A Non-Conforming Dual Approach for Adaptive Trust-Region Reduced Basis Approximation of PDE-Constrained Parameter Optimization

Seminar Advanced Numerical Methods and Applications

Peter Oehme

24.06.2022

Download Handout and Slides

Alternatively:
github.com/peoe/sem-nc-dual-trrb-pdeopt

Outline

Parametrized Partial Differential Equations

Problem Description

For a rectangular parameter space

$$\mathcal{P} := \{\mu \in \mathbb{R}^d \mid (\mu_a)_i \leq \mu_i \leq (\mu_b)_i, i = 1, \dots, d\}$$

find a solution $u \in V$ for

$$a_\mu(u, v) = l_\mu(v) \qquad \forall v \in V,$$

where

- a_μ: continuous, coercive, bilinear, and
- l_μ: continuous, linear.

Optimization over Parameter Domain

Cost Functional

Optimize some functional

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

over the parameter space $\mathcal{P}$.

Optimization over Parameter Domain

Cost Functional

Optimize some functional

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

over the parameter space $\mathcal{P}$.

In our case the quadratic cost functional

$$\mathcal{J}(u, \mu) := \Theta(\mu) + j_\mu(u) + k_\mu(u, u),$$

where

- k_μ: continuous, symmetric, bilinear,
- j_μ: continuous, linear, and
- Θ: arbitrary parameter function.

Adaptive Approach

Idea

Approximate the solution to our pPDE locally, and optimize until this model is no longer sufficient then refine the model.

Adaptive Approach

Idea

Approximate the solution to our pPDE locally, and optimize until this model is no longer sufficient then refine the model.

---

while *not converged* **do**
    Compute locally optimal solution;
    if *some condition* **then**
        Continue to next optimization step;
    else
        Change some parameter of the problem;
        Optimize again;

Adaptive Approach

Idea

Approximate the solution to our pPDE locally, and optimize until this model is no longer sufficient then refine the model.

---

while *not converged* **do**
  Compute locally optimal solution;
  if *some condition* **then**
    Continue with refinement;
  else if *some other condition* **then**
    Repeat optimization step with smaller domain;
  else if *some third condition* **then**
    Refine and possibly repeat optimization step;

Adaptive Approach

Idea

Approximate the solution to our pPDE locally, and optimize until this model is no longer sufficient then refine the model.

Benefits of the Adaptive Approach

We only adapt locally.

We **do not** construct a reduced basis for the entire domain $\mathcal{P}$!

Full Order Model

Standard Finite Element Formulation

For a finite element space find $u \in V$ such that

$$a_\mu(u, v) = l_\mu(v) \quad \forall v \in V$$

Reduced Basis Method, c.f.e.g. [OR15]

Idea

Construct an N-dimensional subspace $V_N \subseteq V$ of the full order model (FOM) space V.

Reduced Basis Method, c.f.e.g. [OR15]

Idea

Construct an N-dimensional subspace $V_N \subseteq V$ of the full order model (FOM) space V.

Construction:

- Greedy Construction
- Discrete Empirical Interpolation
- Proper Orthogonal Decomposition

Comparison FEM/RBM

FE Formulation

For a finite element space find $u \in V$ such that

$$a_\mu(u, v) = l_\mu(v) \quad \forall v \in V$$

RB Formulation

For a finite subspace $V_N \subseteq V$ find $u_N \in V_N$ such that

$$a_\mu(u_N, v_N) = l_\mu(v_N) \quad \forall v_N \in V_N$$

Comparison FEM/RBM

FEM

- sparse system matrix
- large basis

RBM

- dense system matrix
- small basis

Comparison FEM/RBM

FEM

- sparse system matrix
- large basis
- no offline/online decomposition

RBM

- dense system matrix
- small basis
- offline/online decomposition

Offline/Online Decomposition

Parameter Separability

For $u, v \in V_N, p_a, p_l \in \mathbb{N}$

$$a_\mu(u, v) = \sum_{p=0}^{p_a} \theta_p^a(\mu) a_p(u, v),$$
$$l_\mu(v) = \sum_{p=0}^{p_l} \theta_p^l(\mu) l_p(v),$$

where

- θ_i^a, θ_i^l parameter functions, and
- a_i, l_i **parameter independent**.

Offline/Online Decomposition

Offline Assembly

We get the **offline part**

$$\mathbb{A}_p^{i,j} := a_p(\varphi_i, \varphi_j), i = 1, \ldots, N, p = 1, \ldots, p_a,$$
$$\mathbb{L}_p^{i} := l_p(\varphi_i), i = 1, \ldots, N, p = 1, \ldots, p_l,$$

for the reduced basis $\{\varphi_1, \ldots, \varphi_N\}$ of V_N.

Offline/Online Decomposition

Offline Assembly

$$\mathbb{A}_p^{i,j} := a_p(\varphi_i, \varphi_j), \qquad \mathbb{L}_p^i := l_p(\varphi_i)$$

Online Assembly

We get the **online part**

$$\mathbb{A} := \sum_{p=1}^{p_a} \theta_p^a(\mu)\mathbb{A}_p, \qquad \mathbb{L} := \sum_{p=1}^{p_l} \theta_p^l(\mu)\mathbb{L}_p,$$

for the reduced basis $\{\varphi_1, \ldots, \varphi_N\}$ of V_N.

Notes on Differentiability

Directional Derivative, c.f. [Hin+09]

$F : X \to Y$ if

$$dF(x)[h] := \lim_{t \to 0} \frac{F(x + th) - F(x)}{t} \in Y$$

exists.

Notes on Differentiability

Directional Derivative, c.f. [Hin+09]

$F : X \to Y$ if

$$dF(x)[h] := \lim_{t \to 0} \frac{F(x + th) - F(x)}{t} \in Y$$

exists.

Gâteaux Derivative, c.f. [Hin+09]

$F : X \to Y$ if $dF(x) \in \mathcal{L}(X, Y)$.

Notes on Differentiability

Gâteaux Derivative, c.f. [Hin+09]

$F : X \to Y$ if $dF(x) \in \mathcal{L}(X, Y)$.

Fréchet Derivative, c.f. [Hin+09]

$F : X \to Y$ if for $\|h\|_x \to 0$ it holds that

$$\|F(x+th) - F(x) - dF(x)[h]\|_Y = o(\|h\|_X).$$

Notes on Differentiability

Bilinear Form

The chain rule also holds true:

$$\begin{aligned} d_\mu a_\mu(u_\mu, v_\mu) \cdot \nu &= \partial_\mu a_\mu(u_\mu, v_\mu) \cdot \nu + \partial_u a_\mu(u_\mu, v_\mu)[d_\nu u_\mu] + \partial_v a_\mu(u_\mu, v_\mu)[d_\nu v_\mu] \\ &= \partial_\mu a_\mu(u_\mu, v_\mu) \cdot \nu + a_\mu(d_\nu u_\mu, v_\mu) + a_\mu(u_\mu, d_\nu v_\mu). \end{aligned}$$

$d_\nu u_\mu, d_\nu v_\mu$ are also called **sensitivities**.

PDE Constrained Optimization

Motivation

Parameter dependence in PDEs:

- material properties
- velocities
- heat sources/sinks

PDE Constrained Optimization

Motivation

Parameter dependence in PDEs:

- material properties
- velocities
- heat sources/sinks

How to determine optimal parameters?

PDE Constrained Optimization

Problem Description, c.f. [Hin+09]

We consider

$$\min_{\mu \in \mathcal{P}} \mathcal{J}(u, \mu) \qquad s.t.$$
$$e(u, \mu) = 0,$$

where $e_\mu(u, v)$ encodes the PDE constraint

$$e(u, \mu) := l_\mu(v) - a_\mu(u, v) \qquad \forall v \in V \text{ or } V_N$$

Trust-Region Method

Model function, c.f. [Kel99]

Cost functional $\mathcal{J}$ too expensive to efficiently optimize.

Trust-Region Method

Model function, c.f. [Kel99]

Cost functional $\mathcal{J}$ too expensive to efficiently optimize.

Idea: Replace with simpler function $m^{(k)}$ modelling $\mathcal{J}$ around some point $x^{(k)}$.

Trust-Region Method

Model function, c.f. [Kel99]

Replace $\mathcal{J}$ with simpler **model function** $m^{(k)}$ around some point $x^{(k)}$.

Trust-Region

Region where $m^{(k)}$ approximates $\mathcal{J}$ well:

$$\Delta^{(k)} := \left\{ x \in X \mid \left\| x - x^{(k)} \right\|_X \leq \delta^{(k)} \right\}.$$

- $\Delta^{(k)}$: **trust-region**
- $\delta^{(k)}$: **trust-radius**

Trust-Region Method

Trust-Region Method

Idea: Optimize model function locally, i.e.

$$\underset{\|s\| \leq \delta^{(k)}}{\arg\min}\, m^{(k)}(x^{(k)} + s)$$

$(s.t.$ constraints are satisfied$)$.

Trust-Region Method

Trust-Region Method

Idea: Optimize model function locally, i.e.

$$\operatorname*{arg\,min}_{\|s\| \leq \delta^{(k)}} m^{(k)}(x^{(k)} + s)$$

$$(s.t. \text{ constraints are satisfied}).$$

Note: This implies use of a local optimization algorithm!

Enrichment

Problem: Updating of the TR parameters

Enrichment

Problem: Updating of the TR parameters

- Enrich model function every iteration

Enrichment

Problem: Updating of the TR parameters

- Enrich model function if necessary

Enrichment

Problem: Updating of the TR parameters

- Enrich model function if necessary
- Shrink trust-radius if necessary

Enrichment

Problem: Updating of the TR parameters

- Enrich model function if necessary
- Shrink trust-radius if necessary, enlarge trust-radius if possible

BFGS Method

Newton Method, c.f. [NW06]

$$
\begin{aligned}
x^{(k+1)} &:= x^{(k)} + \kappa^{(k)} d^{(k)}, \\
d^{(k)} &:= -\mathcal{H}(x^{(k)})^{-1} \nabla \mathcal{J}(x^{(k)})
\end{aligned}
$$

BFGS Method

Quasi Newton Method, c.f. [Kel99; NW06]

$$x^{(k+1)} := x^{(k)} + \kappa^{(k)} d^{(k)},$$
$$d^{(k)} := -{\mathcal{H}^{(k)}}^{-1} \nabla \mathcal{J}(x^{(k)}),$$

$\mathcal{H}^{(k)}$ is symmetric, pos. def. matrix updated every iteration to approximate the true Hessian.

BFGS Method

BFGS Update Formula

$$\begin{aligned}
\mathcal{H}^{(k+1)} &:= \mathcal{H}^{(k)} + \frac{(y^{(k)} - \mathcal{H}^{(k)} s^{(k)})(y^{(k)} - \mathcal{H}^{(k)} s^{(k)})^T}{(y^{(k)} - \mathcal{H}^{(k)} s^{(k)})^T s^{(k)}}, \\
s^{(k)} &:= x^{(k+1)} - x^{(k)}, \\
y^{(k+1)} &:= \nabla \mathcal{J}(x^{(k+1)}) - \nabla \mathcal{J}(x^{(k)})
\end{aligned}$$

Optimality Conditions

First Order Necessary Conditions, c.f. [Kei+21]

For any locally optimal pair (u, μ) there exists an optimal dual variable $p \in V$ such that the following conditions are satisfied

$$
\begin{aligned}
a_\mu(u, v) &= l_\mu(v) && \forall v \in V \\
a_\mu(u, p) &= \partial_u \mathcal{J}(u, \mu)[v] && \forall v \in V \\
0 &\leq (\nabla_\mu \mathcal{J}(u, \mu) + \nabla_\mu l_\mu(p) - \nabla_\mu a_\mu(u, p)) \cdot (\nu - \mu) && \forall \nu \in \mathcal{P}.
\end{aligned}
$$

Primal and Dual Equations

Primal Residual

For $u, v \in V$, $\mu \in \mathcal{P}$ we define

$$Res_\mu^{pr}(u)[v] := l_\mu(v) - a_\mu(u, v).$$

Dual Residual

For $u, v, p \in V$, $\mu \in \mathcal{P}$ we define

$$Res_\mu^{du}(u, p)[v] := j_\mu(v) + 2k_\mu(v, u) - a_\mu(v, p).$$

Lagrangian: $\mathcal{L}(u, \mu, p) := \mathcal{J}(u, \mu) - Res_\mu^{pr}(u)[p]$

Optimality Conditions — Standard Approach

Reduced Functional

For $\mu \in \mathcal{P}$ we define

$$J_N(\mu) := \mathcal{J}(u_{N,\mu}, \mu),$$

where $u_{N,\mu}$ defines the reduced solution for μ.

Optimality Conditions — Standard Approach

Reduced Functional

For $\mu \in \mathcal{P}$ we define

$$J_N(\mu) := \mathcal{J}(u_{N,\mu}, \mu),$$

where $u_{N,\mu}$ defines the reduced solution for μ.

Then the gradient for all $p \in V_N^{du}$ is

$$(\nabla_\mu J_N(\mu))_i = \partial_u \mathcal{L}(u_{N,\mu}, \mu, p)[d_{\mu_i} u_{N,\mu}] + \partial_{\mu_i} \mathcal{L}(u_{N,\mu}, \mu, p) \quad \forall p \in V_N^{du}.$$

Optimality Conditions — Standard Approach

Then the gradient for all $p \in V_N^{du}$ is

$$\left(\nabla_\mu J_N(\mu)\right)_i = \partial_u \mathcal{L}(u_{N,\mu}, \mu, p)[d_{\mu_i} u_{N,\mu}] + \partial_{\mu_i} \mathcal{L}(u_{N,\mu}, \mu, p) \quad \forall p \in V_N^{du}.$$

Inexact Gradient

For better numerical results later on, we define

$$\left(\tilde{\nabla}_\mu J_N(\mu)\right)_i := \partial_{\mu_i} \mathcal{J}(u_{N,\mu}, \mu) + \partial_{\mu_i} Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}].$$

Optimality Conditions — Standard Approach

Then the gradient for all $p \in V_N^{du}$ is

$$(\nabla_\mu J_N(\mu))_i = \partial_u \mathcal{L}(u_{N,\mu}, \mu, p)[d_{\mu_i} u_{N,\mu}] + \partial_{\mu_i} \mathcal{L}(u_{N,\mu}, \mu, p) \quad \forall p \in V_N^{du}.$$

Inexact Gradient

For better numerical results later on, we define

$$\left(\tilde{\nabla}_\mu J_N(\mu)\right)_i := \partial_{\mu_i} \mathcal{J}(u_{N,\mu}, \mu) + \partial_{\mu_i} Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}].$$

$$\Rightarrow (\nabla_\mu J_N(\mu))_i = \partial_u \mathcal{L}(u_{N,\mu}, \mu, p_{N,\mu})[d_{\mu_i} u_{N,\mu}] + \left(\tilde{\nabla}_\mu J_N(\mu)\right)_i$$

Optimality Conditions — NCD-Corrected Approach

NCD-Corrected Functional, c.f. [Kei+21]

For $u_{N,\mu} \in V_N^{pr}$, $p_{N,\mu} \in V_N^{du}$ we define

$$\mathcal{J}_N(\mu) := \mathcal{L}(u_{N,\mu}, \mu, p_{N,\mu}) = J_N(\mu) + Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}]$$

Optimality Conditions — NCD-Corrected Approach

NCD-Corrected Gradient, Standard Approach, c.f. [Kei+21]

For $v_{N,\mu} \in V_N^{pr}$, $p_{N,\mu} \in V_N^{du}$ we get

$$\begin{aligned}(\nabla_\mu \mathcal{J}_N(\mu))_i &= \partial_{\mu_i} \mathcal{J}(u_{N,\mu}, \mu) \\ &\qquad + \partial_{\mu_i} Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu} + w_{N,\mu}] - \partial_{\mu_i} Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[z_{N,\mu}],\end{aligned}$$

where $z_{N,\mu} \in V_N^{du}$, $w_{N,\mu} \in V_N^{pr}$ satisfy

$$\begin{aligned} a_\mu(z_{N,\mu}, q) &= -Res_\mu^{pr}(u_{N,\mu})[q] && \forall q \in V_N^{du}, \text{ and} \\ a_\mu(v, w_{N,\mu}) &= Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[v] - 2k_\mu(z_{N,\mu}, v) && \forall v \in V_N^{pr}. \end{aligned}$$

Optimality Conditions — NCD-Corrected Approach

NCD-Corrected Gradient, Sensitivity Approach, c.f. [Kei+21]

For $v_{N,\mu} \in V_N^{pr}, p_{N,\mu} \in V_N^{du}$ we alternatively get

$$\begin{aligned}(\nabla_\mu \mathcal{J}_N(\mu))_i = {} & \partial_{\mu_i}\mathcal{J}(u_{N,\mu},\mu) + \partial_{\mu_i} Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}] \\ & + Res_\mu^{pr}(u_{N,\mu})[d_{\mu_i} p_{N,\mu}] + Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[d_{\mu_i} u_{N,\mu}],\end{aligned}$$

where the sensitivities for some $\nu \in \mathcal{P}$ are defined by

$$\begin{aligned} a_\mu(d_\nu u_{N,\mu}, v) &= \partial_\mu Res_\mu^{pr}(u_{N,\mu})[v] \cdot \nu && \forall v \in V_N^{pr} \\ a_\mu(q, d_\nu p_{N,\mu}) &= \partial_\mu Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[q] \cdot \nu + 2k_\mu(q, d_\nu u_{N,\mu}) && \forall q \in V_N^{du} \end{aligned}$$

Optimality Conditions — NCD-Corrected Approach

Problem: V_N^{pr}, V_N^{du} must not be good approximation spaces for the sensitivities.

Optimality Conditions — NCD-Corrected Approach

Problem: V_N^{pr}, V_N^{du} must not be good approximation spaces for the sensitivities.

Solution: Compute the approximated sensitivities in $V_N^{pr,d_{\mu_i}}, V_N^{du,d_{\mu_i}} \subseteq V$.

Optimality Conditions — NCD-Corrected Approach

Problem: V_N^{pr}, V_N^{du} must not be good approximation spaces for the sensitivities.
Solution: Compute the approximated sensitivities in $V_N^{pr,d_{\mu_i}}, V_N^{du,d_{\mu_i}} \subseteq V$.

Approximate Sensitivities

We define the approximate sensitivities as solutions to

$$
\begin{aligned}
a_\mu(\tilde{d}_{\mu_i} u_{N,\mu}, v) &= \partial_\mu Res_\mu^{pr}(u_{N,\mu})[v] \cdot e_i && \forall v \in V_N^{pr,d_{\mu_i}}, \text{ and} \\
a_\mu(q, \tilde{d}_{\mu_i} p_{N,\mu}) &= \partial_\mu Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[q] \cdot e_i + 2k_\mu(q, \tilde{d}_{\mu_i} u_{N,\mu}) && \forall q \in V_N^{du,d_{\mu_i}}.
\end{aligned}
$$

Optimality Conditions — NCD-Corrected Approach

$$
\begin{aligned}
a_\mu(\tilde{d}_{\mu_i} u_{N,\mu}, v) &= \partial_\mu Res_\mu^{pr}(u_{N,\mu})[v] \cdot e_i && \forall v \in V_N^{pr,d_{\mu_i}}, \\
a_\mu(q, \tilde{d}_{\mu_i} p_{N,\mu}) &= \partial_\mu Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[q] \cdot e_i + 2k_\mu(q, \tilde{d}_{\mu_i} u_{N,\mu}) && \forall q \in V_N^{du,d_{\mu_i}}.
\end{aligned}
$$

Inexact Gradient of NCD-Corrected Functional

We get for the approximate sensitivities:

$$
\begin{aligned}
\big(\tilde{\nabla}_\mu \mathcal{J}_N(\mu)\big)_i := {}& \partial_{\mu_i} \mathcal{J}(u_{N,\mu}, \mu) + \partial_{\mu_i} Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}] \\
& + Res_\mu^{pr}(u_{N,\mu})[\tilde{d}_{\mu_i} p_{N,\mu}] + Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[\tilde{d}_{\mu_i} u_{N,\mu}].
\end{aligned}
$$

a posteriori Error Estimates

Model Reduction Error, c.f. [Qia+17; Kei+21]

For some $\mu \in \mathcal{P}$ we consider the solutions $u_\mu \in V$, $u_{N,\mu} \in V_N^{pr}$, $p_\mu \in V$, $p_{N,\mu} \in V_N^{du}$. Then it holds that

$$\|u_\mu - u_{N,\mu}\| \leq \Delta_{pr}(\mu) := {\alpha_\mu}^{-1} \left\| Res_\mu^{pr}(u_{N,\mu}) \right\|, \text{ and}$$
$$\|p_\mu - p_{N,\mu}\| \leq \Delta_{du}(\mu) := {\alpha_\mu}^{-1} \left(2\gamma_{k_\mu} \Delta_{pr}(\mu) + \left\| Res_\mu^{du}(u_{N,\mu}, p_{N,\mu}) \right\| \right).$$

a posteriori Error Estimates

$$\|u_\mu - u_{N,\mu}\| \leq \Delta_{pr}(\mu), \qquad \|p_\mu - p_{N,\mu}\| \leq \Delta_{du}(\mu)$$

Output Model Reduction Error, c.f. [Qia+17; Kei+21]

For $\mu \in \mathcal{P}, u_\mu \in V, u_{N,\mu} \in V_N^{pr}, p_\mu \in V, p_{N,\mu} \in V_N^{du}$ we have

$$\begin{aligned}|\mathcal{J}(u_\mu, \mu) - J_N(\mu)| \leq \Delta_{J_N}(\mu) &:= \Delta_{pr}(\mu) \left\| Res_\mu^{du}(u_{N,\mu}, p_{N,\mu}) \right\| \\ &\quad + \Delta_{pr}(\mu)^2 \gamma_{k_\mu} + \left| Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}] \right|, \text{ and} \\ |\mathcal{J}(u_\mu, \mu) - \mathcal{J}_N(\mu)| \leq \Delta_{\mathcal{J}_N}(\mu) &:= \Delta_{pr}(\mu) \left\| Res_\mu^{du}(u_{N,\mu}, p_{N,\mu}) \right\| + \Delta_{pr}(\mu)^2 \gamma_{k_\mu}.\end{aligned}$$

a posteriori Error Estimates

$$|\mathcal{J}(u_\mu, \mu) - J_N(\mu)| \leq \Delta_{J_N}(\mu), \qquad |\mathcal{J}(u_\mu, \mu) - \mathcal{J}_N(\mu)| \leq \Delta_{\mathcal{J}_N}(\mu)$$

Output Gradient Model Reduction Error, c.f. [Qia+17; Kei+21]

For $\mu \in \mathcal{P}$, $u_\mu \in V$, $u_{N,\mu} \in V_N^{pr}$, $p_\mu \in V$, $p_{N,\mu} \in V_N^{du}$ we have

$$\left\|\nabla_\mu \mathcal{J}(u_\mu, \mu) - \nabla_\mu \mathcal{J}_N(\mu)\right\|_2 \leq \Delta_{\nabla \mathcal{J}_N}(\mu) := \left\|\Delta^*_{\nabla \mathcal{J}_N}(\mu)\right\|_2, \text{ and}$$
$$\left\|\nabla_\mu \mathcal{J}(u_\mu, \mu) - \tilde{\nabla}_\mu \mathcal{J}_N(\mu)\right\|_2 \leq \Delta_{\tilde{\nabla} \mathcal{J}_N}(\mu) := \left\|\Delta^*_{\tilde{\nabla} \mathcal{J}_N}(\mu)\right\|_2.$$

TR-RB Algorithm

Choice of Model Function

We choose

$$m^{(k)}(\eta) := \mathcal{J}_N^{(k)}(\mu^{(k)} + \eta),$$

where $\eta \in \mathcal{P}$.

TR-RB Algorithm

Choice of Model Function

We choose

$$m^{(k)}(\eta) := \mathcal{J}_N^{(k)}(\mu^{(k)} + \eta),$$

where $\eta \in \mathcal{P}$.

Note: $\mathcal{J}_N \neq \mathcal{J}_N^{(k)}$! These are based upon different spaces $V_N^{pr,k}, V_N^{du,k}$.

TR-RB Algorithm

while *not converged* **do**
    Compute locally optimal solution;
    if *some condition* **then**
        Continue with refinement;
    else if *some other condition* **then**
        Repeat optimization step with smaller domain;
    else if *some third condition* **then**
        Refine and possibly repeat optimization step;

TR-RB Algorithm

while *not converged* **do**
  Compute locally optimal solution;
  if $\mathcal{J}_N^{(k)}(\mu^{(k+1)}) + \Delta_{\mathcal{J}_N^{(k)}}(\mu^{(k+1)}) < \mathcal{J}_N^{(k)}(\mu^{(k)})$ **then**
    Continue with refinement;
  else if *some other condition* **then**
    Repeat optimization step with smaller domain;
  else if *some third condition* **then**
    Refine and possibly repeat optimization step;

TR-RB Algorithm

while *not converged* **do**
  Compute locally optimal solution;
  if *output decrease sufficiently small* **then**
    Continue with refinement;
  else if *some other condition* **then**
    Repeat optimization step with smaller domain;
  else if *some third condition* **then**
    Refine and possibly repeat optimization step;

TR-RB Algorithm

while *not converged* **do**

Compute locally optimal solution;

if *output decrease sufficiently small* **then**

Continue with refinement;

else if $\mathcal{J}_N^{(k)}(\mu^{(k+1)}) + \Delta_{\mathcal{J}_N^{(k)}}(\mu^{(k+1)}) > \mathcal{J}_N^{(k)}(\mu^{(k)})$ **then**

Repeat optimization step with smaller domain;

else if *some third condition* **then**

Refine and possibly repeat optimization step;

TR-RB Algorithm

while *not converged* **do**
    Compute locally optimal solution;
    if *output decrease sufficiently small* **then**
        Continue with refinement;
    else if *output decrease not small enough* **then**
        Repeat optimization step with smaller domain;
    else if *some third condition* **then**
        Refine and possibly repeat optimization step;

TR-RB Algorithm

while *not converged* **do**
  Compute locally optimal solution;
  if *output decrease sufficiently small* **then**
    Continue with refinement;
  else if *output decrease not small enough* **then**
    Repeat optimization step with smaller domain;
  else
    Refine and possibly repeat optimization step;

TR-RB Algorithm

while *not converged* **do**
  Compute locally optimal solution;
  if *output decrease sufficiently small* **then**
    Refine and potentially enlarge trust-radius;
  else if *output decrease not small enough* **then**
    Repeat optimization step with smaller domain;
  else
    Refine and check whether to repeat optimization (shrinking, enlarging, or keeping of trust-radius depending on outcome);

TR-RB Algorithm

Inner Loop (BFGS), c.f. [Qia+17; Kei+21]

The inner (BFGS) step is

$$\mu^{(k,l+1)} := \mu^{(k,l)}(j) := \mathbb{P}(\mu^{(k,l)} + \kappa^j d^{(k,l)}),$$

where

$$\mathbb{P}(\mu)_i := \begin{cases} (\mu_a)_i, (\mu)_i \leq (\mu_a)_i, \\ (\mu)_i, (\mu_a)_i \leq (\mu)_i \leq (\mu_b)_i, \\ (\mu_b)_i, (\mu_b)_i \leq (\mu)_i \end{cases}.$$

Finally,

$$\mu^{(k+1)} := \mu^{(k,L)}.$$

Overview on Convergence

Remarks on the Proof

- $\mathcal{P}$ is compact.
- In [Kei+21], the proof is very short; in [Qia+17] more explicit versions can be found.

Overview on Convergence

Remarks on the Proof

- $\mathcal{P}$ is compact.
- In [Kei+21], the proof is very short; in [Qia+17] more explicit versions can be found.
- Proper **convergence is not shown**, just that "all accumulation points of the sequence of parameters are approximate first order critical points"!

Convergence Arguments

General Idea, c.f. [Qia+17]

For the TR method to converge we require that

1. the error of $m^{(k)}$ can be bounded over all of $\mathcal{P}$,
2. at any $\mu \in \mathcal{P}$ we can reduce the approximation error of $m^{(k)}$ to some small tolerance $\varepsilon > 0$, and
3. $m^{(k)}$ is smooth with a finite gradient everywhere.

Convergence Arguments

Specific Formulations, c.f. [Qia+17]

For our problem we get

- $|\mathcal{J}(u_\mu, \mu) - \mathcal{J}_N(\mu)| \leq \Delta_{\mathcal{J}_N}(\mu)$,
- $\|\nabla_\mu \mathcal{J}(u_\mu, \mu) - \nabla_\mu \mathcal{J}_N(\mu)\|_2 \leq \Delta_{\nabla \mathcal{J}_N}(\mu)$, and
- $\mathcal{J}_N^{(k+1)}(\mu^{(k+1)}) \leq \mathcal{J}_N^{(k)}(\mu^{(k,0)})$.

Convergence Arguments

Specific Formulations, c.f. [Qia+17]

For our problem we get

- $|\mathcal{J}(u_\mu, \mu) - \mathcal{J}_N(\mu)| \leq \Delta_{\mathcal{J}_N}(\mu)$,
- $\|\nabla_\mu \mathcal{J}(u_\mu, \mu) - \nabla_\mu \mathcal{J}_N(\mu)\|_2 \leq \Delta_{\nabla \mathcal{J}_N}(\mu)$, and
- $\mathcal{J}_N^{(k+1)}(\mu^{(k+1)}) \leq \mathcal{J}_N^{(k)}(\mu^{(k,0)})$.

The last is equivalent to

$$\mathcal{J}_N^{(k)}(\mu^{(k+1)}) + \Delta_{\mathcal{J}_N^{(k)}}(\mu^{(k+1)}) < \mathcal{J}_N^{(k)}(\mu^{(k)}).$$

Convergence Arguments

Assumptions

- $\mathcal{J}(u, \mu)$ is strictly positive.
- For every TR iteration k we can find a trust-radius large enough for which the decrease condition is satisfied.

Convergence Arguments

Lemma 1, c.f. [Kei+21, Lemma 4.4]

The line search for a parameter in the trust-region takes finitely many iterations to complete.

Convergence Arguments

Lemma 1, c.f. [Kei+21, Lemma 4.4]

The line search for a parameter in the trust-region takes finitely many iterations to complete.

We want to find individual indices j_1, j_2 satisfying the individual TR conditions.
For the first index, this follows from a result in [Kel99].
The second index can iteratively be found from the iteration $k = 0$ relying on the Lipschitz continuity of the reduced cost functional and its gradient w.r.t. μ. The iteration then follows from the fact that after enrichment the same argument as for $k = 0$ applies, and that we only increase the index after enrichment of the ROM space.

Convergence Arguments

Lemma 2, c.f. [Kei+21, Theorem 4.5]

Every accumulation point of $\mu^{(k)}$ is an approximate first order critical points, i.e. for $\tau > 0$

$$\|\mu - \mathbb{P}(\mu - \nabla_\mu \mathcal{J}(\mu))\|_2 \leq \tau.$$

Convergence Arguments

Lemma 2, c.f. [Kei+21, Theorem 4.5]

Every accumulation point of $\mu^{(k)}$ is an approximate first order critical points, i.e. for $\tau > 0$

$$\|\mu - \mathbb{P}(\mu - \nabla_\mu \mathcal{J}(\mu))\|_2 \leq \tau.$$

$\mathcal{P}$ is compact, whence we can find a convergent subsequence.
After enrichment,

$$q^{(k)}(\mu) := \frac{\Delta_{\mathcal{J}_N^{(k)}}(\mu)}{\mathcal{J}_N^{(k)}(\mu)} = 0,$$

and because V and V_N are finite subspaces, we after at most a finite amount of time have no more approximation error on $\mathcal{P}$. From then on, all points in the convergent subsequence are approximate first order critical points.

Convergence Arguments

Note: This is a convergence argument for a projected newton algorithm in the inner loop!

Convergence Arguments

Note: This is a convergence argument for a projected newton algorithm in the inner loop!

Assumptions, c.f. [Ban+20]

- $\mathcal{J}$ is strictly positive.
- For every TR iteration k we can find a trust-radius large enough for which the decrease condition is satisfied.

Convergence Arguments

Note: This is a convergence argument for a projected newton algorithm in the inner loop!

General Argument

1. We can find a bound on the iterated ROM gradient.
2. We can find a bound on $q^{(k)}$ satisfying the TR conditions.
3. The parameters produced by the inner iteration satisfy the TR conditions.
4. Every accumulation point of the outer parameter sequence is an approximate first order critical point.

Numerical Results, c.f. [Qia+17]

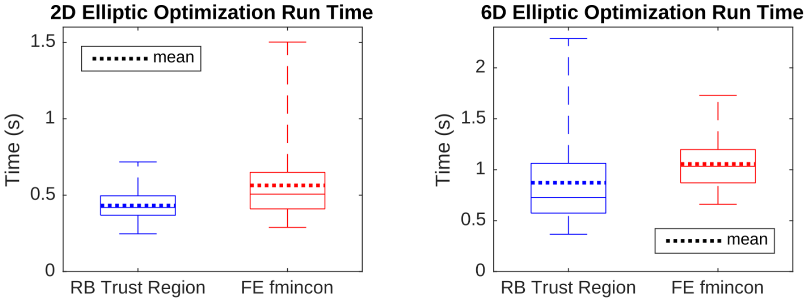


FIG. 2. *Run time comparison for optimizations constrained by elliptic PDEs. In contrast, the traditional offline-online RB approach for a 2D (6D) optimization runs in* 0.04 (0.10) *seconds online, but requires* 1.6 (4800) *seconds offline (on average).*

Numerical Results, c.f. [Qia+17]

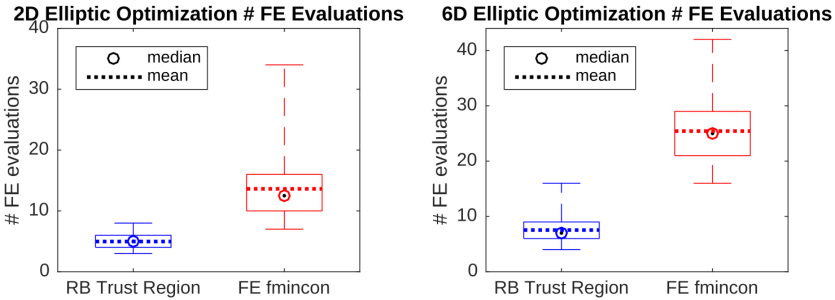
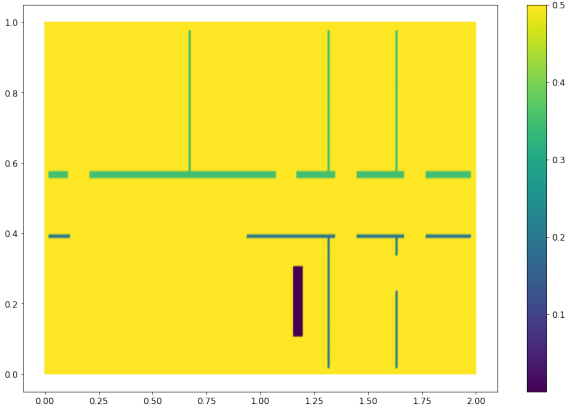
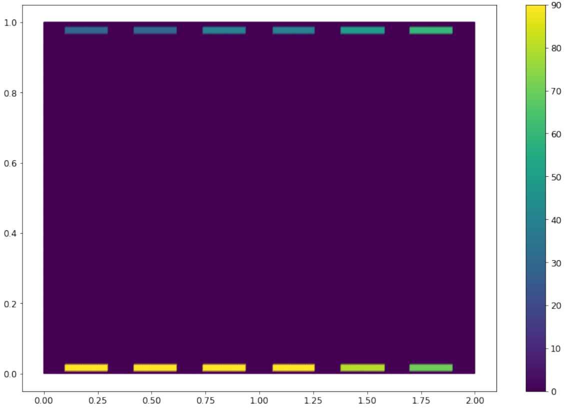


FIG. 3. *Number of full model evaluations required for optimizations constrained by elliptic PDEs. The traditional offline-online RB approach requires* 0 *full evaluations online and an average of* 9 (48) *full evaluations in the* 2*D* (6*D*) *case offline.*

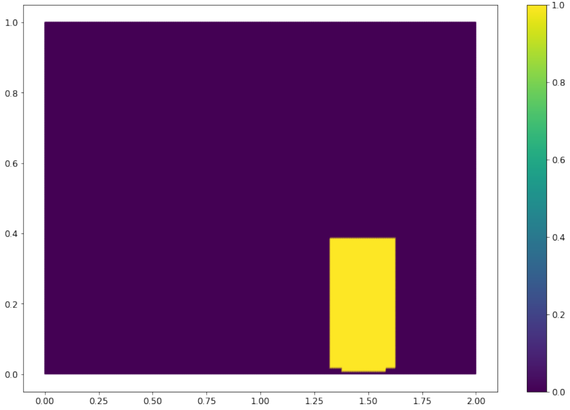
Numerical Results, c.f. github.com/TiKeil



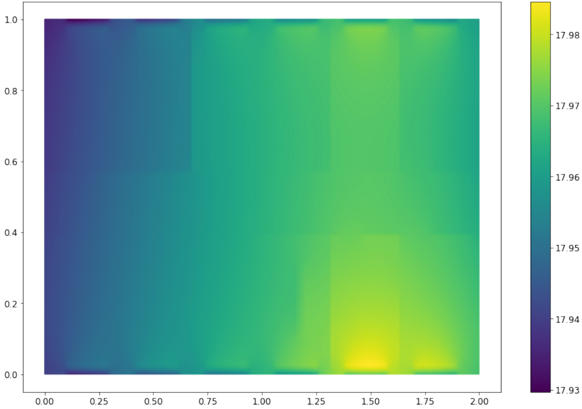
Numerical Results, c.f. github.com/TiKeil



Numerical Results, c.f. github.com/TiKeil



Numerical Results, c.f. github.com/TiKeil



Numerical Results, c.f. [Ban+20]

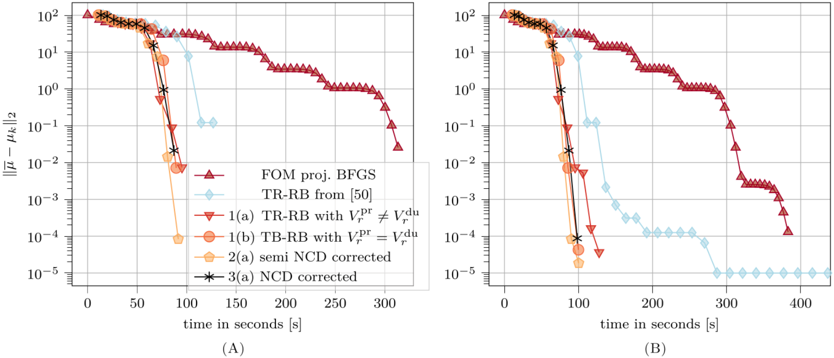


FIGURE 5. Error decay and performance of selected algorithms for two choices of τ_{FOC} for the example from Section 5.3.2 for a single optimization run with random initial guess, compare Figure 2. (A) Result for $\tau_{\mathrm{FOC}} = 5 \cdot 10^{-4}$. (B) Result for $\tau_{\mathrm{FOC}} = 10^{-6}$.

Numerical Results, c.f. [Ban+20]

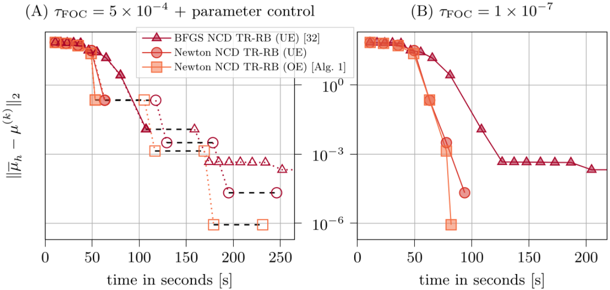


FIGURE 2. Error decay and performance of selected algorithms defined in Section 4.1 for experiment 1 from Section 4.2 with unconditional enrichment (UE) vs. optional enrichment (OE) for a single optimization run with random initial guess $\mu^{(0)}$ for two choices of τ_{FOC} (solid lines) with optional intermediate parameter control according to (2.15) (dotted lines): for each algorithm each marker corresponds to one (outer) iteration of the optimization method and indicates the absolute error in the current parameter, measured against the computed FOM optimum. The dashed black horizontal lines indicate the time taken for the post-processing of the parameter control.

[Ban+20] Stefan Banholzer, Tim Keil, Luca Mechelli, Mario Ohlberger, Felix Schindler, and Stefan Volkwein. "An adaptive projected Newton non-conforming dual approach for trust-region reduced basis approximation of PDE-constrained parameter optimization". In: *arXiv* (Dec. 2020).

[Hin+09] Michael Hinze, Rene Pinnau, Michael Ulbrich, and Stefan Ulbrich. *Optimization with PDE Constraints*. Springer Netherlands, 2009. DOI: 10.1007/978-1-4020-8839-1.

[Kei+21] Tim Keil, Luca Mechelli, Mario Ohlberger, Felix Schindler, and Stefan Volkwein. "A non-conforming dual approach for adaptive Trust-Region reduced basis approximation of PDE-constrained parameter optimization". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 55 (3 May 2021), pp. 1239–1269. ISSN: 28047214. DOI: 10.1051/m2an/2021019.

[Kel99] C. T. Kelley. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics, Jan. 1999. DOI: 10.1137/1.9781611970920.

[NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer New York, 2006. ISBN: 978-0-387-30303-1. DOI: 10.1007/978-0-387-40065-5.

[OR15] Mario Ohlberger and Stephan Rave. "Reduced Basis Methods: Success, Limitations and Future Challenges". In: *arXiv* (Nov. 2015).

[Qia+17] Elizabeth Qian, Martin Grepl, Karen Veroy, and Karen Willcox. "A Certified Trust Region Reduced Basis Approach to PDE-Constrained Optimization". In: *SIAM Journal on Scientific Computing* 39 (5 Jan. 2017), S434–S460. ISSN: 1064-8275. DOI: 10.1137/16m1081981.

Discussion

Thank you for your attention!

Questions?