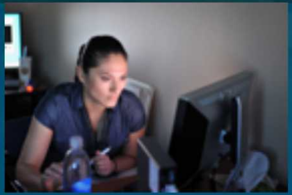
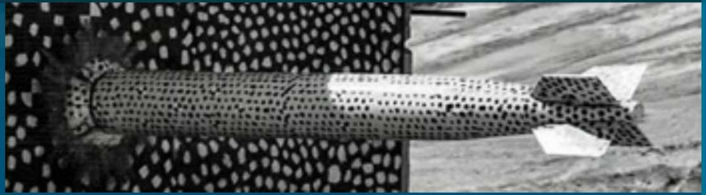
Sandia National Laboratories

SAND2018-8091C

Domain-decomposition least-squares Petrov-Galerkin (DD-LSPG) nonlinear model reduction





PRESENTED BY

Chi Hoang and Kevin Carlberg

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Motivation for DDROM

- Typical ROMs can work very well, e.g., classical RB, POD, PGD, DMD techniques. However, training typical ROMs can be very burdensome for decomposable engineering systems:

  - Require training simulations for the full-system, which is very costly.

  - If a full-system is based on components that can be assembled in different ways, require training for each full-system configuration.

- Main idea: a 'divide and conquer' approach that

  (1) Constructs reduced basis locally for different components/subdomains of the full-system.

  (2) Solve the resulting full-system ROM using ideas from non-overlapping domain decomposition.

  \+ Subdomains/components can be trained separately.

  \+ Full-system ROMs can be assembled from components in arbitrary ways (i.e., Lego blocks).

  \+ Can enforce weak compatibility between subdomains, which mitigates the need for matching meshes, for example

  \+ Applicable to nonlinear systems.

  \+ Multiple different solvers that expose different levels of parallelism.

Literature reviews for DDROM

- Parameterized linear static PDE: Galerkin projection, no hyper-reduction

  RDF [Iapichino 2016]: parametric BCs on local subdomains (heat conduction)

  LRBMS [Ohlberger 2015]: weak constraint with DG (multiscale material homogenization)

  SCRBE [Huynh 2013, Huynh 2015]: strong constraint (heat conduction, structural analysis)

  RBHM [Iapichino 2012]: strong constraint (Stokes equation, cardiovascular network)

  RBEM [Maday 2002]: weak constraint with Lagrange multiplier (potential flows analysis)

- Parameterized nonlinear static PDE: Galerkin projection, hybrid FOM+ROM

  [Kerfriden 2013]: strong constraint (nonlinear fracture mechanics)

  [Baiges 2013]: strong constraint (Navier-Stokes equation with hyper-reduction)

  [Buffoni 2009]: strong constraint (compressible Euler equation)

and many other works.

Key attributes of our proposed methodology

1. Applicable to nonlinear systems.
2. Hyper-reduction is enabled.
3. Subdomain LSPG ROMs can be constructed completely independently (i.e., tailored basis, hyper-reduction).
4. Global problem constructed by exploiting the optimization structure of LSPG ROMs: formulate a nonlinear least squares problem with linear equality constraints associated with possibly weak compatibility.
5. Weak compatibility enables the interface reduced bases to not be perfectly matched. This implies that no global ROM exists! But it ends up giving the best performance.
6. Different kinds of basis functions on the interfaces of subdomains (port, skeleton, and interface) are proposed.
7. The decomposed structure allows for very efficient numerical solvers that can exploit the localized structure: primal-dual monolithic, primal-dual Schur complement, primal monolithic, primal Schur complement, nonlinear primal solvers.
8. In the online stage, each solver composed of assembly and solving steps. The assembly step can be done in parallel, while the solving step is done as whole.

Problem settings

Global FOM approximation: $\boldsymbol{r}(\boldsymbol{x}) = \mathbf{0}$

with residual $\boldsymbol{r} : \mathbb{R}^n \to \mathbb{R}^n$, state $\boldsymbol{x} \in \mathbb{R}^n$

DDFOM re-formulation:

$$\boldsymbol{r}(\boldsymbol{x}) = \sum_{i=1}^{n_\Omega} [\boldsymbol{P}_i^{\boldsymbol{r}}]^T \boldsymbol{r}_i(\boldsymbol{P}_i^\Omega \boldsymbol{x}, \boldsymbol{P}_i^\Gamma \boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^n$$

$$\boldsymbol{x}_i^\Omega := \boldsymbol{P}_i^\Omega \boldsymbol{x} \in \mathbb{R}^{n_i^\Omega}, \ \boldsymbol{x}_i^\Gamma := \boldsymbol{P}_i^\Gamma \boldsymbol{x} \in \mathbb{R}^{n_i^\Gamma}, \ \boldsymbol{x}_i := (\boldsymbol{x}_i^\Omega, \boldsymbol{x}_i^\Gamma)$$

Define a set of "ports", the *j*th port is $P(j)$, compatibility conditions:

$$[\boldsymbol{P}_i^j]^T \boldsymbol{x}_i^\Gamma = [\boldsymbol{P}_l^j]^T \boldsymbol{x}_l^\Gamma, \quad i, l \in P(j)$$

$$\boldsymbol{r}_i(\boldsymbol{x}_i^\Omega, \boldsymbol{x}_i^\Gamma) = 0, \quad i = 1, \ldots, n_\Omega \ \text{subject to} \ \sum_{i=1}^{n_\Omega} \bar{\boldsymbol{A}}_i \boldsymbol{x}_i^\Gamma = 0$$

Reduced order models

DDROM approximation:

Introduce reduced bases: $\boldsymbol{\Phi}_i^{\Omega} \in \mathbb{R}^{n_i^{\Omega} \times p_i^{\Omega}}, \boldsymbol{\Phi}_i^{\Gamma} \in \mathbb{R}^{n_i^{\Gamma} \times p_i^{\Gamma}}, i = 1, \ldots, n_{\Omega}$

Solution approximation: $\boldsymbol{x}_i \approx \tilde{\boldsymbol{x}}_i = (\tilde{\boldsymbol{x}}_i^{\Omega}, \tilde{\boldsymbol{x}}_i^{\Gamma}) = (\boldsymbol{\Phi}_i^{\Omega} \hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma})$

$$\underset{(\hat{\boldsymbol{x}}_1^{\Omega}, \ldots, \hat{\boldsymbol{x}}_{n_{\Omega}}^{\Omega}), (\hat{\boldsymbol{x}}_1^{\Gamma}, \ldots, \hat{\boldsymbol{x}}_{n_{\Omega}}^{\Gamma})}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{n_{\Omega}} \| \boldsymbol{r}_i(\boldsymbol{\Phi}_i^{\Omega} \hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma}) \|_2^2$$

$$\text{subject to} \quad \sum_{i=1}^{n_{\Omega}} \boldsymbol{A}_i \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma} = 0$$

DDGNAT approximation:

$$\underset{(\hat{\boldsymbol{x}}_1^{\Omega}, \ldots, \hat{\boldsymbol{x}}_{n_{\Omega}}^{\Omega}), (\hat{\boldsymbol{x}}_1^{\Gamma}, \ldots, \hat{\boldsymbol{x}}_{n_{\Omega}}^{\Gamma})}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{n_{\Omega}} \| \boldsymbol{\Phi}_i^{r} (\boldsymbol{Z}_i \boldsymbol{\Phi}_i^{r})^{+} \boldsymbol{Z}_i \boldsymbol{r}_i(\boldsymbol{\Phi}_i^{\Omega} \hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma}) \|_2^2$$

$$\text{subject to} \quad \sum_{i=1}^{n_{\Omega}} \boldsymbol{A}_i \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma} = 0$$

Reduced order models…

Interface basis functions types

"Port" basis function	"Skeleton" basis function	"Interface" basis function
1. Isolate global snapshots to ports. 2. Compute separate SVD for each port to form port basis functions (BFs). 3. Stack the port BFs to form interface BFs.	1. Isolate global snapshots to skeleton DOFs. 2. Compute SVD for the skeleton DOFs to create skeleton BFs. 3. Isolate the skeleton BFs to each subdomain interface. 4. Orthogonalize the above to form interface BFs.	1. Isolate global snapshots to subdomain interface. 2. Compute separate SVD for each interface to create interface BFs.

Constraint types

+ Strong constraint: $\boldsymbol{A}_i = \bar{\boldsymbol{A}}_i$

+ Weak constraint: $\boldsymbol{A}_i = \boldsymbol{C}_i \bar{\boldsymbol{A}}_i$

$$i = 1, \dots, n_\Omega$$

Note: ROM & GNAT offline is performed separately on each subdomain

Sequential Quadratic Programming method

Lagrangian:

$$L(\hat{\boldsymbol{x}}_1^{\Omega}, \hat{\boldsymbol{x}}_1^{\Gamma}, \ldots, \hat{\boldsymbol{x}}_{n_\Omega}^{\Omega}, \hat{\boldsymbol{x}}_{n_\Omega}^{\Gamma}, \boldsymbol{\lambda}) := \tfrac{1}{2} \textstyle\sum_{i=1}^{n_\Omega} \| \boldsymbol{r}_i(\boldsymbol{\Phi}_i^{\Omega} \hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma}) \|_2^2 + \textstyle\sum_{i=1}^{n_\Omega} \boldsymbol{\lambda}^T \boldsymbol{A}_i \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma}$$

Necessary optimality condition (KKT condition):

$$\hat{\boldsymbol{r}}_i^{\Omega}(\hat{\boldsymbol{x}}_i^{\Omega}, \hat{\boldsymbol{x}}_i^{\Gamma}) := (\boldsymbol{\Phi}_i^{\Omega})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Omega}} (\boldsymbol{\Phi}_i^{\Omega} \hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma})^T \boldsymbol{r}_i(\boldsymbol{\Phi}_i^{\Omega} \hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma}) = 0,$$

$$\hat{\boldsymbol{r}}_i^{\Gamma}(\hat{\boldsymbol{x}}_i^{\Omega}, \hat{\boldsymbol{x}}_i^{\Gamma}) := (\boldsymbol{\Phi}_i^{\Gamma})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Gamma}} (\boldsymbol{\Phi}_i^{\Omega} \hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma})^T \boldsymbol{r}_i(\boldsymbol{\Phi}_i^{\Omega} \hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma}) + (\boldsymbol{\Phi}_i^{\Gamma})^T \boldsymbol{A}_i^T \boldsymbol{\lambda} = 0,$$

$$\sum_{i=1}^{n_\Omega} \boldsymbol{A}_i \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma} = 0, \quad i = 1, \ldots, n_\Omega.$$

Primal-dual monolithic solver

Use Gauss-Newton approximation, one SQP iteration is defined as

$$
\begin{bmatrix}
\boldsymbol{H}_1^{\Omega\Omega}(\hat{\boldsymbol{x}}_1^{\Omega}, \hat{\boldsymbol{x}}_1^{\Gamma}) & \boldsymbol{H}_1^{\Omega\Gamma}(\hat{\boldsymbol{x}}_1^{\Omega}, \hat{\boldsymbol{x}}_1^{\Gamma}) & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{H}_1^{\Gamma\Omega}(\hat{\boldsymbol{x}}_1^{\Omega}, \hat{\boldsymbol{x}}_1^{\Gamma}) & \boldsymbol{H}_1^{\Gamma\Gamma}(\hat{\boldsymbol{x}}_1^{\Omega}, \hat{\boldsymbol{x}}_1^{\Gamma}) & \cdots & \mathbf{0} & \mathbf{0} & (\boldsymbol{\Phi}_1^{\Gamma})^T \boldsymbol{A}_1^T \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{H}_{n_\Omega}^{\Omega\Omega}(\hat{\boldsymbol{x}}_{n_\Omega}^{\Omega}, \hat{\boldsymbol{x}}_{n_\Omega}^{\Gamma}) & \boldsymbol{H}_{n_\Omega}^{\Omega\Gamma}(\hat{\boldsymbol{x}}_{n_\Omega}^{\Omega}, \hat{\boldsymbol{x}}_{n_\Omega}^{\Gamma}) & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{H}_{n_\Omega}^{\Gamma\Omega}(\hat{\boldsymbol{x}}_{n_\Omega}^{\Omega}, \hat{\boldsymbol{x}}_{n_\Omega}^{\Gamma}) & \boldsymbol{H}_{n_\Omega}^{\Gamma\Gamma}(\hat{\boldsymbol{x}}_{n_\Omega}^{\Omega}, \hat{\boldsymbol{x}}_{n_\Omega}^{\Gamma}) & (\boldsymbol{\Phi}_{n_\Omega}^{\Gamma})^T \boldsymbol{A}_{n_\Omega}^T \\
\mathbf{0} & \boldsymbol{A}_1 \boldsymbol{\Phi}_1^{\Gamma} & \cdots & \mathbf{0} & \boldsymbol{A}_{n_\Omega} \boldsymbol{\Phi}_{n_\Omega}^{\Gamma} & \mathbf{0}
\end{bmatrix}
\begin{bmatrix}
\boldsymbol{p}_1^{\Omega} \\ \boldsymbol{p}_1^{\Gamma} \\ \vdots \\ \boldsymbol{p}_{n_\Omega}^{\Omega} \\ \boldsymbol{p}_{n_\Omega}^{\Gamma} \\ \boldsymbol{p}^{\boldsymbol{\lambda}}
\end{bmatrix}
=
-\begin{bmatrix}
\hat{\boldsymbol{r}}_1^{\Omega}(\hat{\boldsymbol{x}}_1^{\Omega}, \hat{\boldsymbol{x}}_1^{\Gamma}) \\
\hat{\boldsymbol{r}}_1^{\Gamma}(\hat{\boldsymbol{x}}_1^{\Omega}, \hat{\boldsymbol{x}}_1^{\Gamma}) \\
\vdots \\
\hat{\boldsymbol{r}}_{n_\Omega}^{\Omega}(\hat{\boldsymbol{x}}_{n_\Omega}^{\Omega}, \hat{\boldsymbol{x}}_{n_\Omega}^{\Gamma}) \\
\hat{\boldsymbol{r}}_{n_\Omega}^{\Gamma}(\hat{\boldsymbol{x}}_{n_\Omega}^{\Omega}, \hat{\boldsymbol{x}}_{n_\Omega}^{\Gamma}) \\
\sum_{i=1}^{n_\Omega} \boldsymbol{A}_i \boldsymbol{\Phi}_i^{\Gamma} \hat{\boldsymbol{x}}_i^{\Gamma}
\end{bmatrix}
$$

Primal-dual monolithic solver…

where

$$\boldsymbol{H}_i^{\Omega\Omega}(\hat{\boldsymbol{x}}_i^{\Omega}, \hat{\boldsymbol{x}}_i^{\Gamma}) := (\boldsymbol{\Phi}_i^{\Omega})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Omega}} (\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Omega}} (\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma}) \boldsymbol{\Phi}_i^{\Omega}$$

$$\boldsymbol{H}_i^{\Omega\Gamma}(\hat{\boldsymbol{x}}_i^{\Omega}, \hat{\boldsymbol{x}}_i^{\Gamma}) := (\boldsymbol{\Phi}_i^{\Omega})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Omega}} (\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Gamma}} (\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma}) \boldsymbol{\Phi}_i^{\Gamma}$$

$$\boldsymbol{H}_i^{\Gamma\Gamma}(\hat{\boldsymbol{x}}_i^{\Omega}, \hat{\boldsymbol{x}}_i^{\Gamma}) := (\boldsymbol{\Phi}_i^{\Gamma})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Gamma}} (\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Gamma}} (\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma}) \boldsymbol{\Phi}_i^{\Gamma}$$

$$\boldsymbol{H}_i^{\Gamma\Omega}(\hat{\boldsymbol{x}}_i^{\Omega}, \hat{\boldsymbol{x}}_i^{\Gamma}) := (\boldsymbol{\Phi}_i^{\Gamma})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Gamma}} (\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma})^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Omega}} (\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma}) \boldsymbol{\Phi}_i^{\Omega}.$$

Update

$$\hat{\boldsymbol{x}}_i^{\Gamma} \leftarrow \hat{\boldsymbol{x}}_i^{\Gamma} + \alpha \boldsymbol{p}_i^{\Gamma}, \quad i = 1, \ldots, n_{\Omega}$$

$$\hat{\boldsymbol{x}}_i^{\Omega} \leftarrow \hat{\boldsymbol{x}}_i^{\Omega} + \alpha \boldsymbol{p}_i^{\Omega}, \quad i = 1, \ldots, n_{\Omega}$$

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \alpha \boldsymbol{p}^{\boldsymbol{\lambda}},$$

Primal-dual monolithic solver…

Algorithm 1: Assembling procedure of primal–dual monolithic solver

1: Update the ROM state;

2: Compute residuals $\boldsymbol{r}_i(\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma})$ and Jacobians $\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Omega}}(\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma})$, $\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{x}_i^{\Gamma}}(\boldsymbol{\Phi}_i^{\Omega}\hat{\boldsymbol{x}}_i^{\Omega}, \boldsymbol{\Phi}_i^{\Gamma}\hat{\boldsymbol{x}}_i^{\Gamma})$ from each subdomain;

3: Compute all terms in [Stationary condition] from each subdomain;

4: "Stamping" all terms into big system ;

Algorithm 1: Solving procedure of primal–dual monolithic solver

5: Solve the big system;

6: Extract search directions $\boldsymbol{p}_i^{\Omega}$, $\boldsymbol{p}_i^{\Gamma}$, $\boldsymbol{p}^{\boldsymbol{\lambda}}$;

7: Update solutions;

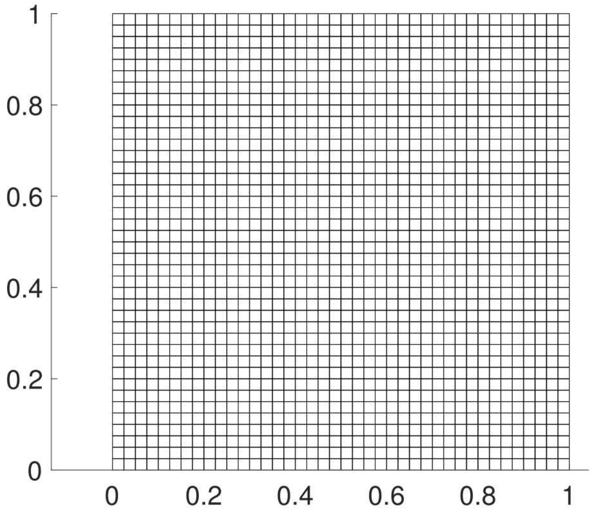
Numerical examples

The FE governing equation:

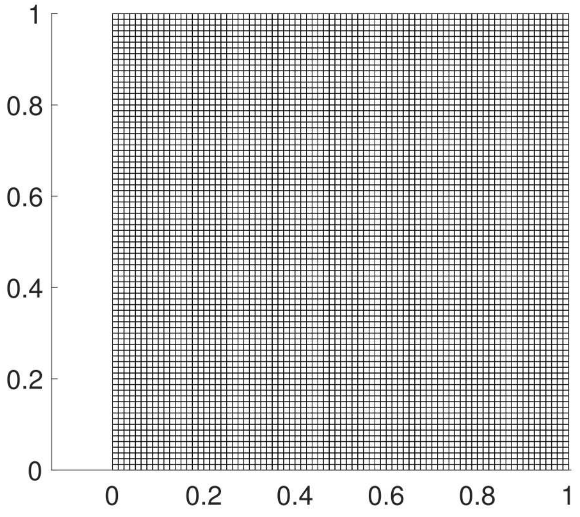
$$-\nabla^2 u + \frac{\mu_1}{\mu_2}\left(e^{\mu_2 u} - 1\right) = 100\sin(2\pi x_1)\sin(2\pi x_2)$$

[Grepl 2007]
[Chaturantabut 2010]

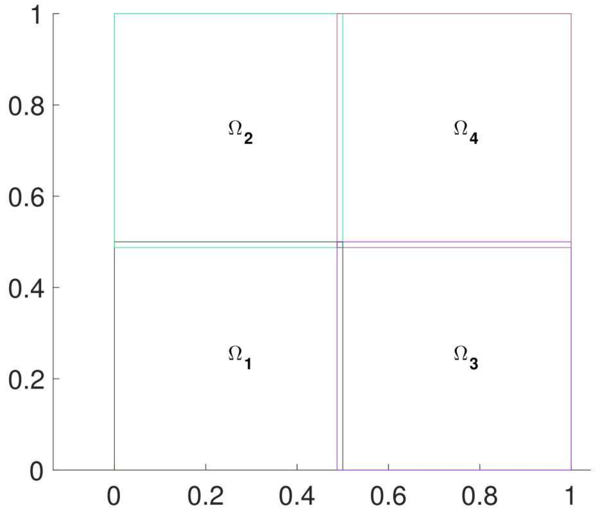
$$\mu = (\mu_1, \mu_2) \in \mathcal{D} = [0.01, 10]^2,\ |\Xi_{\text{train}}| = 400$$



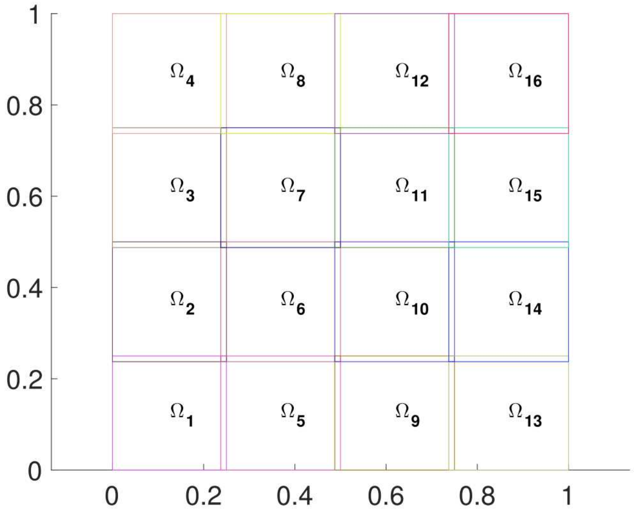
"Coarse" 40x40 elem.



"Fine" 80x80 elem.



2x2 configuration



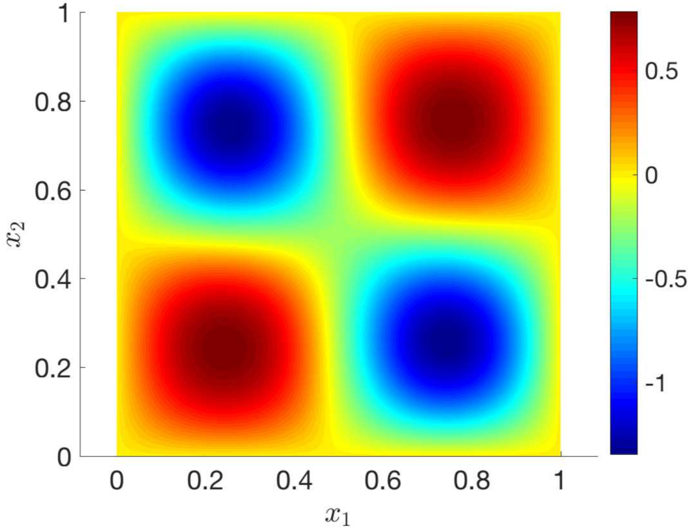
4x4 configuration

Numerical examples: one online computation

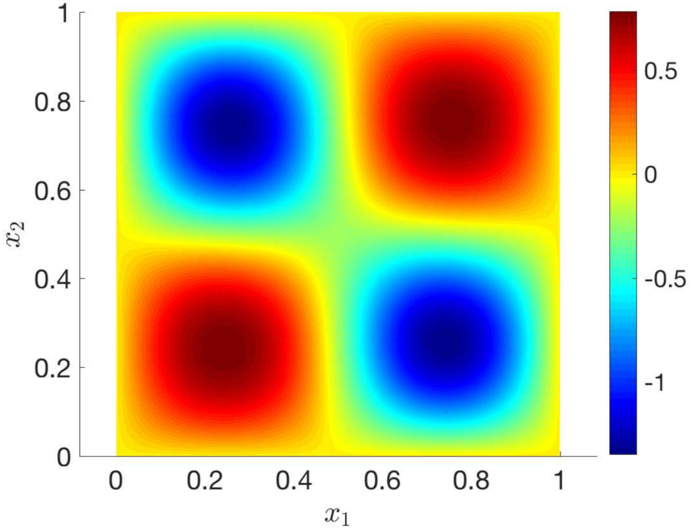
Table 1: Heat equation, 2x2 “fine” configuration. ROM method performance at point $\mu = (5.005, 5.005) \notin \Xi_{\text{train}}$ for one online computation ($1 \leq i \leq n_\Omega$)

constraint type	strong					
basis function type	port		skeleton		interface	
method	LSPG	GNAT	LSPG	GNAT	LSPG	GNAT
energy rate on Ω_i	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-5}$
energy rate on Γ_i	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-5}$	$1-10^{-5}$
energy rate on $\boldsymbol{r}_i$		$1-10^{-12}$		$1-10^{-12}$		$1-10^{-12}$
$n_i^z / n_i^{\boldsymbol{r}}$		2		2		2
relative error	0.0026	0.0012	0.0025	0.0019	0.9912	0.9920
speedup	1.29	2.80	1.29	2.78	1.19	2.60

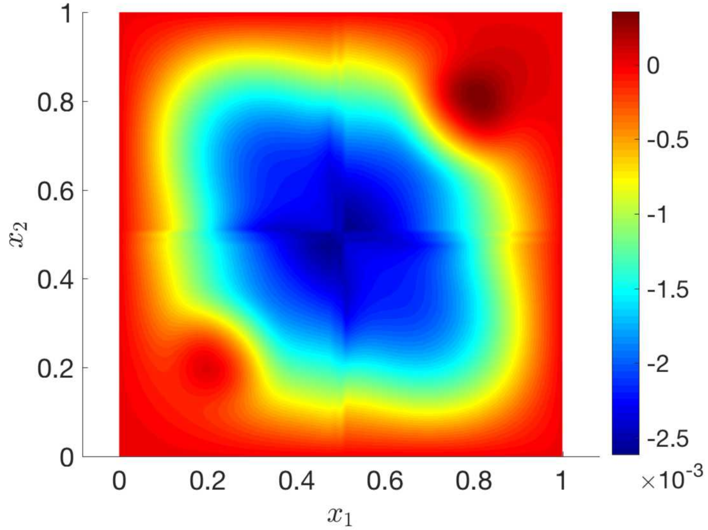
Numerical examples : DDROM and DDGNAT



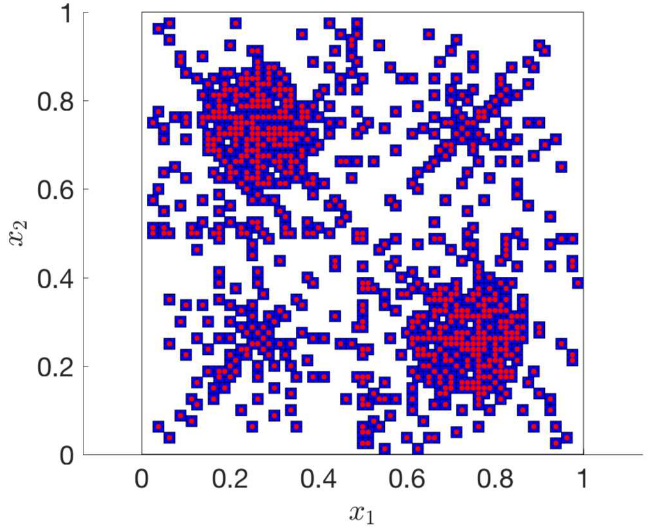
DDFEM solution



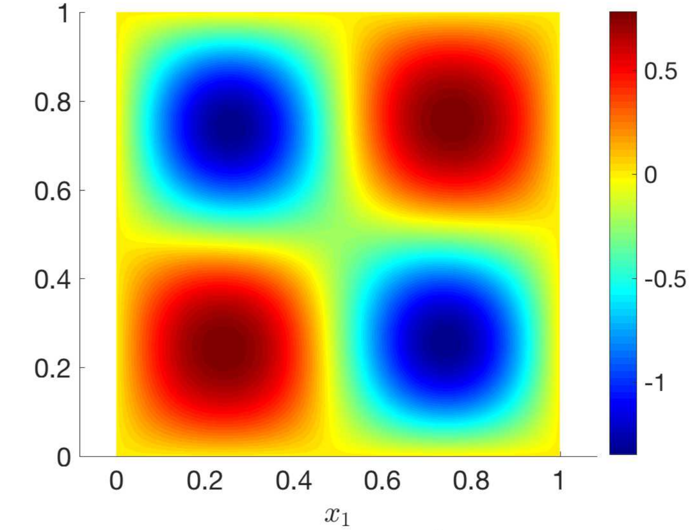
DDROM solution



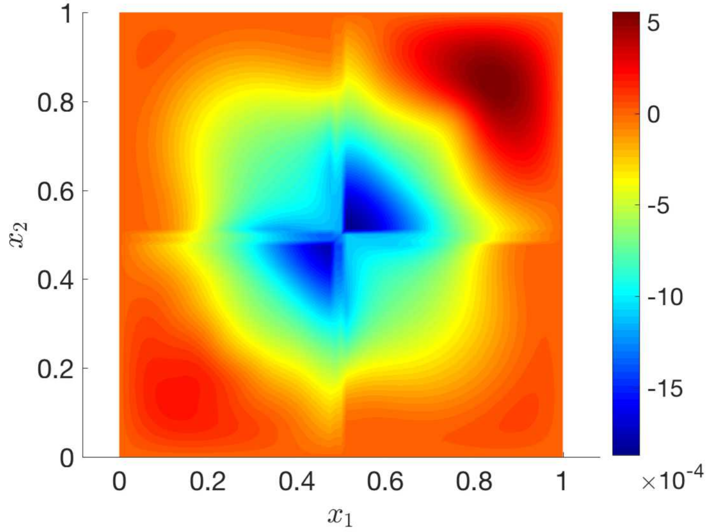
DDROM error



Sample mesh



DDGNAT solution



DDGNAT error

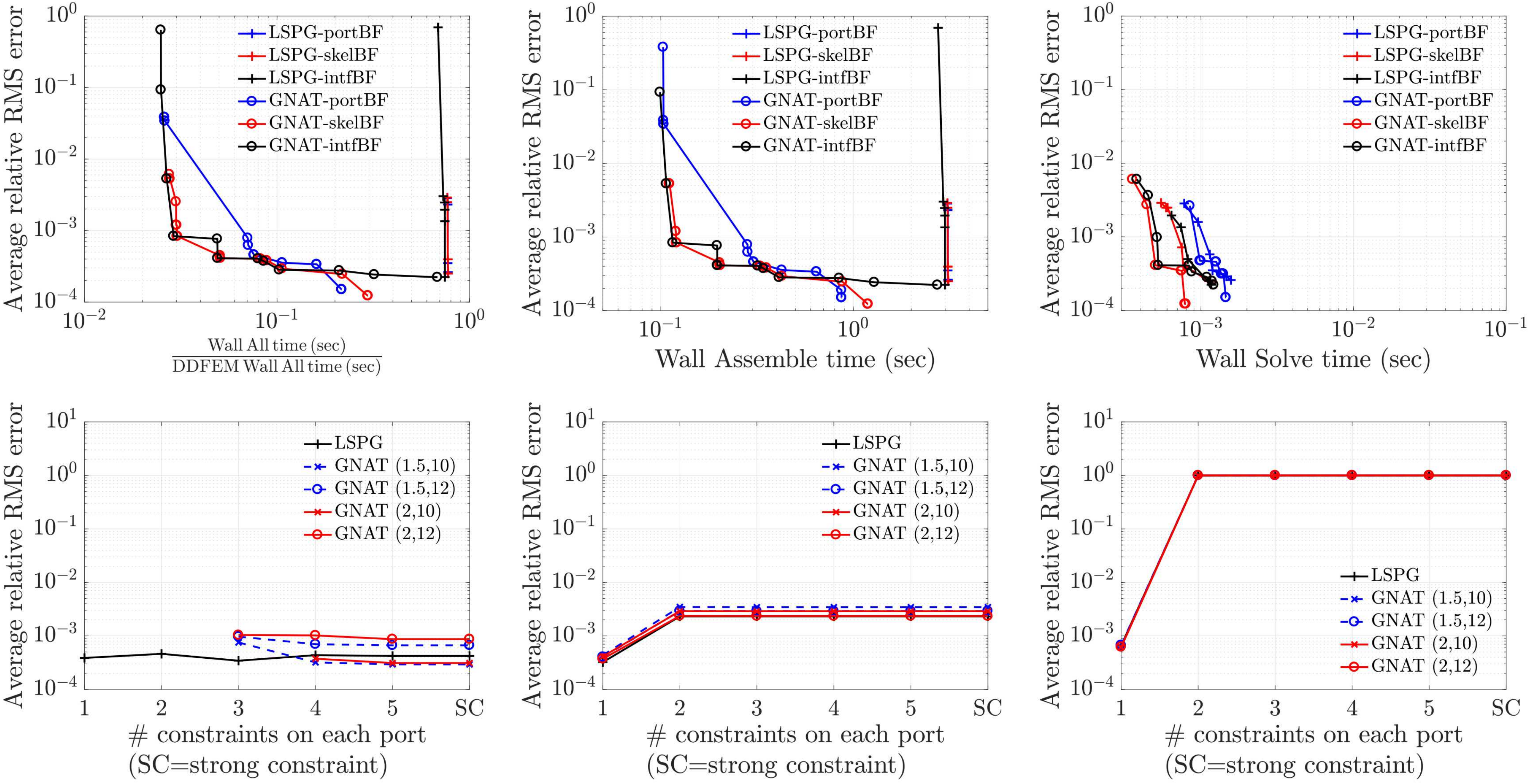
Numerical examples : many online computations

Table 1: Heat equation. ROM method-parameters at point $\mu = (5.005, 5.005) \notin \Xi_{\text{train}}$ for many online computations

method	LSPG	GNAT
energy rate on Ω_i	$\{1-10^{-5}, 1-10^{-8}\}$	$\{1-10^{-5}, 1-10^{-8}\}$
energy rate on Γ_i	$\{1-10^{-5}, 1-10^{-8}\}$	$\{1-10^{-5}, 1-10^{-8}\}$
energy rate on $\boldsymbol{r}_i$		$\{1-10^{-6}, 1-10^{-8}, 1-10^{-10}, 1-10^{-12}\}$
$n_i^z/n_i^{\boldsymbol{r}}$		{1, 1.5, 2, 4}
constraint type	{1, 2, 3, 4, 5, strong}	{1, 2, 3, 4, 5, strong}
BF type	{portBF, skelBF, intfBF}	{portBF, skelBF, intfBF}

Numerical examples : many online computations

2x2 fine config.



portBF, erO=8, erG=8 skelBF, erO=8, erG=8 intfBF, erO=8, erG=8

Summary

- Port type: (+) global problem (don't need weak constraint), but (-) large # dof. Expect strong constraints to be best. Otherwise, we get inconsistent solutions on a port.
- Skeleton type: ideal because (+) global problem (don't need weak constraint), and (+) it minimizes effective # dof, but (-) ``not practical''. Expect strong constraints to be best. Otherwise, we get inconsistent solutions on a port.
- Interface type: (+) practical, but (-) no global problem (need weak constraint). Expect weak constraints to be best. Otherwise, we may have a very low-dimensional global problem.

THANKS FOR YOUR ATTENTION

QUESTION?