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

Peter Oehme

24 June 2022

Abstract

Reduced basis methods are a relatively new approach to make multi-query scenarios for elliptic parametric partial differential equations (PDEs) with a large amount of parameters feasible. Parameter optimization motivated by practical examples then motivates the desire to optimize such systems w.r.t. the underlying parameter. In 2017 a certified trust-region method was proposed to tackle this specific problem, and in 2021 the employment of an NCD-corrected approach yielded even better results. The objective of this handout is to establish very basic foundations in the individual components, and try to explain how they come together in the adaptive trust-region reduced basis algorithm. This handout and the slides can be found at github.com/peoe/sem-nc-dual-trrb-pdeopt.

1 Overview

1.1 Problem setting

The method under investigation in [11] considers the following problem: Given a rectangular parameter domain

$$\mathcal{P} := \{\mu \in \mathbb{R}^P \mid \mu_a \leq \mu \leq \mu_b\} \subseteq \mathbb{R}^P,$$

where the inequalities are to be considered componentwise, we consider for a real-valued Hilbert space V the optimization problem

$$\min_{(u,\mu)\in V\times\mathcal{P}} \mathcal{J}(u,\mu) \quad s.t. \quad a_\mu(u,v) = l_\mu(v) \quad \forall v \in V. \tag{1}$$

Here a_μ is a continuous, coercive, bilinear form, and l_μ is a continuous, linear functional; and $\mathcal{J}$ is a quadratic cost functional defined by the continuous, symmetric, bilinear form k_μ, a continuous, linear functional j_μ, and some parameter function Θ through

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

The constraint $a_\mu(u,v) = l_\mu(v)\ \forall v \in V$ encodes that u must be a solution of the underlying PDE in variatonal form.

To tackle this problem in a repeatable manner, an adaptive trust-region method is employed on a reduced basis model working with a non-conforming dual (NCD) basis approach. The advantages of this procedure are:

- higher accuracy of approximations in contrast to conforming methods, and
- increased performance when compared to other model reduction approaches.

1.2 Structure

In the Section 2 we introduce the individual parts required to understand the contents of the paper [11]. We introduce the theory of reduced basis methods from a background in finite elements, distinguishing Gâteaux and Fréchet differentiability to state the main objective of PDE constrained optimization, and motivate the two main optimization frame works used to implement the final algorithm. Afterwards, Section 3 highlights the differences between the standard approach to optimality systems in reduced order modelling and the NCD-corrected approach, concluding with an overview of the *a posteriori* error analysis necessary to show a certified approximation property. The adaptive trust-region reduced basis algorithm researched by [15, 11] is lastly stated in Section 4 with a short addendum of convergence proofs. The proofs of two selected error estimates can be found in Appendix A.

2 Preliminaries

2.1 Reduced Basis Method

This section introduces the reduced basis method (RBM) motivated by a consideration of parametrized PDEs with a large number of parameters. This might be the case in optimization, parameter identification or other similar multi-query scenarios where the repeated computation might otherwise be too expensive. A quick overview of RBM the reduced basis method will be given here, while its advantages, limitations, and challenges can be found in works such as [14], [16] or [9].

We start off with a generic description of the finite element method. The central problem for this method is to find some $u \in V$ such that the following equation may be satisfied

$$a_\mu(u, v) = l_\mu(v) \qquad \forall v \in V, \tag{2}$$

where a_μ is a continuous, coercive, bilinear form and l_μ is a continuous linear functional on some function space V. In general, V is supposed to be a high dimensional function space used to obtain a model with negligible error when compared to the true solution. This model is called the full order model (FOM). We note here that the resulting system of equations is far too expensive to compute for mutliple parameter values in sequence, motivating the introduction of RBM.

The first step is to only consider a certain N-dimensional subspace $V_N \subseteq V$ of the FOM space, where $N << \dim(V)$. As a result the problem we want to solve is to find some $u_N \in V_N$ such that

$$a_\mu(u_N, v) = l_\mu(v) \qquad \forall v \in V_N, \tag{3}$$

is satisfied. Thus far, the complexity to solve this problem for multiple parameters has been reduced, however we can do even better. The key assumption to achieve this is the so-called **parameter-separability**. This means that we can decompose a_μ and l_μ in the following manner

$$a_\mu(u, v) = \sum_{p=1}^{p_a} \theta_p^a(\mu)\, a_p(u, v), \qquad l_\mu(v) = \sum_{p=1}^{p_l} \theta_p^l(\mu)\, l_p(v), \tag{4}$$

where the θ are parameter functions for a_μ and l_μ respectively, the p are the number of parameters for a_μ and l_μ respectively, and the a_p and l_p are the parameter independent components for a_μ and l_μ respectively.

With this assumption in hand, we can introduce the idea which enables RBM to repeatedly compute the solution for different parameters: the **offline/online decomposition**. The principal idea is to first assemble the parameter independent system of equations and later only compute the parameter dependent parts for each parameter. In this procedure, the computationally complex assembly is done only once (**offline phase**), while the computationally faster addition of parameters can be cheaply repeated multiple times (**online phase**). We thus assemble

$$\mathbb{A}_m^{i,j} := a_m(\varphi_j, \varphi_i), \qquad \mathbb{L}_n^i := l_n(\varphi_i), \qquad \text{(offline)}$$

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

for some basis $\{\varphi_i \mid 1 \leq i \leq N\}$ of the space V_N, $1 \leq m \leq p_a$, and $1 \leq n \leq p_l$. We can afterwards obtain our solution by solving the system

$$\mathbb{A}(\mu)\, \underline{u}_N(\mu) = \mathbb{L}(\mu), \qquad u_N(\mu) = \sum_{i=1}^{N} \underline{u}_N(\mu)\, \varphi_i.$$

Overall this approach via offline/online decomposition requires a complexity of $\mathcal{O}(N^2\, p_a) + \mathcal{O}(N\, p_l) + \mathcal{O}(N^3) + \mathcal{O}(N)$, where the first two terms amount to the offline assembly, the third summand originates from solving the assembled system, and the linear term is due to the reassembly of the solution from the solution vector.

One essential question in RBM is the construction of the reduced space V_N. For this there are multiple methods such as:

- Greedy algorithm: construction of the reduced basis by choosing the parameter solution with the largest *a posteriori* error, cf. [5, 18, 8],
- Proper orthogonal decomposition: construction of the reduced basis by left singular values, cf. [13, 7, 8], and
- Discrete empirical interpolation: construction of the reduced basis by finding a unisolvent set for an interpolation operator, cf. [2, 3, 4, 6, 8].

2.2 PDE Constrained Optimization

We now want to introduce the idea of optimization under PDE constraints. This type of optimization is similar to other kinds of optimization in that we can derive certain optimality conditions reliant on the gradient and Hessian of some function or functional. When working functionals it is required that we take care in describing how the derivatives of these functionals are defined and calculated. Hence we first give the definitions of **Gâteaux** and **Fréchet differentiability** in accordance with [10, Section 1.4].

A functional $F : X \to Y$ is called **Gâteaux differentiable** at $x \in X$ if the directional derivative $dF(x) : X \to Y, h \mapsto dF(x)[h]$ is a bounded and linear functional, that is $dF(x) \in \mathcal{L}(X, Y)$, where the directional derivative of F at x is defined by

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

If furthermore the approximation

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

holds for $\|h\|_X \to 0$, we say that F is **Fréchet differentiable** at x. The usual generalization to say that F is Gâteaux/Fréchet differentiable if it is Gâteaux/Fréchet differentiable at every $x \in X$ also applies here.

For our use case we often consider parametrized functionals, for which we get the chain rule

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

where we generally denote the derivative w.r.t. to the parameter with ∂_μ and the derivative w.r.t. the respective first and second argument with ∂_u, and ∂_v.

For the scope of this seminar, we shall consider problems of the form

$$\min_{u \in V, \mu \in \mathcal{P}} \mathcal{J}(u, \mu) \quad \text{s.t.} \quad e(u, \mu) = 0, \tag{5}$$

where we consider V to be some function space, and $\mathcal{P}$ to be the underlying parameter space. In accordance with [10, Subsection 1.1.1 and Equation 1.4, Section 1.6 and Equation 1.81], we call the function $\mathcal{J}$ the **objective function** and

$$e(u, \mu) = 0 \tag{6}$$

the **state equation**.

We can then derive a system of optimality condition for (5), similarly to [10, Section 1.7]. However, this will be covered later on in direct application to our general problem setting.

2.3 Trust-Region Method and BFGS

For the final component of this seminar we take a look at the trust-region method. The benefit of this method is that we can iteratively perform local optimizations for a simpler function than our initial objective function. The general procedure according to [12, Section 3.3] is to approximate the objective function by a model function m_k in a region where we can trust that this approximation is of sufficient quality — the so-called **trust-region** (TR).

Any such trust-region can be described by $\Delta_k := \{x \in X \mid \|x - x_k\| \leq \delta_k\}$, where x_k is the sequence of points obtained from our iterative optimization procedure and δ_k is the radius of the current trust-region, the so-called **trust-radius**. We then optimize over the trust-region and compute the next x_k as follows:

$$s_k := \arg\min_{\|s\| \leq \delta_k} m_k(x_k + s) \qquad x_{k+1} := x_k + s_k. \tag{7}$$

Afterwards, we can decide whether or not to accept this step subject to conditions depending on the individual problem such as accuracy of the underlying model function or satisfaction of other previously given constraints. Depending on the decision taken here, we are left with two options:

- if the step is rejected we may decrease the trust-radius and repeat the optimization, and

- if the step is accepted and a certain threshold reached, we may increase the trust-radius and proceed with the next trust-region.

In each iteration of the TR method we have to solve (7) for a certain trust-region. For this matter we involve the BFGS method (Broyden, Fletcher, Goldfarb, Shanno). This is a quasi-Newton method where we perform a line search with some approximation of the Hessian for the Newton steps, cf. [12, Chapter 4]. The iterations in [11] then appear as follows:

$$\mu_{k+1}(j) := \mathbb{P}_{\mathcal{P}}\left(\mu_k + \kappa^j\, d_k\right),$$

where $d_k = -\mathcal{H}_c(\mathcal{J})^{-1}\nabla_\mu \mathcal{J}(\mu_k)$ denotes the descent direction, $\kappa \in (0,1)$ is a step size, and $\mathbb{P}_{\mathcal{P}}$ is the projection operator which ensures that our iterator remains within our constraint domain. The descent direction here is determined as outlined in [12, Section 5.5.3]. We then terminate this iteration at some point to compute the next trust-region.

3 ROM Optimality Systems

In the following, we want to consider optimality conditions for the reduced problem and from there deduce error estimates for the reduced problem. This will involve primal and dual spaces for the conditions, causing the easily computable gradients that one would usually use to derive error estimates to differ from the exact gradients. Only a short overview of the ideas mentioned will be given here, but a longer description can be found in [11, Section 3]. We then move on to state two different approaches to estimating the true gradients in the reduced problem. Combining all this, we finally establish error estimates on the inexact functionals and their gradients.

If we consider (6) for our FOM and ROM problems (2) and (3), we can see that the state equation can be restated through

$$e(u_\mu, \mu) = l_\mu(v) - a_\mu(u_\mu, v) \quad \forall v \in V \text{ or } V_N.$$

We now define the **primal residual** by $Res_\mu^{pr}(u)[v] := l_\mu(v) - a_\mu(u, v)$. Following [10, Corollary 1.3], we get the optimality conditions in [11, Proposition 2.9]. We derive from these the following dual relation

$$a_\mu(v, p_\mu) = \partial_u \mathcal{J}(u_\mu, \mu)[v] = j_\mu(v) + 2k_\mu(v, u_\mu) \quad \forall v \in V \text{ or } V_N,$$

and finally define the **dual residual** in terms of the difference in the dual optimality condition $Res_\mu^{du}(u_\mu, p_\mu)[v] := j_\mu(v) + 2k_\mu(v, u_\mu) - a_\mu(v, p_\mu)$. By the same optimality conditions, all locally optimal points of (5) have to satisfy

$$Res_\mu^{pr}(u_\mu)[v] = 0 \qquad \forall v \in V \text{ or } V_N^{pr}, \tag{8}$$

and

$$Res_\mu^{du}(u_\mu, p_\mu)[v] = 0 \qquad \forall v \in V \text{ or } V_N^{du}. \tag{9}$$

As usual in optimization we also get the **Lagrangian**

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

In this and the following sections, $u_\mu \in V$ will always denote the primal solution u for some parameter μ, and similarly $p_\mu \in V$ the dual solution. If necessary, we will denote the ROM solutions by $u_{N,\mu} \in V_N^{pr}$ ($p_{N,\mu} \in V_N^{du}$ respectively), while we will avoid additional indices for the FOM solutions. We note here that as mentioned in [11, Subsection 3.2] the primal and the dual reduced spaces are in general not the same such that even the dimensions may be assumed to differ. This then gives rise to the notion of a **non-conforming dual** (NCD) approach.

3.1 Standard Approach for a ROM Optimality System

We now introduce a first method to obtain an inexact gradient as proposed in [15] and specifically the variant described in [11, Subsection 3.2]. This method is practical because it enables the approximation of the exact gradient for the reduced solutions. One particuliarity of this idea as mentioned by [15] is that we can afterwards disregard the usual offline/online paradigm during the optimization and thus adapt the reduced approximation on the fly.

Hence, for the **reduced functional** $J_N(\mu) := \mathcal{J}(u_{N,\mu}, \mu)$, when considering the gradient (or rather its componentwise definition) we get

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

and defining the **inexact gradient** we have

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

where in the exact gradient we considered the Lagrangian for the reduced version of (1) and apply $J_N(\mu) = \mathcal{L}(u_{N,\mu}, \mu, p)\ \forall p \in V_N^{du}$. If we then consider $p = p_{N,\mu}$ in the first line and combine them with the second line we obtain

$$\big(\nabla_\mu J_N(\mu)\big)_i = \underbrace{\partial_u \mathcal{L}(u_{N,\mu}, \mu, p_{N,\mu})[d_{\mu_i} u_{N,\mu}]}_{(*)} + \big(\tilde{\nabla}_\mu J_N(\mu)\big)_i.$$

The key insight here is that in general $(*) \neq 0$ because we are not actually considering all elements from V for (9), but just those from the reduced dual space.

3.2 NCD-corrected Approach for a ROM Optimality System

To extend on the result of the previous section, the **NCD-corrected reduced functional** in [11, Subsection 3.3] contains one more step to attain improved error estimates. Specifically, this involves the restriction of the Lagrangian in the dual variable. We get

$$\mathcal{J}_N(\mu) := \mathcal{L}(u_{N,\mu}, \mu, p_{N,\mu}) = J_N(\mu) + Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}]. \tag{10}$$

This is the functional we will be dealing with forthwith, and we define the reduced optimization problem

$$\min_{\mu \in \mathcal{P}} \mathcal{J}_N(\mu). \tag{11}$$

The key insight for the coming error estimates is that the following statement (cf. [11, Proposition 3.3]) about the gradient of (10) holds true:

Proposition 3.1. *We have*

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

where $z_\mu \in V_N^{du}$ and $w_\mu \in V_N^{pr}$ respectively denote the solutions to the following problems:

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

Besides this classical approach we can also try and compute the gradient of the NCD-corrected functional (10) by the sensitivities of the primal and dual reduced solutions (cf. [11, Proposition 3.9]).

Proposition 3.2. *We have*

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

Here, **sensitivities** mean the derivatives of these solutions w.r.t. the parameters:

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

This procedure is more efficient then the one from Proposition 3.1 because the solutions to both equations of Proposition 3.2 can be reused for every component e_i whereas the ones of the former have to be solved every time.

Another catching point is that the reduced sensitivities in V_N^{pr} or V_N^{du} must not be good approximations of the original approximations in V. To accompany these generally high dimensional sensitivities in the reduced spaces, we can take one of two approaches:

- include the sensitivities in the corresponding reduced spaces, or
- distribute the directional sensitivities to problem adapted reduced spaces.

Either way, we obtain $V_N^{pr,d_{\mu_i}}, V_N^{du,d_{\mu_i}} \subseteq V$. In analogy to the derivatives in Proposition 3.2 we define the **approximate partial derivatives** as the solutions to

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

From this, the definition of $\tilde{d}_\nu u_{N,\mu}$, and $\tilde{d}_\nu p_{N,\mu}$ follow in the canonical way. We also want to emphasize that both sensitivities coincide if $V_N^{pr} = V_N^{pr,d_{\mu_i}}$, and $V_N^{du} = V_N^{du,d_{\mu_i}}$ respectively. The **inexact gradient of the NCD-corrected functional** for the approximate sensitivities is defined componentwise as

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

To conclude we define the residuals on V, which we will need later to perform the error analysis, for some $v \in V$ by

$$\begin{aligned} Res_\mu^{pr,d_{\mu_i}}(u_\mu, d_{\mu_i}u_\mu)[v] &:= \partial_{\mu_i} Res_\mu^{pr}(u_\mu)[v] - a_\mu(d_{\mu_i}u_\mu, v), \text{ and} \\ Res_\mu^{du,d_{\mu_i}}(u_\mu, p_\mu, d_{\mu_i}u_\mu, d_{\mu_i}p_\mu)[v] &:= \partial_{\mu_i} Res_\mu^{du}(u_\mu,p_\mu)[v] \\ & \quad + 2k_\mu(v, d_{\mu_i}u_\mu) - a_\mu(v, d_{\mu_i}p_\mu). \end{aligned}$$

3.3 *A posteriori* error analysis

The previous results in this section now enable us to take a look at *a posteriori* error estimators. Most of these involve comparisons of the ROM solution to the FOM solution, so we want to reiterate the notation in that u_μ will always denote the primal solution u for some parameter μ of the FOM, and similarly p_μ the dual solution of the FOM, and $u_{N,\mu}$, $p_{N,\mu}$ the respective solutions of the ROM. This part will then highlight two different approaches for obtaining the error estimates: the standard approach of RBM and the sensitivities approach. As [11] does, we also want to emphasize that most of these results can only be computed efficiently by means of the assumption that the problems are offline/online decomposable.

To begin with we have the usual estimates we would expect from coercive problems. Proofs may be found in [17, 15].

$$\begin{aligned}
&\|u_\mu - u_{N,\mu}\| \leq \Delta_{pr}(\mu) := {\alpha_\mu}^{-1} \left\|Res^{pr}_\mu(u_{N,\mu})\right\|, \text{ and}\\
&\|p_\mu - p_{N,\mu}\| \leq \Delta_{du}(\mu) := {\alpha_\mu}^{-1}\big(2\gamma_{k_\mu}\Delta_{pr}(\mu) + \left\|Res^{du}_\mu(u_{N,\mu}, p_{N,\mu})\right\|\big),
\end{aligned}$$

where α_μ denotes the coercivity constant of a_μ, γ_{k_μ} stands for the modulus of continuity of k_μ and the norms of the residuals are to be understood as the norms of the respective linear functionals they represent. Similar notations shall be used henceforth.

In a similar fashion, we can obtain upper bounds on the errors between the exact and reduced functionals. The first is a result for the simple inexact functional courtesy of [15, Theorem 4], while the second is the analogous statement for the NCD-corrected functional as seen in [11, Proposition 3.6]. We have

$$\begin{aligned}
&|\mathcal{J}(u_\mu, \mu) - J_N(\mu)| \leq \Delta_{J_N}(\mu), \text{ and}\\
&|\mathcal{J}(u_\mu, \mu) - \mathcal{J}_N(\mu)| \leq \Delta_{\mathcal{J}_N}(\mu),
\end{aligned}$$

where we define the respective Δs as follows

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

As mentioned by [11], the second inequality is an improved version of the non-coorected definition avoiding lower order terms.

Continuing to gradients, we get a statement about the inexact and the NCD-corrected reduced gradients in accordance with [11, Proposition 3.8].

$$\begin{aligned}
&\left\|\nabla\mathcal{J}(u_\mu, \mu) - \tilde{\nabla} J_N(\mu)\right\|_2 \leq \left\|\Delta_{\tilde{\nabla} J_N}(\mu)\right\|_2, \text{ and}\\
&\left\|\nabla\mathcal{J}(u_\mu, \mu) - \tilde{\nabla} \mathcal{J}_N(\mu)\right\|_2 \leq \left\|\Delta^*_{\nabla\mathcal{J}_N(\mu)}\right\|_2,
\end{aligned}$$

where the Δs are defined componentwise as follows

$$\begin{aligned}
\left(\Delta_{\tilde{\nabla} J_N}(\mu)\right)_i &:= 2\Delta_{pr}(\mu)\gamma_{\partial_{\mu_i} k_\mu} \|u_{N,\mu}\| + \Delta_{pr}(\mu)\left(\gamma_{\partial_{\mu_i} j_\mu} + \gamma_{\partial_{\mu_i} a_\mu} \|p_{N,\mu}\|\right) \\
&\quad + \Delta_{du}(\mu)\left(\gamma_{\partial_{\mu_i} l_\mu} + \gamma_{\partial_{\mu_i} a_\mu} \|p_{N,\mu}\|\right) + \Delta_{pr}(\mu)\Delta_{du}(\mu)\gamma_{\partial_{\mu_i} a_\mu} \\
&\quad + \Delta_{pr}(\mu)^2 \gamma_{\partial_{\mu_i} k_\mu}, \text{ and} \\
\left(\Delta^*_{\tilde{\nabla} \mathcal{J}_N}(\mu)\right)_i &:= 2\Delta_{pr}(\mu)\gamma_{\partial_{\mu_i} k_\mu} \|u_{N,\mu}\| + \Delta_{pr}(\mu)\left(\gamma_{\partial_{\mu_i} l_\mu} + \gamma_{\partial_{\mu_i} a_\mu} \|p_{N,\mu}\|\right) \\
&\quad + \Delta_{du}(\mu)\left(\gamma_{\partial_{\mu_i} l_\mu} + \gamma_{\partial_{\mu_i} a_\mu} \|p_{N,\mu}\|\right) + \Delta_{pr}(\mu)\Delta_{du}(\mu)\gamma_{\partial_{\mu_i} a_\mu} \\
&\quad + {\alpha_\mu}^{-1}\left(\gamma_{\partial_{\mu_i} l_\mu} + \gamma_{\partial_{\mu_i} a_\mu} \|u_{N,\mu}\|\right)\left\|Res^{du}_\mu(u_{N,\mu}, p_{N,\mu})\right\| \\
&\quad + {\alpha_\mu}^{-1}\left\|Res^{pr}_\mu(u_{N,\mu})\right\|\left(\gamma_{\partial_{\mu_i} j_\mu} + 2\gamma_{\partial_{\mu_i} k_\mu}\|u_{N,\mu}\| + \gamma_{\partial_{\mu_i} a_\mu}\|p_{N,\mu}\|\right) \\
&\quad + 2\gamma_{k_\mu}{\alpha_\mu}^{-2}\left(\gamma_{\partial_{\mu_i} l_\mu} + \gamma_{\partial_{\mu_i} a_\mu}\|u_{N,\mu}\|\right)\left\|Res^{pr}_\mu(u_{N,\mu})\right\| \\
&\quad + \Delta_{pr}(\mu)^2\gamma_{\partial_{\mu_i} k_\mu}.
\end{aligned}$$

Reiterating the conclusion of [11] w.r.t. this estimate the second inequality in general does not provide a better estimate than the simpler inexact reduced gradient. Hence, we conclude the classical approach and look onward to the computation by means of sensitivities.

For a first sensitivity result, we take a look at the error in the primal and dual sensitivities. The next statement follows from the arguments presented in [11, Proposition 3.12, and Proposition 3.13].

Proposition 3.3. *We consider the FOM sensitivities $d_{\mu_i} u_\mu, d_{\mu_i} p_\mu \in V$ and the reduced sensitivities $d_{\mu_i} u_{N,\mu} \in V_N^{pr,d_{\mu_i}}, d_{\mu_i} p_{N,\mu} \in V_N^{du,d_{\mu_i}}$. Then we get*

$$\begin{aligned}
\|d_{\mu_i} u_\mu - d_{\mu_i} u_{N,\mu}\| &\leq \Delta_{d_{\mu_i} pr}(\mu) \\
&:= {\alpha_\mu}^{-1}\left(\gamma_{\partial_{\mu_i} a_\mu}\Delta_{pr}(\mu) + \left\|Res^{pr,d_{\mu_i}}_\mu(u_{N,d_{\mu_i} u_{N,\mu}})\right\|\right),
\end{aligned}$$

and

$$\begin{aligned}
\|d_{\mu_i} p_\mu - d_{\mu_i} p_{N,\mu}\| &\leq \Delta_{d_{\mu_i} du}(\mu) \\
&:= {\alpha_\mu}^{-1}\left(2\gamma_{\partial_{\mu_i} k_\mu}\Delta_{pr}(\mu) + \gamma_{\partial_{\mu_i} a_\mu}\Delta_{du}(\mu) + 2\gamma_{k_\mu}\Delta_{d_{\mu_i} pr}(\mu)\right) \\
&\quad + {\alpha_\mu}^{-1}\left\|Res^{du,d_{\mu_i}}_\mu(u_{N,\mu}, p_{N,\mu}, d_{\mu_i} u_{N,\mu}, d_{\mu_i} p_{N,\mu})\right\|.
\end{aligned}$$

This proposition also holds true in the analogous form for the approximate sensitivities $\tilde{d}_{\mu_i} u_{N,\mu}$, and $\tilde{d}_{\mu_i} p_{N,\mu}$. We then label the corresponding estimators by $\Delta_{\tilde{d}_{\mu_i} pr}(\mu)$ and $\Delta_{\tilde{d}_{\mu_i} du}(\mu)$ instead.

Finally, we can now state the improved *a posteriori* gradient error bound for the NCD-corrected functional. This follows [11, Proposition 3.14] closely.

Proposition 3.4. *We have*

$$\|\nabla_\mu \mathcal{J}(u_\mu, \mu) - \nabla_\mu \mathcal{J}_N(\mu)\|_2 \leq \|\Delta_{\nabla \mathcal{J}_N}(\mu)\|_2,$$

where the vector is defined by the components

$$\begin{aligned}
\left(\Delta_{\nabla \mathcal{J}_N}(\mu)\right)_i &:= \gamma_{\partial_{\mu_i} k_\mu}\Delta_{pr}(\mu)^2 + \gamma_{a_\mu}\Delta_{d_{\mu_i} pr}(\mu)\Delta_{du}(\mu) \\
&\quad + \Delta_{pr}(\mu)\left\|Res^{du,d_{\mu_i}}_\mu(u_{N,\mu}, p_{N,\mu}, d_{\mu_i} u_{N,\mu}, d_{\mu_i} p_{N,\mu})\right\|.
\end{aligned}$$

The analogous assertion for the inexact quantities and estimators holds true as well.

We conclude error analysis by mentioning the second order decay of these estimators in combination with the necessity to always compute the dual norm of the sensitivity residuals (cf. [11]). We can now focus on applying these estimates in a trust-region method to solve problem (11).

4 Adaptive Enrichment for Trust-Region Reduced Basis Approximation

The objective of this section is to describe the trust-region reduced basis method described in [11, Section 4]. To this end we shall stick to a general description, and only mention the relevant theoretical results for convergence at the end. Throughout we shall choose the model function (cf. Subsection 2.3) to be defined as the sequence indexed by $k \geq 0$ as follows

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

The reduced spaces are assumed to be initialized as $V_N^{pr,0} = \{u_\mu\}, V_N^{du,0} = \{p_\mu\}$, with u_μ and p_μ being the FOM solutions for some initial guess $\mu^{(0)} \in \mathcal{P}$. We can then state the inexact reduced version of (7) as introduced in [15, Equation 51]

$$\min_{s \in \mathcal{P}} \mathcal{J}_N^{(k)}(\mu^{(k)} + s) \quad \text{s.t.} \quad \frac{\Delta_{\mathcal{J}_N^{(k)}}(\mu^{(k)} + s)}{\mathcal{J}_N^{(k)}(\mu^{(k)} + s)} \leq \delta^{(k)}, \tag{12}$$

where the equality constraint $Res^{pr}_{\mu^{(k)}+s}(u_{N,\mu^{(k)}+s})[v] = 0$ is hidden in $\mathcal{J}_N^{(k)}$, and $\delta^{(k)}$ is the radius of the trust-region.

For every step of the trust-region method we then have to solve the local problem via BFGS. We iterate for $l \geq 1$ via

$$\begin{aligned} \mu^{(k,l)}(j^{(k,l)}) &:= \mathbb{P}_{\mathcal{P}}\left(\mu^{(k,l)} + \kappa^{j^{(k,l)}} d^{(k,l)}\right), \\ \mu^{(k,l+1)} &:= \mu^{(k,l)}(j^{(k,l)}) \end{aligned}$$

where $\mathbb{P}_{\mathcal{P}}$ is the projection $(\mathbb{P}_{\mathcal{P}}(\mu))_i := \begin{cases} (\mu_a)_i, (\mu)_i \leq (\mu_a)_i, \\ (\mu)_i, (\mu_a)_i \leq (\mu)_i \leq (\mu_b)_i, \\ (\mu_b)_i, (\mu_b)_i \leq (\mu)_i, \end{cases}$ mapping the parameters into the rectangular bounds of $\mathcal{P}$ in analogy to the projection described in Subsection 2.3. Here the descent direction is $d^{(k,l)}$, and $j^{(k,l)}$ is the smallest integer such that the Armijo condition for the BFGS method in [11, Inequality 4.4] and the trust-region inequality condition from (12) are satisfied (cf. [11, Inequality 4.5]). After the termination condition for the BFGS method has been reached at L iterations, see [11, Inequalities 4.6a and 4.6b], we set $\mu^{(k+1)} := \mu^{(k+1,0)} := \mu^{(k,L)}$ for the next (potential) trust-region iteration.

As explained in [11], the main drawback of the trust-region algorithm is that in the standard version the trust-radius may be significantly shrunk, thus missing out on the rate of convergence for the BFGS method. To fix this, we define the threshold $\eta \in [\frac{3}{4}, 1)$ and the criterion for enlarging the radius as proposed in [11, Inequality 4.7] by

$$\rho^{(k)} := \frac{\mathcal{J}(\mu^{(k)}) - \mathcal{J}(\mu^{(k+1)})}{\mathcal{J}_N^{(k)}(\mu^{(k)}) - \mathcal{J}_N^{(k)}(\mu^{(k+1)})} \geq \eta.$$

With this idea in hand, we can now formulate the adaptive trust-region algorithm in Algorithm 1.

One critical point in this alorithm that we have yet to cover are lines 5 and 13. Here we just mentioned an update to the reduced spaces but so far have not yet specified what this means. Essentially there are two options according to [11]:

- Lagrangian reduced spaces: Construct the reduced space by adding the FOM solutions directly related to the primal and dual spaces, i.e. for some $\mu \in \mathcal{P}$ from the optimization $V_N^{pr,k+1} := V_N^{pr,k} \cup \{u_\mu\}, V_N^{du,k+1} := V_N^{du,k} \cup \{p_\mu\}$.

- Aggregated reduced spaces: Construct the reduced space by adding both results to both spaces, i.e. $V_N^{pr,k+1} = V_N^{du,k+1} := V_N^{pr,k} \cup \{u_\mu, p_\mu\}$.

The latter approach has the side effect that the NCD-corrected reduced and the standard reduced functionals coincide, hence giving up all advantages we might gain from pursuing an NCD-corrected approach.

In addition to the different ways one may construct the reduced spaces, [11, Subsection 4.4] lists a short description of the three varying approaches including the standard approach in [15], a semi NCD-corrected approach, and the "fully" NCD-corrected approach taken in [11]. Distinctions between these methods mostly lie in the realm of order of estimation, being distinguished only by the increased computational complexities necessary for more accurate results.

To close, we shall make a few remarks on convergence of the adaptive trust-region reduced basis method. In general, this is based on a few assumptions such as positivity of the functional $\mathcal{J}$, and the existence of trust-radii sufficiently large but in applications these are no hard constraints, and hence can be assumed satisfied. It can then be shown that every accumulation point of the generated parameter sequence is an approximate first order critical point, cf. [11, Theorem 4.5]. A more robust proof has been given in [1], though the method described there employs a projective Newton method instead of BFGS to attain a faster local rate of convergence.

Algorithm 1: Adaptive TR-RB Algorithm, cf. [11, Algorithm 1]

Data: Initial guess $\mu^{(0)}$, initial trust-radius $\delta^{(0)}$, rate of change and safeguard of the radius $\beta_1, \beta_2 \in (0,1)$, tolerance for enlarging the radius η, tolerances for the BFGS method and the first order critical condition (FOC) $0 < \tau_{BFGS} \leq \tau_{FOC} \ll 1$

1 Set $k := 0, loop\_flag := true$;
2 **while** $loop\_flag$ **do**
3   Compute $\mu^{(k+1)}$ by the BFGS method;
4   **if** $\mathcal{J}_N^{(k)}(\mu^{(k+1)}) + \Delta_{\mathcal{J}_N^{(k)}}(\mu^{(k+1)}) < \mathcal{J}_N^{(k)}(\mu^{(k)})$ **then**
5   Accept $\mu^{(k+1)}$, update the reduced spaces at $\mu^{(k+1)}$, and compute $\rho^{(k)}$;
6   **if** $\rho^{(k)} \geq \eta$ **then**
7   Enlarge the radius $\delta^{(k+1)} := \beta_1^{-1}\delta^{(k)}$;
8   **else**
9   Keep $\delta^{(k+1)} := \delta^{(k)}$;
10   **else if** $\mathcal{J}_N^{(k)}(\mu^{(k+1)}) + \Delta_{\mathcal{J}_N^{(k)}}(\mu^{(k+1)}) > \mathcal{J}_N^{(k)}(\mu^{(k)})$ **then**
11   Reject $\mu^{(k+1)}$, shrink the radius $\delta^{(k+1)} := \beta_1\delta^{(k)}$, and go to 3;
12   **else**
13   Update the reduced spaces at $\mu^{(k)}$, and compute $\rho^{(k)}$;
14   **if** $\mathcal{J}_N^{(k+1)}(\mu^{(k+1)}) \leq \mathcal{J}_N^{(k)}(\mu^{(k)})$ **then**
15   Accept $\mu^{(k+1)}$;
16   **if** $\rho^{(k)} \geq \eta$ **then**
17   Enlarge the radius;
18   **else**
19   Keep the radius;
20   **else**
21   Reject $\mu^{(k+1)}$, shrink the radius, and go to 3;
22   **if** *FOC is satisfisfied for* τ_{FOC} **then**
23         $loop\_flag := false$;
24     $k := k + 1$;

References

[1] 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).

[2] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T. Patera. “Une méthode d'«intepolation empirique»: Application à la discrétisation efficace par base réduite d'equations aux dériveés partielles”. In: *Comptes Rendus Mathematique* 339 (9 Nov. 2004), pp. 667–672. ISSN: 1631073X. DOI: 10.1016/j.crma.2004.08.006.

[3] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. “Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations”. In: *International Journal for Numerical Methods in Engineering* 86 (2 Apr. 2011), pp. 155–181. ISSN: 00295981. DOI: 10.1002/nme.3050.

[4] Saifon Chaturantabut and Danny C. Sorensen. “Nonlinear model reduction via discrete empirical interpolation”. In: *SIAM Journal on Scientific Computing* 32 (5 Sept. 2010), pp. 2737–2764. ISSN: 10648275. DOI: 10.1137/090766498.

[5] Ronald DeVore, Guergana Petrova, and Przemyslaw Wojtaszczyk. “Greedy Algorithms for Reduced Bases in Banach Spaces”. In: *Constructive Approximation* 37 (3 June 2013), pp. 455–466. ISSN: 01764276. DOI: 10.1007/s00365-013-9186-2.

[6] Martin Drohmann, Bernard Haasdonk, and Mario Ohlberger. “Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation”. In: *SIAM Journal on Scientific Computing* 34 (2 Mar. 2012). ISSN: 10648275. DOI: 10.1137/10081157X.

[7] B Haasdonk, M Ohlberger, and G Rozza. *A Reduced Basis Method for Evolution Schemes with Parameter-Dependent Explicit Operators*. 2008, pp. 145–168.

[8] Bernard Haasdonk. “Convergence rates of the POD—greedy method”. In: *Mathematical Modelling and Numerical Analysis* 47 (3 2013), pp. 859–873. ISSN: 12903841. DOI: 10.1051/m2an/2012045.

[9] Jan S. Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Reduced Basis Methods*. Springer Science and Business Media B.V., 2016, pp. 27–43. DOI: 10.1007/978-3-319-22470-1_3.

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

[11] 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.

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

[13] K. Kunisch and S. Volkwein. "Galerkin proper orthogonal decomposition methods for parabolic problems". In: *Numerische Mathematik* 90 (1 Nov. 2001), pp. 117–148. ISSN: 0029599X. DOI: 10.1007/s002110100282.

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

[15] 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.

[16] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: An introduction*. Springer International Publishing, Aug. 2015, pp. 1–263. ISBN: 9783319154312. DOI: 10.1007/978-3-319-15431-2.

[17] G. Rozza, D. B.P. Huynh, and A. T. Patera. "Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics". In: *Archives of Computational Methods in Engineering* 15 (3 Sept. 2008), pp. 229–275. ISSN: 11343060. DOI: 10.1007/s11831-008-9019-9.

[18] K. Veroy, C. Prud'Homme, D. V. Rovas, and A. T. Patera. "A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations". In: American Institute of Aeronautics and Astronautics Inc., 2003. ISBN: 9781624100864. DOI: 10.2514/6.2003-3847.

A Additional Proofs

Proof. This is the proof of Proposition 3.1. We shall use the shorter notation $d_{\mu_i}\mathcal{J}_N := d_\mu \mathcal{J}_N \cdot \mu_i$.

We compute using the chain rule

$$\begin{aligned} d_{\mu_i}\mathcal{J}_N(\mu) = \partial_{\mu_i}\mathcal{J}(u_\mu,\mu) + \underbrace{\partial_u \mathcal{J}(u_\mu,\mu)[d_{\mu_i}u_\mu]}_{(I)} + \partial_{\mu_i} Res_\mu^{pr}(u_\mu)[p_\mu] \\ + \underbrace{\left(\partial_u Res_\mu^{pr}(u_\mu)[p_\mu]\right)[d_{\mu_i}u_\mu]}_{(II)} + \underbrace{\left(\partial_v Res_\mu^{pr}(u_\mu)[p_\mu]\right)[d_{\mu_i}p_\mu]}_{(III)}. \end{aligned} \tag{13}$$

When considering the individual terms in (13) we get

$$(I) = j_\mu(d_{\mu_i}u_\mu) + 2k_\mu(d_{\mu_i}u_\mu, p_\mu), \tag{14}$$

where we made use of [11, Remark 2.4];

$$(II) = \left(\partial_u(l_\mu(p_\mu) - a_\mu(u_\mu,p_\mu))\right)[d_{\mu_i}u_\mu] = -a_\mu(d_{\mu_i}u_\mu, p_\mu), \tag{15}$$

because $\partial_u l_\mu \equiv 0$, and the above remark once more; and lastly

$$\begin{aligned} (III) &= \left(\partial_v l_\mu(p_\mu) - a_\mu(u_\mu,p_\mu)\right)[d_{\mu_i}p_\mu] \\ &= l_\mu(d_{\mu_i}p_\mu) - a_\mu(u_\mu, d_{\mu_i}p_\mu) = Res_\mu^{pr}(u_\mu)[d_{\mu_i}p_\mu], \end{aligned} \tag{16}$$

having applied the remark a third time. Summing all three terms we get

$$\begin{aligned} (14) + (15) + (16) = {} & Res_\mu^{pr}(u_\mu)[d_{\mu_i}p_\mu] \\ & + j_\mu(d_{\mu_i}u_\mu) + 2k_\mu(d_{\mu_i}u_\mu, u_\mu) - a_\mu(d_{\mu_i}u_\mu, p_\mu). \end{aligned} \tag{17}$$

We then consider the following derivatives

$$\begin{aligned} \partial_{\mu_i} Res_\mu^{pr}(u_\mu)[w_\mu] = a_\mu(d_{\mu_i}u_\mu, w_\mu) &= Res_\mu^{du}(u_\mu,p_\mu)[d_{\mu_i}u_\mu] - 2k_\mu(z_\mu, d_{\mu_i}u_\mu) \\ &= j_\mu(d_{\mu_i}u_\mu) + 2k_\mu(d_{\mu_i}u_\mu, u_\mu) - a_\mu(d_{\mu_i}u_\mu, p_\mu) - 2k_\mu(z_\mu, d_{\mu_i}u_\mu), \end{aligned} \tag{18}$$

where in the first equality we made use of [11, Proposition 2.5], in the second and third we applied [11, Equation 3.11 and 2.5] respectively;

$$\begin{aligned} \partial_{\mu_i} Res_\mu^{du}(u_\mu,p_\mu)[z_\mu] &= a_\mu(z_\mu, d_{\mu_i}p_\mu) - 2k_\mu(z_\mu, d_{\mu_i}u_\mu) \\ &= -Res_\mu^{pr}(u_\mu)[d_{\mu_i}p_\mu] - 2k_\mu(z_\mu, d_{\mu_i}u_\mu), \end{aligned} \tag{19}$$

with [11, Equation 2.6 and 3.10]. Computing (18) − (19), we obtain

$$\begin{aligned} (18) - (19) = {} & Res_\mu^{pr}(u_\mu)[d_{\mu_i}p_\mu] + j_\mu(d_{\mu_i}u_\mu) + 2k_\mu(d_{\mu_i}u_\mu, u_\mu) - a_\mu(d_{\mu_i}u_\mu, p_\mu) \\ & - 2k_\mu(z_\mu, d_{\mu_i}u_\mu) + 2k_\mu(z_\mu, d_{\mu_i}u_\mu). \end{aligned} \tag{20}$$

Finally, we see that (17) = (20) and thus get the assertion. □

Proof. This is the proof of Proposition 3.2. We once again use the notation introduced in the previous proof.

We consider (17) and see that (16) is equal to the primal residual part of the assertion, and that (14) as well as (15) add up to the dual residual part — hence showing the assertion. □

Proof. This is the proof of Proposition 3.4. We once again use the notation of the previous two proofs, and $e_\mu^{pr} := u_\mu - u_{N,\mu}, e_\mu^{du} := p_\mu - p_{N,\mu}$.

We use the fact that by [11, Equations 3.1 and 3.2] a FOM solution u_μ satisfies $Res_\mu^{pr}(u_\mu)[d_{\mu_i} p_{N,\mu}] = 0$ and $Res_\mu^{du}(u_\mu, p_\mu)[d_{\mu_i} u_{N,\mu}] = 0$ to get

$$\begin{aligned}
(\nabla_\mu \mathcal{J}(u_\mu, \mu) - \nabla_\mu \mathcal{J}_N(\mu))_i &= \partial_{\mu_i} j_\mu(e_\mu^{pr}) + \partial_{\mu_i} k_\mu(u_\mu, u_\mu) - \partial_{\mu_i} k_\mu(u_{N,\mu}, u_{N,\mu}) \\
&+ \partial_{\mu_i} Res_\mu^{pr}(u_\mu)[p_\mu] - \partial_{\mu_i} Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}] \\
&- Res_\mu^{pr}(u_{N,\mu})[d_{\mu_i} p_{N,\mu}] + Res_\mu^{pr}(u_\mu)[d_{\mu_i} p_{N,\mu}] \\
&- Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[d_{\mu_i} p_{N,\mu}] + Res_\mu^{du}(u_\mu, p_\mu)[d_{\mu_i} p_{N,\mu}] \\
&= \partial_{\mu_i} j_\mu(e_\mu^{pr}) + \partial_{\mu_i} k_\mu(u_\mu, u_\mu) - \partial_{\mu_i} k_\mu(u_{N,\mu}, u_{N,\mu}) \\
&+ \partial_{\mu_i} Res_\mu^{pr}(u_\mu)[p_\mu] - \partial_{\mu_i} Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}] \\
&+ \underbrace{Res_\mu^{pr}(e_\mu^{pr})[d_{\mu_i} p_{N,\mu}]}_{(I)} + \underbrace{Res_\mu^{du}(e_\mu^{pr}, e_\mu^{du})[d_{\mu_i} p_{N,\mu}]}_{(II)}.
\end{aligned} \tag{21}$$

Applying the definitions of the residuals and adding terms we see that

$$\begin{aligned}
(I) &= \underbrace{l_\mu(d_{\mu_i} p_{N,\mu}) - l_\mu(d_{\mu_i} p_{N,\mu})}_{=0} - a_\mu(e_\mu^{pr}, d_{\mu_i} p_{N,\mu}) \\
&+ \partial_{\mu_i} Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[e_\mu^{pr}] + 2k_\mu(d_{\mu_i} u_{N,\mu}, e_\mu^{pr}) \\
&- \partial_{\mu_i} Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[e_\mu^{pr}] - 2k_\mu(d_{\mu_i} u_{N,\mu}, e_\mu^{pr}) \\
&= Res_\mu^{du,d_{\mu_i}}(u_{N,\mu}, p_{N,\mu}, d_{\mu_i} u_{N,\mu}, d_{\mu_i} p_{N,\mu})[e_\mu^{pr}] \\
&- \partial_{\mu_i} Res_\mu^{du}(u_{N,\mu}, p_{N,\mu})[e_\mu^{pr}] - 2k_\mu(d_{\mu_i} u_{N,\mu}, e_\mu^{pr}) \\
&= Res_\mu^{du,d_{\mu_i}}(u_{N,\mu}, p_{N,\mu}, d_{\mu_i} u_{N,\mu}, d_{\mu_i} p_{N,\mu})[e_\mu^{pr}] + \partial_{\mu_i} a_\mu(e_\mu^{pr}, p_{N,\mu}) \\
&- 2k_\mu(d_{\mu_i} u_{N,\mu}, e_\mu^{pr}) - \partial_{\mu_i} j_\mu(e_\mu^{pr}) - 2\partial_{\mu_i} k_\mu(e_\mu^{pr}, u_{N,\mu}),
\end{aligned} \tag{22}$$

$$\begin{aligned}
(II) &= j_\mu(d_{\mu_i} u_{N,\mu}) + 2k_\mu(d_{\mu_i} e_\mu^{pr}) \\
&- j_\mu(d_{\mu_i} u_{N,\mu}) - a_\mu(d_{\mu_i} u_{N,\mu}, e_\mu^{du}).
\end{aligned} \tag{23}$$

We then add up all terms from (21), (22) and (23), and with the use of

$$\begin{aligned}
\partial_{\mu_i} k_\mu(e_\mu^{pr}, e_\mu^{pr}) &= \partial_{\mu_i} k_\mu(e_\mu^{pr}, u_\mu) + \partial_{\mu_i} k_\mu(e_\mu^{pr}, u_{N,\mu}) - 2\partial_{\mu_i} k_\mu(e_\mu^{pr}, u_{N,\mu}) \\
&+ \partial_{\mu_i} k_\mu(u_\mu, u_{N,\mu}) - \partial_{\mu_i} k_\mu(u_\mu, u_{N,\mu}) \\
&= \partial_{\mu_i} k_\mu(u_\mu, u_\mu) - \partial_{\mu_i} k_\mu(u_{N,\mu}, u_{N,\mu}) - 2\partial_{\mu_i} k_\mu(e_\mu^{pr}, u_{N,\mu}),
\end{aligned}$$

and

$$\begin{aligned}
&\partial_{\mu_i} Res_\mu^{pr}(u_\mu)[p_\mu] - \partial_{\mu_i} Res_\mu^{pr}(u_{N,\mu})[p_{N,\mu}] \\
&\qquad = \partial_{\mu_i} l_\mu(e_\mu^{pr}) - \partial_{\mu_i} a_\mu(u_\mu, p_\mu) + \partial_{\mu_i} a_\mu(u_{N,\mu}, p_{N,\mu}) \\
&\qquad = \partial_{\mu_i} a_\mu(u_{N,\mu}, p_{N,\mu}) - \partial_{\mu_i} a_\mu(u_\mu, p_\mu) + \partial_{\mu_i} a_\mu(u_\mu, e_\mu^{du}) \\
&\qquad + \underbrace{\partial_{\mu_i} l_\mu(e_\mu^{du}) - \partial_{\mu_i} a_\mu(u_\mu, e_\mu^{du})}_{=\partial_{\mu_i} Res_\mu^{pr}(u_\mu)[e_\mu^{du}]}
\end{aligned}$$

we get

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

and thus the assertion. The same holds for the inexact version. □