

Cours 11: Réduction de modèles, construction des modèles réduits

Emmanuel Franck^{*},

19 septembre, 2024

Master CMSI, M2, Strasbourg

^{*}MACARON project-team, Université de Strasbourg, CNRS, Inria, IRMA, France

The logo for Inria, featuring the word "Inria" in a red, cursive script font.The logo for IRMA, consisting of the letters "IRMA" in a blue, bold, sans-serif font. Below the letters is a horizontal blue line, and underneath that line, the text "Institut de Recherche Mathématique Avancée" is written in a smaller, blue, sans-serif font.

Modèles réduits et projection

- Projection de Galerkin pour les EDP elliptiques

- Projection de Galerkin pour les EDP temporelles

- Hyper-réduction et fermeture pour la réduction nonlinéaire

Modèles réduits et apprentissage d'EDO

- Hyper-réduction et fermeture

Modèles réduits et projection

Projection de Galerkin pour les EDP elliptiques

Projection de Galerkin pour les EDP temporelles

Hyper-réduction et fermeture pour la réduction nonlinéaire

Modèles réduits et apprentissage d'EDO

Hyper-réduction et fermeture

Modèles réduits et projection

Principe de la réduction d'ordre

Hypothèse

Dans la suite on va considérer qu'on a construit Φ ou G à partir de données $(\mathbf{u}_1, \dots, \mathbf{u}_m)$

Objectif

Etre rapidement capable de calculer une solution spatiale ou temporelle pour un nouveau paramètre μ . Pour cela

- On calcule $\hat{\mathbf{u}}(\mu)$ ou $\hat{\mathbf{u}}(t, \mu)$
- On décompresse avec Φ ou G

Comment calculer la solution dans l'espace réduit ?

Solution

Construire un **modèle réduit** en petite dimension dont $\hat{\mathbf{u}}(\mu)$ ou $\hat{\mathbf{u}}(t, \mu)$ sont solutions.

Offline/Online

La construction du modèle peut être coûteuse et est faite une seule fois. On parle de la phase **offline**. Le modèle réduit peut être résolu un nombre important de fois. On parle de phase **online**. La phase **online** doit être très peu coûteuse.

Modèles réduits et projection

Projection de Galerkin pour les EDP elliptiques

Projection de Galerkin pour les EDP temporelles

Hyper-réduction et fermeture pour la réduction nonlinéaire

Modèles réduits et apprentissage d'EDO

Hyper-réduction et fermeture

Méthode de projection I

Idée

Comme pour les méthodes numérique on **restreint/projette les équations sur l'espace d'approximation**.

- On considère

$$\begin{cases} r(x)u(x) + a\partial_x u - d\partial_{xx}u = f(x) \\ u(x=0) = u_l, \quad u(x=L) = u_r \end{cases}$$

- Après discrétisation spatial le problème se réécrit

$$\mathbf{r}(\mathbf{v}, \mathbf{r}_h, \mathbf{f}_h, \boldsymbol{\mu}) := \mathbf{R}_h \mathbf{v} + aD_h \mathbf{v} - dD_{hh} \mathbf{v} - \mathbf{f}_h := L_{R,a,\mu} \mathbf{v} - \mathbf{f}_h = 0$$

avec $\mathbf{r} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathcal{D} \rightarrow \mathbb{R}^n$ correspond au résidu de notre problème en grande dimension.

- On définit notre espace d'approximation **Linéaire**:

$$V_K = \{ \tilde{\mathbf{u}}(\boldsymbol{\mu}) = \mathbf{u}_{ref} + \Phi \hat{\mathbf{u}}(\boldsymbol{\mu}), \quad \hat{\mathbf{u}}(\boldsymbol{\mu}) \in \mathbb{R}^K \}$$

Projection de Galerkin moindre carrés

Il s'agit de la solution du problème:

$$\hat{\mathbf{u}} = \underset{\tilde{\mathbf{v}} \in V_K}{\operatorname{argmin}} \| \mathbf{r}(\tilde{\mathbf{v}}, \mathbf{R}_h, \mathbf{f}_h, \boldsymbol{\mu}) \|_2^2 = \underset{\hat{\mathbf{v}} \in \mathbb{R}^K}{\operatorname{argmin}} \| \mathbf{r}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{v}}, \mathbf{R}_h, \mathbf{f}_h, \boldsymbol{\mu}) \|_2^2$$

Méthode de projection II

Solution de LS Galerkin

La solution de la **minimisation du résidu sur V_n** est donnée par:

$$(L_{R,a,\mu} \Phi_K)^t (L_{R,a,\mu} \Phi_K) \hat{\mathbf{u}} = (L_{R,a,\mu} \Phi_K)^t (\mathbf{f}_h - L_{R,a,\mu} \mathbf{u}_{ref})$$

- **Démonstration:**

- ▶ On cherche à minimiser le problème:

$$h(\hat{\mathbf{v}}) = \| L_{R,a,\mu} \Phi_K \hat{\mathbf{v}} - (\mathbf{f}_h - L_{R,a,\mu} (\mathbf{u}_{ref})) \|_2^2$$

- ▶ On développe:

$$h(\hat{\mathbf{v}}) = \langle L_{R,a,\mu} \Phi_K \hat{\mathbf{v}}, L_{R,a,\mu} \Phi_K \hat{\mathbf{v}} \rangle - 2 \langle L_{R,a,\mu} \Phi_K \hat{\mathbf{v}}, \mathbf{f}_h - L_{R,a,\mu} \mathbf{u}_{ref} \rangle + \| \mathbf{f}_h - L_{R,a,\mu} \mathbf{u}_{ref} \|_2^2$$

- ▶ On calcule le gradient

$$\frac{\partial h(\hat{\mathbf{v}})}{\partial \hat{\mathbf{v}}} = 2 \Phi_K^t L_{R,a,\mu}^t L_{R,a,\mu} \Phi_K \hat{\mathbf{v}} - 2 \Phi_K^t L_{R,a,\mu}^t (\mathbf{f}_h - L_{R,a,\mu} \mathbf{u}_{ref})$$

- ▶ On résout maintenant $\frac{\partial h(\hat{\mathbf{v}}^*)}{\partial \hat{\mathbf{v}}^*} = 0$. On obtient donc

$$2(L_{R,a,\mu} \Phi_K)^t L_{R,a,\mu} \Phi_K \hat{\mathbf{v}}^* - 2(L_{R,a,\mu} \Phi_K)^t (\mathbf{f}_h - L_{R,a,\mu} (\mathbf{u}_{ref})) = 0$$

donc:

$$(L_{R,a,\mu} \Phi_K)^t (L_{R,a,\mu} \Phi_K) \hat{\mathbf{v}}^* = (L_{R,a,\mu} \Phi_K)^t (\mathbf{f}_h - L_{R,a,\mu} (\mathbf{u}_{ref}))$$

- Si la dépendance aux paramètres est affine on peut pré-calculer les matrices. Dans le cas général on aura besoin d'hyper-réduction (cours ROM).

Méthode de projection III

- Le schéma peut réécrit sous forme énergétique;

$$\min_{\mathbf{u}} \mathbf{r}_V(\mathbf{u}, \mathbf{R}_h, f_h, \mu) := \min_{\mathbf{u}} \langle L_{R,a,\mu} \mathbf{u}, \mathbf{u} \rangle - \langle \mathbf{f}_h, \mathbf{u} \rangle$$

Solution de Galerkin

La solution de la **minimisation du résidu sur V_K** est donnée par:

$$\hat{\mathbf{u}} = \min_{\tilde{\mathbf{v}} \in V_K} (\langle L_{R,a,\mu} \tilde{\mathbf{v}}, \tilde{\mathbf{v}} \rangle - \langle \mathbf{f}_h, \tilde{\mathbf{v}} \rangle)$$

équivalent à

$$\hat{\mathbf{u}} = \min_{\hat{\mathbf{v}} \in V_K} (\langle L_{R,a,\mu} (\mathbf{u}_{ref} + \Phi \hat{\mathbf{v}}(\mu)), (\mathbf{u}_{ref} + \Phi \hat{\mathbf{v}}(\mu)) \rangle - \langle \mathbf{f}_h, (\mathbf{u}_{ref} + \Phi \hat{\mathbf{v}}(\mu)) \rangle)$$

- On retrouve la forme des élément finis standard.
- On est entrain d'appliquer les méthode de Ritz Galerkin et Galerkin moindre carré à l'espace réduit.

Solution de Galerkin

La solution de la **minimisation de l'énergie sur V_K** est donnée par:

$$(\Phi_K^t L_{R,a,\mu} \Phi_K) \hat{\mathbf{u}} = \Phi_K^t (\mathbf{f}_h - L_{R,a,\mu} \mathbf{u}_{ref})$$

- **Démonstration:**

- ▶ On cherche à minimiser le problème:

$$h(\hat{\mathbf{v}}) = \langle L_{R,a,\mu} (\Phi_K \hat{\mathbf{v}} + \mathbf{u}_{ref}), \Phi_K \hat{\mathbf{v}} \rangle - \langle \mathbf{f}_h, \Phi_K \hat{\mathbf{v}} \rangle$$

- ▶ On réécrit cela:

$$h(\hat{\mathbf{v}}) = \langle L_{R,a,\mu} \Phi_K \hat{\mathbf{v}}, \Phi_K \hat{\mathbf{v}} \rangle - \langle \mathbf{f}_h - L_{R,a,\mu} \mathbf{u}_{ref}, \Phi_K \hat{\mathbf{v}} \rangle$$

$$h(\hat{\mathbf{v}}) = \langle \Phi_K^t L_{R,a,\mu} \Phi_K \hat{\mathbf{v}}, \hat{\mathbf{v}} \rangle - \langle \Phi_K^t (\mathbf{f}_h - L_{R,a,\mu} \mathbf{u}_{ref}), \hat{\mathbf{v}} \rangle$$

On calcule le gradient qui donne

$$\frac{\partial h(\hat{\mathbf{v}}^*)}{\partial \hat{\mathbf{v}}^*} = \Phi_K^t L_{R,a,\mu} \Phi_K \hat{\mathbf{v}}^* - \Phi_K^t (\mathbf{f}_h - L_{R,a,\mu} \mathbf{u}_{ref})$$

- ▶ En cherchant à l'annuler on conclut la preuve.

Méthode de projection pour décodeur non linéaire

- Dans le cadre non linéaire on peut aussi proposer une projection de Galerkin et Galerkin moindre carrés. On va détailler le cas de la **Galerkin moindre carrés**.

Projection de Galerkin moindre carrés

On se fixe l'espace d'approximation **non linéaire**:

$$W_K = \{ \tilde{\mathbf{u}}(\boldsymbol{\mu}) = (\hat{\mathbf{u}}(\boldsymbol{\mu})), \quad \hat{\mathbf{u}}(\boldsymbol{\mu}) \in \mathbb{R}^K \}$$

Il s'agit de la solution du problème:

$$\hat{\mathbf{u}} = \underset{\tilde{\mathbf{v}} \in W_K}{\operatorname{argmin}} \| \mathbf{r}(\tilde{\mathbf{v}}, \mathbf{R}_h, \mathbf{f}_h, \boldsymbol{\mu}) \|_2^2 = \underset{\hat{\mathbf{v}} \in \mathbb{R}^K}{\operatorname{argmin}} \| \mathbf{r}(\mathbf{G}(\hat{\mathbf{v}}), \mathbf{R}_h, \mathbf{f}_h, \boldsymbol{\mu}) \|_2^2$$

Solution de LS Galerkin dans le cas non linéaire

La solution de la **minimisation du résidu sur W_K** est calculé par un algorithme de Levenberg–Marquardt de la forme:

$$\hat{\mathbf{u}}_{k+1} = \hat{\mathbf{u}}_k + \delta \hat{\mathbf{u}}$$

avec

$$((L_{R,a,\boldsymbol{\mu}}(\partial_{\hat{\mathbf{u}}_k} \mathbf{G}))^t (L_{R,a,\boldsymbol{\mu}}(\partial_{\hat{\mathbf{u}}_k} \mathbf{G})) + \lambda I_d) \delta \hat{\mathbf{u}} = (L_{R,a,\boldsymbol{\mu}}(\partial_{\hat{\mathbf{u}}_k} \mathbf{G}))^t \mathbf{f}_h$$

- L'encodeur peut permettre d'encoder une condition initiale physique.

Modèles réduits et projection

Projection de Galerkin pour les EDP elliptiques

Projection de Galerkin pour les EDP temporelles

Hyper-réduction et fermeture pour la réduction nonlinéaire

Modèles réduits et apprentissage d'EDO

Hyper-réduction et fermeture

Méthode de projection I

- On peut aussi procéder a une projection de Galerkin dans le cas temporel (Ex Neural Galerkin).

- On va considérer ici:

$$\begin{cases} \partial_t \rho + \partial_x f(\rho) = \frac{1}{Re} \partial_{xx} \rho \\ \rho(0, x) = \rho(x, \alpha) \\ \rho(t, x = 0) = r_l, \quad \rho(t, x = L) = r_r \end{cases}$$

- Comme précédemment on introduit le résidu:

$$\mathbf{r}\left(\frac{d\mathbf{u}}{dt}, \mathbf{u}, t; \boldsymbol{\mu}\right) := \frac{d\mathbf{u}}{dt} - \mathbf{f}(\mathbf{x}, t; \boldsymbol{\mu}) = 0, \quad \mathbf{u}(0; \boldsymbol{\mu}) = \mathbf{u}_0(\boldsymbol{\mu})$$

Projection de Galerkin

La **projection de Galerkin** est donnée par la solution du problème

$$\hat{\mathbf{u}} = \underset{\frac{d}{dt} \tilde{\mathbf{v}} \in V_K}{\operatorname{argmin}} \left\| \mathbf{r}\left(\frac{d}{dt} \tilde{\mathbf{v}}, \tilde{\mathbf{v}}, t; \boldsymbol{\mu}\right) \right\|_2^2$$

Ce problème est équivalent à

$$\hat{\mathbf{u}} = \underset{\hat{\mathbf{v}} \in \mathbb{R}^K}{\operatorname{argmin}} \left\| \mathbf{r}\left(\Phi_K \frac{d}{dt} \hat{\mathbf{v}}, \tilde{\mathbf{v}}, t; \boldsymbol{\mu}\right) \right\|_2^2$$

avec

$$\hat{\mathbf{u}}(0; \boldsymbol{\mu}) = \hat{\mathbf{u}}_0(\boldsymbol{\mu}) = \Phi_K^T (\mathbf{u}_0(\boldsymbol{\mu}) - \mathbf{u}_{ref}).$$

Méthode de projection II

Solution de l'approche de LS Galerkin

La solution du problème de minimisation précédent est donnée par:

$$\frac{d\hat{\mathbf{u}}}{dt} = \Phi_K^T \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}}, t; \boldsymbol{\mu}), \quad \hat{\mathbf{u}}(0; \boldsymbol{\mu}) = \hat{\mathbf{u}}_0(\boldsymbol{\mu})$$

- On cherche à minimiser le problème:

$$h(\hat{\mathbf{v}}) = \| \Phi_K \hat{\mathbf{v}} - \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{v}}, t; \boldsymbol{\mu}) \|_2^2$$

- On développe:

$$h(\hat{\mathbf{v}}) = \hat{\mathbf{v}}^T \Phi_K^T \Phi_K \hat{\mathbf{v}} - 2\hat{\mathbf{v}}^T \Phi_K^T \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{v}}, t; \boldsymbol{\mu}) + \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{v}}, t; \boldsymbol{\mu})^T \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{v}}, t; \boldsymbol{\mu})$$

- On calcule le gradient

$$\frac{\partial h(\hat{\mathbf{v}})}{\partial \hat{\mathbf{v}}} = 2\Phi_K^t \Phi_K \hat{\mathbf{v}} - 2\Phi_K^t \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{v}}, t; \boldsymbol{\mu})$$

- On résout maintenant $\frac{\partial h(\hat{\mathbf{v}}^*)}{\partial \hat{\mathbf{v}}^*} = 0$. Pour cela on utilise le fait que Φ est orthogonale, ce qui donne comme solution

$$0 = \hat{\mathbf{v}}^* - \Phi_K^t \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{v}}^*, t; \boldsymbol{\mu})$$

- Il existe des variantes où on minimise le résidu du schéma. On parle de méthode de Petrov-Galerkin.

Méthode de projection pour décodeur non linéaire

- On peut procéder de façon similaire pour un **espace réduit non linéaire**.

Méthode de Galerkin non linéaire

La projection de Galerkin non linéaire est donnée par la solution du problème

$$\hat{\mathbf{u}} = \operatorname{argmin}_{\tilde{\mathbf{v}} \in W_K} \|\mathbf{r}(\tilde{\mathbf{v}}, \tilde{\mathbf{u}}, t; \boldsymbol{\mu})\|_2^2$$

qui est équivalent à:

$$\hat{\mathbf{u}} = \operatorname{argmin}_{\hat{\mathbf{v}} \in \mathbb{R}^K} \left\| \mathbf{r} \left(J_g(\hat{\mathbf{u}}) \frac{d\hat{\mathbf{u}}}{dt}, \mathbf{u}_{ref} + G(\hat{\mathbf{u}}), t; \boldsymbol{\mu} \right) \right\|_2^2$$

avec J_g la Jacobienne de G et

$$W_K = \{ \tilde{\mathbf{v}}(t, \boldsymbol{\mu}) = \mathbf{u}_{ref} + G(\hat{\mathbf{v}}(t, \boldsymbol{\mu})), \quad \hat{\mathbf{v}} \in \mathbb{R}^K \}$$

avec $\hat{\mathbf{u}}(0; \boldsymbol{\mu}) = E(\mathbf{u}_0(\boldsymbol{\mu}))$.

Méthode de projection pour décodeur non linéaire II

Solution de Galerkin nonlinéaire

La solution de l'équation de la projection de Galerkin nonlinéaire est donnée par:

$$\frac{d\hat{\mathbf{u}}}{dt} = J_g(\hat{\mathbf{u}})^+ \mathbf{f}(\mathbf{u}_{ref} + G(\hat{\mathbf{u}}), t; \mu), \quad \hat{\mathbf{u}}(0; \mu) = \hat{\mathbf{u}}_0(\mu)$$

avec le pseudo-inverse $J_g(\hat{\mathbf{u}})^+ = (J_g(\hat{\mathbf{u}})^t J_g(\hat{\mathbf{u}}))^{-1} J_g(\hat{\mathbf{u}})^t$.

- On cherche à minimiser le problème:

$$h(\hat{\mathbf{v}}) = \| J_g(\hat{\mathbf{v}})\hat{\mathbf{v}} - \mathbf{f}(\mathbf{u}_{ref} + G(\hat{\mathbf{v}}), t; \mu) \|_2^2$$

- On développe:

$$h(\hat{\mathbf{v}}) = \hat{\mathbf{v}}^T J_g(\hat{\mathbf{v}})^T J_g(\hat{\mathbf{v}})\hat{\mathbf{v}} - 2\hat{\mathbf{v}}^T J_g(\hat{\mathbf{v}})^T \mathbf{f}(\mathbf{u}_{ref} + G(\hat{\mathbf{v}}), t; \mu) + \mathbf{f}(\mathbf{u}_{ref} + G(\hat{\mathbf{v}}), t; \mu)^T \mathbf{f}(\mathbf{u}_{ref} + G(\hat{\mathbf{v}}), t; \mu)$$

- On calcule le gradient

$$\frac{\partial h(\hat{\mathbf{v}})}{\partial \hat{\mathbf{v}}} = 2J_g(\hat{\mathbf{v}})^T J_g(\hat{\mathbf{v}})\hat{\mathbf{v}} - 2J_g(\hat{\mathbf{v}})^T \mathbf{f}(\mathbf{u}_{ref} + G(\hat{\mathbf{v}}), t; \mu)$$

- On résout maintenant $\frac{\partial h(\hat{\mathbf{v}}^*)}{\partial \hat{\mathbf{v}}^*} = 0$. Pour cela on inverse $J_g(\hat{\mathbf{v}}^*)^T J_g(\hat{\mathbf{v}}^*)$ ce qui donne

$$0 = \hat{\mathbf{v}}^* - (J_g(\hat{\mathbf{v}}^*)^T J_g(\hat{\mathbf{v}}^*))^{-1} J_g(\hat{\mathbf{v}}^*)^T \mathbf{f}(\mathbf{u}_{ref} + G(\hat{\mathbf{v}}^*), t; \mu)$$

ce qui conclut la preuve.

Modèles réduits et projection

Projection de Galerkin pour les EDP elliptiques

Projection de Galerkin pour les EDP temporelles

Hyper-réduction et fermeture pour la réduction nonlinéaire

Modèles réduits et apprentissage d'EDO

Hyper-réduction et fermeture

La projection de Galerkin est t'elle suffisante ?

- Suite a notre réduction de modèle on doit résoudre

$$\frac{d\hat{\mathbf{u}}}{dt} = \Phi_K^T \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}}, t; \mu)$$

ou

$$\frac{d\hat{\mathbf{u}}}{dt} = J_g(\hat{\mathbf{u}})^T \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}}, t; \mu), \quad \hat{\mathbf{u}}(0; \mu) = \hat{\mathbf{u}}_0(\mu)$$

- Pour évaluer le flux de l'équation on doit **repasser par la grande dimension**. Le cout va donc rester trop important. En effet: $\Phi_K \hat{\mathbf{u}} \approx O(nK)$, $\Phi_K^T \mathbf{f} \approx O(nK)$ et $\mathbf{f}(\tilde{\mathbf{u}}) \approx O(n^2)$.
- **Cas particulier :**

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{A}\mathbf{u}(t) + \mathbf{B}(\mathbf{u} \otimes \mathbf{u}) + \mathbf{C}(\mathbf{u} \otimes \mathbf{u} \otimes \mathbf{u})$$

avec des matrices $\mathbf{A} \in \mathcal{M}_{n,n}$, $\mathbf{B} \in \mathcal{M}_{n,n^2}$ et $\mathbf{C} \in \mathcal{M}_{n,n^3}$. Dans le cas le modèle réduit est donné par

$$\frac{d\hat{\mathbf{u}}(t)}{dt} = \hat{\mathbf{A}}\hat{\mathbf{u}} + \hat{\mathbf{B}}(\hat{\mathbf{u}} \otimes \hat{\mathbf{u}}) + \hat{\mathbf{C}}(\hat{\mathbf{u}} \otimes \hat{\mathbf{u}} \otimes \hat{\mathbf{u}})$$

avec $\hat{\mathbf{A}} = \Phi_K^T \mathbf{A} \Phi_K$, $\hat{\mathbf{B}} = \Phi_K^T \mathbf{B} (\Phi_K \otimes \Phi_K)$ et $\hat{\mathbf{C}} = \Phi_K^T \mathbf{C} (\Phi_K \otimes \Phi_K \otimes \Phi_K)$

Gappy POD I

Hyper-réduction

Il s'agit d'une seconde phase pour obtenir un modèle réduit dont le coût d'évaluation est en $O(K^p)$.

- On va traiter d'abord le cas de la réduction linéaire.

Gappy POD

La méthode Gappy-POD est la première approche pour estimer une nonlinéarité en dimension réduite en évitant un calcul coûteux

- On commence par construire une matrice de **Snapshots nonlinéarité**:

$$X_f = [\mathbf{f}(\mathbf{u}(t_0, \boldsymbol{\mu}_1)), \dots, \mathbf{f}(\mathbf{u}(t_{N_t}, \boldsymbol{\mu}_{N_t}))]$$

- On peut appliquer une POD à ces nouveaux snapshots qui est solution du problème de régression

$$\min_{\Phi_K^f \in \mathbb{R}^{n \times k}} \left\| X_f - \Phi_K^f \Phi_K^{t,f} X_f \right\|_F^2, \quad \text{sous contrainte } \Phi_K^{t,f} \Phi_K^f = I_d$$

Gappy POD II

- Maintenant on va donner **une matrice d'échantillonnage** $Z = [e_{i_1}, \dots, e_{i_l}] \in \mathcal{M}_{n,l}(\mathbb{R})$. Cette matrice a pour rôle de sélectionner l indices du vecteur qu'elle multiplie.
- L'idée de la **méthode Gappy-POD** est de trouver la représentation dans l'espace réduit d'un vecteur \mathbf{f}_x qui minimise l'erreur de reconstruction seulement sur certains indices. On a:

$$\hat{\mathbf{f}} = \operatorname{argmin}_{\mathbf{a} \in \mathbb{R}^l} \| Z^t (\Phi_K^f \mathbf{a} - \mathbf{f}_x) \|_2^2$$

- Il s'agit d'un problème de type moindre carré donc la solution est donnée par:

$$(Z^t \Phi_K^f)^t (Z^t \Phi_K^f) \hat{\mathbf{f}} = (Z^t \Phi_K^f)^t Z^t \mathbf{f}_x$$

$$(\Phi_K^{f,t} Z Z^t \Phi_K^f) \hat{\mathbf{f}} = \Phi_K^{f,t} Z Z^t \mathbf{f}_x$$

- La nonlinearité réduite ici est donc estimée

$$\hat{\mathbf{f}} = (\Phi_K^{f,t} Z Z^t \Phi_K^f)^{-1} \Phi_K^{f,t} Z Z^t \mathbf{f}_x$$

- En grande dimension on obtient l'approximation

$$\tilde{\mathbf{f}} = \Phi_K^t (\Phi_K^{f,t} Z Z^t \Phi_K^f)^{-1} \Phi_K^{f,t} Z Z^t \mathbf{f}_x$$

- On obtient donc la formule d'hyper réduction qui estime la nonlinearité de la méthode de Galerkin.

Hyperduction Gappy-POD

Le vecteur nonlinéaire de dimension K du modèle réduit est estimé par le terme:

$$\Phi_K^t \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}}) \approx \Phi_K^t \left(Z^t \Phi_K^f \right)^+ Z^t \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}})$$

avec $A^+ = (A^t A)^{-1} A$ le pseudo-inverse de Penrose.

- Il reste une dépendance en $O(n)$ à l'intérieur de la nonlinéarité.
- On peut, à ce moment-là, différencier plusieurs cas:
 - ▶ Le cas où $\mathbf{f}(\mathbf{u}) = (f(u_1), \dots, f(u_n))^t$. C'est typiquement le genre de non-linéarité qu'on obtient lorsqu'on discrétise **une source nonlinéaire**. Ici on peut remarquer que si on applique une méthode POD Gappy à \mathbf{f} , on a:

$$Z^t \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}}) = \mathbf{u}(Z^t (\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}}))$$

On a donc, dans ce cas, une complexité en $O(K)$ pour estimer la non-linéarité.

- ▶ Le cas où $\mathbf{f}(\mathbf{u}) = B \mathbf{f}_l$ avec $\mathbf{f}_l(\mathbf{u}) = (f_l(u_1), \dots, f_l(u_n))^t$. **Exemple:** une **discrétisation centrée** d'un terme du type $\partial_x^p g(\rho)$. Dans ce cas on réécrit la réduction sous la forme:

$$\Phi_k^t B \mathbf{f}_{loc}(\mathbf{u}_{ref} + \Phi_k \hat{\mathbf{u}}(t), \boldsymbol{\mu}) = \hat{B} \Phi_k^t \mathbf{f}_{loc}(\mathbf{u}_{ref} + \Phi_k \hat{\mathbf{u}}(t), \boldsymbol{\mu})$$

avec $\hat{B} = \Phi_k^t B \Phi_k$ et on applique la stratégie du cas précédent à \mathbf{f}_{loc} . On a dans ce cas une estimation de la nonlinéarité en $O(K^3)$.

- ▶ **Cas plus général:** On effectue que les calculs qui seront utile pour les lignes sélectionnées par Z .

DEIM

L'approche est similaire. Au lieu d'imposer par **régression la valeur aux points sélectionnés on va le faire par interpolation.**

- Cela veut dire que pour certains points (représentée par Z que une non-linéarité approchée $\tilde{\mathbf{f}}(\mathbf{u})$:

$$Z^t \tilde{\mathbf{f}}(\mathbf{u}) = Z^t \mathbf{f}(\mathbf{u})$$

- On applique une POD au snapshots de la non-linéarité qui donne Φ_k^f . On a donc

$$Z^t \Phi_k^f \hat{\mathbf{f}}(\mathbf{u}) = Z^t \mathbf{f}(\mathbf{u})$$

- avec $l = K$ le nombre de points d'interpolation. Si on suppose que la matrice est inversible. On obtient

$$\hat{\mathbf{f}} = (Z^t \Phi_K^f)^{-1} Z^t \mathbf{f}(\mathbf{u})$$

et donc

$$\tilde{\mathbf{f}} = \Phi_k^f (Z^t \Phi_K^f)^{-1} Z^t \mathbf{f}(\mathbf{u})$$

- On va maintenant appliquer cela à notre non-linéarité $\Phi_K^t \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}})$.

Hyper-réduction DEIM

Le vecteur non linéaire de dimension K du modèle est estimé par

$$\Phi_K^t \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}}) \approx \Phi_K^t \Pi_o \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}})$$

avec $Z \in \mathcal{M}_{K,l}(\mathbb{R})$, $\Pi_o = \Phi_K^f (Z^t \Phi_K^f)^{-1} Z^t$ la matrice qu'on appelle le **le projecteur oblique**. Le terme $\Phi_K^t \Phi_K^f (Z^t \Phi_K^f)^{-1}$ est pré-calculé et stocké.

- **Idée pour construire Z** : On va faire une approche **gloutonne**. Le premier indice sera choisi pour minimiser l'erreur de reconstruction du premier mode de Φ_f puis le second indice pour reconstruire au mieux le second mode, etc.

- **Algorithme:**

- ▶ On initialise la base : $\Phi^f = [\phi_1]$.
- ▶ On initialise la matrice des indices: $Z = [e_{i_1}]$ avec $i_1 = \operatorname{argmax}_{i \in \{1, \dots, n\}} |(\phi_1)_i|$
- ▶ Pour $l = 2$ et $l \leq K$:

- On calcul l'interpolation du vecteur ϕ_l en résolvant

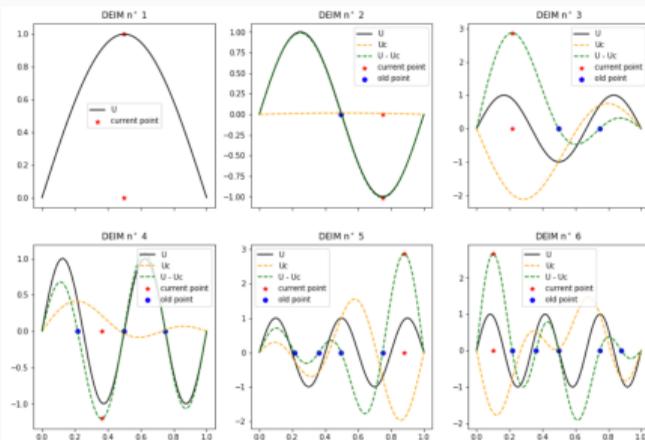
$$Z^t \Phi^f \mathbf{s} = Z^t \phi_l$$

- On calcul le résidu sur l'interpolation de ϕ_l

$$i_l = \operatorname{argmax}_{i \in \{1, \dots, l\}} |\phi_l^i - \Phi^f \mathbf{s}|$$

- ▶ On a $Z = [e_{i_1}, \dots, e_{i_l}]$ et on incrémente la base $\Phi^f = [\phi_1, \dots, \phi_l]$

- Exemple de DEIM



DEIM général

Soit une fonction $\mathbf{f} : V \in \mathbb{R}^d \rightarrow \mathbb{R}^n$. On se donne des vecteurs de \mathbb{R}^n $\Phi = (\phi, \dots, \phi_K)$ linéairement indépendant. Pour un point $\mathbf{u} \in V$ l'approximation DEIM est donnée par

$$\mathbf{f}_a(\mathbf{x}) = \Pi_o \mathbf{f}(\mathbf{x}), \quad \Pi_o = \Phi(Z^t \Phi)^{-1} Z^t$$

et $Z = (\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_K})$. On a

$$\|\mathbf{f} - \mathbf{f}_a\|_2 \leq \| (Z^t \Phi)^{-1} \|_2 \| (I_d - \Phi \Phi^t) \mathbf{f} \|_2 \leq \frac{(1 + 2n)^{K-1}}{\|\Phi_1\|_\infty} \| (I_d - \Phi \Phi^t) \mathbf{f} \|_2$$

Fermeture I

- **POD**: dans ces régimes nonlinéaires la décroissance des valeurs propres est lente. En prenant peu de modes on tronque donc une partie de la physique non négligeable. On se trouve dans le régime dit **sous-résolu**.
- **Exemple**: On suppose que nos données s'expliquent avec 3 modes.

$$\mathbf{x} \approx \hat{\chi}_1(t)\boldsymbol{\phi}_1 + \hat{\chi}_2(t)\boldsymbol{\phi}_2 + \hat{\chi}_3(t)\boldsymbol{\phi}_3$$

avec un modèle qui décrit la propagation en temps qui est obtenu par projection de Galerkin:

$$\begin{cases} \frac{d\hat{\chi}_1(t)}{dt} = F_1(\hat{\chi}_1, \hat{\chi}_2, \hat{\chi}_3) \\ \frac{d\hat{\chi}_2(t)}{dt} = F_2(\hat{\chi}_1, \hat{\chi}_2, \hat{\chi}_3) \\ \frac{d\hat{\chi}_3(t)}{dt} = F_3(\hat{\chi}_1, \hat{\chi}_2, \hat{\chi}_3) \end{cases}$$

- Maintenant on tronque en gardant 2 modes. Le modèle le plus complet pour les décrire est tout naturellement

$$\begin{cases} \frac{d\hat{\chi}_1(t)}{dt} = F_1(\hat{\chi}_1, \hat{\chi}_2, \hat{\chi}_3) \\ \frac{d\hat{\chi}_2(t)}{dt} = F_2(\hat{\chi}_1, \hat{\chi}_2, \hat{\chi}_3) \end{cases}$$

Il est même exact si on connaît $\hat{\chi}_3(t)$. Pour le moment le modèle n'est pas résoluble, car on ne connaît pas $\hat{\chi}_3(t)$.

Notion de fermeture

Une fermeture pour le modèle précédent est une relation du type

$$\hat{x}_3(t) \approx \tau(\hat{x}_1(t), \hat{x}_2(t))$$

- Une projection de Galerkin revient à

$$\begin{cases} \frac{\hat{x}_1(t)}{dt} = F_1(\hat{x}_1, \hat{x}_2, 0) \\ \frac{\hat{x}_2(t)}{dt} = F_2(\hat{x}_1, \hat{x}_2, 0) \end{cases}$$

- Soit le modèle complet

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{f}(\mathbf{u}(t))$$

- et que une méthode de projection donne le modèle réduit:

$$\frac{d\hat{\mathbf{u}}(t)}{dt} = \hat{\mathbf{f}}(\hat{\mathbf{u}}(t))$$

- La **construction d'une fermeture revient à chercher** $\tau()$ tel que le modèle suivant

$$\frac{d\hat{\mathbf{u}}_c(t)}{dt} = \hat{\mathbf{f}}(\hat{\mathbf{u}}_c(t)) + \tau(\hat{\mathbf{u}}_c(t))$$

soit plus précis que le modèle réduit.

Fermeture III

Fermeture idéale

Soit le modèle $\frac{d\mathbf{u}(t)}{dt} = \mathbf{f}(\mathbf{u}(t))$. Soit la projection de Galerkin:

$$\frac{d\hat{\mathbf{u}}(t)}{dt} = \Phi_K^t \mathbf{f}(\Phi_K \hat{\mathbf{u}}(t))$$

Alors une **fermeture** est un terme $\tau(\hat{\mathbf{u}}(t))$ qu'on ajoute pour obtenir l'équation suivante

$$\frac{d\hat{\mathbf{u}}(t)}{dt} = \Phi_K^t \mathbf{f}(\Phi_K \hat{\mathbf{u}}(t)) + \tau(\hat{\mathbf{u}}(t))$$

Elle est dite **idéale** si

$$\tau(\hat{\mathbf{u}}(t)) = \Phi_K^t (\mathbf{f}(\mathbf{u}(t)) - \mathbf{f}(\Phi_K \hat{\mathbf{u}}(t)))$$

- Pour l'équation de Burgers le modèle réduit fermé s'écrit:

$$\frac{d\hat{\mathbf{u}}(t)}{dt} = \hat{B}(\hat{\mathbf{u}} \otimes \hat{\mathbf{x}}) + \frac{1}{Re} \hat{L}\hat{\mathbf{u}} + C(\hat{\mathbf{u}})$$

- La **fermeture consiste souvent à ajouter de la viscosité** $C(\mathbf{u}) = \nu D\hat{L}\mathbf{u}$. Exemples:

$$D_{kk} = \left(\frac{k}{K}\right)^p, \quad D_{kk} = \chi_{k>M}$$

avec k le numéro du mode.

Modèles réduits et projection

Projection de Galerkin pour les EDP elliptiques

Projection de Galerkin pour les EDP temporelles

Hyper-réduction et fermeture pour la réduction nonlinéaire

Modèles réduits et apprentissage d'EDO

Hyper-réduction et fermeture

Modèles réduits et apprentissage d'EDO

Modèles réduits et projection

Projection de Galerkin pour les EDP elliptiques

Projection de Galerkin pour les EDP temporelles

Hyper-réduction et fermeture pour la réduction nonlinéaire

Modèles réduits et apprentissage d'EDO

Hyper-réduction et fermeture

Hyper-réduction et apprentissage

- Pour la réduction nonlinéaire on peut aussi construire une hyper-réduction de type Gappy. Il s'agit de construction difficile qu'on ne traitera pas ici.
- Autre solution: **L'apprentissage**.
- On va ici considérer une réduction nonlinéaire. Le cas linéaire s'obtient avec $G(\hat{\mathbf{u}}) = \Phi_K \hat{\mathbf{u}}$

Hyper-réduction par apprentissage

On considère le modèle

$$\frac{d\hat{\mathbf{u}}}{dt} = J_g(\hat{\mathbf{u}})^+ \mathbf{f}(\mathbf{u}_{ref} + G(\hat{\mathbf{u}}), \mathbf{t}; \boldsymbol{\mu}),$$

On appelle **hyper-réduction par apprentissage de modèle** la résolution suivante

$$\min_{\theta} \sum_{i=1}^m \| J_g(\hat{\mathbf{u}}_i)^+ \mathbf{f}(\mathbf{u}_{ref} + G(\hat{\mathbf{u}}_i), \mathbf{t}_i; \boldsymbol{\mu}_i) - \hat{\mathbf{f}}_{\theta}(\hat{\mathbf{u}}_i) \|_2^2$$

avec $\hat{\mathbf{u}}_i = E(\mathbf{u}_i)$ et E l'encodeur associé.

- On peut utiliser des modèles paramétriques comme des réseaux de neurones ou des modèles de type Sindy.

Fermeture et apprentissage

- On peut construire la fermeture par apprentissage. En pratique c'est utile en cas de réduction linéaire. **Les méthodes de réduction nonlinéaire évite en général ses difficultés.**

Fermeture par apprentissage de modèle

. On considère le modèle

$$\frac{d\hat{\mathbf{u}}}{dt} = \Phi_K^t \mathbf{f}(\mathbf{u}_{ref} + \Phi_K(\hat{\mathbf{u}}), t; \boldsymbol{\mu}),$$

On appelle **hyper-réduction par apprentissage de modèle** la résolution suivante

$$\min_{\theta} \sum_{i=1}^m \|\Phi_K^t(\mathbf{f}(\mathbf{u}_i) - \mathbf{f}(\mathbf{u}_{ref} + \Phi_K \hat{\mathbf{u}}_i)) - \tau(\hat{\mathbf{u}}_i)\|_2^2$$

avec $\hat{\mathbf{u}}_i = E(\mathbf{u}_i)$ et E l'encodeur associé.

Fermeture par apprentissage de trajectoire

On appelle **hyper-réduction par apprentissage de trajectoires** la résolution suivante

$$\min_{\theta} \sum_{i=1}^m \|\Phi^t \mathbf{u}^{i+k} - \hat{\mathbf{u}}^{i+k}\|_2^2$$

sous la contrainte $\hat{\mathbf{u}}^{i+k} = S \circ \dots \circ S(\hat{\mathbf{u}}^i, \tau_{\theta})$ avec S le schéma. On retrouve l'apprentissage d'EDO

Apprentissage de modèles

- Plus généralement on peut apprendre le modèle que ce soit en linéaire ou nonlinéaire.
- On perd les garanties théoriques type DEIM mais on apprend un modèle potentiellement plus efficace (**il contient l'hyper-réduction et la fermeture**).

Modèle réduit par apprentissage de modèle

On appelle **réduction par apprentissage de modèle** la résolution suivante

$$\min_{\theta} \sum_{i=1}^m \left\| \left(\frac{d\hat{\mathbf{u}}}{dt} \right)_i - J_{\theta}(\hat{\mathbf{u}}_i) + f_{\theta}(\hat{\mathbf{u}}_i) \right\|_2^2$$

avec $\hat{\mathbf{u}}_i = E(\mathbf{u}_i)$, E l'encodeur associé et $(\frac{d\hat{\mathbf{u}}}{dt})_i$ estimée à partir des données.

Modèle réduit par apprentissage de trajectoire

On appelle **réduction par apprentissage de trajectoires** la résolution suivante

$$\min_{\theta} \sum_{i=1}^m \left\| E(\mathbf{u}^{i+k}) - \hat{\mathbf{u}}^{i+k} \right\|_2^2, \quad \text{ou} \quad \min_{\theta} \sum_{i=1}^m \left\| \mathbf{u}^{i+k} - D(\hat{\mathbf{u}}^{i+k}) \right\|_2^2$$

sous la contrainte $\hat{\mathbf{u}}^{i+k} = S_{\theta} \circ \dots \circ S_{\theta}(\hat{\mathbf{u}}^i)$ avec S le schéma. Apprentissage d'EDO par DpO.

- En basse dimension on peut apprendre l'EDO, le flot, par des réseaux, des modèles Sindy etc.

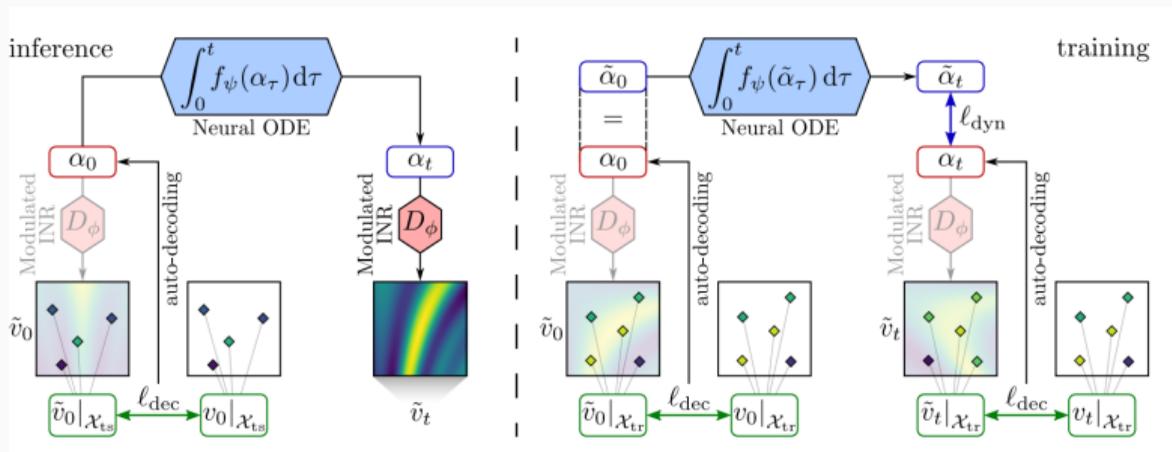
Apprentissage de bout en bout

- Pour plus de simplicité en cas de décodeur nonlinéaire on va souvent **apprendre le modèle**.

Entraînement "end-to-end"

consiste à effectuer l'apprentissage du modèle réduit, de l'encodeur et du décodeur en même temps. On peut mettre des fonctions de coût sur la partie AE, sur la dynamique latente ou sur l'ensemble

- Exemple d'entraînement global:



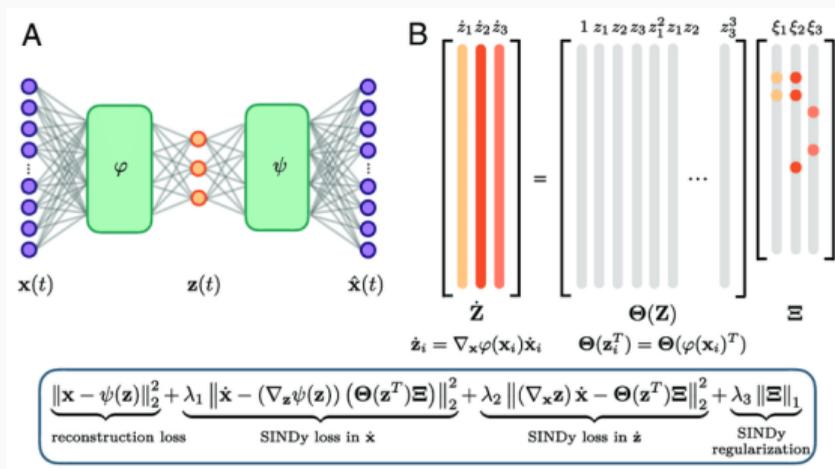
Apprentissage de bout en bout

- Pour plus de simplicité en cas de décodeur nonlinéaire on va souvent **apprendre le modèle**.

Entraînement "end-to-end"

consiste à effectuer l'apprentissage du modèle réduit, de l'encodeur et du décodeur en même temps. On peut mettre des fonctions de coût sur la partie AE, sur la dynamique latente ou sur l'ensemble

- Exemple d'entraînement global:



Apprentissage de bout en bout

- Pour plus de simplicité en cas de décodeur nonlinéaire on va souvent **apprendre le modèle**.

Entraînement "end-to-end"

consiste à effectuer l'apprentissage du modèle réduit, de l'encodeur et du décodeur en même temps. On peut mettre des fonctions de coût sur la partie AE, sur la dynamique latente ou sur l'ensemble

- Exemple d'entraînement global:

