Sensitivity analysis and non-intrusive two-grid reduced basis methods

IRMA - Séminaire Equations aux dérivées partielles

Elise Grosjean [1]

[1]ENSTA-Paris

ENSTA

INSTITUT POLYTECHNIQUE DE PARIS

Research activities

Development, analysis and application of Non-Intrusive Reduced Basis (NIRB) methods for the simulation of parametric PDEs

NIRB (Non-Intrusive Reduced Basis) method

- Several numerical analyses of the 2-grid method
  - FEM/FV context
  - Elliptic/Parabolic equations/Sensitivity analysis
- Development of new NIRB methods
- Non-intrusive implementation in a Python module [1] and development of a website on RBM [2]
- 3D Applications
  - Offshore wind turbines
  - Optimization of parameters for meniscus tissue

[1] https://gitlab.com/mor_dicus
[2] https://reducedbasis.github.io

Introduction

Optimization of parameters

- ◇ Parameter identification & Sensitivity analysis : How the parameters influence the model?
- ◇ NIRB 2-grid method[1] [2]
- ◇ NIRB 2-grid method with parabolic equations[3]
- ◇ NIRB methods in the context of sensitivity analysis[4]

---

[1] R. Chakir, Y. Maday, *A two-grid finite-element/reduced basis scheme for the approximation of the solution of parameter dependent PDE.* 2009.

[2] E. G., Y. Maday, *Error estimate of the non-intrusive reduced basis method with finite volume schemes.* 2021.

[3] E. G., Y. Maday, *Error estimate of the non-intrusive reduced basis two-grid method with parabolic equations.* 2023.

[4] E. G., B. Simeon, *The non-intrusive reduced basis two-grid method applied to sensitivity analysis.* 2024.

Parametric problem

Parameter identification & Sensitivity analysis

Parametric problem

IVP

$$\mathcal{P} : (\mathbf{u}^0, t, \mathbf{x}, \mathbf{p}) \to \mathbf{u}(t, \mathbf{x}; \mathbf{p}), \quad t \in [0, T]\; \mathbf{x} \in \Omega,\; \mathbf{p} \in \mathbb{R}^P.$$

Numerical solution

$$\mathcal{P}_h : (\mathbf{u}_h^0, t^k, \mathbf{x}, \mathbf{p}) \to \mathbf{u}_h^k(\mathbf{x}; \mathbf{p}), \quad k \in 0, \dots, N_T,\; \mathbf{x} \in \Omega,\; \mathbf{p} \in \mathbb{R}^P.$$

Measurements with true parameter $\overline{\mathbf{p}}$

$$\overline{\mathbf{u}}(t, \mathbf{x}), \quad t \in]0, T], \quad \mathbf{x} \in \Omega,$$

$$\overline{\mathbf{u}}(0, \mathbf{x}) = \overline{\mathbf{u}}^0(\mathbf{x}), \quad \mathbf{x} \in \Omega.$$

Parameter identification

$$\mathcal{F}(\mathbf{p}) = \frac{1}{2}\sum_{k=1}^{T/\Delta t} \underbrace{\|\mathbf{u}_h^k(\mathbf{p}) - \overline{\mathbf{u}}^k\|_{L^2}^2}_{\|\mathrm{err}(t^k;\mathbf{p})\|_{L^2}^2},$$

Parameter identification

$$\mathcal{F}(\mathbf{p}) = \frac{1}{2} \sum_{k=1}^{T/\Delta t} \underbrace{\|\mathbf{u}_h^k(\mathbf{p}) - \overline{\mathbf{u}}^k\|_{L^2}^2}_{\|\mathrm{err}(t^k;\mathbf{p})\|_{L^2}^2},$$

$$\Downarrow$$

Gradient descent

Sensitivities

$$\mathbf{S}_j^k(\mathbf{x};\mathbf{p}) := \partial_{p_j}\mathbf{u}_h^k(\mathbf{x};\mathbf{p}) \quad \text{or} \quad \partial_{p_j}\mathcal{F}(\mathbf{p})$$

Normalized sensitivity coefficients[1]

$$C_j = \partial_{p_j}\mathcal{F}(\mathbf{p}) \times \frac{p_j}{\mathcal{F}(\mathbf{p})}, \quad j = 1,\dots,P.$$

[1] E. Borgonovo, E. Plischke, *Sensitivity analysis: a review of recent advances.* 2016.

Sensitivity analysis

$$\mathcal{P}_{IVP} \quad \begin{aligned} &\partial_t u(t,\mathbf{x};\mathbf{p}) = f(u,t,\mathbf{x},\mathbf{p}), \text{ in } \Omega\times]0,T], \\ &u(0,\mathbf{x};\mathbf{p}) = u^0(\mathbf{x},\mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Forward method $S_j = \partial_{p_j} u,\ j = 1,\dots,P$	
Backward method $(\partial_{p_j}\mathcal{F})_{j=1,\dots,P}$	

Sensitivity analysis

$$\mathcal{P}_{IVP} \qquad \begin{aligned} &\partial_t u(t,\mathbf{x};\mathbf{p}) = f(u,t,\mathbf{x},\mathbf{p}), \text{ in } \Omega\times]0,T], \\ &u(0,\mathbf{x};\mathbf{p}) = u^0(\mathbf{x},\mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Forward method $S_j = \partial_{p_j} u,\ j = 1,\dots,P$	$\mathcal{P}_j \left\{ \begin{aligned} &\partial_t S_j = \partial_u f \cdot S_j + \partial_{p_j} f, \text{ in } \Omega\times]0,T], \\ &S_j(0) = S_j^0, \text{ in } \Omega, \\ &+ BCs. \end{aligned} \right.$
Backward method $(\partial_{p_j}\mathcal{F})_{j=1,\dots,P}$	

Sensitivity analysis

$$\mathcal{P}_{IVP} \qquad \partial_t u(t,\mathbf{x};\mathbf{p}) = f(u,t,\mathbf{x},\mathbf{p}),\ \text{in } \Omega\times]0,T],$$
$$u(0,\mathbf{x};\mathbf{p}) = u^0(\mathbf{x},\mathbf{p}),\ \text{in } \Omega,$$
$$+\ BCs.$$

Forward method $S_j = \partial_{p_j} u,\ j = 1,\ldots,P$	$\mathcal{P}_j \left\{ \begin{array}{l} \partial_t S_j = \partial_u f \cdot S_j + \partial_{p_j} f,\ \text{in } \Omega\times]0,T], \\ S_j(0) = S_j^0,\ \text{in } \Omega, \\ +\ BCs. \end{array} \right.$
Backward method $(\partial_{p_j}\mathcal{F})_{j=1,\ldots,P}$	$\mathcal{L}(u,\lambda;\mathbf{p}) = \mathcal{F}(\mathbf{p}) + \int_0^T \int_\Omega \lambda\cdot(f - \partial_t u)\ \mathrm{dx}\ \mathrm{dt}$

Sensitivity analysis

$$\mathcal{P}_{IVP} \qquad \begin{aligned} &\partial_t u(t,\mathbf{x};\mathbf{p}) = f(u,t,\mathbf{x},\mathbf{p}), \text{ in } \Omega\times]0,T], \\ &u(0,\mathbf{x};\mathbf{p}) = u^0(\mathbf{x},\mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Forward method $S_j = \partial_{p_j} u,\ j = 1,\dots,P$	$\mathcal{P}_j \left\{ \begin{aligned} &\partial_t S_j = \partial_u f \cdot S_j + \partial_{p_j} f, \text{ in } \Omega\times]0,T], \\ &S_j(0) = S_j^0, \text{ in } \Omega, \\ &+ BCs. \end{aligned} \right.$
Backward method $(\partial_{p_j}\mathcal{F})_{j=1,\dots,P}$	$\mathcal{P}^* \left\{ \begin{aligned} &\partial_t \lambda = -\lambda \cdot \partial_u f - \partial_u \mathrm{err}, \text{ in } \Omega \times [0,T[, \\ &\lambda(T) = 0, \text{ in } \Omega, \\ &+BCs. \end{aligned} \right.$

Sensitivity analysis

$$\mathcal{P}_{IVP} \qquad \begin{aligned} &\partial_t u(t,\mathbf{x};\mathbf{p}) = f(u,t,\mathbf{x},\mathbf{p}), \text{ in } \Omega\times]0,T], \\ &u(0,\mathbf{x};\mathbf{p}) = u^0(\mathbf{x},\mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Forward method $S_j = \partial_{p_j} u,\ j = 1,\dots,P$	$\mathcal{P}_j \left\{ \begin{aligned} &\partial_t S_j = \partial_u f \cdot S_j + \partial_{p_j} f, \text{ in } \Omega\times]0,T], \\ &S_j(0) = S_j^0, \text{ in } \Omega, \\ &+ BCs. \end{aligned} \right.$
Backward method $(\partial_{p_j}\mathcal{F})_{j=1,\dots,P}$	$\partial_{p_j}\mathcal{F} = \partial_{p_j}\mathcal{L} = \int_0^T \int_\Omega \lambda_j \cdot \partial_{p_j} f \,\mathrm{dx}\,\mathrm{dt}$

Sensitivity analysis

$$\mathcal{P}_{IVP} \qquad \begin{aligned} &\partial_t u(t, \mathbf{x}; \mathbf{p}) = f(u, t, \mathbf{x}, \mathbf{p}), \text{ in } \Omega \times]0, T], \\ &u(0, \mathbf{x}; \mathbf{p}) = u^0(\mathbf{x}, \mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Forward method $S_j = \partial_{p_j} u,\ j = 1, \ldots, P$	IVP + P systems to solve ...
Backward method $(\partial_{p_j} \mathcal{F})_{j=1,\ldots,P}$	IVP + 1 system to solve ...

How to reduce the computational costs of these parameter-dependent problems?

Introduction

$$PDE : \mu \to u(\mu)$$
$$\mu \in \mathcal{G} : \text{Parameter}$$
$$u(\mu) : \text{Solution}$$

We are interested in approximating functions depending on space/time and parameter-dependent.
We denote by $u(x;\mu)$ these functions.
$\mu = (\mathbf{p}, t)$

Reduced Basis Methods (RBM)

- ◇ Resolution by a classical discretization method (FEM/FV ...)
- ◇ A system of size dependent of the number of degrees of freedom $\mathcal{N}$ needs to be solved!
- ◇ How to decrease the High-Fidelity (HF) code execution runtimes when we need to solve the problem for **many different parameter values**?

Reduced Basis Methods (RBM)

- Resolution by a classical discretization method (FEM/FV . . .)
- A system of size dependent of the number of degrees of freedom $\mathcal{N}$ needs to be solved!
- How to decrease the High-Fidelity (HF) code execution runtimes when we need to solve the problem for **many different parameter values**?
  Reduce the dimension of the algebraic system arizing from the discretization of a PDE.
- RBM: The solution is obtained with a projection of the HF problem onto a small subspace.
- RBM: We denote by $(\Phi_i)_{i=1,\dots,N}$ the reduced basis associated to this space, obtained by a POD algorithm.

Reduced Basis Methods (RBM)

$PDE : \mu \to u(\mu)$

$\mu \in \mathcal{G}$: Parameter

$u(\mu)$: Solution

$$\mathcal{P}_{IVP} \quad \begin{aligned} &\partial_t u(t, \mathbf{x}; \mathbf{p}) = f(u, t, \mathbf{x}, \mathbf{p}), \text{ in } \Omega\times]0, T], \\ &u(0, \mathbf{x}; \mathbf{p}) = u^0(\mathbf{x}, \mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Reduced Basis Methods (RBM)

$$PDE : \mu \to u(\mu)$$

$$\mu \in \mathcal{G} : \text{Parameter}$$

$$u(\mu) : \text{Solution}$$

$$\mathcal{P}_{IVP} \quad \begin{aligned} &\partial_t u(t, \mathbf{x}; \mathbf{p}) = f(u, t, \mathbf{x}, \mathbf{p}), \text{ in } \Omega \times]0, T], \\ &u(0, \mathbf{x}; \mathbf{p}) = u^0(\mathbf{x}, \mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Aim of the reduced basis methods (RBM)

Solve the PDE as quickly as possible when it has to be evaluated for many parameter values

Applications

- Fitting parameters to data
- Real-time simulations

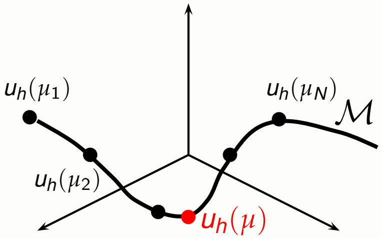
Reduced Basis Methods (RBM)

$PDE : \mu \rightarrow u(\mu)$

$\mu \in \mathcal{G}$: Parameter

$u(\mu)$: Solution

$$\mathcal{P}_{IVP} \quad \begin{aligned} &\partial_t u(t, \mathbf{x}; \mathbf{p}) = f(u, t, \mathbf{x}, \mathbf{p}), \text{ in } \Omega\times]0, T], \\ &u(0, \mathbf{x}; \mathbf{p}) = u^0(\mathbf{x}, \mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$



Solution manifold

Offline Part

$\mathcal{M} = \{u_h(\mu) | \, \mu \in \mathcal{G}\}$

X^N Reduced space

Parameters $\mu_1, \ldots, \mu_N \in \mathcal{G}$

Online Part

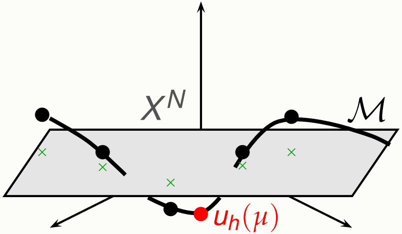
Reduced Basis Methods (RBM)

$PDE : \mu \to u(\mu)$

$\mu \in \mathcal{G}$: Parameter

$u(\mu)$: Solution

$$\mathcal{P}_{IVP} \quad \begin{aligned} &\partial_t u(t,\mathbf{x};\mathbf{p}) = f(u,t,\mathbf{x},\mathbf{p}), \text{ in } \Omega\times]0,T], \\ &u(0,\mathbf{x};\mathbf{p}) = u^0(\mathbf{x},\mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$



Solution manifold

Offline Part

$\mathcal{M} = \{u_h(\mu) |\ \mu \in \mathcal{G}\}$

X^N Reduced space

Parameters $\mu_1, \ldots, \mu_N \in \mathcal{G}$

Online Part

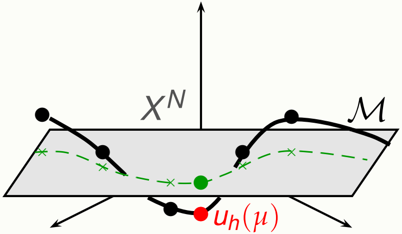
Reduced Basis Methods (RBM)

$PDE : \mu \to u(\mu)$

$\mu \in \mathcal{G}$: Parameter

$u(\mu)$: Solution

$$\mathcal{P}_{IVP} \quad \begin{aligned} &\partial_t u(t, \mathbf{x}; \mathbf{p}) = f(u, t, \mathbf{x}, \mathbf{p}), \text{ in } \Omega \times]0, T], \\ &u(0, \mathbf{x}; \mathbf{p}) = u^0(\mathbf{x}, \mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$



Offline Part

$\mathcal{M} = \{u_h(\mu) | \ \mu \in \mathcal{G}\}$

X^N Reduced space

Parameters $\mu_1, \ldots, \mu_N \in \mathcal{G}$

Online Part

Approximation onto X^N

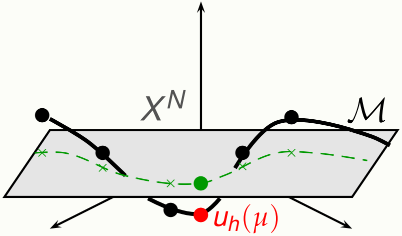
Reduced Basis Methods (RBM)

$PDE : \mu \rightarrow u(\mu)$

$\mu \in \mathcal{G}$: Parameter

$u(\mu)$: Solution

$$\mathcal{P}_{IVP} \quad \begin{aligned} &\partial_t u(t,\mathbf{x};\mathbf{p}) = f(u,t,\mathbf{x},\mathbf{p}), \text{ in } \Omega\times]0,T], \\ &u(0,\mathbf{x};\mathbf{p}) = u^0(\mathbf{x},\mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$



Solution manifold

Offline Part

$\mathcal{M} = \{u_h(\mu) |\ \mu \in \mathcal{G}\}$

X^N Reduced space

Parameters $\mu_1, \dots, \mu_N \in \mathcal{G}$

Online Part

Approximation onto X^N

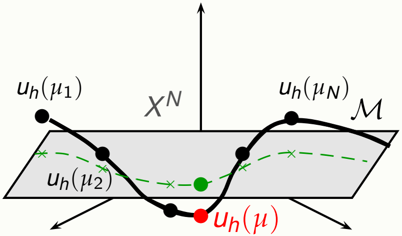
[a] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, P. Wojtaszczyk, *Convergence rates for greedy algorithms in reduced basis methods.* 2011.

Classical greedy algorithm

for $k = 1, \dots, N$:

$$\mu_k = \arg\max_{\mu \in \mathcal{G}} \| u_h(\mu) - P^{k-1}(u_h(\mu)) \|_{L^2(\Omega)}$$

P^{k-1} := Projection onto previous RB

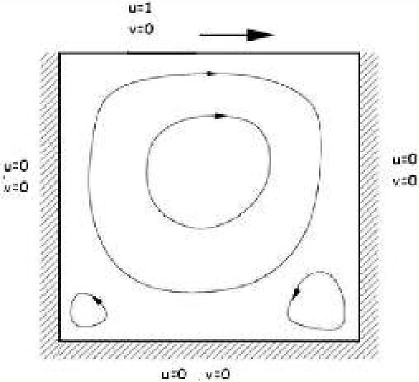


Solution manifold

Orthonormal RB in $L^2 := \Phi_j^h,\ j = 1, \dots, N$

2D lid driven cavity problem

Stationary Navier-Stokes equation in the unit square (+ Dirichlet boundary cdts)



2D steady Navier-stokes equation:

$$
\begin{aligned}
(\mathbf{u}\cdot\nabla)\mathbf{u} - \mu\Delta\mathbf{u} + \nabla p = 0, &\text{ on } \Omega,\\
\nabla\cdot\mathbf{u} = 0, &\text{ on } \Omega,\\
\mathbf{u}_{|\Gamma_{up}} = (1,0)&,\\
\mathbf{u}_{|\partial\Omega\setminus\Gamma_{up}} = (0,0)&,
\end{aligned}
$$

where $\mathbf{u} = (u, v)$ is the velocity,
p is the pressure,
Γ_{up} is the upper border,
$\mu = 1/Re$, and Re is the Reynolds number.

RBM

- The $(\Phi_n)_{n=1,\dots,+\infty}$ of the POD of order n are orthonormal (and the associated eigenvalues positive).
- Any realization of u can be approximated by:

$$\Pi_{V_n} u(x;\mu) = u^N(x;\mu) = \sum_{n=1}^{N} (u(\mu), \Phi_n)\Phi_n(x),$$

POD Energy

$E(\Phi_1,\dots,\Phi_n) = \sum_{i=1}^{ntrain} \|u_i - \Pi_{V_n} u_i\|_2^2$. We have:

$$E(\Phi_1,\dots,\Phi_n) = \sum_{n+1}^{r} \lambda_i,$$

So, we can select N such that $E(\Phi_1,\dots,\Phi_N) \leq \varepsilon$ for a prescribed tolerance. The number of modes required for the RB can be given by the Relative Information Content:

$$I(N) = \frac{\sum_{k=1}^{N} \lambda_k}{\sum_{i=1}^{r} \lambda_i}.$$

POD Galerkin & Results on the model problem

Instead of using **u** as before, we replace it by $\bar{\mathbf{u}} + \sum_{j=1}^{N} \alpha_j^k \Phi_j$

On the **2D lid driven cavity problem**, we obtain:

$$\mathcal{A}_i + \sum_{j=1}^{N} \mathcal{B}_{ij}\alpha_j^k + \sum_{j=1}^{N}\sum_{r=1}^{N} \mathcal{C}_{ijr}\alpha_j^k \alpha_r^{k-1} = 0,$$

where $\mathcal{A} \in \mathbb{R}^N, \mathcal{B} \in \mathbb{R}^{N\times N}$ and $\mathcal{C}_{ijr} = \mathbb{R}^{N\times N^2}$. e.g. $\mathcal{A}_i = \underbrace{\mu(\nabla \bar{\mathbf{u}}, \nabla \Phi_i)}_{A1} + \underbrace{(\bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}}, \Phi_i)}_{A2}$

Online stage

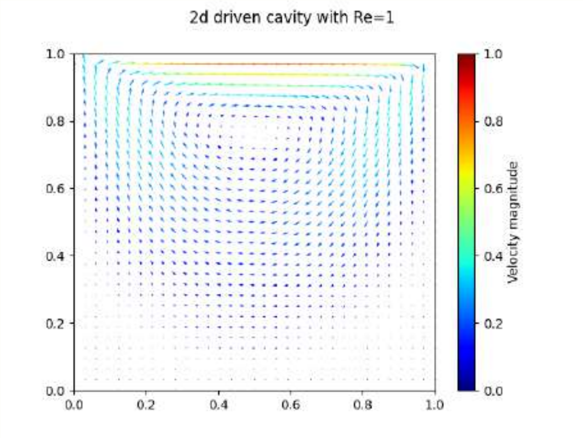
Input: New parameter of interest μ

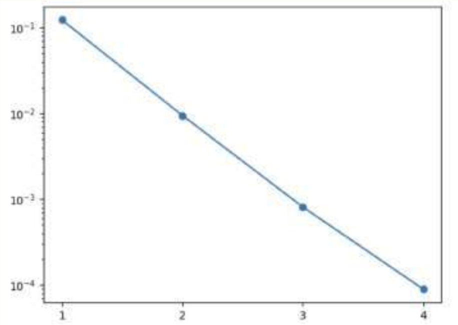
$$M_{ij} = \mathcal{B}_{ij} + \sum_{k=1}^{N} \mathcal{C}_{ijl}\alpha_l^{k-1}$$

Output:

$$\mathbf{M}\alpha^k(\mu) = \mathcal{A}, \qquad \mathbf{u}^N(x;\mu) = \bar{\mathbf{u}} + \sum_{j=1}^{N} \alpha_j(\mu)\Phi_h^j(x).$$

POD Galerkin & Results on the model problem





1−RIC (logscale)

Eigenvalues: [2.2E-02, 2.9E-03, 2.2E-04, 1.9E-05,...]

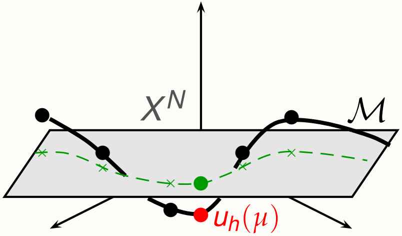
Reduced Basis Methods (RBM)

$PDE : \mu \to u(\mu)$

$\mu \in \mathcal{G}$: Parameter

$u(\mu)$: Solution

$$\mathcal{P}_{IVP} \quad \begin{cases} \partial_t u(t, \mathbf{x}; \mathbf{p}) = f(u, t, \mathbf{x}, \mathbf{p}), \text{ in } \Omega \times]0, T], \\ u(0, \mathbf{x}; \mathbf{p}) = u^0(\mathbf{x}, \mathbf{p}), \text{ in } \Omega, \\ + BCs. \end{cases}$$



Offline Part

$\mathcal{M} = \{u_h(\mu) | \ \mu \in \mathcal{G}\}$

X^N Reduced space

Parameters $\mu_1, \dots, \mu_N \in \mathcal{G}$

Online Part

Approximation onto X^N

[a] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, P. Wojtaszczyk, *Convergence rates for greedy algorithms in reduced basis methods.* 2011.

NIRB methods

Several definitions of non-intrusivity . . .

Reduced basis methods

- Galerkin-POD[a]
- Petrov-Galerkin RBM[b]
- EIM[c]
- POD-DL-ROM[d]
- 2-grid method[e]
- POD-I[f] (Reg, NN)
- PBDW[g]
- PGD[h]
- Surrogate problems[i]
- Operator inference[j]

...

[a] J.L. Lumley. *The structure of inhomogeneous turbulent flows.* 1967

[b] K. Carlberg, M. Barone, H. Antil. *Galerkin v. least-squares Petrov-Galerkin projection in nonlinear model reduction.* 2017

[c] M. Barrault, Y. Maday, NC. Nguyen, AT. Patera*An 'empirical interpolation'method: application to efficient reduced-basis discretization of partial differential equations.*2004.

[d] S. Fresca, A. Manzoni *POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by pod.*2022

[e] Chakir, R. & Maday, Y. *A two-grid finite-element/reduced basis scheme for the approximation of the solution of parameter dependent PDE.*2009

[f] D. Xiao, F. Fang, C. Pain, G. Hu. *Non-intrusive reduced-order modelling of the NavierStokes equations based on RBF interpolation.* 2015.

[g] J.K. Hammond, R. Chakir, F. Bourquin, Y. Maday. *PBDW: A non-intrusive Reduced Basis Data Assimilation method and its application to an urban dispersion modeling framework.*2019

[h] X. Zou, M. Conti, P. Diez and F. Auricchio *A non-intrusive proper generalized decomposition scheme with application in biomechanics.*2017

[i] T. Guo, O. Roko, K. Veroy,*Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method.*2021

[j] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, K. Willcox, *Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms.* 2021

Reduced basis methods: https://reducedbasis.github.io

- Galerkin-POD[a]
- Petrov-Galerkin RBM[b]
- EIM[c]
- POD-DL-ROM[d]
- 2-grid method[e]
- POD-I[f] (Reg, NN)
- PBDW[g]
- PGD[h]
- Surrogate problems[i]
- Operator inference[j]

...

[a] J.L. Lumley. *The structure of inhomogeneous turbulent flows.* 1967

[b] K. Carlberg, M. Barone, H. Antil. *Galerkin v. least-squares Petrov-Galerkin projection in nonlinear model reduction.* 2017

[c] M. Barrault, Y. Maday, NC. Nguyen, AT. Patera*An 'empirical interpolation'method: application to efficient reduced-basis discretization of partial differential equations.*2004.

[d] S. Fresca, A. Manzoni *POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by pod.*2022

[e] Chakir, R. & Maday, Y. *A two-grid finite-element/reduced basis scheme for the approximation of the solution of parameter dependent PDE.*2009

[f] D. Xiao, F. Fang, C. Pain, G. Hu. *Non-intrusive reduced-order modelling of the NavierStokes equations based on RBF interpolation.* 2015.

[g] J.K. Hammond, R. Chakir, F. Bourquin, Y. Maday. *PBDW: A non-intrusive Reduced Basis Data Assimilation method and its application to an urban dispersion modeling framework.*2019

[h] X. Zou, M. Conti, P. Diez and F. Auricchio *A non-intrusive proper generalized decomposition scheme with application in biomechanics.*2017

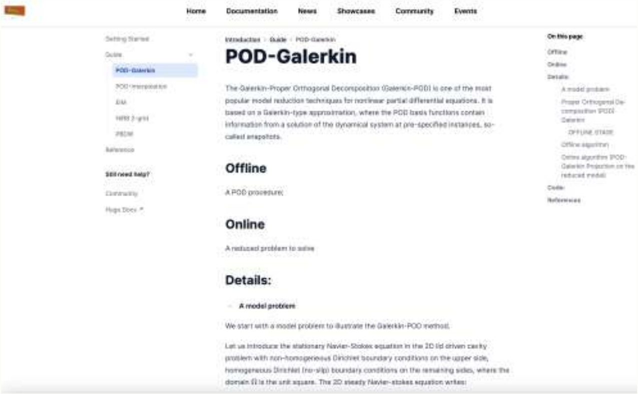
[i] T. Guo, O. Roko, K. Veroy,*Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method.*2021

[j] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, K. Willcox, *Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms.* 2021

Reduced basis methods: https://reducedbasis.github.io

- ◇ Galerkin-POD[a]
- ◇ Petrov-Galerkin RBM[b]
- ◇ EIM[c]
- ◇ POD-DL-ROM[d]
- ◇ 2-grid method[e]
- ◇ POD-I[f] (Reg, NN)
- ◇ PBDW[g]
- ◇ PGD[h]
- ◇ Surrogate problems[i]
- ◇ Operator inference[j]

...



Reduced basis methods: https://reducedbasis.github.io

- Galerkin-POD[a]
- Petrov-Galerkin RBM[b]
- EIM[c]
- POD-DL-ROM[d]
- 2-grid method[e]
- POD-I[f] (Reg, NN)
- PBDW[g]
- PGD[h]
- Surrogate problems[i]
- Operator inference[j]
- ...

We are going to use in this example the famous 2D-driven cavity problem with the Finite Element Method (FEM), which consists in solving on a unit square (denoted Ω) the following equations:

$$\begin{aligned} &-\nu\Delta u + (u\cdot\nabla)u + \nabla p = 0, \text{ in } \Omega,\\ &\nabla . u = 0, \text{ in } \Omega,\\ &(u_1,u_2) = (1,0), \text{ on } \Omega_{up} := \partial\Omega \cap \{y=1\},\\ &(u_1,u_2) = (0,0), \text{ on } \partial\Omega\backslash\Omega_{up}, \end{aligned}$$

where $u = (u_1,u_2) \in V := H^1_{d,0}(\Omega)^2 = \{u \in H^1(\Omega)^2, \gamma_{\partial\Omega\backslash\Omega_{up}} u = 0, \gamma_{up} u = (1,0)\}$ (γ stands for the trace operator) represents the velocity of the incompressible fluid, $p \in L^2(\Omega)$ its pressure, and $\nu = \frac{1}{Re}$ where Re is the Reynolds parameter. Here, the Reynolds number is our parameter of interest ($\mu = Re$). For the nonlinearity we adopt a fixed-point iteration scheme, and after multiplying by test functions q and v (resp. for pressure and velocity), which in variational form reads:

$$\nu(\nabla u^k, \nabla v) + ((u^{k-1}\cdot\nabla)u^k, v) - (p^k, \nabla\cdot v) - (q, \nabla\cdot u^k) + 10^{-10}(p^k, q) = 0, \text{in } \Omega,$$

where u^{k-1} is the previous step solution, and we iterate until a threshold is reached (until $\|u^k - u^{k-1}\| < \varepsilon$). Here, with the term $10^{-10}(p^k, q)$, we impose the average of the pressure $\int_\Omega p^k$ to be equal to 0. For more details on how we derive this formulation, visit the link : https://github.com/grosjean1/navierStokes (FEM.pdf).

We employ Taylor-Hood elements to get a proper solution (e.g. P2-P1 for the tuple velocity-pressure) and obtain the system $\mathbf{Kx} = \mathbf{f}$ to solve where $\mathbf{K} = \begin{pmatrix} \mathbf{A} & -\mathbf{B}^T \\ -\mathbf{B} & 10^{-10}\mathbf{C} \end{pmatrix}$, $\mathbf{x}$ stands for the tuple velocity-pressure (u_1^k, u_2^k, p^k), and where the assembled matrix $\mathbf{A}$ corresponds to the bilinear part $\nu(\nabla u^k, \nabla v) + ((u^{k-1}\cdot\nabla)u^k), v)$, the matrix B to the bilinear part $(p^k, \nabla\cdot v)$ and $\mathbf{C}$ is the mass matrix applied to the pressure variable ($(\cdot,\cdot)$ represents either the L^2 inner-product onto the velocity space or onto the pressure space).

The dirichlet boundary conditions are imposed with a penalization method, called with scikit-FEM by the line:

solve(*condense(K,f, x=uvp, D=D)),

where x=uvp gives the values at the boundaries of the velocity and D refers to the boundary decomposition.

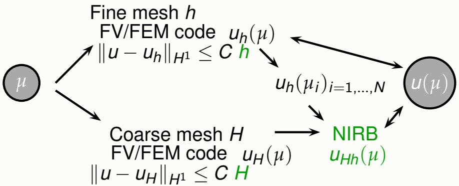
```
""" First we define a mesh for the unit square with the boundary decomposition """
mesh= skfem.MeshTri().refined(4).with_boundaries(
    {
        "left": lambda x: x[0] == 0,
        "right": lambda x: x[0] == 1,
        "down": lambda x: x[1] == 0,
        "up": lambda x: x[1] == 1,

    }
)
```

NIRB two-grid method

NIRB 2-grid method

$u(\mu)$: Exact solution - $u^N_{Hh}(\mu)$: NIRB Solution[1]



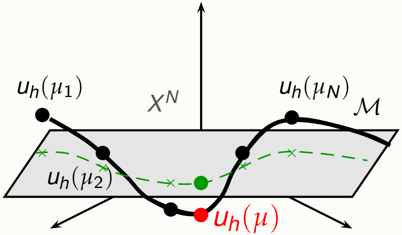
Decomposition

$$u_h(\mathbf{x}, \mu) = \sum_j a_j^h(\mu)\, \Phi_j^h(\mathbf{x}),$$

$(\Phi_j^h)_{j=1,\dots,N} \in X_h^N$: modes (RB)

Optimal coefficients: $a_j^h(\mu) = (u_h(\mu), \Phi_j^h(\mathbf{x}))$,

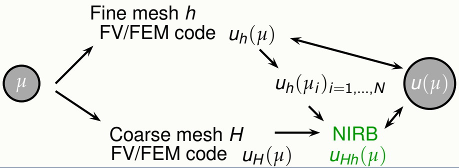
Our choice: $b_j^H(\mu) = (u_H(\mu), \Phi_j^h(\mathbf{x}))$



Discretized solution manifold

NIRB 2-grid method

$u(\mu)$: Exact solution - $u^N_{Hh}(\mu)$: NIRB Solution[1]



Decomposition

$$u_h(\mathbf{x},\mu) = \sum_j a_j^h(\mu)\,\Phi_j^h(\mathbf{x}),$$

$(\Phi_j^h)_{j=1,\dots,N} \in X_h^N$: modes (RB)

Optimal coefficients: $a_j^h(\mu) = (u_h(\mu), \Phi_j^h(\mathbf{x}))$,

Our choice: $b_j^H(\mu) = (u_H(\mu), \Phi_j^h(\mathbf{x}))$

Post-treatment : rectification[1]

$B_{i,j} = b_j^H(\mu^i) \to A_{i,j} = a_j^h(\mu^i) \in \mathbb{R}^{N\times N}$

Least-square approach :

$$\arg\min_{R^i} \|BR^i - A^i\|_2^2 + \varepsilon \|R^i\|_2^2.$$

$$R_i = (B^T B + \varepsilon I_N)^{-1} B^T A_i,\ \forall i = 1,\cdots,N.$$

$$u^N_{Hh}(\mu) = \sum_{i,j=1}^{N} R_{ij}\, b_j^H(\mu)\,\Phi_i$$

[1] Chakir, R. & Maday, Y. (2009). *A two-grid finite-element/reduced basis scheme for the approximation of the solution of parameter dependent PDE.*

NIRB two-grid method applied to parabolic equations

FEM Error estimates within the elliptic context

$$\partial_t u - \nabla \cdot (A(p)\nabla u) = f, \text{ dans } \Omega \times]0, T],$$
$$u(\mathbf{x}, 0) = u^0(\mathbf{x}), \text{ in } \Omega,$$
$$u = 0, \text{ on } \partial\Omega, \; t \in [0, T],$$

$\mu = (t, p) \in \mathcal{G} \subset \mathbb{R}^+ \times \mathbb{R}^P$: parameter

$u(t, \mathbf{x}; p)$: unknowns

FEM error estimates

Theorem [E G.-Y M.] (H^1 Error estimate)[2]

Pod-greedy[3] or greedy-greedy algorithm for the BR construction

$$\forall n,\ \|u(t^n)(p) - u_{Hh}^{N,n}(p)\|_{H^1(\Omega)} \leq \overbrace{\varepsilon(N)}^{T_1} + \underbrace{C_1(p)(h+\Delta t_F)}_{T_2} + \overbrace{C_2(N)(H^2+\Delta t_G^2)}^{T_3}$$

$C_1,\ C_2$: constants independent of h et H

$u_h^n(p^k) = u_h(p^k, t^n) \in V_h$, for $n = 1, \ldots, N_T$: Fine snapshots

$u_H^m(p) = u_H(p, \tilde{t}^m) \in V_H$, for $m = 1, \ldots, M_T$: Coarse solution, $H \gg h$.

---

[1] V. Thomée, *Galerkin finite element methods for parabolic problems*, 1984.

[2] E. G., Y. Maday, *Error estimate of the non-intrusive reduced basis two-grid method with parabolic equations.* 2023.

[3] B. Haasdonk, *Convergence rates of the pod-greedy method*, 2013.

FEM error estimates

Theorem [E G.-Y M.] (H^1 Error estimate)[2]

$$\forall n,\ \|u(t^n)(p) - u_{Hh}^{N,n}(p)\|_{H^1(\Omega)} \leq \overbrace{\varepsilon(N)}^{T_1} + \underbrace{C_1(p)(h + \Delta t_F)}_{T_2} + \overbrace{C\sqrt{\lambda_N}(H^2 + \Delta t_G^2)}^{T_3}$$

λ_N: $\forall v \in X_h^N, \int_\Omega \nabla \Phi_h \cdot \nabla v = \lambda \int_\Omega \Phi_h \cdot v,$
and C_1, C: constants independent of h, H and N.

$$\|u(\tilde{t}^m) - u_H^m\|_{L^2(\Omega)} \leq C\, H^2 \Big[\|u^0\|_{H^2(\Omega)} + \int_0^{\tilde{t}^m} \|u_t\|_{H^2(\Omega)}\, ds\Big]$$
$$+ C\, \Delta t_G^2 \Big[\Big(\int_0^{\tilde{t}^m} \|u_{ttt}\, ds\|^2_{L^2(\Omega)}\Big)^{1/2} + \Big(\int_0^{\tilde{t}^m} \|\Delta u_{tt}\|^2_{L^2(\Omega)}\, ds\Big)^{1/2}\Big].$$

[1] V. Thomée, *Galerkin finite element methods for parabolic problems*, 1984.

[2] E. G., Y. Maday, *Error estimate of the non-intrusive reduced basis two-grid method with parabolic equations.* 2023.

Decomposition

Projection coefficients

$$u_h(\mathbf{x}, t^n; p) = \sum_j a_j^h(p, t^n)\, \Phi_j^h(\mathbf{x}), \quad n = 1, \dots, N_T$$

$(\Phi_j^h)_{j=1,\dots,N} \in X_h^N$: L^2-orthonormalized basis functions (modes)

Coefficients $a_j^h(\mu) = a_j^h(p, t^n)$

Optimal coefficients: $(u_h(p, t^n), \Phi_j^h(\mathbf{x})), n = 1, \dots, N_T$

Our choice: $(I_n(u_H(p, \tilde{t}^m)), \Phi_j^h(\mathbf{x}))_{L^2},\ m = 1, \dots, M_T$

NIRB methods applied to sensitivity analysis

Sensitivity analysis

$$\mathcal{P}_{IVP} \qquad \begin{aligned} &\partial_t u(t,\mathbf{x};\mathbf{p}) = f(u,t,\mathbf{x},\mathbf{p}), \text{ in } \Omega\times]0,T], \\ &u(0,\mathbf{x};\mathbf{p}) = u^0(\mathbf{x},\mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Forward method $S_j = \partial_{p_j} u,\ j = 1,\dots,P$	IVP + P systems to solve ...
Backward method $(\partial_{p_j}\mathcal{F})_{j=1,\dots,P}$	IVP + 1 system to solve ...

FEM Error estimates

Theorem [E G.-B. S.] (H^1 error estimate)[1]

$$\forall n,\ \|S(t^n)(p) - S_{Hh}^{N,n}(p)\|_{H^1(\Omega)} \leq \overbrace{\varepsilon(N)}^{T_1} + \underbrace{C_1(p)(h + \Delta t_F)}_{T_2} + \overbrace{C(p)\sqrt{\lambda_N}(H^2 + \Delta t_G^2)}^{T_3}$$

λ_N: $\forall v \in X_h^N, \int_\Omega \nabla\Phi_h \cdot \nabla v = \lambda \int_\Omega \Phi_h \cdot v$,
and C_1, C: Constants independent of h, H and N.
Pod-greedy or greedy-greedy algorithm for the RB construction[2]

$u_h^n(p^k) = u_h(p^k, t^n) \in V_h$, for $n = 1, \dots, N_T$: Fine snapshots
$u_H^m(p) = u_H(p, \tilde{t}^m) \in V_H$, for $m = 1, \dots, M_T$: Coarse solution, $H \gg h$.

$$\|S_H^m - S(\tilde{t}^m)\|_{L^2} \leq CH^2\Big[\|S^0\|_{H^2} + \int_0^{\tilde{t}^m} \|S_t\|_{H^2}\,\mathrm{ds} + C(\mu)\Big[\int_0^{\tilde{t}^m} \|u_t\|_{H^2}^2\,\mathrm{ds}\Big]^{1/2}\Big]$$
$$+ C\Delta t_G^2\Big(\int_0^{\tilde{t}^m} \|S_{ttt}\|_{L^2}\,\mathrm{ds} + \Big[\int_0^{\tilde{t}^m} \|\Delta u_{tt}\|_{L^2}^2\,\mathrm{ds}\Big]^{1/2} + C(\mu)\Big[\int_0^{\tilde{t}^m} [\|u_{ttt}\|_{L^2}^2\,\mathrm{ds}]^{1/2} + \int_0^{\tilde{t}^m} \|\Delta S_{tt}\|_{L^2}\,\mathrm{ds}\Big]\Big).$$

[1] E. G., Y. Maday, *Error estimate of the non-intrusive reduced basis two-grid method with parabolic equations.* 2023.
[2] B. Haasdonk, *Convergence rates of the pod-greedy method.* 2013.

Rectification post-treatment

For each $p_k^i, k = 1, \ldots, P, i = 1, \ldots, N$

$S_h^k(p^i, t^n) = \partial_{p_k} u(p^i, t^n)$

Fine coefficients

$$\boxed{a_j^h(p^i, t^n) = (S_h^k(p^i, t^n), \zeta_j^k)}$$

Coarse coefficients

$$\boxed{b_j^H(p^i, t^n) = (I_n(S_H^k(p^i, \tilde{t}^m), \zeta_j^k))}$$

Rectification post-treatment

$$\mathcal{P}_{IVP} : \begin{cases} \partial_t u(t, \mathbf{x}; \mathbf{p}) = f(u, t, \mathbf{x}, \mathbf{p}), \text{ in } \Omega \times]0, T], \\ u(0, \mathbf{x}; \mathbf{p}) = u^0(\mathbf{x}, \mathbf{p}), \text{ in } \Omega, \\ +BCs. \end{cases}$$

Forward method $S_j = \partial_{p_j} u, \ j = 1, \ldots, P$	IVP + P systems to solve . . .
Backward method $(\partial_{p_j} \mathcal{F})_{j=1,\ldots,P}$	IVP + 1 system to solve . . .

Rectification post-treatment

For each $p_k^i, k = 1, \ldots, P, i = 1, \ldots, N$
$S_h^k(p^i, t^n) = \partial_{p_k} u(p^i, t^n)$

Fine coefficients

$$\boxed{a_j^h(p^i, t^n) = (S_h^k(p^i, t^n), \zeta_j^k)}$$

Coarse coefficients

$$\boxed{b_j^H(p^i, t^n) = (I_n(u_H(p^i, \tilde{t}^m), \Phi_j))}$$

Inputs-outputs for our model,

$$\mathcal{D} = \{(\mathbf{B}, \mathbf{A})\},$$

Rectification post-treatment

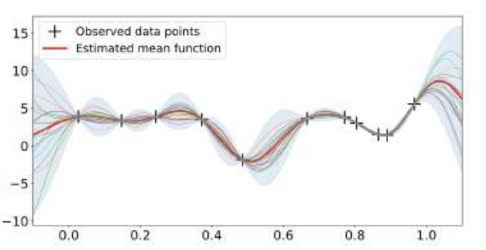
$$\mathcal{P}_{IVP} : \begin{cases} \partial_t u(t, \mathbf{x}; \mathbf{p}) = f(u, t, \mathbf{x}, \mathbf{p}), \text{ in } \Omega \times]0, T], \\ u(0, \mathbf{x}; \mathbf{p}) = u^0(\mathbf{x}, \mathbf{p}), \text{ in } \Omega, \\ + BCs. \end{cases}$$

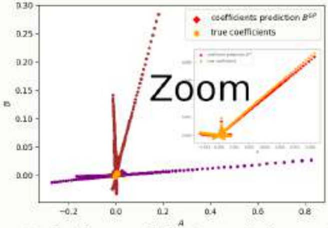
Forward method $S_j = \partial_{p_j} u,\ j = 1, \dots, P$	IVP system to solve . . .
Backward method $(\partial_{p_j} \mathcal{F})_{j=1,\dots,P}$	IVP + 1 system to solve . . .

Two-grid algorithm with a non-deterministic approach : GPR model[1]

GP = Probability distribution over possible functions that fit a set of points.

GPR model = provides uncertainty estimations together with prediction values





Projection coefficients prediction on X^N

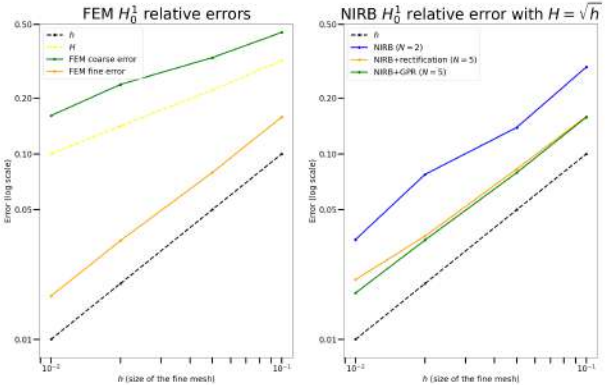
[1] G., E. & Simeon, B. (2024). The non-intrusive reduced basis two-grid method applied to sensitivity analysis.

[2] Guo, M., & Hesthaven, J. S. (2019). Data-driven reduced order modeling for time-dependent problems.

GPR NIRB 2-grid method

$$\partial_t u - \nabla\cdot(A(p)\nabla u) = f, \text{ in } \Omega\times]0,T],$$
$$u(\mathbf{x},0) = u^0(\mathbf{x}), \text{ in } \Omega,$$
$$u = 0, \text{ on } \partial\Omega,\ t\in[0,T],$$

$$\mathcal{D} = \{(\mathbf{B},\mathbf{A})\},$$
$$f:\mathbb{R}^{N\times N_T}\to\mathbb{R}^{N\times N_T},\ f(\mathbf{A}_k)=\mathbf{B}_k,\ k=1,\dots,N.$$



Direct sensitivities : $\ell^\infty(0,\dots,N_T;H_0^1(\Omega))$ relative errors

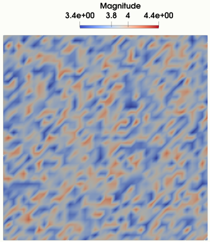
Brusselator equations

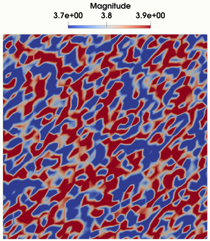
$$\partial_t u = a + uv^2 - (b+1)u + \alpha \Delta u$$
$$\partial_t v = bu - uv^2 + \alpha \Delta v,$$
$$+ \ BCs.$$

$(u(\mathbf{x}, t; \mathbf{p}), v(\mathbf{x}, t; \mathbf{p}))$: Unknowns

$\boldsymbol{p} = (a, b, \alpha) \in \mathbb{R}^3$: Variable parameter

Brusselator equations





Brusselator solutions

Dot Product kernel [1]

$$\kappa(\mathbf{x},\mathbf{x}') = 1 + \mathbf{x}\cdot\mathbf{x}'.$$

$$\text{for } (\mathbf{x},\mathbf{x}') \in \mathbb{R}^{N\times N_T} \times \mathbb{R}^{N\times N_T}, f(\mathbf{x}) \sim GP(0,\kappa(\mathbf{x},\mathbf{x}')), \mathbf{y} = f(\mathbf{x}),$$

Parameters a-b-α	Fine error	Coarse error	True projection S_{hh}^N	NIRB + rectification	NIRB+GPR
3-2-0.01	6.4×10^{-2}	3.1×10^{-1}	6.4×10^{-2}	6.4×10^{-2}	6.8×10^{-2}
3-3-0.01	6.2×10^{-2}	3.0×10^{-1}	6.2×10^{-2}	6.2×10^{-2}	6.9×10^{-2}
3-4-0.01	8.6×10^{-2}	3.9×10^{-1}	8.6×10^{-2}	8.6×10^{-2}	1.4×10^{-1}
4-2-0.0005	7.3×10^{-2}	3.3×10^{-1}	7.3×10^{-2}	7.3×10^{-2}	6.8×10^{-2}
4-3-0.0005	7.1×10^{-2}	3.3×10^{-1}	7.1×10^{-2}	7.1×10^{-2}	7.7×10^{-2}
4-4-0.0005	8.5×10^{-2}	3.8×10^{-1}	8.5×10^{-2}	8.5×10^{-2}	9.2×10^{-2}

relative $\ell^\infty(0,\dots,N_T;\ H^1(\Omega))$ errors (and $\ell^\infty(0,\dots,M_T;\ H^1(\Omega))$ for the coarse ones) with leave-one-out strategy $N = 40$ for the parameter b

[1] N. C. Nguyen and J. Peraire. *Gaussian functional regression for output prediction: Model assimilation and experimental design.* 2016.

NIRB on the adjoint

$$\mathcal{P}_{IVP} \qquad \begin{aligned} &\partial_t u(t, \mathbf{x}; \mathbf{p}) = f(u, t, \mathbf{x}, \mathbf{p}), \text{ in } \Omega \times]0, T], \\ &u(0, \mathbf{x}; \mathbf{p}) = u^0(\mathbf{x}, \mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Forward method $S_j = \partial_{p_j} u,\ j = 1, \dots, P$	
Backward method $(\partial_{p_j} \mathcal{F})_{j=1,\dots,P}$	$\mathcal{P}^* \left\{ \begin{aligned} &\partial_t \lambda = -\lambda \cdot \partial_u f - \partial_u \mathrm{err}, \text{ in } \Omega \times [0, T[, \\ &\lambda(T) = 0, \text{ in } \Omega, \\ &+ BCs. \end{aligned} \right.$

NIRB on the adjoint

$$\mathcal{P}_{IVP} \qquad \begin{aligned} &\partial_t u(t,\mathbf{x};\mathbf{p}) = f(u,t,\mathbf{x},\mathbf{p}), \text{ in } \Omega\times]0,T], \\ &u(0,\mathbf{x};\mathbf{p}) = u^0(\mathbf{x},\mathbf{p}), \text{ in } \Omega, \\ &+ BCs. \end{aligned}$$

Forward method $S_j = \partial_{p_j} u,\ j = 1,\dots,P$	
Backward method $(\partial_{p_j}\mathcal{F})_{j=1,\dots,P}$	$\partial_{p_j}\mathcal{F} = \partial_{p_j}\mathcal{L} = \int_0^T \int_\Omega \lambda_j \cdot \partial_{p_j} f \,\mathrm{dx}\,\mathrm{dt}$

Numerical results on the heat equation

Table: Adjoint formulation without noise: Maximum absolute $\ell^\infty(0,\dots,N_T;H_0^1(\Omega))$ error over the parameters with $N=15$

Δt_F-Δt_G	NIRB rectified error	fine error	true projection error	coarse error
0.01-0.1	4.9×10^{-6}	7.0×10^{-6}	7.0×10^{-6}	3.3×10^{-4}
0.02-0.1414	9.1×10^{-6}	8.1×10^{-6}	8.1×10^{-6}	7.0×10^{-4}
0.05-0.22	1.2×10^{-4}	1.0×10^{-4}	1.0×10^{-4}	1.3×10^{-3}
0.1-0.32	2.2×10^{-4}	1.7×10^{-4}	1.7×10^{-4}	2.3×10^{-3}

Table: Adjoint formulation with noise: Maximum absolute $\ell^\infty(0,\dots,N_T;H_0^1(\Omega))$ error over the parameters with $N=15$ and noisy measurements

Δt_F-Δt_G	NIRB rectified error	fine error	true projection error	coarse error
0.01-0.1	6.7×10^{-4}	6.6×10^{-4}	6.7×10^{-4}	4.0×10^{-3}
0.02-0.1414	1.4×10^{-3}	1.3×10^{-3}	1.3×10^{-3}	6.0×10^{-3}
0.05-0.22	6.3×10^{-3}	6.3×10^{-3}	6.3×10^{-3}	8.1×10^{-3}
0.1-0.32	3.2×10^{-3}	3.0×10^{-3}	3.0×10^{-3}	1.0×10^{-2}

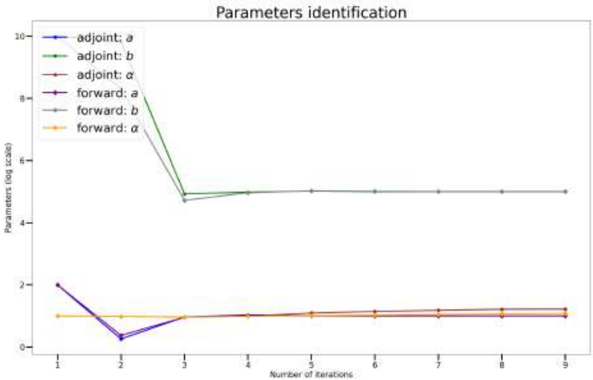
Numerical results on the Brusselator

Objective values with $h = 0.02 \simeq H^2$

Parameter	fine values	rectified NIRB $N = 5$	rectified NIRB $N = 40$
a	-2.74×10^{-6}	-4.97×10^{-3}	-5.70×10^{-5}
b	1.17×10^{-5}	1.66×10^{-2}	1.66×10^{-4}
α	-6.73×10^{-4}	-9.54×10^{-2}	-3.8×10^{-2}

Parameter identification results

NIRB with Gauss-Newton algorithm



Identification of $a = 1, b = 5$ and $\alpha = 2$

Parameter identification results

Table: FEM runtimes (h:min:sec)

FEM high fidelity solver	FEM coarse solution
00:07:05	00:00:58

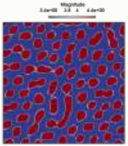
Table: NIRB runtimes ($N = 40$, h:min:sec)

	Offline	Online
Rectified NIRB	01:25:00	00:04:02
NIRB-GP	01:32:00	00:03:57

[1] https://github.com/grosjean1/SensitivityAnalysisWithNIRBTwoGridMethod/

Conclusions & Perspectives

Sensitivity analysis & NIRB with parabolic equations



Perspectives

Two-grid a-posteriori & rectification error estimates

Analysis with the GPR model

Conclusions & Perspectives

Sensitivity analysis & NIRB with parabolic equations

Merci !

Perspectives

- Two-grid a-posteriori & rectification error estimates
- Analysis with the GPR model