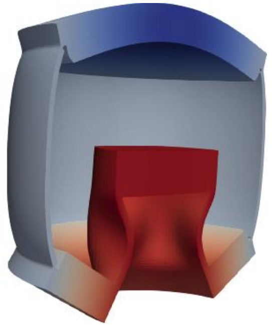
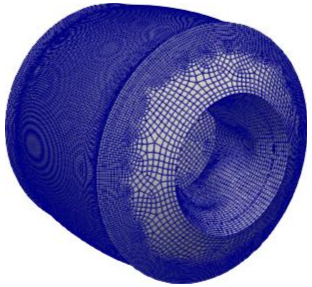
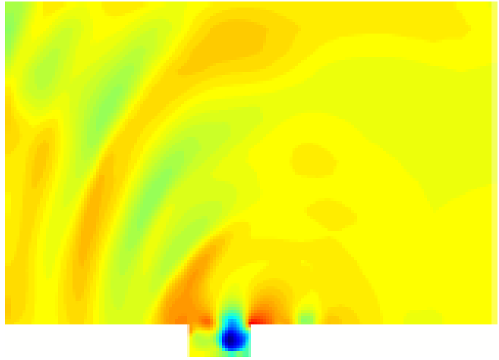
Preconditioned Least-Squares Petrov-Galerkin Reduced Order Models for Fluid and Solid Mechanics Problems







Payton Lindsay[1], Jeff Fike[1], **Irina Tezaur**[1], Kevin Carlberg[2]

[1]Sandia National Labs, [2]Meta Research Labs/U Washington

ECCOMAS 2022 June 6-9, 2022 Oslo, Norway

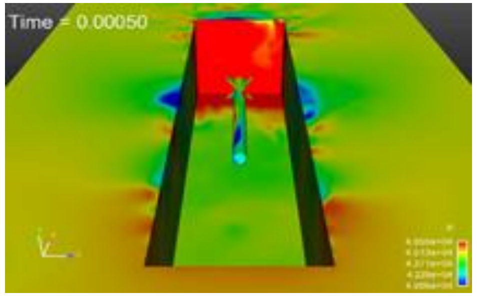
SAND2021-11141C

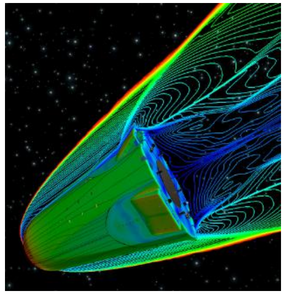
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Motivation

Despite improved algorithms and powerful supercomputers, “**high-fidelity**” **models** are often too expensive for use in a design or analysis setting.

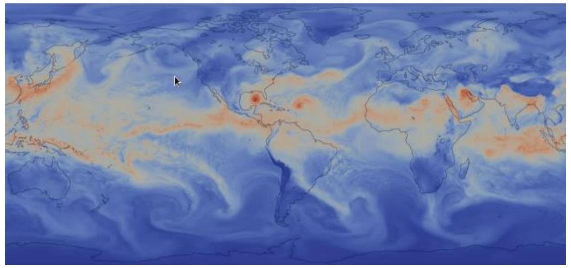
Sandia application areas in which this situation arises:



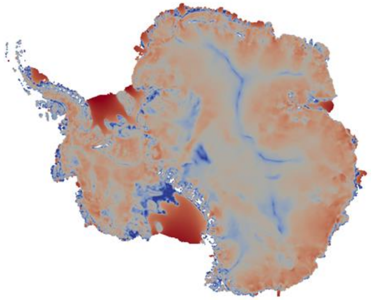


- **Captive-carry** and **re-entry environments:** Large Eddy Simulations (LES) runs require very fine meshes and can take on the order of *weeks*.

- **Fastener failure modeling:** modeling fastener behavior in a full system presents meshing and computational challenges, which limits the number of configurations that can be studied.



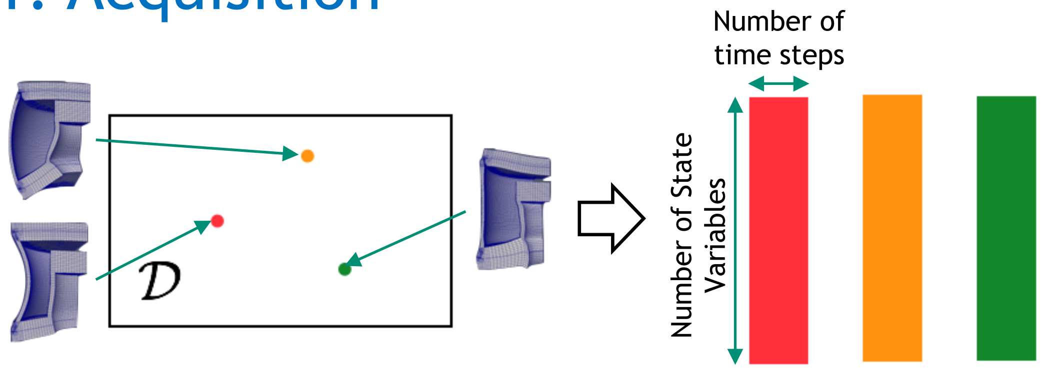
- **Climate modeling** (e.g., land-ice, atmosphere): high-fidelity simulations too costly for uncertainty quantification (UQ); Bayesian inference of high-dimensional parameter fields is intractable.



POD/LSPG* Approach to Model Reduction

Full Order Model (FOM) = Ordinary Differential Equation (ODE): $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$

1. Acquisition

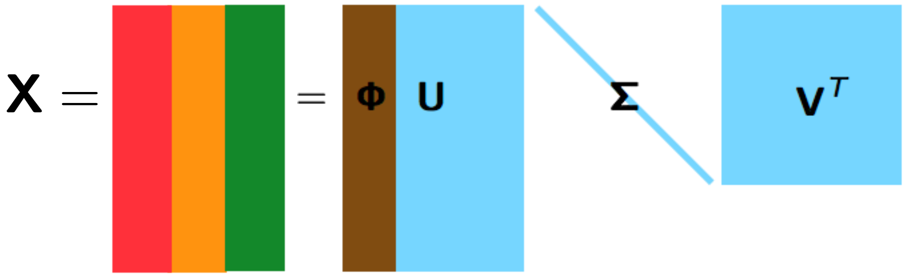


Solve ODE at different design points

Save solution data

2. Learning

Proper Orthogonal Decomposition (POD):



3. Reduction

Choose ODE Temporal Discretization

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

$$\Downarrow$$

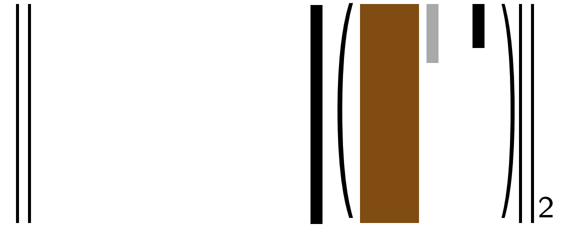
$$\mathbf{r}^n(\mathbf{x}^n; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \dots, T$$

Reduce the number of unknowns

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t)$$

Minimize the Residual

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \| \mathbf{r}^n(\boldsymbol{\Phi}\,\hat{\mathbf{v}}; \boldsymbol{\mu}) \|_2$$

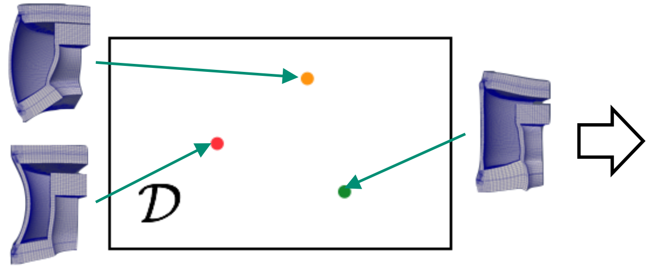


*Least-Squares Petrov-Galerkin Projection [K. Carlberg *et al.*, 2011; K. Carlberg *et al.*, 2017]

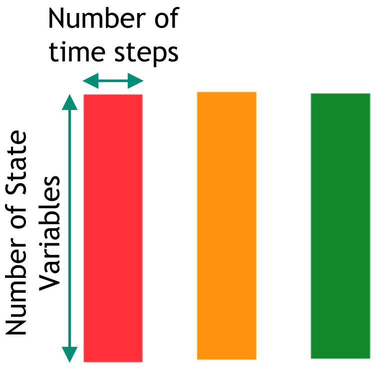
POD/LSPG* Approach to Model Reduction

Full Order Model (FOM) = Ordinary Differential Equation (ODE): $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$

1. Acquisition



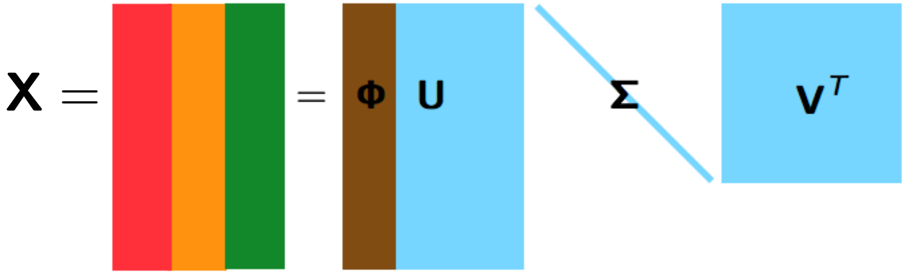
Solve ODE at different design points



Save solution data

2. Learning

Proper Orthogonal Decomposition (POD):



3. Reduction

Choose ODE Temporal Discretization

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

$$\Downarrow$$

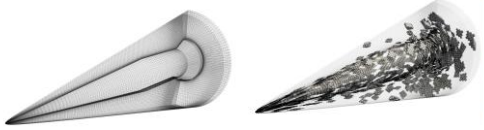
$$\mathbf{r}^n(\mathbf{x}^n; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \dots, T$$

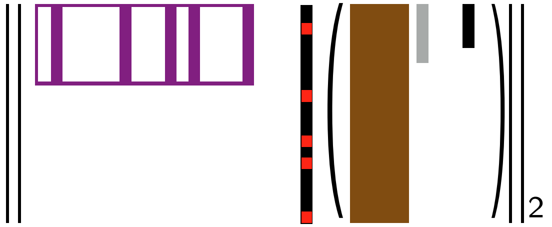
Reduce the number of unknowns

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t)$$

Minimize the Residual

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \mathbf{A}\, \mathbf{r}^n(\boldsymbol{\Phi}\,\hat{\mathbf{v}}; \boldsymbol{\mu}) \right\|_2$$





Hyper-reduction/sample mesh

*Least-Squares Petrov-Galerkin Projection [K. Carlberg *et al.*, 2011; K. Carlberg *et al.*, 2017]

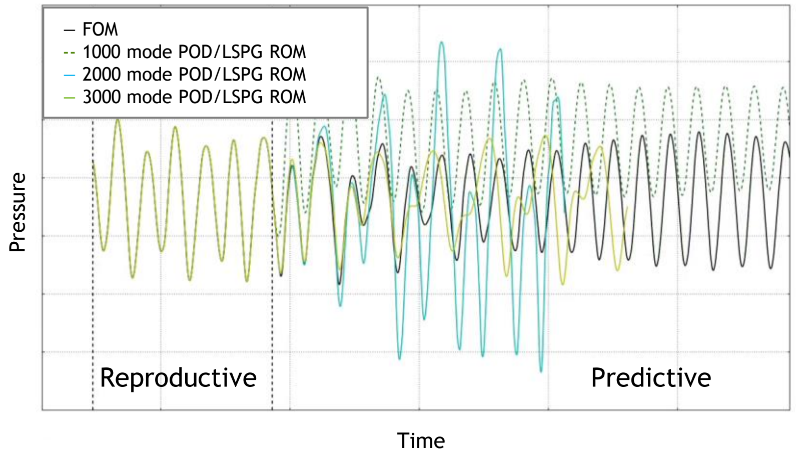
POD/LSPG Approach to Model Reduction

Advantages of POD/LSPG projection:

- Computes a solution that **minimizes the l_2-norm** of the time-discrete residual arising in each Δt
  - ➢ Ensures that adding basis vectors yields a **monotonic decrease** in the least-squares objective function defining the underlying minimization problem [Carlberg *et al.*, 2011]
- Possesses **better stability** and **accuracy** than POD/Galerkin for certain classes of problems (e.g., compressible flow) [Carlberg *et al.*, 2013; Carlberg *et al.*, 2017; Tezaur *et al.*, 2018].

Room for improvement for realistic predictive applications:

- Accuracy for **predictive problems** can be inadequate
- Method may **fail to converge** for some realistic problems run in the predictive regime
- Method may struggle when applied to **problems** with **disparate scales** [Washabaugh, 2016]



Mitigation: introduction of preconditioning into LSPG ROM formulation.

Preconditioned LSPG ROMs

LSPG Formulation:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{y}\in\mathbb{R}^M}{\operatorname{argmin}} ||\boldsymbol{r}(\boldsymbol{\Phi y})||_2^2$$

Optimization problem

$$\delta\hat{\boldsymbol{x}}_{\text{PG}}^{(k)} = \underset{\boldsymbol{y}\in\mathbb{R}^M}{\operatorname{argmin}} ||\boldsymbol{J}^{(k)}\boldsymbol{\Phi y} + \boldsymbol{r}^{(k)}||_2^2$$
$$\hat{\boldsymbol{x}}_{\text{PG}}^{(k)} = \hat{\boldsymbol{x}}_{\text{PG}}^{(k-1)} + \alpha_k \delta\hat{\boldsymbol{x}}_{\text{PG}}^{(k)}$$
$$\tilde{\boldsymbol{x}}_{\text{PG}}^{(k)} = \boldsymbol{\Phi}\hat{\boldsymbol{x}}_{\text{PG}}^{(k)}$$

Gauss-Newton iteration

Normal equations

$$\boldsymbol{J}_{\text{PG}}^{(k)}\delta\hat{\boldsymbol{x}}_{\text{PG}}^{(k)} = -\boldsymbol{r}_{\text{PG}}^{(k)}$$
$$\boldsymbol{J}_{\text{PG}}^{(k)} := \boldsymbol{\Phi}^T\boldsymbol{J}^{(k)T}\boldsymbol{J}^{(k)}\boldsymbol{\Phi}$$
$$\boldsymbol{r}_{\text{PG}}^{(k)} := \boldsymbol{\Phi}^T\boldsymbol{J}^{(k)T}\boldsymbol{r}^{(k)}$$

Preconditioned LSPG Formulation:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{y}\in\mathbb{R}^M}{\operatorname{argmin}} ||\boldsymbol{M}\boldsymbol{r}(\boldsymbol{\Phi y})||_2^2$$

Optimization problem

$$\delta\hat{\boldsymbol{x}}_{\text{PPG}}^{(k)} = \underset{\boldsymbol{y}\in\mathbb{R}^M}{\operatorname{argmin}} ||\boldsymbol{M}^{(k)}(\boldsymbol{J}^{(k)}\boldsymbol{\Phi y} + \boldsymbol{r}^{(k)})||_2^2$$
$$\hat{\boldsymbol{x}}_{\text{PPG}}^{(k)} = \hat{\boldsymbol{x}}_{\text{PPG}}^{(k-1)} + \alpha_k \delta\hat{\boldsymbol{x}}_{\text{PPG}}^{(k)}$$
$$\tilde{\boldsymbol{x}}_{\text{PPG}}^{(k)} = \boldsymbol{\Phi}\hat{\boldsymbol{x}}_{\text{PPG}}^{(k)}$$

Gauss-Newton iteration

Normal equations

$$\boldsymbol{J}_{\text{PPG}}^{(k)}\delta\hat{\boldsymbol{x}}_{\text{PG}}^{(k)} = -\boldsymbol{r}_{\text{PG}}^{(k)}$$
$$\boldsymbol{J}_{\text{PPG}}^{(k)} := \boldsymbol{\Phi}^T\boldsymbol{J}^{(k)T}\boldsymbol{M}^{(k)T}\boldsymbol{M}^{(k)}\boldsymbol{J}^{(k)}\boldsymbol{\Phi}$$
$$\boldsymbol{r}_{\text{PPG}}^{(k)} := \boldsymbol{\Phi}^T\boldsymbol{J}^{(k)T}\boldsymbol{M}^{(k)T}\boldsymbol{M}^{(k)}\boldsymbol{r}^{(k)}$$

Preconditioned LSPG ROMs

Adding preconditioning to the POD/LSPG formulation can **improve** not only **ROM efficiency** but also **ROM accuracy**.

Ideal preconditioned ROM emulates projection of FOM solution increment onto POD basis.

- Upper limit on ROM accuracy is obtained by taking **solution increment** computed by FOM, $\delta \boldsymbol{x}^{(k)}$, at each time step k and **projecting it** onto the **POD basis**:

$$\boxed{\delta \widetilde{\boldsymbol{x}}^{(k)} = \boldsymbol{\Phi}(\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \delta \boldsymbol{x}^{(k)}} \qquad (1)$$

- **Ideal preconditioned** ROM ($\boldsymbol{M}^{(k)} = (\boldsymbol{J}^{(k)})^{-1}$) gives rise to "projected solution increment" (1).
- As quality of preconditioner is improved ($\boldsymbol{M}^{(k)} \to (\boldsymbol{J}^{(k)})^{-1}$), ROM solution **approaches** most accurate ROM solution possible for a given basis $\boldsymbol{\Phi}$.

Preconditioning can get different residual components on approximately the same scale.

- Minimizing the raw (unweighted) residual $\boldsymbol{r}$ can be problematic for systems of PDEs where different variables have **drastically different magnitudes** (e.g., dimensional PDEs, multi-physics) [Washabaugh, 2016].
- Adding a preconditioner can **scale** the ROM residual to get all the equations to be roughly of the same order.

Preconditioning LSPG formulation changes the norm defining the residual minimization.

- Norm change can **improve residual-based stability constant** bounding ROM solution's **error** (κ_0, $\kappa_1 \to 1$) .

$$\frac{1}{\kappa_0} ||\boldsymbol{r}(\widetilde{\boldsymbol{w}})||_2 \leq ||\boldsymbol{w} - \widetilde{\boldsymbol{w}}||_2 \leq \frac{1}{\kappa_1} ||\boldsymbol{r}(\widetilde{\boldsymbol{w}})||_2$$

Numerical Examples: Albany and SPARC codes

Albany multi-physics finite element code

- Open-source[1], parallel, C++ code
- Component-based design for rapid development
- Contains a wide variety of constitutive models for mechanical/thermo-mechanical problems.
- Makes extensive use of libraries from the open-source Trilinos project[2], including preconditioners from the Ifpack library

Problems tested: quasi-static mechanical and **thermo-mechanical** with prediction across **material parameter space**[3].

[1]https://github.com/SNLComputation/Albany/releases/tag/MOR_support_end
[2]https://github.com/trilinos/trilinos
[3]P. Lindsay *et al.*, *IJNME*, 2022 (accepted), https://arxiv.org/abs/2203.12180

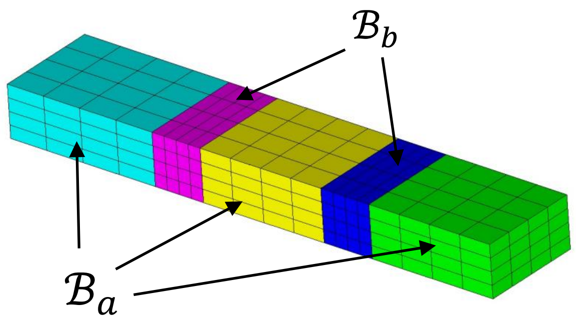
SPARC[4] Flow Solver

- Next-generation transonic and hypersonic C++ CFD code developed at Sandia
- Simulates compressible flow
- Used for analyses involving captive carry and reentry vehicles
- Primary discretization is cell-centered finite volume method
- Leverages libraries from the Trilinos project[2]

Problems tested: transient **compressible laminar flow** over an open cavity with **prediction in time**

[4]Sandia Parallel Aerodynamics and Reentry Code

Mechanical Beam (Albany)



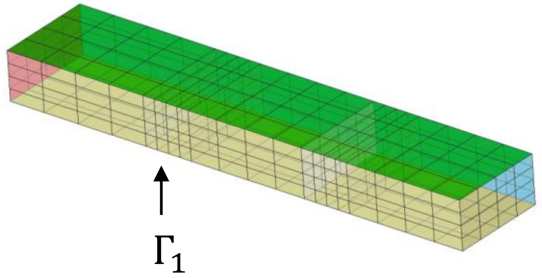
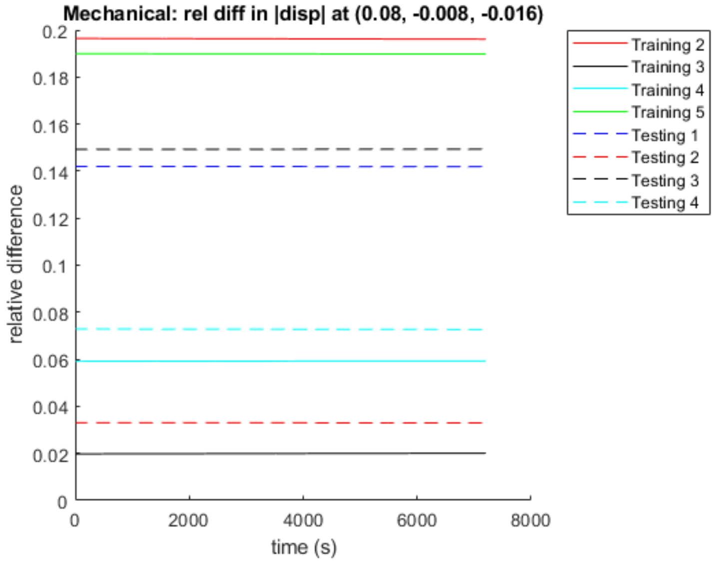


Table 2. Parameters in block $\mathcal{B}_b$ for mechanical beam problem.

Regime	Case	$E_b(\times 10^{11})$ [Pa]	ν_b	ρ_b [kg/m^3]
training	1	1.38002	0.28028	9194.74
	2	2.11826	0.332646	7683.22
	3	1.82559	0.395908	6150.4
	4	1.56036	0.350415	9067.35
	5	1.68463	0.256473	7466.27
testing	1	1.50293	0.244704	6466.96
	2	1.54545	0.304329	6774.12
	3	1.47145	0.367092	8362.44
	4	1.703	0.32	7920

- **Mechanical** problem involving **Neohookean** material
- 2 sets of **material blocks**, $\mathcal{B}_a$ and $\mathcal{B}_b$, each having set of material params
  - Material parameters in block $\mathcal{B}_a$ are fixed
  - Material parameters in block $\mathcal{B}_b$ are varied (see Table 2)
- Linearly varying **time-dependent pressure** BC is prescribed on Γ_1; other boundaries are fixed
- Problem is run **quasi-statically** to pseudo-time t =7200s with 1340 dofs
- **Training** is performed for 6 sets of parameters; **testing/prediction** is performed for 4 sets of parameters (see Table 2)
  - **Nontrivial variations** in displacement (up to 20%) are observed with the parameter variations considered (right figure)



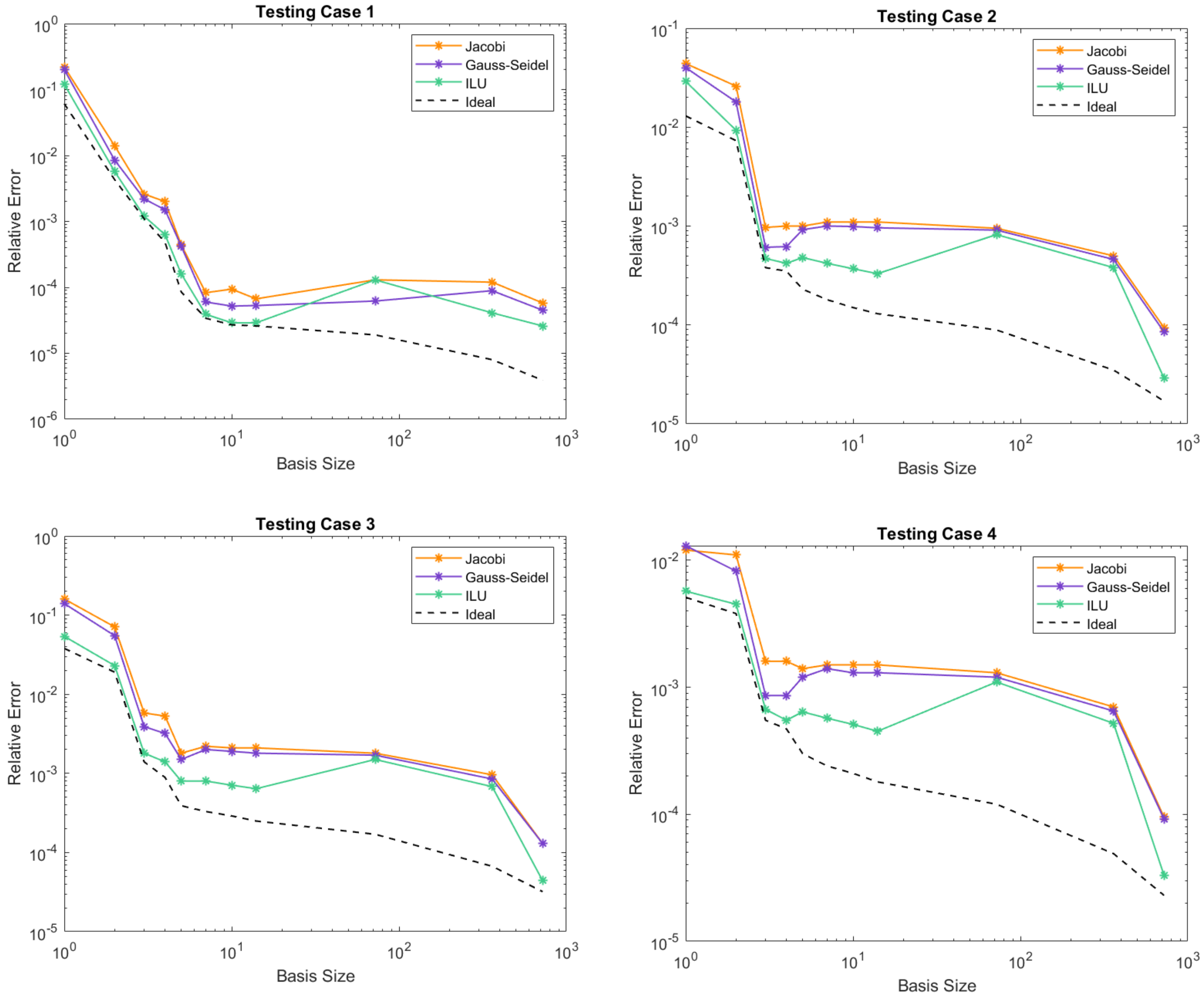
[Lindsay *et al.*, in prep.]

Mechanical Beam (Albany)

- Figure plots **global relative error** in approximate ROM solutions:

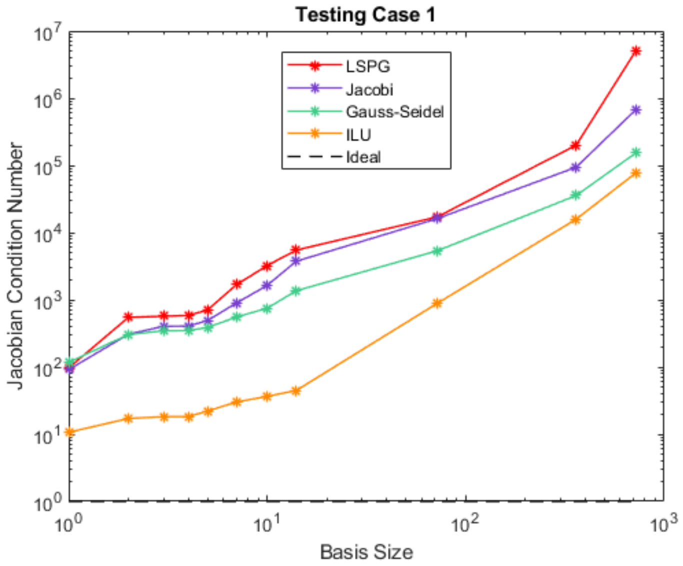
$$\epsilon := \frac{\sum_{i=0}^{P} \|\boldsymbol{x}_i - \widetilde{\boldsymbol{x}}_i\|_2}{\sum_{i=0}^{P} \|\boldsymbol{x}_i\|_2}$$

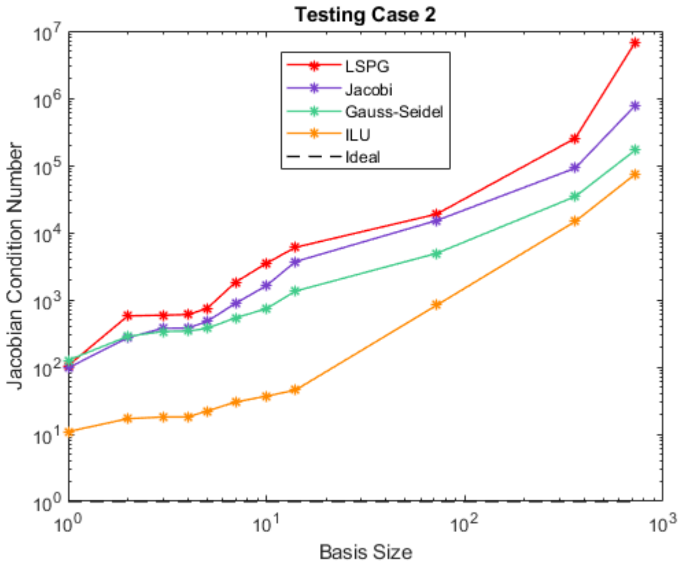
- **Preconditioners evaluated**: Jacobi, Gauss-Seidel, ILU and $(\boldsymbol{J}^{(k)})^{-1}$ (denoted by "Ideal")
- Nonlinear solver for unpreconditioned LSPG ROM **did not converge** for any of the basis sizes considered.
- More sophisticated preconditioners deliver **smaller errors.**

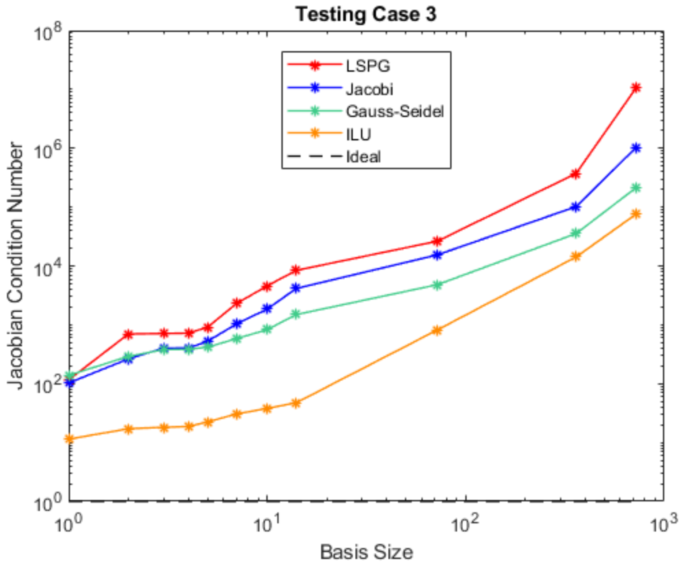


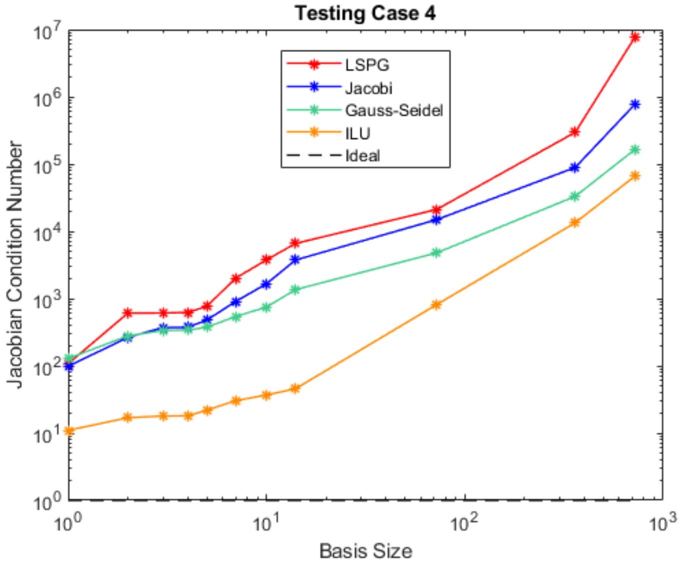
Mechanical Beam (Albany)

- Figure plots **condition numbers** of reduced Jacobian ($\boldsymbol{J}_{\mathrm{PPG}}^{(k)}$ or $\boldsymbol{J}_{\mathrm{PG}}^{(k)}$) for each ROM.
- A **moderate reduction in condition number** is obtained through preconditioning strategies.
- Most sophisticated ILU preconditioner gives rise to a reduced Jacobian with the **smallest condition number.**



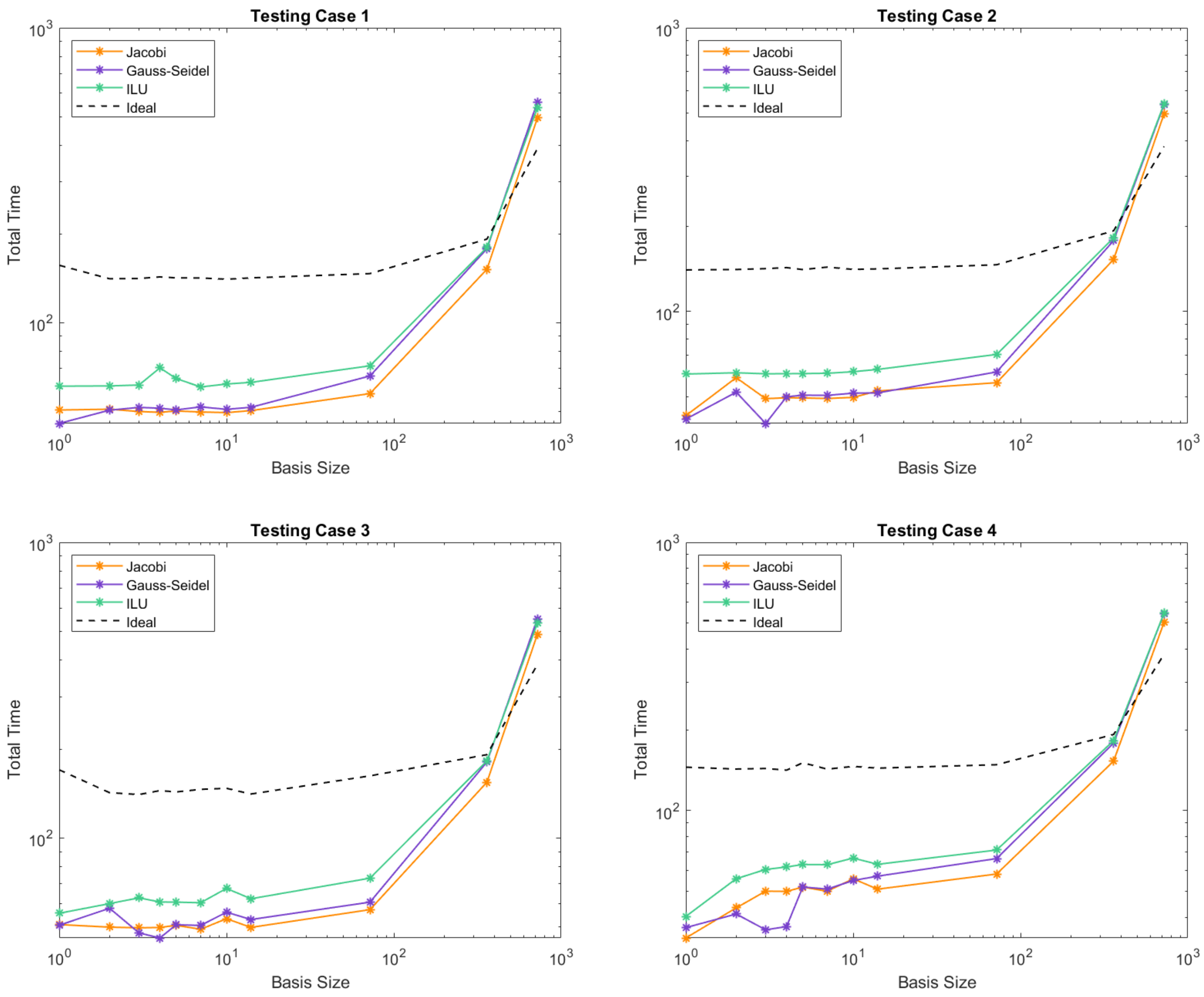






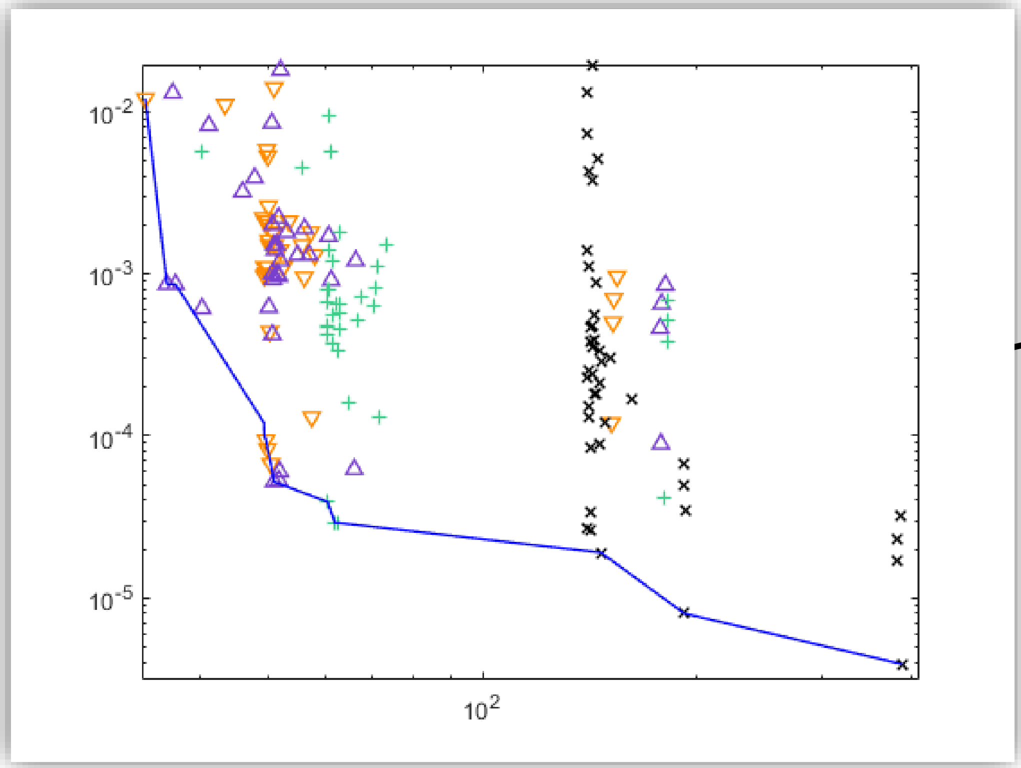
Mechanical Beam (Albany)

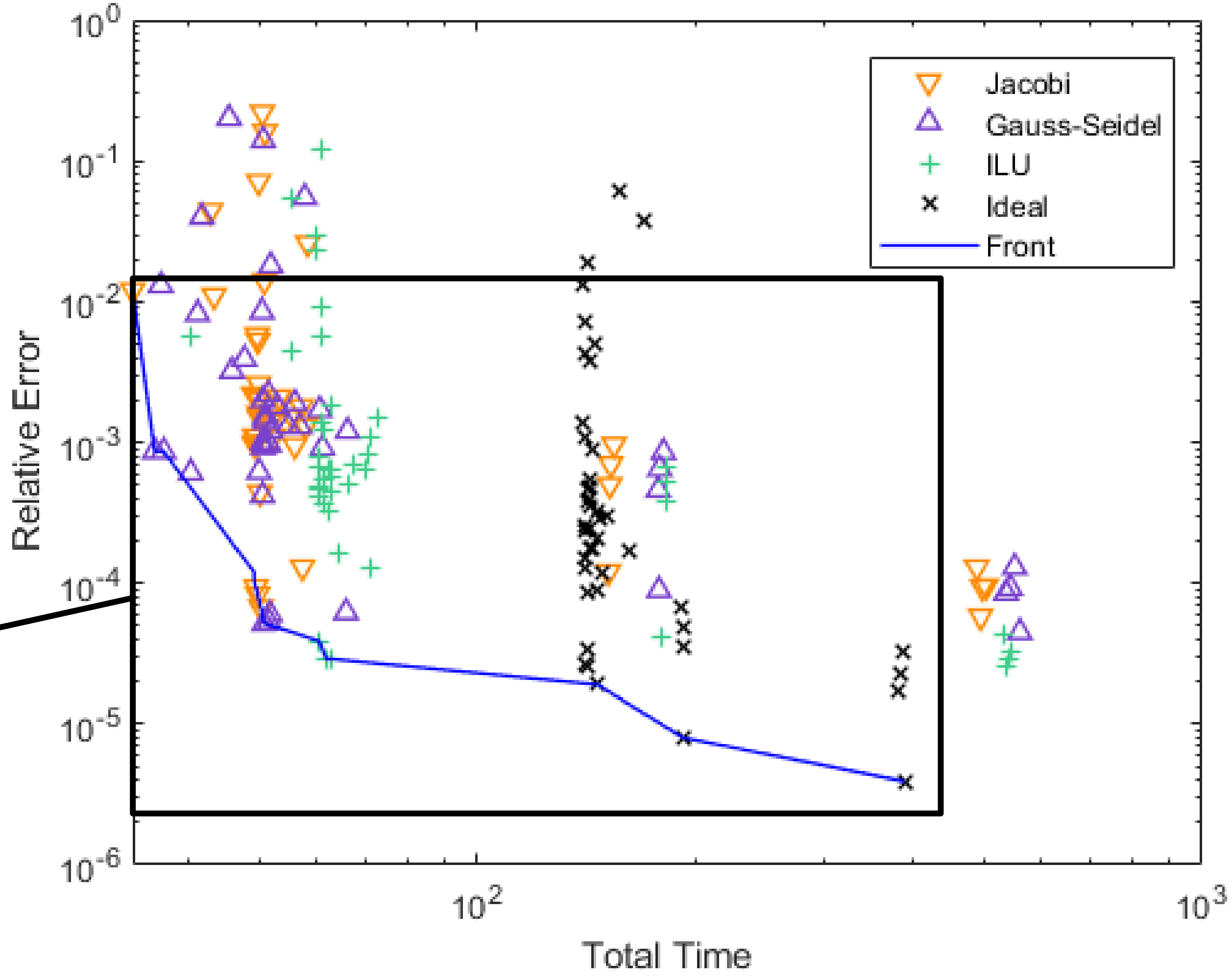
- Figures shows **CPU-times** for all ROMs considered
- As expected, the **projected solution increment** is the **most expensive** to compute



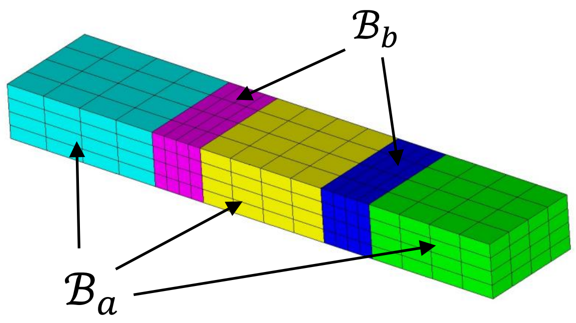
Mechanical Beam (Albany)

The **best preconditioner** given error/CPU-time requirements can be inferred from **Pareto plot** shown here.





Thermo-Mechanical Beam (Albany)



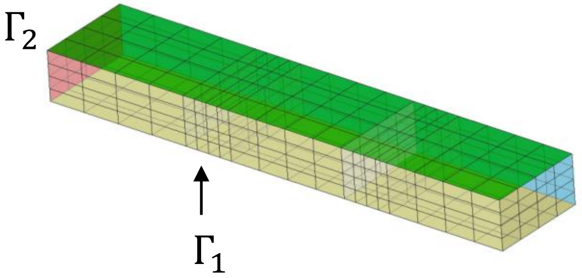
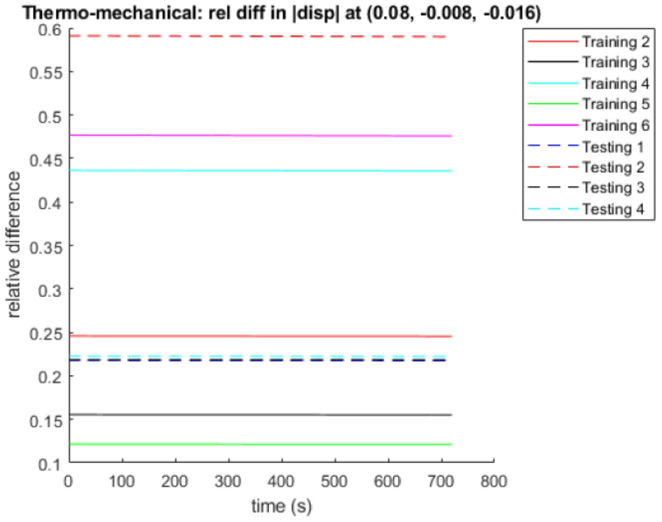


Table 3. Parameters in block $\mathcal{B}_b$ for thermo-mechanical beam problem.

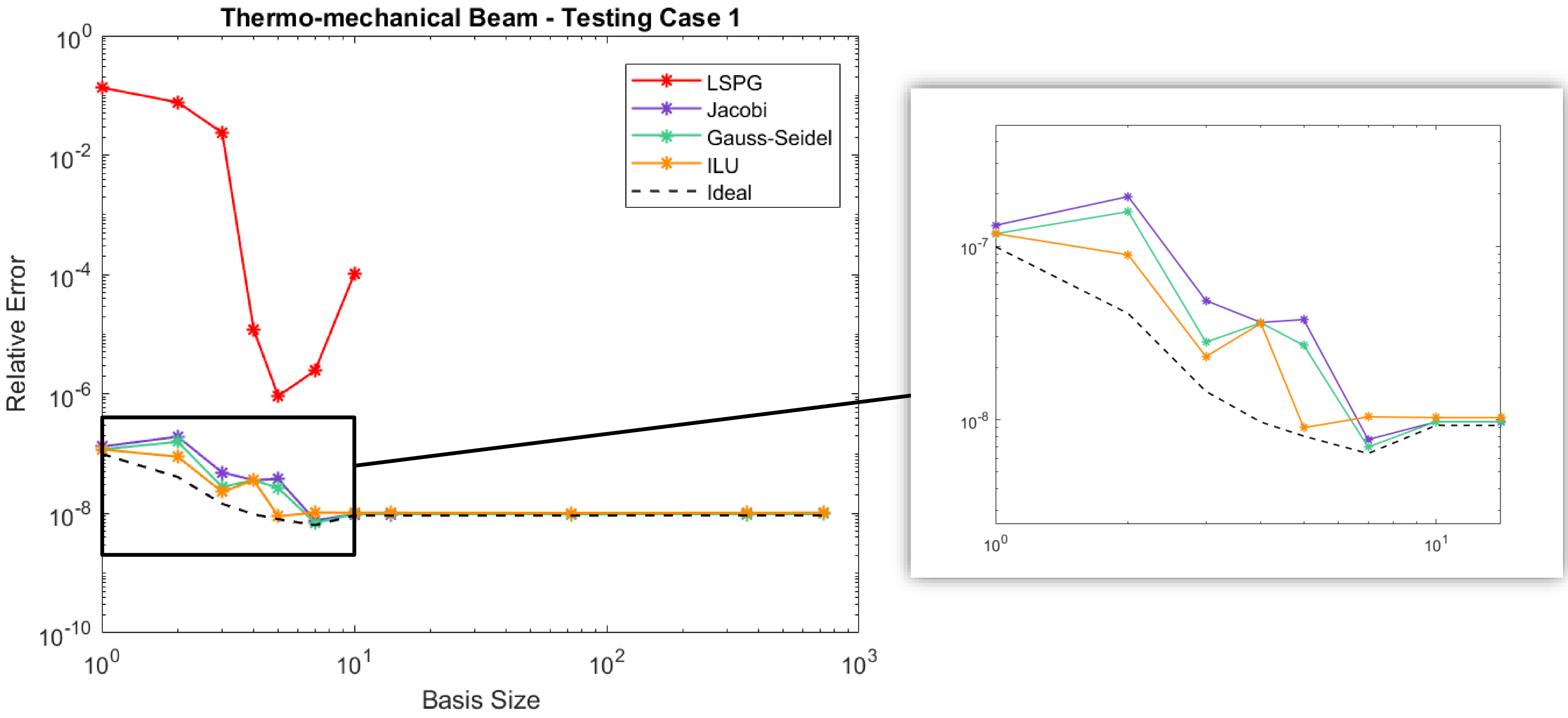
Regime	Case	$E_b(\times 10^9)$ [Pa]	ν_b	$\rho_b(\times 10^{-5})$ [kg/m^3]	$T_{b,\mathrm{ref}}$ [K]
training	1	2.01313	0.285907	7.94827	273.657
	2	1.71637	0.332083	6.93965	318.406
	3	1.96881	0.3478	9.37181	301.406
	4	1.28954	0.29427	9.14636	365.378
	5	1.61326	0.262464	6.32164	223.434
	6	1.54724	0.374118	7.31561	245.778
testing	1	1.52473	0.27925	8.80694	266.674
	2	1.31153	0.345538	7.58234	333.462
	3	1.37015	0.246513	7.73303	345.942
	4	1.703	0.32	7.92	293

- Coupled **thermo-mechanical** problem involving **Neohookean** material
- 2 sets of **material blocks**, $\mathcal{B}_a$ and $\mathcal{B}_b$, each having set of material params
  - ➢ Material parameters in block $\mathcal{B}_a$ are fixed
  - ➢ Material parameters in block $\mathcal{B}_b$ are varied (see Table 3)
- Linearly varying **time-dependent pressure** and **temperature** BC is prescribed on Γ_1 and Γ_2, respectively; other boundaries are fixed
- Problem is run **quasi-statically** to pseudo-time t =7200s with 2100 dofs
- **Training** is performed for 6 sets of parameters; **testing/prediction** is performed for 4 sets of parameters (see Table 3)
  - ➢ **Significant variations** in displacement (up to 60%) are observed with the parameter variations considered (right figure)

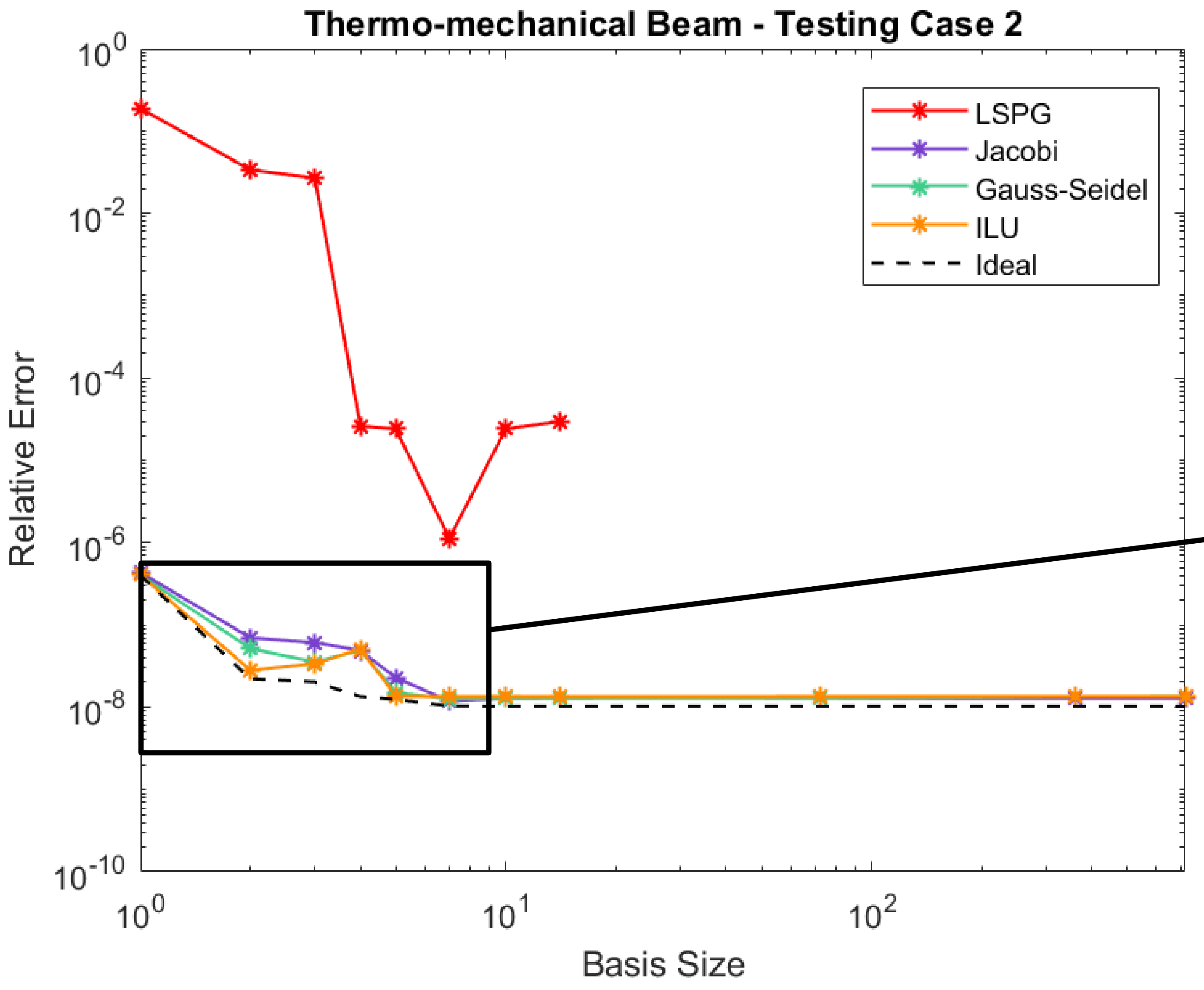


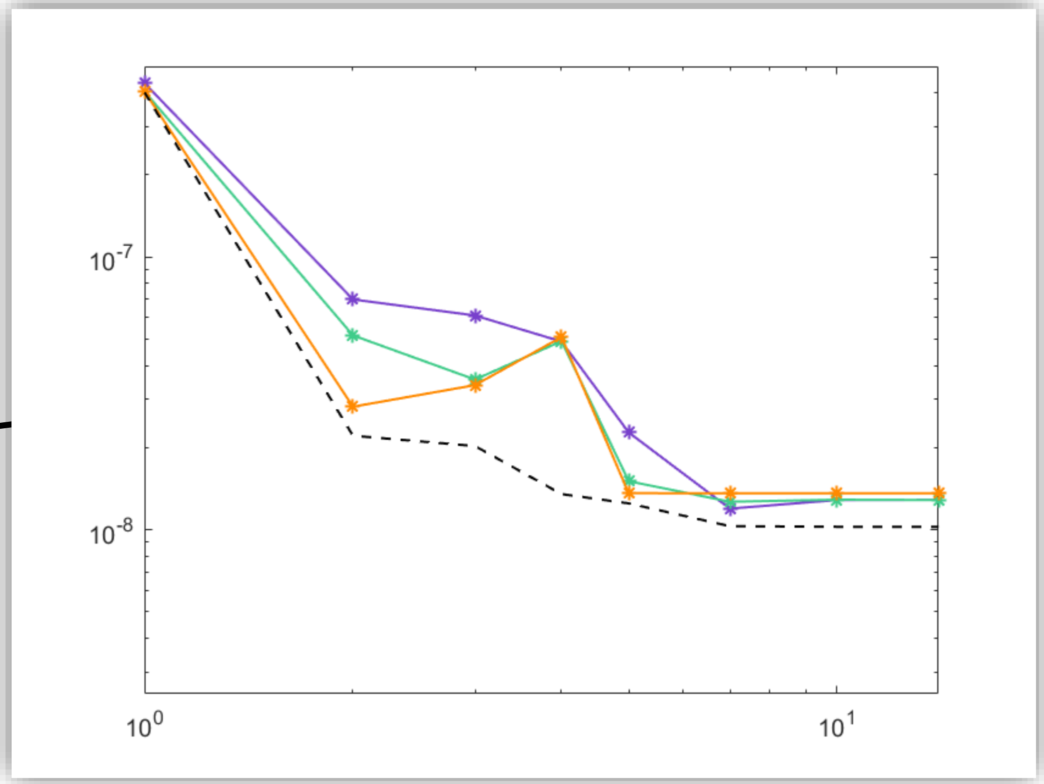
[Lindsay *et al.*, in prep.]

Thermo-Mechanical Beam (Albany)

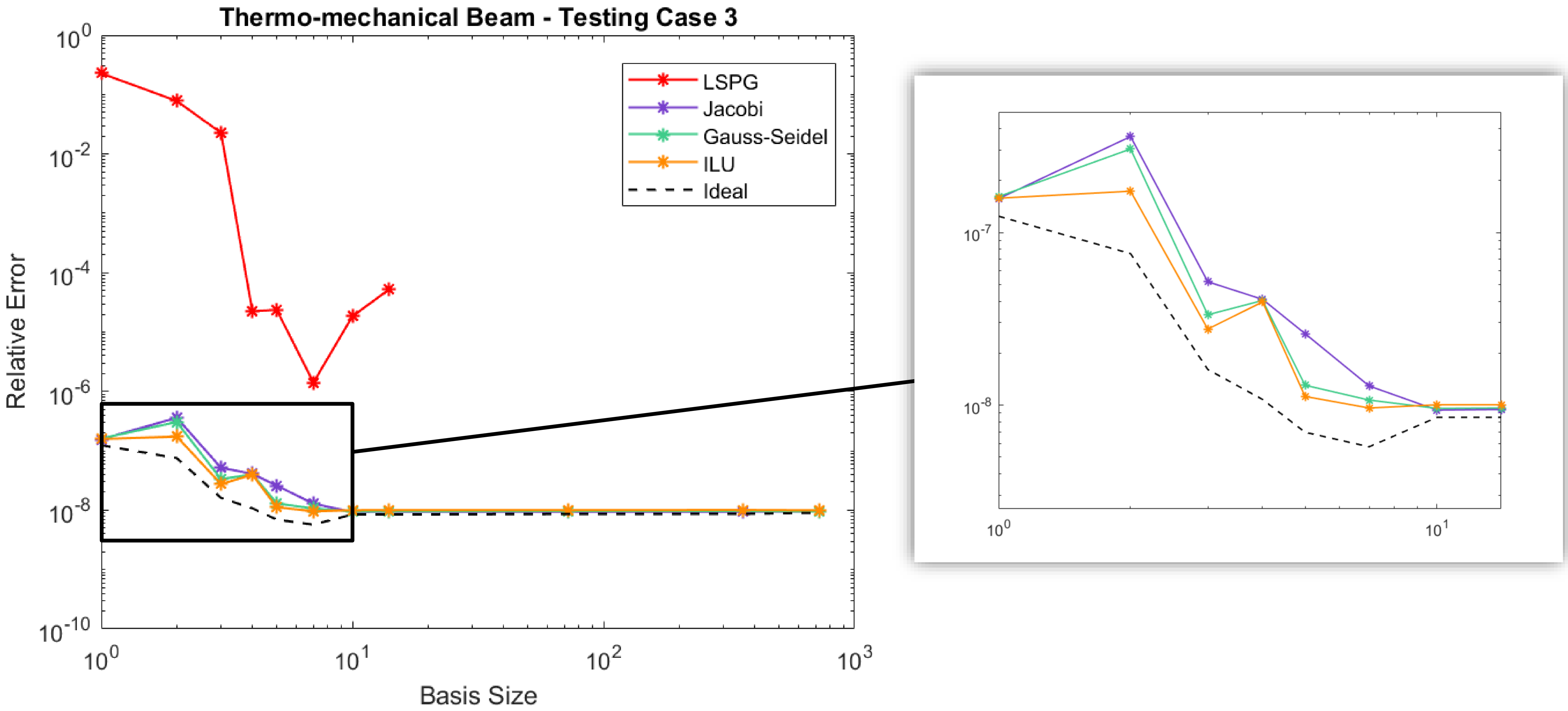


Thermo-Mechanical Beam (Albany)

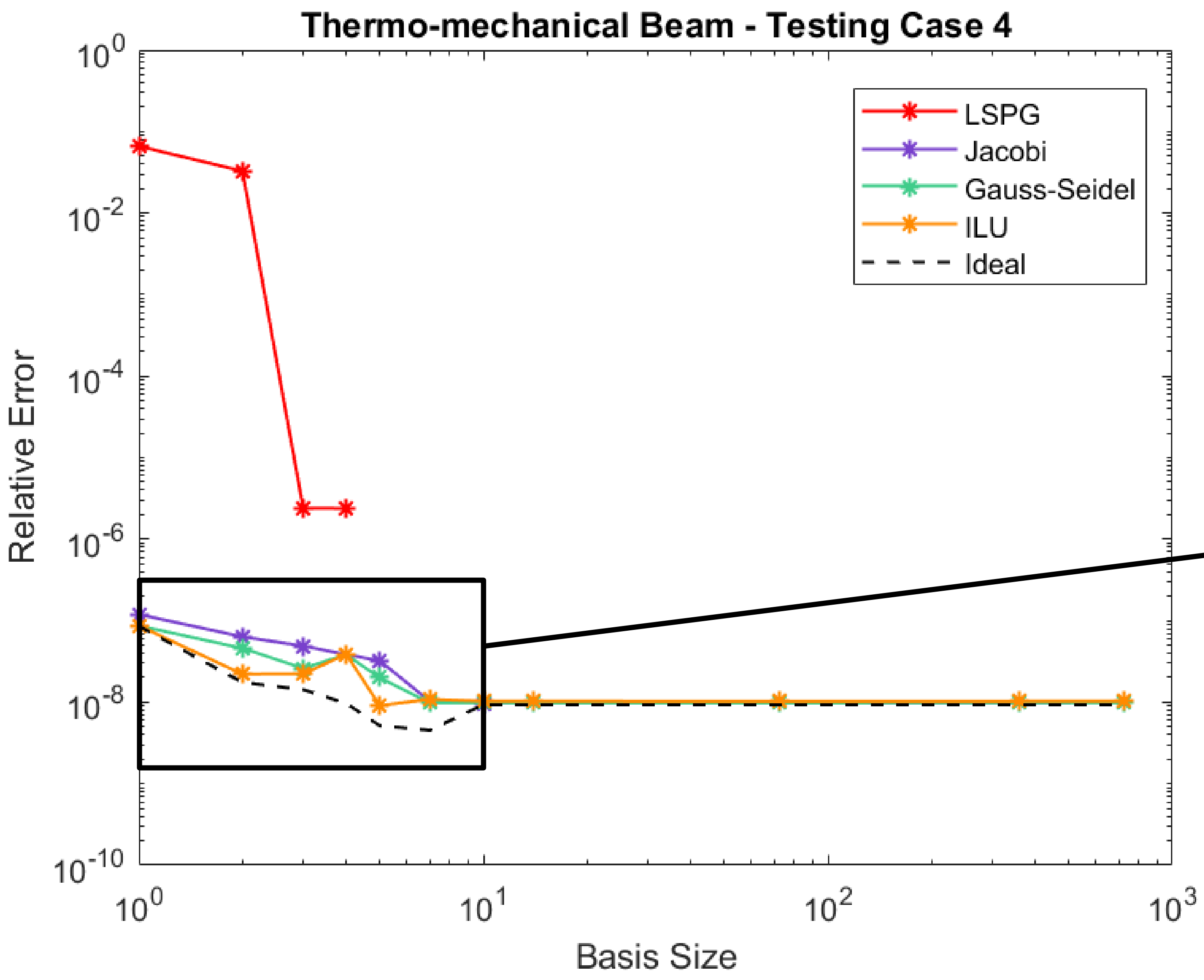


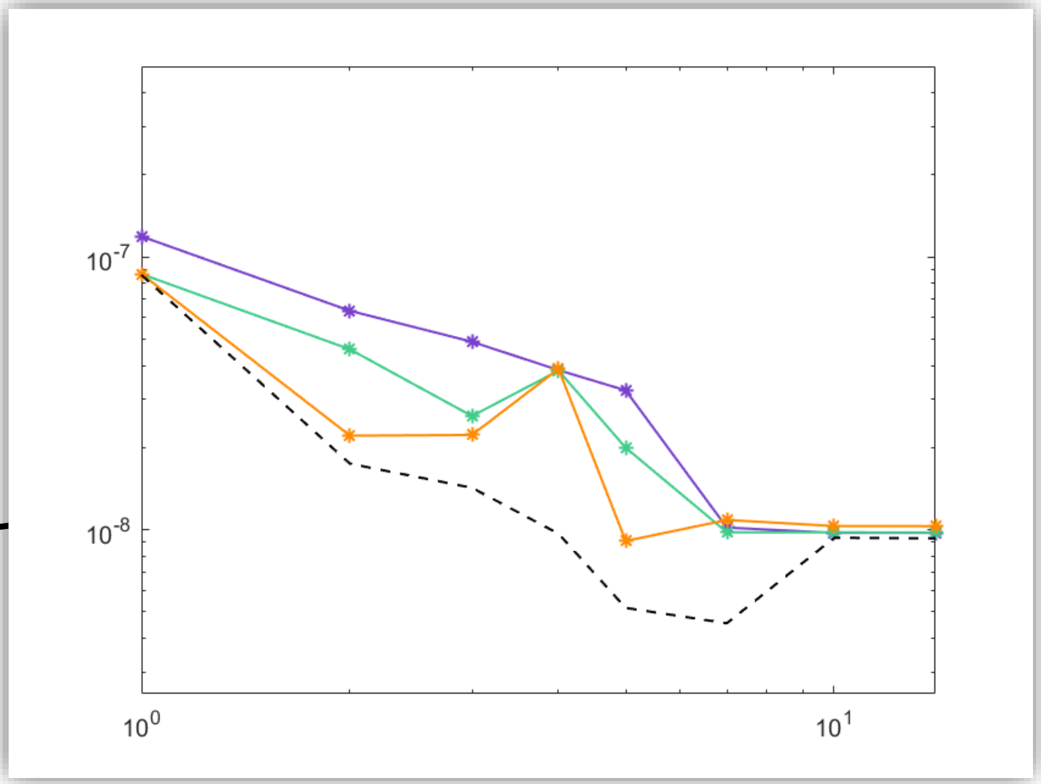


Thermo-Mechanical Beam (Albany)



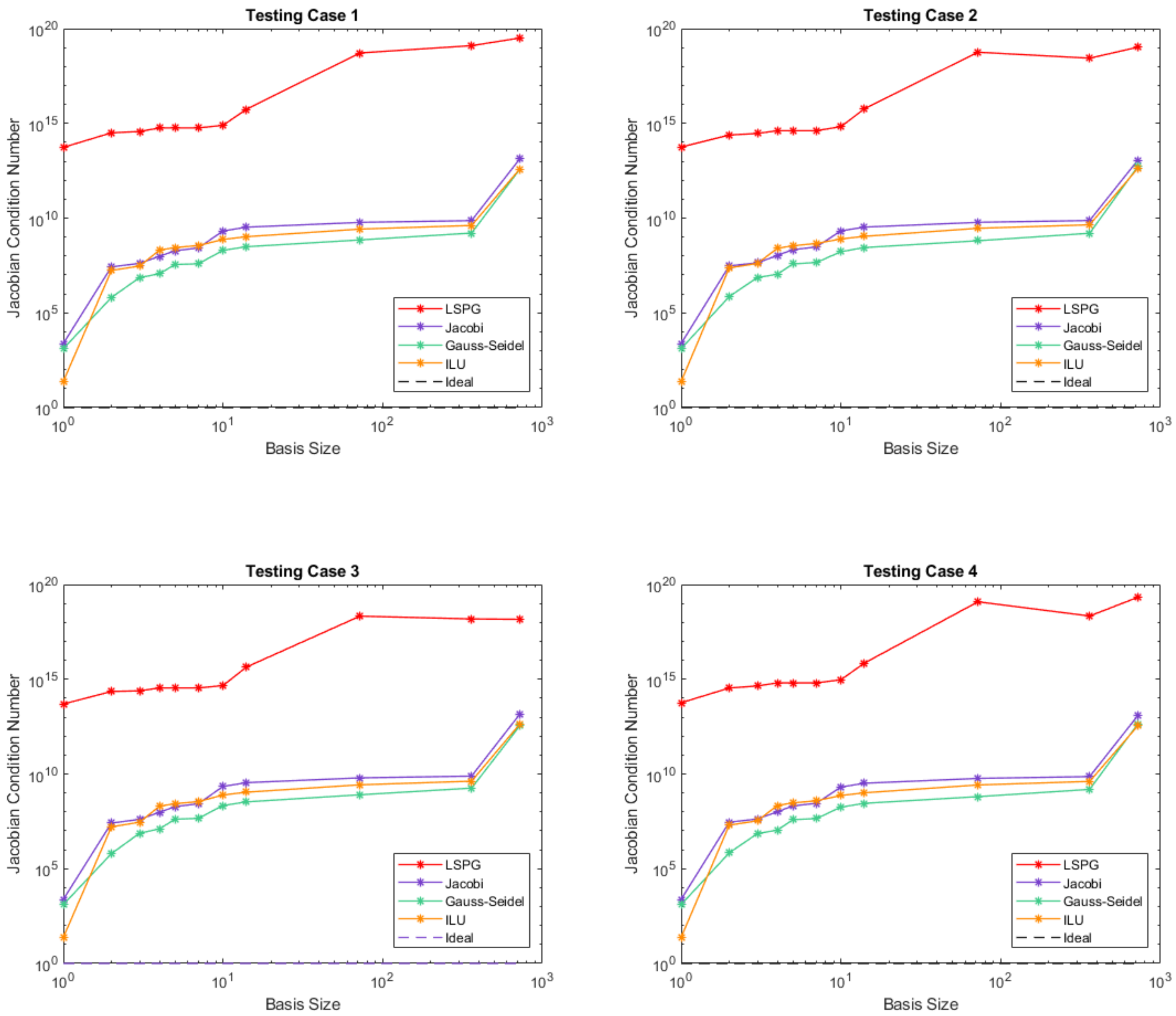
Thermo-Mechanical Beam (Albany)





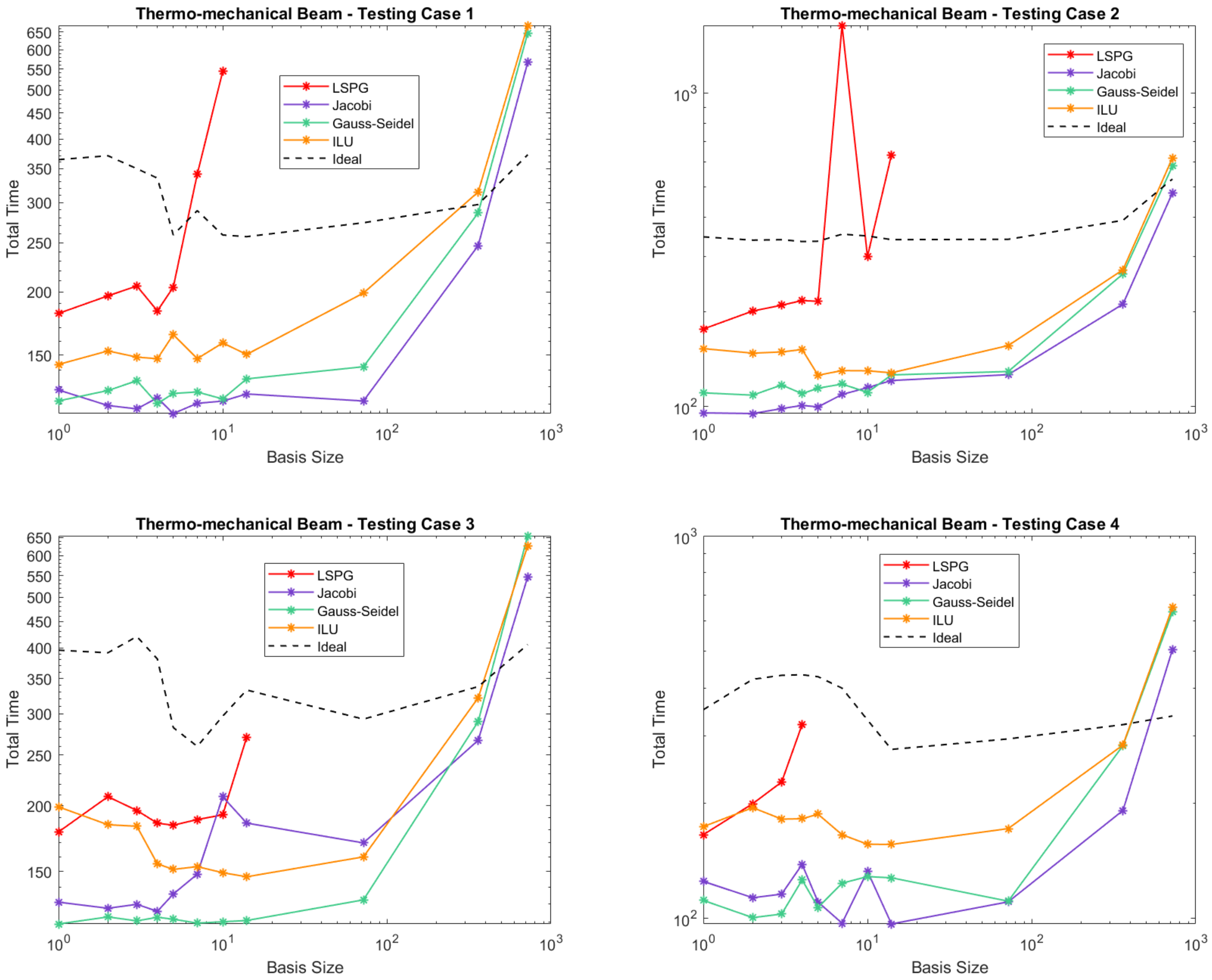
Thermo-Mechanical Beam (Albany)

- Figure plots **condition numbers** of reduced Jacobian ($\boldsymbol{J}_{\mathrm{PPG}}^{(k)}$ or $\boldsymbol{J}_{\mathrm{PG}}^{(k)}$) for each ROM.
- Reduced Jacobians for regular LSPG ROM are **very ill-conditioned** ($> O(10^{14})$)
  - ➢ Ill-conditioning is due to extreme differences in scale b/w displacement and temperature solutions (9 orders of magnitude)
- Results demonstrate that **simple preconditioning** strategy can reduce condition numbers by as many as **10 orders of magnitude**
- As expected, projected solution increment reduced Jacobian has **perfect condition number**



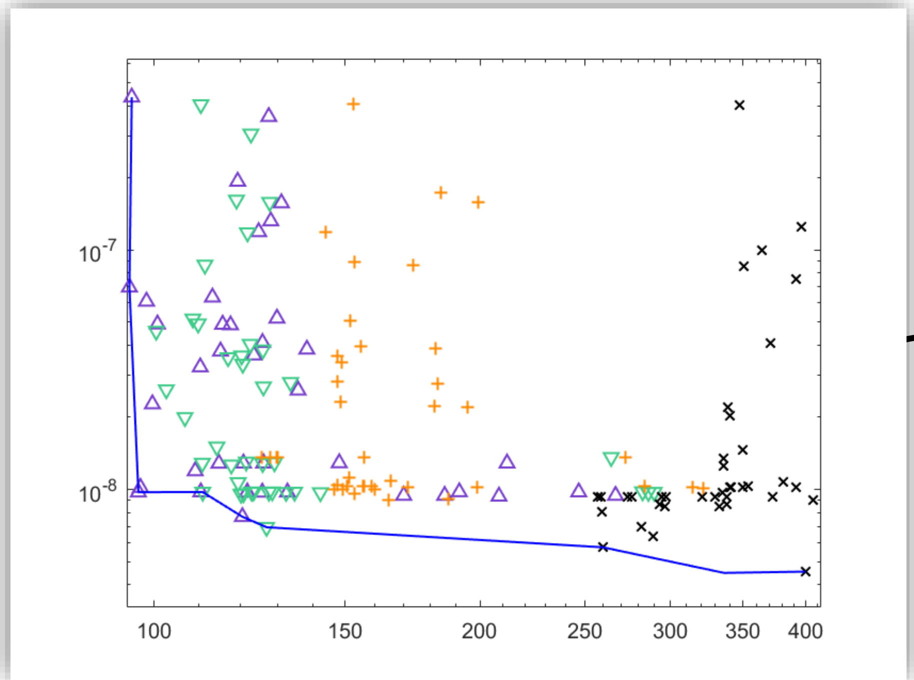
Thermo-Mechanical Beam (Albany)

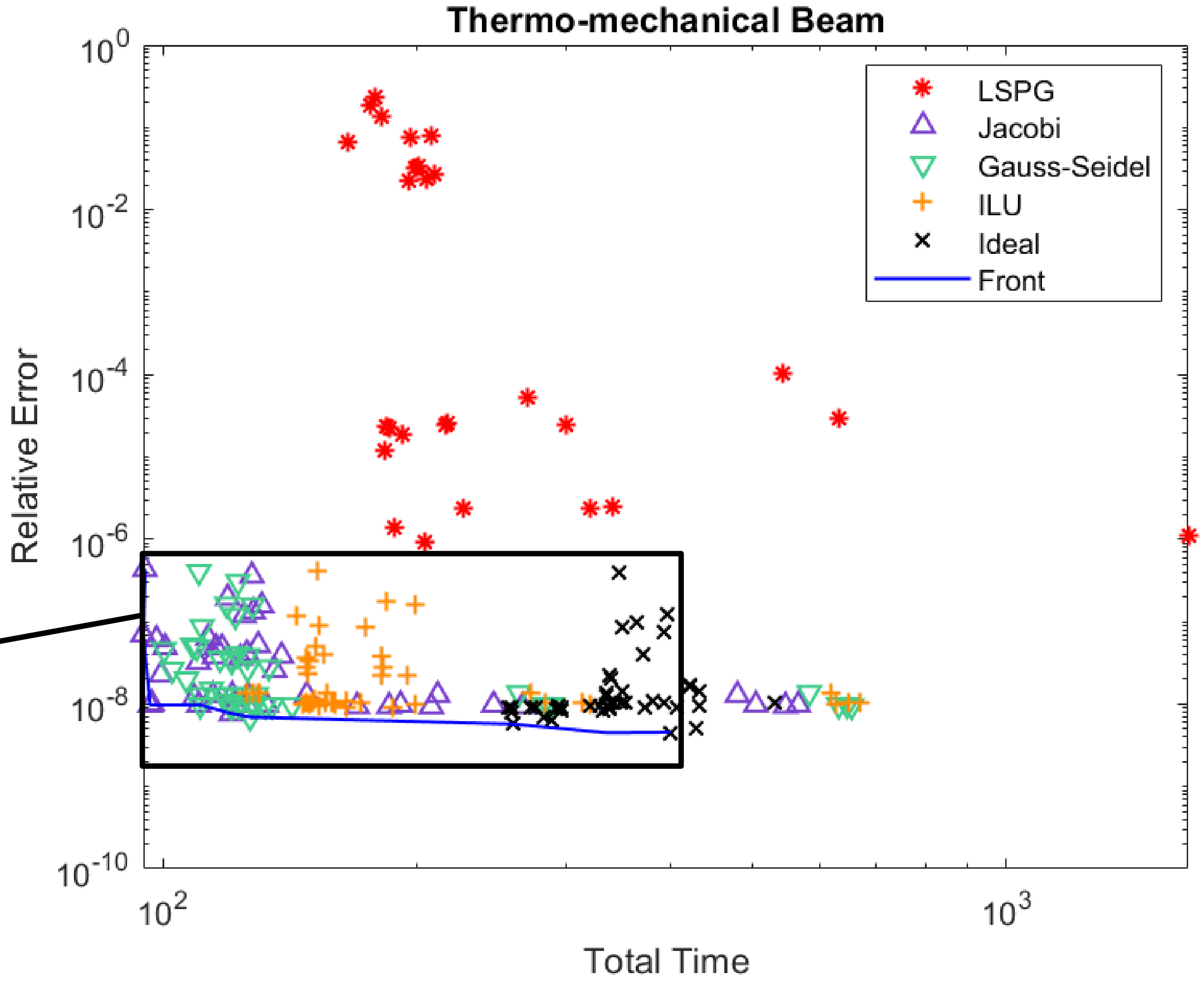
- Figures shows **CPU-times** for all ROMs considered
- In general, **preconditioned LSPG ROMs** achieve **CPU-times smaller than** unpreconditioned LSPG ROM
- As expected, the **projected solution increment** is the **most expensive** to compute in general



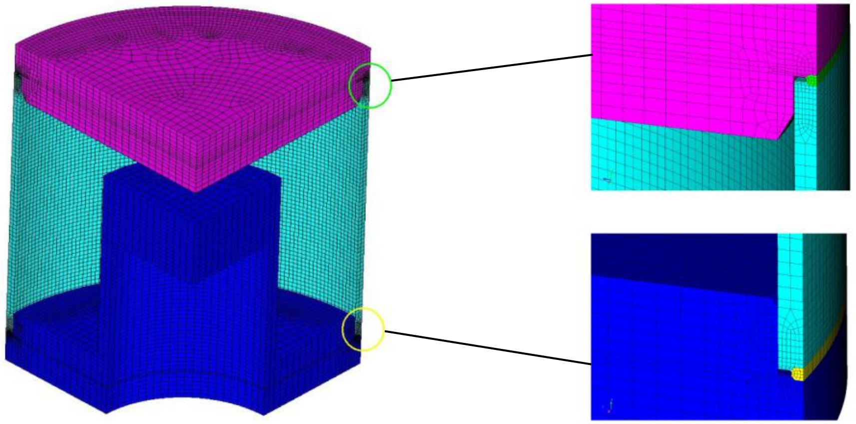
Thermo-Mechanical Beam (Albany)

Pareto plot results confirm that there is a **significant computational advantage** in applying preconditioning





Thermo-Mechanical Pressure Vessel (Albany)



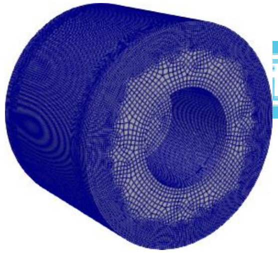
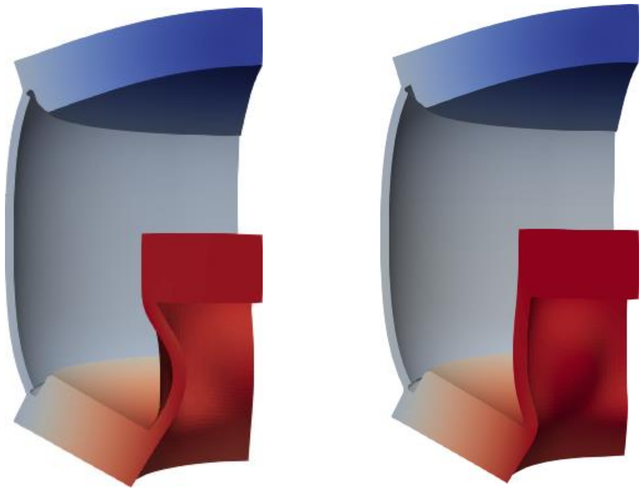


Table 1. Parameters in block $\mathcal{B}_b$ for thermo-mechanical pressure vessel problem.

Regime	Case	$E_b(\times 10^9)$ [Pa]	ν_b	$\rho_b(\times 10^{-3})$ [kg/m^3]	$T_{b,\mathrm{ref}}$ [K]
training	1	1.64424	0.39524	8.33058	311.094
	2	1.77118	0.300065	9.67843	267.396
	3	1.9893	0.32161	7.17625	223.746
	4	1.45551	0.266385	6.67746	331.116
testing	1	2.06416	0.391368	7.79804	252.102
	2	1.703	0.32	7.92	293

- Coupled **thermo-mechanical** problem involving **Neohookean** material
  - **Multi-physics problem:** temperature and displacement solutions differ by **9 orders of magnitude**; 370K dofs
- 2 sets of **material blocks**, $\mathcal{B}_a$ and $\mathcal{B}_b$, each having set of material params
  - Material parameters in block $\mathcal{B}_a$ (magenta, cyan) are fixed
  - Material parameters in block $\mathcal{B}_b$ (green, yellow, blue) are varied
- Pressure vessel is **heated** and **pressurized** from the inside
- Problem is run **quasi-statically** to pseudo-time t =720s
- **Training** is performed for 4 sets of parameters; **testing/prediction** is performed for 2 sets of parameters (see Table 1)



Testing 1 Testing 2

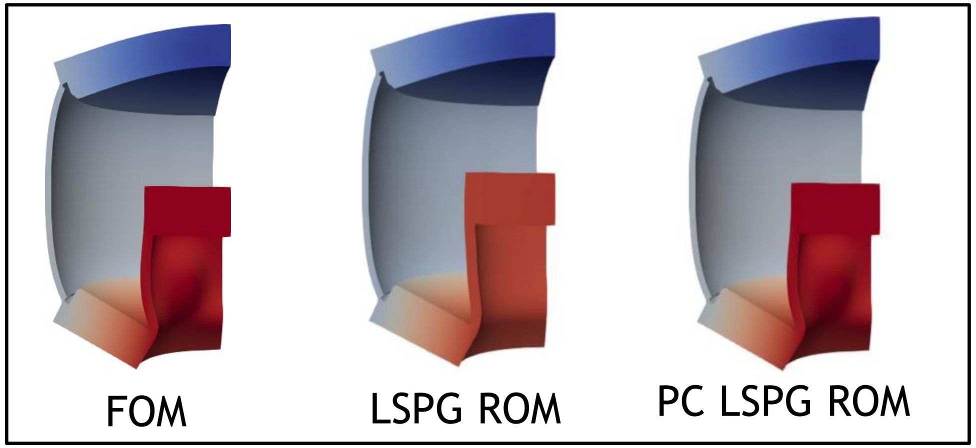
[Lindsay *et al.*, *IJNME*, 2022 (accepted)]

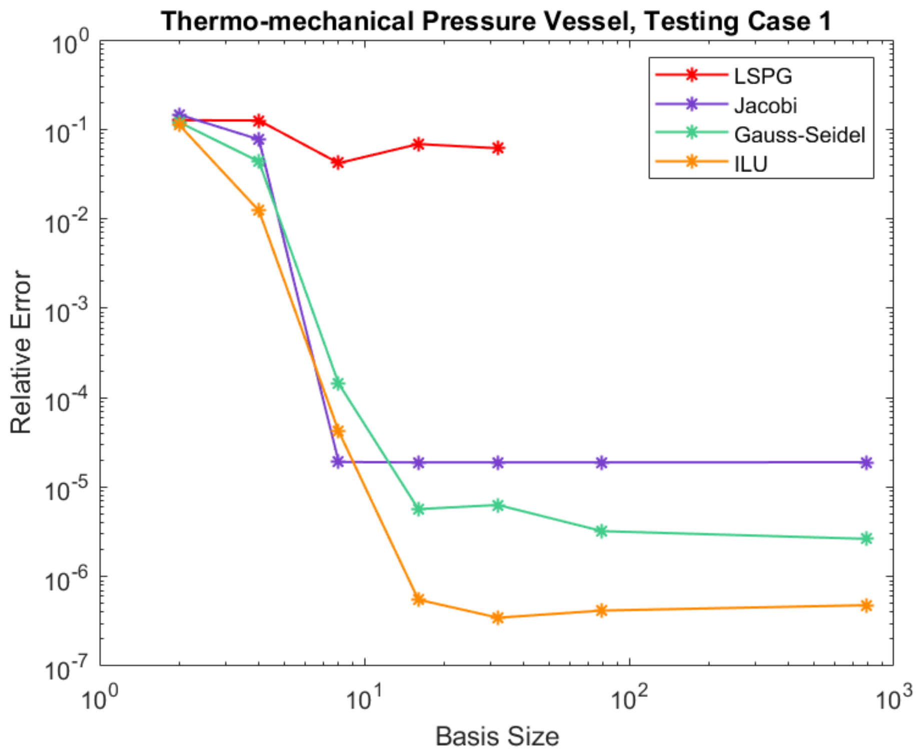
Thermo-Mechanical Pressure Vessel (Albany)

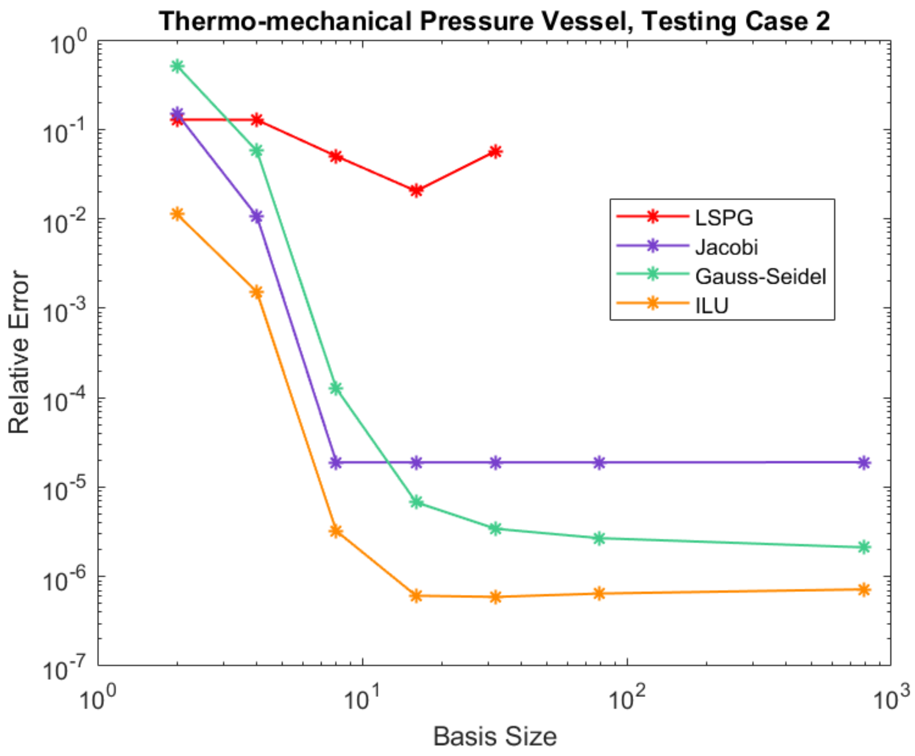
- Global relative error:

$$\epsilon := \frac{\sum_{i=0}^{P} ||\boldsymbol{x}_i - \widetilde{\boldsymbol{x}}_i||_2}{\sum_{i=0}^{P} ||\boldsymbol{x}_i||_2}$$

- Seven basis sizes evaluated: 2,4,8,16, 32,79,790



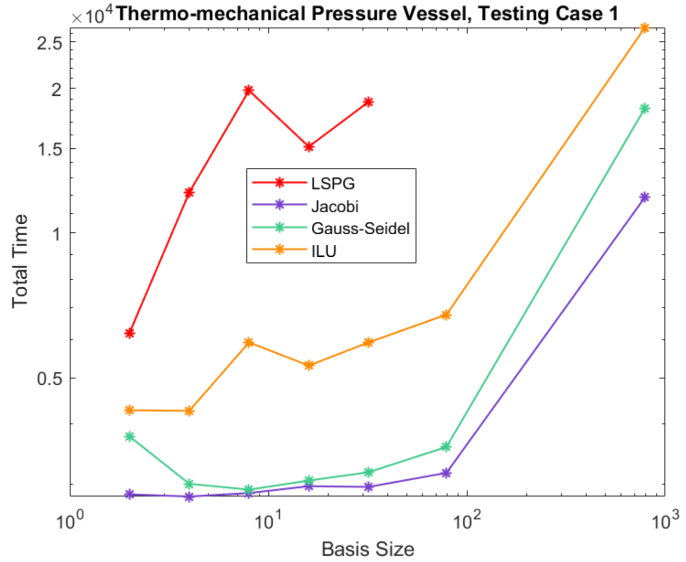


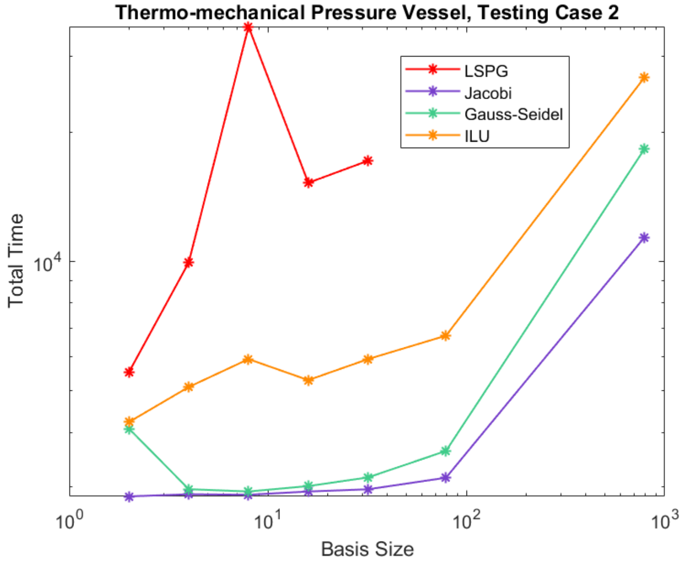


- Preconditioned (PC) LSPG ROMs are up to **5 orders of magnitude more accurate** than LSPG ROMs.
- LSPG ROMs **do not converge** for larger basis sizes.
- Accuracy is **improved** by improving the preconditioner.

Thermo-Mechanical Pressure Vessel (Albany)

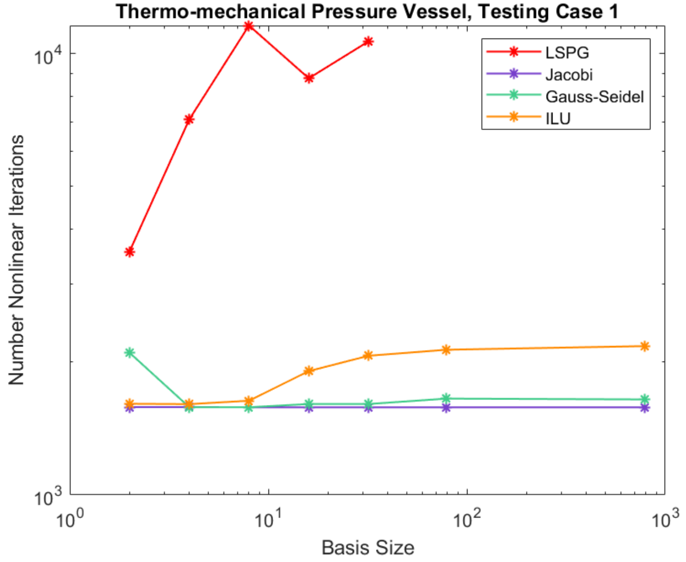
Top figures: CPU times

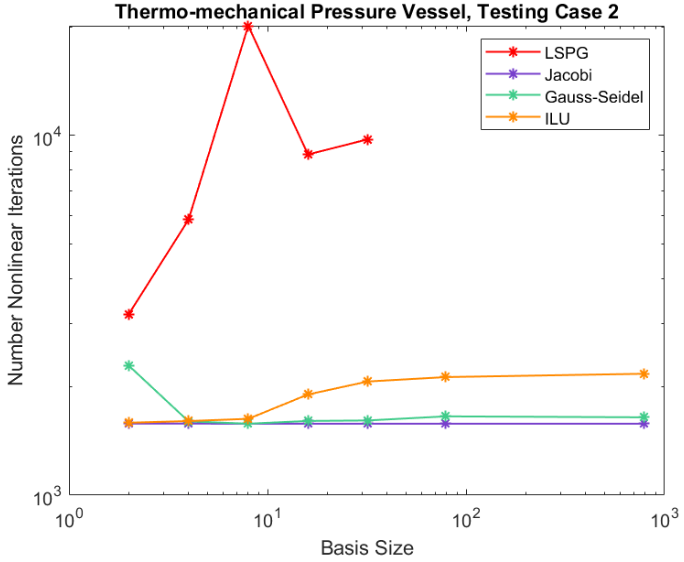




- Preconditioned LSPG ROMs are up to **12x faster** than vanilla LSPG ROMs
- More sophisticated preconditioners lead to **greater CPU** times

Bottom figures: # nonlinear iterations

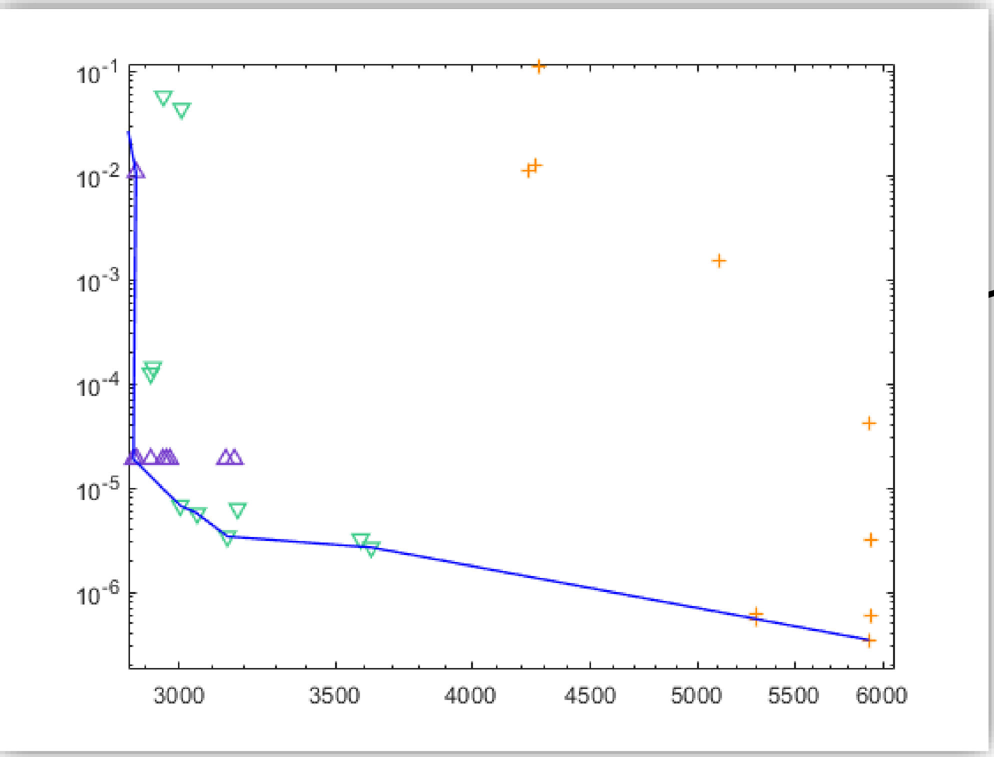


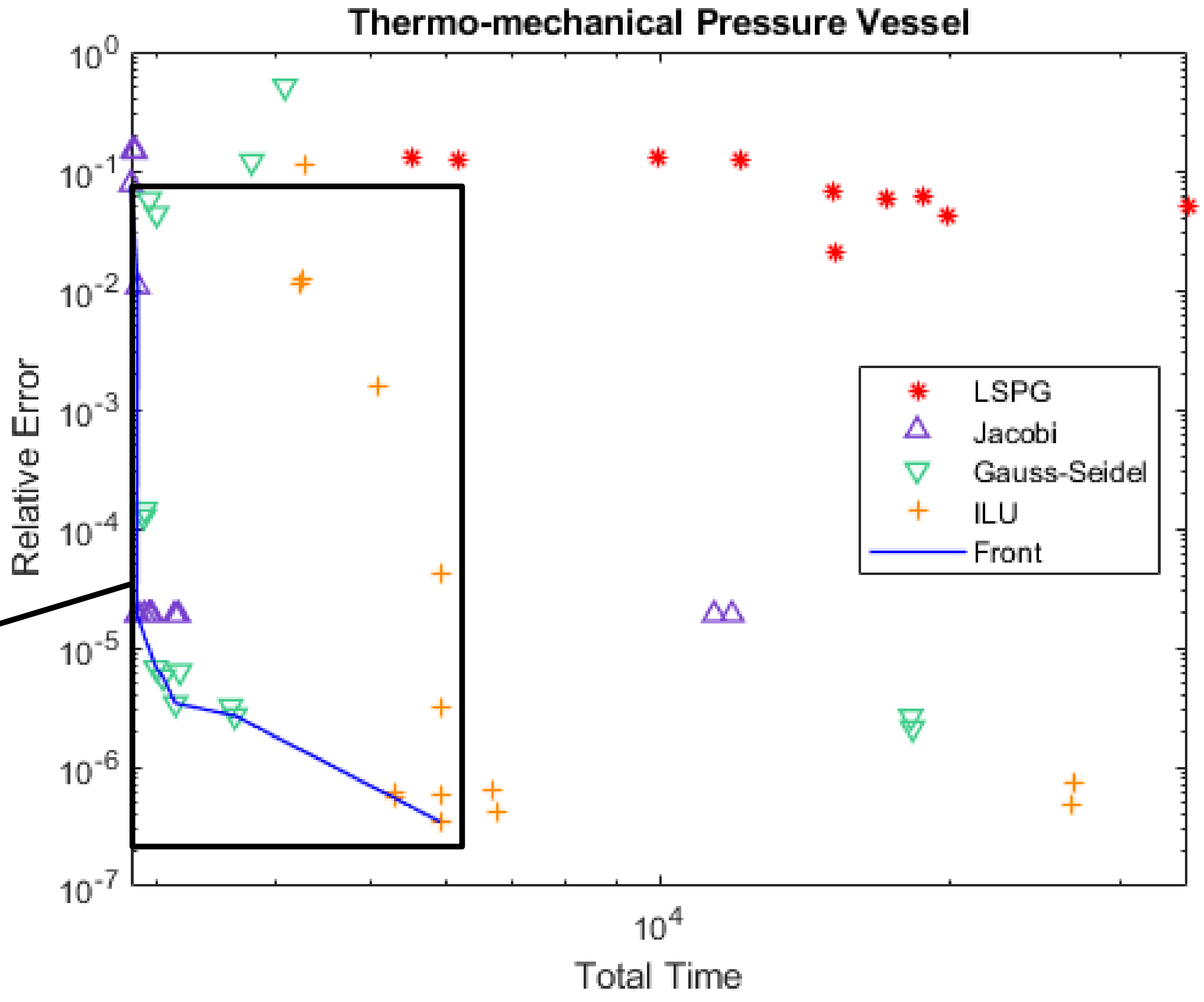


- Speed-ups are largely due to **reduction in # nonlinear iterations** (by factor of >12x)

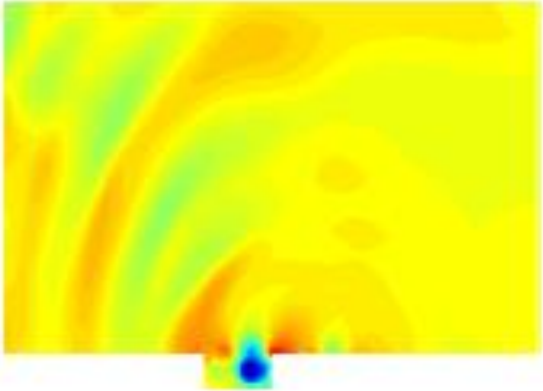
Thermo-Mechanical Pressure Vessel (Albany)

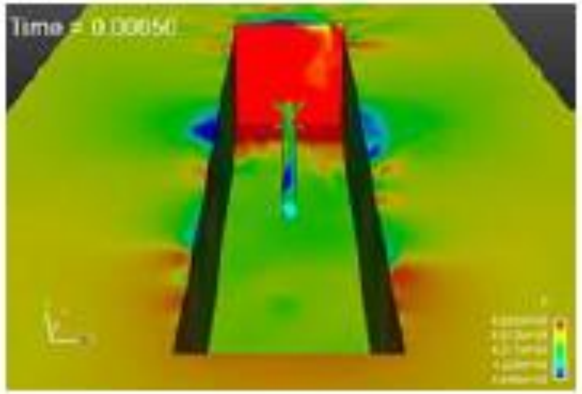
Pareto plot confirms **competitiveness** of preconditioned LSPG ROMs.



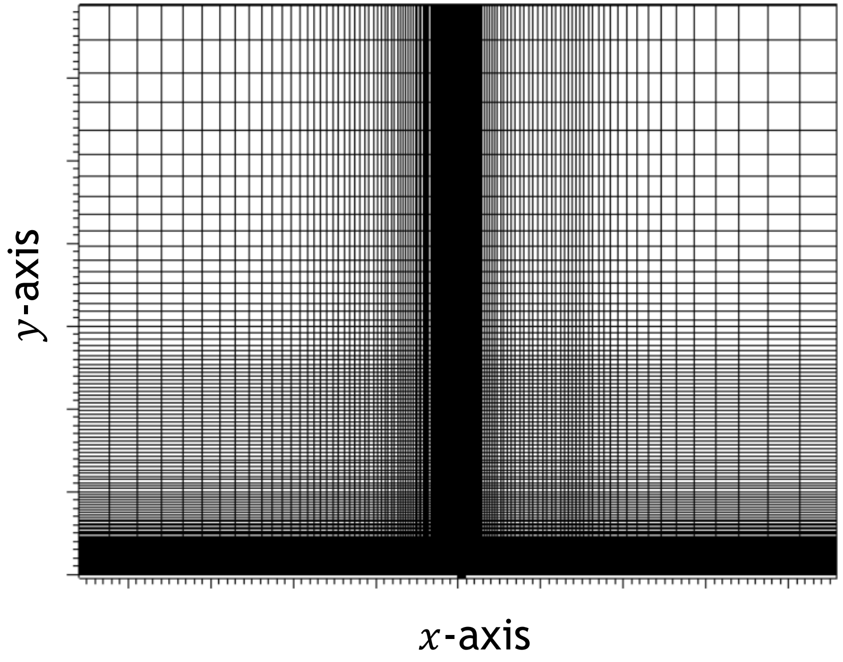


Compressible Cavity Flow (SPARC)



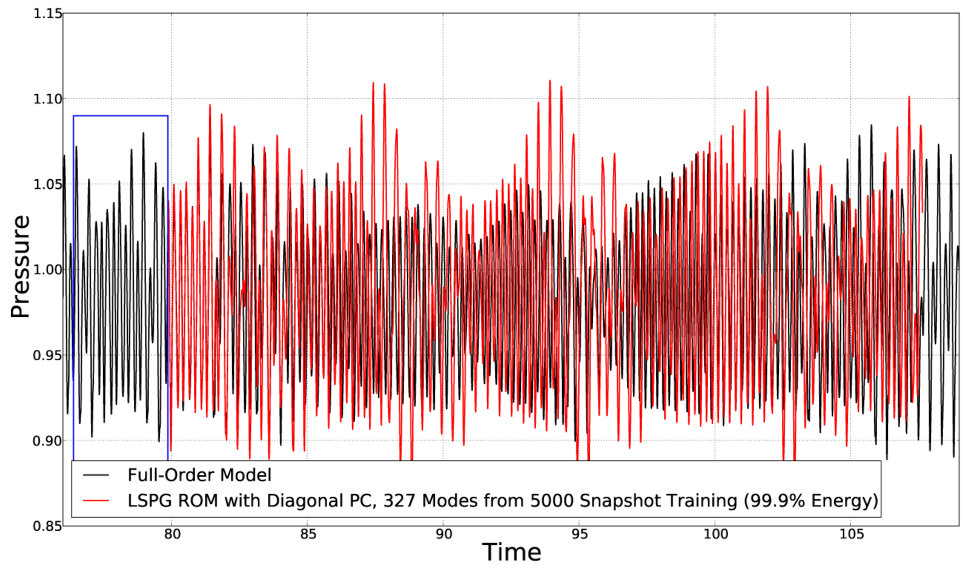


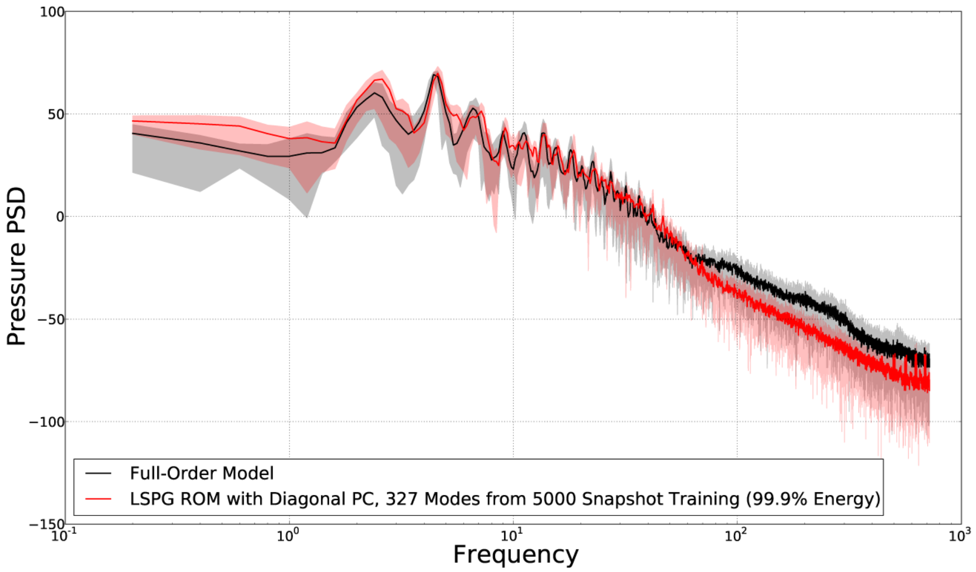
- 2D **viscous laminar flow** around an **open cavity** geometry
  - ➢ Simple model for the **captive carry scenario**
- Mach number = 0.6, Reynolds number ≈ 3000
- Problem is run **non-dimensionally**
- Domain is discretized using 104,500 hexahedral cells (right)
- Of primary interest are **long-time predictive simulations**
  - ➢ ROM is run at **same parameters** as FOM but **much longer in time**
  - ➢ **Relevant QOIs:** statistics of the flow (e.g., pressure power spectral densities or PSDs)



[Tezaur *et al*. 2017; Fike *et al*. 2018]

Compressible Cavity Flow (SPARC)





- **Figure top left:** pressure time history for a point halfway up the downstream wall of the cavity for an LSPG ROM having 327 modes with a Jacobi preconditioner
- **Figure bottom left:** pressure PSD for the signal in the top left figure (solid line is mean PSD, shaded regions indicate range of values used to construct the mean)
- **Preconditioned LSPG ROM** captures well the **pressure PSD**, including its peaks (Rossiter modes) and the **RMS OASPL**[1]
- Vanilla LSPG ROM **did not run successfully**

Method	RMS OASPL[1] in dB	% Difference from FOM
FOM	66.176	–
Ideal	67.552	2.08%
LSPG	N/A	N/A
LSPG + Jacobi PC	68.033	2.80%

[1]Overall sound pressure level

Summary & Future Work

Summary:

- Adding **preconditioning** to the LSPG formulation gives rise to ROMs with **improved accuracy** and **robustness**, especially in the predictive regime
  - Preconditioning attempts to **emulate projection of FOM solution increment onto POD basis** (the ROM "best-case scenario" for a given basis)
  - Preconditioning can **ensure all components of residual** being minimized are of the **same magnitude**
  - Preconditioning **changes the norm** in which the residual is minimized, which can **improve residual-based stability constant** bounding the ROM solution's **error**
  - Results on **predictive (across parameter space) thermo-mechanical** and **predictive (in time) compressible flow** problems are **compelling**

Ongoing/future work:

- **Manuscript in preparation:** J. Fike, P. Lindsay, K. Carlberg, I. Tezaur. "Preconditioned Least-Squares Petrov-Galerkin Reduced Order Models for Compressible Flows", *in prep*.
- **Application** of preconditioned LSPG approach to **more sophisticated problems** relevant to Sandia's mission spaces
  - Preconditioning LSPG ROMs has been helpful for **hypersonic aero**, **thermal/ablation** and **reacting hypersonic flow** problems

The End

For more details (mechanical/thermo-mechanical application), please see the following pre-print, which was just accepted for publication in *IJNME*:

Cornell University

arXiv > math > arXiv:2203.12180

Mathematics > Numerical Analysis

[Submitted on 23 Mar 2022]

Preconditioned Least-Squares Petrov-Galerkin Reduced Order Models

Payton Lindsay, Jeffrey Fike, Irina Tezaur, Kevin Carlberg

This paper introduces a methodology for improving the accuracy and efficiency of reduced order models (ROMs) constructed using the least-squares Petrov-Galerkin (LSPG) projection method through the introduction of preconditioning. Unlike prior related work, which focuses on preconditioning the linear systems arising within the ROM numerical solution procedure to improve linear solver performance, our approach leverages a preconditioning matrix directly within the LSPG minimization problem. Applying preconditioning in this way can improve ROM accuracy for several reasons. First, preconditioning the LSPG formulation changes the norm defining the residual minimization, which can improve the residual-based stability constant bounding the ROM solution's error. The incorporation of a preconditioner into the LSPG formulation can have the additional effect of scaling the components of the residual being minimized, which can be beneficial for problems with disparate scales. Importantly, we demonstrate that an 'ideal preconditioned' LSPG ROM (a ROM preconditioned with the inverse of the Jacobian of its corresponding full order model, or FOM) emulates projection of the FOM solution increment onto the reduced basis, a lower bound on the ROM solution error for a given reduced basis. By designing preconditioners that approximate the Jacobian inverse, a ROM whose error approaches this lower bound can be obtained. The proposed approach is evaluated in the predictive regime on several mechanical and thermo-mechanical problems within the Albany HPC code. We demonstrate numerically that the introduction of simple Jacobi, Gauss-Seidel and ILU preconditioners into the Proper Orthogonal Decomposition/LSPG formulation reduces significantly the ROM solution error, the reduced Jacobian condition number, the number of nonlinear iterations required to reach convergence, and the wall time.

Subjects: **Numerical Analysis (math.NA)**; Mathematical Software (cs.MS)
Cite as: arXiv:2203.12180 **[math.NA]**
(or arXiv:2203.12180v1 **[math.NA]** for this version)
https://doi.org/10.48550/arXiv.2203.12180

Thank you for your Attention!

Start of Backup Slides