Journal of Computational Physics 504 (2024) 112863

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

MORe DWR: Space-time goal-oriented error control for incremental POD-based ROM for time-averaged goal functionals

Hendrik Fischer [a,b,*], Julian Roth [a,b], Thomas Wick [a,b], Ludovic Chamoin [b], Amelie Fau [b]

[a] Leibniz Universität Hannover, Institut für angewandte Mathematik, AG Wissenschaftliches Rechnen, Welfengarten 1, 30167 Hannover, Germany
[b] Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS, LMPS - Laboratoire de Mécanique Paris-Saclay, 91190 Gif-sur-Yvette, France

ARTICLE INFO

Keywords:
Tensor-product space-time reduced-order modeling
Dual-weighted residual method
Goal-oriented error control
Incremental proper orthogonal decomposition

ABSTRACT

In this work, the dual-weighted residual (DWR) method is applied to obtain an error-controlled incremental proper orthogonal decomposition (POD) based reduced order model. A novel approach called MORe DWR (Model Order Reduction with Dual-Weighted Residual error estimates) is being introduced. It marries tensor-product space-time reduced-order modeling with time slabbing and an incremental POD basis generation with goal-oriented error control based on dual-weighted residual estimates. The error in the goal functional is being estimated during the simulation and the POD basis is being updated if the estimate exceeds a given threshold. This allows an adaptive enrichment of the POD basis in case of unforeseen changes in the solution behavior. Consequently, the offline phase can be skipped, the reduced-order model is being solved directly with the POD basis extracted from the solution on the first time slab and –if necessary– the POD basis is being enriched on-the-fly during the simulation with high-fidelity finite element solutions. Therefore, the full-order model solves can be reduced to a minimum, which is demonstrated on numerical tests for the heat equation and elastodynamics using time-averaged quantities of interest.

1. Introduction

Model order reduction (MOR) by means of the proper orthogonal decomposition (POD) has been applied for cheap surrogate modeling to a plethora of partial differential equations (PDEs) [1–11]. Therein, the dynamics is projected onto a set of POD modes that constitute an approximate basis for the solution manifold to reduce the cost of running expensive high-fidelity simulations. This proper orthogonal decomposition based reduced-order modeling (POD-ROM) is a truth approximation because it yields a compressed representation of an a priori known solution trajectory. To avoid the necessity of these expensive high-fidelity simulations beforehand, we use error estimates to only locally perform high-fidelity calculations.

The dual-weighted residual method is used in this work to switch between ROM and high-fidelity computations. The space-time dual-weighted residual (DWR) method is an extension of the DWR method for stationary problems introduced in [12–14], which is based on seminal prior work of Johnson and co-workers [15]; an overview on the usage with adaptive predictive multiscale modeling was published by Oden [16]. The space-time DWR method has been applied to parabolic PDEs by Schmich (Besier) and Vexler [17],

* Corresponding author at: Leibniz Universität Hannover, Institut für angewandte Mathematik, AG Wissenschaftliches Rechnen, Welfengarten 1, 30167 Hannover, Germany.
E-mail address: fischer@ifam.uni-hannover.de (H. Fischer).

https://doi.org/10.1016/j.jcp.2024.112863
Received 3 April 2023; Received in revised form 4 December 2023; Accepted 13 February 2024

Available online 19 February 2024

Schmich (Besier) [18] and Besier and Rannacher [19], Endtmayer et al. [20], and in the authors' own works [21,22]. Moreover, it has been applied to hyperbolic PDEs in the dissertation of Rademacher [23] and to the wave equation by Bangerth et al. [24]. Since the theory for the error estimation is formulated in spatio-temporal function spaces and requires space-time variational formulations, we employ a space-time finite element method (FEM) discretization; see for instance [25] and the many references cited therein. Space-time finite elements for the heat equation have been studied, e.g., in [17,26] and for the elastodynamics equation, e.g., in [27,24,28]. Similar space-time FEM implementations can be found in FEniCS in [29] and in NGSolve in [30,31].

In recent years, space-time formulations have been applied to model order reduction [32–35], including a windowed space-time approach for more efficiency [36]. Additional applications of space-time model order reduction include optimal control [37] and classical error estimates and hyper-reduction estimates using discrete empirical interpolation [38]. Further, the advantages of space-time formulations in the context of certified reduced basis methods were extensively investigated in [39–42]. It has been demonstrated that in long-time settings, where traditional time-marching L^2 error bounds grow exponentially in time and are not efficient, space-time formulations allow sharp and effective error bounds.

A lot of research on DWR error estimates for hyper-reduction with reduced quadrature rules has been done by Yano [43,44]. Another reduced-order modeling approach employing goal-oriented error estimates has been proposed by Meyer and Matthies [45], where the estimates have been used to remove POD basis vectors that are not relevant for the accurate computation of the quantity of interest. Finally, related methods include the proper generalized decomposition (PGD) [46] and hierarchical model (HiMod) reduction [47–49], which uses estimates for the POD in the transverse direction of the dynamics.

In this work, we propose a different methodology for POD-ROM computations in which only a small portion of the solution trajectory is being computed with the expensive full-order-model (FOM) and the reduced-order-model (ROM) is being updated on-the-fly when the error estimates exceed a prescribed tolerance. This is being accomplished by combining POD-ROM with the incremental POD and space-time dual-weighted residual error estimates. We work out the algorithmic details, resulting in a final newly proposed algorithm for incremental ROM. The presented incremental POD implementation relies on additive rank-b updates of the singular value decomposition [50,51] and has successfully been applied in a parallel SIMD framework to the incremental model order reduction of high-dimensional turbulent fluid flows [52]. In contrast to previous work on incremental POD, e.g. [53,54], this allows updates with multiple snapshots at once, which is beneficial in a space-time setting. In addition, the SIMD parallelization provides scalability [52] for future work without the disadvantage of a priori knowledge of the final POD size required in hierarchical approaches such as in [55].

As previously mentioned, as an overall framework, we employ a space-time setting. More concretely, we rely on the tensor-product space-time FEM implementation from [22] based on the FEM library deal.II [56,57]. The final algorithm is implemented and demonstrated for time-averaged goal functionals with various settings that include parabolic problems (heat equation) and second-order hyperbolic problems (elastodynamics). The main objective is to show the decrease in computational cost by keeping the accuracy of the numerical solutions. Moreover, the error estimator and the goal functional are compared in terms of effectivities.

The outline of this paper is as follows: In Section 2, we formulate the problem for the heat equation and elastodynamics and discretize them with tensor-product space-time finite elements. Next, in Section 3 we recapitulate POD-based reduced-order modeling and depict its extension to tensor-product space-time POD-ROM. Then, in Section 4 the theories for the space-time error estimates and the incremental model order reduction are elucidated. In Section 5, numerical tests in 1+1D, 2+1D and 3+1D are being conducted for the heat equation and elastodynamics. Finally, our findings are summarized in Section 6.

2. Problem formulation and discretization

2.1. Model problem formulation

Let $\tilde{d} \in \mathbb{N}$ with $\tilde{d}$ depending on whether the problem is vector- or scalar-valued, i.e. for the heat equation we have $\tilde{d} = 1$, whereas for elastodynamics in u-formulation (where u denotes the displacements) we have $\tilde{d} = d$ and for the (u, v)-formulation (where u is as before and v denotes the velocity), we have $\tilde{d} = 2d$, where $d \in \{1, 2, 3\}$ is the spatial dimension. In the problem description, $I := (0, T)$ denotes the temporal domain and $\Omega \subset \mathbb{R}^d$ a sufficiently smooth spatial domain. Here, the spatial boundary is split into a Dirichlet boundary $\Gamma_D \subseteq \partial\Omega$ and a Neumann boundary $\Gamma_N \subsetneq \partial\Omega$ with $\Gamma_D \cap \Gamma_N = \emptyset$. We consider the abstract time-dependent problem: Find $u : \bar{\Omega} \times \bar{I} \to \mathbb{R}^{\tilde{d}}$ such that

$$\begin{aligned} \partial_t u + \mathcal{A}(u) &= f && \text{in } \Omega \times I, \\ u &= u_D && \text{on } \Gamma_D \times I, \\ \mathcal{B}(u) &= g_N && \text{on } \Gamma_N \times I, \\ u &= u^0 && \text{in } \Omega \times \{0\}, \end{aligned} \tag{1}$$

with possibly nonlinear spatial operator $\mathcal{A}$, boundary operator $\mathcal{B}$ and sufficiently regular right-hand side f. Here, u_D is the Dirichlet boundary function, g_N is the Neumann boundary function and u^0 is the initial condition. Choosing a suitable continuous spatial function space $V := V(\Omega)$, a continuous temporal functional space $X := X(I, \cdot)$ and time-dependent Sobolev space $X(I, V(\Omega))$ (see e.g., [58–60]) mapping from I into $V(\Omega)$, we can define the continuous spatio-temporal variational formulation as: Find $u \in u_D + X(I, V(\Omega))$ such that

$$A(u)(\varphi) := ((\partial_t u, \varphi)) + ((\mathcal{A}(u), \varphi)) + (u(0), \varphi(0))$$

$$= (\!(f,\varphi)\!) + \langle\!\langle g_N - \mathcal{B}(u),\varphi\rangle\!\rangle_{\Gamma_N} + (u^0,\varphi(0)) =: F(\varphi) \qquad \forall \varphi \in X(I,V(\Omega)),$$

where we use the notation

$$(f,g) := (f,g)_{L^2(\Omega)} := \int_\Omega f\cdot g \,\mathrm{d}x, \qquad (\!(f,g)\!) := (f,g)_{L^2(I,L^2(\Omega))} := \int_I (f,g)\,\mathrm{d}t,$$

$$\langle f,g\rangle := \langle f,g\rangle_{L^2(\Gamma)} := \int_\Gamma f\cdot g \,\mathrm{d}s, \qquad \langle\!\langle f,g\rangle\!\rangle := (f,g)_{L^2(I,L^2(\Gamma))} := \int_I \langle f,g\rangle\,\mathrm{d}t.$$

In this notation, $f\cdot g$ represents the Euclidean inner product if f and g are scalar- or vector-valued and it stands for the Frobenius inner product if f and g are matrices. We notice that some partial differential equations that fall into this framework are the heat equation and more generally parabolic problems. With a bit of abuse of notation, elastodynamics formulated as a first-order-in-time system can also be written in the above form, which we however specify below for the sake of mathematical precision.

2.1.1. Heat equation

The strong formulation of the heat equation reads: Find the temperature $u : \bar{\Omega}\times\bar{I}\to\mathbb{R}$ such that

$$\partial_t u - \Delta_x u = f \qquad \text{in } \Omega\times I,$$

with $\mathcal{A}(u) := -\Delta_x u$ in (1). The initial and boundary conditions are given by

$$u = u^0 \qquad \text{on } \Omega\times\{0\},$$

$$u = 0 \qquad \text{on } \partial\Omega\times I.$$

For the functional framework of the weak formulation, we follow [17] and first setup $V(\Omega) := H_0^1(\Omega)$, the dual space $H_0^1(\Omega)^*$ (it is well known that $H^{-1}(\Omega) = H_0^1(\Omega)^*$), where we have the Gelfand triple $V \hookrightarrow L^2(\Omega) \hookrightarrow V^*$. With this, we define the Hilbert space

$$X(I,V(\Omega)) := \{u|\ u\in L^2(I,V(\Omega)),\ \partial_t u \in L^2(I,H_0^1(\Omega)^*)\}.$$

It is well known that the space $X(I,V(\Omega))$ is embedded continuously in $C(\bar{I},L^2(\Omega))$; see e.g., [60][Paragraph 5.9.2, Theorem 3] or [58][Chapter XVIII] such that the initial conditions u^0 are well defined. Moreover, we notice that

$$\{u|\ u\in L^2(I,V(\Omega)),\ \partial_t u \in L^2(I,H_0^1(\Omega)^*)\} \cong \{u\in L^2(I,H_0^1(\Omega)) \cap H^1(I,(H_0^1(\Omega))^*)\}.$$

Consequently, $X(I,V(\Omega))$ constitutes the trial space in the weak form. The test functions are defined in the sense of distributions and due to density arguments (see again [60][Chapter 7] or [58][Chapter XVIII]) they can be taken as well from $X(I,V(\Omega))$. We thus arrive at the continuous variational formulation:

Formulation 2.1 *(Continuous variational formulation of the heat equation).* Find $u\in X(I,V(\Omega))$ such that

$$A(u)(\varphi) := (\!(\partial_t u,\varphi)\!) + (\!(\nabla_x u,\nabla_x\varphi)\!) + (u(0),\varphi(0)) = (\!(f,\varphi)\!) + (u^0,\varphi(0)) =: F(\varphi) \qquad \forall\varphi\in X(I,V(\Omega)).$$

For this variational formulation, we use $u^0\in L^2(\Omega)$ and $f\in L^2(I,H_0^1(\Omega)^*)$ [59,58,60].

2.1.2. Elastodynamics equation

The strong formulation of linear elastodynamics in three spatial dimensions reads: Find the displacement $u : \bar{\Omega}\times\bar{I}\to\mathbb{R}^d$ such that

$$\partial_{tt} u - \nabla_x\cdot\sigma(u) = f \qquad \text{in } \Omega\times I,$$

with an external force $f := f(t,x)$ and

$$\sigma(u) = 2\mu E(u) + \lambda\,\mathrm{tr}(E(u))\mathbb{1}_{d\times d}, \qquad \text{(stress tensor)}$$

$$E(u) = \frac{1}{2}(\nabla_x u + (\nabla_x u)^T), \qquad \text{(linearized strain tensor)}$$

where $\mathbb{1}_{d\times d}\in\mathbb{R}^{d\times d}$ is the identity matrix and the Lamé parameters are $\mu>0$ and $\lambda > -\frac{2}{3}\mu$. The initial conditions are given by

$$u = u^0 \qquad \text{on } \Omega\times\{0\},$$

$$\partial_t u = v^0 \qquad \text{on } \Omega\times\{0\}.$$

As boundary conditions, we prescribe

$$u = 0 \qquad \text{on } \Gamma_D \times I,$$
$$\mathcal{B}(u) = \sigma(u) \cdot n = g_N \qquad \text{on } \Gamma_N \times I,$$

with n denoting the normal unit vector to Γ_N. We convert this into a first-order system (as it is usually done in the literature [27,24] and [28][Chapter 1]) in time and solve for displacement $u : \bar{\Omega} \times \bar{I} \to \mathbb{R}^d$ and velocity $v : \bar{\Omega} \times \bar{I} \to \mathbb{R}^d$ such that

$$\partial_t v - \nabla_x \cdot \sigma(u) = f \qquad \text{in } \Omega \times I,$$
$$\partial_t u - v = 0 \qquad \text{in } \Omega \times I,$$

with $\mathcal{A}(u,v) := \begin{pmatrix} -v \\ -\nabla_x \cdot \sigma(u) \end{pmatrix}$ and $\partial_t(u,v) := \begin{pmatrix} \partial_t u \\ \partial_t v \end{pmatrix}$ in (1). We still have the same initial and boundary conditions with the only difference that we now have

$$v = v^0 \qquad \text{on } \Omega \times \{0\},$$
$$v = 0 \qquad \text{on } \Gamma_D \times I.$$

For setting up the functional framework, we refer to [60][Chapter 7.2]. Specifically, for the variational formulation, we use $u^0 \in H^1_{\Gamma_D,0}(\Omega)^d$, which is the space of weakly differentiable functions that vanish on Γ_D, $v^0 \in L^2(\Omega)^d, g_N \in L^2(I, L^2(\Gamma_N)^d)$. First, as for the heat equation, we notice that for the scalar-valued wave equation the initial conditions are well-defined for [60][Paragraph 5.9.2, Theorem 2]

$$u \in L^2(I, H^1_0(\Omega)), \quad \partial_t u \in L^2(I, L^2(\Omega)), \quad \partial_{tt} u \in L^2(I, H^{-1}(\Omega),$$

then $u \in C(\bar{I}, L^2(\Omega)$ and $\partial_t u \in C(\bar{I}, H^{-1}(\Omega))$. These results transfer to the elastic wave equation that we consider here such that we can define the Hilbert space (with respect to our boundary conditions):

$$X(I, V(\Omega)) := \{u |\, u \in L^2(I, H^1_{\Gamma_D,0}(\Omega)^d), \quad \partial_t u \in L^2(I, L^2(\Omega)^d), \quad \partial_{tt} u \in L^2(I, H^1_{\Gamma_D,0}(\Omega)^d)^* \}.$$

The corresponding isometric isomorphic function spaces for the displacements and velocities are:

$$X(I, V^u(\Omega)) := L^2(I, H^1_{\Gamma_D,0}(\Omega)^d) \cap H^1(I, L^2(\Omega)^d) \cap H^2\left(I, (H^1_{\Gamma_D,0}(\Omega)^d)^*\right),$$
$$X(I, V^v(\Omega)) := L^2(I, L^2(\Omega)^d) \cap H^1\left(I, (H^1_{\Gamma_D,0}(\Omega)^d)^*\right),$$
$$X(I, V(\Omega)) := X(I, V^u(\Omega)) \times X(I, V^v(\Omega)).$$

We thus solve the continuous variational formulation:

Formulation 2.2 *(Continuous variational formulation of the elastodynamics equation).* Find $U = (u, v) \in X(I, V(\Omega))$ such that

$$A(U)(\Phi) = F(\Phi) \qquad \forall \Phi = (\varphi^u, \varphi^v) \in X(I, V(\Omega)),$$

where

$$A(U)(\Phi) := ((\partial_t v, \varphi^u)) + ((\sigma(u), \nabla_x \varphi^u)) + (v(0), \varphi^u(0)) + ((\partial_t u, \varphi^v)) - ((v, \varphi^v)) + (u(0), \varphi^v(0)),$$
$$F(\Phi) := ((f, \varphi^u)) + (v^0, \varphi^u(0)) + \langle\langle g_N, \varphi^u \rangle\rangle_{\Gamma_N} + (u^0, \varphi^v(0)).$$

2.2. Tensor-product space-time FEM discretization

We follow our recent work on space-time adaptivity for the Navier-Stokes equations [22] and use tensor-product space-time finite elements (FEM) with discontinuous Galerkin finite elements in time (dG) and continuous Galerkin finite elements in space (cG). Using the tensor-product of the temporal and spatial basis functions is a special case of the broad class of space-time finite element methods [25]. We will now explain tensor-product space-time FEM at the example of the heat equation, where the function spaces can be found in [17] and the slabwise tensor-product space-time implementation is being outlined in [21] and [61]. We assume that the spatial mesh remains fixed, which simplifies the analysis and the implementation. Furthermore, we outline the extension of this methodology to elastodynamics.

2.2.1. Discretization in time

Let $\mathcal{T}_k := \{ I_m := (t_{m-1}, t_m) \mid 1 \le m \le M \}$ be a partitioning of time, i.e. $\bar{I} = [0,T] = \bigcup_{m=1}^{M} \bar{I}_m$, where the subscript k denotes the temporal discretization parameter. We now introduce broken continuous level function spaces

$$\tilde{X}(\mathcal{T}_k, V(\Omega)) := \{ v \in L^2(I, L^2(\Omega)) \mid v|_{I_m} \in X(I_m, V(\Omega)) \quad \forall I_m \in \mathcal{T}_k \}$$

for the heat equation and

$$\tilde{X}(\mathcal{T}_k, V^u(\Omega)) := \{v \in L^2(I, L^2(\Omega)^d) \mid v|_{I_m} \in X(I_m, V^u(\Omega)) \quad \forall I_m \in \mathcal{T}_k\},$$
$$\tilde{X}(\mathcal{T}_k, V^v(\Omega)) := \{v \in L^2(I, L^2(\Omega)^d) \mid v|_{I_m} \in X(I_m, V^v(\Omega)) \quad \forall I_m \in \mathcal{T}_k\},$$
$$\tilde{X}(\mathcal{T}_k, V(\Omega)) := \tilde{X}(\mathcal{T}_k, V^u(\Omega)) \times \tilde{X}(\mathcal{T}_k, V^v(\Omega))$$

for the elastodynamics equation. These broken function spaces [62] are required, since we want to perform a conforming (w.r.t. the broken spaces) discontinuous Galerkin discretization in time and thus need to allow for discontinuities between time intervals/temporal elements. Due to these discontinuities, we define the limits of f at time t_m from above and from below for a function f as

$$f_m^{\pm} := \lim_{\epsilon \searrow 0} f(t_m \pm \epsilon),$$

and the jump of the function value of f at time t_m as

$$[f]_m := f_m^+ - f_m^-.$$

The function spaces enable us to include discontinuities in the variational formulations:

Formulation 2.3 *(Time-discontinuous variational formulation of the heat equation).* Find $u \in \tilde{X}(\mathcal{T}_k, V(\Omega))$ such that

$$\tilde{A}(u)(\varphi) = \tilde{F}(\varphi) \qquad \forall \varphi \in \tilde{X}(\mathcal{T}_k, V(\Omega)),$$

where

$$\tilde{A}(u)(\varphi) := \sum_{m=1}^{M} \int_{I_m} (\partial_t u, \varphi) + (\nabla_x u, \nabla_x \varphi)\, \mathrm{d}t + \sum_{m=1}^{M-1} ([u]_m, \varphi_m^+) + (u_0^+, \varphi_0^+),$$
$$\tilde{F}(\varphi) := ((f, \varphi)) + (u^0, \varphi_0^+).$$

Formulation 2.4 *(Time-discontinuous variational formulation of the elastodynamics equation).* Find $U = (u, v) \in \tilde{X}(\mathcal{T}_k, V(\Omega))$ such that

$$\tilde{A}(U)(\Phi) = \tilde{F}(\Phi) \qquad \forall \Phi = (\varphi^u, \varphi^v) \in \tilde{X}(\mathcal{T}_k, V(\Omega)),$$

where

$$\tilde{A}(U)(\Phi) := \sum_{m=1}^{M} \int_{I_m} (\partial_t v, \varphi^u) + (\sigma(u), \nabla_x \varphi^u) + (\partial_t u, \varphi^v) - (v, \varphi^v)\, \mathrm{d}t$$
$$+ \sum_{m=1}^{M-1} (([v]_m, \varphi_m^{u,+}) + ([u]_m, \varphi_m^{v,+})) + (v_0^+, \varphi_0^{u,+}) + (u_0^+, \varphi_0^{v,+}),$$
$$\tilde{F}(\Phi) := ((f, \varphi^u)) + (v^0, \varphi_0^{u,+}) + \langle\langle g_N, \varphi^u \rangle\rangle_{\Gamma_N} + (u^0, \varphi_0^{v,+}).$$

We have the inclusions $X(I, \cdot) \subset \tilde{X}(\mathcal{T}_k, \cdot)$, since for continuous functions the jump terms vanish, and thus the variational Formulation 2.3 and Formulation 2.4 are consistent.

Next, we define the semi-discrete space for the heat equation as

$$X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V(\Omega)) := \left\{ v_k \in L^2(I, L^2(\Omega)) \,\middle|\, v_k|_{I_m} \in P_r(I_m, H_0^1(\Omega)) \right\} \subset \tilde{X}(\mathcal{T}_k, V(\Omega))$$

and for the elastodynamics equation as

$$X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^u(\Omega)) := \left\{ v_k \in L^2(I, L^2(\Omega)^d) \,\middle|\, v_k|_{I_m} \in P_r(I_m, H^1_{\Gamma_D,0}(\Omega)^d) \right\} \subset \tilde{X}(\mathcal{T}_k, V^u(\Omega)),$$
$$X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^v(\Omega)) := X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^u(\Omega)),$$
$$X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V(\Omega)) := X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^u(\Omega)) \times X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^v(\Omega)),$$

where the space-time function spaces $\tilde{X}(\mathcal{T}_k, \cdot)$ have been discretized in time with the dG method of order $r \in \mathbb{N}_0$ (dG(r)). Typical choices in our work for the temporal degree are $r = 1$ and $r = 2$. Here, $P_r(I_m, Y)$ is the space of polynomials of order r, which map from the time interval I_m into the space Y. The dG(r) time discretization for the case $r = 1$ is illustrated in Fig. 1.
The locations of the temporal degrees of freedom (DoFs) are defined by quadrature rules. Due to the discontinuity of the temporal discretization, various quadrature rules can be chosen, the most common being Gauss-Lobatto, Gauss-Legendre and Gauss-Radau. In Fig. 1 the locations of the temporal degrees of freedom are chosen at the ends of the time intervals, which corresponds to Gauss-Lobatto quadrature. In Section 5, we use Gauss-Legendre and Gauss-Lobatto quadrature in time to demonstrate the versatility of our method concerning the choice of the temporal quadrature formula.

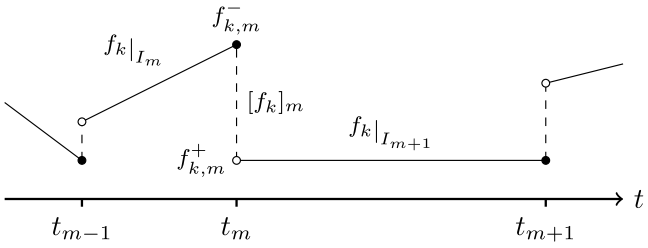


Fig. 1. dG(1) time discretization.

It has been derived in [63] (see also the classical textbooks [64,65]) that the dG(0) time-discretization is a variant of the backward Euler scheme. Higher-order schemes are derived as well and it was established that $dG(r_p)$ discretizations, where $r_p \in \mathbb{N}_0$ is the polynomial degree, are generically implicit and A-stable.

2.2.2. Discretization in space

For the spatial discretization of the variational formulation, we use a fixed mesh $\mathcal{T}_h$, which consists of intervals in one dimension and of quadrilateral (2D) or hexahedral (3D) elements in higher dimensions. Here, the subscript h represents the spatial discretization parameter We can then use element-wise polynomial functions of up to order $s \in \mathbb{N}$ as our spatial function space, i.e.,

$$V_h^s := V_h^s(\mathcal{T}_h) := \left\{ v \in C(\bar{\Omega}) \middle| v|_K \in \mathcal{Q}_s(K) \quad \forall K \in \mathcal{T}_h \right\}$$

for the heat equation and for the elastodynamics equation

$$V_h^{s,u} := V_h^{s,u}(\mathcal{T}_h) := \left\{ v \in C(\bar{\Omega})^d \middle| v|_K \in (\mathcal{Q}_s(K))^d \quad \forall K \in \mathcal{T}_h \right\} =: V_h^{s,v}(\mathcal{T}_h) =: V_h^{s,v},$$

where $\mathcal{Q}_s(K)$ is being constructed by mapping tensor-product polynomials of degree s from the reference element $(0,1)^d$ to the element K. The fully discrete function space for the heat equation is then given by

$$X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s) := \left\{ v_{kh} \in L^2(I, L^2(\Omega)) \,\middle|\, v_{kh}|_{I_m} \in P_r(I_m, V_h^s) \quad \forall I_m \in \mathcal{T}_k \right\}$$

and for the elastodynamics equation

$$X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s) := \left\{ v_{kh} \in L^2(I, L^2(\Omega)^{2d}) \,\middle|\, v_{kh}|_{I_m} \in P_r(I_m, V_h^s) \quad \forall I_m \in \mathcal{T}_k \right\},$$
$$V_h^s := V_h^{s,u} \times V_h^{s,v}.$$

Thus, the fully discrete variational formulation reads for the heat equation:

Find $u_{kh} \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s)$ such that

$$\tilde{A}(u_{kh})(\varphi_{kh}) = \tilde{F}(\varphi_{kh}) \quad \forall \varphi_{kh} \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s).$$

Moreover, the fully discrete variational formulation for the elastodynamics equation reads:

Find $U_{kh} := (u_{kh}, v_{kh}) \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s)$ such that

$$\tilde{A}(U_{kh})(\Phi_{kh}) = \tilde{F}(\Phi_{kh}) \quad \forall \Phi_{kh} = (\varphi_{kh}^u, \varphi_{kh}^v) \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s).$$

2.2.3. Slabwise discretization

Finally, we want to remark that the fully discrete variational formulations do not need to be solved on the entire space-time cylinder $\Omega \times I$, but can also be solved sequentially on space-time slabs

$$S_l^n := \Omega \times \left(\bigcup_{m=l}^{n} I_m \right),$$

where $1 \le l \le n \le M$. As mentioned previously, we can then get the space-time FEM basis on S_l^n by taking the tensor-product of the spatial and the temporal finite element basis functions. This simplifies the finite element discretization of the abstract time-dependent problem (1), since the main prerequisite is a FEM code for the stationary problem $\mathcal{A}(u) = f$ in Ω. Furthermore, tensor-product space-time FEM allows for larger flexibility in the choice of temporal discretization, since changing the temporal degree of the space-time discretization can be performed simply by changing the polynomial degree of the temporal finite elements. Due to the tensor-product structure of the space-time FE basis, it is straightforward how proper orthogonal decomposition (POD) based reduced-order modeling can be performed, since on an abstract level only the spatial finite element basis needs to be replaced by the spatial POD basis.

For the heat equation on the space-time slab S_l^n with $n-l+1$ time intervals, we arrive at the linear equation system

$$\begin{pmatrix} A & & & & \mathbf{0} \\ B & A & & & \\ & B & A & & \\ & & \ddots & \ddots & \\ \mathbf{0} & & & B & A \end{pmatrix} \begin{pmatrix} U_l \\ U_{l+1} \\ U_{l+2} \\ \vdots \\ U_n \end{pmatrix} = \begin{pmatrix} F_l - BU_{l-1} \\ F_{l+1} \\ F_{l+2} \\ \vdots \\ F_n \end{pmatrix} \tag{2}$$

or in brevity

$$A_{S_l^n} U_{S_l^n} = F_{S_l^n} \tag{3}$$

with

$$A = C_k \otimes M_h + M_k \otimes K_h,$$

$$B = -D_k \otimes M_h,$$

where we use the spatial matrices

$$M_h = \left\{ (\varphi_h^{(j)}, \varphi_h^{(i)}) \right\}_{i,j=1}^{\#\mathrm{DoFs}(\mathcal{T}_h)},$$

$$K_h = \left\{ (\nabla_x \varphi_h^{(j)}, \nabla_x \varphi_h^{(i)}) \right\}_{i,j=1}^{\#\mathrm{DoFs}(\mathcal{T}_h)},$$

and the temporal matrices

$$M_k = \left\{ \int_{I_m} \varphi_k^{(j)} \cdot \varphi_k^{(i)} \, \mathrm{d}t \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)},$$

$$C_k = \left\{ \int_{I_m} \partial_t \varphi_k^{(j)} \cdot \varphi_k^{(i)} \, \mathrm{d}t + \varphi_{k,m-1}^{(j),+} \cdot \varphi_{k,m-1}^{(i),+} \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)},$$

$$D_k = \left\{ \varphi_{k,m-1}^{(j),-} \cdot \varphi_{k,m-1}^{(i),+} \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)}.$$

Note that $U_l, \dots, U_n$ are space-time vectors themselves, where $U_m \in \mathbb{R}^{\#\mathrm{DoFs}(I_m) \cdot \#\mathrm{DoFs}(\mathcal{T}_h)}$ with $m = l, \dots, n$ is the coefficient vector of the solution u_{kh} on the time interval I_m, i.e., for the dG(r) method in time with temporal quadrature points $t_1, \dots, t_{r+1}$ we have

$$U_m = \begin{pmatrix} U_m(t_1) \\ \vdots \\ U_m(t_{r+1}) \end{pmatrix}, \quad m = 1, \dots, M,$$

where M is the total number of time intervals. In particular, if we use space-time slabs that contain only one temporal element, then we only need to solve the linear system

$$AU_m = F_m - BU_{m-1}$$

for each time slab $S_m := S_m^m = \mathcal{T}_h \times I_m$. For efficiency reasons, in the remainder of this paper, we only consider such slabs of size one.

For the elastodynamics equation, the space-time FEM linear system can be derived similarly. The linear system and time-stepping formulations for dG(1) and dG(2) with Gauss-Lobatto quadrature in time can be found in Appendix A.

Remark 2.5. Although the linear systems for the heat equation in this section and for the elastodynamics equation in Appendix A have been presented as the tensor product of temporal and spatial matrices, tensor-product space-time FEM can be applied to a much larger class of problems. For example, if the PDE contains coefficients that depend on space and time, it is not always possible to decompose a space-time linear system into this tensor-product structure. Nevertheless, our implementation of tensor-product space-time FEM is general enough to also deal with these kinds of problems, since it does not rely on a tensor-product of the linear system but only on the tensor-product structure of the finite element basis.

3. Reduced-order modeling

3.1. POD-ROM

The increase in computational power in the last decades has made it possible to exploit high-performance computing for large-scale numerical simulations. Nevertheless, in some scenarios, e.g. for multiphysics problems, high-performance computing can be computationally expensive, in particular also having a large carbon footprint and enormous energy consumption. These circumstances motivate the application of model order reduction (MOR) techniques on the premise of a large computational speedup to satisfy these demands. In this work, we mainly deal with projection-based reduced basis methods (RBM) [66–72] since this methodology aims at efficient treatments by providing both an approximate solution procedure and efficient error estimates [66]. Here, the critical observation is that instead of using projection spaces with general approximation properties (e.g. finite element method) problem-specific approximation spaces are chosen and then can be used for the discretization of the original problem [73]. Based on these spaces and the assumption that the solution evolves smoothly in a low-dimensional solution manifold (equivalent to a small Kolmogorov N-width [74,6,67]), a reduced-order model (ROM) can be constructed that represents with sufficient accuracy the physical problem of interest using a significantly smaller number of degrees of freedom [73].

In order to construct the reduced spaces, the solution manifold is empirically explored by means of solutions of the full-order model as developed in Section 2.2. Then, a proper orthogonal decomposition (POD) is conducted on these snapshots of the high-fidelity solution to obtain the reduced basis functions [69,6,75,76,1,77–81]. The following Theorem 3.1 states that the POD basis is optimal in a least-squares sense. The proof is provided by Gubisch and Volkwein in [82].

Theorem 3.1 *(POD basis). Let* $Y = [Y_1, \dots, Y_q] := [U_1(t_1), \dots, U_1(t_{r+1}), U_2(t_1), \dots, U_M(t_{r+1})] \in \mathbb{R}^{n\times q}$ *with* $q = M \cdot \#DoFs(I_m)$, $n = \#DoFs(\mathcal{T}_h)$ *and rank* $d \leq \min(n,q)$ *be the snapshot matrix with a (spatial) column vector for each temporal degree of freedom. Moreover, let* $Y = \Psi\Sigma\Phi^T$ *be its singular value decomposition with* $\Sigma = diag(\sigma_1, \dots, \sigma_d) \in \mathbb{R}^{d\times d}$ *and orthogonal matrices* $\Psi = [\psi_1, \dots, \psi_d] \in \mathbb{R}^{n\times d}$, $\Phi = [\phi_1, \dots, \phi_d] \in \mathbb{R}^{q\times d}$. *Then for* $1 \leq N \leq d$ *the optimization problem is*

$$\min_{\tilde\psi_1,\dots,\tilde\psi_N \in \mathbb{R}^n} \sum_{j=1}^{q} \left\| Y_j - \sum_{i=1}^{N} \left(Y_j, \tilde\psi_i\right)_{\mathbb{R}^n} \tilde\psi_i \right\|_{\mathbb{R}^n}^2 \quad s.t. \quad (\tilde\psi_i, \tilde\psi_j)_{\mathbb{R}^n} = \delta_{ij} \ \forall 1 \leq i,j \leq N, \tag{$\mathbf{P}^N$}$$

where $\{\tilde\psi_i\}_{i=1}^N \subset \mathbb{R}^n$, *and which is being solved by the left-singular vectors* $\{\psi_i\}_{i=1}^N \subset \mathbb{R}^n$ *and it holds that*

$$\sum_{j=1}^{q} \left\| Y_j - \sum_{i=1}^{N} \left(Y_j, \psi_i\right)_{\mathbb{R}^n} \psi_i \right\|_{\mathbb{R}^n}^2 = \sum_{i=N+1}^{d} \sigma_i^2 = \sum_{i=N+1}^{d} \lambda_i \tag{4}$$

with $\lambda_i = \sigma_i^2$ *being the positive eigenvalues of* $Y^T Y$.

Thus, the decay rate of the singular values plays an essential role in the feasibility of the POD approach. If the sum of the squared truncated singular values is sufficiently small for a relatively small N, we can utilize a linear combination of a few basis functions ψ_i for a good approximation of elements Y_j living in the high-dimensional FE space. Although the error of an obtained rank-N approximation can be determined by Equation (4), this does not yield an intuitive measure for rank determination. Thus, a widely used criterion to determine the quality of the POD basis heuristically refers to its retained energy or information content $\varepsilon(N)$, cf. [83,82,69]. The latter is defined by

$$\varepsilon(N) = \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{d} \sigma_i^2} = \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{q} ||U_i||_{\mathbb{R}^n}^2}. \tag{5}$$

For a detailed presentation of the construction possibilities of the POD basis, we refer to [6][Chap. 2].

3.2. Tensor-product space-time POD-ROM

In order to reduce the space-time full-order system (2) of Section 2.2 the general spatial FEM space V_h is replaced by a problem-specific low-dimensional space $V_N = \text{span}\{\varphi_N^1, \dots, \varphi_N^N\}$ obtained by means of POD. This yields the reduced variational formulation: Find $u_N \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_N)$ such that

$$\tilde A(u_N)(\varphi) = \tilde F(\varphi) \qquad \forall \varphi \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_N).$$

The reduced basis matrix can be formed by the concatenation of the reduced basis vectors, viz.

$$\Psi_N = \begin{bmatrix} \psi_1 & \dots & \psi_N \end{bmatrix} \in \mathbb{R}^{\#\text{DoFs}(\mathcal{T}_h)\times N}. \tag{6}$$

Subsequently, the slabwise discretization for the space-time slab S_l^n with $n - l + 1$ time intervals is obtained in analogy to the full-order model of Section 2.2.3. In the case of the heat equation, we utilize the linear equation system described in (2) and reduce the given matrices in an affine manner. Thus, we arrive at

$$\begin{pmatrix} A_N & & & & \mathbf{0} \\ B_N & A_N & & & \\ & B_N & A_N & & \\ & & \ddots & \ddots & \\ \mathbf{0} & & & B_N & A_N \end{pmatrix} \begin{pmatrix} U_{N_l} \\ U_{N_{l+1}} \\ U_{N_{l+2}} \\ \vdots \\ U_{N_n} \end{pmatrix} = \begin{pmatrix} F_{N_l} - B_N U_{N_{l-1}} \\ F_{N_{l+1}} \\ F_{N_{l+2}} \\ \vdots \\ F_{N_n} \end{pmatrix} \tag{7}$$

or in brevity

$$A_N U_{N,S_l^n} = F_{N,S_l^n} \tag{8}$$

with the reduced components

$$A_N = C_k \otimes M_N + M_k \otimes K_N, \tag{9a}$$

$$B_N = -D_k \otimes M_N, \tag{9b}$$

$$F_{N_i} = Z_N^T F_i, \quad l \le i \le n, \tag{9c}$$

based on the spatial reduced matrices

$$M_N = Z_N^T M Z_N,$$

$$K_N = Z_N^T K Z_N.$$

4. A posteriori goal-oriented error-controlled reduced-order modeling

For further analysis, we consider homogeneous Dirichlet boundary conditions to simplify the presentation, i.e. $u_D = 0$. Let a goal functional, also called output functional, $J : \tilde{X}(\mathcal{T}_k, V(\Omega)) \to \mathbb{R}$ of the form

$$J(u) = \int_0^T J_1(u(t))\, \mathrm{d}t + J_2(u(T)) \tag{10}$$

be given, which represents a physical quantity of interest (QoI). Here, T denotes the end time as before. In this work, we consider time-averaged goal functionals, i.e., $J_2 = 0$ and leave $J_2 \neq 0$ for future studies. Now, we want to reduce the difference between the quantity of interest of a fine solution u^{fine} and a coarse solution u^{coarse}, i.e.,

$$J(u^{\text{fine}}) - J(u^{\text{coarse}}) \tag{11}$$

subject to the constraint that the variational formulation of the time-dependent problem (1) is being satisfied for u^{fine}, i.e. $\tilde{A}(u^{\text{fine}})(\varphi) = \tilde{F}(\varphi)$ for all test functions φ. Possible choices for the fine and the coarse solution could be $u^{\text{fine}} := u \in X(I, V(\Omega)), u^{\text{coarse}} := u_k \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V(\Omega))$ to control the error caused by the temporal discretization or $u^{\text{fine}} := u_k \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V(\Omega))$, $u^{\text{coarse}} := u_{kh} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h)$, with $V_h := V_h^s$ for the heat equation and $V_h := V_h^s = V_h^{s,u} \times V_h^{s,v}$ for the elasto-dynamics equation, to control the error caused by the spatial discretization. For more information on space-time error control, we refer the interested reader to [18,21,22] and for general information on spatial error control to [12–14,84]. As an extension, in this work we restrict ourselves to the control of the error introduced by reduced-order modeling and thus we consider the full-order-model (FOM) solution $u^{\text{fine}} := u_{kh}^{\text{FOM}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}})$ as the fine solution, and the reduced-order-model (ROM) solution $u^{\text{coarse}} := u_{kh}^{\text{ROM}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}})$ as the coarse solution, with $V_h^{\text{ROM}} \subset V_h^{\text{FOM}} =: V_h$. First efforts of incorporating the dual-weighted residual (DWR) method in reduced-order modeling have been undertaken by Meyer and Matthies [45], where after computing some snapshots and creating the reduced basis, they used the DWR error estimator to determine which basis vectors have the largest error contribution and only use them for the reduced-order model. This can be thought of as a goal-oriented adaptive coarsening of the reduced basis. In this work, we focus on another objective, namely the enrichment of the reduced basis depending on the temporal evolution of the quantities of interest. This can be thought of as a goal-oriented adaptive refinement[1] of the reduced basis, in which we propose to accurately and efficiently compute the solution over the whole temporal domain.

4.1. Space-time dual-weighted residual method

For the constrained optimization problem (11), we define the Lagrange functional for the fine problem as

$$\mathcal{L}_{\text{fine}} : X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}) \times X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}) \to \mathbb{R},$$

[1] In principle coarsening would also be possible, but is not the objective in this work. For coarsening, we would need to follow the work of Meyer and Matthies [45].

$$(u^{\text{fine}}, z^{\text{fine}}) \mapsto J(u^{\text{fine}}) - \tilde{A}(u^{\text{fine}})(z^{\text{fine}}) + \tilde{F}(z^{\text{fine}}),$$

and for the coarse problem as

$$\mathcal{L}_{\text{coarse}} : X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}}) \times X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}}) \to \mathbb{R},$$
$$(u^{\text{coarse}}, z^{\text{coarse}}) \mapsto J(u^{\text{coarse}}) - \tilde{A}(u^{\text{coarse}})(z^{\text{coarse}}) + \tilde{F}(z^{\text{coarse}}).$$

The stationary points $(u^{\text{fine}}, z^{\text{fine}})$ and $(u^{\text{coarse}}, z^{\text{coarse}})$ of the Lagrange functionals $\mathcal{L}_{\text{fine}}$ and $\mathcal{L}_{\text{coarse}}$ need to satisfy the Karush-Kuhn-Tucker first-order optimality conditions.

4.1.1. Primal problem

Firstly, these stationary points are solutions to the equations

$$\mathcal{L}'_{\text{fine},z}(u^{\text{fine}}, z^{\text{fine}})(\delta z^{\text{fine}}) = -\tilde{A}(u^{\text{fine}})(\delta z^{\text{fine}}) + \tilde{F}(\delta z^{\text{fine}}) = 0 \quad \forall \delta z^{\text{fine}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}),$$
$$\mathcal{L}'_{\text{coarse},z}(u^{\text{coarse}}, z^{\text{coarse}})(\delta z^{\text{coarse}}) = -\tilde{A}(u^{\text{coarse}})(\delta z^{\text{coarse}}) + \tilde{F}(\delta z^{\text{coarse}}) = 0 \quad \forall \delta z^{\text{coarse}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}}).$$

We call these equations the primal problems and their solutions u^{fine} and u^{coarse} the primal solutions. We observe that the primal solution can be obtained by solving the original problem, e.g. the heat or the elastodynamics equation, forward in time.

4.1.2. Adjoint problem

Secondly, the stationary points must also satisfy the equations

$$\mathcal{L}'_{\text{fine},u}(u^{\text{fine}}, z^{\text{fine}})(\delta u^{\text{fine}}) = J'_u(u^{\text{fine}})(\delta u^{\text{fine}}) - \tilde{A}'_u(u^{\text{fine}})(\delta u^{\text{fine}}, z^{\text{fine}}) = 0 \quad \forall \delta u^{\text{fine}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}),$$
$$\mathcal{L}'_{\text{coarse},u}(u^{\text{coarse}}, z^{\text{coarse}})(\delta u^{\text{coarse}}) = J'_u(u^{\text{coarse}})(\delta u^{\text{coarse}}) - \tilde{A}'_u(u^{\text{coarse}})(\delta u^{\text{coarse}}, z^{\text{coarse}}) = 0 \quad \forall \delta u^{\text{coarse}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}}).$$

These equations are called the adjoint or dual problems and their solutions z^{fine} and z^{coarse} are the adjoint solutions. Hence, to obtain the adjoint solution, we need to solve an additional equation, the adjoint problem

$$\tilde{A}'_u(u)(\delta u, z) = J'_u(u)(\delta u). \tag{12}$$

Note that even for nonlinear PDEs and goal functionals the adjoint problem is linear since the semi-linear form in the variational formulation of the PDE is linear in the test functions, however the primal solution enters as it is well-known [13].

Remark 4.1. For linear PDEs, like the heat or the elastodynamics equation, the left-hand side of the adjoint problem (12) simplifies to

$$\tilde{A}'_u(u)(\delta u, z) = \tilde{A}(\delta u)(z).$$

For linear goal functionals, like the mean-value functional, the right-hand side of the adjoint problem (12) reduces to

$$J'_u(u)(\delta u) = J(\delta u).$$

In particular for a linear problem, i.e. linear PDE and goal functional, we have the adjoint problem

$$\tilde{A}(\delta u)(z) = J(\delta u), \tag{13}$$

which does not depend on the primal solution u anymore.

By Remark (4.1), the adjoint problem for the heat equation reads

$$\tilde{A}(\delta u)(z) = J'_u(u)(\delta u)$$
$$\Leftrightarrow \sum_{m=1}^{M} \int_{I_m} (\partial_t \delta u, z) + (\nabla_x \delta u, \nabla_x z)\, \mathrm{d}t + \sum_{m=1}^{M-1} ([\delta u]_m, z_m^+) + (\delta u_0^+, z_0^+) = J'_u(u)(\delta u).$$

We now use integration by parts in time to move the time derivative from the test function δu to the adjoint solution z and we get

$$\sum_{m=1}^{M} \int_{I_m} (\delta u, -\partial_t z) + (\nabla_x \delta u, \nabla_x z)\, \mathrm{d}t - \sum_{m=1}^{M-1} (\delta u_m^-, [z]_m) + (\delta u_M^-, z_M^-) = J'_u(u)(\delta u).$$

For the elastodynamics equation the adjoint problem can be derived in a similar fashion as

$$\sum_{m=1}^{M} \int_{I_m} (\delta v, -\partial_t z^u) + (\sigma(\delta u), \nabla_x z^u) + (\delta u, -\partial_t z^v) - (\delta v, z^v)\, \mathrm{d}t$$

$$- \sum_{m=1}^{M-1} \left((\delta v_m^-, [z^u]_m) + (\delta u_m^-, [z^v]_m) \right) + (\delta v_M^-, z_M^{u,-}) + (\delta u_M^-, z_M^{v,-}) = J'_U(U)(\delta U).$$

We notice that both adjoint problems now run backward in time.

4.1.3. Error identity and temporal localization for linear problems

For the sake of simplicity, we assume that we are dealing with a linear PDE and goal functional. Then we have the error identity

$$J(u^{\text{fine}}) - J(u^{\text{coarse}}) = -\tilde{A}(u^{\text{coarse}})(z^{\text{fine}}) + \tilde{F}(z^{\text{fine}}) =: \eta. \tag{14}$$

The proof relies on both the linearity of the goal functional and the PDE, and the definition of the adjoint and primal problems:

$$J(u^{\text{fine}}) - J(u^{\text{coarse}}) = J(u^{\text{fine}} - u^{\text{coarse}}) = \tilde{A}(u^{\text{fine}} - u^{\text{coarse}})(z^{\text{fine}}) = -\tilde{A}(u^{\text{coarse}})(z^{\text{fine}}) + \tilde{F}(z^{\text{fine}}).$$

In the DWR literature, first, for obtaining spatial and temporal discretization error control, due to Galerkin orthogonality the discrete solutions may be inserted, and then estimates by interpolation estimates. Next, we remark that this kind of error identity (14) would be useless, because for most applications z^{fine} is the analytical solution which is not known a priori and replacing it by z^{coarse} yields bad error estimates, even identical to zero results if z^{coarse} is in the discrete function space corresponding to z^{fine}. Thus, for FEM discretization error control the dual weights $z^{\text{fine}} - z^{\text{coarse}}$ are being used, often as previously written in terms of interpolation error estimates, which can be approximated by post-processing of the dual solution. However, in our case $z^{\text{fine}} := z_{kh}^{\text{FOM}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}})$ is the full-order-model dual solution, which is computable but comes with an expense. Moreover, in our numerical experiments we will observe that using a reduced-order-model dual solution $z^{\text{coarse}} := z_{kh}^{\text{ROM}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, \tilde{V}_h^{\text{ROM}})$ still produces excellent error estimates for our problems if the dual reduced basis is sufficiently large. We point out that the reduced dual spatial function space $\tilde{V}_h^{\text{ROM}}$ needs to differ from the reduced primal spatial function space V_h^{ROM} if we want to capture the dynamics of the dual problem and want to have a non-zero error estimator.

To localize the error in time, we just need to assemble the primal residual (14) slabwise. In particular, to localize the error to each time interval I_m, we simply need to assemble the primal residual on each time interval separately. More concretely, for the heat equation the error on the time interval can be computed from the primal linear equation system, the coarse primal solution and the fine dual solution by

$$\eta|_{I_m} = \sum_{i=1}^{\#\text{DoFs}(I_m)} \left\{ (Z_m^{\text{coarse}})^T \left(-AU_m^{\text{coarse}} + F_m - BU_{m-1}^{\text{coarse}} \right) \right\}_i. \tag{15}$$

The error estimator on the time interval I_m for elastodynamics can be derived analogously by using the linear system (A.1) of the primal problem.

To test whether we need to use the fine dual solution for our error estimates or whether we can replace it with a coarse dual solution, we use the effectivity index as a measure of the quality of our error estimator. The effectivity index is the ratio of the estimated and the true errors, i.e.,

$$\text{I}_{\text{eff}} := \left| \frac{\eta}{J(u^{\text{fine}}) - J(u^{\text{coarse}})} \right|. \tag{16}$$

We desire $\text{I}_{\text{eff}} \approx 1$, since then the error estimator can reliably predict the reduced-order-modeling error and we also observe this in the numerical tests in Section 5.

4.1.4. Space-time dual-weighted residual method for nonlinear problems

For nonlinear problems, like the heat equation with nonlinear goal functional in Section 5.2, we do not have an error identity anymore as in (14) for the linear case. Based on the proof in [13][Proposition 2.3], we have the following error representation formula.

Theorem 4.2 *(Error representation for nonlinear problems).*

$$J(u^{fine}) - J(u^{coarse}) = -\tilde{A}(u^{coarse})(z^{fine}) + \tilde{F}(z^{fine}) + R,$$

with the quadratic remainder term

$$R = \int_0^1 \Big[\tilde{A}''_{uu}(u^{coarse} + s(u^{fine} - u^{coarse}))(u^{fine} - u^{coarse}, u^{fine} - u^{coarse}, z^{fine}) - J''_{uu}(u^{coarse} + s(u^{fine} - u^{coarse}))(u^{fine} - u^{coarse}, u^{fine} - u^{coarse}) \Big] \cdot s \, \mathrm{d}s.$$

Proof. In the following we will show that $R = J(u^{\text{fine}}) - J(u^{\text{coarse}}) + \tilde{A}(u^{\text{coarse}})(z^{\text{fine}}) - \tilde{F}(z^{\text{fine}})$ holds. For abbreviation, we use the notation $u := u^{\text{fine}}$, $\tilde{u} := u^{\text{coarse}}$ and $z := z^{\text{fine}}$. Then, using integration by parts we get

$$R = \int_0^1 \left[\tilde{A}''_{uu}(\tilde{u}+s(u-\tilde{u}))(u-\tilde{u},u-\tilde{u},z) - J''_{uu}(\tilde{u}+s(u-\tilde{u}))(u-\tilde{u},u-\tilde{u})\right]\cdot s \,\mathrm{d}s$$

$$= -\int_0^1 \left[\tilde{A}'_{u}(\tilde{u}+s(u-\tilde{u}))(u-\tilde{u},z) - J'_{u}(\tilde{u}+s(u-\tilde{u}))(u-\tilde{u})\right]\cdot 1 \,\mathrm{d}s + \left[\tilde{A}'_{u}(u)(u-\tilde{u},z) - J'_{u}(u)(u-\tilde{u})\right]\cdot 1 - 0.$$

We observe that $\tilde{A}'_u(u)(u-\tilde{u},z) - \tilde{J}'_u(u)(u-\tilde{u}) = 0$, since $z := z^{\text{fine}}$ is the solution of the fine dual problem. Thus, by the fundamental theorem of calculus and $\tilde{A}(u)(z) = \tilde{F}(z)$, we have

$$R = -\left[\tilde{A}(u)(z) - J(u) - \tilde{A}(\tilde{u})(z) + J(\tilde{u})\right] = J(u) - J(\tilde{u}) + \tilde{A}(\tilde{u})(z) - \tilde{F}(z).$$

This completes the proof. □

To make the error estimator computable, we neglect the quadratic remainder term and arrive at the same primal error estimator (14) as for linear problems

$$\eta := -\tilde{A}(u^{\text{coarse}})(z^{\text{fine}}) + \tilde{F}(z^{\text{fine}}).$$

Similarly as before, we replace the full-order dual solution z^{fine} in the error estimator with a reduced-order-model dual solution z^{coarse}. Note that due to these approximations, the effectivity index for nonlinear problems is expected not to be close to 1. Clearly, for highly nonlinear problems (e.g., quasi-linear or fully nonlinear) and nonlinear goal functionals, both estimator parts are necessary as demonstrated in [85][Figure 4] and [86][Sec. 6.5]. However, in our numerical tests, we see that the estimated error still yields a reasonable approximation to the true error.

4.2. Error estimator based ROM updates

In this section, we present our novel approach of a goal-oriented incremental reduced-order model. In the MORe DWR method, we marry a reduced-order model with a DWR-based error estimator and an incremental version of the POD algorithm. The MORe DWR method addresses the problems that occur when a reduced-order model has to deal with solution behavior that is not already captured and incorporated during basis generation. In general, this yields an increasing error between full- and reduced-order solutions. Thus, the presented approach aims to detect changes in solution behavior, or more precisely, differences in the evaluated quantities of interest by means of the full or reduced model during the temporal evolution. If the error increases to intolerable heights, the method allows an adaptive on-the-fly basis enrichment with snapshots of the new behavior. Hence, the reduced model can be incrementally modified until the error is sufficiently small.

In more detail, we rely on the space-time reduced-order model presented in Section 3.2 and apply our findings on error control of Section 4.1. The use of an error estimate rather than an analytical error bound entails practical advantages since its application is more versatile and we can use the method even if no error bounds are known. Further, an incremental basis generation is mandatory for the method to reduce computational operations and thus to be fast. The incremental SVD satisfies these requirements and allows an update only requiring the prior SVD and the new snapshots. The incremental SVD is presented in Section 4.2.1. In this context, we also introduce the incremental POD as a trimmed version of the incremental SVD. Subsequently, the overall MORe DWR framework is depicted in Section 4.2.2. Here, all the ingredients are assembled and the final algorithm is presented.

In summary, our novel approach avoids a computationally heavy offline phase and directly solves the reduced model. Full-order solves are only required for the basis enrichment and are held to a minimum. Moreover, the reduced evaluation of the quantity of interest can be error-controlled.

4.2.1. Incremental proper orthogonal decomposition

This section aims to derive an algorithm that updates an already existing truncated SVD (tSVD) or solely its left-singular (POD) vectors according to modifications of the snapshot matrix without recomputing the whole tSVD or requiring access to the snapshot matrix. This methodology can then be used to update the POD incrementally by appending additional snapshots to the snapshot matrix. For this purpose, we rely on the general approach of an additive rank-b modification of the SVD, mainly developed by [50,51] and applied to the model-order reduction of fluid flows in [52]. Although this approach provides a variety of possible modifications, e.g. resizing of the matrix, modification of individual values, or exchanging rows and columns, we are merely interested in the updates of columns, i.e. adding columns to the matrix, and thus restrict the proceeding on this. The following steps are based on [52][Section 2.2].

We start with a given snapshot matrix $Y \in \mathbb{R}^{\#\text{DoFs}(\mathcal{T}_h)\times\tilde{m}}$ that includes $\tilde{m} > 0$ snapshots. Usually, $\tilde{m}$ is equal or connected to the number of already computed time steps. Further, we have the rank-N tSVD $\Psi\Sigma\Phi^{\mathrm{T}}$ of the matrix Y. Additionally, let $b \in \mathbb{N}$ newly computed snapshots $\{U_1, \dots, U_b\}$ be stored in the bunch matrix

$$B = \begin{bmatrix} u_1 & \dots & u_b \end{bmatrix} \in \mathbb{R}^{\#\text{DoFs}(\mathcal{T}_h)\times b}. \tag{17}$$

We now aim to compute the tSVD that is updated by the information contained in the bunch matrix B according to

$$\tilde{\Psi}\tilde{\Sigma}\tilde{\Phi}^T = \tilde{Y} = \begin{bmatrix} Y & B \end{bmatrix}$$

without explicitly recomputing Y or $\tilde{Y}$ due to performance and memory reasons which was the original motivation of Brand's work on the incremental SVD, cf. [50,51].

Therefore, we write the column update as an additive operation given as

$$\begin{bmatrix} Y & B \end{bmatrix} = \begin{bmatrix} Y & 0_{\#\text{DoFs}(\mathcal{T}_h)\times b} \end{bmatrix} + B \begin{bmatrix} 0_{b\times\tilde{m}} & I_{b\times b} \end{bmatrix} \tag{18}$$

to apply the additive rank-b modification to the SVD according to [52] and obtain the rank-$\tilde{N}$ tSVD of $\tilde{Y}$ with $\tilde{N} \leq N + b$ and

$$\tilde{\Phi} = \begin{bmatrix} \Phi & 0 \\ 0 & I \end{bmatrix} \Phi'(:, 1:\tilde{N}), \tag{19}$$

$$\tilde{\Sigma} = \Sigma'(1:\tilde{N}, 1:\tilde{N}), \tag{20}$$

$$\tilde{\Psi} = \begin{bmatrix} \Psi & Q_{\mathrm{B}} \end{bmatrix} \Psi'(:, 1:\tilde{N}), \tag{21}$$

where $F = \Psi'\Sigma'\Phi'^T \in \mathbb{R}^{(N+b)\times(N+b)}$ is the SVD of

$$F = \begin{bmatrix} \Sigma & \Psi^{\mathrm{T}} B \\ 0 & R_{\mathrm{B}} \end{bmatrix} \tag{22}$$

and $Q_B \in \mathbb{R}^{\#\text{DoFs}(\mathcal{T}_h)\times b}$ and $R_B \in \mathbb{R}^{b\times b}$ are given by the QR decomposition

$$Q_{\mathrm{B}} R_{\mathrm{B}} = (I - \Psi\Psi^{\mathrm{T}})B \in \mathbb{R}^{\#\text{DoFs}(\mathcal{T}_h)\times b}. \tag{23}$$

For the POD basis update, we identify Ψ and $\tilde{\Psi}$ with the previous and updated versions of the reduced basis matrix Ψ_N including the POD vectors, respectively. We also neglect the update of the right-singular vectors in (19), since it does not provide any additional benefit apart from extra computational effort for the reduced-order model, cf. Theorem 3.1. The singular values are considered for the rank determination but they come within zero computational cost. In conclusion, (20)-(22) serve as the basis for the on-the-fly or incrementally computed POD (iPOD) in this paper.

An additional technical observation: For bunch matrices with small column rank b, the iPOD algorithm is invoked frequently, and algebraic subspace rotations possibly involved do not preserve orthogonality, cf. [51,87–90]. Hence, a numerically induced loss of orthogonality of the POD basis vectors can occur. In order to deal with this problem an additional orthonormalization of $\begin{bmatrix} \Psi & Q_{\mathrm{B}} \end{bmatrix}$ is recommended. Algorithm 1 drafts the implementation of an incremental POD update. Here, Ψ_N is the reduced basis matrix of (6) with its singular values $\Sigma = [\sigma_1, \ldots, \sigma_N] \in \mathbb{R}^N$, and $\tilde{\Psi}_N$ and $\tilde{\Sigma} = [\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{\tilde{N}}]$ denote their incremental updates.

In addition, the bunch matrix B introduced in (17) including b snapshots is used as an input. The information content captured by the reduced basis is determined by the energy threshold ε. We provide a prototypical implementation of the incremental POD in Python under https://github.com/Hendrik240298/Incremental_POD.

Algorithm 1 Incremental POD update.

Input: Reduced basis matrix $\Psi_N \in \mathbb{R}^{\#\text{DoFs}(\mathcal{T}_h)\times N}$, singular value vector $\Sigma = [\sigma_1, \ldots, \sigma_N] \in \mathbb{R}^N$, bunch matrix $B \in \mathbb{R}^{\#\text{DoFs}(\mathcal{T}_h)\times b}$, and energy threshold $\varepsilon \in [0, 1]$.
Output: Reduced basis matrix $\tilde{\Psi}_N \in \mathbb{R}^{\#\text{DoFs}(\mathcal{T}_h)\times \tilde{N}}$, singular value vector $\tilde{\Sigma} = [\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{\tilde{N}}]$

$H = \Psi_N^T B$
$P = B - \Psi_N H$
$[Q_P,\ R_P] = \mathrm{QR}(P)$
$Q = [\Psi_N\ Q_P]$
$F = \begin{bmatrix} \mathrm{diag}(\Sigma) & H \\ 0 & R_P \end{bmatrix}$
if Q not orthogonal **then**
    $[Q,\ R] = \mathrm{QR}(Q)$
    $F = RF$
$[\Psi', \Sigma'] = \mathrm{SVD}(F)$
$\tilde{N} = \min\{N \in \mathbb{N} \mid \varepsilon(N) \geq \varepsilon,\ \ 1 \leq N \leq d\}$
$\tilde{\Sigma} = \mathrm{diag}(\Sigma')(1:\tilde{N})$
$\tilde{\Psi}_N = Q\Psi'(:, 1:\tilde{N})$

Note that checking if the orthogonality is preserved can be computationally expensive. Thus, we resort to a heuristic approach by sole validation if the first and last columns of a matrix are orthogonal.

4.2.2. Goal-oriented error-controlled incremental ROM

In this section, we assemble the space-time ROM presented in Section 3.2 and the incremental POD of Section 4.2.1 with the findings on goal-oriented error control of Section 4.1. This yields an adaptive goal-oriented incremental reduced-order model. Firstly, next to the slab definition we introduce the parent-slab notion as a further decomposition of the space-time domain. A parent-slab unifies several slabs that are consecutive in time and is defined as

$$P_k^r = \{ S_l^n \mid l \geq k \ \wedge\ n \leq r \}.$$

Now, our approach is designed to work without any prior knowledge or exploration of the solution manifold while also attempting to minimize the full-order operations. Thus, we aim to solve the reduced-order model parent-slab wise and –if necessary– adaptively enrich the reduced basis by means of the iPOD with full-order solutions of the parent-slab until the reduced basis is good enough to meet a given estimated error tolerance for the chosen goal functional. For this, we identify the fine and coarse solutions introduced in the DWR method with the finite element and reduced basis solutions, respectively, and estimate the error on each slab of the parent-slab. The full-order solution used for the basis enrichment is computed on the slab where the error is the largest. We remark that both the primal and dual full-order solutions are computed on this slab and are used to enrich the primal and dual bases. So, for each enrichment two full-order solves are conducted. After having finished this iterative process on a parent-slab, the obtained basis is transferred to the proceeding parent-slab and is used as a starting point to solve the reduced-order model where the whole procedure is repeated. So if the solution behavior on the next parent-slab only differs slightly from the already observed behavior, the reduced basis at hand should be able to reproduce most of the behavior. Thus, few basis updates would be sufficient such that a fast computation of the reduced solution can be expected. However, if the solution behavior changes drastically the error estimate will detect this and further refinements of the basis will be conducted to ensure that the solution meets the error tolerance. We emphasize that the dual problem is solved on each parent slab independently, without using the dual solutions of the following slabs as an initial condition since these are not computed yet. Instead, on each parent slab the dual problem is solved with a zero initial condition. This division into subproblems does not fully equal the system described in Section 4.1.2 but yields an approach without needing to solve the dual problem over the whole time domain, thus enabling an efficient solution process. Nevertheless, there may be certain problems or goal functionals for which this shortcut does not yield good results. Especially, for end-time functionals, i.e. $J_1 = 0$ and $J_2 \neq 0$, the dual solution would be zero on all but the last parent slab. We try to possibly counteract the induced error of employing a surrogate for the exact dual problem by adding a validation loop at the end of the simulation. There, the ROM and error estimation are conducted on a single parent slab, which translates to the computation of the dual problem over the full temporal domain as stated in Section 4.1.2.

We observe that this procedure is perfectly compatible with the adaptive basis selection based on DWR estimates presented by Meyer and Matthies in [45] to reduce the dimension of the reduced space. Thus, if incorporated it would be possible to either enrich or delude the reduced basis adjusted to the problem statement.

The resulting approach is outlined in detail in Algorithm 2 and illustrated in Fig. 2 for two consecutive parent slabs. For the sake of simplicity, we decompose the space-time cylinder in K parent-slabs of fixed length L and enumerate them with respect to time, viz. $P_1, P_2, \dots, P_K$. In order to identify the affiliation of a slab to a parent-slab P_k, the slabs it contains are denoted by $S^1_{P_k}, S^2_{P_k}, \dots, S^L_{P_k}$ with $1 \leq k \leq K$. The discretized primal systems for each slab $S^j_{P_k}$ are expressed in (3) and (8) for the full- and reduced-order models, respectively. For the dual problem

$$A' Z_{S^l_{P_k}} = J_{S^l_{P_k}} \quad \text{and} \tag{24}$$

$$A'_{N_d} Z_{N_d, S^l_{P_k}} = J_{N_d, S^l_{P_k}} \tag{25}$$

denote the discretized full- and reduced-order systems of the adjoint problem (12). To distinguish the components of the primal and dual reduced-order model they got the labels p and d, respectively. Further, the evaluation of the error estimator (15) on slab $S^l_{P_k}$ is given by $\eta_{N, S^l_{P_k}}\left(U_{N_p, S^l_{P_k}}, Z_{N_d, S^l_{P_k}}\right)$. Note that the reduced primal and dual solutions are deployed to enable an evaluation independent of the full-order system size and thus a fast error evaluation. In addition, the incremental POD (Algorithm 1) is referred to by the abbreviation iPOD with the reduced basis, new snapshots bundled in the snapshot matrix, and singular values as input and the new POD basis as output. Lastly, in Algorithm 2 the incremental ROM has both a primal and a dual reduced basis as input in the form of the reduced basis matrix. Depending on the situation, these inputs may be different. If no prior information is available, the bases can be initialized by computing the primal and dual solution snapshots on the first single slab in time. However, if prior reduced bases are available, e.g. from a previous simulation on a different parameter configuration, these bases can be used as an initial guess and then be altered by the goal-oriented adaptation performed in the algorithm. This indicates the applicability of the MORe DWR approach to reduced-order modeling of parameterized problems. Here, the method can be an efficient substitute for the full-order model in the offline phase and build a reduced basis tailored to a quantity of interest with a minimum of full-order model solves.

Furthermore, we note that similar to the mere approximation error of the POD in (4) the physical interpretation of the error estimate is not intuitive. Therefore, we are considering a relative measurement of the approximation quality. However, the full-order solutions are not available for a normalization of the error so that we resort to

$$J\left(U_{S^l_{P_k}}\right) \approx J\left(U_{N_p, S^l_{P_k}}\right) + \eta_{N, S^l_{P_k}}.$$

This results in the relative error estimator $\eta^{rel}_{N, S^l_{P_k}}$ on slab $S^l_{P_k}$ defined by

$$\eta^{rel}_{N, S^l_{P_k}} = \frac{\eta_{N, S^l_{P_k}}}{J\left(U_{S^l_{P_k}}\right)} \approx \frac{\eta_{N, S^l_{P_k}}}{J\left(U_{N_p, S^l_{P_k}}\right) + \eta_{N, S^l_{P_k}}}. \tag{26}$$

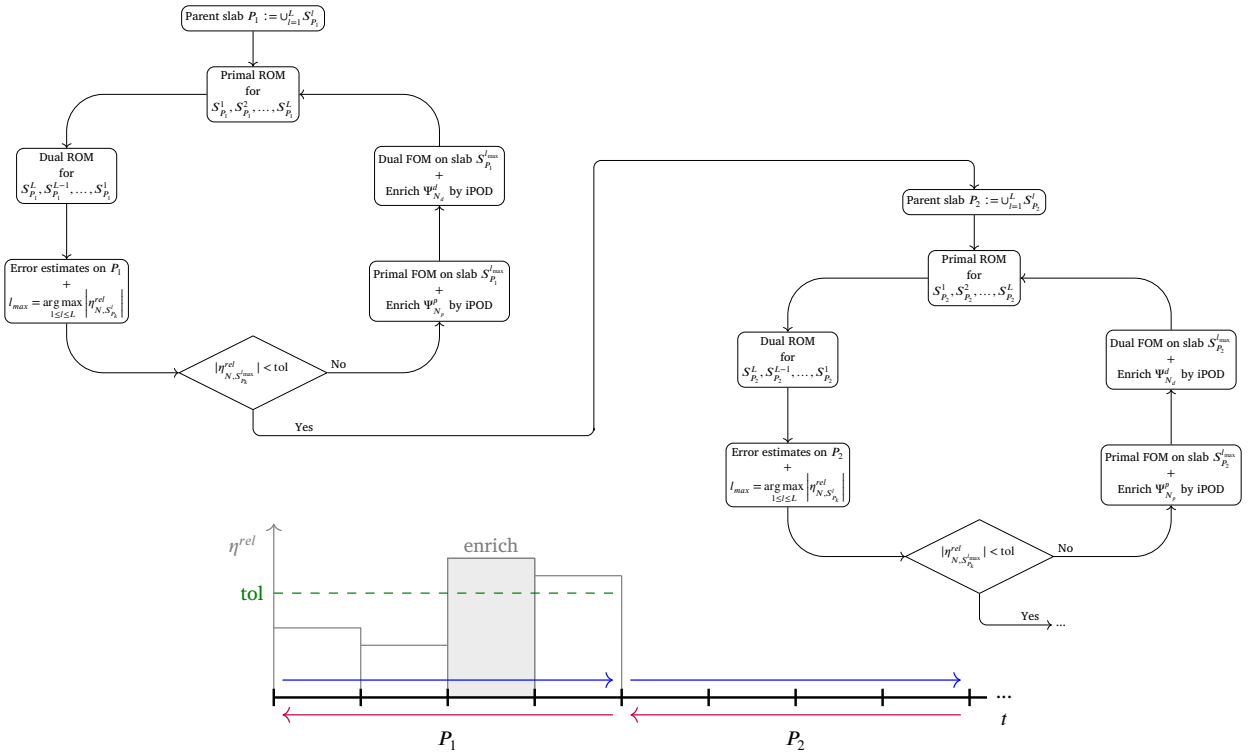


Fig. 2. MORe DWR algorithm illustrated for two consecutive parent slabs.

Algorithm 2 MORe DWR algorithm.

Input: Initial condition $U_0 := U(t_0)$, primal and dual reduced basis matrices $\Psi^p_{N_p}$ and $\Psi^d_{N_d}$, energy threshold $\varepsilon \in [0,1]$ and error tolerance $\text{tol} > 0$.

Output: Primal and dual reduced basis matrices $\Psi^p_{N_p}$ and $\Psi^d_{N_d}$ and reduced primal solutions U_{N_p, I_m} for all $1 \le m \le M$.

for $k = 1, 2, \ldots, K$ **do** ▷ Loop over parent slabs
  while $\eta_{max} > tol$ **do**
    for $l = 1, 2, \ldots, L$ **do** ▷ Primal ROM on k-th parent slab
      Solve reduced primal system (8): $A_{N_p} U_{N_p, S^l_{P_k}} = F_{N_p, S^l_{P_k}}$
    for $l = L, L-1, \ldots, 1$ **do** ▷ Dual ROM on k-th parent slab
      Solve reduced dual system (25): $A'_{N_d} Z_{N_d, S^l_{P_k}} = J_{N_d, S^l_{P_k}}$
    for $l = 1, 2, \ldots, L$ **do** ▷ Error estimates on k-th parent slab
      Compute error estimate: $\eta^{rel}_{N, S^l_{P_k}} \left(U_{N_p, S^l_{P_k}}, Z_{N_d, S^l_{P_k}} \right)$
    $\eta_{max} = \max_{1 \le l \le L} \left| \eta^{rel}_{N, S^l_{P_k}} \right|$
    if $\eta_{max} > tol$ **then**
      $l_{max} = \arg\max_{1 \le l \le L} \left| \eta^{rel}_{N, S^l_{P_k}} \right|$
      Solve primal full-order system (3): $A U_{S^{l_{max}}_{P_k}} = F_{S^{l_{max}}_{P_k}}$
      Update primal reduced basis: $\Psi^p_{N_p} = \text{iPOD}(\Psi^p_{N_p}, \Sigma_{N_p}, [U_{S^{l_{max}}_{P_k}}(t_1), \ldots, U_{S^{l_{max}}_{P_k}}(t_{r+1})], \varepsilon)$
      Solve dual full-order system (24): $A' Z_{S^{l_{max}}_{P_k}} = J_{S^{l_{max}}_{P_k}}$
      Update dual reduced basis: $\Psi^d_{N_d} = \text{iPOD}(\Psi^d_{N_d}, \Sigma_{N_d}, [Z_{S^{l_{max}}_{P_k}}(t_1), \ldots, Z_{S^{l_{max}}_{P_k}}(t_{r+1})], \varepsilon)$
      Update reduced system components and error estimator w.r.t. (9)

——— Validation loop ——— ▷ This is an optional validation of the model
for $k = 1, 2, \ldots, K$ **do** ▷ Primal ROM on whole temporal domain
  for $l = 1, 2, \ldots, L$ **do**
    Solve primal reduced system: $A_{N_p} U_{N_p, S^l_{P_k}} = F_{N_p, S^l_{P_k}}$
for $k = K, K-1, \ldots, 1$ **do** ▷ Dual ROM on whole temporal domain
  for $l = L, L-1, \ldots, 1$ **do**
    Solve dual reduced system: $A'_{N_d} Z_{N_d, S^l_{P_k}} = J_{N_d, S^l_{P_k}}$
for $k = 1, 2, \ldots, K$ **do** ▷ Error estimates on whole temporal domain
  for $l = 1, 2, \ldots, L$ **do**
    Compute slab estimate: $\eta^{rel}_{N, S^l_{P_k}} (U_{N_p, S^l_{P_k}}, Z_{N_d, S^l_{P_k}})$

In addition to the previously mentioned steps, we add an optional validation loop whose purpose depends on the application. Specifically, it consists in recomputing the whole reduced solutions with the final reduced basis and evaluating its error again. If the generated reduced basis is meant to be reused, the additional validation of its accuracy ensures that the reduced basis is well suited to approximate the solution for the whole time domain. This is mainly the case in multi-query evaluations, e.g. in an optimization process or if the MORe DWR method is used for manifold exploration. However, if the only purpose is a one-time evaluation of a quantity of interest, the validation can be ignored for performance reasons.

5. Numerical tests

In order to demonstrate our methodology, we perform numerical tests on three different problem configurations. For the first two numerical tests, we perform computations for the heat equation in 1+1D and 2+1D. For the former, we use a linear goal functional and for the latter, we use a nonlinear goal functional. To demonstrate the flexibility of our temporal discretization, we use Gauss-Legendre quadrature points in time for the heat equation and a dG(1) time discretization. As the third problem configuration, we consider a 3+1D cantilever beam as a benchmark problem from elastodynamics. For this problem, we use Gauss-Lobatto quadrature points in time, which are the support points for conventional time-stepping schemes, and we use a dG(2) time discretization.

All our computations have been performed on a personal computer with an Intel i5-7600K CPU @ 3.80GHz × 4 and 16GB of RAM. The space-time FEM codes have been written in deal.II [56,57] and the reduced-order modeling has been performed with NumPy [91] and SciPy [92]. The data between the codes is exchanged via the hard disk.

5.1. 1+1D heat equation

For our first numerical test, we construct a 1+1D heat equation problem; see Formulation 2.1. We consider the spatial domain $\Omega = (0,1)$ and the temporal domain $I = (0,4)$. We use a single moving heat source that changes its temperature after each second and moves through the spatial domain with a heating interval width of 0.1 from $x = 0.1$ to $x = 0.9$ and then back to $x = 0.1$. For this, we use the right-hand side function

$$f(t,x) := \begin{cases} 0.2 & t \in (0,1),\ -0.05 \le x - 0.4t - 0.1 \le 0.05, \\ -0.5 & t \in (1,2),\ -0.05 \le x - 0.4t - 0.1 \le 0.05, \\ 1.0 & t \in (2,3),\ -0.05 \le x + 0.4(t-2) - 0.9 \le 0.05, \\ -0.75 & t \in (3,4),\ -0.05 \le x + 0.4(t-2) - 0.9 \le 0.05. \end{cases}$$

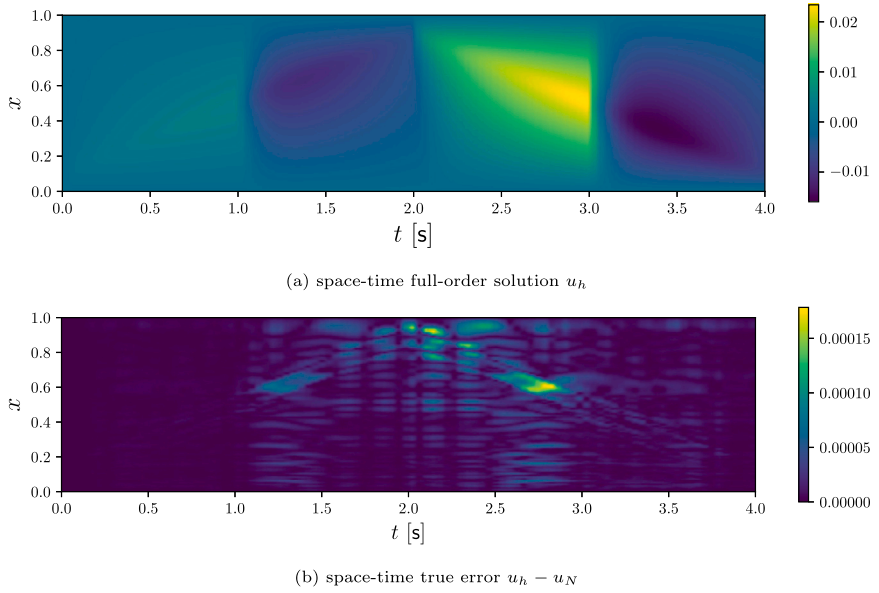
We use a zero initial condition, homogeneous Dirichlet boundary conditions, and the time-averaged mean value goal functional $J(u) := \frac{1}{4}\int_0^4 \int_0^{\frac{1}{2}} u(t,x)\,\mathrm{d}x\,\mathrm{d}t$. We point out that the goal functional does not have support on the entire spatial domain, but only on its lower half $(0, \frac{1}{2}) \subsetneq \Omega$.

For the reduced-order model, we choose that the primal and dual reduced bases have to preserve $\varepsilon = 1 - 10^{-8}$ of the information. As previously stated, we resort to the relative error estimate $\eta^{rel}_{N,S^l_{P_k}}$ developed in (26) and allow errors up to a tolerance of 1%. We consider this to be a reasonable tolerance for many applications. The full-order model is characterized by $n = 513$ and $q = 1,280$ DoFs in space and time, respectively. This gives us a total of $n \cdot q = 656,640$ space-time degrees of freedom. The resulting full-order system is solved in $0.27\,s$. Further, the temporal domain is split up into $M = 640$ time slabs. For the incremental ROM, we choose a total amount of $K = 32$ parent-slabs on which the slabs are evenly distributed, i.e. $L = 40$.

In Fig. 3, we display the full-order space-time solution u_h as well as the true error $u_h - u_N$ between the full-order space-time solution u_h and the reduced-order space-time solution u_N obtained using MORe DWR. Looking at the error, we observe that the reduced-order model becomes less accurate for $x \in (\frac{1}{2}, 1)$ than for $x \in (0, \frac{1}{2})$, which is the spatial support of the goal functional. This shows that our incremental POD is goal-oriented. In Fig. 4a, we compare the time trajectories of the goal functional restricted to each time slab for the full-order space-time solution u_h and the reduced-order space-time solution u_N. It illustrates that both trajectories are not distinguishable from each other indicating that the reduced-order model captures the temporal evolution of the quantity of interest accurately even with changing solution behavior. This good approximation quality can also be observed when regarding the time-averaged goal functional. We obtain $J(u_h) = 2.0607 \cdot 10^{-4}$ and $J(u_N) = 2.0585 \cdot 10^{-4}$ yielding a relative error of $\eta_{max} = 0.1095\%$. We compare the temporal error estimate with the exact temporal error on each slab in Fig. 4b. The general tendencies of both curves are similar. The exact error is on average more than one magnitude smaller than the error tolerance of 1% (indicated by a green dashed line). Furthermore, the error tolerance acts as a real error bound since it is never violated, despite the use of error estimates in the MORe DWR approach.

Table 1 gives an overview of simulation results for different error tolerances comprised between 0.1% and 10%. The listed characteristics are: the relative error, computational speedup, the total number of FOM solves, POD basis sizes for the primal and dual problem, prediction capability of error estimator, and the effectivity index from (16). Here, the number of FOM solves sums up all primal and dual solves and the basis sizes are shown in the pattern primal | dual. The prediction capability is visualized by means of a confusion matrix. The prediction on each slab is assigned to one of the four cases:

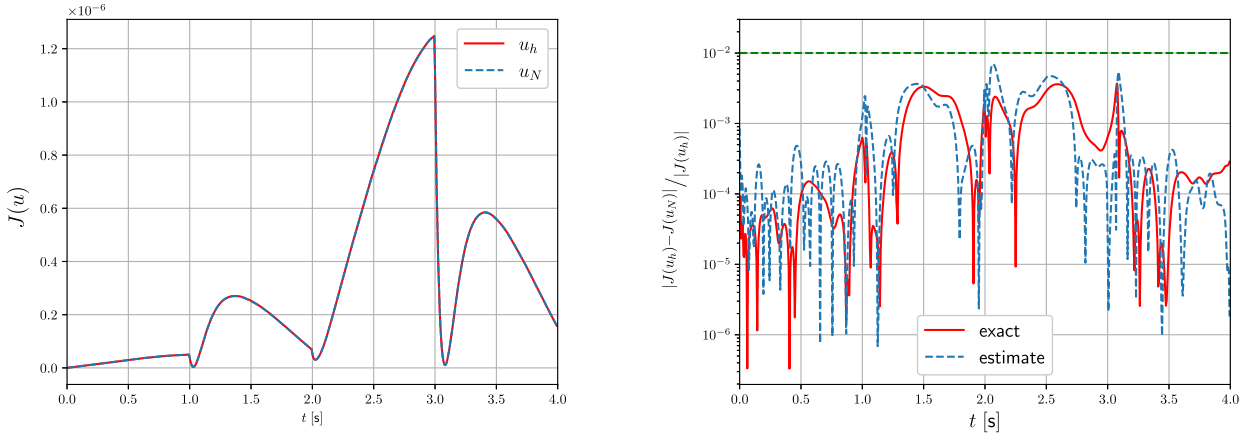
$$\text{error} > \text{tol} \wedge \text{estimate} < \text{tol} \quad | \quad \text{error} < \text{tol} \wedge \text{estimate} > \text{tol} \quad | \quad \text{error} > \text{tol} \wedge \text{estimate} > \text{tol} \quad | \quad \text{error} < \text{tol} \wedge \text{estimate} < \text{tol}.$$



(a) space-time full-order solution u_h

(b) space-time true error $u_h - u_N$

Fig. 3. Space-time solution and error for the 1+1D heat equation.



(a) (b)

Fig. 4. Temporal evolution of (a) the goal functional and (b) the time slab-wise error estimator compared to the true error for the 1+1D heat equation. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

We note that the four possible scenarios are sorted according to the severity of the consequences of their occurrence. So, the first two cases indicate mispredictions of the estimator. Here, the first case is the least desirable since then the error estimator underestimates the true error, which can lead to a too small reduced basis to meet the desired error tolerance. The second case is less fatal since then the true error is being overestimated by the error estimator, which can cause the reduced basis to be slightly larger than necessary. The last two cases are less harmful since the estimate correctly predicts the error. However, the third case is also not optimal, since it shows that after the incremental basis enrichment, in the validation loop, there are still slabs on which the error tolerance is being exceeded. Therefore, we expect that for an efficient method (almost) all slabs fall in the last category, where the error tolerance is being met and the error estimate is also below the tolerance.

We observe that with a rise in the tolerance the relative error as well as the speedup –compared to our highly resolved full-order model– increase. We remark that the actual computational speedups strongly depend on multiple variables such as the chosen time and space discretization of the full-order model, the implementation of the method and hardware specifications. Overall our method also benefits from more efficient full-order models but for coarse discretizations the speedup obtained by adaptive model order reduction will be less prominent or in the worst case not existent. Note that the relative error is at least one order of magnitude smaller than the tolerance, which aligns with the results of Fig. 4b. The difference in magnitude can be explained by the fact that the tolerance has to be met slabwise while the relative error is evaluated over the whole time domain. The speedup is explained by the decreasing amount of FOM solves and smaller POD bases for both the primal and dual problem w.r.t. the given tolerance.

Table 1
Performance of MORe DWR for the 1+1D heat equation depending on the tolerance in the goal functional.

Tolerance	Relative error	Speedup	FOM solves	Basis size	Prediction	Effectivity
0.1%	0.0001%	1.0203	48	29 | 26	0 | 2 | 0 | 638	1.0143
1%	0.0011%	1.7060	30	20 | 17	0 | 0 | 0 | 640	1.0121
2%	0.0015%	2.3779	22	17 | 13	0 | 1 | 0 | 639	1.0142
5%	0.0095%	2.6840	16	14 | 10	0 | 4 | 0 | 636	1.0288
10%	0.0089%	2.6919	16	15 | 10	0 | 0 | 0 | 640	1.0165

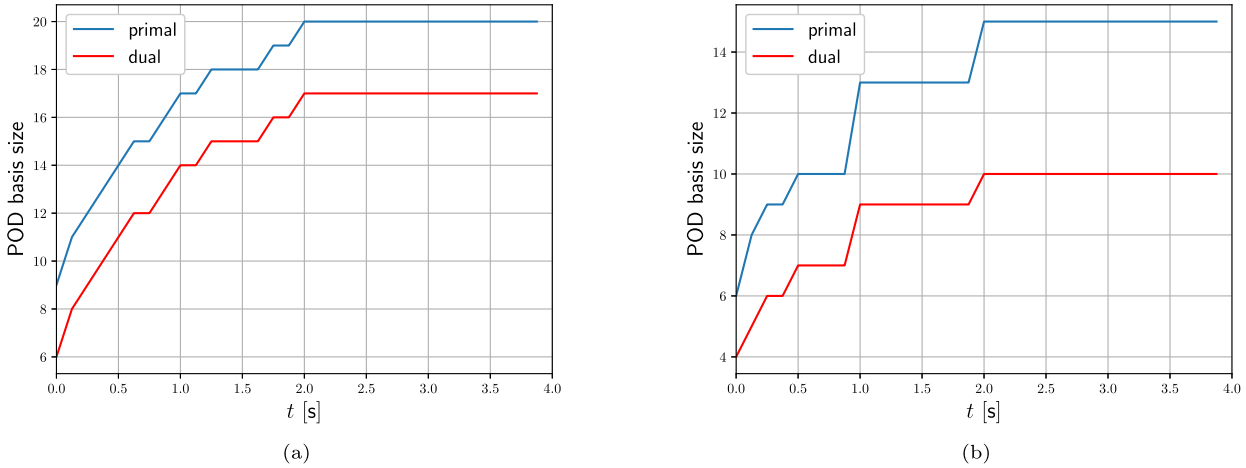


Fig. 5. Temporal evolution of the reduced basis size for a relative error tolerance of (a) 1% and (b) 10% for the 1+1D heat equation.

Furthermore, the estimator predicts the relationship of the error to the tolerance in over 99% of the cases right with the incorrect predictions being overestimations. Similarly, for the effectivity index, a slight worsening can be seen with rising tolerance since then replacing the full-order dual solution in the error estimator with the reduced-order dual solution introduces additional errors.

Finally, we demonstrate the incremental nature of our MORe DWR approach in Fig. 5. In this context, we illustrate the on-the-fly basis generation by plotting the primal and dual reduced basis size over the time domain and compare its evolution for the tolerances of 1% and 10%. The results indicate a steeper and more granular increase of both the primal and dual basis size if the tolerance is smaller. Nevertheless, we observe a steady basis size for all bases and tolerances after around 2 seconds. If we take the movement of the heat source into account, this is exactly the time the source needs to travel once through the spatial domain. Thus, after this, no new information is added to the system that would trigger a further basis enrichment.

5.2. 2+1D heat equation

In the second numerical experiment, we test MORe DWR on a 2+1D heat equation problem. We consider the spatial domain $\Omega = (0,1)^2$ and the temporal domain $I = (0,10)$. We create a moving heat source of oscillating temperature that rotates around the midpoint of the spatial domain Ω as shown in Fig. 6. For this, we use the right-hand side function

$$f(t,x) := \begin{cases} \sin(4\pi t) & \text{if } (x_1 - p_1)^2 + (x_2 - p_2)^2 < r^2, \\ 0 & \text{else,} \end{cases}$$

with $x = (x_1, x_2)$, midpoint $p = (p_1, p_2) = (\frac{1}{2} + \frac{1}{4}\cos(2\pi t), \frac{1}{2} + \frac{1}{4}\sin(2\pi t))$ and radius of the trajectory $r = 0.125$. In addition, a zero initial condition and homogeneous Dirichlet boundary conditions are applied. In contrast to the goal functional in Section 5.1, we test the method for a nonlinear goal functional $J(u) := \frac{1}{10}\int_0^{10}\int_\Omega u(t,x)^2 \,\mathrm{d}x\,\mathrm{d}t$.

For the reduced-order model, we choose that the primal and dual reduced bases have to preserve $\varepsilon = 1 - 10^{-8}$ of the information. Similar to the previous one-dimensional scenario, we resort to the relative error estimate $\eta^{rel}_{N,S^l_{P_k}}$ and allow errors up to a tolerance of 1%. The full-order model is characterized by $n = 4,225$ and $q = 4,096$ DoFs in space and time, respectively. This gives us a total of $n \cdot q = 17,305,600$ space-time degrees of freedom. The resulting full-order system is solved in 220 s. Further, the temporal domain is split up into $M = 2,048$ time slabs. Note that we chose this temporal discretization to ensure a sufficiently accurate full-order solution. However, it might be possible that a coarser temporal discretization would be enough. For the incremental ROM, we choose a total amount of $K = 128$ parent-slabs on which the slabs are evenly distributed, i.e. $L = 16$.

Firstly, in Fig. 7a we compare the time trajectories of the goal functional restricted to each time slab for the full-order space-time solution u_h and the reduced-order space-time solution u_N. It illustrates that both trajectories are not distinguishable from each

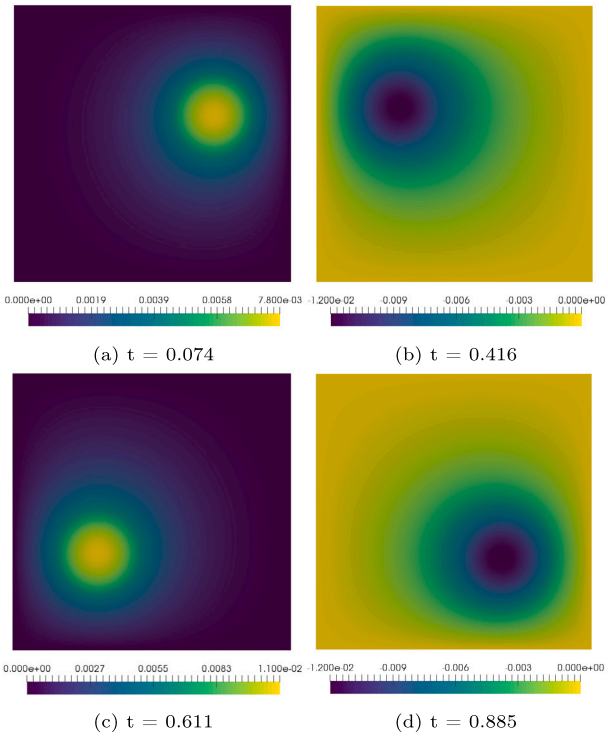


Fig. 6. Full-order solution snapshots for the 2+1D heat equation.

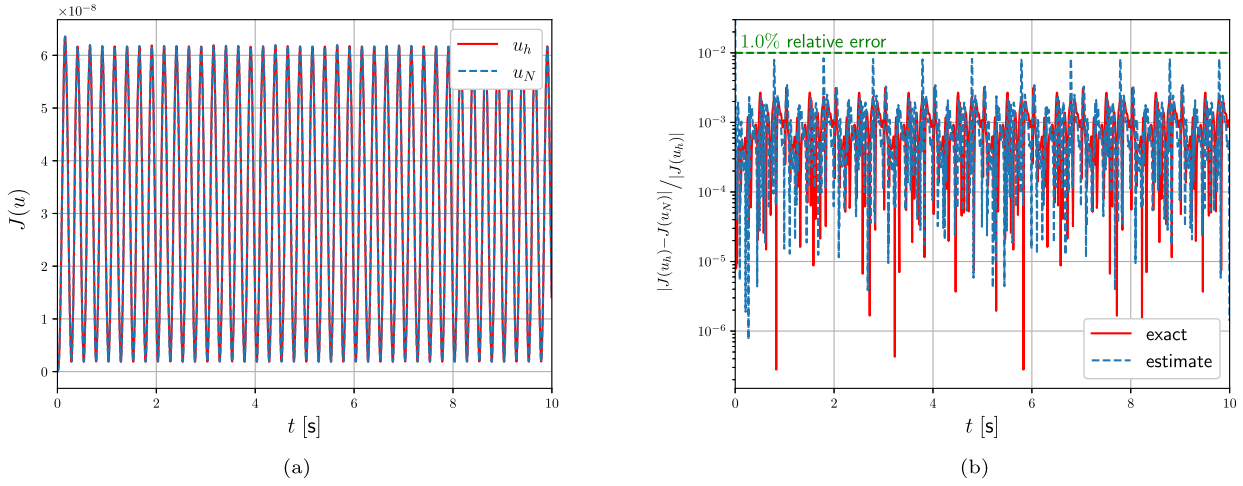


Fig. 7. Temporal evolution of (a) the goal functional and (b) the time interval-wise error estimator compared to the true error for the 2+1D heat equation.

other although the solution behavior is constantly changing. Furthermore, good approximation quality can also be observed when regarding the time-averaged goal functional. We obtain $J(u_h) = 6.4578 \cdot 10^{-5}$ and $J(u_N) = 6.4577 \cdot 10^{-5}$ yielding a relative error of $\eta_{max} = 0.0016\%$. This implies that the incremental ROM can replicate nonlinear goal functionals within a given tolerance.

In Fig. 7b, the exact temporal errors and their estimation on each slab are compared. Further, for illustration we indicate the error tolerance of 1% in this plot. The results show that both the exact and estimated errors meet the given error tolerance on all slabs. Overall, the estimate shows a similar trajectory to the exact error. However, we can observe spikes in the exact error that are not completely covered by the estimation. Nevertheless, these deflections remain without consequences. Table 2 presents simulation results for a range of error tolerances. The quantities we consider are the following: the relative error, computational speedup, the total number of FOM solves, POD basis sizes for the primal and dual problem, prediction capability of the error estimator, and the effectivity index. For definitions of these quantities, we refer to Section 5.1. We can observe that with a rise in the tolerance the relative error as well as the speedup increase. Again, the relative error is much smaller than the tolerance. Note in contrast to the 1D linear scenario the relaxation of the error tolerance has a greater impact on the speedup. This can be explained by the evolution of

Table 2
Incremental reduced-order modeling summary for the 2+1D heat equation depending on the tolerance in the goal functional.

Tolerance	Relative error	Speedup	FOM solves	Basis size	Prediction	Effectivity
0.1%	0.0019%	7.7	150	92 | 78	0 | 35 | 0 | 2013	0.7524
1%	0.0017%	27.5	80	55 | 44	0 | 1 | 0 | 2047	0.2771
2%	0.0628%	29.6	66	47 | 36	0 | 9 | 0 | 2039	3.9181
5%	0.9162%	44.8	44	33 | 25	0 | 1 | 0 | 2047	1.2254
10%	0.9243%	50.0	38	31 | 23	79 | 28 | 17 | 1924	1.5474

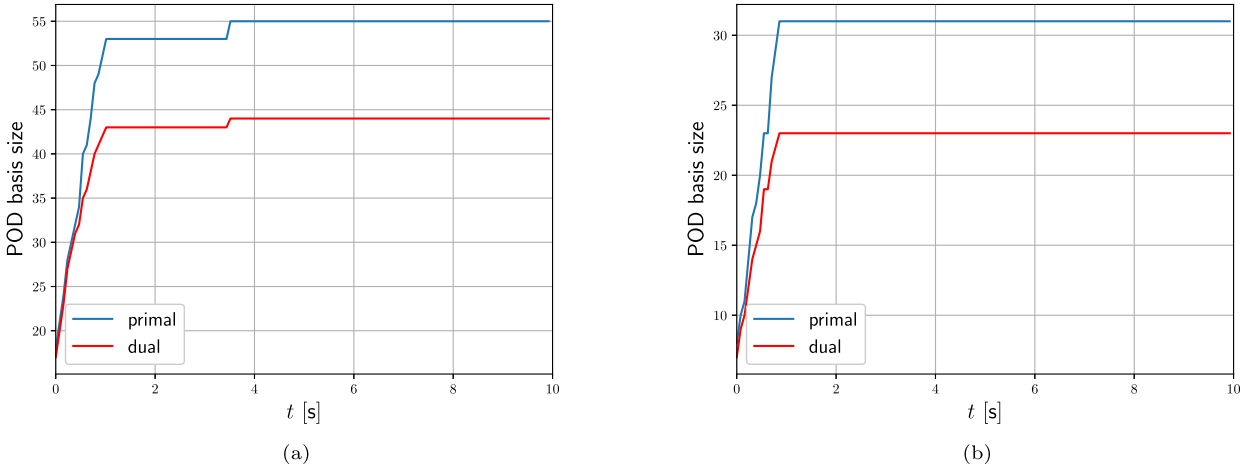


Fig. 8. Temporal evolution of the reduced basis size for a relative error tolerance of (a) 1% and (b) 10% for the 2+1D heat equation.

the amount of FOM solves and the POD bases w.r.t. the given tolerance. Furthermore, the estimator predicts the relationship of the error to the tolerance in approximately $94-99\%$ of the cases right with most of the incorrect predictions being overestimations. An exception exists for tol $= 10\%$ where a drop of 5% in the prediction capability can be observed indicating the dual basis is too small to accurately estimate the error. An adapted tolerance for the information content of the dual basis could counteract that problem. Nevertheless, the obtained reduced goal functional still meets the error tolerance. The largest difference with the linear case holds the evaluation of the effectivity index. We observe that in contrast to the linear case, the effectivity indices show larger fluctuations around 1, which have been expected due to the nonlinear goal functional. However, the effectivity indices are still in an acceptable range yielding good results. Finally, we observe that for a large tolerance of 10% we have a few mispredictions, i.e. on 79 slabs the true error is greater than the tolerance while the estimated error is smaller than the tolerance, and on 28 slabs the error estimator is greater than the tolerance while the true error is smaller than the tolerance. Additionally, for this tolerance we have 17 slabs on which both true and estimated errors are larger than the tolerance. This decay of MORe DWR performance can be explained by the replacement of the fine dual solution z^{fine} in the DWR error estimator by the coarse dual solution z^{coarse}. If we make the POD bases for the primal and dual problems too small, then this approximation might cause additional errors and lead to a worse performance of our method. Next, Fig. 8 sketches the incremental nature of the MORe DWR approach. The on-the-fly basis generation is shown by plotting the primal and dual reduced basis size over the time domain and comparing its evolution for the tolerances of 1% and 10%. The results indicate a steep increase of both the primal and dual basis sizes in the first second of the simulation that reflects one round trip of the oscillating heat source through the spatial domain. Again, for a more restrictive tolerance, more basis refinements can be observed, yielding a faster increase in basis sizes. After the first round trip of the heat source, the basis size remains almost unchanged with only one basis enlargement for the tolerance of 1% at around $t = 4$ s. This is grounded in the periodic behavior of the chosen numerical experiment that does not add any further information to the system. Thus, less or no further basis enrichments have to be performed to meet the given error tolerance.

So far, we have used the strategy of replacing the dual problem with an approximate dual problem to avoid unnecessary computations because of the backward time flow as discussed in Section 4.2.2. The above results demonstrate the successful application of this strategy. However, from a theoretical perspective, an inexact dual problem is solved. Hence, we now use the 2+1D heat problem to examine the MORe DWR method conducted on a single parent slab by setting $K = 1$ and thus using the exact dual problem. For this, Fig. 9 depicts the corresponding temporal evolution of the time interval-wise error estimator evaluation for the error tolerances 0.1%, 1.0% and 10.0%. The error behavior for all tolerances is very similar. In each instance, the actual error and its estimate coincide, and are generally slightly below the specified tolerance. But, several small overshoots of the real error where the estimate stays below the tolerance can be observed. Compared to the multi parent slab counterpart above, more of these mispredictions happen. Nevertheless, the comparison of the effectivity indices of the one parent slab with the multiple parent slab scenarios shows an improvement if using a single slab. More precisely, the effectivity indices change from 0.752 to 1.2148, from 0.2771 to 1.1846 and from 1.5474 to 1.3907

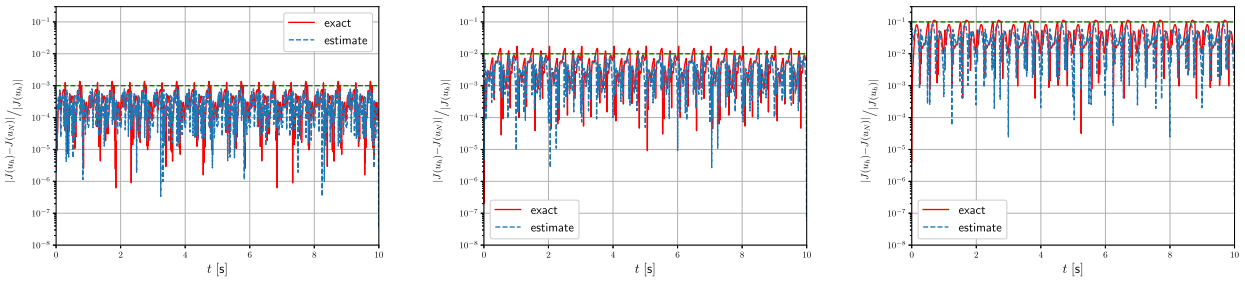


Fig. 9. Temporal evolution of the time interval-wise error estimator compared to the true error for (a) tol = 0.1%, (b) tol = 1.0% and (c) tol = 10.0% on a single parent slab ($K = 1$) for the 2+1D heat equation.

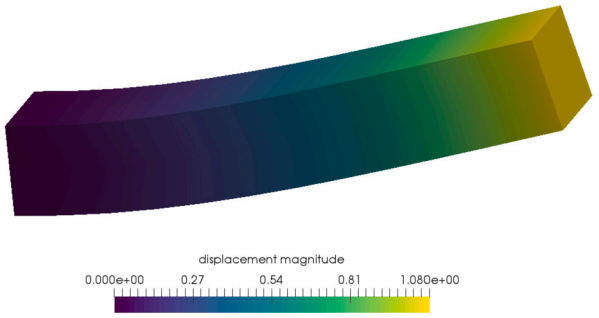


Fig. 10. Full-order solution snapshot at t = 5.75 for the elastodynamics equation.

for the relative tolerances 0.1%, 1.0% and 10.0% respectively. In summary, using the fully backward dual problem yields improved effectivity indices but also suffers from small mispredictions which could be counterbalanced by reducing the MORe DWR tolerance slightly below the real desired error tolerance. Besides the error estimation capability, also the resulting speed-ups are of interest. If using a single parent slab, the MORe DWR method obtains the speed-up factors 2.6, 6.3 and 20.2 for the relative tolerances 0.1%, 1.0% and 10.0% respectively. Compared to Table 2, this shows that the usage of multiple parent slabs and thus approximating the dual problem yields better speed-ups while retaining similar quality in error control.

5.3. 3+1D elastodynamics equation

In the third numerical experiment, we choose Formulation 2.2 and investigate the method on a 3+1D elastodynamics problem. We consider a rectangular beam spanning the spatial domain $\Omega = (0,6) \times (0,1) \times (0,1)$. Further, the temporal domain $I = (0,40)$ is regarded. We induce an oscillation in the vertical direction by defining a force $f(t,x)$ acting on the upper boundary of the beam $\Gamma_{\text{up}} = (0,6) \times (0,1) \times \{x_3 = 1\}$. In the first part of the experiment the beam is lifted up by means of the acting force as shown in Fig. 10. Thereafter, the force is slowly eliminated such that the beam begins to swing.

For this, we use

$$g(t) := \begin{cases} f_{\max} \frac{t}{t_1} & x_3 = 1 \ \wedge \ t \leq t_1, \\ f_{\max} \left(1 - \frac{t - t_1}{t_2 - t_1}\right) & x_3 = 1 \ \wedge \ t_1 < t \leq t_2, \\ 0 & \text{else}, \end{cases}$$

with maximal acting force $f_{\max} = 0.5$ and $t_1 = 5$ and $t_2 = 6$ being the time points until the force increases or decreases, respectively. Together with the beam being clamped at the boundary $\Gamma_{\text{clamped}} = \{x_1 = 0\} \times (0,1) \times (0,1)$ this yields the boundary conditions in Section 2.1.2

$$\begin{aligned} u &= 0 \qquad \text{in } I \times \Gamma_{\text{clamped}}, \\ v &= 0 \qquad \text{in } I \times \Gamma_{\text{clamped}}, \\ \sigma(u) \cdot n &= 0 \qquad \text{in } I \times \partial\Omega \setminus (\Gamma_{\text{clamped}} \cup \Gamma_{\text{up}}), \\ \sigma(u) \cdot n &= g(t) \qquad \text{in } I \times \Gamma_{up}. \end{aligned}$$

Furthermore, the homogeneous initial conditions are given by

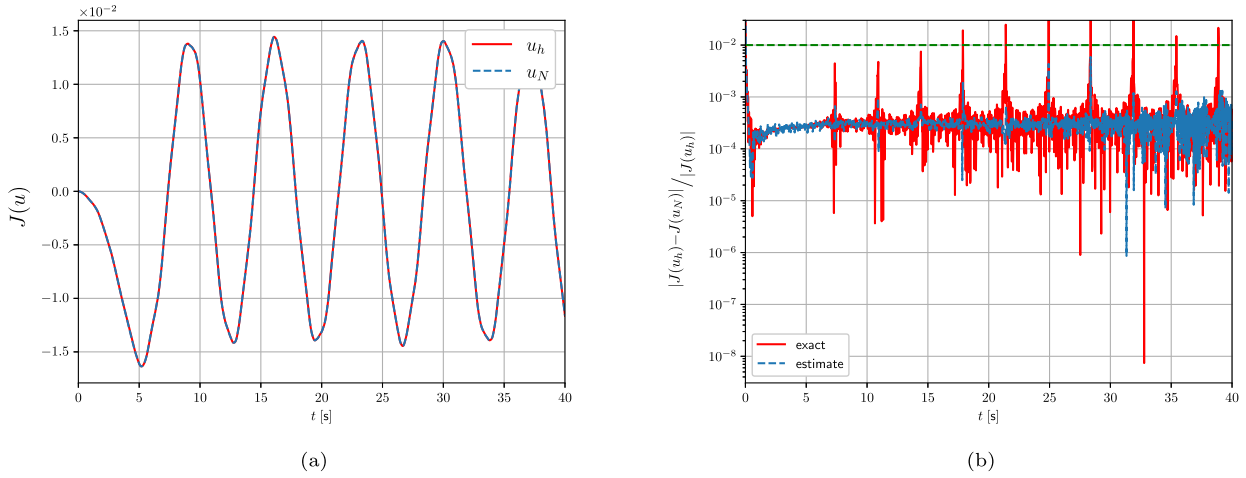


Fig. 11. Temporal evolution of (a) the goal functional and (b) the time interval-wise error estimator compared to the true error for the 3+1D elastodynamics equation.

$$u(0) = 0 \qquad \text{in } \Omega,$$
$$v(0) = 0 \qquad \text{in } \Omega.$$

We choose the time-averaged stress acting on the clamped boundary Γ_{clamped} denoted by $J(u) := \frac{1}{40}\int_0^{40}\int_{\Gamma_{\text{clamped}}} \sigma(u(t,x))\cdot n \,\mathrm{d}x\,\mathrm{d}t$ as the goal functional. For the reduced-order model, we decide that the primal and dual reduced bases have to preserve $\varepsilon = 1 - 10^{-11}$ of the information. Again, we resort to the relative error estimate $\eta^{rel}_{N,S^l_{P_k}}$ and allow errors up to a tolerance of 1%. The full-order model is characterized by $n = 1,875$ and $q = 4,800$ DoFs in space and time, respectively. This gives us a total of $n \cdot q = 9,000,000$ space-time degrees of freedom. The resulting full-order system is solved in $62.7\,s$. For the sake of concentrating on reasonable run times, we employ a not too fine spatial discretization. This ensures that the use of a direct solver is still feasible with respect to memory consumption such that the numerical tests can be reproduced on a desktop computer. It is important to emphasize that the central objective of this work is not the enhancement of preconditioners for the full-order model, but rather the development of an efficient adaptive reduced-order model. Note that similar to the 2+1D heat equation, the temporal discretization might be finer than necessary.

Further, the temporal domain is split up to $M = 1,600$ time slabs. For the incremental ROM, we choose a total amount of $K = 50$ parent-slabs on which the slabs are evenly distributed, i.e. $L = 32$.

We compare the time trajectories of the goal functional restricted to each time slab for the full-order space-time solution u_h and the reduced-order space-time solution u_N in Fig. 11a. The results show that both trajectories are indistinguishable from each other and the oscillating behavior can be mimicked by the reduced goal functional. Furthermore, the good approximation quality can also be observed when regarding the time-averaged goal functional. We obtain $J(u_h) = -1.3100$ and $J(u_N) = -1.3097$ yielding a relative error of $\eta_{max} = 0.0023\%$ which is smaller than the desired error tolerance of 1%.

In Fig. 11b, we plot the exact temporal errors and their estimation on each slab for comparison. For illustration purposes, we indicate the error tolerance of 1% in this plot. The results show that both quantities are on average in the same order of magnitude while the standard deviation of the real error appears to be larger. Thus, there exist spikes in the error that are not captured by the error estimation. Most of the time, this has no consequence but on one slab the error tolerance is exceeded slightly. In order to investigate the violation of the error tolerance, we present in Table 3 simulation results for a range of error tolerances. Therefore, we show the relative error, the prediction measures and the effectivity indices to examine the error estimation. Further, the computational speedup, total number of FOM solves and the POD basis sizes for the primal and dual problems are displayed. For definitions of these quantities, we refer to Section 5.1. We can observe that while most of the mispredictions are poor underestimations of the error, there are only a few of them. In addition, except for a relative tolerance of 2% the effectivity indices are near to the optimum of 1 and the relative errors meet the tolerance in all scenarios. Additionally, when reviewing the performance measurement, we can determine differences to the previous heat problems. The resulting speedups as well as the FOM solves are near constant for all tolerances. We also observe that the total amount of FOM solves and the size of the POD bases are not monotonically decreasing w.r.t. the error tolerance like in the heat equation setting. A reason for this behavior can be assigned to the behavior of the error itself. Using a smaller tolerance can lead to more reduced basis enrichment early on. Larger tolerances lead to smaller initial reduced basis and so errors further on due to the small basis size, which is compensated by enlarging the basis in a later stage.

Finally, the incremental nature of our MORe DWR approach is depicted in Fig. 12. We allow insights into the on-the-fly basis generation by plotting the primal and dual reduced basis size over the time domain and comparing its evolution for the tolerances of 1% and 10%. Similar to the previous scenarios, we observe a steep increase in both the primal and dual basis sizes at the beginning of the simulation. In addition, we see further changes in the reduced basis sizes in the latter course of the simulation. While in the case of a tolerance of 10% the basis sizes only experience one additional ascent and then remain constant, in the case of a relative

Table 3
Incremental reduced-order modeling summary for the 3+1D elastodynamics equation depending on the tolerance in the goal functional.

Tolerance	Relative error	Speedup	FOM solves	Basis size	Prediction	Effectivity
0.1%	0.0067%	6.5	50	118 | 96	14 | 0 | 1 | 1585	1.0346
1%	0.0229%	5.4	56	108 | 111	8 | 0 | 1 | 1591	0.9987
2%	0.0025%	6.2	46	117 | 99	2 | 0 | 0 | 1598	0.5741
5%	0.0020%	4.7	52	118 | 111	1 | 0 | 0 | 1599	1.0123
10%	0.0511%	6.0	46	116 | 111	10 | 0 | 0 | 1590	0.9985

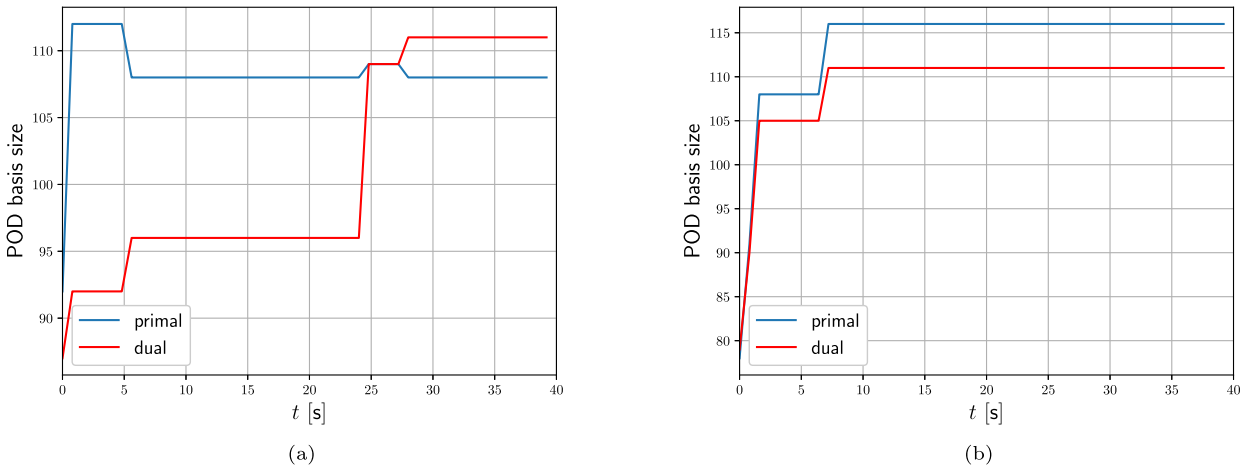


Fig. 12. Temporal evolution of the reduced basis size for a relative error tolerance of (a) 1% and (b) 10% for the 3+1D elastodynamics equation.

tolerance of 1%, more dynamic basis size adjustments can be observed. Specifically, the primal basis size decreases shortly after the steep initial increase, but then remains approximately constant. In contrast, the dual basis size remains stable after the initial enrichment for a while but then again increases steeply after around 24 s.

6. Conclusion and outlook

In this work, we proposed a novel incremental POD-ROM method with on-the-fly basis enrichment based on space-time DWR error estimates for linear PDEs, namely the heat equation and elastodynamics, and linear and nonlinear time-averaged goal functionals. This methodology can be applied to a wide class of problems since its efficiency has been demonstrated. The effectivity indices for linear problems are almost exactly one, which makes the error estimates reliable in practice, and for nonlinear goal functionals, we had also good effectivity indices. For nonlinear PDEs and goal functionals possibly full DWR is needed. Additionally, we had speedups of up to 50, while the error between the FOM and the ROM solution remained within our prescribed tolerance. Consequently, e.g. the expensive high-fidelity computations in the offline stage of the reduced basis method could be replaced by our incremental POD method. A future aspect is time point error control using J_2 at T (or some intermediate $t^* \in (0,T]$) in (10). The extreme case are end time values $J_2(u(T))$ within our proposed incremental approach since such goal functionals may nonetheless need dual sensitivity measures on the entire time interval I, which is a priori jeopardized by our proposed incremental approach. Nevertheless, if conducted on a single parent slab, the MORe DWR method yields satisfying findings of the adjoint measures on the entire time interval I. This gives hope that J_2 cases can be treated as well using a single parent slab, which we however refer to ongoing work. Another interesting topic for future work would be the extension of this method to dynamical, adaptive spatial meshes to further speed up the computations.

CRediT authorship contribution statement

Hendrik Fischer: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Julian Roth:** Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Thomas Wick:** Conceptualization, Formal analysis, Funding acquisition, Resources, Supervision, Writing – review & editing. **Ludovic Chamoin:** Conceptualization, Formal analysis, Funding acquisition, Supervision, Writing – review & editing. **Amelie Fau:** Conceptualization, Formal analysis, Funding acquisition, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors acknowledge the funding of the German Research Foundation (DFG) within the framework of the International Research Training Group on Computational Mechanics Techniques in High Dimensions GRK 2657 under Grant Number 433082294. In addition, we thank Hendrik Geisler (Leibniz University Hannover, GRK 2657) for fruitful discussions and comments. The support of the French-German University through the French-German Doctoral college "Sophisticated Numerical and Testing Approaches" (CDFA-DFDK 19-04) is also acknowledged. We thank the anonymous reviewers for their extensive questions that helped to improve the manuscript.

Appendix A. Space-time linear system and dG(r) time-stepping formulation for elastodynamics

The space-time discretization of the elastodynamics equation on a slab with a single temporal element and a dG(r) in time discretization is discussed in this appendix. Using the fully discrete variational formulation 2.4 of elastodynamics, we arrive at the linear equation system

$$\left[C_k \otimes M_h + M_k \otimes K_h + D_k^1 \otimes M_h\right] U_m = F_m + \left[D_k^2 \otimes M_h\right] U_{m-1}, \tag{A.1}$$

where we use the spatial matrices

$$M_h = \left\{ (\varphi_h^{v,(j)}, \varphi_h^{u,(i)}) + (\varphi_h^{u,(j)}, \varphi_h^{v,(i)}) \right\}_{i,j=1}^{\#\mathrm{DoFs}(\mathcal{T}_h)},$$

$$K_h = \left\{ (\sigma(\varphi_h^{u,(j)}), \nabla_x \varphi_h^{u,(i)}) + (\varphi_h^{v,(j)}, \varphi_h^{v,(i)}) \right\}_{i,j=1}^{\#\mathrm{DoFs}(\mathcal{T}_h)},$$

and the temporal matrices

$$M_k = \left\{ \int_{I_m} \varphi_k^{(j)} \cdot \varphi_k^{(i)} \, \mathrm{d}t \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)},$$

$$C_k = \left\{ \int_{I_m} \partial_t \varphi_k^{(j)} \cdot \varphi_k^{(i)} \, \mathrm{d}t \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)},$$

$$D_k^1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & & \\ \vdots & & \ddots & \\ 0 & & & 0 \end{pmatrix}, \qquad D_k^2 = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ & & 0 & 0 \\ & \iddots & & \vdots \\ 0 & & & 0 \end{pmatrix}.$$

Here, the solution vector U_m for the dG(r) method in time with temporal quadrature points $t_1, \dots, t_{r+1}$ is given by

$$U_m = \begin{pmatrix} U_m(t_1) \\ \vdots \\ U_m(t_{r+1}) \end{pmatrix} = \begin{pmatrix} u_m(t_1) \\ v_m(t_1) \\ \vdots \\ u_m(t_{r+1}) \\ v_m(t_{r+1}) \end{pmatrix}.$$

To derive the dG(r) time-stepping formulation, we now only need to evaluate the temporal matrices M_k and C_k by integrating over $(0, k)$, where $k := t_m - t_{m-1}$ is the time step size, and by plugging in the dG$-Q^r$ basis functions on $(0, k)$, which coincide with the Q^r basis functions since we only have one single element and use Gauss-Lobatto quadrature.

A.1. dG(1) *formulation of elastodynamics*

By inserting the basis functions $\varphi_k^{(1)} = 1 - \frac{t}{k}, \varphi_k^{(2)} = \frac{t}{k}$ into the temporal matrices M_k and C_k we get

$$\int_0^k \varphi_k^{(1)} \cdot \varphi_k^{(1)} \,\mathrm{d}t = \int_0^k \left(1 - \frac{t}{k}\right)^2 \,\mathrm{d}t = \int_0^k 1 - \frac{2t}{k} + \frac{t^2}{k^2} \,\mathrm{d}t = \frac{k}{3} = \int_0^k \varphi_k^{(2)} \cdot \varphi_k^{(2)} \,\mathrm{d}t,$$

$$\int_0^k \varphi_k^{(1)} \cdot \varphi_k^{(2)} \,\mathrm{d}t = \int_0^k \varphi_k^{(2)} \cdot \varphi_k^{(1)} \,\mathrm{d}t = \int_0^k \left(1 - \frac{t}{k}\right) \cdot \frac{t}{k} \,\mathrm{d}t = \int_0^k \frac{t}{k} - \frac{t^2}{k^2} \,\mathrm{d}t = \frac{k}{6},$$

as well as

$$\int_0^k \partial_t \varphi_k^{(1)} \cdot \varphi_k^{(2)} \,\mathrm{d}t = \int_0^k \partial_t \left(1 - \frac{t}{k}\right) \cdot \frac{t}{k} \,\mathrm{d}t = \int_0^k -\frac{t}{k^2} \,\mathrm{d}t = -\frac{1}{2} = \int_0^k \partial_t \varphi_k^{(1)} \cdot \varphi_k^{(1)} \,\mathrm{d}t,$$

$$\int_0^k \partial_t \varphi_k^{(2)} \cdot \varphi_k^{(2)} \,\mathrm{d}t = \int_0^k \partial_t \left(\frac{t}{k}\right) \cdot \frac{t}{k} \,\mathrm{d}t = \int_0^k \frac{t}{k^2} \,\mathrm{d}t = \frac{1}{2} = \int_0^k \partial_t \varphi_k^{(2)} \cdot \varphi_k^{(1)} \,\mathrm{d}t.$$

Consequently, the dG(1) time-stepping formulation for elastodynamics reads

$$\left[\frac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \otimes M_h + \frac{k}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \otimes K_h\right] \begin{pmatrix} U_m(t_{m-1}) \\ U_m(t_m) \end{pmatrix} = \begin{pmatrix} F_m(t_{m-1}) + U_{m-1}(t_{m-1}) M_h \\ F_m(t_m) \end{pmatrix},$$

where we use the fact that the temporal quadrature points for dG(1) are t_{m-1} and t_m.

A.2. dG(2) *formulation of elastodynamics*

Repeating the procedure from A.1 with quadratic basis functions, we arrive at the dG(2) time-stepping formulation for elastodynamics

$$\left[\frac{1}{6}\begin{pmatrix} 3 & 4 & -1 \\ -4 & 0 & 4 \\ 1 & -4 & 3 \end{pmatrix} \otimes M_h + \frac{k}{30}\begin{pmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{pmatrix} \otimes K_h\right] \begin{pmatrix} U_m(t_{m-1}) \\ U_m(t_{m-\frac{1}{2}}) \\ U_m(t_m) \end{pmatrix} = \begin{pmatrix} F_m(t_{m-1}) + U_{m-1}(t_{m-1}) M_h \\ F_m(t_{m-\frac{1}{2}}) \\ F_m(t_m) \end{pmatrix},$$

where we use the fact that the temporal quadrature points for dG(2) are t_{m-1}, $t_{m-\frac{1}{2}} := t_{m-1} + \frac{k}{2}$ and t_m.

Remark A.1. The dG(1)and dG(2) formulations can also be found in Section 7.1 and Section 7.2 in [93] for an ODE model.

References

[1] L. Sirovich, Turbulence and the dynamics of coherent structures. I. Coherent structures, Q. Appl. Math. 45 (1987) 561–571.
[2] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems, Numer. Math. 90 (2001) 117–148.
[3] Z. Luo, J. Chen, P. Sun, X. Yang, Finite element formulation based on proper orthogonal decomposition for parabolic equations, Sci. China Ser. A, Math. 52 (2009) 585–596.
[4] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, Int. J. Numer. Methods Eng. 102 (2015) 1136–1161.
[5] S. Volkwein, Lecture notes in "Proper orthogonal decomposition: theory and reduced-order modelling", http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Book.pdf, 2013.
[6] P. Benner, W. Schilders, S. Grivet-Talocia, A. Quarteroni, G. Rozza, L. Miguel Silveira, Model Order Reduction: Volume 2: Snapshot-Based Methods and Algorithms, De Gruyter, 2020.
[7] G. Kerschen, J.-C. Golinval, A.F. Vakakis, L.A. Bergman, The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview, Nonlinear Dyn. 41 (2005) 147–169.
[8] P. Benner, A. Cohen, M. Ohlberger, K. Willcox, Model Reduction and Approximation: Theory and Algorithms, SIAM, Philadelphia, 2015.
[9] M. Abbaszadeh, M. Dehghan, A. Khodadadian, N. Noii, C. Heitzinger, T. Wick, A reduced-order variational multiscale interpolating element free Galerkin technique based on proper orthogonal decomposition for solving Navier–Stokes equations coupled with a heat transfer equation: nonstationary incompressible Boussinesq equations, J. Comput. Phys. 426 (2021) 109875.
[10] Q. Wang, N. Ripamonti, J.S. Hesthaven, Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism, J. Comput. Phys. 410 (2020) 109402.
[11] M. Girfoglio, A. Quaini, G. Rozza, A POD-Galerkin reduced order model for the Navier–Stokes equations in stream function-vorticity formulation, Comput. Fluids 244 (2022) 105536.
[12] R. Becker, R. Rannacher, A feed-back approach to error control in finite element methods: basic analysis and examples, East-West J. Numer. Math. 4 (1996) 237–264.
[13] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta Numer. 10 (2001) 1–102.
[14] W. Bangerth, R. Rannacher, Adaptive Finite Element Methods for Differential Equations, Birkhäuser Verlag, 2003.

[15] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Introduction to adaptive methods for differential equations, Acta Numer. 4 (1995) 105–158.
[16] J.T. Oden, Adaptive multiscale predictive modelling, Acta Numer. 27 (2018) 353–450.
[17] M. Schmich, B. Vexler, Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations, SIAM J. Sci. Comput. 30 (2008) 369–393.
[18] M. Schmich, Adaptive Finite Element Methods for Computing Nonstationary Incompressible Flows, Ph.D. thesis, Heidelberg University, 2009.
[19] M. Besier, R. Rannacher, Goal-oriented space-time adaptivity in the finite element Galerkin method for the computation of nonstationary incompressible flow, Int. J. Numer. Methods Fluids 70 (2012) 1139–1166.
[20] B. Endtmayer, U. Langer, A. Schafelner, Goal-oriented adaptive space-time finite element methods for regularized parabolic p-Laplace problems, https://doi.org/10.48550/ARXIV.2306.07167, 2023.
[21] J.P. Thiele, T. Wick, Numerical modeling and open-source implementation of variational partition-of-unity localizations of space-time dual-weighted residual estimators for parabolic problems, J. Sci. Comput. (2024), accepted.
[22] J. Roth, J.P. Thiele, U. Köcher, T. Wick, Tensor-Product Space-Time Goal-Oriented Error Control and Adaptivity With Partition-of-Unity Dual-Weighted Residuals for Nonstationary Flow Problems, Comput. Methods Appl. Math. 24 (2024) 185–214.
[23] A. Rademacher, Adaptive finite element methods for nonlinear hyperbolic problems of second order, Ph.D. thesis, Technische Universität Dortmund, 2009.
[24] W. Bangerth, M. Geiger, R. Rannacher, Adaptive Galerkin finite element methods for the wave equation, Comput. Methods Appl. Math. 10 (2010) 3–48.
[25] U. Langer, O. Steinbach (Eds.), Space-Time Methods: Application to Partial Differential Equations, Radon Series on Computational and Applied Mathematics, vol. 25, de Gruyter, Berlin, 2019.
[26] A. Schafelner, Space-time finite element methods, Ph.D. thesis, Johannes Kepler University Linz, 2021.
[27] T. Hughes, G. Hulbert, Space-time finite element methods for elastodynamics: formulations and error estimates, Comput. Methods Appl. Mech. Eng. 66 (1988) 339–363.
[28] W. Dörfler, M. Hochbruck, J. Köhler, A. Rieder, R. Schnaubelt, C. Wieners, Wave Phenomena: Mathematical Analysis and Numerical Approximation, Oberwolfach Seminars, Birkhäuser, Cham, 2022.
[29] M. Loveland, E. Valseth, M. Lukac, C. Dawson, Extending FEniCS to work in higher dimensions using tensor product finite elements, J. Comput. Sci. 64 (2022) 101831.
[30] C. Lehrenfeld, F. Heimann, J. Preuß, H. von Wahl, 'ngsxfem': add-on to NGSolve for geometrically unfitted finite element discretizations, J. Open Sour. Softw. 6 (2021) 3237.
[31] J. Preuß, Higher order unfitted isoparametric space-time FEM on moving domains, Master's thesis, Georg-August Universität Göttingen, 2018.
[32] Y. Choi, P.N. Brown, B. Arrighi, R. Anderson, Space-time reduced order model for large-scale linear dynamical systems with application to Boltzmann transport problems, J. Comput. Phys. 424 (2019) 109845.
[33] Y. Kim, K. Wang, Y. Choi, Efficient space–time reduced order model for linear dynamical systems in Python using less than 120 lines of code, Mathematics 9 (2021) 1690.
[34] R. Tenderini, N. Mueller, S. Deparis, Space-time reduced basis methods for parametrized unsteady Stokes equations, https://doi.org/10.48550/ARXIV.2206.12198, 2022.
[35] F. Ekre, F. Larsson, K. Runesson, R. Jänicke, A posteriori error estimation for numerical model reduction in computational homogenization of porous media, Int. J. Numer. Methods Eng. 121 (2020) 5350–5380.
[36] Y.S. Shimizu, E.J. Parish, Windowed space–time least-squares Petrov–Galerkin model order reduction for nonlinear dynamical systems, Comput. Methods Appl. Mech. Eng. 386 (2021) 114050.
[37] F. Zoccolan, M. Strazzullo, G. Rozza, A streamline upwind Petrov-Galerkin reduced order method for advection-dominated partial differential equations under optimal control, https://doi.org/10.48550/ARXIV.2301.01973, 2023.
[38] M. Bernreuther, S. Volkwein, An adaptive certified space-time reduced basis method for nonsmooth parabolic partial differential equations, https://doi.org/10.48550/ARXIV.2212.13744, 2022.
[39] K. Steih, K. Urban, Space-time reduced basis methods for time-periodic partial differential equations, IFAC Proc. Vol. 45 (2012) 710–715.
[40] K. Urban, A. Patera, An improved error bound for reduced basis approximation of linear parabolic problems, Math. Comput. 83 (2014) 1599–1615.
[41] M. Yano, A.T. Patera, K. Urban, A space-time hp-interpolation-based certified reduced basis method for Burgers' equation, Math. Models Methods Appl. Sci. 24 (2014) 1903–1935.
[42] M. Yano, A space-time Petrov–Galerkin certified reduced basis method: application to the Boussinesq equations, SIAM J. Sci. Comput. 36 (2014) A232–A266.
[43] M. Yano, Goal-oriented model reduction of parametrized nonlinear partial differential equations: application to aerodynamics, Int. J. Numer. Methods Eng. 121 (2020) 5200–5226.
[44] M.K. Sleeman, M. Yano, Goal-oriented model reduction for parametrized time-dependent nonlinear partial differential equations, Comput. Methods Appl. Mech. Eng. 388 (2022) 114206.
[45] M. Meyer, H.G. Matthies, Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods, Comput. Mech. 31 (2003) 179–191.
[46] F. Chinesta, P. Ladevèze, E. Cueto, A short review in model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. 18 (2011) 395–404.
[47] S. Perotto, A. Zilio, Space–time adaptive hierarchical model reduction for parabolic equations, Adv. Model. Simul. Eng. Sci. 2 (2015).
[48] D. Baroli, C.M. Cova, S. Perotto, L. Sala, A. Veneziani, Hi-POD solution of parametrized fluid dynamics problems: preliminary results, Research Report, MOX, Dipartimento di Matematica, Politecnico di Milano, 2016.
[49] S. Perotto, M.G. Carlino, F. Ballarin, Model reduction by separation of variables: a comparison between hierarchical model reduction and proper generalized decomposition, in: S.J. Sherwin, D. Moxey, J. Peiró, P.E. Vincent, C. Schwab (Eds.), Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018, vol. 134, Springer, 2020, pp. 61–77.
[50] M. Brand, Incremental singular value decomposition of uncertain data with missing values, in: European Conference on Computer Vision, Springer, 2002, pp. 707–720.
[51] M. Brand, Fast low-rank modifications of the thin singular value decomposition, Linear Algebra Appl. 415 (2006) 20–30.
[52] N. Kühl, H. Fischer, M. Hinze, T. Rung, An incremental singular value decomposition approach for large-scale spatially parallel & distributed but temporally serial data–applied to technical flows, Comput. Phys. Commun. 296 (2024) 109022.
[53] G.M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, K. Chand, Limited-memory adaptive snapshot selection for proper orthogonal decomposition, Int. J. Numer. Methods Eng. 109 (2017) 198–217.
[54] P. Phalippou, S. Bouabdallah, P. Breitkopf, P. Villon, M. Zarroug, 'On-the-fly' snapshots selection for Proper Orthogonal Decomposition with application to nonlinear dynamics, Comput. Methods Appl. Mech. Eng. 367 (2020) 113120.
[55] C. Himpe, T. Leibner, S. Rave, Hierarchical approximate proper orthogonal decomposition, SIAM J. Sci. Comput. 40 (2018) A3267–A3292.
[56] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, D. Wells, The deal.II finite element library: design, features, and insights, Comput. Math. Appl. 81 (2021) 407–422.
[57] D. Arndt, W. Bangerth, M. Feder, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Sticko, B. Turcksin, D. Wells, The `deal.II` library, version 9.4, J. Numer. Math. 30 (2022) 231–246.

[58] R. Dautray, J.-L. Lions, Mathematical Analysis and Numerical Methods for Science and Technology: Volume 5 Evolution Problems I, Springer-Verlag, Berlin-Heidelberg, 2000.
[59] J. Wloka, C. Thomas, M. Thomas, Partial Differential Equations, Cambridge University Press, 1987.
[60] L.C. Evans, Partial Differential Equations, American Mathematical Society, 2010.
[61] J. Thiele, Error-controlled space-time finite elements, algorithms, and implementations for nonstationary problems, Ph.D. thesis, Leibniz University Hannover, 2023.
[62] D. Di Pietro, A. Ern, Mathematical Aspects of Discontinuous Galerkin Methods, Mathématiques et Applications, Springer, Berlin, Heidelberg, 2011.
[63] M. Delfour, W. Hager, F. Trochu, Discontinuous Galerkin methods for ordinary differential equations, Math. Comput. 36 (1981) 455–473.
[64] R. Rannacher, Numerik 1: Numerik gewöhnlicher Differentialgleichungen, Heidelberg University Publishing, 2017.
[65] K. Eriksson, K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Computational Differential Equations, Computational Differential Equations, Cambridge University Press, 1996.
[66] B. Haasdonk, M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations, ESAIM: Math. Model. Numer. Anal. 42 (2008) 277–302.
[67] J.S. Hesthaven, G. Rozza, B. Stamm, Parametrized differential equations, in: J.S. Hesthaven, G. Rozza, B. Stamm (Eds.), Certified Reduced Basis Methods for Parametrized Partial Differential Equations, SpringerBriefs in Mathematics, Springer International Publishing, 2016, pp. 15–25.
[68] L. Bertagna, A. Veneziani, A model reduction approach for the variational estimation of vascular compliance by solving an inverse fluid–structure interaction problem, Inverse Probl. 30 (2014) 055006.
[69] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, Model Order Reduction in Fluid Dynamics: Challenges and Perspectives, Reduced Order Methods for Modeling and Computational Reduction, vol. 9, Springer International Publishing, 2014, pp. 235–273.
[70] B. Haasdonk, Chapter 2: reduced basis methods for parametrized PDEs—a tutorial introduction for stationary and instationary problems, in: Model Reduction and Approximation, Society for Industrial and Applied Mathematics, 2017, pp. 65–136.
[71] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Arch. Comput. Methods Eng. 15 (2008) 229–275.
[72] N.-C. Nguyen, G. Rozza, A.T. Patera, Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation, Calcolo 46 (2009) 157–185.
[73] G. Rozza, Shape design by optimal flow control and reduced basis techniques, Ph.D. thesis, 2005.
[74] A. Kolmogoroff, Über die beste Annäherung von Funktionen einer gegebenen Funktionenklasse, Ann. Math. (1936) 107–110.
[75] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. 40 (2002) 492–515.
[76] S.S. Ravindran, A reduced-order approach for optimal control of fluids using proper orthogonal decomposition, Int. J. Numer. Methods Fluids 34 (2000) 425–448.
[77] K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, AIAA J. 40 (2002) 2323–2330.
[78] E.A. Christensen, M. Brøns, J.N. Sørensen, Evaluation of proper orthogonal decomposition–based decomposition techniques applied to parameter-dependent nonturbulent flows, SIAM J. Sci. Comput. 21 (1999) 1419–1434.
[79] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Comput. Methods Appl. Mech. Eng. 196 (2007) 1030–1047.
[80] A. Caiazzo, T. Iliescu, V. John, S. Schyschlowa, A numerical investigation of velocity–pressure reduced order models for incompressible flows, J. Comput. Phys. 259 (2014) 598–616.
[81] J. Baiges, R. Codina, S. Idelsohn, Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 72 (2013) 1219–1243.
[82] M. Gubisch, S. Volkwein, Chapter 1: proper orthogonal decomposition for linear-quadratic optimal control: theory and algorithms, in: Model Reduction and Approximation, Society for Industrial and Applied Mathematics, 2017, pp. 3–63.
[83] C. Gräßle, M. Hinze, POD reduced-order modeling for evolution equations utilizing arbitrary finite element discretizations, Adv. Comput. Math. 44 (2018) 1941–1978.
[84] B. Endtmayer, Multi-goal oriented a posteriori error estimates for nonlinear partial differential equations, Ph.D. thesis, Johannes Kepler University Linz, 2021.
[85] B. Endtmayer, U. Langer, T. Wick, Two-side a posteriori error estimates for the dual-weighted residual method, SIAM J. Sci. Comput. 42 (2020) A371–A394.
[86] B. Endtmayer, U. Langer, T. Wick, Multigoal-oriented error estimates for non-linear problems, J. Numer. Math. 27 (2019) 215–236.
[87] H. Fareed, J.R. Singler, Y. Zhang, J. Shen, Incremental proper orthogonal decomposition for PDE simulation data, Comput. Math. Appl. 75 (2018) 1942–1960.
[88] C. Bach, D. Ceglia, L. Song, F. Duddeck, Randomized low-rank approximation methods for projection-based model order reduction of large nonlinear dynamical problems, Int. J. Numer. Methods Eng. 118 (2019) 209–241.
[89] H. Fareed, J.R. Singler, A note on incremental POD algorithms for continuous time data, Appl. Numer. Math. 144 (2019) 223–233.
[90] Y. Zhang, An answer to an open question in the incremental SVD, https://doi.org/10.48550/ARXIV.2204.05398, 2022.
[91] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (2020) 357–362.
[92] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: fundamental algorithms for scientific computing in Python, Nat. Methods 17 (2020) 261–272.
[93] T. Richter, A. Springer, B. Vexler, Efficient numerical realization of discontinuous Galerkin methods for temporal discretization of parabolic problems, Numer. Math. 124 (2013) 151–182.