Reduced order methods: application to reactor core spatial dynamics

SERPENT and Multiphysics workshop

Alberto Sartori[*,§]

Joint work with: Antonio Cammi[*], Lelio Luzzi[*] and Gianluigi Rozza[§]

[*]Politecnico di Milano, Department of Energy, CeSNEF-Nuclear Engineering Division

[§]SISSA mathLab, International School for Advanced Studies, Mathematics Area

February 27, 2015 · LPSC - Grenoble

Outline

Reduced order methods: application to reactor core spatial dynamics

Gen IV nuclear reactors vs. current nuclear reactors

- No experimental data
- More severe constraints
- Innovative systems

What it is needed

- More accurate simulation tools
- Better description of the phenomena

What it is usually done

- Control: point-wise kinetics
- Design: coupled-code techniques

Reduced order methods: application to reactor core spatial dynamics

Gen IV nuclear reactors vs. current nuclear reactors

- No experimental data
- More severe constraints
- Innovative systems

What it is needed

- More accurate simulation tools
- Better description of the phenomena

What it is usually done

- Control: point-wise kinetics — Spatial description: ρ, Φ
- Design: coupled-code techniques — Multi-Physics

What are Reduced Order Models?

Reduced representations of the full order problem

No reduced physics

Finite Element

- $\mathcal{N}$ *local generic* basis functions

ROM

- *N global characteristic* basis functions

$$N \ll \mathcal{N}$$

Reduction via projection

- Galerkin projection onto an *ad hoc* built basis
- Proper Orthogonal Decomposition (POD)
- Reduced Basis method (RB)

Offline-online splitting

"Truth" Parametrized PDEs → Reduced Basis construction (POD-greedy) → ROM preparation

Reduced Basis construction (POD-greedy) → Error Bounds

Offline-online splitting

"Truth" Parametrized PDEs → Reduced Basis construction (POD-greedy) → ROM preparation / Error Bounds

Parameter μ → ROM → Output / Error Bounds

Worked problem

Goals

- fast-running simulation tool (control-oriented)
- spatial effects

Multi-group time-dependent neutron diffusion equation

$$\begin{cases} \underline{\underline{V}}^{-1}\dfrac{\partial \underline{\Phi}}{\partial t} & = \nabla \cdot \left(\underline{\underline{D}}\nabla\underline{\Phi}\right) - \underline{\underline{\Sigma_a}}\underline{\Phi} - \underline{\underline{\Sigma_s}}\underline{\Phi} + (1-\beta)\underline{\chi_p}\underline{F}^T\underline{\Phi} + \sum_m \lambda_m \underline{\chi_m} C_m \\ \dfrac{\partial C_m(t)}{\partial t} & = -\lambda_m C_m(t) + \beta_m \underline{F}^T\underline{\Phi} \qquad \text{for } m = 1, \cdots, 8 \end{cases}$$

$$\underline{\Phi}(\mathbf{r}, t) \simeq \sum_i \underline{\underline{b_i}}(\mathbf{r})\underline{a_i}(t)$$

1. Modal Method (MM)
2. Proper Orthogonal Decomposition (POD)

POD as a minimization problem

- Given multiple snapshots: $\{y_j\}_{j=1}^{n} \subset X$

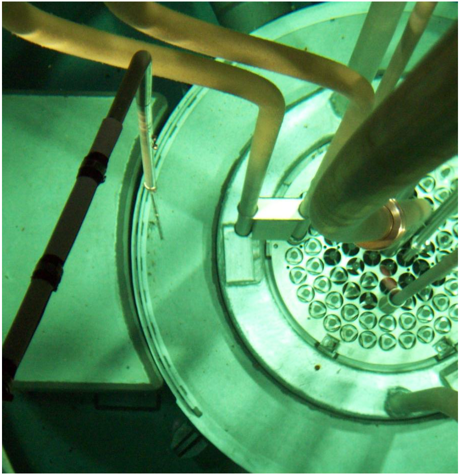
Proper Orthogonal Decomposition

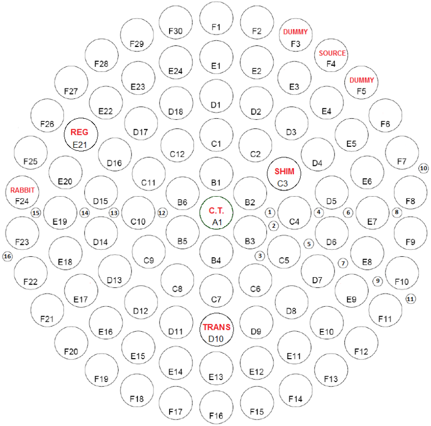
for any $\ell \in \{1, \dots, n\}$ solve

$$\min \sum_{j=1}^{n} \alpha_j \| y_j - \sum_{i=1}^{\ell} \langle y_j, \psi_i \rangle_X \, \psi_i \|_X^2 \quad \text{s.t.} \quad \{\psi_i\}_{i=1}^{\ell} \subset X, \langle \psi_i, \psi_j \rangle_X = \delta_{ij}, \alpha_j > 0$$

Methodological approach

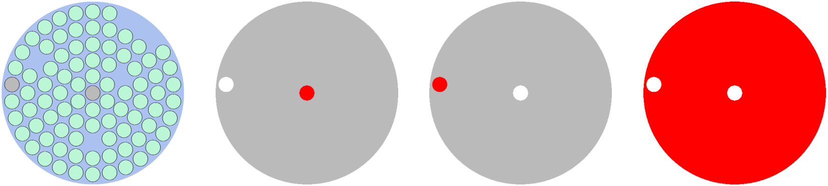
- Reference to the TRIGA Mark II nuclear reactor



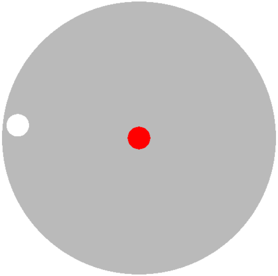


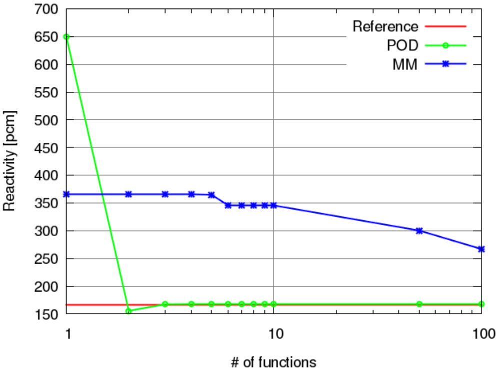
Methodological approach

- Reference to the TRIGA Mark II nuclear reactor
- Simulated perturbations



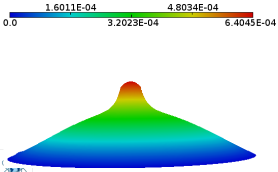
- Up to 100 basis functions
- Reference solution: eigenvalue calculation
- ARCO Project (INFN)



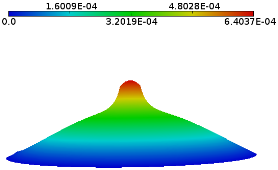


Neutron flux [$1/cm^2$ s]

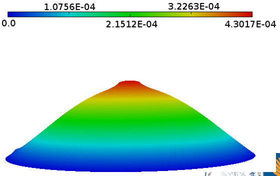
Reference

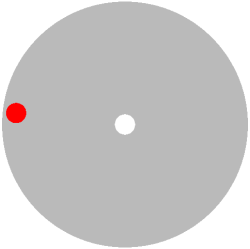


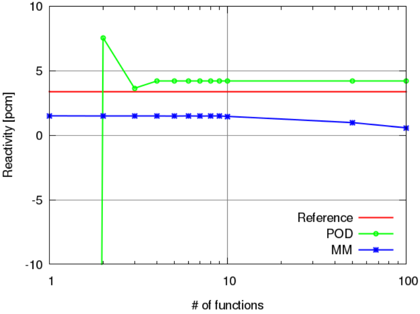
POD

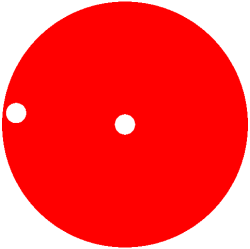


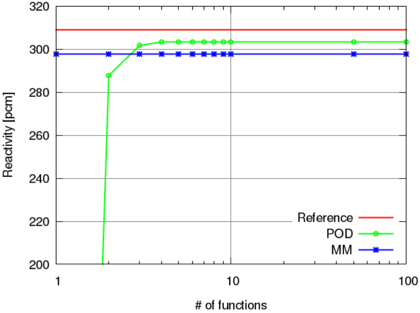
MM



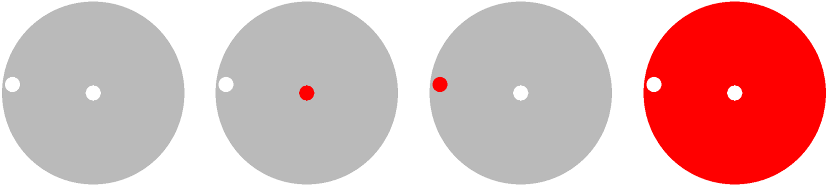


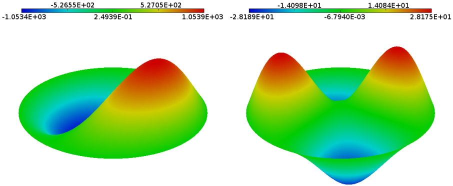

Reactivity [pcm]
10
5
0
-5
-10
1
10
100
# of functions
Reference
POD
MM



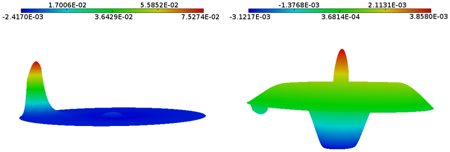

Reactivity [pcm]
320
300
280
260
240
220
200
1
10
100
# of functions
Reference
POD
MM

Basis functions





Eigenfunctions



POD

Next step

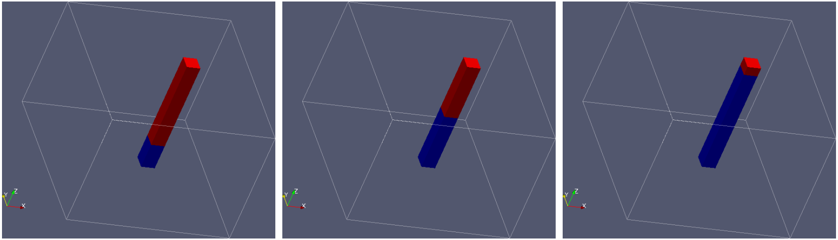
Lessons learned

- ROMs are suitable for *parametrized* PDEs
- parameters can be both physical and geometrical
- basis can be trained
- hierarchical space is obtained

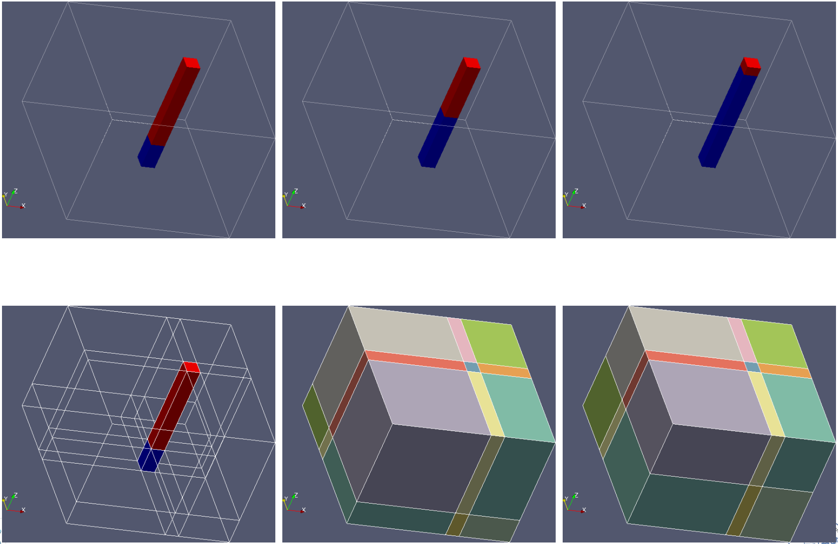
➠ Modelling of control rod movement

1. Time-dependent
2. Stationary (eigenvalue problem)

Control rod movement $\rightarrow$ geometric parametrization



Control rod movement → geometric parametrization



Modelling approach

- TRIGA Mark II nuclear reactor
- finite element method
- reduced basis method

Multi-group time-dependent neutron diffusion equation

$$\begin{cases} \underline{\underline{V}}^{-1}\dfrac{\partial \underline{\Phi}}{\partial t} & = \nabla \cdot \left(\underline{\underline{D}}\nabla\underline{\Phi}\right) - \underline{\underline{\Sigma}}_a\underline{\Phi} - \underline{\underline{\Sigma}}_s\underline{\Phi} + (1-\beta)\underline{\chi}_p\underline{F}^T\underline{\Phi} + \sum_m \lambda_m \underline{\chi}_m C_m \\ \dfrac{\partial C_m(t)}{\partial t} & = -\lambda_m C_m(t) + \beta_m \underline{F}^T\underline{\Phi} \qquad \text{for } m = 1, \cdots, 8 \end{cases}$$

Mathematical challenges

- Set of 10 coupled parabolic PDEs
- Reaction dominated time dependent (small Δt)
- Non-symmetric
- Non-coercive (small β)

Essential ingredients of Reduced Basis

1. Affine decomposition of the *parametrized* FE bilinear forms

$$a(u, v; \mu) = \sum_q \Theta^q(\mu) a^q(u, v)$$

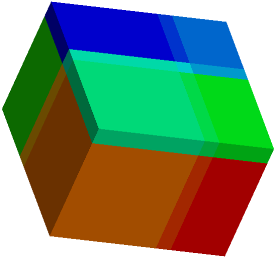
2. Galerkin projection onto an *ad hoc* built basis $\mathcal{Z} = \left[\xi_1^{\mathcal{N}} | \ldots | \xi_N^{\mathcal{N}}\right]$

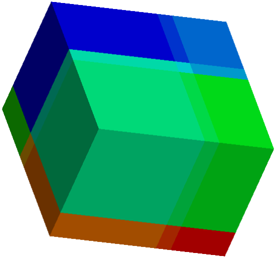
$$u_{\mathcal{N}} \rightarrow u_N = \sum_{i=1}^{N} u_{N,i}(\mu) \xi_i^{\mathcal{N}} \quad (1)$$

$$a^q(u, v) \rightarrow A_N^q(u, v) = \mathcal{Z}^T a^q(u, v) \mathcal{Z} \quad (2)$$

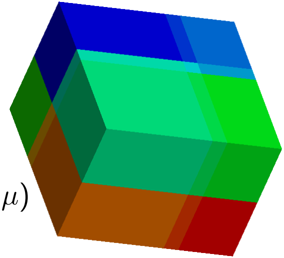
3. Rigorous *a posteriori* error estimation w.r.t. high-fidelity FE

How to get $\Theta^q(\mu)$





$$\Omega^l_o(\mu) = \mathcal{T}^l(\Omega^l;\mu)$$



How to get $\Theta^q(\mu)$

Original and reference subdomains are linked *via* a mapping $\mathcal{T}(\cdot;\mu)$:

$$\Omega_o^l(\mu) = \mathcal{T}^l(\Omega^l;\mu) \quad 1 \le l \le L_{\mathrm{dom}} \tag{3}$$

where

$$\mathcal{T}_i^l(\mathbf{x};\mu) = C_i^l(\mu) + \sum_{j=1}^{d} G_{ij}^l(\mu) x_j, \qquad 1 \le i \le d \tag{4}$$

The following terms can be defined

$$J^l(\mu) = |\det(G^l(\mu))|, \qquad 1 \le l \le L_{\mathrm{dom}} \tag{5}$$

$$D^l(\mu) = (G^l(\mu))^{-1}, \qquad 1 \le l \le L_{\mathrm{dom}} \tag{6}$$

$$\mathcal{G}^l(\mu) = \begin{bmatrix} D^l(\mu) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \qquad 1 \le l \le L_{\mathrm{dom}} \tag{7}$$

How to get $\Theta^q(\mu)$

The problem can be formulated *on the original domain* as follows

$$a_o(w, v; \mu) = \sum_{l=1}^{L_{\mathrm{dom}}} \int_{\Omega_o^l(\mu)} \left[\frac{\partial w}{\partial x} \ \frac{\partial w}{\partial y} \ w \right] \mathcal{K}_o^l(\mu) \begin{bmatrix} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \\ v \end{bmatrix} \tag{8}$$

The problem can be reformulated *on the reference domain* as follows

$$a(w, v; \mu) = \sum_{l=1}^{L_{\mathrm{dom}}} \int_{\Omega^l} \left[\frac{\partial w}{\partial x} \ \frac{\partial w}{\partial y} \ w \right] \mathcal{K}^l(\mu) \begin{bmatrix} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \\ v \end{bmatrix} \tag{9}$$

$$\mathcal{K}^l(\mu) = J^l(\mu)\, \mathcal{G}^l(\mu)\, \mathcal{K}_o^l(\mu) \ \left(\mathcal{G}^l(\mu)\right)^T \tag{10}$$

How to get $\Theta^q(\mu)$

Definition

$$a(w, v; \mu) = \sum_q \Theta^q(\mu) a^q(w, v)$$

The affine formulation can be derived by expanding the expression (9) in terms of the subdomains Ω_l and the different entries of $\mathcal{K}^l_{ij}$ leading to

$$a(w, v; \mu) = \underbrace{\mathcal{K}^1_{11}(\mu)}_{\Theta^1(\mu)} \underbrace{\int_{\Omega_1} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1}}_{a^1(w,v)} + \underbrace{\mathcal{K}^1_{22}(\mu)}_{\Theta^2(\mu)} \underbrace{\int_{\Omega_1} \frac{\partial w}{\partial x_2} \frac{\partial v}{\partial x_2}}_{a^2(w,v)} + \dots$$

Formulation on the *original* domain

$$
\begin{aligned}
m_l &= \int_{\Omega_o^l(\mu)} \left[\frac{1}{\mathrm{v}_1} \Phi_1 \psi_{\Phi_1} + \frac{1}{\mathrm{v}_2} \Phi_2 \psi_{\Phi_2} + \sum_{i=1}^{8} c_i \psi_{c_i} \right] \\
a_l^1 &= \int_{\Omega_o^l(\mu)} \left[D_1 \frac{\partial \Phi_1}{\partial x} \frac{\partial \psi_{\Phi_1}}{\partial x} + D_2 \frac{\partial \Phi_2}{\partial x} \frac{\partial \psi_{\Phi_2}}{\partial x} \right] \\
&+ \int_{\Omega_o^l(\mu)} \left[D_1 \frac{\partial \Phi_1}{\partial y} \frac{\partial \psi_{\Phi_1}}{\partial y} + D_2 \frac{\partial \Phi_2}{\partial y} \frac{\partial \psi_{\Phi_2}}{\partial y} \right] \\
&+ \int_{\Omega_o^l(\mu)} \left[\Sigma_{a_1} + \Sigma_{s_{1\to 2}} - (1-\beta)\nu\Sigma_{f_1} \right] \Phi_1 \psi_{\Phi_1} - \int_{\Omega_o^l(\mu)} \left[\Sigma_{s_{2\to 1}} + (1-\beta)\nu\Sigma_{f_2} \right] \Phi_2 \psi_{\Phi_1} \\
&- \int_{\Omega_o^l(\mu)} \sum_{i=1}^{8} \lambda_i c_i \psi_{\Phi_1} - \int_{\Omega_o^l(\mu)} \Sigma_{s_{1\to 2}} \Phi_1 \psi_{\Phi_2} + \int_{\Omega_o^l(\mu)} \left[\Sigma_{a_2} + \Sigma_{s_{2\to 1}} \right] \Phi_2 \psi_{\Phi_2} \\
&- \int_{\Omega_o^l(\mu)} \sum_{i=1}^{8} \beta_i \nu \Sigma_{f_1} \Phi_1 \psi_{c_i} - \int_{\Omega_o^l(\mu)} \sum_{i=1}^{8} \beta_i \nu \Sigma_{f_2} \Phi_2 \psi_{c_i} + \int_{\Omega_o^l(\mu)} \sum_{i=1}^{8} \lambda_i c_i \psi_{c_i} \\
a_l^2 &= \int_{\Omega_o^l(\mu)} \left[D_1 \frac{\partial \Phi_1}{\partial z} \frac{\partial \psi_{\Phi_1}}{\partial z} + D_2 \frac{\partial \Phi_2}{\partial z} \frac{\partial \psi_{\Phi_2}}{\partial z} \right]
\end{aligned}
$$

A greedy algorithm

compute $u_{\mathcal{N}}(\boldsymbol{\mu}_1)$;
$X_1^{RB} = \mathrm{span}\{u_{\mathcal{N}}(\boldsymbol{\mu}_1)\}$;
for $N = 2 : N_{\mathrm{max}}$
    $\boldsymbol{\mu}_N = \arg\max_{\boldsymbol{\mu} \in \Xi_{\mathrm{train}}} \|u_{N-1}(\boldsymbol{\mu}) - u_{\mathcal{N}}(\boldsymbol{\mu})\|$;
    $\varepsilon_{N-1} = \|u_{N-1}(\boldsymbol{\mu}) - u_{\mathcal{N}}(\boldsymbol{\mu})\|$;
    if $\varepsilon_{N-1} \leq \varepsilon^{*}_{\mathrm{tol}}$
        $N_{\mathrm{max}} = N - 1$;
        break;
    end
    compute $u_{\mathcal{N}}(\boldsymbol{\mu}_N)$;
    $X_N^{RB} = X_{N-1}^{RB} \cup \mathrm{span}\, u_{\mathcal{N}}(\boldsymbol{\mu}_N)$;
end

A greedy algorithm

compute $u_{\mathcal{N}}(\boldsymbol{\mu}_1)$;
$X_1^{RB} = \mathrm{span}\{u_{\mathcal{N}}(\boldsymbol{\mu}_1)\}$;
for $N = 2 : N_{\max}$
  $\boldsymbol{\mu}_N = \arg\max_{\boldsymbol{\mu} \in \Xi_{\mathrm{train}}} \Delta_{N-1}(\boldsymbol{\mu})$;
  $\varepsilon_{N-1} = \Delta_{N-1}(\boldsymbol{\mu})$;
  if $\varepsilon_{N-1} \leq \varepsilon^*_{\mathrm{tol}}$
    $N_{\max} = N - 1$;
    break;
  end
  compute $u_{\mathcal{N}}(\boldsymbol{\mu}_N)$;
  $X_N^{RB} = X_{N-1}^{RB} \cup \mathrm{span}\, u_{\mathcal{N}}(\boldsymbol{\mu}_N)$;
end

A POD-greedy algorithm

set $\mathcal{Z} = \emptyset$;
compute $\left\{u_{\mathcal{N}}^{k}(\boldsymbol{\mu}^*), 1 \leq k \leq K\right\}$;
while $N \leq N_{\max}$
  $\{\chi_m, 1 \leq m \leq M_1\} = \mathrm{POD}\left(\left\{u_{\mathcal{N}}^{k}(\boldsymbol{\mu}^*), 1 \leq k \leq K\right\}, M_1\right)$;
  $\mathcal{Z} \leftarrow \{\mathcal{Z}, \{\chi_m, 1 \leq m \leq M_1\}\}$;
  $N \leftarrow N + M_2$;
  $\{\xi_n, 1 \leq n \leq N\} = \mathrm{POD}\,(\mathcal{Z}, N)$;
  $X_N^{RB} = \mathrm{span}\,\{\xi_n, 1 \leq n \leq N\}$;
  $\boldsymbol{\mu}^* = \arg\max_{\boldsymbol{\mu} \in \Xi_{\mathrm{train}}} \Delta_N^k|_{k=K}(\boldsymbol{\mu}^*)$;
  $\varepsilon_N = \Delta_N^k|_{k=K}(\boldsymbol{\mu}^*)$;
  if $\varepsilon_N \leq \varepsilon_{\mathrm{tol}}$
    break ;
  end
end

Error bounds

$$\|u_N(t^k;\mu) - u_{\mathcal{N}}(t^k;\mu)\| \leq \Delta_N(t^k;\mu) \quad 1 \leq k \leq K$$

$$\Delta_N(t^k;\mu) = \left(\frac{\Delta t}{\alpha_{\mathrm{LB}}^{\mathcal{N}}(\mu)} \sum_{m=1}^{k} \varepsilon_N^2(t^m;\mu) \right)^{1/2}$$

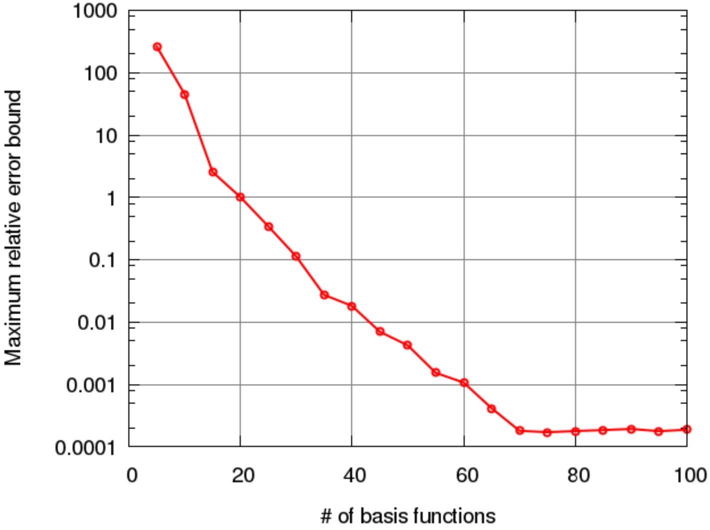
$\varepsilon_N(t^k;\mu)$ is the dual norm of the residual $r_N(v;t^k;\mu)$

$$\varepsilon_N(t^k;\mu) = \sup_{v \in X^{\mathcal{N}}} \frac{r_N(v;t^k;\mu)}{\|v\|_X}, \quad 1 \leq k \leq K$$

where $r_N(v;t^k;\mu)$ is the residual associated with the RB approximation

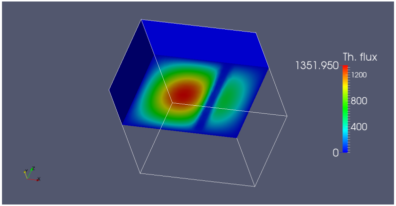
$$\begin{aligned} r_N(v;t^k;\mu) = f(v) - \frac{1}{\Delta t} m\left(u_N^k(\mu) - u_N^{k-1}(\mu), v;\mu\right) \\ - a(u_N^k(\mu), v;\mu), \quad \forall v \in X^{\mathcal{N}}, 1 \leq k \leq K \end{aligned} \tag{11}$$

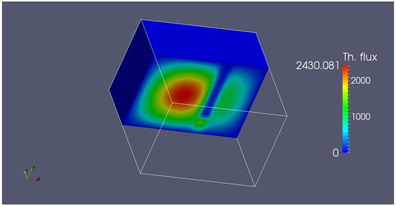
Offline phase



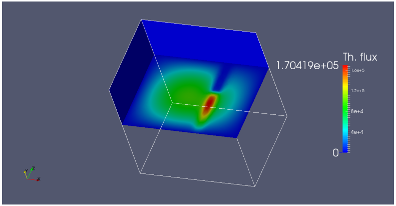
Online phase – ROM

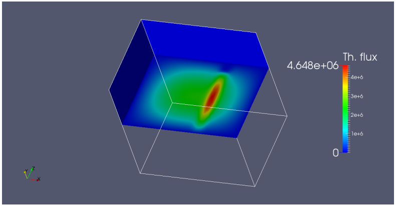
Neutron flux prediction [1/cm^2 s]





$\mu = -16\,\text{cm}$ $\qquad$ $\mu = -10\,\text{cm}$





$\mu = 3\,\text{cm}$ $\qquad$ $\mu = 13\,\text{cm}$

Online phase – statistics

Relative error bounds of the neutron flux, at the last time step.

	Error bounds
$\mu = -16\,\text{cm}$	9.36×10^{-6}
$\mu = -10\,\text{cm}$	2.48×10^{-4}
$\mu = 3\,\text{cm}$	4.65×10^{-4}
$\mu = 13\,\text{cm}$	1.85×10^{-4}

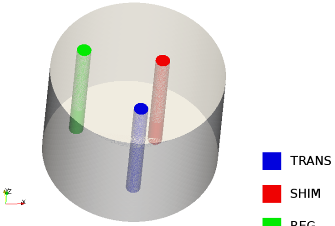
Effectivity: ratio between error bound and true error.

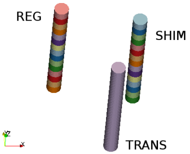
Average	Maximum	Minimum
16.749	52.826	3.574

Computational time per cpu per time step.

	Truth	ROM	Speed-up
$\mu = -16\,\text{cm}$	148.35 s	2.2 ms	67 432
$\mu = -10\,\text{cm}$	148.02 s	2.1 ms	70 486
$\mu = 3\,\text{cm}$	148.68 s	2.5 ms	59 472
$\mu = 13\,\text{cm}$	148.72 s	2.4 ms	61 967

Alternative approach





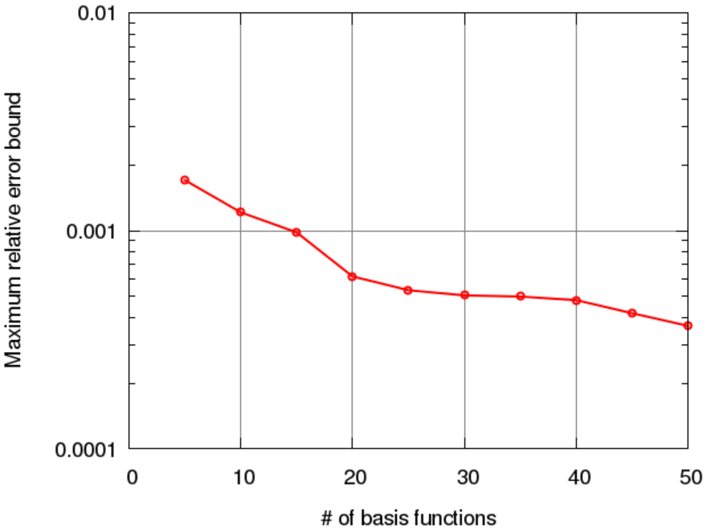
Parametrized formulation

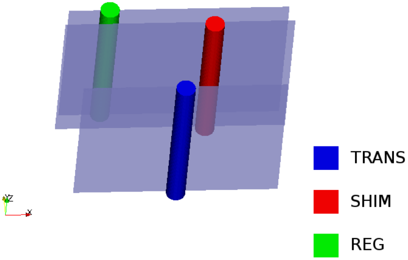
Within each subdomain of the control rods

$$
\begin{aligned}
m_l &= \mu_l\, m_r + (1-\mu_l)\, m_w, \quad \mu_l = 0 \text{ or } 1 \\
a_l &= \mu_l\, a_r + (1-\mu_l)\, a_w, \quad \mu_l = 0 \text{ or } 1
\end{aligned}
$$

- 31 m_l
- 31 a_l

Offline phase

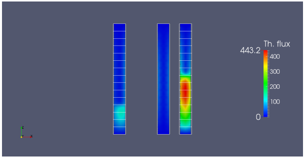




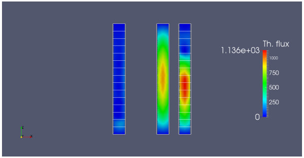
Employed sections to visualize the thermal neutron flux.

Online phase – ROM

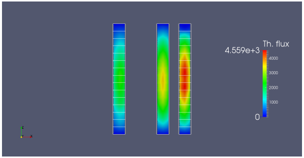
Neutron flux prediction [1/cm^2 s]



TRANS is inserted, and the height of the REG and SHIM is 4 and 8 “steps”, respectively



TRANS is withdrawn, and the height of the REG and SHIM is 2 and 10 “steps”, respectively



TRANS is withdrawn, and the height of the REG and SHIM is 16 and 16 “steps”, respectively

Online phase – statistics

Relative error bounds in L^2 norm of the neutron flux, at the last time step.

$\mu = \{\text{TRANS, REG, SHIM}\}$	Error bounds
$\mu = \{0, 4, 8\}$	6.02×10^{-5}
$\mu = \{1, 2, 10\}$	1.92×10^{-4}
$\mu = \{1, 16, 16\}$	1.14×10^{-4}

Effectivity: ratio between the error bound and true error.

Average	Maximum	Minimum
4.768	10.480	1.670

Computational time per cpu per time step.

$\mu = \{\text{TRANS, REG, SHIM}\}$	Truth	ROM	Speed-up
$\mu = \{0, 4, 8\}$	191.0 s	110 ms	1736
$\mu = \{1, 2, 10\}$	190.4 s	110 ms	1730
$\mu = \{1, 16, 16\}$	191.1 s	112 ms	1705

Next step

Accomplishments

- First RB for control rod movement
- Time dependent
- Two approaches
- Speed-up $>$ 3 orders of magnitude
- Good accuracy

Quasi-static approach

- Parametrized generalized eigenvaule calculation
- Reactivity from ROM
- Black-box

Modelling approach

- TRIGA Mark II nuclear reactor
- finite element method
- reduced basis method

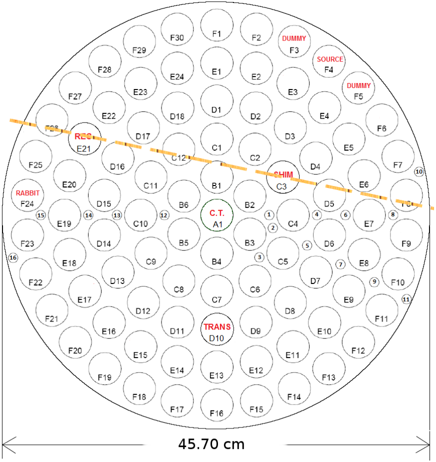
Stationary neutron diffusion equation: generalized eigenvalue problem

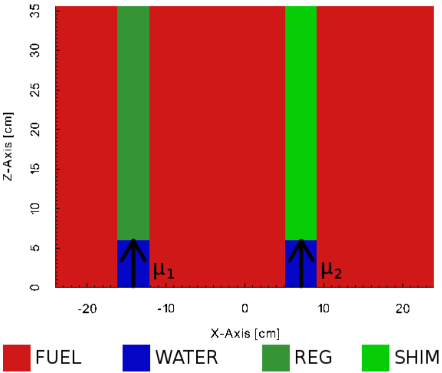
$$\left(-\nabla \cdot \underline{\underline{D}}\nabla + \underline{\underline{\Sigma_a}} + \underline{\underline{\Sigma_s}}\right) \underline{\phi} = \lambda_{\min} \underline{\chi_t} \underline{F}^T \underline{\phi}$$

Challenges

- high accuracy (up to pcm)
- no greedy algorithm

Parametrization

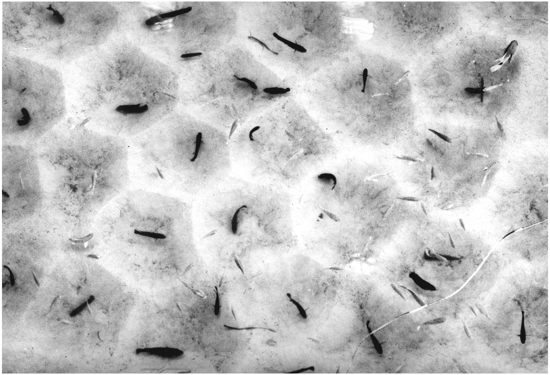




How to efficiently explore the parameter space?

- no efficient a posterior error estimation available
- no greedy algorithm

Centroidal Voronoi Tessellation

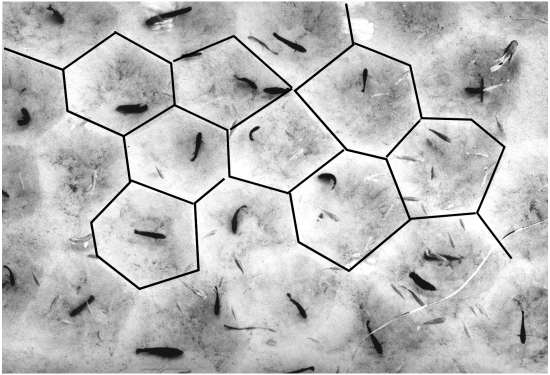


$$V_i = \{\mathbf{x} \in \Omega \quad | \quad |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \ldots, k,\ j \neq i\}$$

How to efficiently explore the parameter space?

- no efficient a posterior error estimation available
- no greedy algorithm

Centroidal Voronoi Tessellation

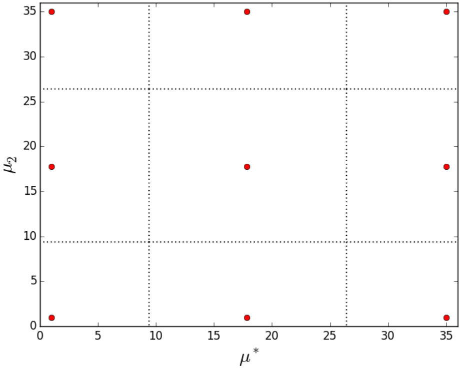


$$V_i = \{\mathbf{x} \in \Omega \quad | \quad |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \dots, k,\ j \neq i\}$$

How to compute $\mathbf{z}_i$?

$$V_i = \{\mathbf{x} \in \Omega \quad | \quad |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \ldots, k,\ j \neq i\}$$

Given an initial set of (μ_1, μ_2)
Compute the snapshots



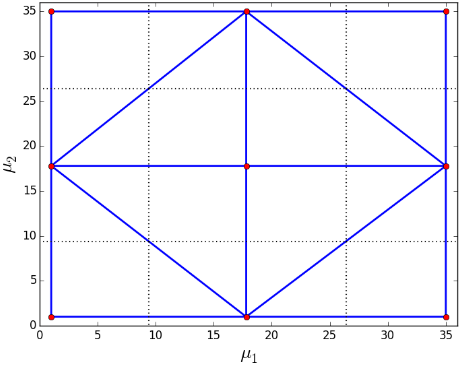
How to compute $\mathbf{z}_i$?

$$V_i = \{\mathbf{x} \in \Omega \quad | \quad |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \dots, k, \; j \neq i\}$$

Compute a Delaunay triangulation

Compute the residuals in each vertex

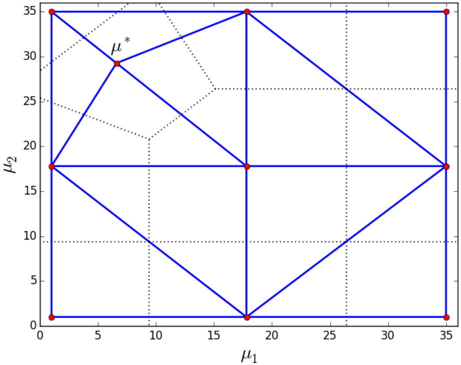
For each triangle, compute the sum of the residuals times the area



How to compute $\mathbf{z}_i$?

$$V_i = \{\mathbf{x} \in \Omega \quad | \quad |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \ldots, k, \ j \neq i\}$$

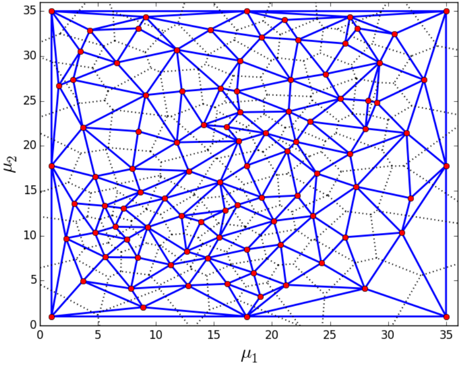
Next μ^* is the barycenter of the "worst" element



How to compute $\mathbf{z}_i$?

$$V_i = \{\mathbf{x} \in \Omega \quad | \quad |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \ldots, k, \; j \neq i\}$$

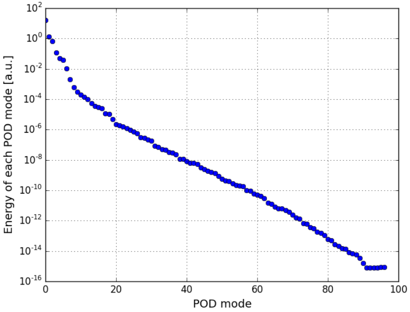
Repeat



How to compute $\mathbf{z}_i$?

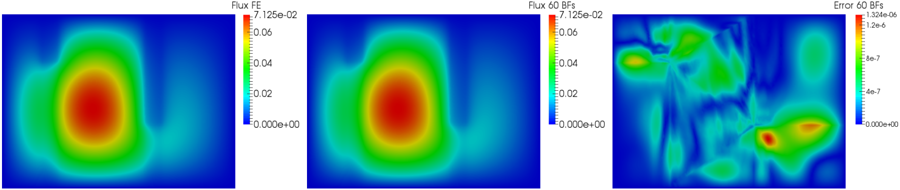
$$V_i = \{\mathbf{x} \in \Omega \quad | \quad |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j| \quad \text{for } j = 1, \dots, k,\ j \neq i\}$$

POD on the vector of the snapshots



Representative results

Normalized neutron flux [a.u.]



Ref. | Reduced | Spatial error

	Computational time			
N	Finite Element	ROM	Speed-up	$\lvert\rho^{\mathcal{N}} - \rho^{N}\rvert$
10	33.423 s	5.7 ms	$\sim$ 5800	196.998 pcm
20	33.423 s	6.4 ms	$\sim$ 5200	19.134 pcm
40	33.423 s	9.4 ms	$\sim$ 3500	1.551 pcm
60	33.423 s	16.2 ms	$\sim$ 2000	0.436 pcm

Next step

Accomplishments

- Several ROMs for handling spatial effects
- Set of parametrized PDEs
- “Only” neutronics

What about multiphysics? Is a fast-running MP simulation tool achievable?

➠ Proof of concept of a parametrized MP LFR single channel

➠ Strategy for handling the coupling for Offline/Online split

➠ Develop a ROM

NUCLEAR REACTORS GROUP

LFR - Single channel analysis

Neutronics

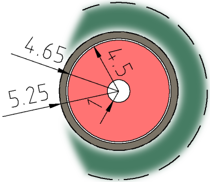
- Six-group diffusion equation
- $\Sigma(T,\rho) = \frac{\rho}{\rho_0}\left[\Sigma_0 + \alpha \log\left(\frac{T}{T_0}\right)\right]$

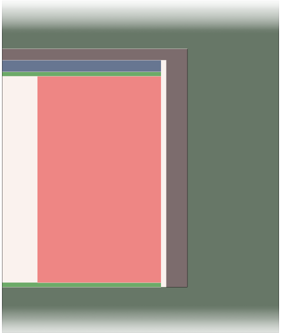
Thermo-hydraulics

- $k - \varepsilon$ turbolence model
- Heat source from neutron flux

Thermo-mechanics

- Linear elasticity
- Moving mesh





Aufiero et al. Nuclear Engineering and Design, 256 (2013), 14-27.

Parametrized LFR single channel

Neutronics

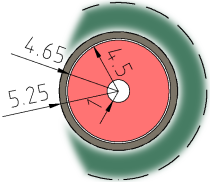
- Six-group diffusion equation
- $\Sigma(T,\rho) = \frac{\rho}{\rho_0}\left[\Sigma_0 + \alpha \log\left(\frac{T}{T_0}\right)\right]$

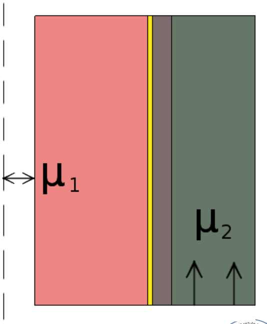
Thermo-hydraulics

- Uniform velocity profile
- K_T from previous LFR MP model
- Heat source from neutron flux

Parameters

- Pellet hole radius $\mu_1 = (1.0, 4.3)$
- Lead inlet velocity $\mu_2 = (0.8, 1.6)$





How to achieve an Offline/Online split

Abstract formulation

$$a(\phi, v; \mu, \Sigma(T)) = \lambda_{\min} m(\phi, v; \mu, \Sigma(T)), \quad \forall v \in V^{\mathcal{N}} \tag{12}$$

$$d(T, w; \mu) = f(w; \mu, S(\phi)), \quad \forall w \in W^{\mathcal{N}} \tag{13}$$

First step: affine decomposition on μ

$$a(\phi, v; \mu, \Sigma(T)) = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) a^q(\phi, v; \Sigma(T)) \tag{14}$$

$$m(\phi, v; \mu, \Sigma(T)) = \sum_{q=1}^{Q_m} \Theta_m^q(\mu) m^q(\phi, v; \Sigma(T)) \tag{15}$$

How to achieve an Offline/Online split

Abstract formulation

$$a\left(\phi, v; \mu, \Sigma(T)\right) = \lambda_{\min} m\left(\phi, v; \mu, \Sigma(T)\right), \quad \forall v \in V^{\mathcal{N}} \tag{12}$$

$$d(T, w; \mu) = f\left(w; \mu, S(\phi)\right), \quad \forall w \in W^{\mathcal{N}} \tag{13}$$

First step: affine decomposition on μ

$$a\left(\phi, v; \mu, \Sigma(T)\right) = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) a^q\left(\phi, v; \Sigma(T)\right) \tag{14}$$

$$m\left(\phi, v; \mu, \Sigma(T)\right) = \sum_{q=1}^{Q_m} \Theta_m^q(\mu) m^q\left(\phi, v; \Sigma(T)\right) \tag{15}$$

Non-linear terms

Representative term

$$c(T^{N_T}, \Phi^{N_f}, v) = \int_{\Omega} \Sigma(T^{N_T})\, \Phi^{N_f}\, v \,\mathrm{d}\mathbf{x} \tag{16}$$

$$\Sigma(T^{N_T}) = \sum_{q=1}^{N_T} \sigma_q\, \xi_q^T \tag{17}$$

Non-linear terms

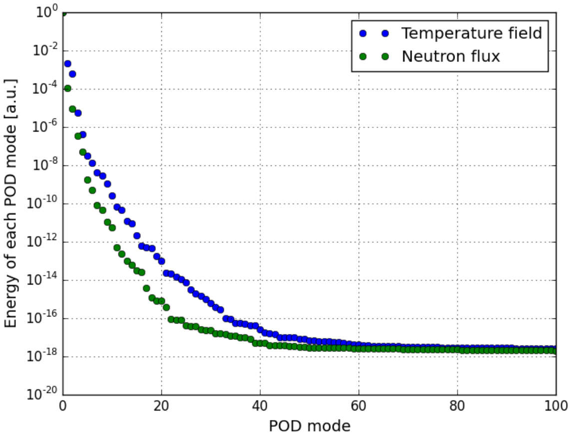
Representative term

$$c(T^{N_T}, \Phi^{N_f}, v) = \int_\Omega \Sigma(T^{N_T})\, \Phi^{N_f}\, v \,\mathrm{d}\mathbf{x} \tag{16}$$

$$\begin{aligned} c(T^{N_T}, \Phi^{N_f}, v) &= \int_\Omega \sum_{q=1}^{N_T} \sigma_q\, \xi_q^T\, \Phi^{N_f}\, v \,\mathrm{d}\mathbf{x} = \sum_{q=1}^{N_T} \sigma_q \int_\Omega \xi_q^T\, \Phi^{N_f}\, v \,\mathrm{d}\mathbf{x} \\ &= \sum_{q=1}^{N_T} \sigma_q\, c^q\left(\xi_q^T\right) \end{aligned} \tag{17}$$

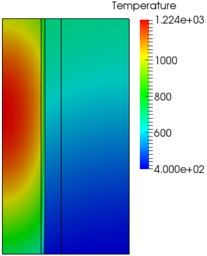
$$C_N^q = \mathcal{Z}_f' C_{\mathcal{N}}^q \mathcal{Z}_f \tag{18}$$

Basis functions

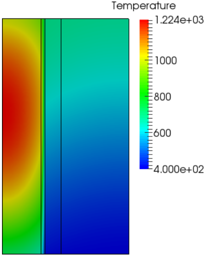


Representative results

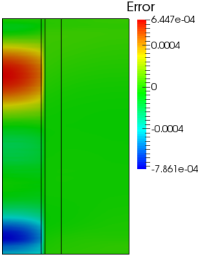
Temperature field [C°]



Ref.



Reduced



Spatial error

	Computational time				
Plot	Finite Element	ROM	Speed-up	Error λ	Error OI
NO	∼60 min	∼0.3 s	∼ 11 800	2.2031 pcm	3.1615e-7
YES	∼60 min	∼2.3 s	∼ 1500	2.2031 pcm	3.1615e-7

Conclusions

✓ Focus on methodological approach

✓ First RB in nuclear engineering

✓ Control rod movement
  - Time dependent
  - Eigenvalue problem

✓ Preliminary Multi-physics ROM

Future developments

- More detailed description of TRIGA Mark II
- Multi-physics RB with thermal-expansion effects
- Reduced Basis Element Method (RBEM)
- Uncertainty quantification
- “Exotic” reduced order models (e.g., SERPENT + OpenFOAM)

Thank you for your kind attention!

POD as a minimization problem

- Given multiple snapshots: $\{y_j\}_{j=1}^{n} \subset X$

Proper Orthogonal Decomposition

for any $\ell \in \{1, \ldots, n\}$ solve

$$\min \sum_{j=1}^{n} \alpha_j \| y_j - \sum_{i=1}^{\ell} \langle y_j, \psi_i \rangle_X \, \psi_i \|_X^2 \quad \text{s.t.} \quad \{\psi_i\}_{i=1}^{\ell} \subset X, \langle \psi_i, \psi_j \rangle_X = \delta_{ij}, \alpha_j > 0$$

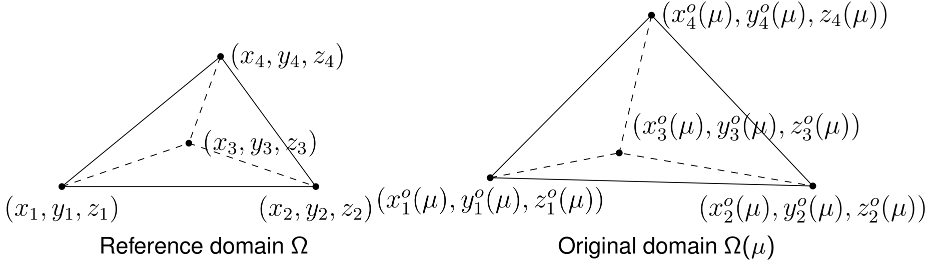
Method of the snapshots

- Given a discrete set of train $\Xi_{\text{train}}(\underline{\underline{D}}, \underline{\underline{\Sigma_a}}, \underline{\underline{\Sigma_s}})$
- Collect N snapshots $s_1(\boldsymbol{\mu}_1), \ldots, s_N(\boldsymbol{\mu}_N),\ \boldsymbol{\mu}_i \in \Xi_{\text{train}}\ i = 1, \ldots, N$
- Build a vector of the snapshots $X = [s_1, \ldots, s_N]$
- Perform the *singular value decomposition* on C, we obtain

$$X = U\,S\,V$$

- The columns of U are the POD orthonormal basis
- The *energy* of each POD is proportional to the corresponding singular value

How to compute the affine transformation



How to compute the affine transformation

$$\mathbb{B} = \begin{bmatrix} 1 & 0 & 0 & x_1 & y_1 & z_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 \\ 1 & 0 & 0 & x_2 & y_2 & z_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 \\ 1 & 0 & 0 & x_3 & y_3 & z_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x_3 & y_3 & z_3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & x_3 & y_3 & z_3 \\ 1 & 0 & 0 & x_4 & y_4 & z_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x_4 & y_4 & z_4 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & x_4 & y_4 & z_4 \end{bmatrix}, \quad \mathbb{V}(\mu) = \begin{bmatrix} x_1^o(\mu) \\ y_1^o(\mu) \\ z_1^o(\mu) \\ x_2^o(\mu) \\ y_2^o(\mu) \\ z_2^o(\mu) \\ x_3^o(\mu) \\ y_3^o(\mu) \\ z_3^o(\mu) \\ x_4^o(\mu) \\ y_4^o(\mu) \\ z_4^o(\mu) \end{bmatrix}, \tag{19}$$

$$\begin{aligned} &[C_1(\mu),\ C_2(\mu), C_3(\mu),\ G_{11}(\mu),\ G_{12}(\mu),\ G_{13}(\mu),\ G_{21}(\mu),\ G_{22}(\mu),\ G_{23}(\mu), \\ &G_{31}(\mu),\ G_{32}(\mu),\ G_{33}(\mu)]^T = \mathbb{B}^{-1}\,\mathbb{V}(\mu). \end{aligned} \tag{20}$$

NUCLEAR REACTORS GROUP

Parabolic problem

High order model – weak formulation

$$\frac{1}{\Delta t} m(u^k(\mu), v; \mu) + a(u^k(\mu), v; \mu) = f(v) + \frac{1}{\Delta t} m(u^{k-1}(\mu), v; \mu), \quad \forall v \in X^{\mathcal{N}}, 1 \leq k \leq K$$

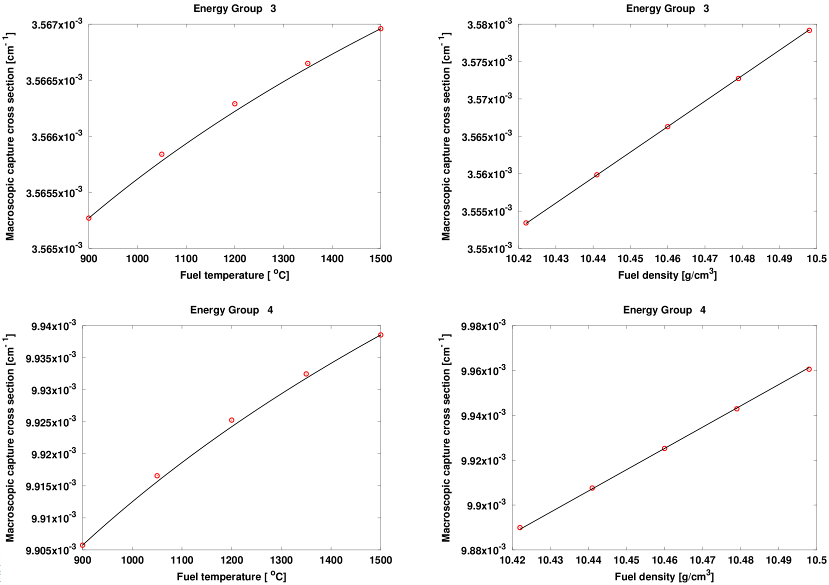
Reduced order model – algebraic formulation

$$\left[\frac{1}{\Delta t} \sum_{q=1}^{Q_m} \Theta_m^q(\mu) M_N^q + \sum_{q=1}^{Q_a} \Theta_a^q(\mu) A_N^q \right] u_N(t^k; \mu) = f_N + \frac{1}{\Delta t} \sum_{q=1}^{Q_m} \Theta_m^q(\mu) M_N^q \, u_N(t^{k-1}; \mu)$$

NUCLEAR REACTORS GROUP

SISSA

Validation of the cross-sections functional form



NUCLEAR REACTORS GROUP

SISSA