Deep reduced models for linear and nonlinear waves equations

R. Cote[2], E. Franck[1,2], E. Opshtein[2], L. Navoret[1,2],
C. Schnoebelen[1], G. Steimer[1,2], V. Vigon[1,2]

14/03/203

Journée Ondes du Sud Ouest, Toulouse

[1]Inria Nancy Grand Est, France
[2]IRMA, Strasbourg university, France

Outline

Introduction

Reduced order modeling I

- We are interested in the following type of parametric problems:

$$\left\{\begin{array}{ll} \partial_t u + \mathcal{N}(u, \partial_x u, \partial_{xx} u, \boldsymbol{\alpha}) = 0, & \text{in } \Omega_{\boldsymbol{\beta}} \\ u(t=0,x) = u_0(x, \boldsymbol{\gamma}) \end{array}\right.$$

  with all the parameters $\boldsymbol{\mu} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$.
- Solving this PDE for many parameters is important for control optimal, inverse problems, uncertain propagation.
- After a spatial discretization we have:

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{\mu})$$

  with $\boldsymbol{x}(t) \in \mathbb{R}^d$ and $d >> 1$.
- Solve many times this problem is very costly.

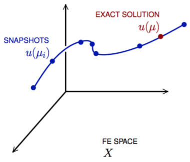
Idea of ROM

Construct a reduced model valid for a subset of $\boldsymbol{\mu}$ and use it for optimal control or other applications.

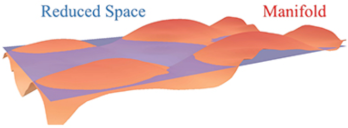
Reduced order modeling II

Principle

- **Manifold assumption**: The solutions live in a manifold of small dimensions (dimension of $\boldsymbol{\mu}$)
- **Idea**: determinate the manifold and project the equation on this manifold.



- The classical approach uses the assumption than the manifold is closed to a hyperplane.



POD approach + Galerkin Projection

Hyperplan Assumption

$$\boldsymbol{x}(t) \approx \tilde{\boldsymbol{x}}(t) = \boldsymbol{x}_{ref} + \Phi\hat{\boldsymbol{x}}(t)$$

with the decoder $\Phi \in \mathbb{R}^{d,m}$ and $m << d$

- How determinate Φ ? We construct a snapshot matrix:

$$X = \left\{x(t_1, \boldsymbol{\mu}_1) - x_{ref}, ..., x(t_{n_t}, \boldsymbol{\mu}_{n_\mu}) - x_{ref}\right\} \in \mathbb{R}^{d, n_t \times n_\mu}$$

- POD method solve the following problem:

$$\min_{\Phi, \Phi^t\Phi = I_d} \parallel X - \Phi\Phi^t X \parallel_F$$

- The solution is given by the m eigenvectors associated with the m maximal eigenvalues of XX^t.

Reduced model

- We make a Galerkin projection: represent the solution of the space $Vect(\Phi) + x_{ref}$ and project the derivative of time on the test space: $Vect(\Phi)$
- Result:

$$\frac{d\hat{\boldsymbol{x}}(t)}{dt} = \Phi^t \boldsymbol{F}(\boldsymbol{x}_{ref} + \Phi\hat{\boldsymbol{x}})$$

Applications to equation

- Damped wave equation

$$\partial_{tt} u - c^2 \Delta u = 0$$

- First order version: $v = \partial_t u$

$$\left\{ \begin{array}{l} \partial_t u = v \\ \partial_t v = c^2 \partial_{xx} u \end{array} \right.$$

- Energy balance:

$$\frac{d}{dt} \int_{\Omega} \left(\frac{v^2}{2c^2} + \frac{(\partial_x u)^2}{2} \right) = 0$$

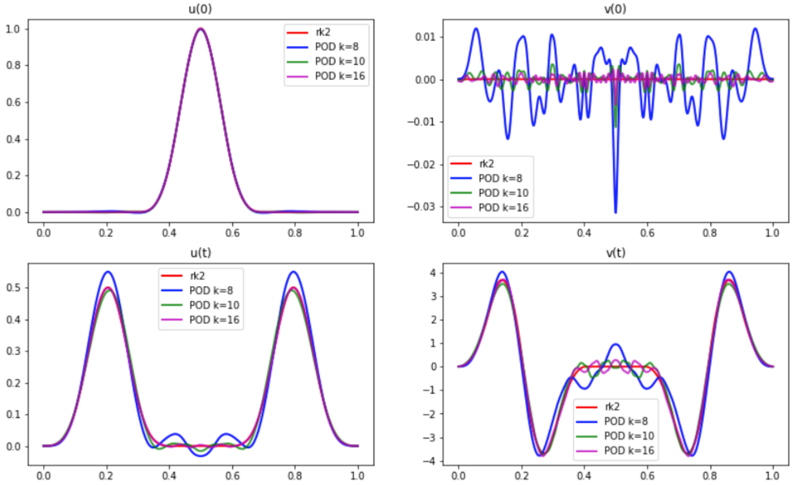
- We apply the POD + Galerkin method:

$$\frac{d}{dt} \begin{pmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{v}} \end{pmatrix} = \hat{A} \begin{pmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{v}} \end{pmatrix} + \Phi^t \begin{pmatrix} \boldsymbol{u}_{ref} \\ \boldsymbol{v}_{ref} \end{pmatrix}$$

with $\hat{A} = \Phi^t \left(\begin{pmatrix} 0 & I_d \\ -c^2 D_{hh} & 0 \end{pmatrix} \right) \Phi$ precomputed and D_{hh} the discrete Laplacian.

Results

- We compress the wave equation using POD.
- In the data set we take 20 values of $c \in [0.2, 0.6]$



- Less efficient than for diffusion problems. Gibbs phenomena.

Structure preserving linear reduction

Hamiltonian structure of general wave equation

- The POD does not work well for wave equation. How improve that ?
- Energy balanced:

$$\frac{d}{dt}\int_{\Omega}\left(\frac{v^2}{2c^2}+\frac{(\partial_x u)^2}{2}\right)=0$$

- Discretization with staggered grids (or other structures preserving method) we obtain:

$$\frac{d}{dt}\begin{pmatrix} \boldsymbol{u}_h \\ \boldsymbol{v}_h \end{pmatrix}=\mathcal{J}\nabla H(\boldsymbol{u}_h,\boldsymbol{v}_h)$$

with

$$\mathcal{J}=\left(\begin{pmatrix} d & I_d \\ -I_d & 0 \end{pmatrix}\right), \quad H(\boldsymbol{u}_h,\boldsymbol{v}_h)=\Delta_x\sum_{i=1}^{N}\left(v_i^2+\frac{(u_{i+1}-u_i)^2}{2\Delta x^2}+\frac{(u_i-u_{i-1})^2}{2\Delta x^2}\right)$$

Hamiltonian systems

We speak about Hamiltonian system. By construction the Hamiltonian is conserved in time. It allows assuring the stability of the system.

Symplectic flot and Symplectic scheme

- The Hamiltonian systems are a key object in symplectic geometric.
- **Symplectic map**: maps which preserves the symplectic form.
- The map: $(u, v) = \phi((q, p)) \in \mathbb{R}^{n_o}$, with $(q, p) \in \mathbb{R}^{n_i}$ with $n_i < n_o$ is a symplectic map if

$$(\nabla_{(q,p)}\phi)^t \mathcal{J}_{n_o} (\nabla_{(q,p)}\phi) = \mathcal{J}_{n_i}$$

- Galerkin projection with symplectic map preserve the Hamiltonian structure:

$$\frac{d}{dt}\left(\begin{array}{c} u \\ v \end{array}\right) = \mathcal{J}_{n_o}\nabla H(u, v) \rightarrow \frac{d}{dt}\left(\begin{array}{c} p \\ q \end{array}\right) = \mathcal{J}_{n_i}\nabla H(\phi(p, q))$$

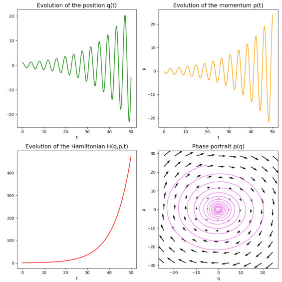
- Example: Pendulum.

Remark

The symplectic map are important tools.
Example we use a time scheme:

$$\begin{pmatrix} u^{n+1} \\ v^{n+1} \end{pmatrix} = \phi_{\Delta t} \begin{pmatrix} u^n \\ v^n \end{pmatrix}$$

The scheme where $\phi_{\Delta t}$ is a symplectic map admits better stability results.



Euler explicite

Symplectic flot and Symplectic scheme

- The Hamiltonian systems are a key object in symplectic geometric.
- **Symplectic map**: maps which preserves the symplectic form.
- The map: $(u, v) = \phi((q, p)) \in \mathbb{R}^{n_o}$, with $(q, p) \in \mathbb{R}^{n_i}$ with $n_i < n_o$ is a symplectic map if

$$(\nabla_{(q,p)}\phi)^t \mathcal{J}_{n_o} (\nabla_{(q,p)}\phi) = \mathcal{J}_{n_i}$$

- Galerkin projection with symplectic map preserve the Hamiltonian structure:

$$\frac{d}{dt}\begin{pmatrix} u \\ v \end{pmatrix} = \mathcal{J}_{n_o} \nabla H(u, v) \rightarrow \frac{d}{dt}\begin{pmatrix} p \\ q \end{pmatrix} = \mathcal{J}_{n_i} \nabla H(\phi(p, q))$$

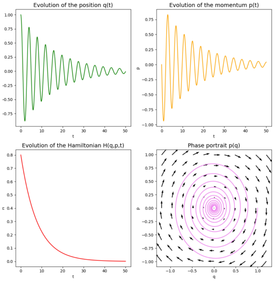
- Example: Pendulum.

Remark

The symplectic map are important tools.
Example we use a time scheme:

$$\begin{pmatrix} u^{n+1} \\ v^{n+1} \end{pmatrix} = \phi_{\Delta t}\begin{pmatrix} u^n \\ v^n \end{pmatrix}$$

The scheme where $\phi_{\Delta t}$ is a symplectic map admits better stability results.



Euler implicite

- The Hamiltonian systems are a key object in symplectic geometric.
- **Symplectic map**: maps which preserves the symplectic form.
- The map: $(u, v) = \phi((q, p)) \in \mathbb{R}^{n_o}$, with $(q, p) \in \mathbb{R}^{n_i}$ with $n_i < n_o$ is a symplectic map if

$$(\nabla_{(q,p)}\phi)^t \mathcal{J}_{n_o}(\nabla_{(q,p)}\phi) = \mathcal{J}_{n_i}$$

- Galerkin projection with symplectic map preserve the Hamiltonian structure:

$$\frac{d}{dt}\begin{pmatrix} u \\ v \end{pmatrix} = \mathcal{J}_{n_o}\nabla H(u, v) \rightarrow \frac{d}{dt}\begin{pmatrix} p \\ q \end{pmatrix} = \mathcal{J}_{n_i}\nabla H(\phi(p, q))$$

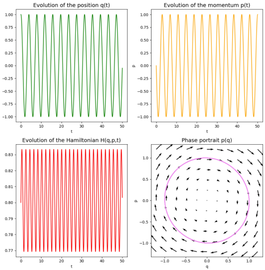
- Example: Pendulum.

Remark

The symplectic map are important tools.
Example we use a time scheme:

$$\begin{pmatrix} u^{n+1} \\ v^{n+1} \end{pmatrix} = \phi_{\Delta t}\begin{pmatrix} u^{n} \\ v^{n} \end{pmatrix}$$

The scheme where $\phi_{\Delta t}$ is a symplectic map admits better stability results.



Euler Symplectique

PSD

The idea of structure preserving reduction: propose a POD-type method which is a symplectic map. We speak about PSD.

- PSD method (Hestaven and al) solve the following problem:

$$\min_{A, A^t A = I_d, A^t \mathcal{J} A = \mathcal{J}} \parallel X - AA^t X \parallel_F$$

  - We construct snapshots matrix:

$$X = \left\{ u(t_1, \boldsymbol{\mu}_1), ..., u(t_{n_t}, \boldsymbol{\mu}_{n_\mu}), v(t_1, \boldsymbol{\mu}_1), ..., v(t_{n_t}, \boldsymbol{\mu}_{n_\mu}) \right\}$$

  - We compute a POD on X to obtain Φ
  - We obtain the decoder:

$$A = \left(\begin{pmatrix} \Phi & 0 \\ 0 & \Phi \end{pmatrix} \right)$$

- We obtain a Hamiltonian reduced model:

$$\frac{d}{dt} \begin{pmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{v}} \end{pmatrix} = \mathcal{J}_m \nabla H \left(\Phi \begin{pmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{v}} \end{pmatrix} \right)$$

- **Hyper-reduction**: method to construct $\hat{H} \left(\begin{pmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{v}} \end{pmatrix} \right)$.

Results for linear wave equation I

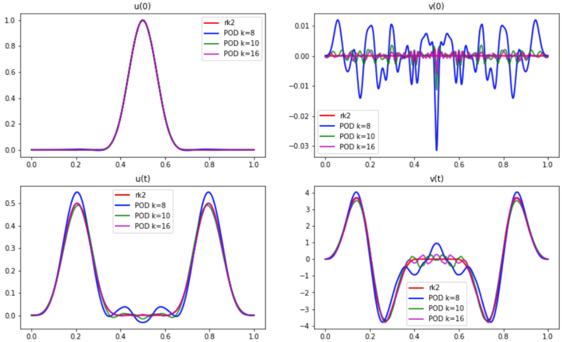
- General wave equations: $\partial_{tt}\mathbf{u} - \partial_x\left[\nabla_U V(\partial_x \mathbf{u})\right] = 0$. First order form:

$$\begin{cases} \partial_t \mathbf{u} = \mathbf{v} \\ \partial_t \mathbf{v} = \partial_x \left[\nabla_{\mathbf{u}} V(\partial_x \mathbf{u})\right] \end{cases}$$

with

$$H(\mathbf{u},\mathbf{v}) = \int_\Omega \left(\frac{1}{2} \mid \mathbf{v} \mid^2 + \nabla_{\mathbf{u}} V(\partial_x \mathbf{u})\right) dx$$

- We compress the linear wave equation using POD and PSD.
- In the data set we take 20 values of $c \in [0.2, 0.6]$



Results for linear wave equation I

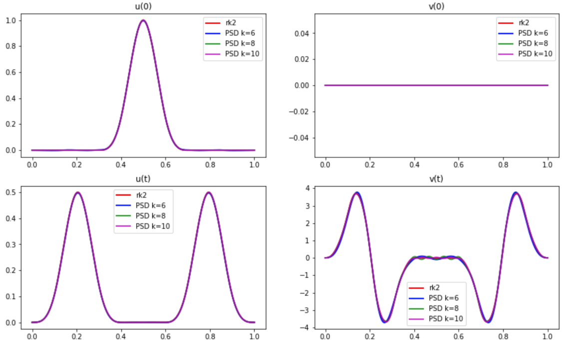
- General wave equations: $\partial_{tt}\mathbf{u} - \partial_x \left[\nabla_U V(\partial_x \mathbf{u})\right] = 0$. First order form:

$$\begin{cases} \partial_t \mathbf{u} = \mathbf{v} \\ \partial_t \mathbf{v} = \partial_x \left[\nabla_{\mathbf{u}} V(\partial_x \mathbf{u})\right] \end{cases}$$

  with

$$H(\mathbf{u},\mathbf{v}) = \int_\Omega \left(\frac{1}{2} \mid \mathbf{v} \mid^2 + \nabla_{\mathbf{u}} V(\partial_x \mathbf{u}) \right) dx$$

- We compress the linear wave equation using POD and PSD.
- In the data set we take 20 values of $c \in [0.2, 0.6]$

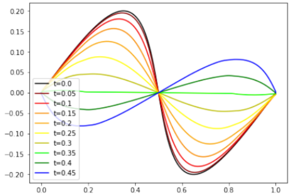


Results for linear wave equation II

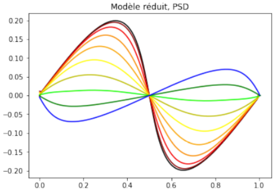
- We solve linear wave equation for Piano string (Chabassier 10)

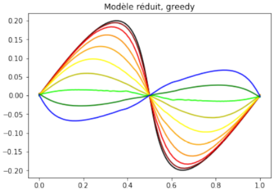
$$\begin{cases} \partial_{tt} u_1 = \partial_x((1-\alpha)\partial_x u_1) \\ \partial_{tt} u_2 = \partial_{xx} u_2 \end{cases}$$

  with $\alpha \approx 0.5$
- Solution in high dimension:



- Solution in low dimension with 5 mods:



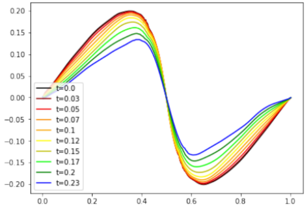


Results for nonlinear wave equations

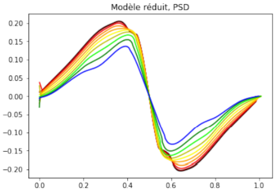
- We solve nonlinear wave equation for Piano string (Chabassier 12) with fixed parameters.

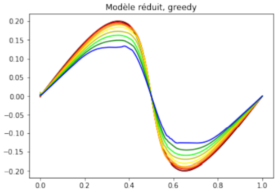
$$\begin{cases} \partial_{tt} u_1 = \partial_x \left[(1-\alpha)\partial_x u_1 + \alpha \partial_x u \partial_x v + \frac{1}{2}(\partial_x u)^3 \right] \\ \partial_{tt} u_2 = \partial_x (\partial_x u_2 + \frac{\alpha}{2} (\partial_x u)^2) = 0 \end{cases}$$

  with $\alpha \approx 0.8$
- Solution in high dimension (200 cells):



- Solution in low dimension with 5 mods without hyper-reduction:





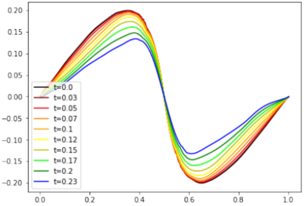
Results for nonlinear wave equations

- We solve nonlinear wave equation for Piano string (Chabassier 12) with fixed parameters.

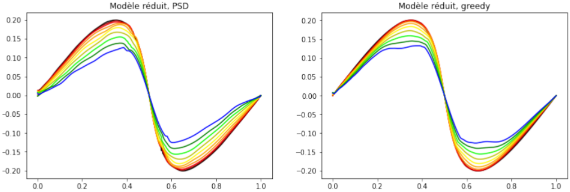
$$\begin{cases} \partial_{tt} u_1 = \partial_x \left[(1-\alpha)\partial_x u_1 + \alpha \partial_x u \partial_x v + \frac{1}{2}(\partial_x u)^3 \right] \\ \partial_{tt} u_2 = \partial_x (\partial_x u_2 + \frac{\alpha}{2} (\partial_x u)^2) = 0 \end{cases}$$

with $\alpha \approx 0.8$

- Solution in high dimension (200 cells):



- Solution in low dimension with 20 mods without hyper-reduction:



Nonlinear reduction

Principle of nonlinear-reduction

- We make the assumption that the manifold solution can be approximate by a hyperplane. Not realistic for strongly nonlinear PDE.

Nonlinear assumption

$$\boldsymbol{x}(t) \approx \tilde{\boldsymbol{x}}(t) = G(\hat{\boldsymbol{x}}(t))$$

with $G(.) : \mathbb{R}^m \rightarrow \mathbb{R}^d$ and $m << d$

- How construct the decoder G:
  - neural networks like auto-encoder,
  - manifold learning approach (extension to POD for manifold) + regression.

aim

Combine nonlinear reduction method and structure preserving one.

- Solution:
  - Weakly symplectic decoder (Buchfink an al 2021).
  - Non Symplectic decoder but Hamiltonian reduced models (our work).
  - Symplectic decoder (open question).

Maching learning: principle

- Supervised learning: we want approximate

$$y = f(x) + \epsilon$$

  with ϵ some noise and f unknown.
- We know a set $((x_1, y_1), ..., (x_n, y_n))$. We use a parametric function and find the parameters solving:

$$\min_{\theta} \sum_{i=1}^{N} \parallel y_i - f_\theta(x_i) \parallel_2^2$$

- **Which parametric functions**? Neural network.

Layer

A layer is a function $L_l(\mathbf{x}_l) : \mathbb{R}^{d_l} \rightarrow \mathbb{R}^{d_{l+1}}$ given by

$$L_l(\mathbf{x}_l) = \sigma(A_l \mathbf{x}_l + \mathbf{b}_l),$$

$A_l \in \mathbb{R}^{d_{l+1}, d_l}$, $\mathbf{b} \in \mathbb{R}^{d_{l+1}}$ and $\sigma()$ a nonlinear function applied component by component.

Neural network

A neural network is parametric function obtained by composition of layers:

$$f_\theta(\mathbf{x}) = L_n \circ .... \circ L_1(\mathbf{x})$$

with θ the trainable parameters composed of all the matrices $A_{l,l+1}$ and biases $\mathbf{b}_l$.

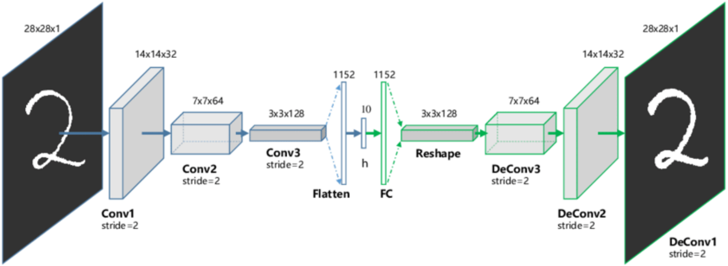
Auto-encoder

Auto-encoder

We propose two networks $E_{\theta_e}(x) : \mathbb{R}^d \rightarrow \mathbb{R}^m$ and $D_{\theta_d}(x) : \mathbb{R}^m \rightarrow \mathbb{R}^d$ with $m << d$ such that

$$\min_{\theta_e,\theta_d} \sum_{i=1}^{n} \parallel x_i - D_{\theta_d}(E_{\theta_e}(x_i)) \parallel_2^2$$

- For high-dimensional data living on grids we use Convolutional neural networks.
- Example of CAE:



Strategy

Coupling nonlinear reduction with learning reduced Hamiltonian ODE (HNN) in the reduced space.

- How learn a Hamiltonian system.
  - We estimate the derivative of data $\left\{(\frac{dy}{dt})_1, ...(\frac{dy}{dt})_n\right\}$ with finite difference and we solve

$$\min_{\theta} \sum_{i=1}^{n} \parallel (\frac{dy}{dt})_i - \mathcal{J}\nabla H_\theta(y_i) \parallel_2^2$$

  - If we define the scheme $S_\theta(y_i) = y_i + \Delta t \mathcal{J}\nabla H_\theta(y_i)$ we minimize:

$$\min_{\theta} \sum_{i=1}^{n} \parallel y_{i+L} - \underbrace{S_\theta \circ ... \circ S_\theta(y_i)}_{\text{L times}} \parallel_2^2$$

  - The gradient of $S_\theta \circ ... \circ S_\theta$ can be computed using automatic differentiation tools. We speak about differentiable physics.

Full nonNonlinear-reduction with Hnn model II

Final loss

- AE loss:

$$\min_{\theta_e,\theta_d} \sum_{i=1}^{n} \| x_i - D_{\theta_d}(E_{\theta_e}(x_i)) \|_2^2$$

- HNN loss:

$$\min_{\theta} \sum_{i=1}^{n} \| E_{\theta_e}(x_{i+L}) - \underbrace{S_\theta \circ ... \circ S_\theta(E_{\theta_e}(x_i))}_{\text{L times}} \|_2^2$$

- Coupling loss (to enforce the encoded trajectory to be conservative):

$$\min_{\theta} \sum_{i=1}^{n} \| H_\theta(E_{\theta_e}(x_{i+L})) - H_\theta(E_{\theta_e}(x_i)) \|_2^2$$

- Full loss:

$$\min_{\theta} \sum_{i=1}^{n} \| x_{i+L} - D_{\theta_d}\underbrace{(S_\theta \circ ... \circ S_\theta(E_{\theta_e}(x_i)))}_{\text{L times}} \|_2^2$$

- We use CNN network for E_{θ_e} and D_{θ_d}.
- We use fully-connected network for H_θ.

Results linear wave

- We solve

$$\begin{cases} \partial_t u = v \\ \partial_t v = c^2 \partial_{xx} u \end{cases}$$

  with varying 20 values of $c \in [0.2, 0.6]$ in the data set.
- Result:

Models		$c = 0.2385$		$c = 0.3798$		$c = 0.5428$	
	dim/error	error u	error v	error u	error v	error u	error v
AE+HNN	$k = 2$	$2.4e^{-4}$	$4.2e^{-3}$	$5.3e^{-4}$	$9.5e^{-3}$	$3.4e^{-4}$	$6.6e^{-3}$
	$k = 1$	$2.1e^{-4}$	$9.2e^{-3}$	$2.2e^{-4}$	$7.6e^{-3}$	$3.6e^{-4}$	$9.0e^{-3}$
PSD	$k = 4$	$5e^{-2}$	$1.38e^{-1}$	$5.05e^{-2}$	$1.9e^{-1}$	$5e^{-2}$	$2.4e^{-1}$
	$k = 5$	$5.5e^{-3}$	$3.4e^{-2}$	5.9^{e-3}	$4.9e^{-2}$	$6.3e^{-3}$	$6.4e^{-2}$
	$k = 6$	$3.5e^{-4}$	$9e^{-3}$	$3.3e^{-4}$	$1.1e^{-2}$	$3.2e^{-4}$	$1.3e^{-2}$
POD	$k = 10$	$1.9e^{-3}$	$1.2e^{-2}$	$9.7e^{-4}$	$1.5e^{-2}$	$3.7e^{-3}$	$6.4e^{-2}$
	$k = 15$	$8.5e^{-4}$	$1.2e^{-2}$	$3.2e^{-4}$	$8.5e^{-3}$	$1.6e^{-3}$	$3.5e^{-2}$
	$k = 20$	$3.9e^{-4}$	$6.2e^{-3}$	$1.3e^{-4}$	$3.1e^{-3}$	$4.8e^{-4}$	$1.4e^{-2}$

- We use a HNN: $[24, 12, 12, 12, 6]+$ tanh. CNN: convolutional block with 2 convolution by block + 4 dense layers $[256, 128, 64, 32]$ + elu activation.

Remark

Our approach made similar result than PSD with the reduced dimension $k = 6$ or $k = 7$ and the POD with $k = 20$

Results nonlinear wave

- We solve

$$\begin{cases} \partial_t u = v \\ \partial_t v = \mu_1 \partial_x (W'(u, \mu_2) + g'(u, \mu_3)) \end{cases}$$

with

$$W(x, \mu) = \frac{1}{2}x^2 + \sin(\mu x), \quad g(x, \mu) = 10\mu x^3$$

and have 20 triplets (μ_1, μ_2, μ_3) in the dataset.

- Three tests (more and more nonlinear):

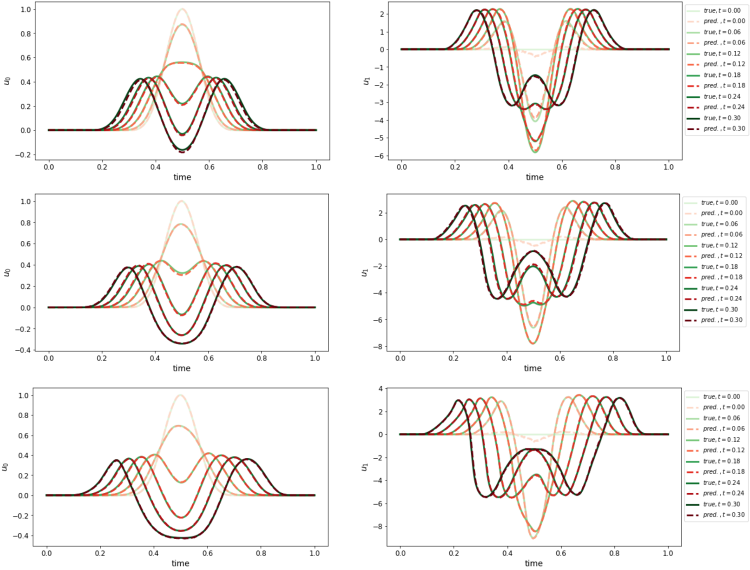
Model		Test 1		Test 2		Test 3	
	dim/error	error u	error v	error u	error v	error u	error v
AE+HNN	$k=3$	$4.8e^{-4}$	$3.1e^{-3}$	$9.7e^{-4}$	$1.2e^{-2}$	$3.5e^{-4}$	$4.6e^{-3}$
PSD	$k=16$	$9.3e^{-4}$	$1.7e^{-2}$	$8.3e^{-4}$	$1.8e^{-2}$	$1.8e^{-3}$	$3.0e^{-2}$
	$k=18$	$5.2e^{-4}$	$1.2e^{-2}$	$5.1e^{-4}$	$1.2e^{-2}$	$9.4e^{-4}$	$2.6e^{-2}$
	$k=24$	$3.2e^{-4}$	$7.2e^{-3}$	$3.1e^{-4}$	$7.4e^{-3}$	$4.0e^{-4}$	$1.5e^{-2}$
POD	$k=24$	$1.4e^{-2}$	$1.2e^{-1}$	$1.7e^{-2}$	$1.8e^{-1}$	$1.8e^{-2}$	$2.6e^{-1}$
	$k=40$	$7.6e^{-3}$	$1.17e^{-1}$	$1.1e^{-2}$	$1.1e^{-1}$	$1.1e^{-2}$	$2.5e^{-1}$

Remark

Our approach made similar results than PSD with the reduced dimension $k=18$ or $k=20$ and better than POD with $k=60$

Results nonlinear wave II

- Tests 1/2/3



Conclusion

Conclusion

Conclusion

The nonlinear reduction allows compressing more the parametric PDE. We can enforce the Hamiltonian structure at the reduced level and ensure more stability.

Future works

Adapt the method to treat nonlinear Vlasov equations for plasma physics (PhD of G. Steimer)

Future works II

Master and PhD Thesis of C. Schnoebelen:

- Space-time structure preserving methods for complex wave equations like Galbrun/linear MHD (DeRham Sequance + Symplectic scheme).
- Enhanced structure preserving methods by neural networks.
- Symplectic nonlinear decoder for Hamiltonian reduction.
- Reduced order modeling for varying medium wave equations.
- Extension to sphere case.