PDE-Constrained Parameter Optimization

Tim Keil

October 7, 2021 – Morning session

Institute for Analysis and Numerics - WWU Münster

PDE-Constrained Parameter Optimization

- linear objective functional $\mathcal{J} : V \times \mathcal{P} \to \mathbb{R}$

$$\mathcal{J}(u, \mu) = j_\mu(u)$$

(linear) PDE-Constrained Optimization

Find a local minimizer of the reduced objective functional $\hat{\mathcal{J}}(\mu) := \mathcal{J}(u_\mu, \mu)$

$$\min_{\mu \in \mathcal{P}} \hat{\mathcal{J}}(\mu), \qquad (\hat{P})$$

where $u_\mu \in V$ denotes the solution of the state equation

$$a_\mu(u_\mu, v) = l_\mu(v) \qquad \forall\, v \in V, \qquad \text{(P.b)}$$

Assumptions:

- Real-valued Hilbert space V (e.g. $V = H_0^1(\Omega)$).
- Compact Banach space $\mathcal{P}$ with simple constraints (e.g. $\mathcal{P} = \{\mu \in \mathbb{R}^p \,|\, \mu_a \leq \mu \leq \mu_b\}$ for $p \in \mathbb{N}$).
- $a_\mu : V \times V \to \mathbb{R}$ continuous (and coercive) bilinear form.
- $l_\mu, j_\mu : V \to \mathbb{R}$ continuous linear functionals.
- a_μ, l_μ, j_μ Fréchet differentiable w.r.t. μ.
- a_μ, l_μ, j_μ parameter separable.

$$a_\mu(u, v) = \sum_{q=1}^{Q} \theta^q(\mu) a^q(u, v).$$

PDE-Constrained Parameter Optimization

- linear objective functional $\mathcal{J} : V \times \mathcal{P} \to \mathbb{R}$

$$\mathcal{J}(u, \mu) = j_\mu(u)$$

(linear) PDE-Constrained Optimization

Find a local minimizer of the reduced objective functional $\hat{\mathcal{J}}(\mu) := \mathcal{J}(u_\mu, \mu)$

$$\min_{\mu \in \mathcal{P}} \hat{\mathcal{J}}(\mu), \qquad (\hat{P})$$

where $u_\mu \in V$ denotes the solution of the state equation

$$a_\mu(u_\mu, v) = l_\mu(v) \qquad \forall\, v \in V, \qquad \text{(P.b)}$$

Challenges:

- Efficient evaluation of at least $\hat{\mathcal{J}}(\mu)$ requires efficient evaluation of the state equation (P.b).
- Exact Gradient $\nabla \hat{\mathcal{J}}(\mu)$ or even Hessian $\hat{\mathcal{H}}(\mu)$ also require the solution of additional equations.
- High dimensional parameter space $\mathcal{P}$.
- Multiscale/large scale context.

Efficient optimization method using MOR techniques:

1. How to reduce the model? <– this talk
2. Which optimization method to apply?

Full Order Model

Discretization: Computational grid τ_h, approximate space $V_h \approx V$ (e.g. CG FE)

Discrete state equation

Find $u_\mu \in V_h$ such that

$$a_\mu(u_{h,\mu}, v) = l_\mu(v) \qquad \forall\, v \in V_h.$$

Primal residual $r_\mu^{\text{pr}}(u_{h,\mu})[v] := l_\mu(v) - a_\mu(u_{h,\mu}, v)$.

- FOM objective functional

$$\hat{\mathcal{J}}(\mu) = \mathcal{J}(u_{h,\mu}, \mu)$$

How to compute gradient information $\nabla_\mu \hat{\mathcal{J}}(\mu)$ for the optimization method?

- using **finite differences** without an analytical expression of the gradient.
- using **sensitivities** of the primal solution $d_\mu u_{h,\mu}$.
- using the **adjoint** approach (Lagrangian ansatz).

Full Order Model

Sensitivities of the primal equation

For every i, find $d_{\mu_i} u_{h,\mu} \in V_h$, such that

$$a_\mu(d_{\mu_i} u_{h,\mu}, v) = \partial_{\mu_i} r_\mu^{\text{pr}}(u_{h,\mu})[v] \qquad \forall v \in V_h$$

dual equation

Find $p_{h,\mu} \in V_h$ such that

$$a_\mu(v, p_{h,\mu}) = \partial_u \mathcal{J}(u_{h,\mu}, \mu)[v] = j_\mu(v) \qquad \forall v \in V_h.$$

- FOM objective functional

$$\hat{\mathcal{J}}(\mu) = \mathcal{J}(u_{h,\mu}, \mu)$$

- Gradient with sensitivities

$$\begin{aligned}(\nabla_\mu \mathcal{J}(\mu))_i &= \partial_{\mu_i} J(u_{h,\mu}, \mu) + \partial_u J(u_{h,\mu}, \mu)[d_{\mu_i} u_{h,\mu}] \\ &= \partial_{\mu_i} J(u_{h,\mu}, \mu) + J(d_{\mu_i} u_{h,\mu}, \mu)\end{aligned}$$

Instead:

- Functional with the Lagrangian ansatz

$$\hat{\mathcal{J}}(\mu) = \mathcal{J}(u_{h,\mu}, \mu) + \underbrace{r_\mu^{\text{pr}}(u_{h,\mu})[p_{h,\mu}]}_{=0} = \mathcal{L}(u_{h,\mu}, \mu, p_{h,\mu})$$

- Gradient with the adjoint approach

$$(\nabla_\mu \mathcal{J}(\mu))_i = \partial_{\mu_i} J(u_{h,\mu}, \mu) + \partial_{\mu_i} r_\mu^{\text{pr}}(u_{h,\mu})[p_{h,\mu}]$$

Reduced Order Model

- Problem adapted Reduced Basis (RB) space: $V_r^{\text{pr}} \subseteq V_h$ of low dimension

Reduced state equation

Find $u_{r,\mu} \in V_r^{\text{pr}}$ such that

$$a_\mu(u_{r,\mu}, v) = l_\mu(v) \qquad \forall\, v \in V_r^{\text{pr}}.$$

Reduced dual equation

Find $p_{r,\mu} \in V_r^{\text{pr}}$ such that

$$a_\mu(v, p_{r,\mu}) = j_\mu(v) \qquad \forall\, v \in V_r^{\text{pr}}.$$

- Standard ROM ansatz

$$\hat{\mathcal{J}}_r(\mu) = \mathcal{J}(u_{r,\mu}, \mu)$$
$$\nabla_\mu \hat{\mathcal{J}}_r(\mu) = \nabla_\mu \mathcal{J}(u_{r,\mu}, \mu) + \nabla_\mu r_\mu^{\text{pr}}(u_{r,\mu})[p_{r,\mu}].$$

Note: This is the easy case for a symmetric a_μ. In general, the dual system should be reduced with a separate basis V_r^{du}.

Optimization methods

Reduced optimization problem

Find a locally optimal solution $\bar{\mu}_r$ of

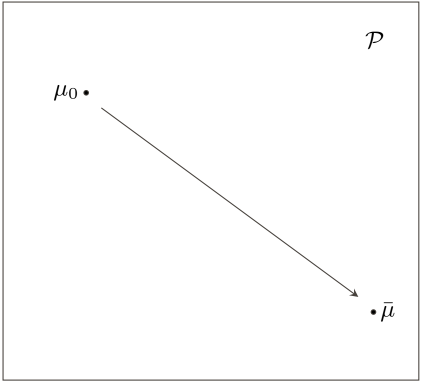
$$\min_{\mu \in \mathcal{P}} \hat{\mathcal{J}}_r(\mu). \qquad (\hat{\mathsf{P}}_r)$$

Remaining task:

1. Find snapshots for the basis construction of V_r^{pr}
2. Optimization method

Traditional: Expensive offline phase to build the model (e.g. with Greedy) + cheap online phase for the optimization

Let's try this with pyMOR



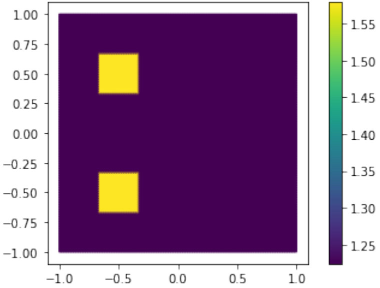
An elliptic model problem

We consider a domain $\Omega := [-1,1]^2$, a parameter set $\mathcal{P} := [0,\pi]^2$ and the elliptic equation

$$-\nabla \cdot \big(\lambda(\mu)\nabla u_\mu\big) = l$$

with data functions

$$\begin{aligned}
l(x,y) &= \tfrac{1}{2}\pi^2 \cos(\tfrac{1}{2}\pi x)\cos(\tfrac{1}{2}\pi y),\\
\lambda(\mu) &= \theta_0(\mu)\lambda_0 + \theta_1(\mu)\lambda_1,\\
\theta_0(\mu) &= 1.1 + \sin(\mu_0)\mu_1,\\
\theta_1(\mu) &= 1.1 + \sin(\mu_1),\\
\lambda_0 &= \chi_{\Omega\backslash\omega},\\
\lambda_1 &= \chi_\omega,\\
\omega &:= [-\tfrac{2}{3},-\tfrac{1}{3}]^2 \cup ([-\tfrac{2}{3},-\tfrac{1}{3}]\times[\tfrac{1}{3},\tfrac{2}{3}]),\\
\theta_{\mathcal{J}}(\mu) &:= 1 + \frac{1}{5}(\mu_0+\mu_1).
\end{aligned}$$



$$\mathcal{J}(\mu) := \theta_{\mathcal{J}}(\mu)\, f_\mu(u_\mu)$$

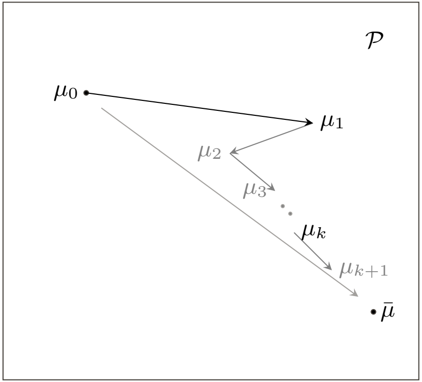
Live demo !

Further optimization methods

Bottleneck: For high dimensional parameter spaces, it becomes unfavorable to construct a "perfect" RB space for the entire parameter space. -> large offline phase.

A better idea: Adaptive enrichment

- Start with a bad RB space.
- After each optimization step, ask an **estimator** whether to update the RB space.
- Only enrich the model along the path of optimization.
- No need for an offline time.

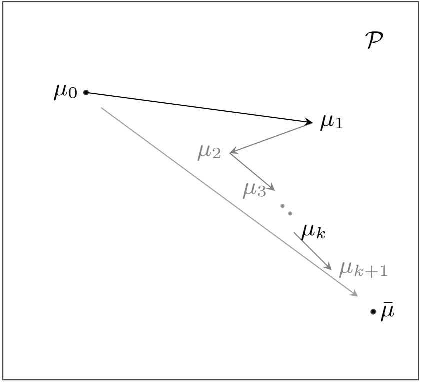


Further optimization methods

Bottleneck: For high dimensional parameter spaces, it becomes unfavorable to construct a "perfect" RB space for the entire parameter space. -> large offline phase.

A better idea: Adaptive enrichment

- Start with a bad RB space.
- After each optimization step, ask an **estimator** whether to update the RB space.
- Only enrich the model along the path of optimization.
- No need for an offline time.



More details and explanations:
`https://docs.pymor.org/main/tutorial_optimization.html`

More advances that are worth mentioning

1. For the case $V_r^{\text{pr}} \neq V_r^{\text{du}}$ (e.g. for non-symmetric operators), it is recommended to use a **corrected output functional**

$$\hat{\mathcal{J}}_r'(\mu) = \mathcal{J}(u_{r,\mu}, \mu) + r_\mu^{\text{pr}}(u_{r,\mu})[p_{r,\mu}]$$

which enables a better approximation and error estimation.
See **[Haasdonk'17]** for details.
Note: This is currently in the pyMOR pipeline. (PR #1418)

More advances that are worth mentioning

1. For the case $V_r^{\text{pr}} \neq V_r^{\text{du}}$ (e.g. for non-symmetric operators), it is recommended to use a **corrected output functional**

$$\hat{\mathcal{J}}_r'(\mu) = \mathcal{J}(u_{r,\mu}, \mu) + r_\mu^{\text{pr}}(u_{r,\mu})[p_{r,\mu}]$$

which enables a better approximation and error estimation.
See **[Haasdonk'17]** for details.
Note: This is currently in the pyMOR pipeline. (PR #1418)

2. For the adaptive procedure it is beneficial to use an **error aware trust-region algorithm** for certified optimization.
   - optimize as long as we "trust" the model, otherwise enrich.
   - Further reading:
     [Yue/Meerbergen'13], **[Qian et al'17]**, **[K. et al'21]**.
   - The numerical experiments in **[K. et al'21]** have been produced with pyMOR. Please approach me if you are interested !!

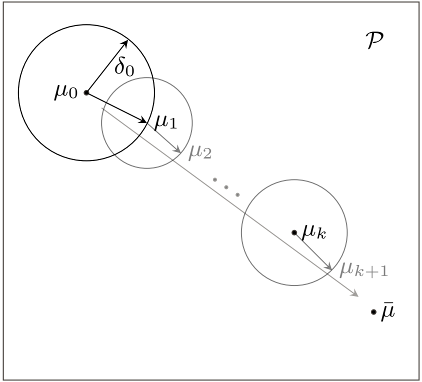


Fig: Visualization of TR algorithms.

Bibliography

B. Haasdonk.
Reduced basis methods for parametrized pdes: A tutorial introduction for stationary and instationary problems.
Model reduction and approximation: theory and algorithms, 15:65, 2017.

T. Keil, L. Mechelli, M. Ohlberger, F. Schindler, and S. Volkwein.
A non-conforming dual approach for adaptive trust-region reduced basis approximation of pde-constrained parameter optimization.
ESAIM Math. Model. Numer. Anal., 2021.

E. Qian, M. Grepl, K. Veroy, and K. Willcox.
A certified trust region reduced basis approach to pde-constrained optimization.
SIAM Journal on Scientific Computing, 39(5):S434–S460, 2017.

Y. Yue and K. Meerbergen.
Accelerating optimization of parametric linear systems by model order reduction.
SIAM Journal on Optimization, 23(2):1344–1370, 2013.

Questions? :)