}(\cdot)$ with neural networks.

Main idea

$$
\begin{aligned}
y^{\varepsilon}(t_0+h) &\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0+h}(y^{\varepsilon}(0))\right) \\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}(y^{\varepsilon}(0))\right)\right) \\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}(y^{\varepsilon}(0))\right)\right) \\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)(y^{\varepsilon}(t_0)) \qquad (27) \\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\Phi^{\widetilde{F^{\varepsilon,[n_\varepsilon]}_h}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)(y^{\varepsilon}(t_0))
\end{aligned}
$$

Main idea

$$\begin{aligned}
y^\varepsilon(t_0+h) &\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0+h}(y^\varepsilon(0))\right)\\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}(y^\varepsilon(0))\right)\right)\\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}(y^\varepsilon(0))\right)\right)\\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)(y^\varepsilon(t_0)) \qquad (27)\\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\Phi^{\widetilde{F^{\varepsilon,[n_\varepsilon]}_h}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)(y^\varepsilon(t_0))
\end{aligned}$$

Main idea

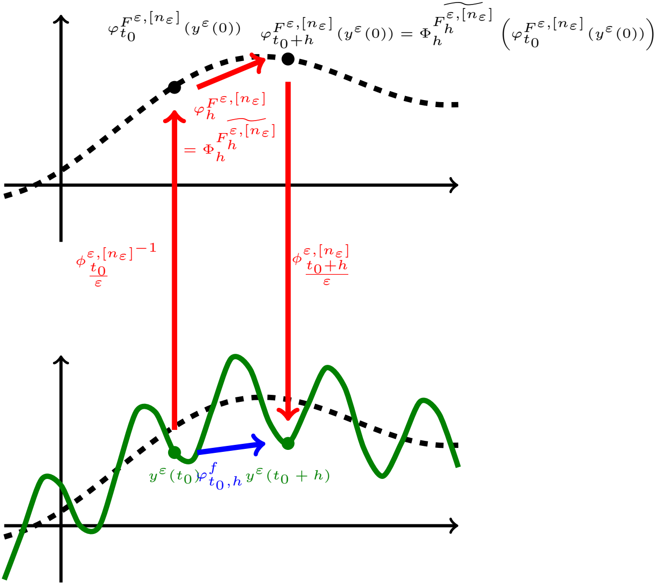
$$
\begin{aligned}
y^\varepsilon(t_0+h) &\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0+h}(y^\varepsilon(0))\right)\\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}(y^\varepsilon(0))\right)\right)\\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}(y^\varepsilon(0))\right)\right)\\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)(y^\varepsilon(t_0)) \qquad (27)\\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\Phi_h^{\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)(y^\varepsilon(t_0))
\end{aligned}
$$

Main idea

$$
\begin{aligned}
y^{\varepsilon}(t_0+h) &\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0+h}\left(y^{\varepsilon}(0)\right)\right) \\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}\left(y^{\varepsilon}(0)\right)\right)\right) \\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}\left(y^{\varepsilon}(0)\right)\right)\right) \\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)\left(y^{\varepsilon}(t_0)\right) \qquad (27) \\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\Phi^{\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)\left(y^{\varepsilon}(t_0)\right)
\end{aligned}
$$

Main idea

$$
\begin{aligned}
y^{\varepsilon}(t_0+h) &\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0+h}(y^{\varepsilon}(0))\right) \\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}(y^{\varepsilon}(0))\right)\right) \\
&\approx \phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)\left(\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{t_0}(y^{\varepsilon}(0))\right)\right) \\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\varphi^{F^{\varepsilon,[n_\varepsilon]}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)(y^{\varepsilon}(t_0)) \qquad (27) \\
&\approx \left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0+h}{\varepsilon}}\circ\Phi^{\widetilde{F^{\varepsilon,[n_\varepsilon]}_h}}_{h}\circ\left(\phi^{\varepsilon,[n_\varepsilon]}_{\frac{t_0}{\varepsilon}}\right)^{-1}\right)(y^{\varepsilon}(t_0))
\end{aligned}
$$


$\varphi_{t_0}^{F^{\varepsilon,[n_\varepsilon]}}(y^\varepsilon(0))$
$\varphi_{t_0+h}^{F^{\varepsilon,[n_\varepsilon]}}(y^\varepsilon(0)) = \Phi_h^{\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}}\left(\varphi_{t_0}^{F^{\varepsilon,[n_\varepsilon]}}(y^\varepsilon(0))\right)$
$\varphi_h^{F^{\varepsilon,[n_\varepsilon]}}$
$= \Phi_h^{\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}}$
$\phi_{\frac{t_0}{\varepsilon}}^{\varepsilon,[n_\varepsilon]^{-1}}$
$\phi_{\frac{t_0+h}{\varepsilon}}^{\varepsilon,[n_\varepsilon]}$
$y^\varepsilon(t_0)$
$\varphi_{t_0,h}^{f}$
$y^\varepsilon(t_0+h)$

Neural networks structures

- **Structure of F_θ (approximation of $\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}(y)$):**

$$F_\theta(y,h,\varepsilon) = \langle f\rangle(y) + R_{\theta,F}(y,h,\varepsilon) \quad (28)$$

- **Structure of ϕ_θ (approximation of $\phi_\tau^{\varepsilon,[n_\varepsilon]}(y)$):**

$$\phi_\theta(\tau,y,\varepsilon) = y + \varepsilon\left[R_\theta(\cos(\tau),\sin(\tau),y,\varepsilon) - R_\theta(1,0,y,\varepsilon)\right] \quad (29)$$

- **Structure of $\phi_{\theta,-}$ (approximation of $\phi_\tau^{\varepsilon,[n_\varepsilon]^{-1}}(y)$):**

$$\phi_{\theta,-}(\tau,y,\varepsilon) = y + \varepsilon\left[R_{\theta,-}(\cos(\tau),\sin(\tau),y,\varepsilon) - R_{\theta,-}(1,0,y,\varepsilon)\right] \quad (30)$$

Neural networks structures

- **Structure of F_θ (approximation of $\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}(y)$):**

$$F_\theta(y,h,\varepsilon) = \langle f \rangle(y) + R_{\theta,F}(y,h,\varepsilon) \tag{28}$$

- **Structure of ϕ_θ (approximation of $\phi_\tau^{\varepsilon,[n_\varepsilon]}(y)$):**

$$\phi_\theta(\tau,y,\varepsilon) = y + \varepsilon\left[R_\theta(\cos(\tau),\sin(\tau),y,\varepsilon) - R_\theta(1,0,y,\varepsilon)\right] \tag{29}$$

- **Structure of $\phi_{\theta,-}$ (approximation of ${\phi_\tau^{\varepsilon,[n_\varepsilon]}}^{-1}(y)$):**

$$\phi_{\theta,-}(\tau,y,\varepsilon) = y + \varepsilon\left[R_{\theta,-}(\cos(\tau),\sin(\tau),y,\varepsilon) - R_{\theta,-}(1,0,y,\varepsilon)\right] \tag{30}$$

Neural networks structures

- **Structure of F_θ (approximation of $\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}(y)$):**

$$F_\theta(y,h,\varepsilon) = \langle f\rangle(y) + R_{\theta,F}(y,h,\varepsilon) \tag{28}$$

- **Structure of ϕ_θ (approximation of $\phi_\tau^{\varepsilon,[n_\varepsilon]}(y)$):**

$$\phi_\theta(\tau,y,\varepsilon) = y + \varepsilon\left[R_\theta(\cos(\tau),\sin(\tau),y,\varepsilon) - R_\theta(1,0,y,\varepsilon)\right] \tag{29}$$

- **Structure of $\phi_{\theta,-}$ (approximation of $\phi_\tau^{\varepsilon,[n_\varepsilon]^{-1}}(y)$):**

$$\phi_{\theta,-}(\tau,y,\varepsilon) = y + \varepsilon\left[R_{\theta,-}(\cos(\tau),\sin(\tau),y,\varepsilon) - R_{\theta,-}(1,0,y,\varepsilon)\right] \tag{30}$$

Neural networks structures

- **Structure of F_θ (approximation of $\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}(y)$):**

$$F_\theta(y,h,\varepsilon) = \langle f\rangle(y) + R_{\theta,F}(y,h,\varepsilon) \tag{28}$$

- **Structure of ϕ_θ (approximation of $\phi_\tau^{\varepsilon,[n_\varepsilon]}(y)$):**

$$\phi_\theta(\tau,y,\varepsilon) = y + \varepsilon\left[R_\theta(\cos(\tau),\sin(\tau),y,\varepsilon) - R_\theta(1,0,y,\varepsilon)\right] \tag{29}$$

- **Structure of $\phi_{\theta,-}$ (approximation of $\phi_\tau^{\varepsilon,[n_\varepsilon]^{-1}}(y)$):**

$$\phi_{\theta,-}(\tau,y,\varepsilon) = y + \varepsilon\left[R_{\theta,-}(\cos(\tau),\sin(\tau),y,\varepsilon) - R_{\theta,-}(1,0,y,\varepsilon)\right] \tag{30}$$

Data creation

- **Repeat for $0 \leqslant k \leqslant K-1$.**
- **Select time step** $h^k \in [h_-, h_+]$ randomly.
- **Select high oscillation parameter** $\varepsilon^k \in [\varepsilon_-, \varepsilon_+]$ randomly.
- **Select initial conditions** $t_0^k \in [0, 2\pi]$ and $y_0^k \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi^f_{t_0^k, h^k}(y_0^k)$ with accurate but expensive integrator.

Data creation

- **Repeat for $0 \leqslant k \leqslant K-1$.**
- **Select time step** $h^k \in [h_-, h_+]$ randomly.
- **Select high oscillation parameter** $\varepsilon^k \in [\varepsilon_-, \varepsilon_+]$ randomly.
- **Select initial conditions** $t_0^k \in [0, 2\pi]$ and $y_0^k \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi^f_{t_0^k, h^k}(y_0^k)$ with accurate but expensive integrator.

Data creation

- **Repeat for $0 \leqslant k \leqslant K-1$.**
- **Select time step** $h^k \in [h_-, h_+]$ randomly.
- **Select high oscillation parameter** $\varepsilon^k \in [\varepsilon_-, \varepsilon_+]$ randomly.
- **Select initial conditions** $t_0^k \in [0, 2\pi]$ and $y_0^k \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi^f_{t_0^k, h^k}(y_0^k)$ with accurate but expensive integrator.

Data creation

- **Repeat for** $0 \leqslant k \leqslant K-1$**.**
- **Select time step** $h^k \in [h_-, h_+]$ randomly.
- **Select high oscillation parameter** $\varepsilon^k \in [\varepsilon_-, \varepsilon_+]$ randomly.
- **Select initial conditions** $t_0^k \in [0, 2\pi]$ and $y_0^k \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi^f_{t_0^k, h^k}(y_0^k)$ with accurate but expensive integrator.

Data creation

- **Repeat for $0 \leqslant k \leqslant K-1$.**
- **Select time step** $h^k \in [h_-, h_+]$ randomly.
- **Select high oscillation parameter** $\varepsilon^k \in [\varepsilon_-, \varepsilon_+]$ randomly.
- **Select initial conditions** $t_0^k \in [0, 2\pi]$ and $y_0^k \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi^f_{t_0^k, h^k}(y_0^k)$ with accurate but expensive integrator.

Data creation

- **Repeat for** $0 \leqslant k \leqslant K-1$**.**
- **Select time step** $h^k \in [h_-, h_+]$ randomly.
- **Select high oscillation parameter** $\varepsilon^k \in [\varepsilon_-, \varepsilon_+]$ randomly.
- **Select initial conditions** $t_0^k \in [0, 2\pi]$ and $y_0^k \in \Omega \subset \mathbb{R}^d$ randomly.
- **Compute exact flow** $y_1^{(k)} = \varphi^f_{t_0^k, h^k}(y_0^k)$ with accurate but expensive integrator.

Training

- $Loss_{Train}$ **optimization:**

$$
\begin{aligned}
Loss_{Train} &= \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|y_1^k - \hat{y_1}^k\right|^2 \\
&+ \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2 \\
&+ \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2
\end{aligned}
\tag{31}
$$

Training

- $Loss_{Train}$ **optimization:**

$$
\begin{aligned}
Loss_{Train} &= \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|y_1^k - \hat{y_1}^k\right|^2 \qquad (31)\\
&+ \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2\\
&+ \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2
\end{aligned}
$$

ODE part:

$$\hat{y_1}^k = \phi_\theta\left(\frac{t_0^k + h^k}{\varepsilon^k}, \Phi_{h^k}^{F_\theta(\cdot,h^k,\varepsilon^k)}\left(\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k}, y_0^k, \varepsilon^k\right)\right), \varepsilon^k\right). \qquad (32)$$

+ Auto-encoder part

Training

- $Loss_{Train}$ **optimization:**

$$
\begin{aligned}
Loss_{Train} &= \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|y_1^k - \hat{y_1}^k\right|^2 \qquad (31)\\
&+ \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)(y_0^k) - y_0^k\right|^2\\
&+ \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)(y_0^k) - y_0^k\right|^2
\end{aligned}
$$

ODE part:

$$
\hat{y_1}^k = \phi_\theta\left(\frac{t_0^k+h^k}{\varepsilon^k}, \Phi_{h^k}^{F_\theta(\cdot,h^k,\varepsilon^k)}\left(\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},y_0^k,\varepsilon^k\right)\right),\varepsilon^k\right). \qquad (32)
$$

+ Auto-encoder part

Training

- $Loss_{Train}$ **optimization:**

$$
\begin{aligned}
Loss_{Train} \quad &= \quad \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|y_1^k - \hat{y_1}^k\right|^2 \qquad (31)\\
&+ \quad \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2\\
&+ \quad \frac{1}{K_0}\sum_{k=0}^{K_0-1}\left|\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2
\end{aligned}
$$

ODE part:

$$
\hat{y_1}^k = \phi_\theta\left(\frac{t_0^k + h^k}{\varepsilon^k}, \Phi_{h^k}^{F_\theta(\cdot,h^k,\varepsilon^k)}\left(\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k}, y_0^k, \varepsilon^k\right)\right), \varepsilon^k\right). \qquad (32)
$$

+ Auto-encoder part

Training

- **Good training:** $Loss_{Train}$ has the same decay pattern as:

$$
\begin{aligned}
Loss_{Test} \quad = \quad & \frac{1}{K-K_0}\sum_{k=K_0}^{K-1}\left|y_1^k - \hat{y_1}^k\right|^2 \\
+ \quad & \frac{1}{K-K_0}\sum_{k=K_0}^{K-1}\left|\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2 \\
+ \quad & \frac{1}{K-K_0}\sum_{k=K_0}^{K-1}\left|\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2.
\end{aligned}
\tag{33}
$$

- **At the end:** Good approximation of $\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}$ and $\phi^{\varepsilon,[n_\varepsilon]}$ after training (only one training required).

Training

- **Good training:** $Loss_{Train}$ has the same decay pattern as:

$$\begin{aligned}
Loss_{Test} &= \frac{1}{K-K_0}\sum_{k=K_0}^{K-1}\left|y_1^k - \hat{y_1}^k\right|^2 \\
&+ \frac{1}{K-K_0}\sum_{k=K_0}^{K-1}\left|\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2 \\
&+ \frac{1}{K-K_0}\sum_{k=K_0}^{K-1}\left|\phi_{\theta,-}\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\circ\phi_\theta\left(\frac{t_0^k}{\varepsilon^k},\cdot,\varepsilon^k\right)\left(y_0^k\right) - y_0^k\right|^2.
\end{aligned} \tag{33}$$

- **At the end:** Good approximation of $\widetilde{F_h^{\varepsilon,[n_\varepsilon]}}$ and $\phi^{\varepsilon,[n_\varepsilon]}$ after training (only one training required).

Numerical integration inspired from Micro-Macro method

- **Numerical integration for Macro part:**

$$\begin{cases} & v_{\theta,0} & = & y^{\varepsilon}(0) \\ \forall n \in \mathbb{N}, & v_{\theta,n+1} & = & \Phi_h^{F_\theta(\cdot,h,\varepsilon)}(v_{\theta,n}) \end{cases} \tag{34}$$

- **Numerical integration for Micro part:** We introduce Micro part vector field:

$$\begin{aligned} g_\theta(\tau,w,v,\varepsilon) &:= f(\tau,\phi_\theta(\tau,v,\varepsilon)+w) - \frac{1}{\varepsilon}\partial_\tau\phi_\theta(\tau,v,\varepsilon) \\ &- \partial_y\phi_\theta(\tau,v,\varepsilon)F_\theta(v,0,\varepsilon) \end{aligned} \tag{35}$$

$$\begin{cases} & w_{\theta,0} & = & 0 \\ \forall n \in \mathbb{N}, & w_{\theta,n+1} & = & \Phi_{nh,h}^{g_\theta\left(\frac{nh}{\varepsilon},\cdot,v_{\theta,n},\varepsilon\right)}(w_{\theta,n}) \end{cases} \tag{36}$$

- **Approximation of the solution:**

$$\forall n \in \mathbb{N}, \quad y_{\theta,n}^{\varepsilon} = \phi_\theta\left(\frac{nh}{\varepsilon}, v_{\theta,n}, \varepsilon\right) + w_{\theta,n}. \tag{37}$$

Numerical integration inspired from Micro-Macro method

- **Numerical integration for Macro part:**

$$\left\{\begin{array}{rcl} & v_{\theta,0} & = & y^{\varepsilon}(0) \\ \forall n \in \mathbb{N}, & v_{\theta,n+1} & = & \Phi_h^{F_\theta(\cdot,h,\varepsilon)}(v_{\theta,n}) \end{array}\right. \tag{34}$$

- **Numerical integration for Micro part:** We introduce Micro part vector field:

$$\begin{array}{rcl} g_\theta\left(\tau,w,v,\varepsilon\right) & := & f\left(\tau,\phi_\theta\left(\tau,v,\varepsilon\right)+w\right)-\frac{1}{\varepsilon}\partial_\tau\phi_\theta\left(\tau,v,\varepsilon\right) \\ & - & \partial_y\phi_\theta\left(\tau,v,\varepsilon\right)F_\theta\left(v,0,\varepsilon\right) \end{array} \tag{35}$$

$$\left\{\begin{array}{rcl} & w_{\theta,0} & = & 0 \\ \forall n \in \mathbb{N}, & w_{\theta,n+1} & = & \Phi_{nh,h}^{g_\theta\left(\frac{nh}{\varepsilon},\cdot,v_{\theta,n},\varepsilon\right)}(w_{\theta,n}) \end{array}\right. \tag{36}$$

- **Approximation of the solution:**

$$\forall n \in \mathbb{N}, \quad y^{\varepsilon}_{\theta,n} = \phi_\theta\left(\frac{nh}{\varepsilon}, v_{\theta,n}, \varepsilon\right) + w_{\theta,n}. \tag{37}$$

Numerical integration inspired from Micro-Macro method

- **Numerical integration for Macro part:**

$$\begin{cases} & v_{\theta,0} = y^{\varepsilon}(0) \\ \forall n \in \mathbb{N}, & v_{\theta,n+1} = \Phi_h^{F_\theta(\cdot,h,\varepsilon)}(v_{\theta,n}) \end{cases} \tag{34}$$

- **Numerical integration for Micro part:** We introduce Micro part vector field:

$$\begin{aligned} g_\theta(\tau,w,v,\varepsilon) &:= f(\tau,\phi_\theta(\tau,v,\varepsilon)+w) - \frac{1}{\varepsilon}\partial_\tau\phi_\theta(\tau,v,\varepsilon) \\ &- \partial_y\phi_\theta(\tau,v,\varepsilon)F_\theta(v,0,\varepsilon) \end{aligned} \tag{35}$$

$$\begin{cases} & w_{\theta,0} = 0 \\ \forall n \in \mathbb{N}, & w_{\theta,n+1} = \Phi_{nh,h}^{g_\theta\left(\frac{nh}{\varepsilon},\cdot,v_{\theta,n},\varepsilon\right)}(w_{\theta,n}) \end{cases} \tag{36}$$

- **Approximation of the solution:**

$$\forall n \in \mathbb{N}, \quad y_{\theta,n}^{\varepsilon} = \phi_\theta\left(\frac{nh}{\varepsilon}, v_{\theta,n}, \varepsilon\right) + w_{\theta,n}. \tag{37}$$

Numerical integration inspired from Micro-Macro method

- Let introduce both learning errors:

$$\delta_\phi := \left\|\frac{\phi^{[p]} - \phi_\theta}{\varepsilon}\right\|_{W^{1,\infty}(\Omega\times[0,2\pi]),L^\infty\left([0,\varepsilon_+]\right)} \quad (38)$$

$$\delta_F := \left\|\widetilde{F_h^{\varepsilon,[p],[q]}} - F_\theta\right\|_{L^\infty\left(\Omega\times[0,h_+]\times[0,\varepsilon_+]\right)} \quad (39)$$

Theorem (M.B., P.Chartier, M.Lemou, F.Méhats - 2024)

Assuming that

- *For* $f_1, f_2 : [0,T] \times \Omega \to \mathbb{R}^d$ *smooth enough and* $t \in [0,T]$, *we have, for all* $h > 0$,

$$\left|\left|\Phi_{t,h}^{f_1} - \Phi_{t,h}^{f_2}\right|\right|_{L^\infty(\Omega)} \leqslant Ch \left|\left|f_1 - f_2\right|\right|_{L^\infty(\Omega)} \tag{40}$$

for some positive constant $C > 0$ *independent from* f_1 *and* f_2.

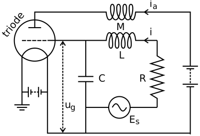
- *For* $f : [0,T] \times \Omega \to \mathbb{R}^d$ *smooth enough and* $t \in [0,T]$, *there exists* $L_f > 0$ *s.t. for all* $y_1, y_2 \in \Omega$, *for all* $h > 0$, *we have*

$$\left|\Phi_{t,h}^{f}(y_1) - \Phi_{t,h}^{f}(y_2)\right| \leqslant (1 + L_f h)|y_1 - y_2| \tag{41}$$

Then there exists positive constants C', M' *such that:*

$$\underset{0 \leqslant n \leqslant N}{Max} |e_{\theta,n}| \quad \leqslant \quad M'h^p + C'\left[\delta_F + \delta_\phi\right] \tag{42}$$

Numerical test: Van der Pol oscillator - Forward Euler



source: wikipedia.org

$$\left\{\begin{array}{rcl} \dot{q} & = & \frac{1}{\varepsilon}p \\ \dot{p} & = & -\frac{1}{\varepsilon}q + \left(\frac{1}{4} - q^2\right)p. \end{array}\right. \tag{43}$$

by making the variable change $(y_1, y_2) \longmapsto S\left(\frac{t}{\varepsilon}\right)(q,p)$, where $\tau \longmapsto S(\tau)$ is given by

$$\tau \longmapsto S(\tau) = \begin{bmatrix} \cos(\tau) & -\sin(\tau) \\ \sin(\tau) & \cos(\tau) \end{bmatrix} \tag{44}$$

we get the system:

$$\left\{\begin{array}{rcl} \dot{y_1}(t) & = & -\sin\left(\frac{t}{\varepsilon}\right)\left[\frac{1}{4} - \left(y_1(t)\cos\left(\frac{t}{\varepsilon}\right) + y_2(t)\sin\left(\frac{t}{\varepsilon}\right)\right)^2\right]\left[-y_1(t)\sin\left(\frac{t}{\varepsilon}\right) + y_2(t)\cos\left(\frac{t}{\varepsilon}\right)\right] \\ \dot{y_2}(t) & = & \cos\left(\frac{t}{\varepsilon}\right)\left[\frac{1}{4} - \left(y_1(t)\cos\left(\frac{t}{\varepsilon}\right) + y_2(t)\sin\left(\frac{t}{\varepsilon}\right)\right)^2\right]\left[-y_1(t)\sin\left(\frac{t}{\varepsilon}\right) + y_2(t)\cos\left(\frac{t}{\varepsilon}\right)\right] \end{array}\right.$$

Numerical test: Van der Pol oscillator - Forward Euler

Parameters	
# Math Parameters:	
Interval where time steps are selected:	$[h_-, h_+] = [10^{-3}, 10^{-1}]$
Interval where small parameters are selected:	$[\varepsilon_-, \varepsilon_+] = [10^{-3}, 1]$
Time for ODE simulation:	$T = 1$
Time step for ODE simulation:	$h = 0.01$
High oscillation parameter for ODE simulation:	$\varepsilon = 0.1$
# Machine Learning Parameters:	
Domain where initial data are selected:	$\Omega = [-2, 2]^2$
Number of data:	$K = 20\,000\,000$
Proportion of data for training:	80% - $K_0 = 16\,000\,000$
Hidden layers per MLP:	2
Neurons on each hidden layer:	200
Epochs:	200

Computational time for data creation: 2 Days 8 h 2 min 30 s
Computational time for training: 3 Days 13 h 50 min 30 s

Numerical test: Van der Pol oscillator - Forward Euler

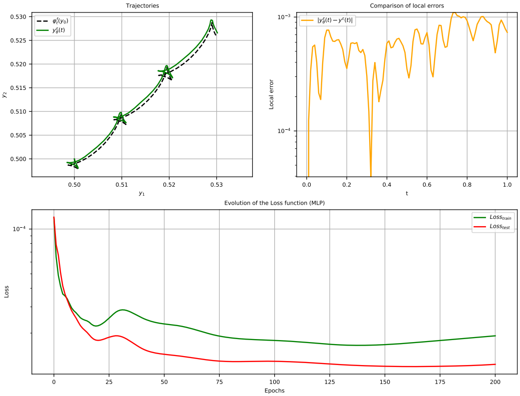


Figure: Comparison between $Loss$ decays (green: $Loss_{Train}$, red: $Loss_{Test}$), trajectories (dashed dark: exact flow, green: numerical flow with learned vector fields and local error (yellow) in the case $\varepsilon = 0.1$.

Numerical test: Van der Pol oscillator - Forward Euler

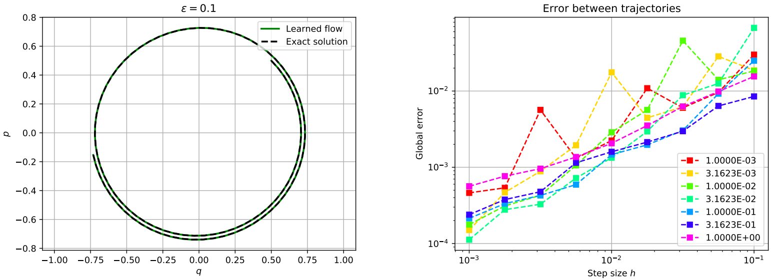
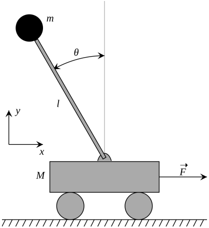


Figure: Left: Comparison between trajectories (dashed dark: exact flow, green: numerical flow with learned vector field) in the case $\varepsilon = 0.1$ after the inverse variable change. Right: Integration errors (each color corresponds to a parameter ε).

Numerical test: Inverted Pendulum - Midpoint



source: Wikipedia.org

$$\begin{cases} \dot{y_1^\varepsilon}(t) & = & y_2^\varepsilon(t) + \sin\left(\frac{t}{\varepsilon}\right)\sin\left(y_1^\varepsilon(t)\right) \\ \dot{y_2^\varepsilon}(t) & = & \sin\left(y_1^\varepsilon(t)\right) - \frac{1}{2}\sin\left(\frac{t}{\varepsilon}\right)^2 \sin\left(2y_1^\varepsilon(t)\right) - \sin\left(\frac{t}{\varepsilon}\right)\cos\left(y_1^\varepsilon(t)\right) y_2^\varepsilon(t) \end{cases} \tag{45}$$

Numerical test: Inverted Pendulum - Midpoint

Parameters	
# Math Parameters:	
Interval where time steps are selected:	$[h_-, h_+] = [10^{-3}, 10^{-1}]$
Interval where small parameters are selected:	$[\varepsilon_-, \varepsilon_+] = [10^{-3}, 1]$
Time for ODE simulation:	$T = 1$
Time step for ODE simulation:	$h = 0.01$
High oscillation parameter for ODE simulation:	$\varepsilon = 0.05$
# Machine Learning Parameters:	
Domain where initial data are selected:	$\Omega = [-2, 2]^2$
Number of data:	$K = 1\,000\,000$
Proportion of data for training:	80% - $K_0 = 800\,000$
Hidden layers per MLP:	2
Neurons on each hidden layer:	200
Epochs:	200

Computational time for data creation: 1 h 44 min 19 s
Computational time for training: 9 h 11 min 8 s

Numerical test: Inverted Pendulum - Midpoint

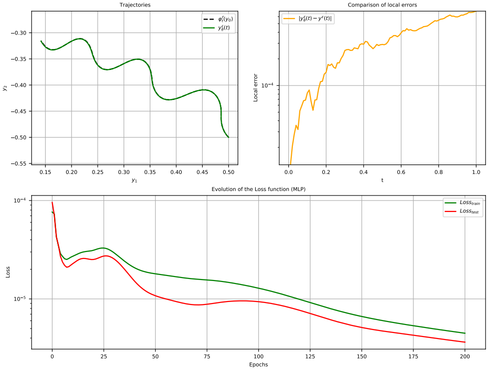


Figure: Comparison between *Loss* decays (green: $Loss_{Train}$, red: $Loss_{Test}$), trajectories (dashed dark: exact flow, green: numerical flow with learned vector fields and local error (yellow) in the case $\varepsilon = 0.05$.

Numerical test: Inverted Pendulum - Midpoint

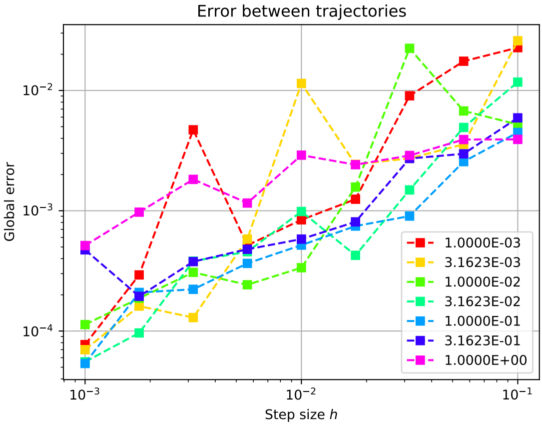


Figure: Integration errors (each color corresponds to a parameter ε).

Conclusion - Main results:

- Performing of existing numerical methods to solve autonomous ODE's by usine Machine Learning to approximate truncated modified field[1].
- Approximation of averaged field and high oscillation generator (truncations) for highly oscillatory ODE's by adding auto-encoders[2].

Research perspectives:

- Include geometric properties for approximation of truncated averaged field and high oscillation generator (e.g. Hamiltonian equation of divergence free).

[1] M. B., Philippe Chartier, Mohammed Lemou & Florian Méhats, Machine Learning Methods for Autonomous Ordinary Differential Equations, 2023, accepted in *Communications in Mathematical Sciences* in December 2023.

[2] M. B., Philippe Chartier, Mohammed Lemou & Florian Méhats, *Machine Learning Methods for Highly Oscillatory Differential Equations*, 2024, to be submitted.

Conclusion - Main results:

- Performing of existing numerical methods to solve autonomous ODE's by usine Machine Learning to approximate truncated modified field[1].
- Approximation of averaged field and high oscillation generator (truncations) for highly oscillatory ODE's by adding auto-encoders[2].

Research perspectives:

- Include geometric properties for approximation of truncated averaged field and high oscillation generator (e.g. Hamiltonian equation of divergence free).

[1] M. B., Philippe Chartier, Mohammed Lemou & Florian Méhats, Machine Learning Methods for Autonomous Ordinary Differential Equations, 2023, accepted in *Communications in Mathematical Sciences* in December 2023.

[2] M. B., Philippe Chartier, Mohammed Lemou & Florian Méhats, *Machine Learning Methods for Highly Oscillatory Differential Equations*, 2024, to be submitted.

Conclusion - Main results:

- Performing of existing numerical methods to solve autonomous ODE's by usine Machine Learning to approximate truncated modified field[1].
- Approximation of averaged field and high oscillation generator (truncations) for highly oscillatory ODE's by adding auto-encoders[2].

Research perspectives:

- Include geometric properties for approximation of truncated averaged field and high oscillation generator (e.g. Hamiltonian equation of divergence free).

---

[1] M. B., Philippe Chartier, Mohammed Lemou & Florian Méhats, Machine Learning Methods for Autonomous Ordinary Differential Equations, 2023, accepted in *Communications in Mathematical Sciences* in December 2023.

[2] M. B., Philippe Chartier, Mohammed Lemou & Florian Méhats, *Machine Learning Methods for Highly Oscillatory Differential Equations*, 2024, to be submitted.